GE Fanuc IC695ALG728

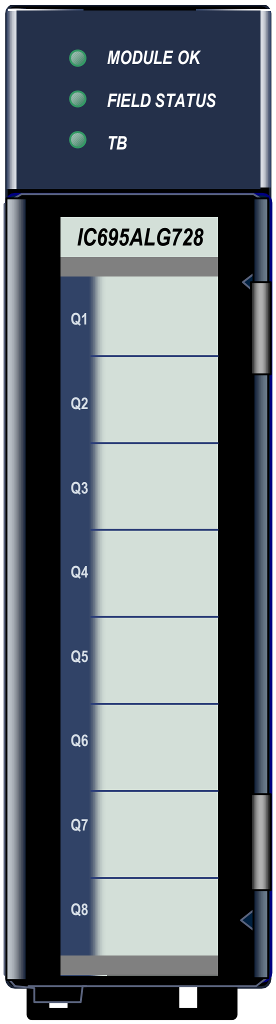
http://www.pdfsupply.com/automation/ge-fanuc/rx3i-pacsystem/IC695ALG728

Rx3i PacSystem

Analog Output HART Module, 8 channels,that is configurable IC695A IC695AL IC695ALG

919-535-3180
sales@pdfsupply.com

Analog Output Module, 8 ChannelVoltage/Current with HART, IC695ALG728



Non-Isolated Analog Voltage/Current Output module IC695ALG728 provides 8 configurable voltage or current output channels with HART version 5.0 communications capability on each channel. The module has two internal HART modems. Four single-ended channels are multiplexed with each HART modem.

Analog channels can be configured for these output ranges:

- Current: 0 to 20mA, 4 to 20mA
- Voltage: +/- 10V, 0 to 10V

Channels that will use HART communications must be configured for the 4-20mA range.

Module Features

- Completely software-configurable, no module jumpers to set
- Individually enable or disable channels
- Clamping and Alarm Limits
- Latching of Alarms
- Configurable output bias
- Rapid channel acquisition times based on filter frequency
- Full autocalibration
- On-board error-checking
- Configurable scaling and offsets per channel
- High alarm, low alarm, high-high alarm, low-low alarm detection and reporting selectable per channel
- Module fault reporting
- Configurable Hold Last State or Output Defaults
- Version 5.0 HART communications

This module can be used with a Box-style (IC694TBB032), Extended Box-style (IC694TBB132), Spring-style (IC694TBS032), or Extended Spring-style (IC694TBS132) Terminal Block. Extended terminal blocks provide the extra shroud depth needed for shielded wiring. Terminal Blocks are ordered separately. The module must be located in an RX3i Universal Backplane. It requires an RX3i CPU with firmware version 3.5 or later. Machine Edition Version 5.0 SP3 Logic Developer-PLC or later must be used for configuration.

Isolated +24 VDC Power

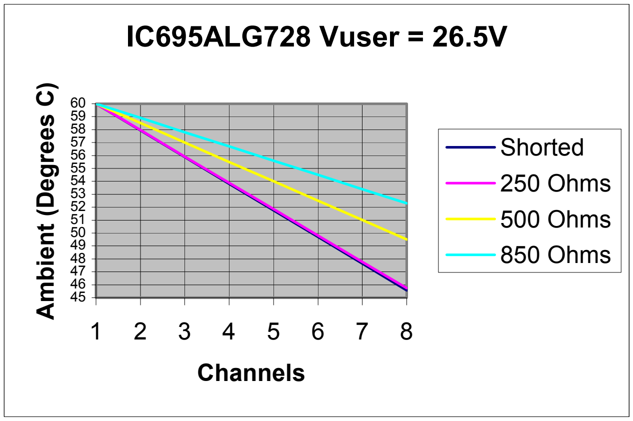
The module must receive its 24 VDC power from an external source. The external source must be connected directly to the module's terminal block. It cannot be connected via the TB1 connector on the RX3i Universal Backplane.

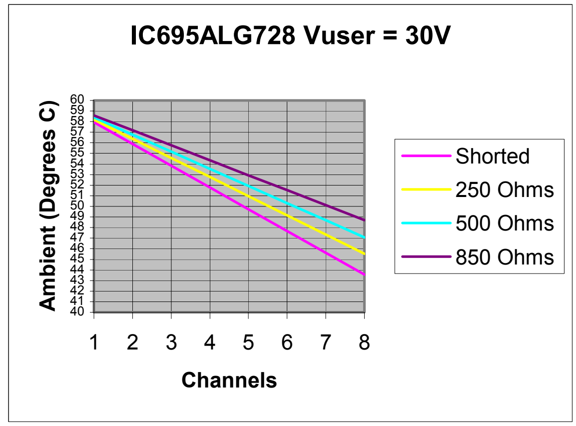
Specifications: IC695ALG728

Output Ranges	Current: 0 to 20mA, 4 to 20mA Voltage: +/- 10V, 0 to 10V
Backplane Power Requirements	380 mA maximum at 3.3V
Power Dissipation within Module (Vuser=24V)	7.25Watts maximum
Thermal Derating	TBD
External Power Supply V_{user}	Voltage Range: +19.2V to +30VDC Current required: 250mA maximum
Resolution	+/-10V: 15.9 bits, 0 to 10V: 14.9 bits, 0 to 20mA: 15.9 bits, 4 to 20mA : 15.6 bits
HART Communications	Version 5.0 HART protocol
Output Data Format	Configurable as floating point IEEE 32 bit or 16-bit integer in a 32-bit field
Analog Update Rate (Determined by I/O scan time, application dependent.)	16mS with HART (approximate, all eight channels) 8mS without HART (approximate, all eight channels)
HART Data Scan Time (in seconds) The HART data scan can consist of up to two acquisition cycles (similar but asynchronous to the analog scan time).. Each cycle includes a specific set of channels. For ALG728: 1-4, 5-8 are separate channel groups. Total HART scan time depends on the number of acquisition cycles in the scan, number of retries, enabling/disabling of slot variables, , and use of pass-thru commands. If slot variables are enabled, update times are doubled.	
1 HART device In group	Each HART Data channel updates every 0.7 second (typical)
2 HART devices in group	Each HART Data channel updates every 1.9 seconds (typical)
Output Overvoltage Protection	Current outputs only: -30V for 60 seconds, +30V for one hour
Calibrated Accuracy	Accurate to within 0.15% of full scale at 25°C Accurate to within 0.30% of full scale at 60°C In the presence of severe RF interference (IC 801-3, 10V/M), accuracy may be degraded to +/-1% FS.
Output Load Reactance	Current: 10µH maximum, Voltage: 1µF maximum
Maximum Output Load	Current: 850 Ohms maximum at V_{user} = 20V Voltage: 2 Kohms minimum
Output Gain Drift	Voltage output: 20ppm per degree C typical Current output: 35ppm per degree C typical
Output Settling Time	Voltage Output: 2ms, 0 to 95% Current output with HART: 70mS, 0 to 95% Current output without HART: 23ms, 0 to 95%
Isolation, Field to Backplane	2550VDC for one second
Maximum Compliance Voltage	V_{user} – 3V (minimum) to V_{user} (maximum)

Refer to Appendix A for product standards and general specifications.

Output Points vs. Temperature, Current Mode





LEDs

The **Module OK** LED indicates module status. The **Field Status** LED indicates whether the external +24 VDC power supply is present and is above the minimum level and whether or not faults are present. All LEDs are powered from the backplane power bus.

LED	***Indicates***
Module OK	ON Green: Module OK and configured. Quick Flashing Green: Module performing powerup sequence. Slow Flashing Green or Amber: Module OK but not configured. OFF: Module is defective or no backplane power present
Field Status	ON Green No faults on any enabled channel, Terminal Block is present, and field power is present. ON Amber and TB Green: Terminal Block is installed, fault on at least one channel, or field power is not present. ON Amber and TB Red: Terminal Block not fully removed, field power still detected. OFF and TB Red: Terminal block not present and no field power is detected.
TB	ON Red: Terminal block not present or not fully seated. See above. ON Green: Terminal block is present. See above. OFF: No backplane power to module.

Configuration Parameters: IC695ALG728

Module Parameters		
Parameter	***Default***	***Description***
Outputs Reference Address	%AQxxxxx	Starting address for the module's output data. This defaults to the next available %AQ block. The format of this data is shown on page 11-33.
Outputs Reference Length	ALG728: 16	The number of words used for the module's output data. This parameter cannot be changed.
Output Command Feedback Reference Address	%AIxxxxx	Stating address for the module's command feedback data. This defaults to the next available %AI address after a non-zero length is configured.
Output Command Feedback Length	0	The number of words used for the module's command feedback data. Length defaults to 0. It can be set to 8 or 16, depending on the module type being configured.
Diagnostic Reference Address	%Ixxxxx	Starting address for the channel diagnostics status data. This defaults to the next available %I block. The format of this data is shown on page 11-34.
Diagnostic Reference Length	0	Read Only. The number of bit reference bits required for the Channel Diagnostics data. Default is 0, which means mapping of Channel Diagnostics is disabled. Change this to a non-zero value to enable Channel Diagnostics mapping. Maximum length is 256 bits.
Module Status Reference Address	%Ixxxxx	Starting address for the module's status data. This defaults to the next available %I block. The format of this data is shown on page 11-35.
Module Status Reference Length	0	Read Only. The number of bits (0 or 32) required for the Module Status data. Default is 0, which means mapping of Module Status data is disabled. Change this to a non-zero value to enable Module Status data mapping.

Continued…

Analog Output Commanded Feedback

The module returns a copy of the analog output data received from CPU in its corresponding channel analog input shared memory. Output Feedback can be monitored to check the values being sent to the channels. The data is in the same scaled format as the output data for each channel. During normal operation this feedback data should match the actual output data after one or more PLC scans of module inputs. During faults, ramping, overrange, and clamping conditions, the analog output data may differ from the commanded output.

<table>
<tr><th colspan="4">Module Parameters</th></tr>
<tr><th>Parameter</th><th>Default</th><th colspan="2">Description</th></tr>
<tr><td>HART Data Scan Control</td><td>No data</td><td colspan="2">Selects whether the CPU will automatically scan from the HART module: no data, changed data only, or all data for each HART-enabled channel. See the below for details of memory usage.
Dynamic Data Only: the first 18 words or 288 bits of HART data per input device.
All Data: all of the HART data (88 words or 1408 bits for each HART input device.</td></tr>
<tr><td>HART Pass-thru Service Options</td><td>Once per two channel scans</td><td colspan="2">Selects whether the module will automatically service a HART pass-through command each 1, 2, or 4 channel scans or only upon change of HART device configuration or if data hasn't been read for 10 seconds (Pass-Thru Only). If Pass-Thru Only is selected, scan data is not available to the application program.</td></tr>
<tr><td>HART Status Reference Address</td><td></td><td colspan="2">Starting address of the HART Status data. Format of this data is shown on page 11-41.</td></tr>
<tr><td>HART Status Reference Length</td><td></td><td colspan="2">Length of the HART Status data; 4 words or 64 bits.</td></tr>
<tr><td>HART Data Reference Address</td><td></td><td colspan="2">Starting address for the module's HART data in %I, %Q, %AI, %AQ, %R, %W, %G, %M, or %T memory. Format of this data is shown on page 11-42.</td></tr>
<tr><td rowspan="5">HART Data Reference Length</td><td rowspan="5">0</td><td colspan="2">Length of the HART data. If Data Scan Control is set to no data, the length is 0. The length is automatically set according to the selection made for HART Data Scan Control:</td></tr>
<tr><td>HART Data Scan Control</td><td>HART Data Reference Length</td></tr>
<tr><td>No Data</td><td>0</td></tr>
<tr><td>Dynamic Data Only</td><td>Highest HART-enabled Channel Number X (18 words or 288 bits)</td></tr>
<tr><td>All Data</td><td>Highest HART-enabled Channel Number X (88 words or 1408 bits)</td></tr>
</table>

Continued…

Module Parameters		
Parameter	***Default***	***Description***
I/O Scan Set	1	The scan set 1 – 32 to be assigned to the module by the RX3i CPU.
Channel Faults w/o Terminal Block	Disabled	Enabled / Disabled: Controls whether channel faults and configured alarm responses will be generated after a Terminal Block removal. The default setting of Disabled means channel faults and alarms are suppressed when the Terminal Block is removed. This parameter does not affect module faults including the Terminal Block loss/add fault generation.
Module Fault Reporting Enabled	Enabled	Enabled / Disabled. Controls whether the module will report faults resulting from either loss of field power or overtemperature conditions.
Field Power Removed Enabled	Enabled	Enabled / Disabled. With Module Fault Reporting enabled, this parameter controls reporting of Field Power Removed module faults.
Over Temp Enabled	Enabled	Enabled / Disabled. With Module Fault Reporting enabled, this parameter controls reporting of Overtemperature module faults.
Module Interrupt Reporting Enabled	Disabled	Enabled / Disabled.
Field Power Removed Enabled	Disabled	Enabled / Disabled. With Module Interrupt Reporting enabled, this parameter controls interrupts for Field Power Removed module faults.
Over Temp Enabled	Disabled	Enabled / Disabled. With Module Interrupt Reporting enabled, this parameter controls interrupts for Overtemperature module faults.

Channel Parameters		
Parameter	***Default***	***Description***
Range Type	Disabled Current	Sets up the type of output to be used for each channel. Choices are: Disabled Voltage, Disabled Current, Voltage/Current. Channels used for HART communications must have Range Type set to Voltage/Current.
Range (Only for Range Type Voltage/Current)	-10V to +10V	For voltage/current: -10V to +10V, 0V to +10V, 4mA to 20 mA, 0mA to 20 mA. Channels used for HART communications must have Range set to 4mA to 20A.
Channel Value Format	32-bit Floating Point	16-bit integer or 32-bit floating point
Outputs Default	Force to Default Value	Controls the state the output will be set to in Outputs Disabled mode (stop), if a fault occurs, if power is lost, or if the configuration is cleared. Choices are Hold Last State, or default to a specific configured default value.

Continued …

OverTemperature

If OverTemperature is enabled, the module generates an OverTemperature alarm if the module’s internal temperature is too great for the number of outputs that are on at the same time. In addition to the configurable options for OverTemperature fault reporting and interrupts, an over temperature condition is also indicated by the OverTemperature bit in the module’s Status Reference data. Detection of the OverTemperature status bit is always enabled.

Range Type

Each channel on the module that will be used should be configured for Voltage/Current. Its voltage or current range and other parameters can then be configured as needed. If the channel output will not be used and is not wired, select either “Disabled” option. If a channel is disabled, it is not necessary to configure any of its other parameters.

If the channel is wired to a current output, but will not presently be used, select "Disabled Current". This will set the channel's current output to 0mA (the channel’s voltage output will be non-zero).

If the channel is wired to a voltage output, but will not presently be used, select "Disabled Voltage". This will set the channel's voltage output to 0V (the channel’s current output will be non-zero).

Output Defaults

If Hold Last State is enabled, an output will hold its last commanded value when the CPU indicates Outputs Not Enabled, or if one of the fault conditions listed below occurs. If Hold Last State is disabled, the output is commanded to go to the Default Value. The Default Value must be set within the selected output range. If both Default Value and Ramp Rate are enabled, the channel will ramp to the default value. Fault conditions are:

- CPU outputs are not enabled
- Backplane power is not ok. In that case, there is no ramping, even if ramping is enabled.
- Loss of communications from CPU.
- Loss of I/O communications.
- Loss of field power.

Outputs Default Notes

- Hot Removal of the module in an I/O Enabled mode will cause all outputs to Hold Last State (even channels configured for Force to Default Value). If that operation is not desirable, the outputs can be forced to default by first turning off field power and removing the module's Terminal Block before hot-removing the module.
- Resetting the module using SVC_REQ 24 causes all channels to Hold Last State even if Default Value is configured. The application program must handle output defaulting before execution of the Service Request.
- Default Ramp Rate configuration is ignored if backplane power from the power supply is lost. Channels configured for Default Value go to the default value immediately.
- The first time a configuration is stored following a return of backplane power, the Default Ramp rate is not used. Any channel configured for Default Value goes to its default value immediately. If analog power was not lost and the same configuration is restored on the next powerup, the channel state is unchanged from the time the power was lost. The Default Ramp Rate is used for any subsequent reconfiguration.

Output Default Conditions and Actions

Condition	***Hold Last State or Default Value***	***Default Ramp Rate Enabled***	***Outputs Enabled and Ramp Rate Enabled***	***Channel Output Setting*** (Except where indicated, field power is assumed to be present).
Outputs Enabled and No Faults	N/A	N/A	No	Output goes to its commanded value from reference memory; defaults don't apply.
	N/A	N/A	Yes	Output is ramped to the commanded output from reference memory at the Outputs Enabled ramp rate. Defaults don't apply.
Outputs Disabled, Fault Mode, or Reconfiguration	Default Value	No	N/A	Output is set to the Default Value
	Default Value	Yes	N/A	Output is ramped to the Default Value at the Default ramp rate, starting at the last commanded value before entering mode.
	Hold Last State	N/A	N/A	Output is held at the last commanded value
Loss of Backplane Power or First Configuration Store after Powerup	Default Value	N/A	N/A	Output is set to the Default Value.
	Hold Last State	N/A	N/A	Output is held at last commanded value.
Hot Removal, Reset with SVCREQ 24 or Cleared Configuration	N/A	N/A	N/A	Output is held at last commanded value.
Loss of Field Power	N/A	N/A	N/A	All outputs go to 0V and 0mA.

Channel Parameters, *continued*

Parameter	***Default***	***Description***
High Scale Value (Eng Units)	The defaults for the 4 Scaling parameters depend on the configured Range Type and Range. Each Range and Range Type have a different set of defaults.	Note: Scaling is disabled if both High Scale Eng. Units equals High Scale A/D Units and Low Scale Eng. Units equals Low Scale A/D Units. Default = High A/D Limit of selected range type.
Low Scale Value (Eng Units)		Default is Low A/D Limit of selected range type. Must be lower than the high scaling value.
High Scale Value (A/D Units)		Default is High A/D Limit of selected range type. Must be greater than the low scaling value.
Low Scale Value (A/D Units)		Default is Low A/D Limit of selected range type.

Continued …

Output Scaling

By default, the module converts a floating point value from the CPU into a voltage or current output over the entire span of its configured Range. For example, if the Range of a channel is 4 to 20mA, the module accepts channel output values from 4.000 to 20.000. By modifying one or more of the four channel scaling parameters (Low/High Scale Value parameters) from their defaults, the scaled Engineering Unit range can be changed for a specific application. Scaling is always linear and inverse scaling is possible. All alarm values apply to the scaled Engineering Units value, not to the A/D units value.

The scaling parameters only set up the linear relationship between two sets of corresponding values. They do not have to be the limits of the output.

Example

In this example, the application should interpret 32000 counts as +10V and –32000 counts as -10V . The following channel configuration will scale a +/-10V output channel to +/-32000 counts.

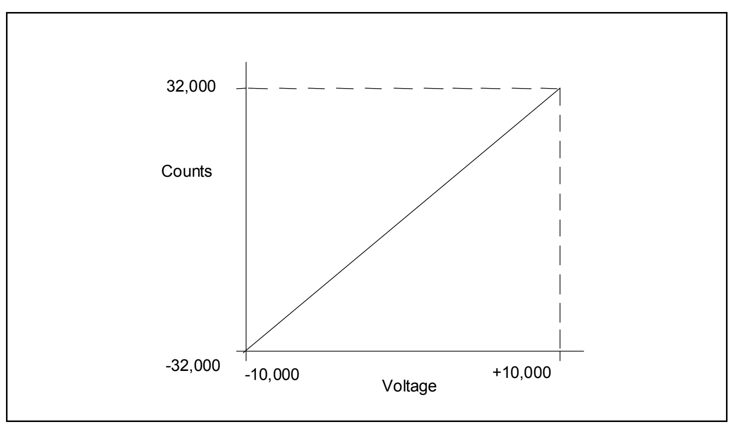
Channel Value Format = 16 Bit Integer

High Scale Value (Eng Units) = 32000.0

Low Scale Value (Eng Units) = -32000.0

High Scale Value (A/D Units) = 10.000

Low Scale Value (A/D Units) = -10.000



Channel Parameters, continued		
Parameter	***Default***	***Description***
High Alarm (Eng Units)	The defaults depend on the configured Range.	All of the alarm parameters are specified in Engineering Units. When the configured value is reached or below (above), a Low (High) Alarm is triggered.
Low Alarm (Eng Units)		
Outputs Enabled Ramp Rate (Eng Units)	0.0	The rate in Engineering Units at which the output will change during normal operation.
Default Ramp Rate (Eng Units)	0.0	The rate in Engineering Units at which the output will change if a fault condition occurs or if outputs are not enabled.
Output Clamping Enabled	Disabled	Enabled / Disabled. See description below.
Upper Clamp Limit (Eng Units)	The defaults depend on the configured Range.	The Upper Clamp Limit must be greater than the Lower Clamp Limit. This parameter can be used to restrict the output to a range that is narrower than its configured Range Type. For example, a channel configured for –10V to +10V could be restricted to -8V to +7.5V.
Lower Clamp Limit (Eng Units)		
Default Value (Eng Units)	0.0	If Hold Last State is disabled, the output is commanded to go to the Default Vale when the CPU is not in Outputs Enabled mode or under certain fault conditions.
User Offset (Eng Units)	0.0	A configurable value that can be used to change the base of the channel. This value is added to the scaled value of the channel before alarm-checking.

Continued …

Lower, Upper Clamp and Alarms

Alarms can be used to indicate when the module has been commanded to meet or exceed the configured high or low limits for each channel. These are set at six configurable alarm trigger points:

- High Alarm and Low Alarm
- Upper Clamp and Lower Clamp
- Overrange and Underrange Alarm

Each alarm is individually configurable per channel to generate diagnostics bit status, fault alarms, or interrupt alarms.

If a channel is commanded higher than the Upper Clamp value, the output is set to the Upper Clamp value and an Upper Clamp condition is indicated. If a channel is commanded lower than the Lower Clamp value, the output is set to the Lower Clamp value and a Lower Clamp condition is indicated.

The High and Low Alarm checks are performed on the engineering units output value after possibly being adjusted by ramping, clamping, and fault conditions.

Channel Parameters, *continued*

Parameter	***Default***	***Description***
Diagnostic Reporting Enable *If Diagnostic Reporting is enabled, the additional parameters listed below can be used to enable specific types of alarms.*	Disabled	*Diagnostic Reporting Enable options* are used to enable reference memory reporting of alarms into the Diagnostic Reference area. *Fault Reporting Enable option*s enable fault logging of alarms into the I/O Fault Table. These parameters enable or disable the individual diagnostics features of a channel. When any of these parameters is enabled, the module uses associated parameters to perform the enabled feature. For example, if Over Range is enabled in the “Diagnostic Reporting Enable” menu, the module will set the Over Range bit in the Diagnostic Reference for the channel. If any of these parameters is disabled, the module does not react to the associated alarm conditions. *For example, if Low Alarm Enable is set to Disabled in the “Fault Reporting Enable” menu, the Low Alarm fault is not logged in the I/O Fault Table when Low Alarm is detected on the channel.*
Fault Reporting Enable *If Fault Reporting is enabled, the additional parameters listed below can be used to enable specific types of Faults.*	Disabled	
Interrupts Enable	Disabled	
Low Alarm Enable	Disabled	
High Alarm Enable	Disabled	
Under Range Enable	Disabled	
Over Range Enable	Disabled	
Lower Clamp Alarm Enable	Disabled	
Upper Clamp Alarm Enable	Disabled	

Alarming and Fault Reporting

The Diagnostic Reporting Enable, Fault Reporting Enable, and Interrupt Enable configuration parameters can be used to enable different types of responses for individual channel alarms. By default, all responses are disabled on every channel. Any combination of alarm enables can be configured for each channel.

- If Diagnostic Reporting is enabled, the module reports channel alarms in reference memory at the channel's Diagnostic Reference address.
- If Fault Reporting is enabled, the module logs a fault log in the I/O Fault table for each occurrence of a channel alarm.
- If Interrupts are enabled, an alarm can trigger execution of an Interrupt Block in the application program, as explained below.

Using Interrupts

To properly configure an I/O Interrupt, the Interrupt enable bit or bits must be set in the module's configuration. In addition, the program block that should be executed in response to the channel interrupt must be mapped to the corresponding channel's reference address.

Example:

In this example, the Channel Values Reference Address block is mapped to %AQ0001-%AQ0008. An I/O Interrupt block should be triggered if a High Alarm condition occurs on channel 2.

- Configure the High-Alarm condition.
- Set the High-Alarm Interrupt Enable flag for Channel 2 in the module configuration.

Channel 2's reference address corresponds to %AQ00003 (2 Words per channel), so the interrupt program block Scheduling properties should be set for the "I/O Interrupt" Type and "%AQ0003" as the Trigger.

Fault Reporting and Interrupts

These modules have separate enable/disable options for Diagnostic Reporting and Interrupts. Normally, disabling a diagnostic (such as Low/High Alarm or Over/Under range) in the configuration means that its diagnostic bit is never set. However, if interrupts are enabled for a condition and that interrupt occurs, the diagnostic bit for that condition is also set during the I/O Interrupt block logic execution. The next PLC input scan always clears this interrupt status bit back to 0, because Diagnostic Reporting has it disabled.

Channel Parameters continued		
Parameter	***Default***	***Description***
HART Communications	Disabled	Enabled/disabled. Set HART Communications to enabled if the channel will use HART communications. Enabling HART communications forces the channel to 4-20mA operation.
HART Slot Variables	Disabled	Enabled/disabled. If HART Slot Variables is enabled, the module will periodically send HART command #33 to request data. Channel variables will be read and placed in the HART scan block channel data. For each slot, the variable assignment code can be set between 0 and 255.
Slot Code 0, 1, 2, 3	1	The slot transmitter variable assignment code that will be used to retrieve data from the connected HART device. These values are used in the request data for HART command #33.

Output Module Data Formats

This section explains how the module uses separate reference areas that can be assigned during module configuration:

- *Output Value Reference Data*, required memory for the analog output channel values.
- *Output Channel Diagnostic Reference Data*, optional memory for channel faults and alarms.
- *Module Status Reference Data,* optional memory for general module status data.

In addition, during configuration, optional *HART Reference Data,* memory can be assigned. See the section “HART Reference Data” later in this chapter for details.

Output Value Reference Data

The module receives its channel data from its configured output words, beginning at its assigned *Outputs Reference Address*. Each channel occupies 2 words, whether the channel is used or not:

Outputs Reference Address	***Contains Data for:***
+0, 1	Channel 1
+2, 3	Channel 2
+4, 5	Channel 3
+6, 7	Channel 4
+8, 9	Channel 5
+10, 11	Channel 6
+12, 13	Channel 7
+14, 15	Channel 8

Depending on its configured Channel Value Format, each enabled channel output reference location is read as a 32-bit floating point or 16-bit integer value.

In the 16-bit integer mode, low word of the 32-bit channel data area contains the 16-bit integer channel value. The high word (upper 16-bits) of the 32-bits is ignored. The full range of the 16-bit integer is a signed decimal value from +32767 to –32768.

Because the channel reference location is 32 bits, it is possible for the application program to write 32-bit signed decimal values to the output reference. However, the program logic must restrict the magnitude of the value to the range +32767 to –32768. Exceeding this range will result in misinterpretation of the sign bit, and incorrect output channel operation.

Output Channel Diagnostic Reference Data

The module can optionally be configured to report channel diagnostics status data to the CPU. The CPU stores this data at the module's configured *Diagnostic Reference Address*. Use of this feature is optional.

The diagnostics data each channel occupies 2 words (whether the channel is used or not):

Diagnostic Reference Address	***Contains Diagnostics Data for:***
+0, 1	Channel 1
+2, 3	Channel 2
+4, 5	Channel 3
+6, 7	Channel 4
+8, 9	Channel 5
+10, 11	Channel 6
+12, 13	Channel 7
+14, 15	Channel 8

When a diagnostic bit equals 1, the alarm or fault condition is present on the channel. When a bit equals 0 the alarm or fault condition is either not present or detection is not enabled in the configuration for that channel.

For each channel, the format of this data is:

Bit	***Description***
1	Low Alarm Exceeded = 1
2	High Alarm Exceeded = 1
3	Underrange = 1
4	Overrange = 1
5 – 20	Reserved (set to 0).
21	Lower Clamp Active = 1
22	Upper Clamp Active = 1
23 - 32	Reserved (set to 0).

Module Status Reference Data

The module can also optionally be configured to return 4 bits of module status data to the CPU. The CPU stores this data in the module's 32-bit configured *Module Status Data reference* area.

Bit	***Description***
1	Module OK (1 = OK, 0 = failure, or module is not present)
2	Terminal Block Present (1 = Present, 0 = Not present)
3	Field Power (0 = Present, 1 = Not present)
4	Module Overtemperature (0 = Not overtemperature, 1 = Approaching or exceeding overtemperature)
5 - 32	Reserved

Terminal Block Detection

The module automatically checks for the presence of a Terminal Block. The module's TB LED indicates the state of the terminal block. It is green when the Terminal Block is present or red if it is not.

Faults are automatically logged in the CPU's I/O Fault table when the terminal block is inserted or removed from a configured module in the system. The fault type is Field Fault and the fault description indicates whether the fault is a "Loss of terminal block" or an "Addition of terminal block". If a Terminal Block is not present while a configuration is being stored, a "Loss of terminal block" fault is logged.

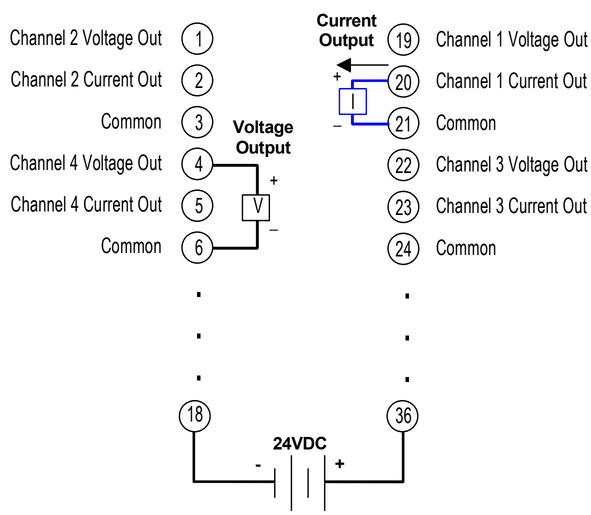
Bit 2 of the Module Status Reference indicates the status of the terminal block. To enable Module Status reporting, the Module Status Reference must be configured. During operation, the PLC must be in an I/O Enabled mode for the current Module Status to be scanned and updated in reference memory.

Field Wiring: IC695ALG728

The table below lists wiring connections for the module. There are no shield terminals. For shielding, tie cable shields to the ground bar along the bottom of the backplane. M3 tapped holes are provide in the ground bar for this purpose.

Terminal	Assignment	Assignment	Terminal
1	Channel 2 Voltage Out	Channel 1 Voltage Out	19
2	Channel 2 Current Out	Channel 1 Current Out	20
3	Common (COM)	Common (COM)	21
4	Channel 4 Voltage Out	Channel 3 Voltage Out	22
5	Channel 4 Current Out	Channel 3 Current Out	23
6	Common (COM)	Common (COM)	24
7	Channel 6 Voltage Out	Channel 5 Voltage Out	25
8	Channel 6 Current Out	Channel 5 Current Out	26
9	Common (COM)	Common (COM)	27
10	Channel 8 Voltage Out	Channel 7 Voltage Out	28
11	Channel 8 Current Out	Channel 7 Current Out	29
12	Common (COM)	Common (COM)	30
13	Common (COM)	Common (COM)	31
14	Common (COM)	Common (COM)	32
15	Common (COM)	Common (COM)	33
16	Common (COM)	Common (COM)	34
17	Common (COM)	Common (COM)	35
18	Common (COM)	+24V In	36

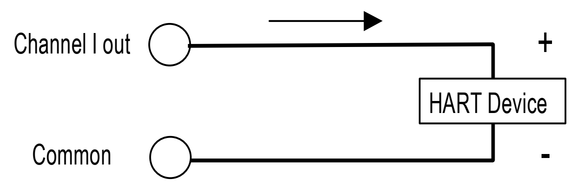
Each channel can be individually-configured to operate as a voltage output or a current output, not both simultaneously. All the common terminals are connected together internally. so any common terminal can be used for the negative lead of the external power supply.



HART Device ConnectionsError! Bookmark not defined.

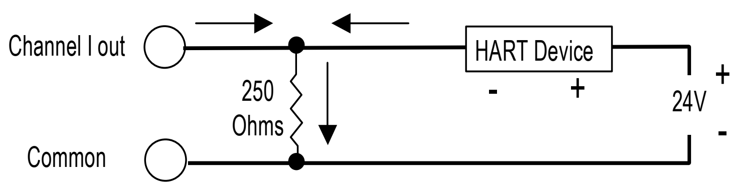
Example connections for 2-wire transmitters are shown below.

Connecting a HART Output Device



Connecting a HART Input Transmitter to an Output Channel

In this type of application, the HART output module, IC695ALG728, cannot read the analog current level from the HART device. However, the module can communicate with the HART signal. There is no analog input to the module.



HART Reference Data

If *HART Data Scan Control* is configured as “Dynamic Data Only” or “All Data”, the CPU automatically scans the HART data listed on the next page into the module’s configured *HART Data Reference Address*. The data length depends on whether All Data or Dynamic Data is selected.

This data includes response data associated with several HART Pass-Thru Commands. The module stores this data, and then passes it to the CPU either in the automatic HART data scan described above, or in response to function blocks in the application program.

Note that invalid or uninitialized REAL (floating point) data will be set to NaN (Not-A-Number).

HART Status Data

In addition to the HART Reference Data, the module reports the status of HART communications in its configured HART Status Reference Address. The length of this data is 4 words / 64 bits:

Word 1	Device Present, one bit per channel. Channel 1 in lowest bit. For 8-channel module, bits 9-16 are not used.
Word 2	Device Initializing, one bit per channel. Channel 1 in lowest bit. For 8-channel module, bits 9-16 are not used.
Words 3 and 4	Not used

HART Data Format

		Byte Offset		Field Description	Type
		Begin	End		
All HART Data	Dynamic HART Data Only	0x0000	0x0003	HART Primary Variable, CMD#3, Bytes 5-8	REAL
		0x0004	0x0007	HART Secondary Variable CMD#3, Bytes 10-13	REAL
		0x0008	0x000b	HART Tertiary Variable CMD#3, Bytes 15-18	REAL
		0x000c	0x000f	HART Fourth Variable CMD#3, Bytes 20-23	REAL
		0x0010	0x0013	Slot 0 value. CMD#33, Bytes 2-5	REAL
		0x0014	0x0017	Slot 1 value CMD#33, Bytes 8-11	REAL
		0x0018	0x001b	Slot 2 value CMD#33, Bytes 14-17	REAL
		0x001c	0x001f	Slot 3 value CMD#33, Bytes 20-23	REAL
		0x0020		HART communication status byte from the last HART command response. See page 11-48.	BYTE
		0x0021		HART device status byte from the last HART command response.	BYTE
		0x0022		Spare for alignment.	BYTE*2
		0x0024		HART device Manufacturer ID. CMD#0, Byte 1	BYTE
		0x0025		HART device type code. CMD#0, Byte 2	BYTE
		0x0026		Minimum number of preambles device requires. CMD#0, Byte 3	BYTE
		0x0027		HART Universal command code. CMD#0, Byte 4	BYTE
		0x0028		HART Transmitter specific revision. CMD#0, Byte 5	BYTE
		0x0029		HART device software revision number. CMD#0, Byte 6	BYTE
		0x002A		HART device hardware revision number. CMD#0, Byte 7	BYTE
		0x002B		HART flags. CMD#0, Byte 8	BYTE
		0x002C	0x002F	HART device ID number. CMD#0, Byte 9-11	BYTE*4
		0x0030	0x0037	8 character device tag. CMD#13, Bytes 0-5 in unpacked ASCII.	BYTE*8
		0x0038	0x0047	Device Descriptor. CMD#13, Bytes 6-17 in unpacked ASCII	BYTE*16
		0x0048		HART Primary Variable Units. CMD#3, Byte 4	BYTE
		0x0049		HART Secondary Variable Units. CMD#3, Byte 9, 0 if not present.	BYTE
		0x004a		HART Tertiary Variable Units. CMD#3, Byte 14, 0 if not present.	BYTE
		0x004b		HART Fourth Variable Units. CMD#3, Byte 19, 0 if not present.	BYTE
		0x004c		HART Primary Variable Code. CMD#50, Byte 0	BYTE
		0x004d		HART Secondary Variable Code. CMD#50, Byte 1	BYTE
		0x004e		HART Tertiary Variable Code. CMD#50, Byte 2	BYTE
		0x004f		HART Fourth Variable Code. CMD#50, Byte 3	BYTE
		0x0050		Units code for range parameter. CMD#15, Byte 2	BYTE
		0x0051	0x0053	Spare for alignment	BYTE*3
		0x0054	0x0057	Low transmitter range for analog signal in eng. units. CMD#15, Bytes 3-6	REAL
		0x0058	0x005b	High transmitter range for analog signal in eng. units. CMD#15, Bytes 7-10	REAL
		0x005c		Slot 0 units code. CMD#33, Byte 1	BYTE
		0x005d		Slot 1 units code. CMD#33, Byte 7	BYTE
		0x005e		Slot 2 units code. CMD#33, Byte 13	BYTE
		0x005f		Slot 3 units code. CMD#33, Byte 19	BYTE
		0x0060		Slot 0 variable code. CMD#33, Byte 0	BYTE
		0x0061		Slot 1 variable code. CMD#33, Byte 6	BYTE
		0x0062		Slot 2 variable code. CMD#33, Byte 12	BYTE
		0x0063		Slot 3 variable code. CMD#33, Byte 18	BYTE
		0x0064	0x0083	32 character message. CMD#12, Bytes 0-23 unpacked ASCII.	BYTE*32
		0x0084	0x0087	Stored date in the field device. CMD#13, Bytes 18-20.	BYTE*4
		0x0088	0x008b	The final assembly number is used for identifying the material and electronics that comprise the field device. CMD#16, Bytes 0-2	BYTE*4
		0x008c	0x00a4	The extended status returned by HART command 48.	BYTE*25
		0x00a5	0x00af	Spare	BYTE*11

COMMREQs for HART Modules

Two Communication Request (COMMREQ) functions can be used in the application program to communicate with RX3i HART modules.

COMMREQ 1, Get HART Device Information, reads

COMMREQ 2, Send HART Pass-Thru Command

Get HART Device Information, COMMREQ 1 Command Block

<table>
<tr><th>Word Offset</th><th>Value
Dec (Hex)</th><th colspan="3">Definition</th></tr>
<tr><td>Word 1</td><td>8 (0008)</td><td colspan="3">Length of command Data Block in words beginning at Word 7.</td></tr>
<tr><td>Word 2</td><td>0 (0000)</td><td colspan="3">Always 0 (no-wait mode request)</td></tr>
<tr><td rowspan="10">Word 3</td><td rowspan="10"></td><td colspan="3">Memory type of COMMREQ status word. (Words 3 and 4 specify the starting address where the status word will be written.) It can be:</td></tr>
<tr><td>Memory Type</td><td>Memory Type</td><td>Decimal code</td></tr>
<tr><td>%I</td><td>Discrete input table (Bitmode)</td><td>70</td></tr>
<tr><td>%Q</td><td>Discrete output table (Bit mode)</td><td>72</td></tr>
<tr><td>%I</td><td>Discrete input table (Byte mode)</td><td>16</td></tr>
<tr><td>%Q</td><td>Discrete output table (Byte mode)</td><td>18</td></tr>
<tr><td>%R</td><td>Register memory</td><td>8</td></tr>
<tr><td>%W</td><td>Word memory</td><td>196</td></tr>
<tr><td>%AI</td><td>Analog input table</td><td>10</td></tr>
<tr><td>%AQ</td><td>Analog output table</td><td>12</td></tr>
<tr><td>Word 4</td><td>0-based offset</td><td colspan="3">COMMREQ status word address minus 1.
Example: if Words 3 and 4 contain values of 8 and 9 respectively, the status word will be written to %R10.</td></tr>
<tr><td>Word 5, 6</td><td>0 (0000)</td><td colspan="3">Reserved</td></tr>
<tr><td>Word 7</td><td>1 (0001)</td><td colspan="3">Command code for the COMMREQ to be executed.
Get HART Device Information = 1.</td></tr>
<tr><td>Word 8</td><td>1 (0001)</td><td colspan="3">Number of Response Reference areas that follow (does not include COMMREQ status word). Always 1.</td></tr>
</table>

<table>
<tr><th>Word Offset</th><th>Value
Dec (Hex)</th><th colspan="3">Definition</th></tr>
<tr><td rowspan="10">Word 9</td><td rowspan="10"></td><td colspan="3">Memory type for the reply data. (Words 9—12 specify the starting address where the response will be written.)</td></tr>
<tr><td>Memory Type</td><td>Memory Type</td><td>Decimal code</td></tr>
<tr><td>%I</td><td>Discrete input table (Byte mode)</td><td>16</td></tr>
<tr><td>%Q</td><td>Discrete output table (Byte mode)</td><td>18</td></tr>
<tr><td>%W</td><td>Word memory</td><td>196</td></tr>
<tr><td>%R</td><td>Register memory</td><td>8</td></tr>
<tr><td>%AI</td><td>Analog input table</td><td>10</td></tr>
<tr><td>%AQ</td><td>Analog output table</td><td>12</td></tr>
<tr><td>%T</td><td>Discrete temporary memory (Byte)</td><td>20</td></tr>
<tr><td>%M</td><td>Discrete internal memory (Byte)</td><td>22</td></tr>
<tr><td>Word 10</td><td>0 (0000)</td><td colspan="3">Bit Offset (must be 0 for all requests).</td></tr>
<tr><td>Word 11</td><td>0-based offset (low word).</td><td colspan="3">Starting address to which the response will be written. The value entered is the 0-based offset from the beginning of PLC memory for the memory type specified in Word 9. This offset is in bytes or words depending on the memory type specified. Valid ranges of values depend on the PLC’s memory ranges.
Example: If Words 9 and 11 contain values of 8 and 250 respectively, the response will be written to %R251.</td></tr>
<tr><td>Word 12</td><td>0-based offset (high word)</td><td colspan="3">High word of offset. Value = 0 for most memory types. High word is non-zero only on if %W memory is used.</td></tr>
<tr><td>Word 13</td><td>Words: 90 (005A)
Bytes: 180 (00B4)</td><td colspan="3">Maximum size of response area. Must be 90 if word memory type is used; 180 if discrete memory type is used.</td></tr>
<tr><td>Word 14</td><td>Range 1-16.</td><td colspan="3">Channel Number 1-16 (valid range depends on module channel count and single-ended versus differential mode)</td></tr>
</table>

COMMREQ Status Word

The COMMREQ status word for the Get HART Device Information command is shown below.

Value Dec (Hex)	Description
0 (0000)	Device has not yet processed the COMMREQ.
1 (0001)	Command Complete **Note:** This status does not necessarily mean success. Some commands have reply data that must also be checked.
2 (0002)	Command Terminated – module busy
3 (0003)	Command Terminated – invalid command
4 (0004)	Command Terminated – invalid command data
5 (0005)	Command Terminated – not enough data
6 (0006)	Not used
7 (0007)	Command Terminated – not enough memory in reply area The command did not specify sufficient PLC memory for the reply. Command will be ignored.
8 (0008)	Command Terminated – command-specific error. See Additional Code in the Status Block for more information.
265 (0109)	Error, Hart device not connected
521 (0209)	Error, Channel not HART-enabled
777 (0309)	Error, Analog Output Module, No field power
1033 (0409)	Error. HART command now allowed
1289 (0509)	Error. Invalid HART command

Get HART Device Information, COMMREQ 1: Reply Data Format

The response to a Get HART Device Information COMMREQ is written to the PLC memory location specified in words 9-12 of the COMMREQ.

Byte	Name	Description
1, 2	Command Code	Echo of Command code. (0x0001)
3, 4	Channel Number	Echo of Channel Number
5-8	HART Primary Variable	CMD#3, Bytes 5-8. Type: REAL
9-12	HART Secondary Variable	CMD#3, Bytes 10-13 Type: REAL
13-16	HART Tertiary Variable	CMD#3, Bytes 15-18. Type: REAL
17-20	HART Fourth Variable	CMD#3, Bytes 20-23. Type: REAL
21-24	Slot 0 value	CMD#33, Bytes 2-5. Type: BYTE
25-28	Slot 1 value	CMD#33, Bytes 8-11. Type: BYTE
29-32	Slot 2 value	CMD#33, Bytes 14-17. Type: BYTE
33-36	Slot 3 value	CMD#33, Bytes 20-23. Type: BYTE
37	HART communication status byte from the last HART command response, see next page	
38	HART device status byte from the last HART command response, see next page.	
39-40	Spare for alignment.	Type: BYTE
41	HART device Manufacturer ID. CMD#0, Byte 1	Type: BYTE
42	HART device type code. CMD#0, Byte 2	Type: BYTE
43	Minimum number of preambles device requires	CMD#0, Byte 3. Type: BYTE
44	HART Universal command code	CMD#0, Byte 4. Type: BYTE
45	HART Transmitter specific revision	CMD#0, Byte 5 Type: BYTE
46	HART device software revision number	CMD#0, Byte 6 Type: BYTE
7	HART device hardware revision number	CMD#0, Byte 7 Type: BYTE
48	HART flags	CMD#0, Byte 8 Type: BYTE
49-52	HART device ID number	CMD#0, Byte 9-11 Type: 4 BYTEs
53-60	8 character device tag.	CMD#13, Type: 8 BYTEs. Bytes 0-5 are unpacked ASCII
61-76	Device Descriptor	CMD#13, TYPE: 16 BYTEs. Bytes 6-17 are unpacked ASCII
77	HART Primary Variable Units	CMD#3, Byte 4. Type: BYTE
78	HART Secondary Variable Units	CMD#3, Byte 9, 0 if not present. Type: BYTE
79	HART Tertiary Variable Units	CMD#3, Byte 14, 0 if not present. Type: BYTE
80	HART Fourth Variable Units	CMD#3, Byte 19, 0 if not present. Type: BYTE
81	HART Primary Variable Code	CMD#50, Byte 0 Type: BYTE
82	HART Secondary Variable Code	CMD#50, Byte 1 Type: BYTE
83	HART Tertiary Variable Code	CMD#50, Byte 2 Type: BYTE
84	HART Fourth Variable Code	CMD#50, Byte 3 Type: BYTE
85	Units code for range parameter	CMD#15, Byte 2 Type: BYTE
86-88	Spare for alignment	3 BYTEs
89-92	Low transmitter range for analog signal in engineering units	CMD#15, Bytes 3-6 Type: REAL
93-96	High transmitter range for analog signal in engineering units	CMD#15, Bytes 7-10 Type: REAL
97	Slot 0 units code	CMD#33, Byte 1 Type: REAL
98	Slot 1 units code	CMD#33, Byte 7 Type: REAL
99	Slot 2 units code	CMD#33, Byte 13 Type: REAL
100	Slot 3 units code	CMD#33, Byte 19 Type: REAL
101	Slot 0 variable code	CMD#33, Byte 0 Type: REAL
102	Slot 1 variable code	CMD#33, Byte 6 Type: REAL
103	Slot 2 variable code	CMD#33, Byte 12 Type: REAL
104	Slot 3 variable code	CMD#33, Byte 18 Type: REAL
105-136	32 character message	CMD#12, Bytes 0-23 unpacked ASCII. Type: 32 BYTEs
137-140	Stored date in the field device	CMD#13, Bytes 18-20. Type 4 BYTEs
141-144	Number identifying the field device’s material and electronics	CMD#16, Bytes 0-2. Type 4 BYTEs
145-169	The extended status returned by HART command 48.	Type: 25 BYTEs
170-180	Spare for alignment	Type: 11 BYTEs

Sending a HART Pass-Thru Command to a HART Device

The HART module automatically uses several HART Pass-thru commands as described earlier in this chapter, In addition, the application program can use the Send HART Pass-Thru Command (COMMREQ 2) to send HART Pass-Thru commands to an RX3i HART module. The HART module stores the data returned by the command in its on-board memory. This data can then be scanned automatically by the CPU or read as needed from the application program.

A list of Pass-Thru commands is included in this section. The RX3i HART module then passes the command to the intended HART input or output device. Responses to HART Pass-Thru commands are available to the application program in the COMMREQ replies.

The Send HART Pass-Thru Command COMMREQ automatically fills in the Start Character, Address, Byte Count, Status, and the checksum. The RX3i HART module waits until the data from the HART device is available before it replies to this command, so the application program does not have to query the module for the response. The application program must check the COMMREQ Status word to determine when the reply data is available. The reply is returned between 750mS and 8 seconds later. The reply time depends on the number of channels enabled, the pass thru rate selected, and whether other pass-thru operations are occurring at the same time.

Only one application program Pass-Thru command per channel is allowed at a time. If another request is made on a channel that has a Pass-Thru in-progress, the module returns a COMMREQ Status Word = 0x0002 (module busy).

HART Pass-Thru Command Block, COMMREQ 2

<table>
<tr><th rowspan="2">Word Offset</th><th colspan="2">Value</th><th rowspan="2" colspan="3">Definition</th></tr>
<tr><th>Dec</th><th>Hex</th></tr>
<tr><td>Word 1</td><td>10+x</td><td>000A + x</td><td colspan="3">Length of command Data Block in words beginning at Word 7</td></tr>
<tr><td>Word 2</td><td>0</td><td>0000</td><td colspan="3">Always 0 (no-wait mode request)</td></tr>
<tr><td rowspan="10">Word 3</td><td rowspan="10" colspan="2"></td><td colspan="3">Memory type of COMMREQ status word. It can be:</td></tr>
<tr><th>Memory Type</th><th>Memory Type</th><th>Decimal code</th></tr>
<tr><td>%I</td><td>Discrete input table (Bit mode)</td><td>70</td></tr>
<tr><td>%Q</td><td>Discrete output table (Btmode)</td><td>72</td></tr>
<tr><td>%I</td><td>Discrete input table (Byte mode)</td><td>16</td></tr>
<tr><td>%Q</td><td>Discrete output table (Byte mode)</td><td>18</td></tr>
<tr><td>%R</td><td>Register memory</td><td>8</td></tr>
<tr><td>%W</td><td>Word memory</td><td>196</td></tr>
<tr><td>%AI</td><td>Analog input table</td><td>10</td></tr>
<tr><td>%AQ</td><td>Analog output table</td><td>12</td></tr>
<tr><td>Word 4</td><td colspan="2">0-based offset</td><td colspan="3">COMMREQ status word address minus 1
Example: if Words 3 and 4 contain values of 8 and 9 respectively, the status word will be written to %R10.</td></tr>
<tr><td>Word 5</td><td>0</td><td>0000</td><td colspan="3">Reserved</td></tr>
<tr><td>Word 6</td><td>0</td><td>0000</td><td colspan="3">Reserved</td></tr>
<tr><td>Word 7</td><td>2</td><td>0002</td><td colspan="3">Command code for the COMMREQ to be executed.
HART Pass-Thru Command = 2</td></tr>
<tr><td>Word 8</td><td>1</td><td>0001</td><td colspan="3">Number of Response Reference areas that follow (does not include COMMREQ status word). Always 1</td></tr>
<tr><td rowspan="10">Word 9</td><td rowspan="10" colspan="2"></td><td colspan="3">Memory type for the reply data. (Words 9—12 specify the starting address where the response will be written).</td></tr>
<tr><th>Memory Type</th><th>Memory Type</th><th>Decimal code</th></tr>
<tr><td>%I</td><td>Discrete input table (Byte mode)</td><td>16</td></tr>
<tr><td>%Q</td><td>Discrete output table (Byte mode)</td><td>18</td></tr>
<tr><td>%W</td><td>Word memory</td><td>196</td></tr>
<tr><td>%R</td><td>Register memory</td><td>8</td></tr>
<tr><td>%AI</td><td>Analog input table</td><td>10</td></tr>
<tr><td>%AQ</td><td>Analog output table</td><td>12</td></tr>
<tr><td>%T</td><td>Discrete temporary memory (Byte)</td><td>20</td></tr>
<tr><td>%M</td><td>Discrete internal memory (Byte)</td><td>22</td></tr>
<tr><td>Word 10</td><td>0</td><td>0000</td><td colspan="3">Bit Offset (must be 0 for all requests)</td></tr>
<tr><td>Word 11</td><td colspan="2">0-based offset (low word)</td><td colspan="3">Starting address to which the response will be written. The value entered is the 0-based offset from the beginning of PLC memory for the memory type specified in Word 9. This offset will be in bytes or words depending on the memory type specified. Valid ranges of values depend on the PLC’s memory ranges.
Example: If Words 9 and 11 contain values of 8 and 250 respectively, the response will be written to %R251.</td></tr>
<tr><td>Word 12</td><td colspan="2">0-based offset (high word)</td><td colspan="3">High word of offset. Value = 0 for most memory types. Would only have a non-zero value if %W memory is used</td></tr>
<tr><td>Word 13</td><td colspan="2">Response data size</td><td colspan="3">Maximum size of response area. Size in bytes if discrete memory type used for response. Size in words if word type used</td></tr>
<tr><td>Word 14</td><td colspan="2">Channel Number (1-16).</td><td colspan="3">Channel Number 1-16 (valid range depends on module channel count and single-ended versus differential mode)</td></tr>
<tr><td>Word 15</td><td colspan="2">HART command (0x0 – 0xff)</td><td colspan="3">HART Pass-Thru Command type. HART Pass-Thru Commands that can be sent to an RX3i HART module are listed in this section.</td></tr>
</table>

Word 16	Command Data byte count	Size in bytes of command data that follows
…	…	…
Word 16+x	…	HART Command Data. Request data must be byte-packed and in big-endian format, PLC CPU format is little-endian, so some commands may require swapping of fields from little-endian to big-endian format as described in this chapter. This is usually needed for floating point data.

HART Pass-Thru Reply Data Format

The RX3i HART module returns the response data below to the CPU memory location specified by words 9-12 of the COMMREQ. Data beginning at Word 7 of the reply is byte-packed and in big-endian format. PLC CPU format is little-endian, so some commands may require swapping of fields from big-endian to little-endian format as described in this chapter. This is usually needed for floating point data.

Word	**Name**	**Description**
1	Command Code	Echo of Command code (0x0002)
2	Channel Number	Echo of Channel Number (same as request)
3	HART command	Echo of HART Pass-Thru Command type. See the tables in this section.
4	HART Status	Low byte is HART Comm Status and high byte is HART Dev Status from HART device response.
5	Spare	Spare for future use. User logic should not check this value because future module revisions may make this non-zero.
6	Response Byte Count (x)	Size in bytes of the response data that follows.
7L	Data Low	First response data byte from device.
7H	Data High	Second response data byte from device.
…	…	…
7+(x-1)/2 L	Data Low	….
7+(x-1)/2 H	Data High	Last response data byte from device.

COMMREQ Status Word

The following table defines the values that can be returned in the COMMREQ status word.

Value ***Dec (Hex)***	***Description***
0 (0000)	Device has not yet processed the COMMREQ.
1 (0001)	Command Complete. This status does not necessarily mean success. Some commands have reply data that must also be checked.
2 (0002)	Command Terminated – module busy
3 (0003)	Command Terminated – invalid command
4 (0004)	Command Terminated – invalid command data
5 (0005)	Command Terminated – not enough data
6 (0006)	Not used
7 (0007)	Command Terminated – not enough memory in reply area. The command did not specify sufficient PLC memory for the reply. Command will be ignored.
8 (0008)	Command Terminated – command-specific error. See Additional Code in the Status Block for more information.
265 (0109)	Error, Hart device not connected
521 (0209)	Error, Channel not HART-enabled
777 (0309)	Error, Analog Output Module, No field power
1033 (0409)	Error. HART command now allowed
1289 (0509)	Error. Invalid HART command

This status information relates to the execution of the COMMREQ function, not to the status of the HART communications. HART communications status is provided in the response data, as shown previously in this section.

HART Pass-Thru Commands and Command Codes for RX3i Modules

Within a HART command, data can be represented as integers, floating point numbers, ASCII text strings, or enumerated item lists. Unmarked data types are 8-, 16, or 24-bit integers (including code values)

Universal Commands		***Data in Command***			***Data in Reply***		
#	**Function**	**Byte**	**Data**	**Type**	**Byte**	**Data**	**Type**
0	Read unique identifier		None		0	"254" (expansion)	
					1	Manufacturer identification code	
					2	Manufacturer device type code	
					3	Number of preambles required	
					4	Universal command revision	
					5	Device-specific command revision	
					6	Software revision	
					7	Hardware revision	Integer
					8	Device function flags: bit 0 = multisensor device, bit 1 = protocol bridge device	Bit
					9-11	Device ID number	
1	Read primary variable		None		0	PV units code	
					1-4	Primary variable (PV)	Floating pt
2	Read current and percent of range		None		0-3	Current (mA)	Floating pt
3	Read current and four predefined dynamic variables		None		0-3	Current (mA)	Floating pt
					4	PV units code	
					5-8	Primary variable (PV)	Floating pt
					9	SV units code	
					10-13	Secondary variable (SV)	Floating pt
					14	TV units code	
					15-18	Third variable (VT)	Floating pt
					19	FV units code	
					20-23	Fourth variable (FV)	Floating pt
6	Write polling address	0	Polling address			same as command	
11	Read unique identifier associated with tag	0-5	Tag	ASCII	0-11	Same as command #0, see above	
12	Read message		None		0-23	Message (32 characters)	ASCII
13	Read tag, descriptor, date		None		0-5	Tag (8 characters)	ASCII
					6-17	Descriptor (16 characters)	ASCII
					18-20	Date	date
14	Read Primary Variable sensor information		None		0-2	Sensor serial number	
					3	Units code for sensor limits and min. span	
					4-7	Upper sensor limit	Floating pt
					8-11	Lower sensor limit	Floating pt
					12-15	Minimum span	Floating pt
15	Read output information		None		0	Alarm select code	
					1	Transfer function code	
					2	PV/range units code	
					3-6	Upper range value	Floating pt
					7-10	Lower range value	Floating pt
					11-14	Damping value (seconds)	Floating pt
					15	Write-protect code	
					16	Private-label distributor code	
16	Read final assembly number		None		0-2	Final assembly number	
17	Write message	0-23	Message (32 characters)	ASCII		Same as command	
18	Write tag, descriptor, date	0-5	Tag (8 characters)	ASCII		Same as command	
		6-17	Descriptor (16 characters)	ASCII			
		18-20	Date	date			
19	Write final assembly number	0-2	Final assembly number			Same as command	

Among the common-practice commands listed below, commands #60 and #62 through #70 are used to configure and control the multiple outputs generated by some multivariable transmitters. Such multiple outputs are numbered 1 to 4, corresponding to the HART dynamic variables: PV (primary variable), SV (secondary variable), TV (third variable) and FV (fourth variable).

Common-Practice Commands		***Data in Command***			***Data in Reply***		
#	***Function***	***Byte***	***Data***	***Type***	***Byte***	***Data***	***Type***
33	Read transmitter variables	0	Transmitted variable code for slot 0		0	Transmitted variable code for slot 0	
		1	Transmitted variable code for slot 1		1	Units code for slot 0	
		2	Transmitted variable code for slot 2		2-5	Variable for slot 0	Floating pt
		3	Transmitted variable code for slot 3		6	Transmitted variable code for slot 1	
		Truncated after last requested code			7	Units code for slot 1	
					8-11	Variable for slot 1	Floating pt
					12	Transmitted variable code for slot 2	
					13	Units code for slot 2	
					14-17	Variable for slot 2	Floating pt
					18	Transmitted variable code for slot 3	
					19	Units code for slot 3	
					20-23	Variable for slot 3	Floating pt
					Truncated after last requested variable		
34	Write damping value	0-3	Damping value (seconds	Floating pt	Same as command		
35	Write range values	0	Range units code		Same as command		
		1-4	Upper range value	Floating pt			
		5-8	Lower range value	Floating pt			
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	0	EEPROM control code: (0 = write to EEPROM 1 = read EEPROM to RAM)		Same as command		
40	Enter/exit fixed current mode	0-3	Current (mA) (0 = fixed current mode)	Floating pt	Same as 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	0	PV units code		Same as command		
45	Trim DAC zero	0-3	Measured current (mA)	Floating pt	Same as command		
46	Trim DAC gain	0-3	Measured current (mA)	Floating pt	Same as command		
47	Write transfer function	0	Transfer function code		Same as command		
48	Read additional device status	none			0-5	Device-specific status	Bit
					6-7	Operational modes (1-5)	
					8-10	Analog outputs saturated*	Bit
					11-13	Analog outputs fixed*	Bit
					14-24	Device-specific status	Bit
					* 24 bites each: LSB…MS refers to analog outputs 1 to 24. Response is truncated after last byte implemented		
49	Write PV sensor serial number	0-2	Sensor serial number		Same as command		
50	Read dynamic variable assignments	None			0	PV transmitter variable code	
					1	SV transmitter variable code	
					2	TV transmitter variable code	
					3	FV transmitter variable code	

Common-Practice Commands		Data in Command			Data in Reply		
#	**Function**	**Byte**	**Data**	**Type**	**Byte**	**Data**	**Type**
51	Read dynamic variable assignments	0	PV transmitter variable code		Same as command		
		1	SV transmitter variable code				
		2	TV transmitter variable code				
		3	FV transmitter variable code				
52	Set transmitter variable zero	0	Transmitter variable code		Same as command		
53	Write transmitter variable units	0	transmitter variable code		Same as command		
		1	transmitter variable units code				
54	Read transmitter variable information	0	Transmitter variable code		0	Transmitter variable code	
					1-3	Transmitter variable sensor serial number	
					4	Transmitter variable limits units code	
					5-8	Transmitter variable upper limit	Floating pt
					9-12	Transmitter variable lower limit	Floating pt
					13-16	Transmitter variable damping value (sec.)	Floating pt
					17-20	Transmitter variable minimum span	Floating pt
55	Write transmitter variable damping value	0	transmitter variable code		Same as command		
		1-4	transmitter variable damping value (seconds)	Floating pt			
56	Write transmitter variable sensor serial number	0	transmitter variable code		Same as command		
		1-3	transmitter variable sensor serial number	Floating pt			
57	Read unit tag, descriptor, date	None			0-5	Unit tag (8 characters)	ASCII
					6-17	Unit descriptor (16 characters)	ASCII
					18-20	Unit date (3 bytes: day, month, year)	Date
58	Write unit tag, descriptor, date	0-5	Unit tag (8 characters)	ASCII	Same as command		
		6-17	Unit descriptor (16 characters)	ASCII			
		18-20	Unit date (3 bytes: day, month, year)	Date			
59	Write number of response preambles	0	Number of response preambles		Same as command		
60	Read analog output and percent of range	0	analog output number code		0	Analog output number code	
					1	Analog output units code	
					2-5	Analog output level	Floating pt
					6-9	Analog output percent of range	Floating pt
61	Read dynamic variables and Primary Variable analog output	None			0	PV analog output units code	
					1-4	PV analog output level	Floating pt
					5	PV units code	
					6-9	Primary variable (PV)	Floating pt
					10	SV analog output units code	
					11-14	Secondary variable	Floating pt
					15	TV analog output units code	
					16-19	Third variable	Floating pt
					20	FV analog output units code	
					21-24	Fourth variable	Floating pt
					Truncated after last supported variable		
62	Read analog outputs (5.1)	0	Analog output number code for slot 0		0	Slot 0 analog output number code	
		1	Analog output number code for slot 1		1	Slot 0 units code	
		2	Analog output number code for slot 2		2-5	Slot 0 level	Floating pt
		3	Analog output number code for slot 3		6	Slot 1 analog output number code	
		Truncated after last requested level			7	Slot 1 units code	
					8-11	Slot 1 level	Floating pt
					12	Slot 2 analog output number code	
					13	Slot 2 units code	
					14-17	Slot 2 level	Floating pt
					18	Slot 3 analog output number code	
					19	Slot 3 units code	
					20-23	Slot 3 level	Floating pt
					Truncated after last requested level		

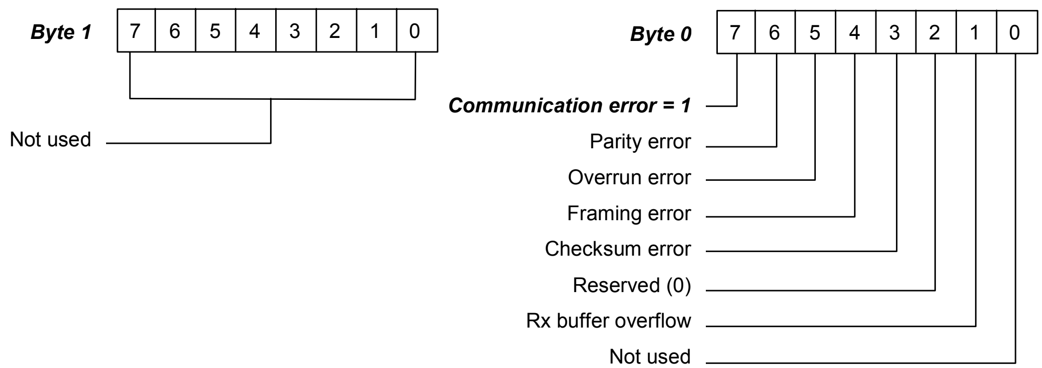
Common-Practice Commands		Data in Command			Data in Reply		
#	Function	Byte	Data	Type	Byte	Data	Type
63	Read analog output information	0	Analog output number code		0	Analog output number code	
					1	Analog output alarm select code	
					2	Analog output transfer function code	
					3	Analog output range units code	
					4-7	Analog output upper range value	Floating pt
					8-11	Analog output lower range value	Floating pt
					12-15	Analog output additional damping value (seconds)	Floating pt
64	Write analog output additional damping value	0	Analog output number code			Same as command	
		1-4	Analog output additional damping value	Floating pt			
65	Write analog output range values	0	Analog output number code			Same as command	
		1	Analog output units code				
		2-5	Analog output upper range value	Floating pt			
		6-9	Analog output lower range value	Floating pt			
66	Enter/exit fixed analog output mode	0	Analog output number code			Same as command	
		1	Analog output units code				
		2-5	Analog output level *	Floating pt			
			* "not a number" exits fixed output mode				
67	Trim analog output zero	0	Analog output number code			Same as command	
		1	Analog output units code				
		2-5	Externally-measured analog output level *	Floating pt			
68	Trim analog output gain	0	Analog output number code			Same as command	
		1	Analog output units code				
		2-5	Externally-measured analog output level *	Floating pt			
69	Write analog output transfer function	0	Analog output number code			Same as command	
		1	Analog output transfer function code				
70	Read analog output endpoint values	0	Analog output number code		0	Analog output number code	
					1	Analog output endpoint units code	
					2-5	Analog output upper endpoint value	Floating pt
					6-9	Analog output lower endpoint value	Floating pt
107	Write burst mode transmitter variable (for command 33)	0	Transmitter variable code for slot 0			Same as command	
		1	Transmitter variable code for slot 1				
		2	Transmitter variable code for slot 2				
		3	Transmitter variable code for slot 3				
108	Write burst mode command number	0	Burst mode command number			Same as command	
109	Burst mode control	0	Burst mode control code (0 = exit, 1 = enter)			Same as command	
110	Read all dynamic variables		none		0	Primary Variable units code	
					1-4	Primary Variable value	Floating pt
					5	Second Variable units code	
					6-9	Second Variable value	Floating pt
					10	Third Variable units code	
					11-14	Third Variable value	Floating pt
					15	Fourth Variable units code	
					16-19	Fourth Variable value	Floating pt

HART Communications Status

Each message from a field slave device includes two bytes of status information, which is also referred to as the “response code”. The format of the HART communications status data is shown in this section.

Response Data with Command Response = 1

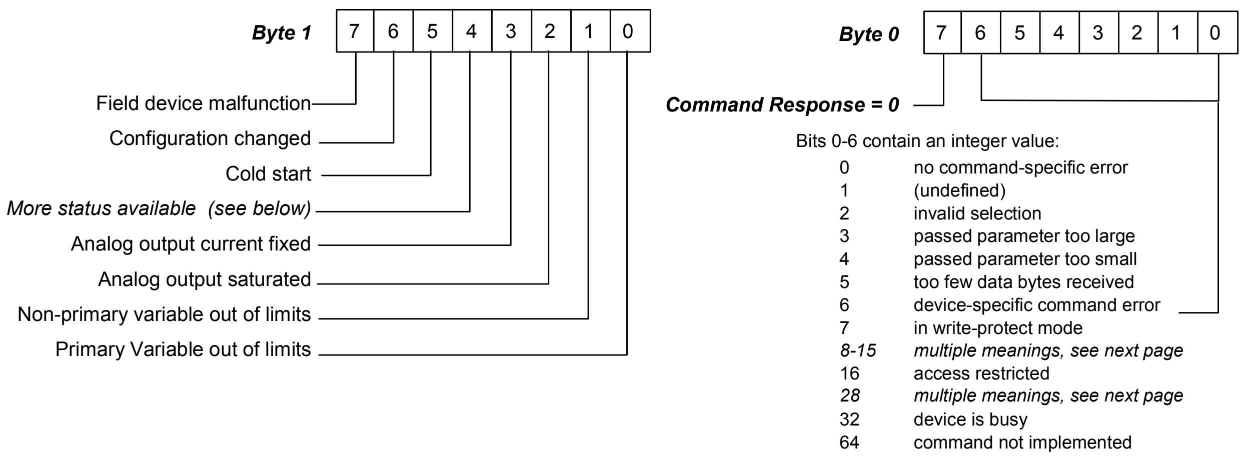
If the Most Significant Bit of the first byte is 1 (Communications Error), an error has occurred in the outgoing HART communication. The rest of the bits individually indicate one or more error conditions:



Error conditions include parity and overrun errors. In addition, a field device will report an overflow of its receive buffer, and any discrepancy between the message content and the received checksum.

Response Data with Command Response = 0

If the Most Significant Bit of the first byte is 0 (Command Response), the outgoing HART communications completed normally. The first byte then contains an integer value (the Command Response code) with the command status. The second byte contains the field device status, indicating the operational state of the slave device:



Field Device Status Codes

The Field Device Status codes in the second byte are explained below.

Field Device Status Code	*Meaning*
Field device malfunction	Measurements may not be correct.
Configuration changed	The configuration has been changed, so the master should check the configuration, and clear the bit by sending Command #38.
Cold start	Set for the first transaction when a field device is powered up.
More status available	The master should issue Command #48 to read more status information.
Analog output fixed	The output has been set to a fixed value for testing. This bit applies only to analog output #1. In a multidrop output device, command #48 may return similar status information for the other outputs.
Analog output saturated	Analog output #1 is out of range.
Primary variable out of limits	The primary measurement is outside the sensor operating limits. The analog signal and the digital signal read by HART commands may be incorrect.
Non-primary variable out of limits	A non-primary measurement is outside the sensor operating limits. The analog signal and the digital signal read by HART commands may be incorrect. Command #48 may provide more information.

Command Codes with Multiple Meanings

The interpretation of Command Codes 8-15 and 28 in the first Command Response byte depends on the command that was issued.

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

Function Blocks to Read HART Data

Proficy Machine Edition release 5.5 includes two custom HART function blocks for use in ladder logic application programs. These function blocks can be used to assign variable names to HART data inputs or on HART Get Device Information response data, beginning at word 3.

- DYN_HART_STRUCT interprets the first 36 bytes of the HART data map for a channel. This function block reads on demand the same data that would be automatically scanned if the module were configured for *HART Data Scan Control* set to “Dynamic Data”.
- ALL_HART_STRUCT interprets the entire HART data for a channel, including the data interpreted by the DYN_HART_STRUCT function block. This function block reads on demand the same data that would be automatically scanned if the module were configured for *HART Data Scan Control* set to “All Data”.

Instead of being read into an assigned CPU reference address, the data read by these Function Blocks is placed into a reference address that is defined by the Function Block.

The HART function blocks are located in the Toolchest under the folder "HART Utilities". To create an instance of one of these function blocks, drag and drop the desired function from the Toolchest folder into LD logic.

DYN_HART_STRUCT

The DYN_HART_STRUCT function block interprets the first 36 bytes on the data (offsets 0x0000 to 0x0023) of the HART data map for a channel. See the example in this section. Use of this function block is not required. It only provides a mechanism to assign variable names to the HART data,

When the DYN_HART_STRUCT function executes (receives power flow), each of the HART variables in reference memory is assigned to DYN_HART_STRUCT instance variables. The instance variable names closely match the names listed in the table, The instance variables can be used as input to other functions, or for debugging purposes. The entire instance can be added to a Data Watch window so that HART data is easily viewed.

The instance data variables are only updated when the function block receives power flow. An instance of the function block is needed for each HART channel on which automatic assignment is desired.

Parameter types:

IN - Type = WORD, Length = 18, Pass-by = Reference

Data Obtained with the DYN_HART_STRUCT Function Block

Byte Offset		***Field Description***	***Data Type***
Begin	***End***		
0x0000	0x0003	HART Primary Variable, CMD#3, Bytes 5-8	REAL
0x0004	0x0007	HART Secondary Variable CMD#3, Bytes 10-13	REAL
0x0008	0x000b	HART Tertiary Variable CMD#3, Bytes 15-18	REAL
0x000c	0x000f	HART Fourth Variable CMD#3, Bytes 20-23	REAL
0x0010	0x0013	Slot 0 value. CMD#33, Bytes 2-5	REAL
0x0014	0x0017	Slot 1 value CMD#33, Bytes 8-11	REAL
0x0018	0x001b	Slot 2 value CMD#33, Bytes 14-17	REAL
0x001c	0x001f	Slot 3 value CMD#33, Bytes 20-23	REAL
0x0020		HART communication status byte from the last HART command response. See page 11-48.	BYTE
0x0021		HART device status byte from the last HART command response. (Dynamic Data)	BYTE
0x0022		Spare for alignment. (Dynamic Data)	BYTE*2

ALL_HART_STRUCT

The ALL_HART_STRUCT function block interprets the entire HART data map for a channel. This is the same data obtained by the DYN_HART_STRUCT block, plus the additional data shown below.

Parameter types: IN - Type = WORD, Length = 88, Pass-by = Reference

Additional Data Obtained with the ALL_HART_STRUCT Function Block

Byte Offset		***Field Description***	***Data Type***
Begin	***End***		
0x0024		HART device Manufacturer ID. CMD#0, Byte 1	BYTE
0x0025		HART device type code. CMD#0, Byte 2	BYTE
0x0026		Minimum number of preambles device requires. CMD#0, Byte 3	BYTE
0x0027		HART Universal command code. CMD#0, Byte 4	BYTE
0x0028		HART Transmitter specific revision. CMD#0, Byte 5	BYTE
0x0029		HART device software revision number. CMD#0, Byte 6	BYTE
0x002A		HART device hardware revision number. CMD#0, Byte 7	BYTE
0x002B		HART flags. CMD#0, Byte 8	BYTE
0x002C	0x002F	HART device ID number. CMD#0, Byte 9-11	BYTE*4
0x0030	0x0037	8 character device tag. CMD#13, Bytes 0-5 in unpacked ASCII.	BYTE*8
0x0038	0x0047	Device Descriptor. CMD#13, Bytes 6-17 in unpacked ASCII	BYTE*16
0x0048		HART Primary Variable Units. CMD#3, Byte 4	BYTE
0x0049		HART Secondary Variable Units. CMD#3, Byte 9, 0 if not present.	BYTE
0x004a		HART Tertiary Variable Units. CMD#3, Byte 14, 0 if not present.	BYTE
0x004b		HART Fourth Variable Units. CMD#3, Byte 19, 0 if not present.	BYTE
0x004c		HART Primary Variable Code. CMD#50, Byte 0	BYTE
0x004d		HART Secondary Variable Code. CMD#50, Byte 1	BYTE
0x004e		HART Tertiary Variable Code. CMD#50, Byte 2	BYTE
0x004f		HART Fourth Variable Code. CMD#50, Byte 3	BYTE
0x0050		Units code for range parameter. CMD#15, Byte 2	BYTE
0x0051	0x0053	Spare for alignment	BYTE*3
0x0054	0x0057	Low transmitter range for analog signal in engineering units. CMD#15, Bytes 3-6	REAL
0x0058	0x005b	High transmitter range for analog signal in engineering units. CMD#15, Bytes 7-10	REAL
0x005c		Slot 0 units code. CMD#33, Byte 1	BYTE
0x005d		Slot 1 units code. CMD#33, Byte 7	BYTE
0x005e		Slot 2 units code. CMD#33, Byte 13	BYTE
0x005f		Slot 3 units code. CMD#33, Byte 19	BYTE
0x0060		Slot 0 variable code. CMD#33, Byte 0	BYTE
0x0061		Slot 1 variable code. CMD#33, Byte 6	BYTE
0x0062		Slot 2 variable code. CMD#33, Byte 12	BYTE
0x0063		Slot 3 variable code. CMD#33, Byte 18	BYTE
0x0064	0x0083	32 character message. CMD#12, Bytes 0-23 unpacked ASCII.	BYTE*32
0x0084	0x0087	Stored date in the field device. CMD#13, Bytes 18-20.	BYTE*4
0x0088	0x008b	The final assembly number is used for identifying the material and electronics that comprise the field device. CMD#16, Bytes 0-2	BYTE*4
0x008c	0x00a4	The extended status returned by HART command 48.	BYTE*25
0x00a5	0x00af	Spare	BYTE*11

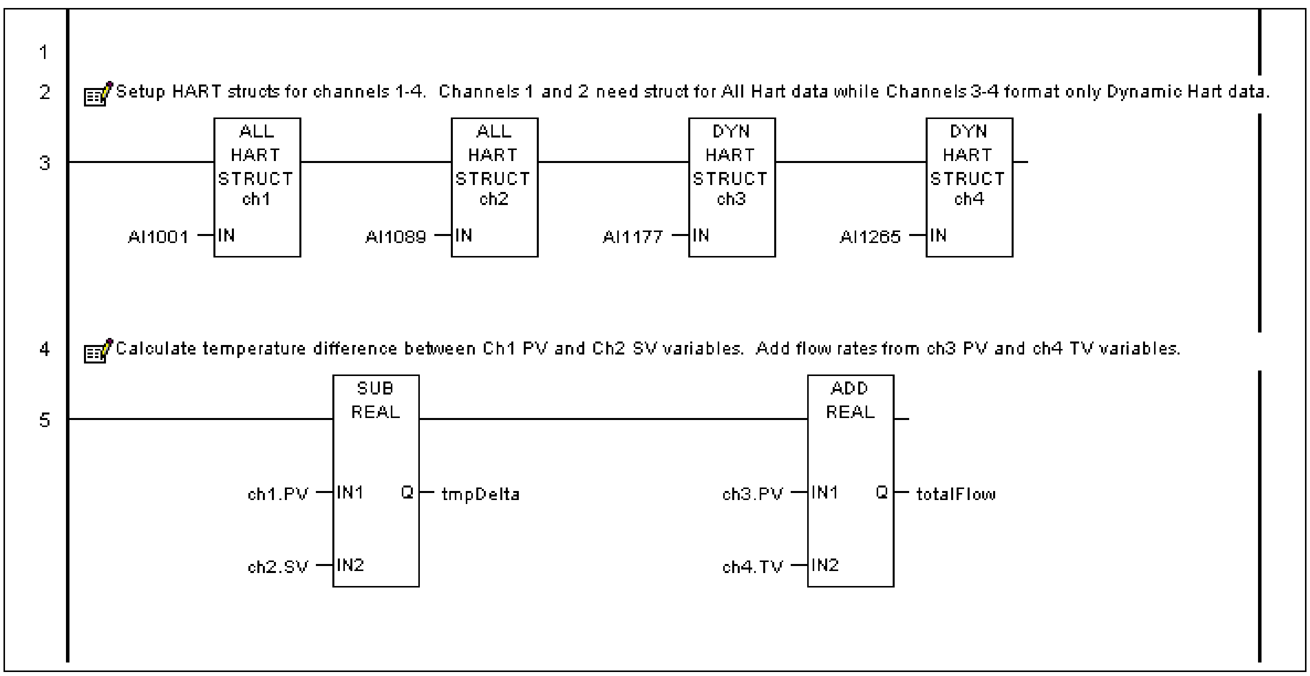
Example 1: ALL_HART_STRUCT and DYN_HART_STRUCT

This example uses the following HART module hardware configuration parameters:

- HART Data Scan Control is set to: All Data
- HART Data Reference Address is configured as: %AI1001

The example shows ALL_HART_STRUCT for channels 1 and 2 on a HART module, and DYN_HART_STRUCT for channels 3 and 4 on the same module. The math function blocks in the example show how the structure instance variables can be used on the HART data.

When using either ALL_HART_STRUCT or DYN_HART_STRUCT, the instance data should be made global. This is done by adding ",g" to the end of the instance name when it is created (for example, "ch1,g" creates a global instance named "ch1"). If the instance data is not global, the instance can only be used in the program block where the ALL_HART_STRUCT or DYN_HART_STRUCT function block is executed.



Converting HART Data to / from RX3i Format

When using HART Pass-Thru COMMREQ (command 2) only, HART data must be both byte-packed and in big-endian format as defined by the HART Specification. Because PACSystems CPUs use little-endian format, floating point values and ASCII data must be reformatted by the program logic.

Note that this conversion is not required when using COMMREQ 1 or for HART data that is automatically scanned.

Two function blocks in the Proficy Machine Edition release 5.5 toolchest can be used to pack/unpack HART ASCII data:

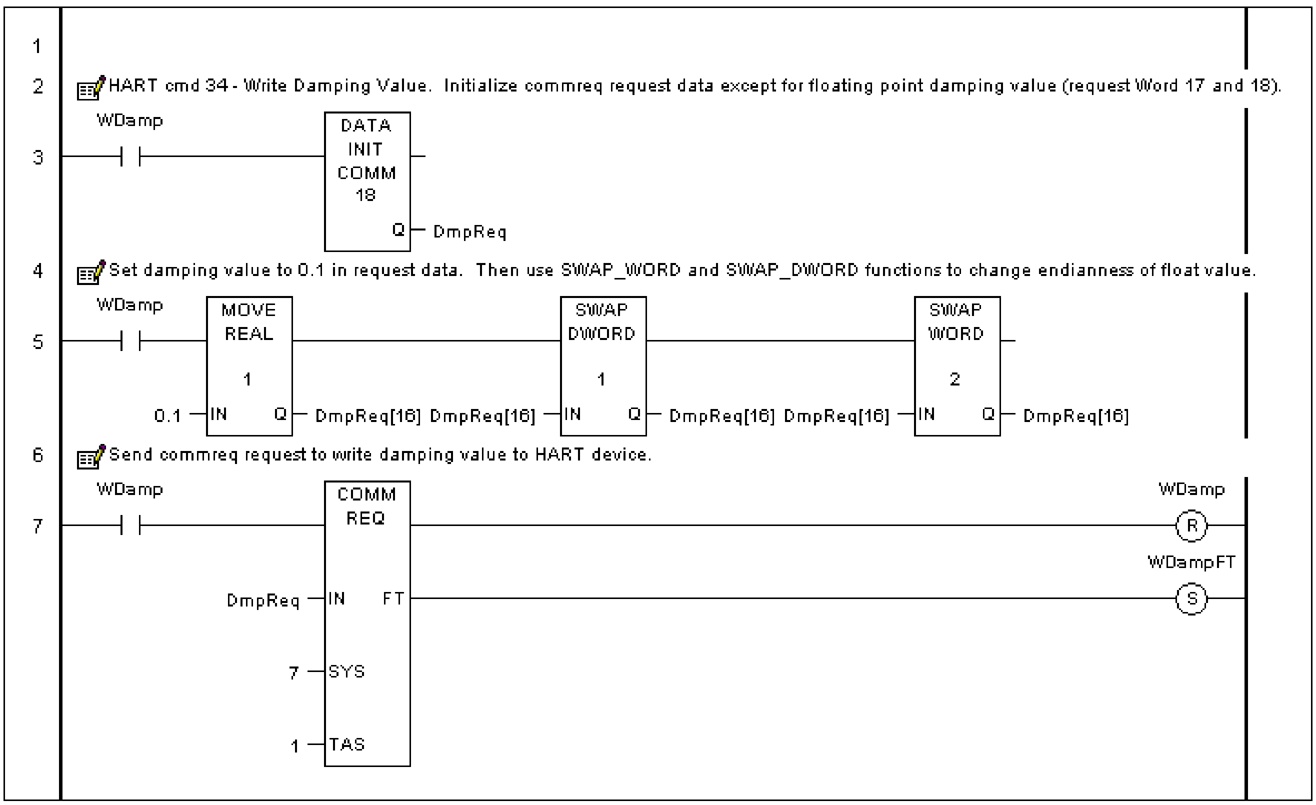
- ASCII_PACK prepares ASCII data before sending it to a HART module.
- ASCII_UNPACK can be used to unpack ASCII characters returned by a HART device.

These are described in the next section.

Converting Floating Point Data (Endian Flip)

Floating point values that begin at word 17 of a Pass-Thru Request must converted TO big-endian format. Floating point values that begin at word 7 of the reply must converted FROM big-endian format.

The basic procedure is to pass any HART float values through the SWAP_DWORD (size = 1) and SWAP_WORD (size = 2) functions, as shown below. This swaps the words within the float dword, and swaps the bytes within the two float words. In this example shows how to format floating point data prior to sending HART command 34, "Write Damping Value".



ASCII_PACK

ASCII_PACK prepares ASCII data before sending it to a HART module using COMMREQ2 (HART Pass-Thru Request). All of the function block parameters are either Type = WORD, Pass-By = Value or word arrays of Type = WORD and Pass-by = Reference. All word-based reference memory types and symbolics can be used.

It is possible to use data of a different type for the array data (for example, using a byte array of data as input to the ASCII_PACKED and ASCII_UNPACKED blocks), although Machine Edition will issue a warning during verification. In the example mentioned, the size of the input byte array would need to be at least as large as the data size of the function block parameter word array.

Parameter types:

IN - Type = WORD, Length = 16, Pass-by = Reference

NUM - Type = WORD, Length = 1, Pass-by = Value

Q - Type = WORD, Length = 12, Pass-by = Reference

Example Function Block: ASCII_PACK

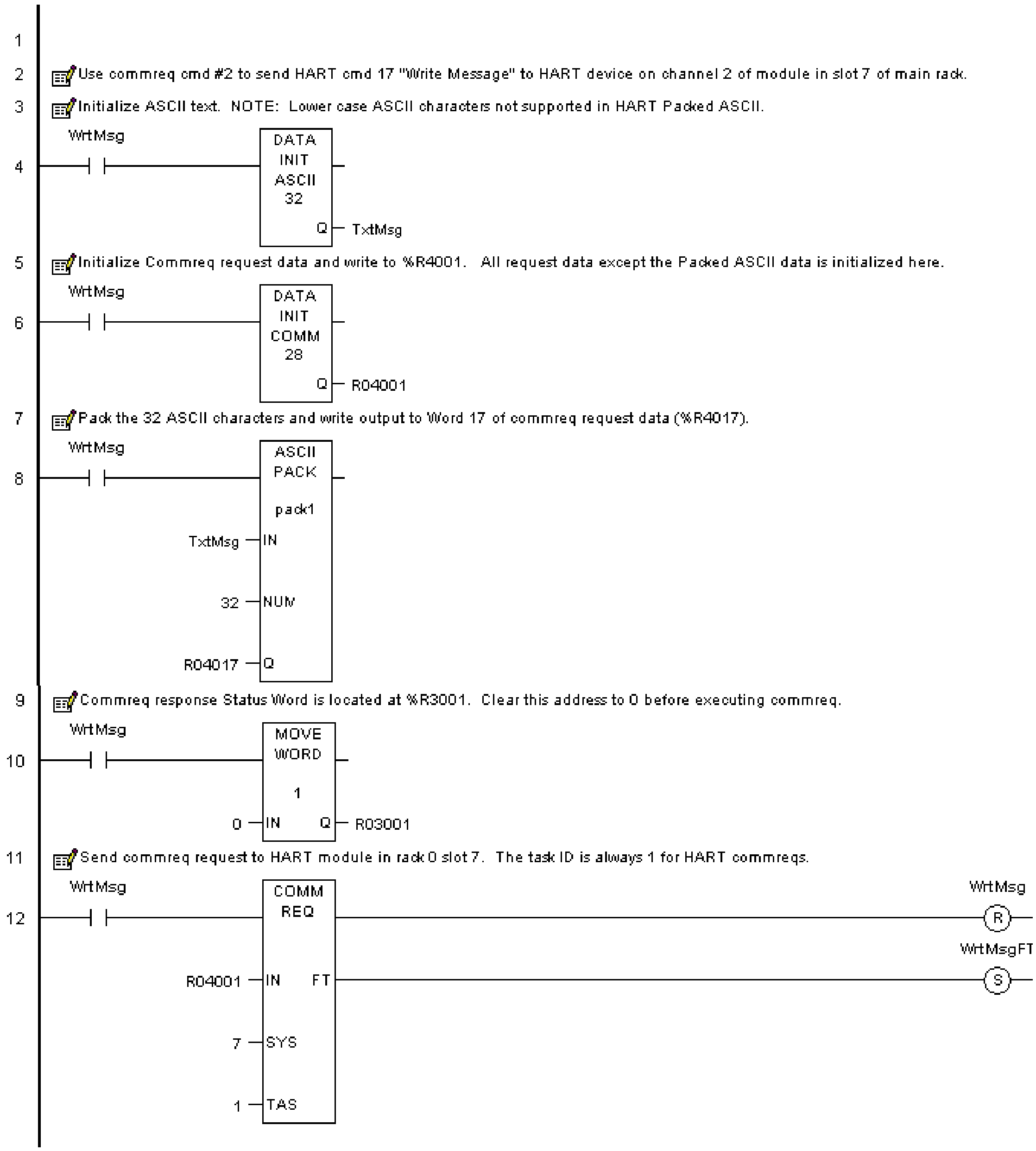
This example shows how ASCII_PACK can be used to prepare ASCII data before using a COMMREQ to send it to a HART module. First, the logic sets up the data that will be used by the COMMREQ. The ASCII_PACK function block packs the 32 ASCII characters of the message and also places that into the COMMREQ data area. After clearing the COMMREQ status to 0, the logic uses a COMMREQ to send HART command 17 "Write Message" to the device.

HART ASCII format packs data into 6 bits per character, 4 characters per 3 bytes. This chart shows the format with the most significant hex digits in the rightmost column and the lowest row.

Less significant

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
2	SP	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

More significant


1
2 Use commreq cmd #2 to send HART cmd 17 "Write Message" to HART device on channel 2 of module in slot 7 of main rack.
3 Initialize ASCII text. NOTE: Lower case ASCII characters not supported in HART Packed ASCII.
WrtMsg
4
DATA INIT ASCII 32
Q TxtMsg
5 Initialize Commreq request data and write to %R4001. All request data except the Packed ASCII data is initialized here.
WrtMsg
6
DATA INIT COMM 28
Q R04001
7 Pack the 32 ASCII characters and write output to Word 17 of commreq request data (%R4017).
WrtMsg
8
ASCII PACK
pack1
TxtMsg IN
32 NUM
R04017 Q
9 Commreq response Status Word is located at %R3001. Clear this address to 0 before executing commreq.
WrtMsg
10
MOVE WORD
1
0 IN Q R03001
11 Send commreq request to HART module in rack 0 slot 7. The task ID is always 1 for HART commreqs.
WrtMsg
12
COMM REQ
R04001 IN FT
7 SYS
1 TAS
WrtMsg
R
WrtMsgFT
S

ASCII_UNPACK

ASCII_UNPACK can be used to unpack ASCII characters returned by a HART device in Pass-Thru Reply data send in response to COMMREQ 2.

All of the function block parameters are either Type = WORD, Pass-By = Value or word arrays of Type = WORD and Pass-by = Reference. All word-based reference memory types and symbolics can be used.

Parameter types:

IN - Type = WORD, Length = 12, Pass-by = Reference
NUM - Type = WORD, Length = 1, Pass-by = Value
Q - Type = WORD, Length = 16, Pass-by = Reference

Example Function Block 3: ASCII_UNPACK:

In this example, after initializing the COMMREQ completion status, the ladder logic sends COMMREQ command #12, "Read Message" to the HART module in Rack 0, Slot 7. When the status = complete and the Response Byte Count =24, showing that all the data is present, the ASCII_UNPACK function block unpacks the HART ASCII data.