# HART® Field Device Specification

# Fisher™ FIELDVUE™ DLC3010 Digital Level Controller (Supported Product)

| HART Revision | Device Type | Device Revision | Firmware Revision |
|---|---|---|---|
| HART 5 | 04 | 1 | 8 |

## Contents




W7977-2

Product removed from sale April 2022

# Introduction

## Scope

The Fisher DLC3010 digital level controller targeted compliance with HART Protocol Revision 5.2. Additionally, an effort was made to provide support for the proposed revision 5/6 compatibility rules. This document provides all the device-specific features and HART communications protocol implementation details. The functionality of this field device is described sufficiently to allow its proper application in a process and its complete support in HART-capable host applications.

## Purpose of this document

This document provides a description of the field device from a HART Communication perspective. Additional product information can be found in the DLC3010 product literature, available from your Emerson sales office.

## Who should use this document

The information contained herein is intended for use as a technical reference for HART-capable host application developers, systems integrators, and knowledgeable end-users. It also provides functional specifications (e.g., commands, enumerations, and performance requirements) used during field device development, maintenance, and testing. Users of this document must be fully trained in HART Protocol requirements and terminology.

## Abbreviations and definitions

| | |
|---|---|
| **Additional Device Status** | Status information returned by Command 48 |
| **Byte** | An 8-bit unsigned integer |
| **Common Table <n>** | A reference to a table in HCF_SPEC-183 (FCG TS20183) Common Tables Specification |
| **Configuration Variables** | Variables which represent nonvolatile values of manufacturing-initialized data or user-specified configuration information. These variables cannot be accessed via Universal or Common Practice Commands. |
| **Device Variable** | Uniquely defined data items within a field device, containing process-related information. They are assigned consecutive code numbers starting with zero. |
| **DLC** | DLC3010 digital level controller product |
| **Dynamic Variable** | A Device Variable mapped to a slot in the set of HART commands that support potential analog channels in the device. Only the first slot is required to have an associated analog channel. |
| **Enumeration** | A pre-defined set of values or text |
| **Float** | IEEE 754 floating point format |
| **HART** | Highway Addressable Remote Transducer |
| **LCD** | Liquid Crystal Display |
| **Lift-Off Voltage** | Minimum supply voltage required at device terminals to guarantee correct behavior, (including HART communication), during both normal operation and while indicating a malfunction |

| | |
|---|---|
| **NVM** | Non-volatile memory |
| **Packed** | Packed ASCII, a special form of characters defined by HART in which four 6-bit ASCII characters are packed into three bytes |
| **RTD** | Resistance Temperature Detector |
| **STO** | Slave Time Out. The time allowed for a slave device to begin its transmission, defined in HCF_SPEC-081 (FCG TS20081) and tested in HCF_TEST-001 section 7.24 DLL024. |
| **Word** | A 16-bit unsigned integer |
| **Uint <n>** | Unsigned integer with bit length n |

## Reference Documentation

HART Smart Communications Protocol Specification Revision 5.0; a group of documents specifying the HART Communication Protocol, physical layers, and Data Link Layers as defined by the FieldComm Group™.

FIELDVUE DLC3010 Digital Level Controller Instruction Manual, D102748X012

FIELDVUE DLC3010 Digital Level Controller Quick Start Guide, D103214X012

Industrial Platinum Resistance Thermometers and Platinum Temperature Sensors, IEC 60751, International Electrotechnical Commission

# Device Identification

| | | | | | |
|---|---|---|---|---|---|
| **Manufacturer Name** | Fisher Controls | | **Model Name(s)** | DLC3010 | |
| **Manufacture ID Code** | 19 | (13 Hex) | **Device Type Code** | 04 | (04 Hex) |
| **HART Protocol Revision** | 5.2 | | **Device Revision** | 1 | |
| **Number of Device Variables** | 5 (effectively 3, as variables 0, 1 and 2 cannot coexist) | | | | |
| **Physical Layers Supported** | FSK (Bell 202 Current) | | | | |
| **Physical Device Category** | Transmitter (two-wire), Non-DC-isolated Bus Device | | | | |

# Product Overview

DLC3010 digital level controllers are used with level sensors to measure liquid level, level of the interface between two liquids, or liquid specific gravity (density). Changes in level or specific gravity exert a buoyant force on a displacer, which rotates the torque tube shaft. This rotary motion is applied to the digital level controller, transformed to an electrical signal and digitized. The digital signal is compensated and processed per user configuration requirements, and converted to a 4-20 mA analog electrical signal. The resulting current output signal is sent to an indicating or final control element. The name plate is located on the top of the DLC3010 assembly and indicates the model name, individual product serial number, and any applicable third party approvals.

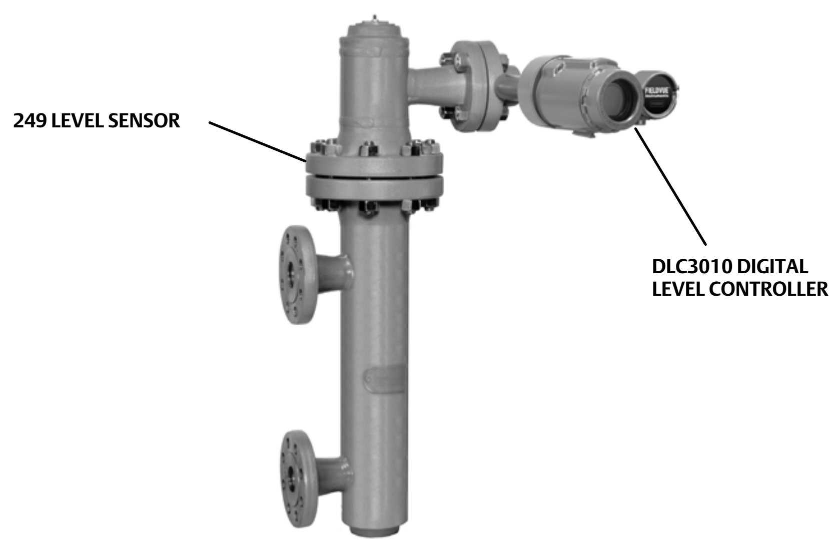
# Product Interfaces

## Process Interface

### Primary Variable Sensor Interface

DLC3010 digital level controllers mount on a wide variety of caged and cageless 249 level sensors. Mounting adaptors are available to allow using them on other manufacturers' displacer type level sensors. The field device is provided with a lever that carries a set of magnets across a Hall-Effect sensor to transform angular position into voltage. The lever is coupled to the buoyancy sensor’s pilot shaft by a clamping bolt and nut. Nominal design rotation of the buoyancy sensor for a full span change of water level at room temperature is 4.4°. To best utilize the accuracy of the transmitter, the amount of input rotation used should be close to this range. For applications that would develop a very small proportional band with standard hardware, (e.g., interface level measurements where the difference between the densities of the two phases is quite small), custom sensor configurations may be specified to improve the mechanical gain.

The available lever travel in the digital level controller is approximately ±6° from the neutral or ‘locked’ position. This allows right- or left-hand-mounted sensors to be used with out mechanical changes to the transmitter. It also allows the digital level controller to be used with sensors having mechanical gain slightly higher than nominal by physically coupling at the center of sensor travel instead of at the lowest process condition.




## Process Temperature Interface

An external 100 Ohm Platinum RTD of 2-wire or 3-wire configuration may be installed to provide process temperature instrumentation. The terminals for the RTD are in the lower bank of the instrument terminal box, and are labeled “Rs”, “R1”, and “R2”, from left to right. “Rs” and “R1” are used for the two wires from the same node of a 3-wire RTD, and must be shorted with a jumper when a two-wire RTD is employed. Refer to the Instruction Manual for additional installation details. When the RTD is installed, configured, and calibrated, and a table of density versus temperature is entered by the user, this input will be used to drive density compensation for the level calibration. If the RTD is not installed, the compensation tables can also be driven by a manually-entered value of process temperature. To disable the density temperature-compensation, the compensation table length is reduced to 1 element.

The process temperature value derived from the RTD may be checked against user-defined alarm thresholds to indicate when the sensor is operating outside of recommended temperature limits.

# Electronics Temperature Interface

An internal temperature sensor mounted near the Hall Sensor is utilized to drive factory-configured temperature compensation for magnetic flux and Hall-effect sensitivity variations. It may also be checked against user-defined alarm thresholds to indicate when the transmitter is operating outside of recommended temperature limits.

# Host interface

## Analog Output: Primary Variable

When available terminal voltage is above the Lift-Off Voltage, the DLC3010 acts as the current source in a two-wire 4-to-20 mA current loop (in point-to-point mode) or draws 4 mA fixed current (in multi-drop mode). This output is provided in the DLC3010's terminal box at two terminals marked "+" and "-". Refer to the Quick Start Guide for connection details. In point-to-point mode, the digital value of the primary variable is mapped to the 4-20 mA signal by the Range Values.

| | Direction | Value (% Range) | Value (mA or V) |
|---|---|---|---|
| **Saturation Limits** | High | PV > +103.13% | 20.5 mA |
| | Low | PV < -1.25% | 3.8 mA |
| **Alarm Indication** | High* | | 22.5 mA |
| | Low* | | 3.7 mA |
| **Maximum Current** | | | 22.5 mA |
| **Multi-Drop Current Draw** | | | 4.0 mA |
| **Lift-Off Voltage** | | | 12 V. |

* User must select one or the other of the alarm indication outputs with a hardware jumper. They are mutually exclusive.

# HART Digital Interface

When available terminal voltage is above the Lift-Off Voltage, the DLC3010 can communicate digitally via Bell 202 FSK HART protocol. This interface is available at test clips in the DLC3010's terminal box on the two terminals marked "+" and "-", or across any convenient impedance on the loop that is sufficient to meet the HART signaling requirements. Refer to the Quick Start Guide for connection details.

# Local Interfaces, Jumpers, and Switches

## Local Displays

A removable Liquid Crystal Display (LCD) assembly is provided. It displays PV %Range on a circular "bar-graph", and different combinations of digital information such as PV %Range, PV in engineering units, Process Temperature, etc. Display symbols or text are provided for:

- Write-lock (key symbol)
- Display numeric field overflow (OFLOW)
- Hardware failure (Hdwr FL)

### Local Jumpers

A fail-mode jumper is provided on the lower face of the LCD assembly (on the upper right quadrant of the electronics module when the LCD is not used). This jumper is internal to the electronics compartment and may only be accessed by removing the main cover. It allows the user to select either the High or Low alarm indication documented in the table above. Only one alarm indication value can be made available during operation. The fault conditions that trigger the alarm are documented in table 2a.

# Device Variables

These variables represent measurements taken by the device (see table 5), and are all in float format. Their values are not directly exposed by any standard HART command. However, they are the set of internal variables from which the Dynamic Variables are selected. Only one of the Liquid Level, Interface Level, or Liquid Density measurements may be computed in a given configuration, so assigning one of these to PV sets up the structure of the measurement algorithm. Temperature variable selections are permanently allocated to SV and TV slots, so their indices are used primarily for units processing. Process temperature is only functional as TV if the RTD is installed, otherwise it is a fixed parameter entered by the user.

## Device Variable 0 - Liquid Level

When Liquid Level is assigned as PV, the process value is derived from the measured rotation of the sensor’s torque tube by using calibration data to convert the change in rotation to a force measurement, then applying displacer weight, displacer volume, and fluid density configuration data to convert force to liquid level. This variable is not computed when it is not assigned as PV.

## Device Variable 1 - Interface Level

When the Interface Level is assigned as PV, the process value is derived from the measured rotation of the sensor’s torque tube by using calibration data to convert the change in rotation to a force measurement, then applying displacer weight, displacer volume, upper fluid density, and lower fluid density configuration data to convert force to interface level. This variable is not computed when it is not assigned as PV.

## Device Variable 2 – Liquid Density

When the Liquid Density is assigned as PV, the process value is derived from the measured rotation of the sensor’s torque tube by by using calibration data to convert the change in rotation to a force measurement, then applying weight and volume configuration data to convert force to fluid density. This variable is not computed when it is not assigned as PV.

## Device Variable 3 – Process Temperature

When an RTD installed in the process fluid is wired to the device and assigned as the Process Temperature Source, the Process Temperature variable is derived from the measured resistance of the RTD via a table related to IEC 60751. A 60-second lag, factory calibration, and user offset adjustments are applied to the signal before reporting.

When “Manual Entry” is assigned as the Process Temperature Source, this variable simply reports a fixed user-entered value.

## Device Variable 4 – Electronics Temperature

This variable is derived from the voltage drop across a semiconductor junction in a chip mounted in the transducer housing. A 60-second lag and factory or user offset adjustments are applied to the signal before reporting.

# Dynamic Variables

Three Dynamic Variables are implemented. The PV is user-selectable to one of the first three Device Variables via Command 51.

| | **Default Meaning** | **Units** |
|---|---|---|
| PV* | Liquid Level | See table 6 |
| SV | Instrument Temperature | See table 6 |
| TV | Process Temperature | See table 6 |
| FV | Not Used | N/A |
| * User selectable | | |

# Status Information

## Device Status

The Field Device Status Byte (see table 2a) is the only status byte defined in the HART 5 protocol. The order and meaning of each of the eight bits within the byte are fixed by the protocol. This byte is one of the status bytes included with each HART response. It is not part of the Command 48 data.

## Extended Device Status

This byte was not defined in the HART 5 specification, so it is not supported.

## Additional Status Integrity Bytes

Three Additional Status bytes providing details of internal monitor states are returned in the Command 48 response. Refer to table 2b for definitions of the bits.

# Universal Commands

The following HART 5.0 Universal Commands are implemented in the DLC3010 firmware:

Command 0: Read Unique HART Identifier
Command 1: Read Primary Variable
Command 2: Read PV Current and Percent Range
Command 3: Read Dynamic Variables and PV Current
Command 6: Write Polling Address
Command 11: Read Unique Identifier with Tag
Command 12: Read Message
Command 13: Read Tag, Descriptor, Date
Command 14: Read PV Sensor Info
Command 15: Read PV Output Info
Command 16: Read Final Assembly Number
Command 17: Write Message
Command 18: WriteTag, Descriptor, Date

---

**Note**

At HART 5, Commands 38 and 48 were in the Common Practice group.

---

Any command can return any of the following response codes:

| | |
|---|---|
| 0 | No command specific errors |
| 5 | Incorrect Byte Count |
| 8 | Warning - value written was truncated (write commands only) |
| 32 | Busy |

In addition, some commands may return additional error codes. See the "Command Specific Response Codes" part of the command description for additional codes. One of the error codes associated with writing data to the instrument is:

| | |
|---|---|
| 7 | In Write Protect Mode |

A more complete list is available in table 1. All commands will check for the required number of data bytes and return Incorrect Byte Count if too few bytes are received. If too many bytes are received for the given command, the extra bytes are ignored. This is the behavior defined by the HART specification, which allows for command expansion with backward compatibility. If additional bytes are added to any command, an older instrument will still accept the command with only the original bytes used and will ignore any of the additional bytes.

# Command 0: Read Unique Identifier

Returns identity information about the field device in HART 5 identity format, including: Device Type, revision levels, and Device ID. This command is implemented by a field device in both Short and Long Frame Formats. Command 0 is the only command that may respond to a short frame address. The Device Type Code will be returned in the expanded three-byte format. ("254", Manufacturer Identification Code, Manufacturer's Device Type Code). The combination of Manufacturer Identification Code, Manufacturer's Device Type Code, and Device Identification Code make up the Unique Identifier required for the Extended Frame Format of the Data Link Layer.

| | **Byte** | **Format** | **Description** | **Returned Value** |
|---|---|---|---|---|
| Request Data bytes | None | | | |
| | 0 | Enum | Expansion code “254” | 254 |
| | 1 | Enum | Manufacturer Identification code [Common Table 8] | 19 |
| | 2 | Enum | Manufacturer's Device Type code [Common Table 1] | 4 |
| | 3 | Uint8 | Number of preambles required for master to slave request, including those required for message detection | 5 |
| | 4 | Uint8 | Revision level of the universal command document implemented by this device. Levels 254 and 255 are reserved. | 5 |
| | 5 | Uint8 | Revision level of the device - specific document implemented. Levels 254 and 255 are reserved | 1 |
| | 6 | Uint8 | Software revision level of this device | 8 |
| | 7.7-7.3 | Uint5 | Hardware revision level. Does not necessarily trace component changes | 1 |
| | 7.2-7.0 | Enum | Physical Signaling Code [Common Table 10] | 0 |
| | 8 | Bits | Flags [Common Table 11] | 0 |
| | 9-11 | Uint24 | (Unique) Device Identification Number | |
| | Code | Class | Description | |
| Response Codes | 6 | Error | Device-Specific Command Error | |

# Command 1: Read Primary Variable

The Primary Variable is returned along with its Units Code.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | 0 | Enum | Primary Variable Units (table 6, DLC3010 Unit Codes) |
| | 1-4 | Float | Primary Variable |
| | **Code** | **Class** | **Description** |
| Response Codes | 6 | Error | Device-Specific Command Error |
| | 8 | Warning | Update Failure |
| | 16 | Error | Access Restricted |

# Command 2: Read Loop Current and Percent of Range

The Loop Current should match the current that would be measured by a milliammeter in series with the field device. The actual implementation reports the commanded value driving the output Digital to Analog converter.

The Percent of Range signal follows the PV as mapped by URV and LRV in normal operation. However, the value reported by Command 2 in the DLC3010 at firmware 8 is derived from the current command, thus reflecting saturation limits and alarm values, instead of following PV out to Sensor Limits.

| | Byte | Format | Description | Returned Value |
|---|---|---|---|---|
| Request Data Bytes | None | | | |
| Response Data Bytes | 0-3 | Float | Primary Variable Loop Current (milliamperes) | |
| | 4-7 | Float | Primary Variable Percent of Range (%) | |
| | **Code** | **Class** | **Description** | |
| Response Codes | 6 | Error | Device-Specific Command Error | |
| | 8 | Warning | Update Failure | |

# Command 3: Read Dynamic Variables and Loop Current

The Loop Current should match the current that would be measured by a milliammeter in series with the field device. The actual implementation reports the commanded value driving the output Digital to Analog converter.

The remaining Response Data include the PV with its Units Code, Process Temperature and Electronics Temperature with Temperature Units Codes.

| | Byte | Format | Description | Returned Value |
|---|---|---|---|---|
| Request Data Bytes | None | | | |
| Response Data Bytes | 0-3 | Float | Primary Variable Loop Current (mA) | |
| | 4 | Enum | Primary Variable Units Code (table 6) | |
| | 5-8 | Float | Primary Variable | |
| | 9 | Enum | Secondary Variable Units Code (table 6) | |
| | 10-13 | Float | Secondary Variable (Electronics Temperature) | |
| | 14 | Enum | Tertiary Variable Units Code (table 6) | |
| | 15-18 | Float | Tertiary Variable (Process Temperature) | |
| | 19 | Enum | Not Used | 250 |
| | 20-23 | Float | Not Computed | NaN |
| | **Code** | **Class** | **Description** | |
| Response Codes | 6 | Error | Device-Specific Command Error | |
| | 8 | Warning | Update Failure | |

# Command 6: Write Polling Address

This is a Data Link Layer Management Command.

This command writes the Polling Address to the field device. The address is used to control the Primary Variable Analog Output and provide a means of device identification in Multidrop installations.

The Primary Variable Analog Output responds to the applied process only when the Polling Address of the device is set to 0. When the address assigned to a device is in the range from 1 through 15, the Analog Output is Not Active and does not respond to the applied process. While the Analog Output is Not Active, the Analog Output is set to its minimum; the Transmitter Status Bit #3, Primary Variable Analog Output Fixed, is set; and the Upscale/Downscale Alarm is disabled. If the Polling Address is changed back to 0, the Primary Variable Analog Output will become Active and respond to the applied process.

In the HART 5 specification, no Read Command was provided for Polling Address. A Device-Specific Command is used to acquire its value for display in the interface.

| | Byte | Format | Description | |
|---|---|---|---|---|
| Request Data bytes | 0 | Uint8 | Polling Address | |
| | | | 0 | Analog Output Active |
| | | | 1 – 15 | Analog Output Not Active |
| | | | 16 - 255 | Invalid |
| Response Data Bytes | 0 | Uint8 | Polling Address | |
| | | | 0 | Analog Output Active |
| | | | 1 – 15 | Analog Output Not Active |
| | | | 16 - 255 | Invalid |
| | **Code** | **Class** | **Description** | |
| Response Codes | 2 | Error | Invalid Selection | |
| | 5 | Error | Too Few Data Bytes Received | |
| | 6 | Error | Device-Specific Command Error | |
| | 7 | Error | In Write-Protect Mode | |

# Command 11: Read Unique Identifier Associated with Tag

This command returns the Expanded Device Type code, revision levels, and Device Identification Number from a device containing the requested Tag. This command is unique in that no response will be made unless the Tag matches that of the device.

The Device Type Code will always be returned in expanded 3-byte format: "254", Manufacturer Identification Code, Manufacturer's Device Type code.

| | Byte | Format | Description | Returned Value |
|---|---|---|---|---|
| Request Data bytes | 0-5 | Packed | Tag | |
| Response Data Bytes | 0 | Enum | Device Type Code for Expansion | 254 |
| | 1 | Enum | Manufacturer Identification Code | 19 |
| | 2 | Enum | Device Type Code | 4 |
| | 3 | Uint8 | Number of Preambles required for request from Master to Slave, including those required for message detect | 5 |
| | 4 | Uint8 | Revision Level of the Universal Command Specification implemented by this device | 5 |
| | 5 | Uint8 | Revision level of the device - specific document implemented. | 1 |
| | 6 | Uint8 | Software revision level of this device | 8 |
| | 7.7-7.3 | Uint5 | Hardware revision level. Does not necessarily trace component changes | 1 |
| | 7.2-7.0 | Enum3 | Physical Signaling Code [Common Table 10] | 0 |
| | 8 | Bits | Flags [Common Table 11] | 0 |
| | 9-11 | Uint24 | Device Identification Number | |
| | **Code** | **Class** | **Description** | |
| Response Codes | 6 | Error | Device-Specific Command Error | |
| | | | | |

# Command 12: Read Message

This command reads a user-defined message contained within the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | None | | |
| Response Data Bytes | 0-23 | Packed | Message String |
| | **Code** | **Class** | **Description** |
| Response Codes | 6 | Error | Device-Specific Command Error |
| | | | |

# Command 13: Read Tag, Descriptor, Date

This command reads the tag, descriptor, and date contained within the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | None | | |
| Response Data Bytes | 0-5 | Packed | Tag |
| | 6-17 | Packed | Descriptor |
| | 18-20 | Uint8[3] | Date; Respectively: day, month, year – base year (1900) |
| | **Code** | **Class** | **Description** |
| Response Codes | 6 | Error | Device-Specific Command Error |
| | | | |

# Command 14: Read Primary Variable Sensor Information

Reads the Primary Variable Sensor Serial Number, Primary Variable Sensor Limits/Minimum Span Units Code, Primary Variable Upper Sensor Limit, Primary Variable Lower Sensor Limit, and Primary Variable Minimum Span for the sensor. The Primary Variable Sensor Limits and Minimum Span Units will be the same as the Primary Variable Units.

**Note**

The sensor serial number is not applicable to the DLC3010 and is set to “0”.

3 bytes (2^24 = 16,777,216 ) were insufficient to code Fisher serial numbers, which have passed the 17,000,000 mark, and may also contain non-numeric elements. We have created a device-specific variable to hold the displacer serial number

| | Byte | Format | Description | Returned Value |
|---|---|---|---|---|
| Request Data bytes | None | | | |
| Response Data Bytes | 0-2 | Uint24 | Not used in DLC3010 - Sensor Serial Number | 000000 |
| | 3 | Enum | Sensor Limits and Minimum Span Units Code | From Cmd 44 |
| | 4-7 | Float | Upper Sensor Limit | Level Offset plus 120% of displacer length, or a density of 1.5 SGU |
| | 8-11 | Float | Lower Sensor Limit | Level Offset minus 20% of displacer length, or a density of 0.1 SGU |
| | 12-15 | Float | Minimum Span | Closest usable spacing between upper/lower range values before accuracy issues need to be considered, 25% of displacer length or a density difference of 0.25 SGU. |
| | **Code** | **Class** | **Description** | |
| Response Codes | 6 | Error | Device-Specific Command Error | |
| | | | | |

# Command 15: Read Primary Variable Output Information

This command reads the Instrument alarm selection code (condition of the hardware jumper), units code for the PV range variables, upper and lower range values, PV damping value, and private label distributor code. This command has the HART 5 structure, 1 byte shorter than the HART 7 version.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | None | | |
| Response Data Bytes | 0 | Enum | Hardware Alarm Selection Code (Ignore)(1) |
| | 1 | Enum | PV Transfer Function Code(2) |
| | 2 | Enum | PV Upper and Lower Range Value Units Code, refer to table 6. |
| | 3-6 | Float | PV Upper Range Value |
| | 7-10 | Float | PV Lower Range Value |
| | 11-14 | Float | PV Damping Value (units of seconds) |
| | 15 | Enum | Write Protect Code (0=Not Protected, 1=Protected) |
| | 16 | Enum | Private Label Distributor Code, refer to table 3 |
| | **Code** | **Class** | **Description** |
| Response Codes | 6 | Error | Device-Specific Command Error |

1. The enumeration assignments for Alarm Selection Code in the DLC3010 do not comply with the definition in HCF_SPEC-183 Common Tables Specification, Table 6. Therefore, it is not referenced in DD menus, and the associated data item is handled by a device-specific command and variable.
2. The value of Transfer Function Code in the DLC3010 was inadvertently hard-coded to ‘1’ during development. The DD uses a post-read action to reset it to the correct value of ‘0’, but it must be ignored when accessed outside of the DD.

# Command 16: Read Final Assembly Number

This command reads a 24-bit user-defined identification number from the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | None | | |
| Response Data Bytes | 0-2 | Uint24 | Final Assembly Number |
| | **Code** | **Class** | **Description** |
| Response Codes | 6 | Error | Device-Specific Command Error |
| | | | |

# Command 17: Write Message

This command allows you to write a 24-character informational message into the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | 0-23 | Packed | Message string |
| Response Data Bytes | 0-23 | Packed | Message String |
| | **Code** | **Class** | **Description** |
| Response Codes | 5 | Error | Too few data bytes |
| | 6 | Error | Device Specific command error |
| | 7 | Error | In write protect mode |

# Command 18: Write Tag, Descriptor, Date

This command writes the tag, descriptor, and date into the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | 0-5 | Packed | Tag |
| | 6-17 | Packed | Descriptor |
| | 18-20 | Uint8[3] | Date; Respectively: day, month, year – base year (1900) |
| Response Data Bytes | 0-5 | Packed | Tag |
| | 6-17 | Packed | Descriptor |
| | 18-20 | Uint8[3] | Date; Respectively: day, month, year – base year (1900) |
| | **Code** | **Class** | **Description** |
| Response Codes | 5 | Error | Too few data bytes |
| | 6 | Error | Device Specific command error |
| | 7 | Error | In write protect mode |

# Command 19: Write Final Assembly Number

This command writes the user-defined final assembly number into the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | 0-2 | Uint24 | Final Assembly Number |
| Response Data Bytes | 0-2 | Uint24 | Final Assembly Number |
| | **Code** | **Class** | **Description** |
| Response Codes | 5 | Error | Too few data bytes |
| | 6 | Error | Device Specific command error |
| | 7 | Error | In write protect mode |

# Common-Practice Commands

The DLC3010 field device supports the following common practice commands:

Command 34: Write Primary Variable Damping Value
Command 35: Write Primary Variable Range Values
Command 36: Set Primary Variable Upper Range Value
Command 37: Set Primary Variable Lower Range Value
Command 38: Reset Configuration Change Flag
Command 40: Enter/Exit Fixed Current Mode
Command 41: Perform Transmitter Self Test
Command 42: Perform Master Reset
Command 44: Write PV Units Code
Command 45: Trim PV Current DAC Zero
Command 46: Trim PV Current DAC Gain
Command 48: Read Additional Transmitter Status
Command 50: Read Dynamic Variable Assignments
Command 51: Write Dynamic Variable Assignments
Command 53: Set Device Variable Units
Command 59: Write Number of Response Preambles

## Burst Mode

This field device supports Burst Mode and the following commands:

Command 108: Write Burst Mode Command Number
Command 109: Burst Mode Control

## Catch Device Variable

This field device does not support Catch Device Variable.

# Command 34: Write Primary Variable Damping Value

Writes the damping value applied to the PV filter, in seconds. This term sets the time constant of the digital filter.

| | Byte | Format | Description | Allowable choices |
|---|---|---|---|---|
| Request Data Bytes | 0-3 | Float | Damping Value for Primary Variable | 0 to 16 seconds |
| Response Data Bytes | 0-3 | Float | Damping Value for Primary Variable | |
| | **Code** | **Class** | **Description** | |
| Response Codes | 3 | Error | Passed parameter too large | |
| | 4 | Error | Passed parameter too small | |
| | 5 | Error | Too few data bytes received | |
| | 7 | Error | In Write Protect mode | |
| | 8 | Warning | Set to nearest possible value | |

# Command 35: Write Primary Variable Range Values

This command writes the primary variable ranging values with the associated units code. The Upper Range Value may be set lower than the Lower Range Value to implement reverse action for the analog output. Changing action via Command 35 is more robust than attempting to use Commands 36 and 37 iteratively for that purpose.

The unit code supplied in this command is not stored in the instrument. The supplied ranges will be converted to current PV units before being applied. Range values must be validated in the DD to assure that they are within the allowable limits and are separated by a minimum span. The unit code is checked in the instrument and Invalid Selection will be returned for a code that is not supported by the device in the current operating mode.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Uint8 | Upper and Lower Range Values Unit Code |
| | 1 - 4 | Float | Primary Variable Upper Range Value |
| | 5 - 8 | Float | Primary Variable Lower Range Value |
| Response Data Bytes | 0 | Uint8 | Upper and Lower Range Values Unit Code |
| | 1 - 4 | Float | Primary Variable Upper Range Value |
| | 5 - 8 | Float | Primary Variable Lower Range Value |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 9 | Error | Lower Range too high |
| | 10 | Error | Lower Range too low |
| | 11 | Error | Upper Range too high |
| | 12 | Error | Upper Range too low |
| | 14 | Warning | Span is too small (Data is accepted) |

# Command 36: Set Primary Variable Upper Range Value

The value of the process applied to the process variable becomes the primary variable upper range value. The lower range value remains the same.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | None | | |
| | **Code** | **Class** | **Description** |
| Response Codes | 7 | Error | In Write Protect mode |
| | 9 | Error | Applied process too high |
| | 10 | Error | Applied process too low |
| | 14 | Warning | Span is too small (Data is accepted) |

# Command 37: Set Primary Variable Lower Range Value

The value of the process applied to the process variable becomes the primary variable lower range value. To maintain the same span, the device will attempt to shift the upper range value by the same amount that the lower range value changes.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | None | | |
| | **Code** | **Class** | **Description** |
| Response Codes | 7 | Error | In Write Protect mode |
| | 9 | Error | Applied process too high |
| | 10 | Error | Applied process too low |
| | 14 | Warning | New Lower Range Value pushed Upper Range Value over limit. Data is accepted in instrument. The Upper Range Value will be clamped at the sensor limit. |
| Note: Response Codes 9 and 10 will take precedence if the new span between LRV and URV is less than the minimum span given by command 15. | | | |

# Command 38: Reset Configuration Change Flag

Resets the configuration changed flag associated with the requesting master[1] in the device status byte. Masters should issue this command after the configuration change has been detected and all needed configuration data has been read from the device.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | None | | |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | Note: The HART specification provides response code 7 (In Write protect mode), but blocking the reset action just because the device is write-protected would defeat the general principle behind the mechanism implemented by this command |
| | | | |
| | | | |
| | | | |

1 The standard now requires that 2 copies of this flag be maintained, one for each type of master. Both copies must be set for any change that affect configuration, and each type of master may only reset its own copy. It must be the responsibility of the host to decide when the flag copy can be cleared, and to make sure it does get cleared.

# Command 40: Enter/Exit Fixed PV Current Mode

The device is placed in the Fixed Primary Variable Current Mode with the PV current set to the value received. A commanded level of ‘0’ forces exit from the Fixed PV Current Mode. This mode is also cleared when power is removed from the device. Value sent is in mA. Range checking is performed in the instrument (3.7 mA to 22.5 mA). The current command will be clamped to these limits if the request violates them.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | 0-3 | Float | Current Value (in mA) requested for Fixed Current Mode |
| Response Data Bytes | 0-3 | Float | Applied PV Current command (in mA) |
| | **Code** | **Class** | **Description** |
| Response Codes | 14 | Error | In Multi-Drop Mode |

# Command 41: Perform Transmitter Self Test

This command is reserved for factory testing only.

# Command 42: Perform Master Reset

Respond immediately and then reset the microprocessor. This is equivalent to a power cycle in the DLC3010.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | None | | |
| Response Data Bytes | None | | |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |

# Command 44: Write Primary Variable Units

This command is used to change the units of the Primary Variable. Range checking is performed in the instrument.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Uint8 | Primary Variable Units Code |
| Response Data Bytes | 0 | Uint8 | Primary Variable Units Code |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |

# Command 45: Trim Primary Variable Current DAC Zero

Trim the zero of the analog output so that the commanded signal current is accurately produced. This trim is typically performed using the Fixed Current Mode by adjusting the associated DAC output to 4.0 mA as observed on a precision meter. Reasonableness range checking must be done in the DD.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0-3 | Float | Externally measured current level (mA) |
| Response Data Bytes | 0-3 | Float | Instrument current level (mA) |
| | **Code** | **Class** | **Description** |
| Response Codes | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 9 | Error | Not in proper current mode |
| | 11 | Error | In Multidrop mode |

# Command 46: Trim Primary Variable Current DAC Gain

Trim the gain of the analog output so that the commanded signal current is accurately produced. This trim is typically performed using the Fixed Current Mode by adjusting the associated DAC output to 20.0 mA as observed on a precision meter. Reasonableness range checking must be done in the DD.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0-3 | Float | Externally measured current level (mA) |
| Response Data Bytes | 0-3 | Float | Instrument current level (mA) |
| | **Code** | **Class** | **Description** |
| Response Codes | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 9 | Error | Not in proper current mode |
| | 11 | Error | In Multidrop mode |

# Command 48: Read Additional Status

Returns status information which is not included in the response code bytes. The free time alarm bit will be cleared by this read action. Command 48 returns 3 bytes of data (see table 2b).

| | Byte | Format | Description |
|---|---|---|---|
| Request Data bytes | None | | |
| Response Data Bytes | 0 | Uint8 | Command 48 Response Byte 0 |
| | 1 | Uint8 | Command 48 Response Byte 1 |
| | 2 | Uint8 | Command 48 Response Byte 2 |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |
| | | | |

# Command 50: Read Dynamic Variable Assignments

This command command reads the variable assigned to the primary variable, namely: Level, Interface, or Density, plus the fixed assignments. Number of variables matches the HART spec, even though only three variables are read from the instrument.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | 0 | Enum | ID of variable returned as the first variable (PV) in Command 3, see table 5 |
| | 1 | Enum | ID of variable returned as the second variable (SV) in Command 3, see table 5<br>Fixed at Electronics Temperature |
| | 2 | Enum | ID of variable returned as the third variable (TV) in Command 3, see table 5<br>Fixed at Process Temperature |
| | 3 | Enum | Code for Unused (250 decimal) |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |

# Command 51: Write Dynamic Variable Assignments

This command assigns the variable to be used as the primary variable: Level, Interface, or Density. It writes only a single variable, not the 4 variables defined in the HART specification. The instrument performs validity checking. Parameters 2, 3, and 4 are not writable and will return what the instrument is using. Note: The instrument may need to be reconfigured to get valid PV units and upper and lower range values after changing the primary variable. Therefore, this command should only be issued from a method that takes care of such clean up.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device variable ID assigned to PV, see table 5 |
| | 1 | Enum | Unused and Ignored by instrument |
| | 2 | Enum | Unused and Ignored by instrument |
| | 3 | Enum | Unused and Ignored by instrument |
| Response Data Bytes | 0 | Enum | Device variable ID assigned to PV |
| | 1 | Enum | Device variable ID assigned to SV<br>Electronics Temperature code from table 5. |
| | 2 | Enum | Device variable ID assigned to TV<br>Process Temperature code from table 5. |
| | 3 | Enum | Code for Unused (250 decimal) |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 7 | Error | In Write Protect Mode |

# Command 53: Set Device Variable Units

This command writes the units associated with the specified parameter. Range checking is performed in the instrument. Note: Process temperature and electronics temperature share the same unit code so writing new units to one will change the units for the other.

| | Byte | Format | Description | Allowable choices |
|---|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device Variable ID | See table 5 |
| | 1 | Enum | Units Code | See table 6 |
| Response Data Bytes | 0 | Enum | Device Variable ID | |
| | 1 | Enum | Units Code | |
| | **Code** | **Class** | **Description** | |
| Response Codes | 5 | Error | Too few data bytes received | |
| | 7 | Error | In Write Protect mode | |
| | 11 | Error | Invalid Variable ID | |
| | 12 | Error | Invalid units code | |

# Command 59: Write Number of Response Preambles

This This is a Data Link Layer Management Command. This command selects the number of preambles to be sent by a device before the start of a response packet. The instrument range checks this value from 5-20. It stores and reads this value from NVM, but will use a RAM copy with a minimum value of 5 no matter what it reads from NVM.

| | Byte | Format | Description | Allowable choices |
|---|---|---|---|---|
| Request Data Bytes | 0 | Uint8 | Number of preambles to be sent prior to the response message from the DLC3010 slave to the Master | 5-20 |
| Response Data Bytes | 0 | Uint8 | Number of preambles to be sent prior to the response message from the DLC3010 slave to the Master | |
| | **Code** | **Class** | **Description** | |
| Response Codes | 3 | Error | Passed parameter is too large | |
| | 4 | Error | Passed parameter is too small | |
| | 5 | Error | Too few data bytes received | |
| | 7 | Error | In Write Protect mode | |

# Command 108: Write Burst Command Number

This command is used to select the response command the device transmits while in Burst Mode. HART 5 did not provide a Read Command for Burst Command Number, so a Device-Specific Command is used to access the value for display.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Burst Command Number (1, 2, or 3) |
| Response Data Bytes | 0 | Enum | Burst Command Number (1, 2, or 3) |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | | | |

# Command 109: Write Burst Mode Control

This command is used to enter and exit Burst Mode. HART 5 did not provide a Read Command for Burst Mode, so a Device-Specific Command is used to access the value for display.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Burst Mode (0 = Off, 1= On) |
| Response Data Bytes | 0 | Enum | Burst Mode |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | | | |

# Device-Specific Commands

The DLC3010 field device supports the following Device-Specific HART Commands.

Command 128: Read Float Variables with Selectable Units
Command 129: Write Float Variables with Selectable Units
Command 130: Read Float Variables with Fixed Units
Command 131: Write Float Variables with Fixed Units
Command 132: Read Byte Variables
Command 133: Write Byte Variables
Command 134: Read NVM (private)
Command 135: Write NVM (private)
Command 137: Read Word Variables
Command 138: Write Word Variables
Command 139: Read Displacer Serial Number
Command 140: Write Displacer Serial Number
Command 142: Read Table Entry
Command 143: Write Table Entry
Command 144: Read Trend Buffer
Command 145: Read Device Variable Alert Params
Command 146: Write Hi Alert
Command 147: Write Hi Hi Alert
Command 148: Write Lo Alert
Command 149: Write Lo Lo Alert
Command 150: Read NVM Count Remaining
Command 151: Capture Zero Reference Angle
Command 152: Set Write Lock

# Command 128: Read Float Variables with Selectable Units

This command is used to read a floating-point variable and its associated units code.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 7 |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 7 |
| | 1 | Enum | Units Code. See table 6 |
| | 2-5 | Float | Value of Selected Variable in Associated Units |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | 5 | Error | Too few data bytes received |

# Command 129: Write Float Variables with Selectable Units

This command is used to write the value of a floating-point variable in its associated units code. The instrument will convert the value from the presented units to standard internal units before storing it in NVM. The unit code is not stored. Range checking must be performed in the DD.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 7 |
| | 1 | Enum | Units Code, see table 6 |
| | 2-5 | Float | Value of Selected Variable in Associated Units |
| Response Data Bytes | 0 | Enum | Variable Index Number |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Value of Selected Variable in Associated Units |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection (Units Code) |
| | 3 | Error | Passed parameter is too large |
| | 4 | Error | Passed parameter is too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 15 | Error | Invalid Index (Variable) |

# Command 130: Read Float Variables with Fixed Units

This command is used to read a floating-point variable without any associated units code. The DD must check for invalid selections.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 8 |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 8 |
| | 1-4 | Float | Value of Selected Variable. |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | 5 | Error | Too few data bytes received |

# Command 131: Write Float Variables with Fixed Units

This command is used to write the value of a floating-point variable without any associated units code. Range checking must be performed in the DD.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 8 |
| | 1-4 | Float | Value of Selected Variable |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 8 |
| | 1-4 | Float | Value of Selected Variable |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | 3 | Error | Passed parameter is too large |
| | 4 | Error | Passed parameter is too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |

# Command 132: Read Byte Variables

This command is used to read a byte variable.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 9 |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 9 |
| | 1 | Enum | Value of Selected Byte Variable |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | 5 | Error | Too few data bytes received |

# Command 133: Write Byte Variables

This command is used to write the value of a byte variable. The instrument only provides range checking for the Hall drive-current setting and the Number of Meter Decimal Places. All other range checking must be performed in the DD.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 9 |
| | 1 | Enum | Value of Selected Variable |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 9 |
| | 1 | Enum | Value of Selected Variable |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 15 | Error | Invalid Index |

# Command 134: Read NVM

Command 134 description is included in the proprietary version of this document.
Contact your Emerson sales office if additional information is required.

# Command 135: Write NVM

Command 135 description is included in the proprietary version of this document.
Contact your Emerson sales office if additional information is required.

# Command 137: Read Word Variables

This command is used to read Word-size parameters. Range checking must be done outside the instrument. For invalid index numbers, the instrument will return 0 for data. (This command was not utilized in the user interface.)

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 10 |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 10 |
| | 1 | Uint8 | LS Byte of Word |
| | 2 | Uint8 | MS Byte of Word |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | | | |

# Command 138: Write Word Variables

This command is used to write Word-size parameters. Range checking must be done by the DD. For valid writes, the configuration changed bit will be set. For invalid writes, the bit will not be set. (This command was not utilized in the user interface.)

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Variable Index Number from table 10 |
| | 1 | Uint8 | LS Byte of Word |
| | 2 | Uint8 | MS Byte of Word |
| Response Data Bytes | 0 | Enum | Variable Index Number from table 10 |
| | 1 | Uint8 | LS Byte of Word |
| | 2 | Uint8 | MS Byte of Word |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid Selection |
| | 9 | Error | Passed value too high |
| | 10 | Error | Passed value too low |

# Command 139: Read Serial Number

This command is used to read the instrument or displacer serial number. If the passed index is invalid, the device sets it to 1.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Serial Number Index: (0 = Displacer, 1 = Instrument) |
| Response Data Bytes | 0 | Enum | Serial Number Index |
| | 1-16 | ASCII | Serial Number string |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |
| | | | |

# Command 140: Write Serial Number

This command is used to write the instrument or displacer serial number.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Serial Number Index: (0 = Displacer, 1 = Instrument) |
| | 1-16 | ASCII | Serial Number string |
| Response Data Bytes | 0 | Enum | Serial Number Index |
| | 1-16 | ASCII | Serial Number string |
| | **Code** | **Class** | **Description** |
| Response Codes | 5 | Error | Too few data bytes received |
| | 15 | Error | Invalid index |

# Command 142: Read Table Entry

This command is used to read an entry from one of the tables in the instrument. The arguments allow selection of the table, and the row of the table to be read. The instrument returns the pair of data associated with the table entry. Range checking for proper table index and row number must be done in the DD.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Table Index from table 12 |
| | 1 | Uint8 | Row Number (0-9) |
| Response Data Bytes | 0 | Enum | Table Index |
| | 1 | Uint8 | Row Number |
| | 2-5 | Float | Temperature Value |
| | 6-9 | Float | Coefficient Value |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 5 | Error | Too few data bytes received |

# Command 143: Write Table Entry

This command is used to write an entry into a compensation table in the instrument. Arguments allow selection of the table, and the row of the table. If the tables are not full, a column 2 (coefficient value) entry of 0.0 will signal the end of valid data for the table. The DD interface must write the last entry of 0.0 for column 2 to signal the device that the table is complete and can be used for computations, unless the table has valid data in all 10 rows.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Table Index from table 12 |
| | 1 | Uint8 | Row Number (0-9) |
| | 2-5 | Float | Temperature Value (Column 1 entry) |
| | 6-9 | Float | Coefficient Value (Column 2 entry) |
| Response Data Bytes | 0 | Enum | Table Index |
| | 1 | Uint8 | Row Number |
| | 2-5 | Float | Temperature Value |
| | 6-9 | Float | Coefficient Value |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 15 | Error | Invalid row number |

# Command 144: Read Trend Buffer

This command is used to read trend data from the instrument. See table 6 for trend variable definitions. The trend buffer is zeroed after each read. This results in 'zero' values being displayed for slots that have not had time to be filled with samples since the last read of the buffer. The trend buffer is continuously updated at the trend interval whether it is read or not. (This feature has been removed from the DD in recent revisions, since the available range for trend interval is quite small.)

| | **Byte** | **Format** | **Description** |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | 0 | Enum | Gap Flag: 0 = No gap in data since last read<br>10 = Illegal trend variable specified<br>11 = Gap in data since last read |
| | 1-4 | Float | First Trend Sample |
| | 5-8 | Float | Second Trend Sample |
| | 9-12 | Float | Third Trend Sample |
| | 13-16 | Float | Fourth Trend Sample |
| | 17-20 | Float | Fifth Trend Sample |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |
| | | | |

# Command 145: Read Device Alert Parameters

This command is used to read the user-configurable alarm thresholds stored in the instrument. It returns all 4 alarm thresholds for the primary variable, and only the Hi and Lo thresholds for temperature alarms. The Hi-Hi and Lo-Lo thresholds for temperature will return Not-a-Number. Only a single set of alarm threshold variables exist in firmware for the Primary Variable, (i.e., there are not separate values maintained for Interface and Density when Level is the active mode). The values must be edited when the Primary Variable code is changed between (Liquid Level or Interface Level) and Liquid Density, as the measurement has different dimensions.

| | **Byte** | **Format** | **Description** |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device Variable Code, see table 5 |
| Response Data Bytes | 0 | Enum | Device Variable Code |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Hi Alarm Threshold |
| | 6-9 | Float | Hi-Hi Alarm Threshold |
| | 10-13 | Float | Lo Alarm Threshold |
| | 14-17 | Float | Lo-Lo Alarm Threshold |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 5 | Error | Too few databytes received |

# Command 146: Write Hi Alert

This command is used to write the Hi alert threshold for the specified Device Variable. Range checking must be done in the DD for both range and valid Device Variable code. The Device Variable should be either one of the two temperatures or the current PV variable code.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device Variable Code, see table 5 |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Hi Alert Threshold |
| Response Data Bytes | 0 | Enum | Device Variable Code |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Hi Alert Threshold |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 15 | Error | Invalid index |

# Command 147: Write Hi Hi Alert

This command is used to write the Hi Hi alert threshold for the Primary Variable. Range checking must be done in the DD for both valid PV code and range. Using a variable code that is not the current PV variable code will result in corrupting the alarm threshold values. The Device Variable index used in the request should be the current PV variable code.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device Variable Code, see table 5 |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Hi Hi Alert Threshold |
| Response Data Bytes | 0 | Enum | Device Variable Code |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Hi Hi Alert Threshold |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 15 | Error | Invalid index |

# Command 148: Write Lo Alert

This command is used to write the Lo alert threshold for the specified Device Variable. Range checking must be done in the DD for both range and valid Device Variable code. The Device Variable should be either one of the two temperatures or the current PV variable code.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device Variable Code, see table 5 |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Lo Alert Threshold |
| Response Data Bytes | 0 | Enum | Device Variable Code |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Lo Alert Threshold |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 15 | Error | Invalid index |

# Command 149: Write Lo Lo Alert

This command is used to write the Lo Lo alert threshold for the Primary Variable. Range checking must be done in the DD for both valid PV code and range. Using a variable code that is not the current PV variable code will result in corrupting the alarm threshold values. The Device Variable index used in the request should be the current PV variable code.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Device Variable Code, see table 5 |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Lo Lo Alert Threshold |
| Response Data Bytes | 0 | Enum | Device Variable Code |
| | 1 | Enum | Units Code |
| | 2-5 | Float | Lo Lo Alert Threshold |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 3 | Error | Passed parameter too large |
| | 4 | Error | Passed parameter too small |
| | 5 | Error | Too few data bytes received |
| | 7 | Error | In Write Protect mode |
| | 15 | Error | Invalid index |

# Command 150: Read NVM Remaining

This command returns the number of NVM write counts remaining in each of the three NVM sections. Their type is "unsigned integer".

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | 0-1 | Uint16 | NVM Writes Remaining in Microprocessor EEPROM |
| | 2-3 | Uint16 | NVM Writes Remaining in On-Board NVM |
| | 4-5 | Uint16 | NVM Writes Remaining in Transducer Board |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |
| | | | |

# Command 151: Capture Zero Reference Angle

This command is used to mark the current pilot shaft position as the zero (differential) buoyancy reference. The instrument will read the internal variable representing shaft rotation, and store it for use in all measurement calculations.

In Level measurement mode, where the SG value represents the differential density between the upper and lower phase, this command should be issued after the lever assembly has been unlocked, and the displacer is at the lowest buoyancy *process* condition.

In Interface or Density measurement mode, the variable being marked must represent true zero buoyancy. Because of this, it is not possible to use this command effectively with an over-weight displacer in those modes. In this condition, the sensor linkage is on a travel stop at zero buoyancy and does not reflect the full theoretical deflection. The required zero-reference value must be back-computed from two calibration data points, and written explicitly with the Write NVM command.

This command does not check write lock status, although it affects the instrument calibration. It therefore should be used only inside a method in the instrument DD, where write protect status can be enforced explicitly.

| | Byte | Format | Description |
|---|---|---|---|
| Request Data Bytes | None | | |
| Response Data Bytes | None | | |
| | **Code** | **Class** | **Description** |
| Response Codes | None | | |
| | | | |

# Command 152: Set Write Lock

This command is used to either set or clear the write lock. This command is also sensitive to Primary/Secondary Master status in the message. If a primary master sets write lock, then only a primary master can clear write lock, not a secondary, and vice versa. However, this variable is stored in RAM and will be reset to the OFF condition (writes enabled) after power-up, providing a work-around if the correct master priority is unavailable.

Write lock status is checked by the following:

Universal commands: 6, 17, 18, 19.

Common Practice commands: 34, 35, 36, 37, 40, 44, 45, 46, 51, 53, 59, 108, 109.

Device-Specific commands: 129, 131, 133, 138, 140, 143, 146, 147, 148, 149, (and 152 with respect to a different master hierarchy).

| | **Byte** | **Format** | **Description** |
|---|---|---|---|
| Request Data Bytes | 0 | Enum | Write Lock request: (0 = OFF, 1 = ON) |
| Response Data Bytes | 0 | Enum | Write Lock state |
| | **Code** | **Class** | **Description** |
| Response Codes | 2 | Error | Invalid selection |
| | 16 | Error | Access Restricted (Another host has possession of the lock) |

# Tables

### Table 1: DLC3010 HART Response Codes

The following codes represent a sub-set of the HART Response Codes which can be returned by the DLC3010 instrument.

| Code | Meaning |
|---|---|
| 0 | No command specific errors |
| 2 | Invalid Selection |
| 3 | Parameter Too Large |
| 4 | Parameter Too Small |
| 5 | Incorrect Byte Count |
| 7 | In Write Protect Mode |
| 8 | Data Truncated (warning) |
| 9 | Lower Range Too High |
| | Not in Proper Mode |
| | Process Too High |
| | Zero Range Too High |
| 10 | Lower Range Too Low |
| | Process Too Low |
| | Zero Range Too Low |
| 11 | In MultiDrop Mode |
| | Invalid Variable Code |
| | Span Range Too High |
| | Starting Address Too Low |
| | Upper Range Too High |
| 12 | Starting Address Too High |
| | Upper Range Too Low |
| 13 | Span Lower Out of Limits |
| | Upper/Lower Out of Limits |
| 14 | New LRV Pushed URV Over Limit (Warning) |
| | Span Too Small (Warning) |
| 15 | Invalid Index |
| 16 | Access Restricted |
| 32 | Busy |
| 64 | Command Not Implemented |

Table 2a: Device Status

| Bit | Name of Status Bit | Meaning |
|---|---|---|
| 7 | Field Device Malfunction | - *This bit should be set whenever the firmware detects a serious error or failure that compromises device operation.* In DLC3010 firmware 8, the only event that sets it is a time-out during writes to Description or Message, caused by a firmware bug involving an improperly initialized timer. Its functional severity is thus downgraded to advisory. |
| 6 | Configuration Changed | Two such bits exist internally, one for each HART master. Both copies are set when any variable, HART message, tag, descriptor or date are changed from HART. Cleared by command 38, separately for each master. This bit survives loss of power. A firmware 8 bug prevents it from being returned to the secondary master. |
| 5 | Cold Start | Set by the firmware whenever a RESET sequence is executed or at initial device power up. Cleared by the first HART command. Two such bits exist internally, one for each HART master. |
| 4 | More Status | Active when any bit in command 48 is active. |
| 3 | Primary Variable Analog Output Fixed | The analog output for the Primary Variable is being held at a fixed value and is not responding to process variations. |
| 2 | Primary Variable Analog Output Saturated | The analog output for the Primary Variable has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further. |
| 1 | Non-Primary Variable Out of Limits | A Device Variable not mapped to the Primary Variable is beyond the operating limits of the device. The “Read Additional Transmitter Status” Command, #48, may be required to identify the variable. This bit is the logical OR of the Middle Status Byte, Bits 3 through 0, (Process Temperature and Electronics Temperature Alarms). |
| 0 | Primary Variable Out of Limits | Primary Variable is beyond the operating limits of the device. |

Table 2b: Additional Status Bytes

| Byte | Bit | Name of Status Bit | Meaning |
|---|---|---|---|
| 0 | 7 | Reserved | |
| | 6 | Reserved | |
| | 5 | Sensor Signal Failed | The torque tube position reading has exceeded the hard-coded limits, either above or below. |
| | 4 | Sensor Drive Failed | The Hall sensor drive current read-back has exceeded the hard-coded limit. |
| | 3 | Reference Voltage Failed | The reference voltage for the A/D has exceeded the hard-coded limits, either above or below. |
| | 2 | NVM Write Limit Exceeded | The total number of writes to NVM has exceeded the hard-coded limit. This bit is an OR of all the NVM write limit flags. |
| | 1 | Free Time Limit Exceeded | There is insufficient free time remaining in the execution period to complete the scheduled tasks. |
| | 0 | Process Temperature Sensor Failed | The apparent resistance measured at the RTD terminals is less than 10 ohms or greater than 320 ohms. |
| 1 | 7 | PV Hi Limit Exceeded | Active when user-configured threshold is violated. |
| | 6 | PV HiHi Limit Exceeded | Active when user-configured threshold is violated. |
| | 5 | PV Lo Limit Exceeded | Active when user-configured threshold is violated. |
| | 4 | PV LoLo Limit Exceeded | Active when user-configured threshold is violated. |
| | 3 | Process Temperature Hi Limit Exceeded | Active when user-configured threshold is violated. |
| | 2 | Process Temperature Lo Limit Exceeded | Active when user-configured threshold is violated. |
| | 1 | Electronics Temperature Hi Limit Exceeded | Active when user-configured threshold is violated. |
| | 0 | Electronics Temperature Lo Limit Exceeded | Active when user-configured threshold is violated. |
| 2 | All | Reserved | |
| "Reserved" bits are always set to 0 | | | |

Table 3: Manufacturer Identification Code

| Code | Meaning |
|---|---|
| 19 | Fisher Controls |

Table 4: Physical Signaling Codes

| Code | Meaning |
|---|---|
| 0 | Bell 202 Current Mode |
| 1 | Bell 202 Voltage Mode |

Table 5: DLC3010 Device Variable Codes
(Index for commands 50, 51, 145, 146, 147, 148, 149)

| Code | Variable | Class | Unit Codes |
|---|---|---|---|
| 0 | Liquid Level | None | Length (table 6) |
| 1 | Interface Level | None | Length (table 6) |
| 2 | Liquid Density | None | Density (table 6) |
| 3 | Process Temperature | None | Temperature (table 6) |
| 4 | Electronics Temperature | None | Temperature (table 6) |

Table 6: DLC3010 Enumeration Codes

Length Units

| Code | Units |
|---|---|
| 44 | feet |
| 45 | meters |
| 47 | inches |
| 48 | centimeters |
| 49 | millimeters |

Density Units

| Code | Units |
|---|---|
| 90 | SGU (specific gravity units) |
| 91 | grams/cubic centimeter |
| 92 | kilograms/cubic meter |
| 93 | pounds/gallon |
| 94 | pounds/cubic feet |
| 95 | grams/milliliter |
| 96 | kilograms/liter |
| 97 | grams/liter |
| 98 | pounds/cubic inch |

Temperature Units

| Code | Units |
|---|---|
| 32 | Centigrade |
| 33 | Fahrenheit |

Weight Units

| Code | Units |
|---|---|
| 60 | gram |
| 61 | kilogram |
| 63 | pound |
| 125 | ounce |

Volume Units

| Code | Units |
|---|---|
| 41 | liter |
| 113 | cubic inches |
| 240 | cubic millimeters |
| 241 | milliliters (cubic centimeters) |

Torque Rate Units

| Code | Units |
|---|---|
| 247 | inch pound/degree |
| 248 | Newton meter/degree |
| 249 | dyne centimeter/degree |

Sensor Mounting

| Code | Meaning |
|---|---|
| 0 | Left of displacer |
| 1 | Right of displacer |

LCD Meter Installation

| Code | Meaning |
|---|---|
| 0 | Not installed |
| 1 | Installed |

LCD Meter Display Mode

| Code | Meaning |
|---|---|
| 0 | All cells off |
| 1 | All cells on |
| 2 | PV only |
| 3 | Alternate:<br>PV, Proc Temp,<br>Degrees Rotation |
| 4 | Percent only |
| 5 | Alternate:<br>PV, Percent |

Table 6: DLC3010 Enumeration Codes (continued)

LCD Meter Decimal Point

| Code | Meaning |
|---|---|
| 0 | Autoranging |
| 1 | 1 digit after decimal point |
| 2 | 2 digits after decimal point |
| 3 | 3 digits after decimal point |
| 4 | 4 digits after decimal point |

Trend Variable

| Code | Meaning |
|---|---|
| 0 | Trending disabled |
| 1 | Primary Variable |
| 2 | Process temperature |
| 3 | Electronics Temperature |

RTD Type

| Code | Meaning |
|---|---|
| 0 | No RTD installed (manual) |
| 1 | 2-wire RTD |
| 2 | 3-wire RTD |

Hall Sensor Drive Level

| Code | Meaning |
|---|---|
| 0 | Low level (≅500 μA) |
| 1 | High level (≅725 μA) |

PV Alert Enable Bits
(1=enabled, 0 = disabled)

| Bit Mask | Monitor |
|---|---|
| 0x80 | Reserved |
| 0x40 | Reserved |
| 0x20 | Reserved |
| 0x10 | Reserved |
| 0x08 | PV Hi |
| 0x04 | PV HiHi |
| 0x02 | PV Lo |
| 0x01 | PV LoLo |

Temperature Alert Enable Bits
(1=enabled, 0 = disabled)

| Bit Mask | Monitor |
|---|---|
| 0x80 | Reserved |
| 0x40 | Reserved |
| 0x20 | Reserved |
| 0x10 | Reserved |
| 0x08 | Process Temperature Hi |
| 0x04 | Process Temperature Lo |
| 0x02 | Electronics Temperature Hi |
| 0x01 | Electronics Temperature Lo |

Table 7: DLC3010 Floating Point Variables with Selectable Units
(Index for commands 128 and 129)

| Code | Parameter |
|---|---|
| 0 | Displacer Weight |
| 1 | Displacer Volume |
| 2 | Displacer Length |
| 3 | Moment Arm Length |
| 4 | Level Offset |
| 5(1) | Process Temperature |
| 6 | Process Temperature Offset |
| 7 | Electronics Temperature Offset |
| 8 | Torque Tube Rate |

1. Process Temperature can only be written when the RTD Type Parameter is set to 0.

Table 8: DLC3010 Floating Point Variables with Fixed Units
(Index for commands 130 and 131)

| Code | Parameter | Fixed Units |
|---|---|---|
| 0[1] | Reference Coupling Point | degrees |
| 1 | Trend Interval | seconds |
| 2[1,2,4] | Free Time *Limit* | % |
| 3[1,4] | Free Time *Remaining* | % |
| 4 | Sensor Drive Limit | µA |
| 5[1,2] | Sensor Drive Value | µA |
| 6 | A/D Reference Limit | counts |
| 7[1,2] | A/D Reference Value | counts |
| 8 | Wire Resistance | Ω |
| 9 | Input Filter Time Constant | seconds |
| 10[1,3] | A/D Torque Tube Input | mV |
| 11 | Reserved | |
| 12 | Reserved | |
| 13 | Sensor Temp Scale (not used) | |
| 14 | Sensor Temp Offset (not used) | |
| 15 | Sensor Amp Scale | dimensionless |
| 16 | Sensor Amp Offset | counts |
| 17 | Sensor Degree Scale | degrees/count |
| 18 | Sensor Degree Offset | counts |
| 19[1,3] | Degrees Rotation | degrees |
| 20 | Magnet Coefficient | dimensionless |
| 21 | Process Temp Scale | dimensionless |

1. Items indicated have Read-Only access through this array.
2. Items indicated are dynamic instrument electronics signals monitored by firmware.
3. Items indicated are precursors to Device Variables that respond to process conditions.
4. These two items have their labels reversed in firmware, as compared to their actual functions.

Table 9: DLC3010 Byte Variables
(Index for commands 132 and 133)

| Code | Parameter |
|---|---|
| 0 | Torque Tube Material |
| 1 | Weight Units |
| 2 | Volume Units |
| 3 | Length Units |
| 4 | Sensor Mounting |
| 5 | LCD Meter Installed |
| 6 | LCD Meter Display Mode |
| 7(1) | LCD Meter Decimal Point |
| 8 | PV Alert Enabled Bit Field |
| 9 | Temperature Alert Enabled Bit Field |
| 10 | Trend Variable Assignment |
| 11 | RTD Type |
| 12 | Torque Rate Units |
| 13 | System A/D Gain Setting |
| 14 | Hall Sensor Drive Signal Level |
| 1. This parameter can only be written if the value of LCD Meter Installed is Yes. | |

Table 10: DLC3010 Word Variables
(index for commands 137 and 138)

| Code | Parameter |
|---|---|
| 0 | DAC Zero |
| 1 | DAC Full Scale |
| 2* | Raw Torque Tube A/D Reading |
| 3* | Raw Process Temperature A/D Reading |
| 4* | Raw Electronics Temperature A/D Reading |
| 5* | Adjusted Torque Tube A/D Reading |
| * Items indicated with an asterisk are Read-Only, as they are dynamic precursors to Device Variables.<br>Items 2-4 are readings from register locations | |

Table 11: DLC3010 NVM Storage Locations
(index for commands 134 and 135)

| Code | Memory Type |
|---|---|
| 1 | Main Electronics Board NVM<br>(buffer for FLASH loads) |
| 2 | Transducer Board NVM<br>(transducer characterization data) |
| 3 | Microprocessor EEPROM<br>(configuration data) |
| 4 | Microprocessor Boot-Sector Flash<br>(boot code and working copy of transducer characterization data) |

Table 12: DLC3010 Coefficient Tables
(index for commands 142 and 143)

| Code | Table Name |
|---|---|
| 0 | Temperature Coefficient for Specific Gravity of Upper Fluid |
| 1 | Temperature Coefficient for Specific Gravity of Lower Fluid |
| 2 | Temperature Coefficient for Torque Rate |

Table 13: DLC3010 Microprocessor RAM and EEPROM Addresses

DLC3010 Microprocessor RAM and EEPROM addresses are defined in the proprietary version of this document. Contact your Emerson sales office if additional information is required.

# Performance

Refer to the DLC3010 Digital Level Controller Instruction Manual (D102748X012) and Product Bulletin 62.1: DLC3010 for details on DLC3010 performance.

## Sampling Rates

The DLC3010 Primary Variable (Level | Interface | Density) is normally updated every 100 milliseconds (10 samples per second).

A Secondary or Tertiary variable sample occurs every 500 milliseconds, but the two are alternated. The sample rate for Temperature variables is therefore 1 per second.

When the output current is below about 4.3 mA, the primary variable sample is skipped when one of the Temperature variables is being sampled that cycle. Therefore, at very low current output, the primary variable sample structure is: 4 samples with 100 millisecond separation, followed by a gap of 200 milliseconds.

The output alarm logic is executed every 200 milliseconds.

The PV alarm logic is executed every 500 milliseconds.

The Temperature alarm logic is executed once a second.

The LCD display is updated every second when displaying one variable, and every two seconds when it is displaying alternating variables.

The analog input filtering has a double pole at about 320 Hz.

The Analog to Digital conversion uses a Sigma/Delta device that runs at a bit rate of 28.8 KHz. The first notch of the decimation filter is at 100 Hz for a theoretical update every 10 milliseconds, but the firmware initiates the conversion only once per 100-millisecond interrupt-driven cycle. The settling time for the Sigma-Delta A/D converter in this mode is 30 milliseconds.

Maximum Delay Time estimate for the primary variable: ~ 250 milliseconds.

# Power-Up

There is a short inrush current transient expected when power is connected to the device.  PV processing begins within 5 seconds. It takes about 1 second for the analog output to settle to 98% of it's commanded value once PV processing has begun. This does not include the input dynamics.

The device does not respond to HART commands until the start-up processing is complete.

# Reset

The effect of Command 42 on the the device firmware processing is the same as a power cycle. A reset removes write protection.

# Self-Test

No user-discernible effect on processing results from issuing Command 41.

Background self-test functions include:

- Serial EEPROM check sum validation
- Write-count-remaining tracking for each NVM region
- Free time tracking for microprocessor cycle
- Primary Sensor drive current monitoring
- Primary sensor A/D self-calibration
- Primary Sensor input measurement conversion range monitoring
- Primary Value output range monitoring (when enabled)
- RTD resistance measurement conversion range monitoring (when installed)
- Electronics and/or Process temperature range monitoring (when enabled)

# Command Response Delay

The device responds to a master request in less than 256 milliseconds (STO @ 1200 bps), as it passed the predecessor of data link layer test DLL024.

# Busy and Delayed-Response

In rare instances the device may respond with the code ‘Busy’ if a write command is still being processed while receiving a new command. Delayed Response mechanisms are not supported by this device.

# Long Messages

Device-Specific Commands for NVM access will permit up to 24 byte payloads in addition to the response code / device status bytes. Some legacy HART multiplexers have been known to truncate command packets that exceed 24 bytes.

# Non-Volatile Memory

There are three sections of non-volatile memory in the device. They utilize serial EEPROM technology. One is embedded in the microprocessor and stores the configuration and calibration constants for the process algorithm. Another is located on the transducer board and contains characterization data for the Hall Sensor and magnet. The final section is a buffer on the main board used only for firmware loading.

# Modes

The device does not support any alternative operating modes other than “Fixed Current” mode, which is used during alarm condition, for multi-drop operation, or for maintenance on the analog output circuit. However, since selection of the PV via command 51 restructures the process measurement algorithm, the device can be said to have 3 different “Measurement” Modes: Liquid Level, Interface Level, or Density.

# Write Protection

There are two Write Protection states for the DLC3010: Not Write Protected or Write Protected. ‘Protected’ prevents configuration and calibration changes to the instrument. The default setting is ‘Not Protected'.

Protection is under software control. The method used for rudimentary write protection is a volatile variable written by command 152. Write Protection states can be applied remotely, but to remove protection, the same priority master must be used. If the primary master sets write lock, then only the primary master can clear write lock, not the secondary. A power cycle or device reset will also remove write protection. Write lock status is checked by the following:

| | |
|---|---|
| Universal commands: | 6, 17, 18, 19 |
| Common Practice commands: | 34, 35, 36, 37, 40, 44, 45, 46, 51, 53, 59, 108, 109 |
| Device-Specific commands: | 129, 131, 133, 138, 140, 143, 146, 147, 148, 149, (and 152 with respect to a different master hierarchy) |

# Damping

There are two digital filters in the signal path that can be modified by the user. Time Constant can be set between 0 and 16 seconds for each. The PV Damping is applied to the entire PV signal after compensation. The Sensor Damping (Input Filter Time Constant, see code 9 in table 8) is applied only to the Torque Tube signal. It is advisable to use only one of these filters at a time. Selection should be based on the noise level in each signal path.

The breakpoint of the analog output filter following the D/A converter is at about 25 Hz.

# Annex A: Compatibility Checklist

| | |
|---|---|
| **Manufacturer, Model, and Revision** | Fisher Controls DLC3010 Device Revision1 |
| **Device type** | Transmitter, two-wire: Liquid level, Interface Level, or Density |
| **HART Protocol revision** | 5.2 |
| **Device Description Available?** | Yes |
| **Number and type of process connections** | 3: Mechanical / magnetic field for buoyancy sensor,<br>Internal sensor for electronics temperature,<br>2 or 3 wire RTD connection for process temperature. |
| **Number and type of host connections** | 1: 4-20 mA output, Bell 202 FSK |
| **Number of Device Variables** | None exposed, 3 mapped to Dynamic, 5 potential |
| **Number of Dynamic Variables** | 3 |
| **Mappable Dynamic Variables?** | Yes. (PV is mappable. SV and TV are fixed.) |
| **Number of Supported Common Practice Commands** | 18 |
| **Number of Device-Specific Commands** | 24 |
| **Bits of additional device status** | 14 |
| **Burst-Mode?** | Yes |
| **Capture Device Variables?** | No |
| **Write protection?** | Yes |

# Annex B: Default Configuration

| Variable | Default Value |
|---|---|
| Weight Unit | pound |
| Volume Unit | $inch^3$ |
| Length Unit | inch |
| Torque Rate Unit | lbf*in/degree |
| Temperature Unit | degree C |
| Displacer Weight | 4.75 (pounds) |
| Displacer Volume | 99. 0 ($inch^3$) |
| Displacer Length | 14. 0 (inch) |
| Driver Rod Length | 8.0 (inch) |
| Torque Rate | 8.67 (lbf*inch/degree) |
| Sensor Mounting | right of displacer |
| Zero Reference Angle | 0.0 (degree) |
| Density of Upper Fluid | 0.0 (SGU) |
| Density of Lower fluid | 1.0 (SGU) |
| PV assignment | Liquid Level |
| PV Unit | inch |
| Level Offset | 0. 0 (inch) |
| Upper Range Value | 14.0 (inch) |
| Lower Range Value | 0. 0 (inch) |

| Variable | Default Value |
|---|---|
| PV Damping | 0.2 (second) |
| Input Filter Time Constant | 0.0 (second) |
| PV Hi Hi Alert Threshold | 14. 0 (inch) |
| PV Hi Alert Threshold | 13. 3 (inch) |
| PV Lo Alert Threshold | 0. 7 (inch) |
| PV Lo Lo Alert Threshold | 0. 0 (inch) |
| PV Alert Deadband | 0. 0 (inch) |
| Electronics Temperature Alert Hi Threshold | 80.0 (degree C) |
| Electronics Temperature Alert Lo Threshold | -40.0 (degree C) |
| Process Temperature Alert Hi Threshold | 232.2 (degree C) |
| Process Temperature Alert Lo Threshold | -200.0 (degree C) |
| Temperature Alert Deadband | 1.0 (degree C) |
| PV Alerts | All disabled |
| Temperature Alerts | All disabled |
| Write Protection | Not Protected |
| LCD Meter Installed | Yes |
| LCD Meter Decimal Point | 1 digit after decimal point |
| LCD Meter Display Mode | Alternate: PV, Percent Range |
| RTD Type | No RTD Installed |
| RTD Wire Resistance | 0.0 (Ohms) |
| Final Assembly Number | 0 |
| Instrument Serial Number | (nameplate value) |
| Displacer Serial Number | (Nameplate value if mounted to sensor, blank otherwise) |
| Tag | (blank, or as specified on requisition) |
| Descriptor | (blank, or as specified on requisition) |
| Date | (Production Date) |
| Message | (blank, or as specified on requisition) |
| Number of Response Preambles | 5 |
| Number of Request Preambles | 5 |
| Polling Address | 0 |
| Burst Mode | Off |
| Burst Command Number | 3 |

# Annex C DLC3010 Parameters as part of a Rosemount 1410 / 1420 WirelessHART Gateway

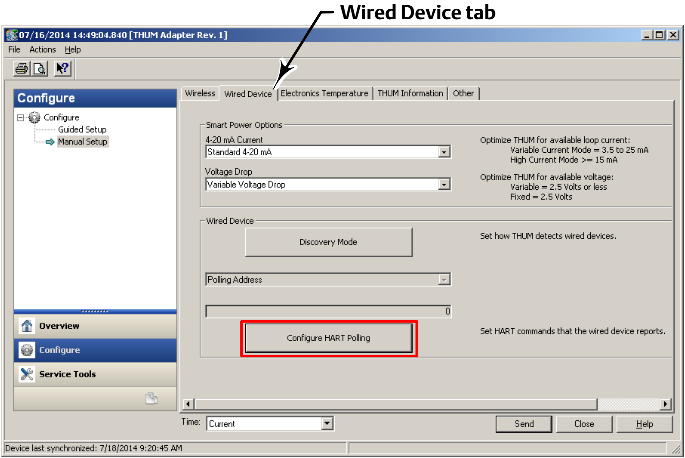
A FIELDVUE DLC3010 can join a wireless network through the addition of a Rosemount 775 THUM WirelessHART adapter. The wireless adapter acts both as a HART modem for communications coming to the DLC3010 from application software and as an independent master issuing commands periodically to the wired device pertaining to it's status. This independently gathered status information is relayed back to the Rosemount 1410 or 1420 Wireless Gateway and is made available to the user either through viewing the HTML interface or via mapping as "Published Data" parameters via the Gateway's MODBUS or OPC outputs.

The PV, SV, and TV variables and Additional Status bytes can be published by using the following steps.

Configure the THUM as follows:

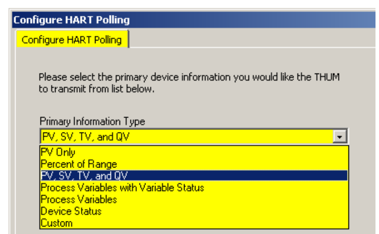
**Navigate to Configure > Manual Setup**
**Under the *Wired Device* tab select *Configure HART Polling***

**Wired Device tab**

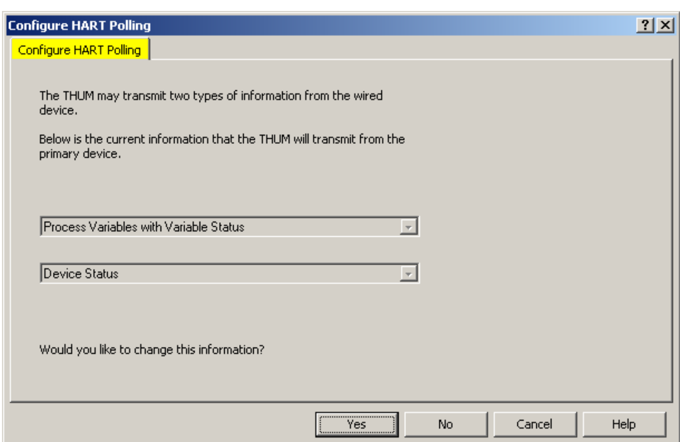



**FOR PRIMARY DEVICE INFORMATION:**

**Select PV, SV, TV, and QV**

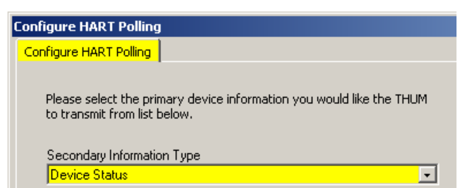



**Select Yes if changes are required**




**FOR SECONDARY DEVICE INFORMATION:**

**Select Device Status**

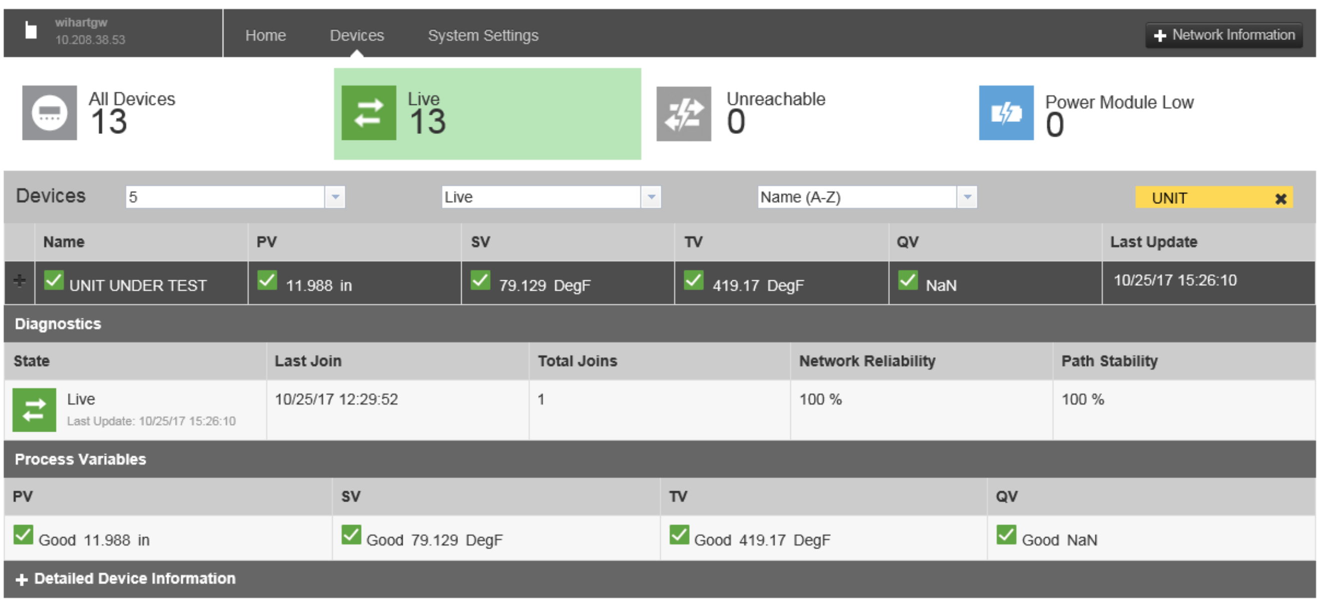


The FIELDVUE instrument is defined on the Gateway by "Tag" (read from the device's "Message" field).

# Smart Wireless Gateway

For each Tag, the Gateway provides updated values for:

- Process Variables
- Additional Status
- Published Data




Published Data, used for the Gateway's OPC and Modbus outputs, falls into the following categories:

## Field Device Identification Values:

Values that define the identity of the DLC3010. These include:

- MANUFACTURER (for "Fisher Controls" the value is "19")
- DEVICE_TYPE (for a DLC3010 the value is "4")
- DEVICE_REVISION
- HARDWARE_REVISION
- SOFTWARE_REVISION
- DEVICE_ID
- UNIVERSAL_REVISION (HART version)
- REQUEST_PREAMBLES
- RESPONSE PREAMBLES

## Variable information:

The Gateway receives updates concerning the DLC3010's configured Dynamic Variables. The information for any given variable is displayed by Dynamic Variable indicator ("PV", "SV", "TV", etc.).

| Designation | Variable ID | Assignment |
|---|---|---|
| PV | 0 | Liquid Level |
| | 1 | Interface Level |
| | 2 | Density |
| SV | 4 | Electronics Temperature |
| TV | 3 | Process Temperature |
| CURRENT | 245 | Loop Current |
| QV | 249 | (Not Used) |

Any of these choices of Dynamic Variable number or designation above can be substituted for the "#" sign in the Variable Identifier fields below.

| Variable Identifier | Comments / Explanation |
|---|---|
| # | See any of the variable identifiers in the table immediately above |
| #_CLASS | Always "0" (No Class assigned) |
| #_CODE | The "Device Variable ID" number in the table immediately above |
| #_HEALTHY | A "true" or "false" indication of the health of the sensor providing this value |
| #_STATUS | Always "192". (Device Variable Status is not defined at HART 5) |
| #_UNITS | Decimal "Variable Units Code" number from the "Units Codes" sections of table 6 |

## Device Status:

The eight standard status conditions (present in ANY HART field device) are all represented by a Boolean "true" or "false" state. These eight conditions, discussed in table 2a, are:

- DEVICE_MALFUNCTION
- CONFIGURATION_CHANGED
- COLD_START
- MORE_STATUS_AVAILABLE
- LOOP_CURRENT_FIXED
- LOOP_CURRENT_SATURATED
- NONPRIMARY_VALUE_OUT_OF_LIMITS
- PRIMARY_VALUE_OUT_OF_LIMITS

## Detailed Device Alerts:

The Alert Groupings are:

- ADDITIONAL_STATUS_0
- ADDITIONAL_STATUS_1
- ADDITIONAL_STATUS_2

Each of these status bytes represent eight individual bits with values that range from “00” to “255”. To determine which of the eight bits are active requires converting a decimal value to its binary equivalent value. Refer to Command 48 and table 2b for details on the individual alert bits inside of each Additional Status Byte.

Emerson Automation Solutions
Marshalltown, Iowa 50158 USA
Sorocaba, 18087 Brazil
Cernay, 68700 France
Dubai, United Arab Emirates
Singapore 128461 Singapore

www.Fisher.com