ALICAT SCIENTIFIC

A Halma company

ALICAT MODBUS MANUAL

Contact Information

World Headquarters,
Tucson, Arizona, USA
info@alicat.com
alicat.com
7641 N Business Park Dr.,
Tucson, AZ 85743 USA
+1 888-290-6060

Europe
europe@alicat.com
Geograaf 24
6921 EW Duiven
The Netherlands
+31(0) 26 203.1651

China & SE Asia
info-cn@alicat.com
alicat.com.cn
3rd Floor, Building 18,
FAMILY Science and
Technology Innovation
Park, No. 155, Yuanke Road
Minhang District
Shanghai, PRC 201109
+86 21-6151-9020

India
india@alicat.com
Alicat Scientific India Pvt. Ltd.
101, Hamilton A Bldg,
Near Hiranandani Hospital,
Hiranandani Estate, Patli
Pada, Ghodbunder Road,
Thane West-400607
Maharashtra, India
+91 22460-81434

DOC-MANUAL-Modbus, Rev 7, November 2024

Contents

Alicat Modbus Instruments

Modbus is a widely available protocol for connecting industrial devices. It is a master/slave protocol, meaning a device operating as a master will poll one or more devices operating as a slave. A master device writes data to, and reads data from, a slave device's registers.

Modbus data is most often read and written as 16-bit "registers," which are typically either signed or unsigned 16-bit integer format. Registers can be either read-write or read-only. When a 32-bit integer or floating point is required, the values are read from a pair of registers. The request and response formats are documented in the "Modicon Modbus Protocol Reference Guide" (https://www.modbus.org/docs/PI_MBUS_300.pdf).

Modbus RTU Instruments

Modbus RTU requires an RS-232 or RS-485 serial port and is compatible with most standard device configurations. Alicat Modbus RTU instruments are available with a variety of connectors, including 8-Pin Mini-DIN, 6-pin Industrial Connectors, DB9, DB15, etc. Refer to your instrument's hardware manual for the receive (Rx) and transmit (Tx) pins for your instrument's particular serial ports.

In Modbus RTU, data is transmitted serially in 8-bit bytes. Each slave device must have a unique device address, or unit number. Each exchange of data consists of a request from the master, followed by a response from the slave. Find more information on the protocol is available in the "Modbus over Serial Line Specification and Implementation Guide" (https://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf).

Connecting to a Master

The Modbus master device sends commands to, and reads data from, the Alicat instrument. Various devices can serve as a Modbus master, including a computer, PLC, etc.

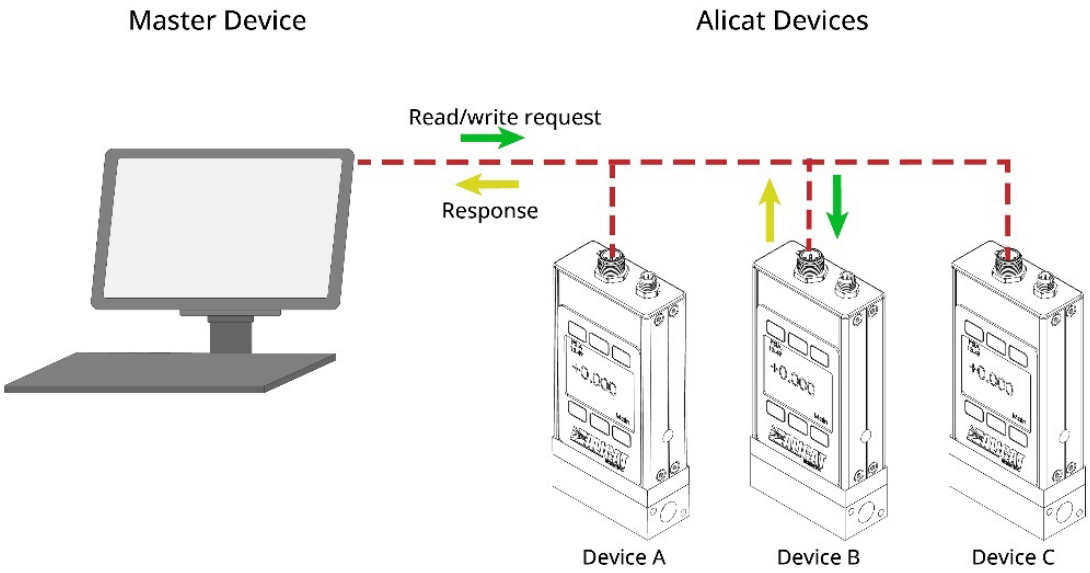
Alicat instruments typically have the following communications settings:

- **Baud:** 19200
- **Data bits:** 8
- **Parity:** none
- **Stop bits:** 1
- **Flow control:** none

✓ ***Note**: Alicat instruments require special firmware to support the even parity configuration that is required by the Modbus RTU specification. Alicat instruments typically support no parity, which is also recommended by the Modbus RTU specification.*

To connect an Alicat instrument to a Modbus master:

1. Connect the Alicat instrument(s) to the master device and network per the diagram below:



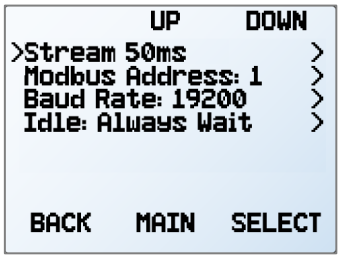
2. Alicat Modbus RTU instruments communicate via RS-232 or RS-485 over a serial port. Instruments with a display have an additional serial configuration menu that is not covered in the standard instrument menu. To access the menu, choose **MENU > SETUP > RS-232** or **MENU > SETUP > RS-485**. Choose the following settings:

   Unit ID / Stream: The Unit ID and Streaming Interval are not applicable for Modbus RTU.

   Modbus address: Each instrument in the network must have a unique ID, from 1 to 247.

   Baud rate: The speed of information transfer over the serial connection. Alicat instruments have a default baud rate of 19200 baud (bits per second). If the master computer or software uses a different baud rate, either change the instrument's baud rate in the **BAUD RATE** menu. Or, change the master computer's baud rate using Windows® Device Manager. Baud rate changes take effect once **SET** is pressed. Some software may need to restart to recognize the change.

   Idle: If the connection is idle (i.e., no valid request is received) for a specified amount of time, the controller can either move to a zero setpoint or maintain the previous setpoint. The idle time will be infinite by default (**Idle: Always Wait**) and can be set up to 99999.9 seconds (1 day, 3 hours, 46 minutes, 39.9 seconds).



If a connection cannot be established, contact Alicat support (see page 2).

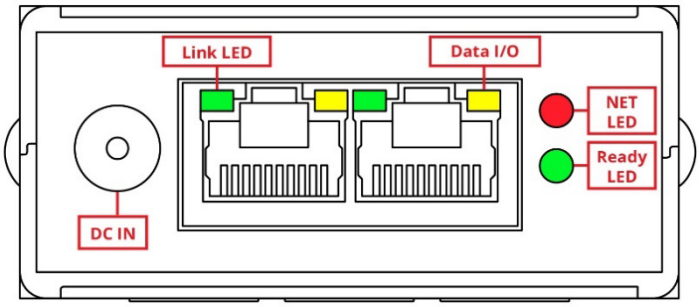
***Note**: When changing the instrument's baud rate, the baud rate for the COM port must change in time to successfully receive the response confirming the new baud rate. If it does not, then the baud rate will change on the instrument, but the response from the serial terminal will not be understood. Since the serial terminal is set to communicate at a specific baud rate during the initial configuration, you will have to redefine the terminal baud rate to continue communications.*

Modbus TCP/IP Instruments

Modbus TCP/IP instruments can communicate via RJ45 Ethernet ports, or via M12 Ethernet ports for waterproof communication. Alicat Modbus TCP/IP instruments have two 10/100Mbps Ethernet ports with an embedded switch. Either port can be used to communicate with the instrument. In addition, the embedded switch supports packet forwarding. Packet forwarding can only change direction with a power cycle; therefore, a ring topology with failover is not supported.

Modbus TCP encapsulates each Modbus RTU request or response data packet into one TCP/IP packet, which is transmitted over an Ethernet network. More information is available in the Modbus Messaging Implementation Guide.

RJ45 Ethernet Ports



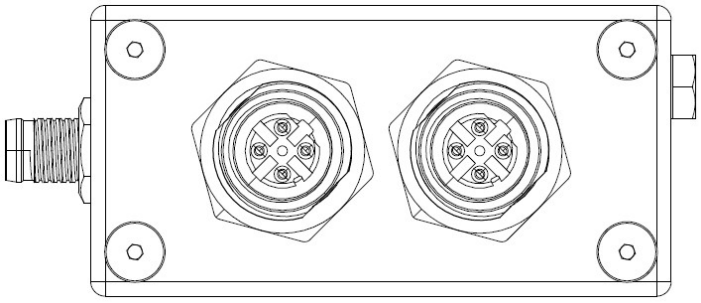
Alicat instruments with Modbus, with RJ45 ports

Each RJ45 port has two LED lights to show when the port is actively connected to the network and when it is transmitting or receiving data. The **Link LED** (green light) shows as a solid green when the port is connected to the network. If the **Link LED** is off, the instrument cannot connect to the network. The **Data I/O LED** (yellow light) blinks as data packets are transmitted or received.

To the right of the two RJ45 ports are the **NET LED** and **Ready LED**. The **NET LED** (the light closer to the back of the instrument) shows when the instrument is connected to a PLC. The **Ready LED** (the light closer to the display or front of the instrument) shows the instrument's operation status. For more information, refer to the following table.

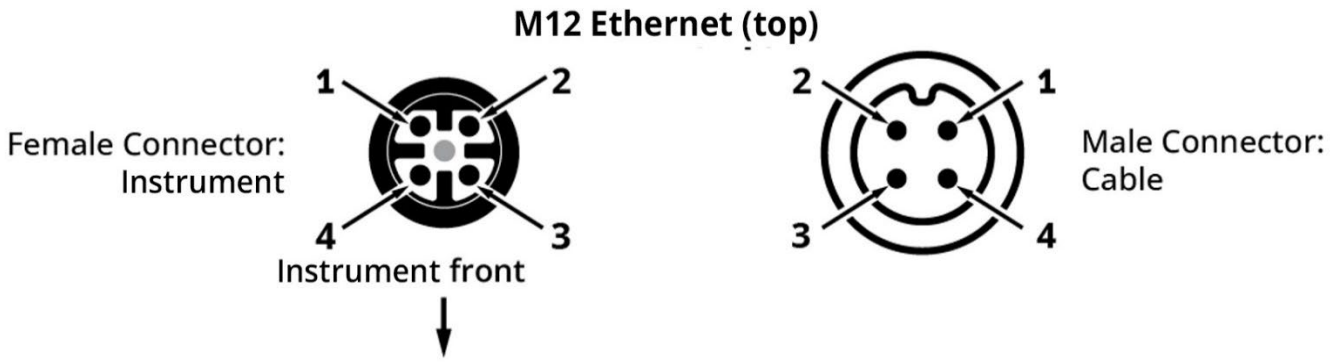
LED Status	Ready LED	NET LED
Off	Startup	Startup
Solid Red	Invalid IP configuration	No active link
Blinking Green	Instrument ready	N/A
Solid Green	N/A	Link active

M12 Ethernet and M8 Power Ports

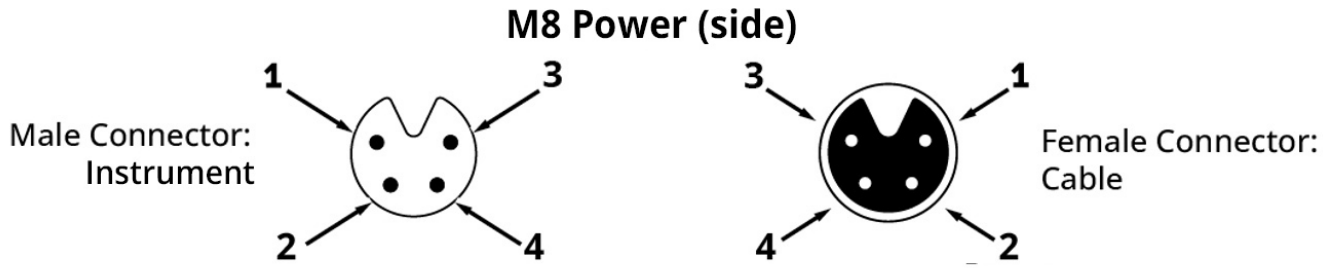


Alicat instruments with Modbus and M12/M8 ports

Alicat Modbus instruments with the M12/M8 Ethernet port configuration have the following pinouts.



Pin	4-Pin M12
1	Tx+: Send data +
2	Rx+: Receive data+
3	Tx-: Send data -
4	Rx-: Receive data -



Pin	4-Pin M8
1	Reserved. Do not connect.
2	Power in: powers the instrument.
3	Reserved. Do not connect.
4	Ground: common ground for power & digital communications.

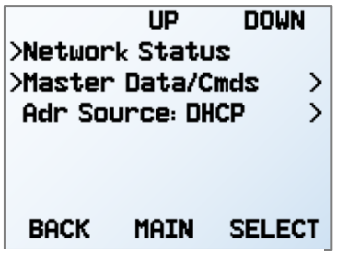
Connecting to a Master

Modbus TCP requires that you know or define IP addresses on the network.

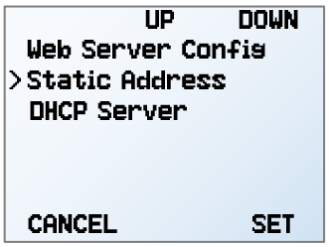
Assign the IP Address

Typically, the IP address can be set through the instrument's front panel screen:

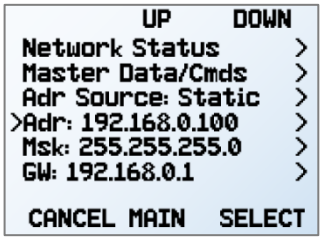
1. Choose **Menu > SETUP > Network**.
2. Choose **ADR Source** and press **SELECT** to choose how the IP address will be set:
   - **Web Server Config** will use settings in the built-in web server to determine the address.
   - **Static Address** will allow the address to be set on the display.
   - **DHCP** will dynamically obtain an address from a DHCP server on a reachable network.
3. Press **SET**. If the source has changed, the instrument will now restart the Ethernet interface to use the new address source. Modbus will be unable to communicate with the instrument for up to 15 seconds.
4. Choose **ADR** and click **SELECT**.
5. For a static address, enter the desired IP address and press **SET**. The instrument will restart the Ethernet interface to use the new address, so Modbus will be unable to communicate with the instrument for up to 15 seconds.
6. Similarly, enter the subnet mask (MSK) and gateway (GW) with appropriate values for your network.



Modbus TCP/IP Network menu showing DHCP option.



ADR Source options

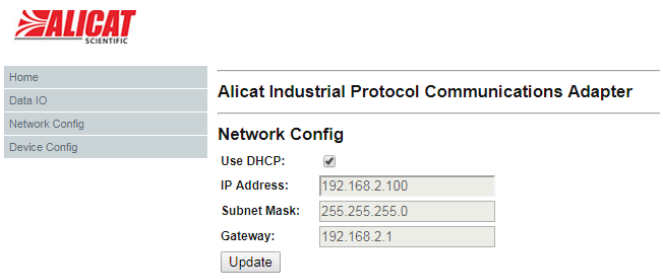


Network Status with Static IP settings.

Assign the IP Address for an Instrument without a Screen

If your Alicat instrument does not have a front panel display, the IP address can be set using the included web server. To access the instrument web server, you will need to know the current IP address of the instrument. By default, Alicat instruments acquire their IP address dynamically from a DHCP server. However, many installations will prefer to use a static IP address. In this case, use the dynamic address to establish communication with the instrument, then set the static IP address using the instruments embedded webserver:

1. Open the active DHCP server and see the current IP address of the Alicat instrument. If a DHCP server is not available, see the Creating a DHCP Server section (page 13) before continuing to Step 2.
2. Using a web browser, enter **http:// [IP address]** . The Alicat Industrial Protocol Communication Adapter interface window will open to the Home tab.

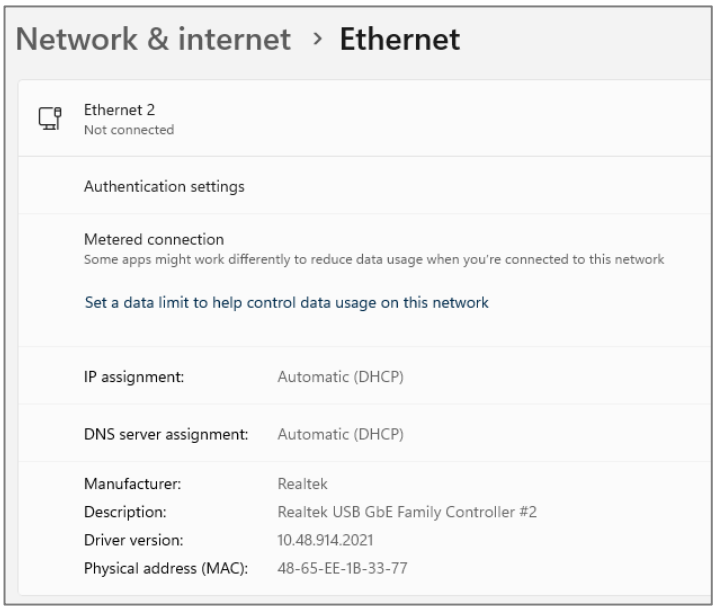


3. Click the **`Network Config`** tab to display the following page, which will include the current, dynamically assigned IP address:
4. Uncheck the **Use DHCP** box to stop using the dynamic address.
5. Enter an appropriate Static IP address.
6. If required, enter/update the appropriate Subnet Mask and/or Gateway.
7. Click **Update**. The instrument will reconfigure for approximately 15 seconds, during which communication will be unavailable.
8. When configuration is complete, complete, the web server will now be available at the new IP address, and the new network configuration settings will be visible in the Industrial Protocol Communication Adapter window.

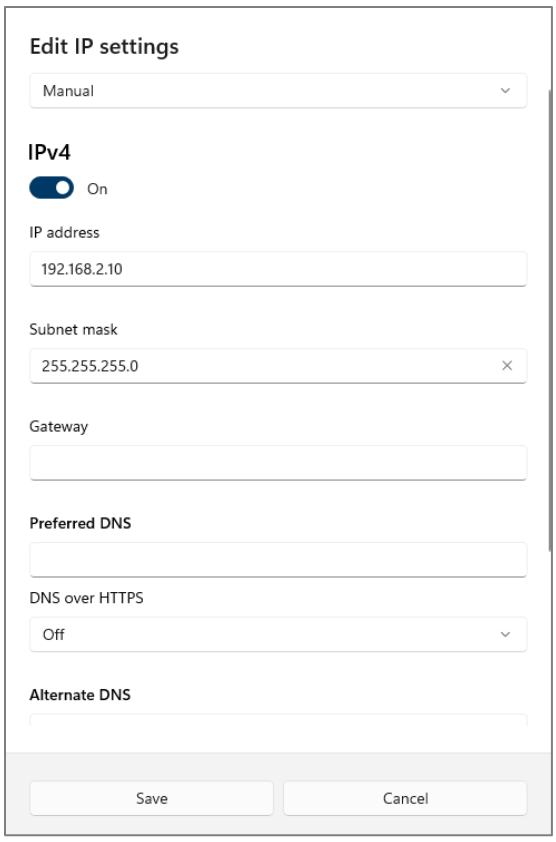
Creating a DHCP server

If a DHCP server has not been set up, you can temporarily set up a server in order to communicate with an device that does not have a screen.

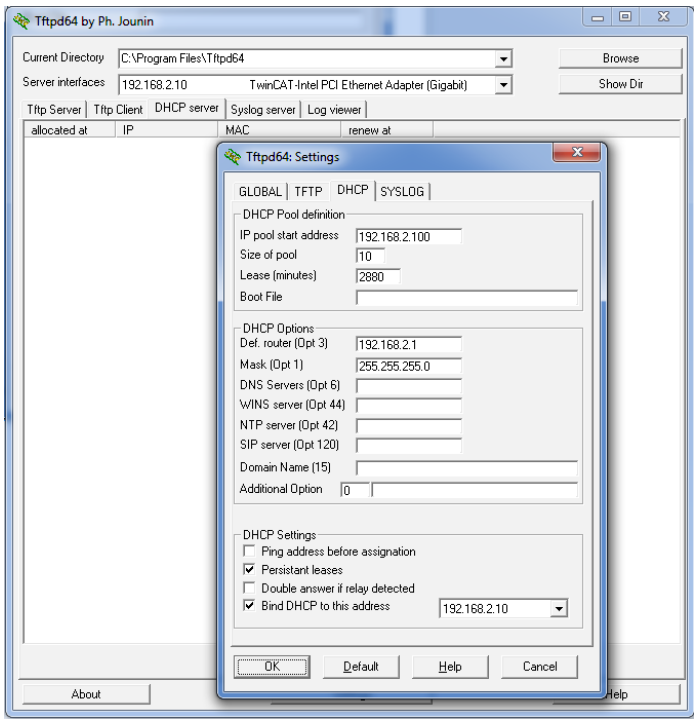
1. On a Windows computer with an unused Ethernet port, open **Settings > Network & Internet**.



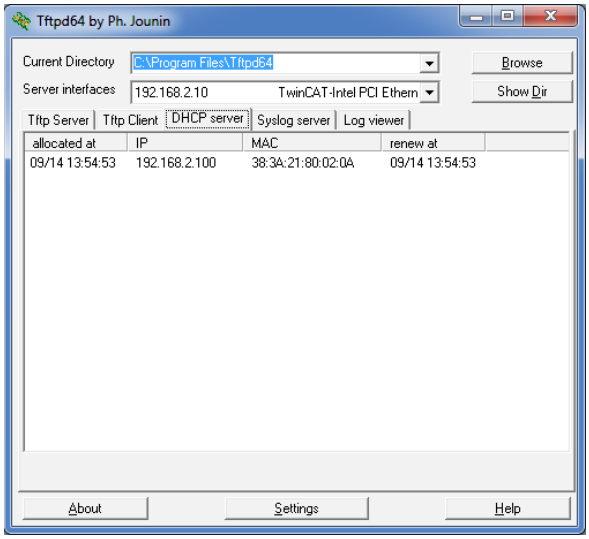
2. Assign a static IP to the unused port (i.e., network interface). Most often, the Subnet Mask will be 255.255.255.0 and gateway will not be set.



3. Configure a DHCP server to give a known IP address to the device on the static network. This example uses the open source Tftpd64 tool (https://pjo2.github.io/tftpd64/):



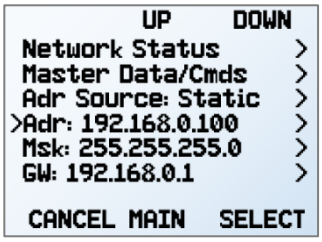
4. Connect your Alicat instrument to the static port.
5. Supply power to the instrument.
6. The address allocated to the instrument by the DHCP server will now appear:



7. The IP address assigned to the Alicat can now be used to communicate with the instrument (page 12).

Modbus TCP/IP Information

Alicat Modbus instruments communicating via TCP/IP have an extra configuration menu, accessible by choosing **MENU > SETUP > Network Setup.**



Network Status with Static IP settings.

Network Status

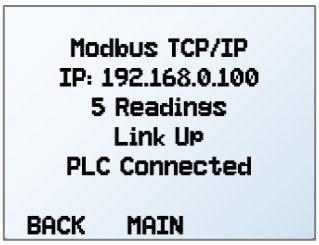
The **Network Status** screen displays various diagnostic information for troubleshooting the network connection. From the Network Status screen, select **Network Status** to access the following information:

IP: The IP address will be shown, indicating that the connection is ready for use. **Configuring** will be shown during startup until the Ethernet port is ready. **Waiting For Network** indicates that the Ethernet cable is not connected, or that the instrument is waiting for the DHCP server to give it an IP address.

N Readings: The number of numeric readings currently available (1–7).

Link: **Link Up** will be displayed when the instrument is connected to the Ethernet network. **Link Down** indicates that no Ethernet cable has been connected to a port.

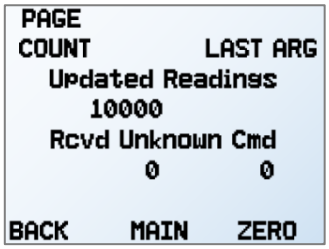
PLC: **PLC Connected** indicates that a connection has been established with a master device. **PLC Not Connected** indicates that there is currently no such connection.



Network Status screen.

Master Data/CMDS

The Master Data/CMDS section can be useful for troubleshooting the network connection. From the Network Status screen, select **Master Data/Cmds** to access the diagnostic data.



Master Data/Cmds screen diagnostics.

Modbus Interface

✓ ***Note**: For more information on Modbus RTU communication commands, please see the Modbus FAQ at alicat.com/using-your-alicat/faq-modbus/*

Functions

The following Modbus functions are supported:

Function Code	Register Address
3	Read holding registers (Modbus RTU only)
4	Read input registers
16	Write multiple registers

Use the instrument registers to interact with the instrument.

Note that the response to function code 16 (Write Multiple Registers) will echo the value provided by the master in the write and not the value of the register after the write. For example, if the setpoint register were written to a value that is not valid for the instrument, it will be adjusted inside the register; however, the value returned will be the original setpoint value. If the real value is desired, a read of the register must be issued. This is consistent with the Modbus specification and demanded by some Modbus implementations.

Reading undefined registers may return 65535 as the value, or may return a Modbus exception response with the ILLEGAL DATA ADDRESS exception code.

Registers

Register Number and Address

A register can be accessed by either a register number or a register address. The tables below include both pieces of data. Note that the Register Number is always one larger than the Register Address. For example, Register 1 means Register Address 0. Some tools use the register number, while others require the register address. If you are not sure what your tool uses and you are having difficulty communicating, try adjusting the register index up or down by 1.

Register Length and Byte Order

Modbus holding and input registers are 16-bit; 32-bit data can be stored using two registers. In the Alicat implementation, fields that span multiple registers have the most significant data in the register with the lower address / number (i.e., "big endian" format). The byte order for all 16-bit values is most-significant byte first. Floating point values are in IEEE-734 single precision floating point format.

Modbus registers have been provided to assist with assuring the correct interpretation of multi register values. Registers 1088 and 1089 (page 36) hold a fixed value that cannot be changed. Read these two registers to determine if the combined value is interpreted in a manner consistent with the device definition. Combine the values read from the two registers, formatting the result as an integer or as a single precision floating-point value. Then, find that combined value in the following table to determine if the bytes are combined in the expected order. If the combined value is not listed, incorrect registers were read (try adding or subtracting one from each address), or another, unusual error occurred.

Hexadecimal Value	4 byte Integer Value	Floating Point Value	Interpretation and Action
3F9E064B	1067320907	1.234567	Ordering matches device definition.
064B3F9E	105594782	3.8226795e-35	Register order is reversed. Swap bytes 1 and 2 with bytes 3 and 4.
9E3F4B06	2654948102 / -1640019194	-1.012697e-20	Register order is correct, but bytes within each register are reversed. Swap bytes 3 and 4, as well as bytes 1 and 2.
4B069E3F	1258724927	8822335.0	Bytes within each register and register order are reversed. Swap bytes 1 and 4, as well as bytes 2 and 3.

Once the combined value read matches the device definition, writing a four-byte integer or a single precision floating-point value can be tested by writing into registers 1086 and 1087, then reading back the intended value with the multi register interpreter that is now known good (these registers do not impact device function in any way). If the device returns an address error when writing to the test registers, add or subtract one from each address.

Standard Readings Registers

Standard readings keep the most common readings in the same registers, regardless of the configuration of the device. Readings that are not valid for the current configuration are indicated with a special value. Any integer readings that are not valid will be read as -2147483648. Any floating point readings that are not valid will be read as hex ffffffff, a quiet NaN.

Current Readings, Integer Format

Register Number	Register Address	Action	Data Type	Description
1296	1295	Read	2 byte bit field	Alarm output status, +1 if alarm 1 is active, +2 if alarm 2 is active. *Added in 10v07.0.*
1297	1296	Read	2 byte unsigned integer	Gas number. *Added in 10v07.0.*
1298–1299	1297–1298	Read	4 byte bit field	Device status. Each of the four Device Status registers have the following meaning for their bits: +1: Temperature overrange active +2: Temperature underrange active +4: Volumetric flow overrange active +8: Volumetric flow underrange active +16: Mass flow overrange active +32: Mass flow underrange active +64: Pressure overrange active +128: Totalizer reached maximum limit active +256: Control loop is being held constant +512: Internal hardware error +1024: Pressure is over user configured limit +2048: Flow overrange during totalization Other bits are reserved and may be 0 or 1 *Added in 10v07.0.*
1300–1301	1299–1300	Read / Write	4 byte integer	Request a new setpoint or read the current setpoint. Scale integer value by 10^(-register 1651 value) for value in engineering units. *Added in 10v07.0.*
1302–1303	1301–1302	Read	4 byte integer	Current valve drive. Scale integer value by 10^(-register 1663 value) for value in engineering units. *Added in 10v07.0.*

1304–1305	1303–1304	Read	4 byte integer	Current pressure reading. This is the current primary pressure value. Scale integer value by 10^(-register 1675 value) for value in engineering units. *Added in 10v07.0.*
1306–1307	1305–1306	Read	4 byte integer	Current secondary pressure reading. This is the current secondary pressure value, if any. This may be from a second pressure sensor or calculated using the barometer, if one is available. Scale integer value by 10^(-register 1687 value) for value in engineering units. *Added in 10v07.0.*
1308–1309	1307–1308	Read	4 byte integer	Current barometric pressure reading. Scale integer value by 10^(-register 1699 value) for value in engineering units. *Added in 10v07.0.*
1310–1311	1309–1310	Read	4 byte integer	Current temperature reading. Scale integer value by 10^(-register 1711 value) for value in engineering units. *Added in 10v07.0.*
1312–1313	1311–1312	Read	4 byte integer	Current volumetric flow reading. Scale integer value by 10^(-register 1723 value) for value in engineering units. *Added in 10v07.0.*
1314–1315	1313–1314	Read	4 byte integer	Current mass flow reading. Scale integer value by 10^(-register 1735 value) for value in engineering units. *Added in 10v07.0.*
1316–1317	1315–1316	Read	4 byte integer	Current totalizer 1 reading. Scale integer value by 10^(-register 1747 value) for value in engineering units. *Added in 10v07.0.*
1318–1319	1317–1318	Read	4 byte integer	Current totalizer 2 reading. Scale integer value by 10^(-register 1759 value) for value in engineering units. *Added in 10v07.0.*
1320–1321	1319–1320	Read	4 byte integer	Current humidity reading. Scale integer value by 10^(-register 1771 value) for value in engineering units. *Added in 10v07.0.*

Current Readings, Float Format

1346	1345	Read	2 byte bit field	Alarm output status, +1 if alarm 1 is active, +2 if alarm 2 is active. *Added in 10v07.0.*
1347	1346	Read	2 byte unsigned integer	Gas number. *Added in 10v07.0.*
1348–1349	1347–1348	Read	4 byte bit field	Device status. See register 1298 for the description (page 17). *Added in 10v07.0.*
1350–1351	1349–1350	Read / Write	4 byte float	Request a new setpoint or read the current setpoint in engineering units. *Added in 10v07.0.*
1352–1353	1351–1352	Read	4 byte float	Current valve drive in engineering units. *Added in 10v07.0.*
1354–1355	1353–1504	Read	4 byte float	Current pressure reading in engineering units. This is the current primary pressure value. *Added in 10v07.0.*
1356–1357	1355–1356	Read	4 byte float	Current secondary pressure reading in engineering units. This is the current secondary pressure value, if any. This may be from a second pressure sensor or calculated using the barometer, if one is available. *Added in 10v07.0.*
1358–1359	1357–1358	Read	4 byte float	Current barometric pressure reading in engineering units. *Added in 10v07.0.*
1360–1361	1359–1360	Read	4 byte float	Current temperature reading in engineering units. *Added in 10v07.0.*
1362–1363	1361–1362	Read	4 byte float	Current volumetric flow reading in engineering units. *Added in 10v07.0.*
1364–1365	1363–1364	Read	4 byte float	Current mass flow reading in engineering units.

				Added in 10v07.0.
1366–1367	1365–1366	Read	4 byte float	Current totalizer 1 reading in engineering units. *Added in 10v07.0.*
1368–1369	1367–1368	Read	4 byte float	Current totalizer 2 reading in engineering units. *Added in 10v07.0.*
1370–1371	1369–1370	Read	4 byte float	Current humidity reading in engineering units. *Added in 10v07.0.*

Setpoint Information

Register Number	Register Address	Action	Data Type	Description
1640	1639	Read	2 byte integer	Setpoint type; one of the statistic values from Appendix A (page 84). *Added in 10v07.0.*
1641	1640	Read	2 byte integer	Setpoint source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1642–1643	1641–1642	Read	4 byte integer	The minimum value of the setpoint, without overrange. Scale integer value by 10^(-register 1651 value) for value in engineering units. *Added in 10v07.0.*
1644–1645	1643–1644	Read	4 byte integer	The maximum value of the setpoint, without overrange. Scale integer value by 10^(-register 1651 value) for value in engineering units. *Added in 10v07.0.*
1646–1647	1645–1646	Read	4 byte float	The minimum value of the setpoint, without overrange, in engineering units. *Added in 10v07.0.*
1648–1649	1647–1648	Read	4 byte float	The maximum value of the setpoint, without overrange, in engineering units. *Added in 10v07.0.*
1650	1649	Read	2 byte integer	Setpoint units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1651	1650	Read	2 byte integer	Setpoint decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Valve Drive Information

Register Number	Register Address	Action	Data Type	Description
1652	1651	Read	2 byte integer	Valve drive type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1653	1652	Read	2 byte integer	Valve drive source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1654–1655	1653–1654	Read	4 byte integer	The minimum value of the valve drive, without overrange. Scale integer value by 10^(-register 1663 value) for value in engineering units. *Added in 10v07.0.*
1656–1657	1655–1656	Read	4 byte integer	The maximum value of the valve drive, without overrange. Scale integer value by 10^(-register 1663 value) for value in engineering units. *Added in 10v07.0.*
1658–1659	1657–1658	Read	4 byte float	The minimum value of the valve drive, without overrange, in engineering units. *Added in 10v07.0.*
1660–1661	1659–1660	Read	4 byte float	The maximum value of the valve drive, without overrange, in engineering units. *Added in 10v07.0.*
1662	1661	Read	2 byte integer	Valve drive units; one of the values from Appendix B (page 86), or 0 if the reading is not available.

				Added in 10v07.0.
1663	1662	Read	2 byte integer	Valve drive decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Pressure Information

Register Number	Register Address	Action	Data Type	Description
1664	1663	Read	2 byte integer	Pressure type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1665	1664	Read	2 byte integer	Pressure source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1666–1667	1665–1666	Read	4 byte integer	The minimum value of the pressure, without overrange. Scale integer value by 10^(-register 1675 value) for value in engineering units. *Added in 10v07.0.*
1668–1669	1667–1668	Read	4 byte integer	The maximum value of the pressure, without overrange. Scale integer value by 10^(-register 1675 value) for value in engineering units. *Added in 10v07.0.*
1670–1671	1669–1670	Read	4 byte float	The minimum value of the pressure, without overrange, in engineering units. *Added in 10v07.0.*
1672–1673	1671–1672	Read	4 byte float	The maximum value of the pressure, without overrange, in engineering units. *Added in 10v07.0.*
1674	1673	Read	2 byte integer	Pressure units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1675	1674	Read	2 byte integer	Pressure decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*
1676	1675	Read	2 byte integer	Secondary pressure type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1677	1676	Read	2 byte integer	Secondary pressure source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1678–1679	1677–1678	Read	4 byte integer	The minimum value of the secondary pressure, without overrange. Scale integer value by 10^(-register 1687 value) for value in engineering units. *Added in 10v07.0.*
1680–1681	1679–1680	Read	4 byte integer	The maximum value of the secondary pressure, without overrange. Scale integer value by 10^(-register 1687 value) for value in engineering units. *Added in 10v07.0.*
1682–1683	1681–1682	Read	4 byte float	The minimum value of the secondary pressure, without overrange, in engineering units. *Added in 10v07.0.*
1684–1685	1683–1684	Read	4 byte float	The maximum value of the secondary pressure, without overrange, in engineering units. *Added in 10v07.0.*
1686	1685	Read	2 byte integer	Secondary pressure units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1687	1686	Read	2 byte integer	Secondary pressure decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

1688	1687	Read	2 byte integer	Barometric pressure type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1689	1688	Read	2 byte integer	Barometric pressure source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1690–1691	1689–1690	Read	4 byte integer	The minimum value of the barometric pressure, without overrange. Scale integer value by 10^(-register 1699 value) for value in engineering units. *Added in 10v07.0.*
1692–1693	1691–1692	Read	4 byte integer	The maximum value of the barometric pressure, without overrange. Scale integer value by 10^(-register 1699 value) for value in engineering units. *Added in 10v07.0.*
1694–1695	1693–1694	Read	4 byte float	The minimum value of the barometric pressure, without overrange, in engineering units. *Added in 10v07.0.*
1696–1697	1695–1696	Read	4 byte float	The maximum value of the barometric pressure, without overrange, in engineering units. *Added in 10v07.0.*
1698	1697	Read	2 byte integer	Barometric pressure units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1699	1698	Read	2 byte integer	Barometric pressure decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Temperature Information

Register Number	Register Address	Action	Data Type	Description
1700	1699	Read	2 byte integer	Temperature type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1701	1700	Read	2 byte integer	Temperature source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1702–1703	1701–1702	Read	4 byte integer	The minimum value of the temperature, without overrange. Scale integer value by 10^(-register 1711 value) for value in engineering units. *Added in 10v07.0.*
1704–1705	1703–1704	Read	4 byte integer	The maximum value of the temperature, without overrange. Scale integer value by 10^(-register 1711 value) for value in engineering units. *Added in 10v07.0.*
1706–1707	1705–1706	Read	4 byte float	The minimum value of the temperature, without overrange, in engineering units. *Added in 10v07.0.*
1708–1709	1707–1708	Read	4 byte float	The maximum value of the temperature, without overrange, in engineering units. *Added in 10v07.0.*
1710	1709	Read	2 byte integer	Temperature units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1711	1710	Read	2 byte integer	Temperature decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Volumetric Flow Information

Register Number	Register Address	Action	Data Type	Description
1712	1711	Read	2 byte integer	Volumetric flow type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1713	1712	Read	2 byte integer	Volumetric flow source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*

1714–1715	1713–1714	Read	4 byte integer	The minimum value of the volumetric flow, without overrange. Scale integer value by 10^(-register 1723 value) for value in engineering units. *Added in 10v07.0.*
1716–1717	1715–1716	Read	4 byte integer	The maximum value of the volumetric flow, without overrange. Scale integer value by 10^(-register 1723 value) for value in engineering units. *Added in 10v07.0.*
1718–1719	1717–1718	Read	4 byte float	The minimum value of the volumetric flow, without overrange, in engineering units. *Added in 10v07.0.*
1720–1721	1719–1720	Read	4 byte float	The maximum value of the volumetric flow, without overrange, in engineering units. *Added in 10v07.0.*
1722	1721	Read	2 byte integer	Volumetric flow units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1723	1722	Read	2 byte integer	Volumetric flow decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Mass Flow Information

Register Number	Register Address	Action	Data Type	Description
1724	1723	Read	2 byte integer	Mass flow type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1725	1724	Read	2 byte integer	Mass flow source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1726–1727	1725–1726	Read	4 byte integer	The minimum value of the mass flow, without overrange. Scale integer value by 10^(-register 1735 value) for value in engineering units. *Added in 10v07.0.*
1728–1729	1727–1728	Read	4 byte integer	The maximum value of the mass flow, without overrange. Scale integer value by 10^(-register 1735 value) for value in engineering units. *Added in 10v07.0.*
1730–1731	1729–1730	Read	4 byte float	The minimum value of the mass flow, without overrange, in engineering units. *Added in 10v07.0.*
1732–1733	1731–1732	Read	4 byte float	The maximum value of the mass flow, without overrange, in engineering units. *Added in 10v07.0.*
1734	1733	Read	2 byte integer	Mass flow units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1735	1734	Read	2 byte integer	Mass flow decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Totalizer Information

Register Number	Register Address	Action	Data Type	Description
1736	1735	Read	2 byte integer	Totalizer 1 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1737	1736	Read	2 byte integer	Totalizer 1 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*

1738–1739	1737–1738	Read	4 byte integer	The minimum value of totalizer 1, without overrange. Scale integer value by 10^(-register 1747 value) for value in engineering units. *Added in 10v07.0.*
1740–1741	1739–1740	Read	4 byte integer	The maximum value of totalizer 1, without overrange. Scale integer value by 10^(-register 1747 value) for value in engineering units. *Added in 10v07.0.*
1742–1743	1741–1742	Read	4 byte float	The minimum value of totalizer 1, without overrange, in engineering units. *Added in 10v07.0.*
1744–1745	1743–1744	Read	4 byte float	The maximum value of totalizer 1, without overrange, in engineering units. *Added in 10v07.0.*
1746	1745	Read	2 byte integer	Totalizer 1units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1747	1746	Read	2 byte integer	Totalizer 1 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*
1748	1747	Read	2 byte integer	Totalizer 2 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1749	1748	Read	2 byte integer	Totalizer 2 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1750–1751	1749–1750	Read	4 byte integer	The minimum value of totalizer 2, without overrange. Scale integer value by 10^(-register 1759 value) for value in engineering units. *Added in 10v07.0.*
1752–1753	1751–1752	Read	4 byte integer	The maximum value of totalizer 2, without overrange. Scale integer value by 10^(-register 1759 value) for value in engineering units. *Added in 10v07.0.*
1754–1755	1753–1754	Read	4 byte float	The minimum value of totalizer 2, without overrange, in engineering units. *Added in 10v07.0.*
1756–1757	1755–1756	Read	4 byte float	The maximum value of totalizer 2, without overrange, in engineering units. *Added in 10v07.0.*
1758	1757	Read	2 byte integer	Totalizer 2 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1759	1758	Read	2 byte integer	Totalizer 2 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Humidity Information

Register Number	Register Address	Action	Data Type	Description
1760	1759	Read	2 byte integer	Humidity type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1761	1760	Read	2 byte integer	Humidity source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1762–1763	1761–1762	Read	4 byte integer	The minimum value of humidity, without overrange. Scale integer value by 10^(-register 1771 value) for value in engineering units. *Added in 10v07.0.*
1764–1765	1763–1764	Read	4 byte integer	The maximum value of humidity, without overrange. Scale integer value by 10^(-register 1771 value) for value in engineering units. *Added in 10v07.0.*
1766–1767	1765–1766	Read	4 byte float	The minimum value of humidity, without overrange, in engineering units. *Added in 10v07.0.*
1768–1769	1767–1768	Read	4 byte float	The maximum value of humidity, without overrange, in engineering units. *Added in 10v07.0.*
1770	1769	Read	2 byte integer	Humidity units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1771	1770	Read	2 byte integer	Humidity decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Optimized Readings Registers

Optimized readings are placed to minimize the number of registers that need to be read in order to retrieve all of the values the device supports. Readings that are not valid for the current configuration are indicated with a special value. Any integer readings that are not valid will be read as -2147483648. Any floating point readings that are not valid will be read as hex ffffffff, a quiet NaN.

Current Readings, Float Format

Register Number	Register Address	Action	Data Type	Description
1010–1011	1009–1010	Read / Write	4 byte float	Request a new setpoint or read the current setpoint. *Added in 6v17.0 (write only prior to 10v07.0).*
1199	1198	Read	2 byte bit field	Alarm output status, +1 if alarm 1 is active, +2 if alarm 2 is active. *Added in 10v07.0.*
1200	1199	Read	2 byte unsigned integer	Gas number. *Added in 6v17.0.*
1201–1202	1200–1201	Read	4 byte bit field	Device status. See register 1298 for the description (page 17). *Added in 6v17.0.*
1203–1204	1202–1203	Read	4 byte float	Reading 1.* *Added in 6v17.0.*
1205–1206	1204–1205	Read	4 byte float	Reading 2.* *Added in 6v17.0.*
1207–1208	1206–1207	Read	4 byte float	Reading 3.* *Added in 6v17.0.*
1209–1210	1208–1209	Read	4 byte float	Reading 4.* *Added in 6v17.0.*
1211–1212	1210–1211	Read	4 byte float	Reading 5.* *Added in 6v17.0.*
1213–1214	1212–1213	Read	4 byte float	Reading 6.* *Added in 6v17.0.*
1215–1216	1214–121	Read	4 byte float	Reading 7.* *Added in 6v17.0.*
1217–1218	1216–1217	Read	4 byte float	Reading 8.* *Added in 6v17.0.*
1219–1220	1218–1219	Read	4 byte float	Reading 9.* *Added in 6v17.0.*
1221–1222	1220–1221	Read	4 byte float	Reading 10.* *Added in 6v17.0.*
1223–1224	1222–1223	Read	4 byte float	Reading 11.* *Added in 6v17.0.*
1225–1226	1224–1225	Read	4 byte float	Reading 12.* *Added in 6v17.0.*
1227–1228	1226–1227	Read	4 byte float	Reading 13.* *Added in 6v17.0.*
1229–1230	1228–1229	Read	4 byte float	Reading 14.* *Added in 6v17.0.*
1231–1232	1230–1231	Read	4 byte float	Reading 15.* *Added in 6v17.0.*
1233–1234	1232–1233	Read	4 byte float	Reading 16.* *Added in 6v17.0.*
1235–1236	1234–1235	Read	4 byte float	Reading 17.* *Added in 6v17.0.*
1237–1238	1236–1237	Read	4 byte float	Reading 18.* *Added in 6v17.0.*
1239–1240	1238–1239	Read	4 byte float	Reading 19.* *Added in 6v17.0.*
1241–1242	1240–1241	Read	4 byte float	Reading 20.* *Added in 6v17.0.*

* Modbus RTU defines all 20 of these reading registers, regardless of the number of valid readings. It will return the invalid value for readings that do not exist. In versions prior to 10v07.0, Modbus TCP/IP will return an invalid address for the register pair if the associated reading does not exist.

Current Readings, Integer Format

1246	1245	Read	2 byte bit field	Alarm output status, +1 if alarm 1 is active, +2 if alarm 2 is active. *Added in 10v07.0.*
1247	1246	Read	2 byte unsigned integer	Gas number. *Added in 10v07.0.*
1248–1249	1247–1248	Read	4 byte bit field	Device status. See register 1298 for the description (page 17). *Added in 10v07.0.*
1250–1251	1249–1250	Read	4 byte integer	Reading 1 Scale integer value by 10^(-register 1411 value) for value in engineering units. *Added in 10v07.0.*

1252–1253	1251–1252	Read	4 byte integer	Reading 2 Scale integer value by 10^(-register 1423 value) for value in engineering units. *Added in 10v07.0.*
1254–1255	1253–1254	Read	4 byte integer	Reading 3 Scale integer value by 10^(-register 1435 value) for value in engineering units. *Added in 10v07.0.*
1256–1257	1255–1256	Read	4 byte integer	Reading 4 Scale integer value by 10^(-register 1447 value) for value in engineering units. *Added in 10v07.0.*
1258–1259	1257–1258	Read	4 byte integer	Reading 5 Scale integer value by 10^(-register 1459 value) for value in engineering units. *Added in 10v07.0.*
1260–1261	1259–1260	Read	4 byte integer	Reading 6 Scale integer value by 10^(-register 1471 value) for value in engineering units. *Added in 10v07.0.*
1262–1263	1261–1262	Read	4 byte integer	Reading 7 Scale integer value by 10^(-register 1483 value) for value in engineering units. *Added in 10v07.0.*
1264–1265	1263–1264	Read	4 byte integer	Reading 8 Scale integer value by 10^(-register 1495 value) for value in engineering units. *Added in 10v07.0.*
1266–1267	1265–1266	Read	4 byte integer	Reading 9 Scale integer value by 10^(-register 1507 value) for value in engineering units. *Added in 10v07.0.*
1268–1269	1267–1268	Read	4 byte integer	Reading 10 Scale integer value by 10^(-register 1519 value) for value in engineering units. *Added in 10v07.0.*
1270–1271	1269–1270	Read	4 byte integer	Reading 11 Scale integer value by 10^(-register 1531 value) for value in engineering units. *Added in 10v07.0.*
1272–1273	1271–1272	Read	4 byte integer	Reading 12 Scale integer value by 10^(-register 1543 value) for value in engineering units. *Added in 10v07.0.*
1274–1275	1273–1274	Read	4 byte integer	Reading 13 Scale integer value by 10^(-register 1555 value) for value in engineering units. *Added in 10v07.0.*
1276–1277	1275–1276	Read	4 byte integer	Reading 14 Scale integer value by 10^(-register 1567 value) for value in engineering units. *Added in 10v07.0.*
1278–1279	1277–1278	Read	4 byte integer	Reading 15 Scale integer value by 10^(-register 1579 value) for value in engineering units. *Added in 10v07.0.*
1280–1281	1279–1280	Read	4 byte integer	Reading 16 Scale integer value by 10^(-register 1591 value) for value in engineering units. *Added in 10v07.0.*
1282–1283	1281–1282	Read	4 byte integer	Reading 17 Scale integer value by 10^(-register 1603 value) for value in engineering units. *Added in 10v07.0.*
1284–1285	1283–1284	Read	4 byte integer	Reading 18 Scale integer value by 10^(-register 1615 value) for value in engineering units. *Added in 10v07.0.*
1286–1287	1285–1286	Read	4 byte integer	Reading 19

				Scale integer value by 10^(-register 1627 value) for value in engineering units. *Added in 10v07.0.*
1288–1289	1287–1288	Read	4 byte integer	Reading 20 Scale integer value by 10^(-register 1639 value) for value in engineering units. *Added in 10v07.0.*

Reading 1 Information

Register Number	Register Address	Action	Data Type	Description
1400	1399	Read	2 byte integer	Reading 1 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1401	1400	Read	2 byte integer	Reading 1 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1402–1403	1401–1402	Read	4 byte integer	The minimum value of reading 1, without overrange. Scale integer value by 10^(-register 1411 value) for value in engineering units. *Added in 10v07.0.*
1404–1405	1403–1404	Read	4 byte integer	The maximum value of reading 1, without overrange. Scale integer value by 10^(-register 1411 value) for value in engineering units. *Added in 10v07.0.*
1406–1407	1405–1406	Read	4 byte float	The minimum value of reading 1, without overrange, in engineering units. *Added in 10v07.0.*
1408–1409	1407–1408	Read	4 byte float	The maximum value of reading 1, without overrange, in engineering units. *Added in 10v07.0.*
1410	1409	Read	2 byte integer	Reading 1 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1411	1410	Read	2 byte integer	Reading 1 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 2 Information

1412	1411	Read	2 byte integer	Reading 2 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1413	1412	Read	2 byte integer	Reading 2 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1414–1415	1413–1414	Read	4 byte integer	The minimum value of reading 2, without overrange. Scale integer value by 10^(-register 1423 value) for value in engineering units. *Added in 10v07.0.*
1416–1417	1415–1416	Read	4 byte integer	The maximum value of reading 2, without overrange. Scale integer value by 10^(-register 1423 value) for value in engineering units. *Added in 10v07.0.*
1418–1419	1417–1418	Read	4 byte float	The minimum value of reading 2, without overrange, in engineering units. *Added in 10v07.0.*
1420–1421	1419–1420	Read	4 byte float	The maximum value of reading 2, without overrange, in engineering units. *Added in 10v07.0.*

1422	1421	Read	2 byte integer	Reading 2 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1423	1422	Read	2 byte integer	Reading 2 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 3 Information

1424	1423	Read	2 byte integer	Reading 3 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1425	1424	Read	2 byte integer	Reading 3 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1426–1427	1425–1426	Read	4 byte integer	The minimum value of reading 3, without overrange. Scale integer value by 10^(-register 1435 value) for value in engineering units. *Added in 10v07.0.*
1428–1429	1427–1428	Read	4 byte integer	The maximum value of reading 3, without overrange. Scale integer value by 10^(-register 1435 value) for value in engineering units. *Added in 10v07.0.*
1430–1431	1429–1430	Read	4 byte float	The minimum value of reading 3, without overrange, in engineering units. *Added in 10v07.0.*
1432–1433	1431–1432	Read	4 byte float	The maximum value of reading 3, without overrange, in engineering units. *Added in 10v07.0.*
1434	1433	Read	2 byte integer	Reading 3 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1435	1434	Read	2 byte integer	Reading 3 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 4 Information

1436	1435	Read	2 byte integer	Reading 4 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1437	1436	Read	2 byte integer	Reading 4 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1438–1439	1437–1438	Read	4 byte integer	The minimum value of reading 4, without overrange. Scale integer value by 10^(-register 1447 value) for value in engineering units. *Added in 10v07.0.*
1440–1441	1439–1440	Read	4 byte integer	The maximum value of reading 4, without overrange. Scale integer value by 10^(-register 1447 value) for value in engineering units. *Added in 10v07.0.*
1442–1443	1441–1442	Read	4 byte float	The minimum value of reading 4, without overrange, in engineering units. *Added in 10v07.0.*
1444–1445	1443–1444	Read	4 byte float	The maximum value of reading 4, without overrange, in engineering units. *Added in 10v07.0.*
1446	1445	Read	2 byte integer	Reading 4 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1447	1446	Read	2 byte integer	Reading 4 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 5 Information

1448	1447	Read	2 byte integer	Reading 5 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1449	1448	Read	2 byte integer	Reading 5 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1450–1451	1449–1450	Read	4 byte integer	The minimum value of reading 5, without overrange. Scale integer value by 10^(-register 1459 value) for value in engineering units. *Added in 10v07.0.*
1452–1453	1451–1452	Read	4 byte integer	The maximum value of reading 5, without overrange. Scale integer value by 10^(-register 1459 value) for value in engineering units. *Added in 10v07.0.*
1454–1455	1453–1454	Read	4 byte float	The minimum value of reading 5, without overrange, in engineering units. *Added in 10v07.0.*
1456–1457	1455–1456	Read	4 byte float	The maximum value of reading 5, without overrange, in engineering units. *Added in 10v07.0.*
1458	1457	Read	2 byte integer	Reading 5 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1459	1458	Read	2 byte integer	Reading 5 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 6 Information

1460	1459	Read	2 byte integer	Reading 6 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1461	1460	Read	2 byte integer	Reading 6 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1462–1463	1461–1462	Read	4 byte integer	The minimum value of reading 6, without overrange. Scale integer value by 10^(-register 1471 value) for value in engineering units. *Added in 10v07.0.*
1464–1465	1463–1464	Read	4 byte integer	The maximum value of reading 6, without overrange. Scale integer value by 10^(-register 1471 value) for value in engineering units. *Added in 10v07.0.*
1466–1467	1465–1466	Read	4 byte float	The minimum value of reading 6, without overrange, in engineering units. *Added in 10v07.0.*
1468–1469	1467–1468	Read	4 byte float	The maximum value of reading 6, without overrange, in engineering units. *Added in 10v07.0.*
1470	1469	Read	2 byte integer	Reading 6 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1471	1470	Read	2 byte integer	Reading 6 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 7 Information

1472	1471	Read	2 byte integer	Reading 7 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1735	1472	Read	2 byte integer	Reading 7 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1474–1475	1473–1474	Read	4 byte integer	The minimum value of reading 7, without overrange. Scale integer value by 10^(-register 1483 value) for value in engineering units. *Added in 10v07.0.*
1476–1477	1475–1476	Read	4 byte integer	The maximum value of reading 7, without overrange. Scale integer value by 10^(-register 1483 value) for value in engineering units. *Added in 10v07.0.*
1478–1479	1477–1478	Read	4 byte float	The minimum value of reading 7, without overrange, in engineering units. *Added in 10v07.0.*
1480–1481	1479–1480	Read	4 byte float	The maximum value of reading 7, without overrange, in engineering units. *Added in 10v07.0.*
1482	1481	Read	2 byte integer	Reading 7 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1483	1482	Read	2 byte integer	Reading 7 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 8 Information

1484	1483	Read	2 byte integer	Reading 8 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1485	1484	Read	2 byte integer	Reading 8 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1486–1487	1485–1486	Read	4 byte integer	The minimum value of reading 8, without overrange. Scale integer value by 10^(-register 1495 value) for value in engineering units. *Added in 10v07.0.*
1488–1489	1487–1488	Read	4 byte integer	The maximum value of reading 8, without overrange. Scale integer value by 10^(-register 1495 value) for value in engineering units. *Added in 10v07.0.*
1490–1491	1489–1490	Read	4 byte float	The minimum value of reading 8, without overrange, in engineering units. *Added in 10v07.0.*
1492–1493	1491–1492	Read	4 byte float	The maximum value of reading 8, without overrange, in engineering units. *Added in 10v07.0.*
1494	1493	Read	2 byte integer	Reading 8 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1495	1494	Read	2 byte integer	Reading 8 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 9 Information

1496	1495	Read	2 byte integer	Reading 9 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1497	1496	Read	2 byte integer	Reading 9 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1498–1499	1497–1498	Read	4 byte integer	The minimum value of reading 9, without overrange. Scale integer value by 10^(-register 1507 value) for value in engineering units. *Added in 10v07.0.*
1500–1501	1499–1500	Read	4 byte integer	The maximum value of reading 9, without overrange. Scale integer value by 10^(-register 1507 value) for value in engineering units. *Added in 10v07.0.*
1502–1503	1501–1502	Read	4 byte float	The minimum value of reading 9, without overrange, in engineering units. *Added in 10v07.0.*
1504–1505	1503–1504	Read	4 byte float	The maximum value of reading 9, without overrange, in engineering units. *Added in 10v07.0.*
1506	1505	Read	2 byte integer	Reading 9 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1507	1506	Read	2 byte integer	Reading 9 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 10 Information

1508	1507	Read	2 byte integer	Reading 10 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1509	1508	Read	2 byte integer	Reading 10 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1510–1511	1509–1510	Read	4 byte integer	The minimum value of reading 10, without overrange. Scale integer value by 10^(-register 1519 value) for value in engineering units. *Added in 10v07.0.*
1512–1513	1511–1512	Read	4 byte integer	The maximum value of reading 10, without overrange. Scale integer value by 10^(-register 1519 value) for value in engineering units. *Added in 10v07.0.*
1514–1515	1513–1514	Read	4 byte float	The minimum value of reading 10, without overrange, in engineering units. *Added in 10v07.0.*
1516–1517	1515–1516	Read	4 byte float	The maximum value of reading 10, without overrange, in engineering units. *Added in 10v07.0.*
1518	1517	Read	2 byte integer	Reading 10 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1519	1518	Read	2 byte integer	Reading 10 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 11 Information

1520	1519	Read	2 byte integer	Reading 11 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1521	1520	Read	2 byte integer	Reading 11 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1522–1523	1521–1522	Read	4 byte integer	The minimum value of reading 11, without overrange. Scale integer value by 10^(-register 1531 value) for value in engineering units. *Added in 10v07.0.*
1524–1525	1523–1524	Read	4 byte integer	The maximum value of reading 11, without overrange. Scale integer value by 10^(-register 1531 value) for value in engineering units. *Added in 10v07.0.*
1526–1527	1525–1526	Read	4 byte float	The minimum value of reading 11, without overrange, in engineering units. *Added in 10v07.0.*
1528–1529	1527–1528	Read	4 byte float	The maximum value of reading 11, without overrange, in engineering units. *Added in 10v07.0.*
1530	1529	Read	2 byte integer	Reading 11 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1531	1530	Read	2 byte integer	Reading 11 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 12 Information

1532	1531	Read	2 byte integer	Reading 12 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1533	1532	Read	2 byte integer	Reading 12 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1534–1535	1533–1534	Read	4 byte integer	The minimum value of reading 12, without overrange. Scale integer value by 10^(-register 1543 value) for value in engineering units. *Added in 10v07.0.*
1536–1537	1535–1536	Read	4 byte integer	The maximum value of reading 12, without overrange. Scale integer value by 10^(-register 1543 value) for value in engineering units. *Added in 10v07.0.*
1538–1539	1537–1538	Read	4 byte float	The minimum value of reading 12, without overrange, in engineering units. *Added in 10v07.0.*
1540–1541	1539–1540	Read	4 byte float	The maximum value of reading 12, without overrange, in engineering units. *Added in 10v07.0.*
1542	1541	Read	2 byte integer	Reading 12 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1543	1542	Read	2 byte integer	Reading 12 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 13 Information

1544	1543	Read	2 byte integer	Reading 13 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1545	1544	Read	2 byte integer	Reading 13 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1546–1547	1545–1546	Read	4 byte integer	The minimum value of reading 13, without overrange. Scale integer value by 10^(-register 1555 value) for value in engineering units. *Added in 10v07.0.*
1548–1549	1547–1548	Read	4 byte integer	The maximum value of reading 13, without overrange. Scale integer value by 10^(-register 1555 value) for value in engineering units. *Added in 10v07.0.*
1550–1551	1549–1550	Read	4 byte float	The minimum value of reading 13, without overrange, in engineering units. *Added in 10v07.0.*
1552–1553	1541–1552	Read	4 byte float	The maximum value of reading 13, without overrange, in engineering units. *Added in 10v07.0.*
1554	1553	Read	2 byte integer	Reading 13 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1555	1554	Read	2 byte integer	Reading 13 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 14 Information

1556	1555	Read	2 byte integer	Reading 14 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1557	1556	Read	2 byte integer	Reading 14 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1558–1559	1557–1558	Read	4 byte integer	The minimum value of reading 14, without overrange. Scale integer value by 10^(-register 1567 value) for value in engineering units. *Added in 10v07.0.*
1560–1561	1559–1560	Read	4 byte integer	The maximum value of reading 14, without overrange. Scale integer value by 10^(-register 1567 value) for value in engineering units. *Added in 10v07.0.*
1562–1563	1561–1562	Read	4 byte float	The minimum value of reading 14, without overrange, in engineering units. *Added in 10v07.0.*
1564–1565	1563–1564	Read	4 byte float	The maximum value of reading 14, without overrange, in engineering units. *Added in 10v07.0.*
1566	1565	Read	2 byte integer	Reading 14 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1567	1566	Read	2 byte integer	Reading 14 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 15 Information

1568	1567	Read	2 byte integer	Reading 15 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1569	1568	Read	2 byte integer	Reading 15 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1570–1571	1569–1570	Read	4 byte integer	The minimum value of reading 5, without overrange. Scale integer value by 10^(-register 1579 value) for value in engineering units. *Added in 10v07.0.*
1572–1573	1571–1572	Read	4 byte integer	The maximum value of reading 15, without overrange. Scale integer value by 10^(-register 1579 value) for value in engineering units. *Added in 10v07.0.*
1574–1575	1573–1574	Read	4 byte float	The minimum value of reading 15, without overrange, in engineering units. *Added in 10v07.0.*
1576–1577	1575–1576	Read	4 byte float	The maximum value of reading 15, without overrange, in engineering units. *Added in 10v07.0.*
1578	1577	Read	2 byte integer	Reading 15 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1579	1578	Read	2 byte integer	Reading 15 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 16 Information

1580	1579	Read	2 byte integer	Reading 16 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1581	1580	Read	2 byte integer	Reading 16 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1582–1583	1581–1582	Read	4 byte integer	The minimum value of reading 16, without overrange. Scale integer value by 10^(-register 1591 value) for value in engineering units. *Added in 10v07.0.*
1584–1585	1583–1584	Read	4 byte integer	The maximum value of reading 16, without overrange. Scale integer value by 10^(-register 1591 value) for value in engineering units. *Added in 10v07.0.*
1586–1587	1585–1586	Read	4 byte float	The minimum value of reading 16, without overrange, in engineering units. *Added in 10v07.0.*
1588–1589	1587–1588	Read	4 byte float	The maximum value of reading 16, without overrange, in engineering units. *Added in 10v07.0.*
1590	1589	Read	2 byte integer	Reading 16 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1591	1590	Read	2 byte integer	Reading 16 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 17 Information

1592	1591	Read	2 byte integer	Reading 17 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1593	1592	Read	2 byte integer	Reading 17 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1594–1595	1593–1594	Read	4 byte integer	The minimum value of reading 17, without overrange. Scale integer value by 10^(-register 1603 value) for value in engineering units. *Added in 10v07.0.*
1596–1597	1595–1596	Read	4 byte integer	The maximum value of reading 17, without overrange. Scale integer value by 10^(-register 1603 value) for value in engineering units. *Added in 10v07.0.*
1598–1599	1597–1598	Read	4 byte float	The minimum value of reading 17, without overrange, in engineering units. *Added in 10v07.0.*
1600–1601	1599–1600	Read	4 byte float	The maximum value of reading 17, without overrange, in engineering units. *Added in 10v07.0.*
1602	1601	Read	2 byte integer	Reading 17 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1603	1602	Read	2 byte integer	Reading 17 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 18 Information

1604	1603	Read	2 byte integer	Reading 18 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1605	1604	Read	2 byte integer	Reading 18 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1606–1607	1605–1606	Read	4 byte integer	The minimum value of reading 18, without overrange. Scale integer value by 10^(-register 1615 value) for value in engineering units. *Added in 10v07.0.*
1608–1609	1607–1608	Read	4 byte integer	The maximum value of reading 18, without overrange. Scale integer value by 10^(-register 1615 value) for value in engineering units. *Added in 10v07.0.*
1610–1611	1609–1610	Read	4 byte float	The minimum value of reading 18, without overrange, in engineering units. *Added in 10v07.0.*
1612–1613	1611–1612	Read	4 byte float	The maximum value of reading 18, without overrange, in engineering units. *Added in 10v07.0.*
1614	1613	Read	2 byte integer	Reading 18 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1615	1614	Read	2 byte integer	Reading 18 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 19 Information

1616	1615	Read	2 byte integer	Reading 19 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1617	1616	Read	2 byte integer	Reading 19 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1618–1619	1617–1618	Read	4 byte integer	The minimum value of reading 19, without overrange. Scale integer value by 10^(-register 1627 value) for value in engineering units. *Added in 10v07.0.*
1620–1621	1619–1620	Read	4 byte integer	The maximum value of reading 19, without overrange. Scale integer value by 10^(-register 1627 value) for value in engineering units. *Added in 10v07.0.*
1622–1623	1621–1622	Read	4 byte float	The minimum value of reading 19, without overrange, in engineering units. *Added in 10v07.0.*
1624–1625	1623–1624	Read	4 byte float	The maximum value of reading 19, without overrange, in engineering units. *Added in 10v07.0.*
1626	1625	Read	2 byte integer	Reading 19 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1627	1626	Read	2 byte integer	Reading 19 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Reading 20 Information

1628	1627	Read	2 byte integer	Reading 20 type; one of the statistic values from Appendix A (page 84), or 0 if the reading is not available. *Added in 10v07.0.*
1629	1628	Read	2 byte integer	Reading 20 source; one of the values from Appendix C (page 91), or 0 if the reading is not available. *Added in 10v07.0.*
1630–1631	1629–1630	Read	4 byte integer	The minimum value of reading 20, without overrange. Scale integer value by 10^(-register 1639 value) for value in engineering units. *Added in 10v07.0.*
1632–1633	1631–1632	Read	4 byte float	The minimum value of reading 20, without overrange. Scale integer value by 10^(-register 1639 value) for value in engineering units. *Added in 10v07.0.*
1634–1635	1633–1634	Read	4 byte float	The minimum value of reading 20, without overrange, in engineering units. *Added in 10v07.0.*
1636–1637	1635–1636	Read	4 byte integer	The maximum value of reading 20, without overrange, in engineering units. *Added in 10v07.0.*
1638	1637	Read	2 byte integer	Reading 20 units; one of the values from Appendix B (page 86), or 0 if the reading is not available. *Added in 10v07.0.*
1639	1638	Read	2 byte integer	Reading 20 decimal places; the number of digits after the implicit decimal place in the current value, minimum value, and maximum value. *Added in 10v07.0.*

Command Registers

See the **Using Commands** section (page 38) for more information.

Register Number	Register Address	Action	Data Type	Description
1000	999	Write	2 byte integer	Requested limited command ID. *Added in 6v17.0.*
		Read		Last executed command ID. *Added in 6v17.0.*
1001	1000	Write	2 byte integer	Requested limited command argument. *Added in 6v17.0.*
		Read		Last executed command result. *Added in 6v17.0.*
1002–1003	1001–1002	Write	4 byte integer	Requested command ID. *Added in 10v07.0.*
		Read		ID of currently executing or last executed command. *Added in 10v07.0.*
1004–1005	1003–1004	Write	4 byte integer	Requested command argument. *Added in 10v07.0.*
		Read		Argument for currently executing or last executed command. *Added in 10v07.0.*
1006–1007	1005–1006	Read	4 byte integer	Status of currently executing or last executed command. *Added in 10v07.0.*
1008–1009	1007–1008	Read	4 byte integer	Return value of the last executed command. *Added in 10v07.0.*
1050	1049	Write	2 byte integer	COMPOSER Mixture gas 1 gas number. *Added in 6v17.0.*
1051	1050	Write	2 byte integer	COMPOSER Mixture gas 1 percentage, 1 = 0.01%, 10000 = 100%. *Added in 6v17.0.*
1052	1051	Write	2 byte integer	COMPOSER Mixture gas 2 gas number. *Added in 6v17.0.*
1053	1052	Write	2 byte integer	COMPOSER Mixture gas 2 percentage, 1 = 0.01%, 10000 = 100%. *Added in 6v17.0.*
1054	1053	Write	2 byte integer	COMPOSER Mixture gas 3 gas number. *Added in 6v17.0.*
1055	1054	Write	2 byte integer	COMPOSER Mixture gas 3 percentage, 1 = 0.01%, 10000 = 100%. *Added in 6v17.0.*
1056	1055	Write	2 byte integer	COMPOSER Mixture gas 4 gas number. *Added in 6v17.0.*
1057	1056	Write	2 byte integer	COMPOSER Mixture gas 4 percentage, 1 = 0.01%, 10000 = 100%. *Added in 6v17.0.*
1058	1057	Write	2 byte integer	COMPOSER Mixture gas 5 gas number. *Added in 6v17.0.*
1059	1058	Write	2 byte integer	COMPOSER Mixture gas 5 percentage, 1 = 0.01%, 10000 = 100%. *Added in 6v17.0.*

Device Information Registers

1086–7	1085–6	Read / Write	4 bytes	User test value. See the discussion of multi register formatting (page 16) for best usage. *Added in 10v19.0.*
1088–9	1087–8	Read	4 bytes	Reads as 1067320907 for a four-byte integer or 1.234567 as a single precision floating point value; see the discussion of multi register formatting (page 16) for best usage. *Added in 10v19.0.*
1090	1089	Read	2 byte integer	Major version number. *Added in 10v07.0.*
1091	1090	Read	2 byte integer	Minor version number. *Added in 10v07.0.*
1092	1091	Read	2 byte integer	Custom version number. *Added in 10v07.0.*
1093	1092	Read	2 byte integer	Internal version number. *Added in 10v07.0.*

1094–5	1093–4	Read	4 byte integer	Serial number. *Added in 10v19.0.*
1096	1095	Read	2 byte integer	Manufacture month (most significant byte, month = value div 256) and day (least significant byte, day = value mod 256). *Added in 10v19.0.*
1097	1096	Read	2 byte integer	Manufacture year. *Added in 10v19.0.*
1098	1097	Read	2 byte integer	Last calibration month (most significant byte, month = value div 256) and day (least significant byte, day = value mod 256). *Added in 10v19.0.*
1099	1098	Read	2 byte integer	Last calibration year. *Added in 10v19.0.*

Legacy Registers

While these Legacy registers are still supported in Modbus RTU, they have all been replaced by other registers that often have more functionality. These registers were never available in Modbus TCP/IP.

Register Number	Register Address	Action	Data Type	Description
21	20	Read / Write	2 byte integer	Closed loop proportional gain.
22	21	Read / Write	2 byte integer	Closed loop derivative gain.
23	22	Read / Write	2 byte integer	Closed loop integral gain.
24	23	Read / Write	2 byte integer	Setpoint, in setpoint counts.
46	45	Read / Write	2 byte integer	Gas number.
65	64	Read / Write	2 byte integer	Slave address.
2041–2042	2040–2041	Read	4 byte float	Current pressure reading in engineering units. This is the current primary pressure value.
2043–2044	2042–2043	Read	4 byte float	Current temperature reading in engineering units.
2045–2046	2044–2045	Read	4 byte float	Current volumetric flow reading in engineering units.
2047–2048	2046–2047	Read	4 byte float	Current mass flow reading in engineering units.
2049–2050	2048–2049	Read	4 byte float	Current setpoint in engineering units.
2051–2052	2050–2051	Read	4 byte float	Current totalizer 1 reading in engineering units.
2053	2052	Read	2 byte integer	Slave address.
2054	2053	Read	2 byte unsigned integer	Gas number.
2055	2054	Read	2 byte integer	Volumetric flow overrange indicator is active. 0 = inactive, nonzero = active.
2056	2055	Read	2 byte integer	Mass flow overrange indicator is active. 0 = inactive, nonzero = active.
2057	2056	Read	2 byte integer	Pressure overrange indicator is active. 0 = inactive, nonzero = active.
2058	2057	Read	2 byte integer	Temperature overrange indicator is active. 0 = inactive, nonzero = active.
2059	2058	Read	2 byte integer	Totalizer overrange indicator is active. 0 = inactive, nonzero = active.

Commands

Using Commands

Commands can be sent using an older method, which is supported by Modbus version 6v17 or later, or using an improved method that is only available in firmware versions 10v00 or later. Both are described below. For either technique, it is recommended to send a **`No Operation`** (commands Id 0) between each command to ensure that a repeated command is never missed. The command ID and/or the command argument must change for the new command to be executed; writing the values that are already in the registers will not execute a new command.

Sending Modbus RTU Commands

Sending a limited command

This procedure is supported by all firmware versions but only supports command IDs < 65535.

1. Send a **`Write Multiple Registers`** message to set register address 999 to the desired command ID and address 1000 to the desired command argument. Note that both values must be in the range of 0–65535. There will be no response until the command has completed. Most commands complete essentially instantly, but some may take a significant amount of time (for example, the Tare command).
2. Send a **`Read Input Registers`** message to read register address 1000. If the returned value is larger than 32768, the command returns an error code. The Status Value table (page 39) shows the meaning for each code. If the value is 32767 or smaller, the value is the return value from the command (many commands only return 0).

Sending any command

This procedure is supported by 10v07.0 or later firmware versions. Any command ID is supported.

Note that all registers here are paired to support four bytes of data. Write the two most significant bytes into the register with the lower address and the two least significant bytes into the register with the higher address. Values that are read back are formatted similarly.

1. Send a **`Write Multiple Registers`** message to set register addresses 1001–1002 to the desired command ID and addresses 1003–1004 to the desired command argument. The command ID is always an unsigned integer. Many commands take an argument that is in an integer format, but a significant number take a single precision floating point value as the argument. See the documentation of the command for the expected format. The instrument will respond as soon as possible, not waiting for the command to complete.
2. Send a **`Read Input Registers`** message to read register addresses 1001–1008. Registers 1001–1004 contain the currently executing or last executed command ID and argument; if they do not match the command requested above, the request has been queued and will be executed as soon as possible. Registers 1005–1006 contain the status of the command. While most commands execute essentially instantly, a 1 in these registers indicates that the command is still in progress. If registers 1001–1004 do not match the requested command ID and argument, or if registers 1005–1006 contain a status of 1, the request command is not yet complete and another Read Input Registers message should be issued to read register addresses 1001–1008. If the status in registers 1005–1006 is nonzero, an error occurred; see the Status Value table (page 39) for the meaning of the error code. If the status is 0, the return value in registers 1007–1008 is valid and will be appropriate for the command; see the command documentation for the format and meaning of this value.

Sending Modbus TCP/IP Commands

Sending a limited command

This procedure is supported by all firmware versions but only supports command IDs < 65535.

1. Send a **`Write Multiple Registers`** message to set register address 999 to the desired command ID and address 1000 to the desired command argument. Note that both values must be in the range of 0–65535. There will be no response until the command has completed.
2. Send a **`Read Input Registers`** message to read register addresses 999–1000. Register 999 contains the currently executing or last executed command ID; if it does not match the command requested above, the request has been queued and will be executed as soon as possible. If register 999 does not match the requested command ID, the request command is not yet complete and another Read Input Registers message should be issued to read register addresses 999–1000. If the status in register 1000 is larger than 32768, an error occurred; see the Status Value table (page 39) for the meaning of the error code. Otherwise, it holds the return value of the command; see the command documentation for the format and meaning of this value.

Sending any command

This procedure is supported by 10v07.0 or later firmware versions. Any command ID is supported.

Note that all registers here are paired to support four bytes of data. Write the two most significant bytes into the register with the lower address and the two least significant bytes into the register with the higher address. Values that are read back are formatted similarly.

1. Send a **`Write Multiple Registers`** message to set register addresses 1001–1002 to the desired command ID and addresses 1003–1004 to the desired command argument. The command ID is always an unsigned integer. Many commands take an argument that is in an integer format, but a significant number take a single precision floating point value as the argument. See the documentation of the command for the expected format. The instrument will respond as soon as possible, not waiting for the command to complete.
2. Send a **`Read Input Registers`** message to read register addresses 1001–1008. Registers 1001–1004 contain the currently executing or last executed command ID and argument; if they do not match the command requested above, the request has been queued and will be executed as soon as possible. Registers 1005–1006 contain the status of the command. While most commands execute essentially instantly, a 1 in these registers indicates that the command is still in progress. If registers 1001–1004 do not match the requested command ID and argument, or if registers 1005–1006 contain a status of 1, the request command is not yet complete and another Read Input Registers message should be issued to read register addresses 1001–1008. If the status in registers 1005–1006 is nonzero, an error occurred; see the Status Value table (page 39) for the meaning of the error code. If the status is 0, the return value in registers 1007–1008 is valid and will be appropriate for the command; see the command documentation for the format and meaning of this value.

Status Value Table

The status may have these values:

Status Value	Status Name	Description	Error Code*
0	SUCCESS	The last command completed successfully.	N/A
1	IN_PROGRESS	A command is currently executing.	N/A
2	INVALID_ID	The ID of the last command is invalid.	32769
3	INVALID_ARGUMENT	The argument of the last command is invalid.	32770
4	UNSUPPORTED	The last command is not supported by the instrument.	32771

5	INVALID_MIX_IDX	The requested gas mix number is invalid.	32772
6	INVALID_MIX_GAS	A gas used in the mix does not exist on the instrument.	32773
7	INVALID_MIX_PCT	The gas mix fractions do not sum to 100%.	32774

* For use with basic / limited commands and found in register address 1000.

Command Flow

No Operation

Command ID: 0

This command does nothing. It is required to separate identical command IDs and arguments. This command may be used before every command or at any time.

Argument: This command ignores any argument.

Data Readings

Use registers 1640–1771 (standard readings, page 17) or 1400–1637 (optimized readings, page 24) to obtain information about a reading type, source, min/max values, units, decimal places for scaling integer values, etc.

Query Reading Type

Firmware: 10v07.0

Command ID: 28

Returns the type of the reading, which will be a statistic value from Appendix A (page 84).

Argument: Use a value from the table below to select the reading type:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading

1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the type is available in the return value.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested *which* argument.

Query Reading Source

Firmware: 10v07.0

Command ID: 27

Returns the source of the reading, which will be one of the values from Appendix A (page **84**).

Argument: Use a value from the table below to select the reading source:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading

1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the source is available in the return value.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested *which* argument.

Query Reading Minimum (Integer)

Firmware: 10v07.0

Command ID: 65538

Returns the minimum value of the reading (not including overrange), as an integer. To obtain the reading in the requested engineering units, multiply the value by 10^(- reading decimal places).

Argument: Use a value from the table below to select the reading type:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading

8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the minimum is available in the return value.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested *which* argument.

Query Reading Minimum (Float)

Firmware: 10v07.0

Command ID: 65536

Returns the minimum value of the reading (not including overrange), in a floating-point format.

Argument: Use a value from the table below to select the reading type:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading

5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the minimum is available in the return value.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested *which* argument.

Query Reading Maximum (Integer)

Firmware: 10v07.0

Command ID: 65539

Returns the maximum value of the reading (not including overrange). To obtain the reading in the requested engineering units, multiply the value by 10^(- reading decimal places).

Argument: Use a value from the table below to select the reading type:

Value	Description
0	Setpoint reading
1	Valve drive reading

2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the maximum is available in the return value.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested *which* argument.

Query Reading Maximum (Float)

Firmware: 10v07.0

Command ID: 65537

Returns the maximum value of the reading (not including overrange), in a floating-point format.

Argument: Use a value from the table below to select the reading type:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the maximum is available in the return value.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested *which* argument.

Query Reading Engineering Units

Firmware: 10v07.0

Command ID: 29

Returns the engineering units of the reading, a value from Appendix B (page **86**).

Argument: Use a value from the table below to select the engineering units to read:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading
1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the units are available in the return value.

An UNSUPPORTED status is set if the offset in the command ID is not associated with the instrument.

An INVALID_ARGUMENT status is set if the instrument has no reading associated with the requested argument.

Set Reading Engineering Units

Firmware: 10v07.0

Command ID: 65300 + offset

Changes the engineering units of the desired reading. When writing the command ID, add the offset of the reading to the command ID (65300).

Offset	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
100	First optimized reading
101	Second optimized reading
102	Third optimized reading
103	Fourth optimized reading
104	Fifth optimized reading
105	Sixth optimized reading
106	Seventh optimized reading
107	Eighth optimized reading
108	Ninth optimized reading
109	Tenth optimized reading
110	Eleventh optimized reading
111	Twelfth optimized reading
112	Thirteenth optimized reading
113	Fourteenth optimized reading
114	Fifteenth optimized reading
115	Sixteenth optimized reading
116	Seventeenth optimized reading
117	Eighteenth optimized reading

118	Nineteenth optimized reading
119	Twentieth optimized reading

Not all readings are independent of each other. Changing the engineering units of the pressure reading also changes the secondary pressure and barometric pressure.

Argument: Use the value of the associated desired engineering units found in **Appendix B** (page 86).

Command response: A SUCCESS status is set if the reading's engineering units are changed. The return value is the value of the requested engineering units.

An UNSUPPORTED status is set if the offset in the command ID is not associated with the instrument.

An INVALID_ARGUMENT status is set if the engineering units are not valid for the reading.

Query Reading Decimal Places

Firmware: 10v07.0

Command ID: 30

Returns the number of digits after the implicit decimal place in the reading current value, minimum value, and maximum value.

Argument: Use a value from the table below to select the engineering units to read:

Value	Description
0	Setpoint reading
1	Valve drive reading
2	Pressure reading
3	Secondary pressure reading
4	Barometric pressure reading
5	Temperature reading
6	Volumetric flow reading
7	Mass flow reading
8	Totalizer 1 reading
9	Totalizer 2 reading
10	Humidity reading
1000	First optimized reading
1001	Second optimized reading
1002	Third optimized reading
1003	Fourth optimized reading
1004	Fifth optimized reading
1005	Sixth optimized reading
1006	Seventh optimized reading
1007	Eighth optimized reading
1008	Ninth optimized reading
1009	Tenth optimized reading

1010	Eleventh optimized reading
1011	Twelfth optimized reading
1012	Thirteenth optimized reading
1013	Fourteenth optimized reading
1014	Fifteenth optimized reading
1015	Sixteenth optimized reading
1016	Seventeenth optimized reading
1017	Eighteenth optimized reading
1018	Nineteenth optimized reading
1019	Twentieth optimized reading

Command response: A SUCCESS status is set if the number of decimal places is available in the return value

An INVALID_ARGUMENT status is set if device has no reading associated with the requested *which* argument.

Totalizer 1 Flow

Firmware: 10v19.0

Command ID: 36

Queries or sets the flow accumulated by Totalizer 1.

Argument: Use a value from the following table to query or set the Totalizer 1 flow accumulated value.

Value	Description
0	Query; do not change the flow being accumulated.
1	Disable this totalizer.
4	Accumulate volumetric flow.
5	Accumulate mass flow.

Command Response: A SUCCESS status is set, and Totalizer 1 will accumulate the requested flow.

An INVALID_ARGUMENT status is set if the flow is not a valid value.

Totalizer 1 Flow Accumulation Mode

Firmware: 10v19.0

Command ID: 34

Queries or sets the flow accumulation mode for Totalizer 1.

Argument: Use a value from the following table to query or set the desired flow accumulation mode:

Value	Description
-1	Query; do not change the current mode.
0	Add positive flow but ignore negative flow.
1	Subtract negative flow but ignore positive flow.
2	Add positive flow and subtract negative flow.
3	Add positive flow. Once flow ceases, reset the volume when flow begins again.

Command Response: A SUCCESS status is set, and Totalizer 1 will accumulate the requested flow.

An INVALID_ARGUMENT status is set if the mode is not a valid value.

Totalizer 1 Accumulation Action When Limit Reached

Firmware: 10v19.0

Command ID: 38

Queries or sets the action taken when Totalizer 1 reaches the maximum value it can accumulate.

Argument: Use the value from the following table to select the action when Totalizer 1 is at its maximum value.

Value	Description
-1	Query; do not change the current mode.
0	Add positive flow but ignore negative flow.
1	Subtract negative flow but ignore positive flow.
2	Add positive flow and subtract negative flow.
3	Add positive flow. Once flow ceases, reset the volume when flow begins again.

Command Response: A SUCCESS status is set, and Totalizer 1 will perform the requested action when the totalizer limit is reached.

An INVALID_ARGUMENT status is set if the limit is not a valid value.

Totalizer 2 Flow

Firmware: 10v19.0

Command ID: 37

Queries or sets the flow accumulated by Totalizer 2.

Argument: Use a value from the following table to query or set the Totalizer 2 flow accumulated value.

Command Response: A SUCCESS status is set, and Totalizer 1 will accumulate the requested flow.

Value	Description
0	Query; do not change the flow being accumulated.
1	Disable this totalizer.
4	Accumulate volumetric flow.
5	Accumulate mass flow.

Command Response: A SUCCESS status is set, and Totalizer 2 will accumulate the requested flow.

An INVALID_ARGUMENT status is set if the flow is not a valid value.

Totalizer 2 Flow Accumulation Mode

Firmware: 10v19.0

Command ID: 35

Queries or sets the flow accumulation mode for Totalizer 2.

Argument: Use a value from the following table to query or set the desired flow accumulation mode:

Command Response: Totalizer 2 will accumulate using the requested mode.

Value	Description
-1	Query; do not change the current mode.
0	Add positive flow but ignore negative flow.
1	Subtract negative flow but ignore positive flow.
2	Add positive flow and subtract negative flow.
3	Add positive flow. Once flow ceases, reset the volume when flow begins again.

Command Response: A SUCCESS status is set, and Totalizer 2 will accumulate the requested flow.

An INVALID_ARGUMENT status is set if the mode is not a valid value.

Totalizer 2 Accumulation Action When Limit Reached

Firmware: 10v19.0

Command ID: 39

Queries or sets the action taken when Totalizer 2 reaches the maximum value it can accumulate.

Argument: Use a value from the following table to select the action when Totalizer 2 is at its maximum value.

Value	Description
-1	Query; do not change the current mode.
0	Add positive flow but ignore negative flow.
1	Subtract negative flow but ignore positive flow.
2	Add positive flow and subtract negative flow.
3	Add positive flow. Once flow ceases, reset the volume when flow begins again.

Command Response: A SUCCESS status is set and Totalizer 2 will perform the requested action when the totalizer limit is reached.

An INVALID_ARGUMENT status is set if the limit is not a valid value.

Standard Flow Reference Pressure (Integer)

Firmware: 10v19.0

Command ID: 65560

Queries or sets the reference pressure for standard flow (flow units that begin with `S`).

Argument: Use 0 or a negative value to query for the pressure used as the reference for standard flow. Use a positive integer to set the reference pressure value, in the same units as the primary pressure; multiply the engineering value by 10^(pressure decimal places) to provide the integer value.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is negative or too large.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Normal Flow Reference Pressure (Integer)

Firmware: 10v19.0

Command ID: 65561

Queries or sets the reference pressure for normal flow (flow units that begin with `N`).

Argument: Use 0 or a negative value to query for the pressure used as the reference for normal flow. Use a positive integer to set the reference pressure value, in the same units as the primary pressure; multiply the engineering value by 10^(pressure decimal places) to provide the integer value.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is negative or too large.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Standard Flow Reference Pressure (Float)

Firmware: 10v19.0

Command ID: 65556

Queries or sets the reference pressure for standard flow (flow units that begin with `S`).

Argument: Use 0 or a negative value to query for the pressure used as the reference for standard flow. Use a positive IEEE-754 single precision floating point value to set the reference pressure, in the same units as the primary pressure.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is not a valid pressure.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Normal Flow Reference Pressure (Float)

Firmware: 10v19.0

Command ID: 65557

Queries or sets the reference pressure for normal flow (flow units that begin with `N`).

Argument: Use 0 or a negative value to query for the pressure used as the reference for normal flow. Use a positive IEEE-754 single precision floating point value to set the reference pressure, in the same units as the primary pressure.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is not a valid pressure.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Standard Flow Reference Temperature (Integer)

Firmware: 10v19.0

Command ID: 65562

Queries or sets the reference temperature for standard flow (flow units that begin with `S`).

Argument: To query the temperature used as the reference for standard flow, use a value that evaluates to -500 or smaller (once the value is scaled to the temperature engineering units). Use a positive integer to set the reference temperature value, in the same units as the temperature; multiply the engineering value by 10^(temperature decimal places) to provide the integer value.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is not a valid temperature.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Normal Flow Reference Temperature (Integer)

Firmware: 10v19.0

Command ID: 65563

Queries or sets the reference temperature for normal flow (flow units that begin with `N`).

Argument: To query the temperature used as the reference for normal flow, use a value that evaluates to -500 or smaller (once the value is scaled to the temperature engineering units). Use a positive integer to set the reference temperature value, in the same units as the temperature; multiply the engineering value by 10^(temperature decimal places) to provide the integer value.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is not a valid temperature.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Standard Flow Reference Temperature (Float)

Firmware: 10v19.0

Command ID: 65558

Queries or sets the reference pressure for standard flow (flow units that begin with `S`).

Argument: To query the temperature used as the reference for standard flow, use a value that evaluates to -500 or smaller. Use a positive IEEE-754 single precision floating point value to set the reference temperature value, in the same units as the temperature.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is not a valid temperature.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Normal Flow Reference Temperature (Float)

Firmware: 10v19.0

Command ID: 65559

Queries or sets the reference temperature for normal flow (flow units that begin with `N`).

Argument: To query the temperature used as the reference for normal flow, use a value that evaluates to -500 or smaller. Use a positive IEEE-754 single precision floating point value to set the reference temperature value, in the same units as the temperature.

Command Response: The reference is updated and available in the return value.

An INVALID_ARGUMENT status is set if the reference is not a valid temperature.

An UNSUPPORTED status is set if the instrument does not support setting the reference.

Flow Averaging Time Constant

Firmware: 10v19.0

Command ID: 50

Queries or sets the averaging time constant for flow readings.

Argument: Use `65535` to query the current value. Otherwise, set the desired averaging time constant as an integer in milliseconds.

Command Response: The time constant is set as requested and is in the return value.

An INVALID_ARGUMENT status is set if the time constant is negative or larger than 65535.

Pressure Averaging Time Constant

Firmware: 10v19.0

Command ID: 47

Queries or sets the averaging time constant for all pressure readings.

Argument: Use 65535 to query the current value. Otherwise, set the desired averaging time constant as an integer in milliseconds.

Command Response: The time constant is set as requested and is in the return value.

An INVALID_ARGUMENT status is set if the time constant is negative or larger than 65535.

Primary Pressure Averaging Time Constant

Firmware: 10v19.0

Command ID: 48

Queries or sets the averaging time constant for the primary pressure sensor.

Argument: Use 65535 to query the current value. Otherwise, set the desired averaging time constant as an integer in milliseconds.

Command Response: The time constant is set as requested and is in the return value.

An INVALID_ARGUMENT status is set if the time constant is negative or larger than 65535.

Secondary Pressure Averaging Time Constant

Firmware: 10v19.0

Command ID: 49

Queries or sets the averaging time constant for the secondary pressure sensor.

Argument: Use 65535 to query the current value. Otherwise, set the desired averaging time constant as an integer in milliseconds.

Command Response: The time constant is set as requested and is in the return value.

An INVALID_ARGUMENT status is set if the time constant is negative or larger than 65535.

Reading Zero Band

Firmware: 10v19.0

Command ID: 44

Queries or sets the value at which readings are reported as zero. This value applies for all pressure and flow readings reported by the instrument; any value closer to zero than the zero band (positive or negative) will be reported as zero. The zero band applies to values transmitted digitally and shown on the display.

Argument: Use 65535 to query the current value. Otherwise, set the desired zero band, with one count being 0.01% of full scale of the reading. 1 count = no zero band, 638 counts = the maximum zero band of 6.38% of reading full scale.

Command Response: The zero band is set as requested and is in the return value.

An INVALID_ARGUMENT status is set if the reading band is not a valid value.

Control

The control commands are commands for Alicat controllers only. These commands manage the setpoint, valve, and totalizer batches.

Autotare Enabled

▶ Controllers

Firmware: 10v19.0

Command ID: 40

Queries, sets, or disables autotare.

Argument: Use the mode from the following table to select the autotare action.

Mode	Description
0	Query the current value.
1	Disable autotare.
2	Enable autotare.

Command Response: A SUCCESS status is set when Autotare is enabled or disabled if requested, and the current enable state is in the return value.

An INVALID_ARGUMENT status is set if the mode is not a valid value.

An UNSUPPORTED status is set if the instrument is not a controller.

Autotare Delay

▶ Controllers

Firmware: 10v19.0

Command ID: 45

Queries or sets the delay between the setpoint being commanded to zero and the start of an autotare.

Argument: Use the mode from the following table to select the autotare action.

Value	Description
0	Query the current delay.
1–255	Set the delay, where each count is 0.1 second. For example, 1 count = 0.1 sec, 255 counts = 25.5 sec

Command Response: A SUCCESS status is set when the autotare delay has been set as requested, and the current delay is in the return value.

An INVALID_ARGUMENT status is set if the delay is not a valid value.

An UNSUPPORTED status is set if the instrument is not a controller.

Query Setpoint Limit (Integer)

▶ Controllers

Firmware: 10v19.0

Command ID: 65548

Queries the minimum or maximum setpoint limits, as an integer.

Argument: Use `0` to query the minimum setpoint limit, or `1` to query the maximum setpoint limit.

Command Response: A SUCCESS status is set when the requested limit is available in the return value.

An INVALID_ARGUMENT status is set if the instrument is not a controller.

Set Minimum Setpoint Limit (Integer)

▶ Controllers

Firmware: 10v19.0

Command ID: 65550

Sets the minimum setpoint limit, as an integer.

Argument: Enter the desired limit as an integer in setpoint engineering units (see Appendix B on page 86); multiply the engineering value by 10^(setpoint decimal places) to provide the integer value.

Command Response: A SUCCESS status is set when the limit is updated and available in the return value.

An INVALID_ARGUMENT status is set if the requested limit is too large or too small.

An UNSUPPORTED status is set if the instrument is not a controller.

Set Maximum Setpoint Limit (Integer)

▶ Controllers

Firmware: 10v19.0

Command ID: 65552

Sets the maximum setpoint limit, as an integer.

Argument: Enter the desired limit as an integer in setpoint engineering units (see Appendix B on page 86); multiply the engineering value by 10^(setpoint decimal places) to provide the integer value.

Command Response: A SUCCESS status is set when the limit is updated and available in the return value.

An INVALID_ARGUMENT status is set if the requested limit is too large or too small.

An UNSUPPORTED status is set if the instrument is not a controller.

Query Setpoint Limit (Float)

▶ Controllers

Firmware: 10v19.0

Command ID: 65549

Queries the minimum or maximum setpoint limits, in a floating-point format.

Argument: Use `0` to query the minimum setpoint limit, or `1` to query the maximum setpoint limit.

Command Response: A SUCCESS status is set when the requested limit is available in the return value.

An INVALID_ARGUMENT status is set if the instrument is not a controller.

Set Minimum Setpoint Limit (Float)

▶ Controllers

Firmware: 10v19.0

Command ID: 65551

Sets the minimum setpoint limit, in a floating-point format.

Argument: Enter the desired limit as an integer in setpoint engineering units (see Appendix B on page 86); multiply the engineering value by 10^(setpoint decimal places) to provide the integer value.

Command Response: A SUCCESS status is set when the limit is updated and available in the return value.

An INVALID_ARGUMENT status is set if the requested limit is too large or too small.

An UNSUPPORTED status is set if the instrument is not a controller.

Set Maximum Setpoint Limit (Float)

▶ Controllers

Firmware: 10v19.0

Command ID: 65553

Sets the maximum setpoint limit, in a floating-point format.

Argument: Enter the desired limit as an integer in setpoint engineering units (see Appendix B on page 86); multiply the engineering value by 10^(setpoint decimal places) to provide the integer value.

Command Response: A SUCCESS status is set when the limit is updated and available in the return value.

An INVALID_ARGUMENT status is set if the requested limit is too large or too small.

An UNSUPPORTED status is set if the instrument is not a controller.

Set Power-up Setpoint

▶ Controllers

Firmware: 7v05.0, also in 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6, 7v00.8, 7v00.9, 7v00.10, 7v00.11, 7v00.12, 7v00.15.

Command ID: 12

Set power-up setpoint stores the current setpoint to be immediately used the next time the controller is powered.

Argument: this command ignores the argument value.

Command Response: A SUCCESS status is set, and the instrument saves the current setpoint as the power-up setpoint.

Setpoint Maximum Ramp (Saved)

▶ Controllers

Firmware: 10v07.0

Command ID: 65546

Sets the maximum ramp rate of the setpoint and saves it across power cycles. This command should not be used more often than every few minutes.

Argument: To query the current maximum ramp rate, use a negative value.

To disable the maximum ramp rate, use a value of 0.

To set a maximum ramp rate, determine the desired rate of full-scale percentage change per millisecond and then multiply that value by 10,000,000. For example, if a controller were to ramp to a setpoint by 1% of full scale every second, the instrument would ramp 0.001% every millisecond. Multiply that 0.001% by 10,000,000 to obtain a value of 10,000. Refer to the following table for further example values.

Maximum Ramp Rate	Value
100% of full scale every millisecond	1000000000
1% of full scale every millisecond	10000000
100% of full scale every second	1000000
100% of full scale every minute	16667
1% of full scale every second	10000
100% of full scale every hour	278
1% of full scale every minute	167
10% of full scale every hour	28

Command Response: A SUCCESS status is set after the command is completed. The return value is the current maximum ramp rate. Multiply the value by 0.0000001 to obtain the ramp rate in percent of full scale per millisecond.

An UNSUPPORTED status is set if the instrument does not support setpoint ramping.

Setpoint Maximum Ramp (Temporary)

▶ Controllers

Firmware: 10v07.0

Command ID: 65547

Sets the maximum ramp rate of the setpoint. This command does not save the maximum ramp rate and it is lost when the instrument loses power.

Argument: To query the current ramp rate, use a negative value.

To disable the ramp rate, use a value of 0.

To set a maximum ramp rate, determine the desired rate of full-scale percentage change per millisecond and then multiply that value by 10,000,000. For example, if a controller were to ramp to a setpoint by 1% of full scale every second, the instrument would ramp 0.001% every millisecond. Multiply that 0.001% by 10,000,000 to obtain a value of 10,000. Refer to the following table for further example values.

Maximum Ramp rate	Value
100% of full scale every millisecond	1000000000
1% of full scale every millisecond	10000000
100% of full scale every second	1000000
100% of full scale every minute	16667
1% of full scale every second	10000
100% of full scale every hour	278
1% of full scale every minute	167
10% of full scale every hour	28

Command Response: A SUCCESS status is set after the command is completed. The return value is the current maximum ramp rate. Multiply the value by 0.0000001 to obtain the ramp rate in percent of full scale per millisecond.

An UNSUPPORTED status is set if the instrument does not support setpoint ramping.

Setpoint Ramp Enabled

▶ Controllers

Firmware: 10v19.0

Command ID: 42

Queries or sets the conditions when the maximum setpoint ramp is enabled.

Argument: Use 65535 to query the current enable state. Use a bit field from the table below to change the state:

Bit	Description
0	Set if the setpoint jumps to the initial value on power up; clear if the setpoint ramps to the initial value from 0.
1	Set if the setpoint jumps when the setpoint is commanded to 0; clear if the setpoint ramps to 0.
2	Set if the setpoint jumps for a setpoint that increases; clear if the setpoint ramps to the higher setpoint.
3	Set if the setpoint jumps for a setpoint that decreases; clear if the setpoint ramps to the lower setpoint.

Command Response: A SUCCESS status is set when the setpoint ramp enables have been set as requested, and the current enables are in the return value.

An INVALID_ARGUMENT status is set if the mode is not valid.

An UNSUPPORTED status is set if the instrument is not a controller or setpoint ramping is not supported.

Hold Valve(s)

▶ Controllers

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 6

Pauses the controller valve(s) and stops any further control of the process. The command can also cancel a current hold.

Argument: Use the value from the following table for the desired action:

Value	Description
0	Cancel valve hold and resume normal closed-loop control.
1	Hold all valves closed.
2	Hold valves at their current positions.
3	Exhaust: Close the upstream valve and fully open the downstream valve. Only supported on dual valve controllers.

Command Response: A SUCCESS status is set if the command sets the valve to the desired argument mode.

An UNSUPPORTED status returns if the requested mode is not supported by the instrument.

An INVALID_ARGUMENT status is set if the argument value is not valid.

Set Active Valve

▶ Controllers

Firmware: 7v16.0, also in 7v15.3.

Command ID: 15

Only available on MCT stream-switching controllers. The command controls which valve is the active valve that controls the process.

Argument: Use the value for the desired valve found in the following table.

Value	Description
0	Upstream valve or only valve
1	Downstream valve or auxiliary valve wired in the downstream location.
2	The first auxiliary valve. *Supported from 10v11.0.*
3	The second auxiliary valve. *Supported from 10v11.0.*
65535	Query the currently active valve. *Supported from 10v11.0.*

Command Response: A SUCCESS status is set if the active valve changes. The return value is the currently active control valve (0 to 3), if the argument is 65535, or 0 otherwise.

An UNSUPPORTED status is set if the instrument is not an MCT controller.

An INVALID_ARGUMENT status is set if the value used is not valid.

Loop Control Variable

▶ Controllers

Firmware: 10v19.0

Command ID: 43

Changes the process variable that drives closed loop control.

Argument: Use 65535 to query the current enable state. Use a value from the table below to choose the process variable to control:

Value	Description
0	Mass Flow
1	Volumetric Flow
2	Differential Pressure
3	Absolute Pressure
4	Gauge Pressure
other	Any setpoint value in Appendix A (page 84)

Command Response: A SUCCESS status if the loop control variable has been updated and the current variable is in the return value.

An INVALID_ARGUMENT status is set if the requested control variable is not applicable to this instrument.

An UNSUPPORTED status is set if the instrument is not a controller.

Set Loop Control Variable

▶ Controllers

Firmware: 7v05.0, also in 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6, 7v00.8, 7v00.9, 7v00.10, 7v00.11, 7v00.12, 7v00.15.

Command ID: 11

Changes the statistic that the controller actively controls. That means a mass flow controller can be changed to control pressure or volumetric flow if needed.

Argument: Use the value for the desired statistic found in the following table.

Value	Description
0	Mass flow
1	Volumetric flow
2	Differential pressure
3	Absolute pressure
4	Gauge pressure
...	Any setpoint value that is found in Appendix A (page 84). *Added in 8v24.0.*

Command Response: A SUCCESS status is set if the loop control variable changes.

An INVALID_ARGUMENT is set if the statistic is not available on the instrument (e.g., trying to control mass flow on a pressure controller).

Set Loop Control Algorithm

▶ Controllers

Firmware: 7v08.0, also in 7v07.1.

Command ID: 13

Changes the active control algorithm.

Argument: Use the value of the desired algorithm found in the following table.

Value	Description
1	PDF closed-loop control algorithm
2	PD^2I closed-loop control algorithm

Command Response: A SUCCESS status is set if the desired loop control algorithm is selected.

An INVALID_ARGUMENT status is set if the value is not a valid algorithm.

Read Closed-Loop Control Gain

▶ Controllers

Firmware: 7v08.0, also in 7v07.1.

Command ID: 14

Returns the current value of the desired loop gain in the loop control algorithm.

Argument: Use the value from the following table to select the desired closed-loop gain to read.

Value	Description
0	Proportional gain (P gain) for PDF or PD^2I
1	Derivative gain (D gain) for PDF or PD^2I
2	Integral gain (I gain) for PD^2I

Command Response: A SUCCESS status is set if the argument is a valid choice. The return value is the gain value of the requested closed-loop gain. The value can be between 0–65535.

An INVALID_ARGUMENT is set if the gain requested is not a valid choice for the loop control algorithm.

Set Proportional Closed-Loop Control Gain

▶ Controllers

Firmware: 6v17.0

Command ID: 8

Changes the proportional gain value (P gain) to the value in the argument.

Argument: Use a value between 0–65535.

Command Response: A SUCCESS status is set when the command completes, and the gain value is set.

Set Integral Closed-Loop Control Gain

▶ Controllers

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 10

Changes the integral gain value (I gain) to the desired value used in the argument.

Argument: Use a value between 0–65535.

Command Response: A SUCCESS status is set when the command completes, and the gain is set.

Set Derivative Closed-Loop Control Gain

▶ Controllers

Firmware: 6v17.0

Command ID: 9

Changes the derivative gain value (D gain) to the desired value in the argument.

Argument: Use a value between 0–65535.

Command Response: A SUCCESS status is set when the command completes, and the gain is set.

Control Gain Gas Compensation

▶ Mass Flow Controllers

Firmware: 10v13.0

Command ID: 62

Sets or queries if gas compensation is enabled for closed loop control gains.

Argument: Use the desired mode from the table below for closed loop gain gas compensation.

Value	Description
0	Do not change the current gas compensation; this is only a query.
1	Disable gas compensation for closed loop gains.
2	Enable gas compensation for closed loop gains.
3	Enable gas compensation for closed loop gains until a closed loop control gain is changed by the user.

Command Response: A SUCCESS status is set if the instrument supports gas compensation of closed loop control gains. The return value is the current mode for closed loop gain gas compensation.

An UNSUPPORTED status is set if the instrument is not a controller or does not support setting the active gas properties.

An INVALID_ARGUMENT status is set if the mode is not one of the valid values.

Inverse Pressure Control

▶ Controllers

Firmware: 7v18.0

Command ID: 16

Queries, enables, or disables inverse pressure control.

Argument: Use a value from the following table.

Value	Description
0	Enable normal pressure control, not saved across power cycles.
1	Enable inverse pressure (usually back pressure) control, not saved across power cycling.
3	Enable normal pressure control, saved across power cycles.
4	Enable inverse pressure (usually back pressure) control, saved across power cycling.
65535	Query if the control loop currently uses inverse action. A*dded in 10v19.0*

Command Response: A SUCCESS status is set if the inverse pressure mode has changed.

An INVALID_ARGUMENT status is set if the argument used is not available on the instrument or does not have an associated pressure control mode.

Closed Loop Active on Zero Pressure Setpoint

▶ Controllers

Firmware: 10v19.0

Command ID: 51

Queries, enables, or disables if closed loop control is active for a zero pressure setpoint. Controlling flow always closes the valve(s) at a zero setpoint. If the control loop is disabled, the valve behavior for a zero pressure setpoint is as follows:

Dual Valve Control	Inverse Pressure Control	Negative Pressure Range	Valve State at Zero Setpoint if not Active Control
No	No	No	Close valve
No	No	Yes	Open valve
No	Yes	No	Open valve

No	Yes	Yes	Close valve
Yes	--	No	Open downstream valve
Yes	--	Yes	Open upstream valve

Argument: Use `65535` to query the current value, `0` to disable closed loop control for a zero pressure setpoint, or `1` to enable closed loop control for a zero pressure setpoint.

Command Response: A SUCCESS status is set if the closed loop control is enabled or disabled as requested, and the current enable state is in the return value.

An INVALID_ARGUMENT status is set if the mode is not a valid value.

An UNSUPPORTED status is set if the instrument is not a controller.

Closed Loop Deadband (Integer)

▶ Controllers

Firmware: 10v19.0

Command ID: 65554

Queries or sets the closed loop deadband limit, as an integer. Control deadband is rarely used except when controlling pressure with a dual valve controller.

Argument: The amount that the process value can drift from the setpoint and remain in the deadband state. Use `-1` to query the current limit. Otherwise, use an integer in the same units as the setpoint; multiply the engineering value by 10^(setpoint decimal places) to provide the integer value.

Command Response: A SUCCESS status is set when the limit is updated and available in the return value.

An INVALID_ARGUMENT status is set if the requested limit is too large or too small.

An UNSUPPORTED status is set if the instrument is not a controller.

Closed Loop Deadband (Float)

▶ Controllers

Firmware: 10v19.0

Command ID: 65555

Queries or sets the closed loop deadband limit, as an IEEE-754 single precision floating point value. Control deadband is rarely used except when controlling pressure with a dual valve controller.

Argument: The amount that the process value can drift from the setpoint and remain in the deadband state. Use `-1` to query the current limit. Otherwise, use a floating point value in the same units as the setpoint.

Command Response: A SUCCESS status is set when the limit is updated and available in the return value.

An INVALID_ARGUMENT status is set if the requested limit is too large or too small.

An UNSUPPORTED status is set if the instrument is not a controller.

Closed Loop Deadband Mode

▶ Controllers

Firmware: 10v19.0

Command ID: 41

Queries or sets the deadband mode for closed loop control.

Argument: Use `0` to query the current mode, `1` to close the valves when the deadband is active, `2` to hold the valve positions when within the deadband.

Command Response: A SUCCESS status is set when the deadband mode has been set as requested, and the mode is in the return value.

An INVALID_ARGUMENT status is set if the mode is not a valid value.

An UNSUPPORTED status is set if the instrument is not a controller.

Query Totalizer Batch (Integer)

▶ Flow Controllers

Firmware: 10v07.0

Command ID: 65543

Queries the total size of the batch from one of the two totalizers.

Argument: Use a value of 1 to query Totalizer 1. Use a value of 2 to query Totalizer 2.

Command Response: A SUCCESS status is set if the requested totalizer is enabled. The return value is the total size of the batch of the requested totalizer. To obtain the volume in the requested totalizer's engineering units, multiply the value by 10^(- totalizer decimal places). The instrument returns a 0 if batching is disabled.

An INVALID_ARGUMENT status is set if the requested totalizer is not enabled.

Query Totalizer Batch (Float)

▶ Flow Controllers

Firmware: 10v07.0

Command ID: 65540

Queries the total size of the batch from one of the two totalizers.

Argument: Use a value of 1 to query Totalizer 1. Use a value of 2 to query Totalizer 2.

Command Response: A SUCCESS status is set if the requested totalizer is enabled on the instrument. The return value is the total size of the batch in the requested totalizer in a single-precision floating point formatted value. This value uses the engineering units of the totalizer. The instrument returns a 0 if batching is disabled.

An INVALID_ARGUMENT status is set if the requested totalizer is not enabled.

Set Totalizer Batch 1 (Integer)

▶ Flow Controllers

Firmware: 10v07.0

Command ID: 65544

Sets the batch volume for Totalizer 1, as an integer.

Argument: Use the desired batch size multiplied by 10^(totalizer decimal places).

To disable batching, use a value of 0.

Command response: A SUCCESS status is set if Totalizer 1 is enabled and the size of the batch is valid. The return value is the batch size.

An UNSUPPORTED status is set if Totalizer 1 is not enabled.

An INVALID_ARGUMENT status is set if the volume of the batch is larger than what Totalizer 1 can hold.

Set Totalizer Batch 1 (Float)

▶ Flow Controllers

Firmware: 10v07.0

Command ID: 65541

Sets the batch volume for Totalizer 1, as an IEEE-754 single precision floating point value.

Argument: Use the desired batch size formatted as a single-precision floating point value in the engineering units of Totalizer 1.

To disable batching, use a value of 0.

Command response: A SUCCESS status is set if totalizer 1 is enabled and the batch size is valid. The return value is the requested batch volume.

An UNSUPPORTED status is set if Totalizer 1 is not enabled.

An INVALID_ARGUMENT status is set if the volume of the batch is larger than what Totalizer 1 can hold.

Set Totalizer Batch 2 (Integer)

▶ Flow Controllers

Firmware: 10v07.0

Command ID: 65545

Sets the batch volume for Totalizer 2, as an integer.

Argument: Use the desired batch size multiplied by 10^(totalizer decimal places). To disable batching, use a value of 0.

Command response: A SUCCESS status is set if Totalizer 2 is enabled and the size of the batch is valid. The return value is the batch size.

An UNSUPPORTED status is set if Totalizer 2 is not enabled.

An INVALID_ARGUMENT status is set if the volume of the batch is larger than what Totalizer 2 can hold.

Set Totalizer Batch 2 (Float)

▶ Flow Controllers

Firmware: 10v07.0

Command ID: 65542

Sets the batch volume for Totalizer 2, as an IEEE-754 single precision floating point value.

Argument: Use the desired batch size formatted as a single-precision floating point value in the engineering units of Totalizer 2. To disable batching, use a value of 0.

Command response: A SUCCESS status is set if Totalizer 2 is enabled and the batch size is valid. The return value is the requested batch volume.

An UNSUPPORTED status is set if Totalizer 2 is not enabled.

An INVALID_ARGUMENT status is set if the volume of the batch is larger than what Totalizer 2 can hold.

Check Setpoint Response Test Setpoint (Setpoint Counts)

▶ Controllers

Firmware: 10v13.0

Command ID: 63

Queries or sets the value to which the setpoint will change when checking the setpoint response.

Argument: Use the desired setpoint value in setpoint counts, 0–65534. To query the test setpoint, use 65535.

Completion Response: A SUCCESS status is set if the test setpoint was successfully set or queried. The return value is the test value that will be used in the setpoint response check, in setpoint counts.

An UNSUPPORTED status is set if the instrument is not a controller.

An INVALID_ARGUMENT status is set if the test setpoint is outside of the range 0–65535.

Check Setpoint Response Test Setpoint (Integer)

▶ Controllers

Firmware: 10v13.0

Command ID: 65565

Queries or sets the value to which the setpoint will change when checking the setpoint response.

Argument: Use the desired setpoint value, scaled in setpoint engineering units by 10^(setpoint decimal places). Use -2147483648 to query the test setpoint.

Command Status: A SUCCESS status is set if the test setpoint was successfully set or queried. The return value is the test value that will be used in the setpoint response check. Scale this integer by 10^ (-setpoint decimal places) for the value in setpoint engineering units.

An INVALID_ARGUMENT status is set if the test setpoint is outside of the valid range.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Test Setpoint (Float)

▶ Controllers

Firmware: 10v13.0

Command ID: 65564

Queries or sets the value to which the setpoint will change when checking the setpoint response.

Argument: Use the desired setpoint value, in the setpoint engineering units. To query the test setpoint, use -3.40282346639e+38 (hexadecimal value ff7fffff). It is formatted as a single-precision floating point value.

Command Status: A SUCCESS status is set if the test setpoint was successfully set or queried. The return value is the test value that will be used in the setpoint response check, formatted as a single-precision floating point value, in the setpoint engineering units.

An INVALID_ARGUMENT status is set if the test setpoint is outside of the valid range.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Collection Time

▶ Controllers

Firmware: 10v13.0

Command ID: 64

Queries or sets the amount of time a setpoint response check collects data after changing the setpoint.

Argument: Use the amount of collection time, in seconds, 1–65535. To query the collection time, use 0.

Command Status: A SUCCESS status is set if the collection time was successfully set or queried. The return value is the amount of time a setpoint response check collects process data after the setpoint is changed, in seconds.

An INVALID_ARGUMENT status is set if the collection time is outside of the range 0–65535.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Run Status

▶ Controllers

Firmware: 10v13.0

Command ID: 66

Queries the execution state of the setpoint response check, and possibly start or abort the response check.

Argument: Use one of the values in the following table.

Value	Notes
0	This is only a query of the check setpoint response state; take no action.
1	Begin a check setpoint response. The instrument will change the setpoint (overriding the analog setpoint if the setpoint source is analog) and collect data for the specified amount of time. The setpoint will return to normal once data collection is complete.
2	Abort a currently executing check setpoint response. If a check is not currently executing, this does nothing.

Command Status: A SUCCESS status is set if the action, if any, was successfully executed. The return value is the status of the currently executing or last completed check setpoint response:

Value	Notes
0	The last check setpoint response has successfully completed and the results calculated.
1	No check setpoint response has been executed since power up.
2	Unable to start a check setpoint response, due to closed loop control gain tuning or another check setpoint response being active.
3	Check setpoint response is currently executing.
4	The data collection was aborted early by user action; the returned values are all zero.
5	The data collection terminated early due to oscillation; the returned values are all zero.

An UNSUPPORTED status is set if the instrument is not a controller.

An INVALID_ARGUMENT status is set if the action is not one of the listed values.

Check Setpoint Response Results Notes

▶ Controllers

Firmware: 10v13.0

Command ID: 67

Queries any notes about unusual circumstances observed during the last completed check setpoint response.

Argument: Always use 0.

Command Status: A SUCCESS status is set if the action, if any, was successfully executed. The return value will be notes regarding unusual circumstances that may affect the accuracy of the results (see the table below). Multiple values in the following table may be added together as a single data collection.

Value	Notes
0	There are no notes on the calculated results; the process data showed typical patterns.
+1	The data collection time was not long enough to ensure reliable calculated results; the values may change with a longer data collection time.
+2	The process variable oscillated significantly. Control was not stable and the calculated results are not reliable.
+4	The process variable moved a small amount compared to the noise observed in the system, so the calculated results may not be reliable. This most commonly happens when the setpoint change is too small or when the process variable oscillated in a manner that was not caught by the oscillation detection algorithm.
+8	A previously closed valve opened during data collection. The results are computed correctly, but the performance will be significantly different from situations that only move an already open valve.
+16	A valve was opened as far as it could go during data collection. The results are computed correctly, but the performance is likely limited by the size of the valve, not the control loop configuration.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Overshoot (Setpoint Counts)

▶ Controllers

Firmware: 10v13.0

Command ID: 68

Queries the overshoot of the last completed setpoint response check.

Argument: Always use 0.

Command Status: A SUCCESS status is set if the overshoot was successfully queried. The return value is the overshoot measured in the last completed setpoint response check, in setpoint counts.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Overshoot (Integer)

▶ Controllers

Firmware: 10v13.0

Command ID: 65567

queries the overshoot of the last completed setpoint response check.

Argument: Always use 0.

Command Status: A SUCCESS status is set if the overshoot was successfully queried. The return value is the overshoot measured in the last completed setpoint response check. Scale this integer by 10^ (-setpoint decimal places) for the value in setpoint engineering units.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Overshoot (Float)

▶ Controllers

Firmware: 10v13.0

Command ID: 65566

Queries the overshoot of the last completed setpoint response check.

Argument: Always use 0.

Command Status: A SUCCESS status is set if the overshoot was successfully queried. The return value is the overshoot measured in the last completed setpoint response check, formatted as a single-precision floating point value, in the setpoint engineering units.

An UNSUPPORTED status is set if the instrument is not a controller.

Check Setpoint Response Timing Results

▶ Controllers

Firmware: 10v13.0

Command ID: 69

Check setpoint response timing results queries the timings measured in the last completed check setpoint response.

Argument: Use the timing parameter to query from the table below:

Value	Notes
0	Query the measured time constant of the last completed check.
1	Query the measured dead time of the last completed check.
2	Query the measured rise time of the last completed check.
3	Query the measured bandwidth of the last completed check.

Command Status: A SUCCESS status is set if the action, if any, was successfully executed. The return value is the timing value requested in the argument:

Value	Notes
0	The closed loop time constant (T63) measured from the trial, in milliseconds. This is appropriate for use in a first-order-plus-dead-time (FOPDT) model of the instrument.
1	The system dead time or delay measured from the control trial, in milliseconds. This is appropriate for use in a first-order-plus-dead-time (FOPDT) model of the instrument.
2	The time required for the closed loop response to go between 10% of the step and 90% of the step during the trial, in milliseconds.
3	The estimated bandwidth of the closed loop response, calculated from the rise time; multiply the number by 0.01 for the bandwidth in hertz.

An UNSUPPORTED status is set if the instrument is not a controller.

An INVALID_ARGUMENT status is set if the value is not one of the values in the table above.

Control Optimization Max Flow (Setpoint Counts)

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 70

Queries or sets the maximum flow that control optimization will intentionally flow, in setpoint counts.

Argument: Use the maximum desired flow in setpoint counts, 1–64000. Use 0 to set the limit to full scale flow. Use 65535 to query the current maximum flow.

Command Status: A SUCCESS status is set if the maximum flow was successfully set or queried. The return value is the current maximum desired flow, in setpoint counts.

An INVALID_ARGUMENT status is set if the maximum flow is outside the range 0–64000 and is not 65535, or if the maximum flow is less than 20% of the full scale flow.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Max Flow (Integer)

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 65569

Queries or sets the maximum flow that control optimization will intentionally flow, as an integer.

Argument: Use the maximum desired flow, scaled to the desired value in setpoint engineering units by 10^(setpoint decimal places). Use 0 to set the limit to full scale flow. Use a value < 0 to query the current maximum flow.

Command response: A SUCCESS status is set if the maximum flow was successfully set or queried. The return value is the current maximum desired flow. Scale this integer by 10^ (-setpoint decimal places) for the value in setpoint engineering units.

An INVALID_ARGUMENT status is set if the maximum flow is less than 20% of the full scale flow and is not 0.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Max Flow (Float)

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 65568

Queries or sets the maximum flow that control optimization will intentionally flow, in a floating-point format.

Argument: Use the maximum desired flow, in the setpoint engineering units. Use 0 to set the limit to full scale flow. Use a value < 0 to read the current maximum flow. This is formatted as a single-precision floating point value.

Command response: A SUCCESS status is set if the maximum flow was successfully set or queried. The return value is the current maximum desired flow, formatted as a single-precision floating point value, in the setpoint engineering units.

An INVALID_ARGUMENT status is set if the maximum flow is less than 20% of the full scale flow and is not 0.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Speed Mode

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 71

Queries or sets the configured speed mode for control optimization.

Argument: Use the speed mode from the table below.

Value	Notes
0	Queries the speed mode.
1	Targets the most versatile usage, which will have the slowest speed of response.
2	Emphasizes versatility over speed of response.
3	Emphasizes faster speed of response over versatility.
4	Maximizes the speed of response (minimize the control loop response time constant). A small amount of overshoot in the response is allowed.

Command response: A SUCCESS status is set if the speed mode has been successfully set or queried. The return value is the current speed mode, per the table above.

An INVALID_ARGUMENT status is set if the speed mode is not listed in the table above.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Goal Time

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 72

Queries or sets the goal response time used by control optimization.

Argument: Use the desired control response time constant in milliseconds. Use 0 to use the fastest possible time constant (the same as the fastest speed mode). Use 65535 to query the goal response time.

The control response after optimization will have a time constant close to this value for the tested setpoints, if the system can support it. If the requested time constant is smaller than the system can reasonably accommodate, optimization will find the gains with the shortest time constant.

If during optimization one of the gains becomes 1 and therefore cannot become smaller to increase the time constant, the resulting time constant may be faster than requested.

Command response: A SUCCESS status is set if the goal response time has been successfully set or queried. The return value is the current desired response time, in milliseconds. If a speed mode is active, the return value will be 0.

An INVALID_ARGUMENT status is set if the goal time is negative.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Control Algorithm

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 73

Queries or sets the control algorithm that control optimization will use to determine gains.

Argument: Use the desired closed loop control algorithm from the table below:

Value	Notes
0	This only a query; do not change the configured mode.
1	Control optimization chooses the most appropriate control algorithm based on instrument configuration and optimization parameters. Note that the algorithm chosen for a particular situation and configuration may change across firmware versions.
2	Force use of the PDF control algorithm. Speed modes will be 2x–3x slower than for other values of the argument.
3	Force use of the PD2I control algorithm.

Command response: A SUCCESS status is set if the control algorithm has been successfully set or queried. The return value is the current control algorithm from the table above.

An INVALID_ARGUMENT status is set if the algorithm is not a valid value from the table above.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Run State

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 74

Queries the current execution state of control optimization, and possibly take an action.

Argument: Use the desired closed loop control algorithm from the table below:

Action	Notes
0	Only query the current state of execution, take no action.
1	Abort any current control optimization.
2	Begin a new control optimization with the current settings values. Any running control optimization will be aborted before starting the new one.
3	The same as action 2, but pause the optimization procedure at each possible point, closing the valve(s).
4	Request that optimization pause and close the valves at the next possible point. A pause is most commonly used to allow a restricted volume that was filled during the optimization process to bleed off to more normal pressure levels. If there is no currently paused optimization, this does nothing.
5	Resume a currently paused optimization.

Command response: A SUCCESS status is set if the requested action, if any, has been performed successfully. The return value is the current activity of the control optimization algorithm:

Mode	Notes
0	Control optimization is not active.
1	Control optimization is actively driving the valves in the instrument.
2	Control optimization is active but has been paused. The valves are closed until this command is issued with a resume argument.
3	An abort has been requested, but control optimization is still active.

An INVALID_ARGUMENT status is set if the action is not in the table of valid values.

An UNSUPPORTED status is set if the instrument is not a controller.

Control Optimization Result

▶ Flow Controllers

Firmware: 10v13.0

Command ID: 75

Control optimization result queries the result of the last completed control optimization.

Argument: Always use 0.

Command response: A SUCCESS status is set if the optimization result has been returned. The return value is the result of the last control optimization:

Result	Notes
0	A control optimization is in progress.
1	The last control optimization completed successfully.
2	The last control optimization was aborted by the user or due to an error.
3	There was an unexpected state during the last control optimization.
4	The maximum flow allowed for optimization is too small.
5	A valve requested to be used for control optimization is not configured for closed loop control.
6	The instrument had no closed loop control configured at the time of the last control optimization.
7	There was not enough flow to successfully complete the last control optimization. Check the feed and outlet pressures.
8	The last control optimization could not find any gains that yielded acceptable performance.

An UNSUPPORTED status is set if the instrument is not a controller.

Administrative

Administrative commands are used to perform actions such as taring the instrument sensors, creating gas mixes, controlling the display, and restoring factory settings.

Perform Tare

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 4

Instructs the instrument to use the current reading as the zero reading.

When performing gauge pressure tares, the sensor must be open to the atmosphere.

Differential pressure tares require a common pressure from which to measure. Absolute pressure tares require an equipped barometer.

Flow tares must be performed when there is no current flow through the process. Flow tares are also best when at the process pressure, or as close to it as possible.

Argument: Use the value of the desired tare found in the following table.

Value	Description
0	Tare gauge or differential pressure
1	Tare absolute pressure
2	Tare mass and/or volumetric flow

Command response: A SUCCESS status is set if the requested tare is performed.

An UNSUPPORTED status is set if the requested tare does not apply (e.g., a mass flow tare on a pressure instrument).

Perform Pressure Sensor Tare

Firmware: 10v07.0

Command ID: 31

Instructs the instrument to use the current gauge or differential pressure reading as the zero reading. An absolute pressure tare uses the barometer's zero reading as the zero reading.

When performing gauge pressure tares the sensor must be open to the atmosphere.

Differential pressure tares require a common pressure to measure from.

Absolute pressure tares require an equipped barometer**.**

Argument: Use the desired time in milliseconds for the tare to take. This can be a value of 0–32767. If a value of 0 is used, a default of 256 milliseconds is used.

Command response: A SUCCESS status is set if the tare is performed.

An INVALID_ARGUMENT status is set if the argument value is less than 0 or greater than 32767.

An UNSUPPORTED status is set if the pressure sensor cannot be tared. This is usually due to trying to tare absolute pressure without a barometer.

Perform Secondary Pressure Sensor Tare

Firmware: 10v07.0

Command ID: 32

Instructs the instrument to use the current pressure reading on the secondary pressure sensor as the zero reading. An absolute pressure tare uses the barometer's zero reading as the zero reading. Not all instruments have a secondary pressure sensor.

When performing gauge pressure tares the sensor must be open to the atmosphere.

Differential pressure tares require a common pressure to measure from.

Absolute pressure tares require an equipped barometer.

Argument: Use the desired time in milliseconds for the tare to take. This can be a value of 0–32767. If a value of 0 is used, a default of 256 milliseconds is used.

Command response: A SUCCESS status is set if the tare is performed.

An INVALID_ARGUMENT status is set if the argument value is less than 0 or greater than 32767.

An UNSUPPORTED status is set if the pressure sensor cannot be tared. This is usually due to attempting to tare absolute pressure without a barometer or not having a secondary pressure sensor.

Perform Flow Tare

▶ Flow instruments

Firmware: 10v07.0

Command ID: 33

Instructs the instrument to use the current flow reading as the zero reading. This command is used for both volumetric and mass flow.

Argument: Use the desired time in milliseconds for the tare to take. This can be a value of 0–32767. If a value of 0 is used, a default of 256 milliseconds is used.

Command response: A SUCCESS status is set if the tare is performed.

An INVALID_ARGUMENT status is set if the argument value is less than 0 or greater than 32767.

An UNSUPPORTED status is set if the pressure sensor cannot be tared. This is usually due to the instrument not reading volumetric or mass flow.

Reset Totalizer

▶ Flow instruments

Firmware: 10v19.0.

Command ID: 80

Resets the requested totalizer to a value of zero.

Argument: Use 1 to reset Totalizer 1, or 2 to reset Totalizer 2.

Command response: A SUCCESS status is set if the reset is complete.

An INVALID_ARGUMENT status is set if the argument is not `1` or `2`.

Reset Totalizer 1

▶ Flow instruments

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 5

Resets Totalizer 1 to a value of zero.

Argument: No argument is used for this command

Command response: A SUCCESS status is set if the reset is complete.

Set Selected Gas

▶ Mass flow instruments

Firmware: *7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.*

Command ID: 1

Updates the selected gas in the instrument.

Argument: Use the index number of the desired gas to measure. The gas index can be found on page 92.

Command response: A SUCCESS status is set if the gas has changed.

An INVALID_ARGUMENT status is set if the requested gas number does not exist.

Create/Update Gas Mix

▶ Mass flow instruments

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 2

Creates or updates a COMPOSER™ gas mix. The mix is a two-step process. First, the desired constituent gas indexes and percentages must be written to the mix registers followed by a write of the Mix Gas command (ID 2) into command register 1000.

Gas mix percentages are interpreted as integer hundredths of a percent and the total percentage must sum to 100%. For example, to specify a mix of 50%, a value of 5000 should be written into the gas percentage. The mix will be performed with the first N gases that have a nonzero percentage. As an example.

If the command argument passed to the mix command is 0 or is omitted, a new gas mix index will be allocated in the next empty gas mix index starting at 255 and working down to 236. If no user mix indices are unused, an error will be returned in the command status.

If the command argument passed is between 236 and 255, the mixture with the specified index will be either created or updated to the new composition. If the specified index is not valid (the command argument is neither 0 nor 236–255), an error will be returned.

Upon completion of mixing, the command result will be updated with the mix result. If the mix was valid, the index of the mixed gas will be returned. If one of the requested gas mix constituents did not exist or the percentage does not add to 100%, an error value will be returned, and the mix will not be created. The created gas will have the name Mn, where n is the gas index.

Argument: Use 0 or 236 - 255. This is the gas index number of the new mixture. A value of 0 instructs the instrument to write to the first available gas index number starting with 255 and moving down. If no gas index numbers are available, then the command will fail.

To update or overwrite a prior custom gas mix, using its gas index number here will overwrite the old mixture.

Command response: A SUCCESS status is set if the gas has changed. The return value is the index number of the gas mix created or updated.

An INVALID_MIX_IDX status is set if the gas index number in the argument is outside of the 236 - 255 range.

An INVALID_MIX_GAS status is set if one or more gases in the mix do not exist on the instrument.

An INVALID_MIX_PCT status is set if the percentages of the gases that make up the mixture do not sum 100%.

Set Gas Mix Constituent Gas Number

▶ Mass Flow instruments

Firmware: 10v19.0, also in 9v07.4.

Command ID: 52, 54, 56, 58, 60

Sets the gas number of the COMPOSER™ gas mix constituent. ID 52 sets the first constituent, 54 sets the second, 56 sets the third, 58 sets the fourth, and 60 sets the fifth.

Argument: Use the number of the COMPOSER gas mix constituent.

Command response: A SUCCESS status is set if the COMPOSER constituent gas number was set to the requested value.

An INVALID_ARGUMENT status is set if the requested gas number does not exist.

Set Gas Mix Constituent Fraction

▶ Mass Flow instruments

Firmware: 10v19.0, also in 9v07.4.

Command ID: 53, 55, 57, 59, 61

Sets the fraction of the COMPOSER™ gas mix constituent. ID 53 sets the first constituent, 55 sets the second, 57 sets the third, 59 sets the fourth, and 61 sets the fifth.

Argument: Use the fraction of the COMPOSER gas mix constituent, 0.01% per count (10000 = 100%).

Command response: A SUCCESS status is set if the COMPOSER constituent gas fraction was set to the requested value.

An INVALID_ARGUMENT status is set if the requested fraction was over 10000 (100%).

Delete Gas Mix

▶ Mass Flow instruments

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 3

Deletes a COMPOSER™ gas mix.

Argument: Use the number of the desired gas to delete.

Command response: A SUCCESS status is set if the gas mix was deleted.

An INVALID_MIX_IDX status is set if the requested gas mix does not exist.

Set Relative Humidity Percentage

▶ Instruments with humidity sensor

Firmware: 9v04.0, /RHM builds only. Added to all builds in 10v07.0

Command ID: 24

Sets the relative humidity level used for gas corrections.

Argument: Use a value between 0–10000. 1 count in the value is equal to 0.01% humidity. For example, use 100 for 1% or 10000 for 100%.

Command response: A SUCCESS status is set if the relative humidity percentage is changed.

An UNSUPPORTED status is set if the relative humidity cannot be set on the instrument.

An INVALID_ARGUMENT status is set if the argument value is outside of 0–10000.

Set Relative Humidity Reference Temperature

▶ Instruments with humidity sensor

Firmware: 10v07.0

Command ID: 25

Sets the relative humidity reference temperature used for gas corrections. This temperature is Celsius.

Argument: Use a value between −3000–10000. This value should match the relative humidity percentage.

The range starts at −30°C and reaches 100°C. 1 count in the value is equal to 0.01°C. For example, a value of −3000 is −30°C and a value of 10000 is 100°C.

Command response: A SUCCESS status is set if the relative humidity reference temperature is changed.

An UNSUPPORTED status is set if the relative humidity cannot be set on the instrument.

An INVALID_ARGUMENT status is set if the argument value is outside of −3000–10000.

Lock/Unlock Display

Firmware: 7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.

Command ID: 7

Locks or unlocks the display on the front of the instrument. When locked, the instrument still responds to button presses, but settings cannot be changed using the display.

Argument: Use a value of 0 to unlock the display. Any other value locks the display.

Command response: A SUCCESS status is set if the display is locked or unlocked as requested.

Flash Display

Firmware: 8v28.0

Command ID: 20

Instructs the instrument to flash its monochrome or TFT display backlight. Pressing any key will stop the flashing.

Argument: Use 65535 to flash the backlight indefinitely. Use 0 to stop the backlight from flashing. Use any other value to flash the backlight for that number of seconds.

Command response: A SUCCESS status is set if the backlight is flashing or stopped as requested.

An UNSUPPORTED status is set if no display is connected to the instrument.

Read Serial Number

Firmware: 10v19.0

Command ID: 65570

Returns the serial number of the instrument.

Argument: Must be 0.

Command response: A SUCCESS status is set if the serial number is available in the return value.

An INVALID_ARGUMENT status is set if the value is not 0.

Read Partial Serial Number

Firmware: 10v19.0

Command ID: 81

Returns a portion of the serial number of the instrument. Included for protocols that only support basic commands.

Argument: Use `0` for the lowest two bytes of the serial number, or `1` for the highest two bytes of the serial number.

Command response: A SUCCESS status is set if the requested portion of the serial number is available in the return value.

An INVALID_ARGUMENT status is set if the value is not 0 or 1.

Read Manufacture Date

Firmware: 10v19.0

Command ID: 82

Returns a portion of the date of manufacture.

Argument: Use a value from the following table:

Value	Description
0	Year

1	Month
2	Day

Command response: A SUCCESS status is set if the requested portion of the manufacture date is available in the return value.

An INVALID_ARGUMENT status is set if the value is not 0, 1, or 2.

Read Calibration Date

Firmware: 10v19.0

Command ID: 83

Returns a portion of the date of calibration.

Argument: Use a value from the following table:

Value	Description
0	Year
1	Month
2	Day

Command response: A SUCCESS status is set if the requested portion of the calibration date is available in the return value.

An INVALID_ARGUMENT status is set if the value is not 0, 1, or 2.

Read Firmware Version

Firmware: 10v19.0

Command ID: 84

Returns a portion of the firmware version.

Argument: Use a value from the following table:

Value	Description
0	Major portion
1	Minor portion
2	Custom portion
3	Internal portion

Command response: A SUCCESS status is set if the portion of the firmware version is available in the return value.

An INVALID_ARGUMENT status is set if the value is not 0, 1, 2, or 3.

Read Configuration Checksum

Firmware: 8v24.0, also in 8v22.2.

Command ID: 17

Computes and returns a checksum of the instrument's calibration and configuration. The checksum is computed at the time of the command initiation and may take 300 milliseconds to complete. A timeout of over 500 milliseconds is recommended.

Any setting that is retained across power cycles is included. Values that may or may not be retained are also included.

Because calibration information is included, two instruments with identical configurations may have different checksums. If an instrument is recalibrated, the checksum may change as well. Changing the instrument firmware may or may not change the checksum.

Any parameter that changes during routine operation (e.g., setpoint) should be set to a known configuration before reading the checksum.

Argument: Must always be 0.

Command response: A SUCCESS status is set when the checksum is complete, and the argument value is 0. The return value is the checksum of the entire instrument configuration. This is a value of 0–65535.

An INVALID_ARGUMENT status is set when the argument value is not 0.

Restore Factory Settings

Firmware: 10v07.0

Command ID: 26

Reverts all the instrument settings and configurations to their values when the instrument was last at Alicat.

! WARNING***: This command should only be used when trying to troubleshoot issues with Alicat support (page 2). All third-party calibrations are removed by performing this command.***

The instrument needs to be power cycled after performing the restore.

Argument: Use a value of 49374 to confirm that a factory restore is the desired result of the command.

Command response: A SUCCESS status is set if the factory restore completes. This may only be available to read briefly before the restore removes the status.

An INVALID_ARGUMENT status is set if the argument value is not 49374.

An UNSUPPORTED status is set if there is an error during the restore process.

Communications

Set Modbus Slave Address

Firmware: *7v05.0, also in 6v17.1, 6v17.2, 6v17.3, 6v20.2, 6v20.3, 6v20.4, 6v20.5, 6v20.6.*

Command ID (Modbus RTU only): 32767

Sets the Modbus slave address of the instrument, 1–247.

An INVALID_ARGUMENT status is set if the requested address is out of range.

Set Communications Watchdog Timeout

Firmware: 8v24.0, also in 8v22.2.

Command ID (Modbus RTU only): 32765

Sets the timeout used for the communications watchdog.

The communications watchdog will execute the configured disconnection logic when more than the timeout amount of time has passed since the last successful communication was received over Modbus. When a disconnection occurs, a COM status will flash on the main display, the setpoint will be set to 0 (unless otherwise configured), and on an MCT instrument, the primary valve will be selected (unless otherwise configured). Receiving a successful communication immediately resets the watchdog and normal operation is restored.

A message with a checksum failure or one addressed to another instrument is not counted as a successful communication. However, a correctly formatted message with a broadcast address, an unsupported function code, an invalid register address, or invalid register value is still counted as a successful communication. For long running commands, the timeout starts as transmission of the response begins.

Argument: The watchdog timeout, 0–65536, in 0.1 seconds. For example, 100 = 10.0 second timeout. A value of 0 disables the watchdog.

Command response: A SUCCESS status is set if the watchdog timeout has been updated.

Hardware

Set Primary Analog Output Value

Firmware: 8v24.0, also in 8v22.2.

Command ID: 18

Sets the value provided by the primary analog output

Argument: The value to output, 0–65535. A value of 0 is equivalent to the minimum of the instrument (e.g., 0 V or 4 mA). A value of 64000 is equivalent to full scale on the instrument (e.g., 5 V or 20 mA)

Command response: A SUCCESS status is set if the primary analog output value has been updated.

An UNSUPPORTED status will be set if the primary analog output is not supported on this instrument or is not configured for this operation.

An INVALID_ARGUMENT will be set if the requested value is out of range.

Set Secondary Analog Output Value

Firmware: 8v24.0, also in 8v22.2.

Command ID: 19

Sets the value provided by the secondary analog output

Argument: The value to output, 0–65535. A value of 0 is equivalent to the minimum of the instrument (e.g., 0 V or 4 mA). A value of 64000 is equivalent to full scale on the instrument (e.g., 5 V or 20 mA)

Command response: A SUCCESS status is set if the secondary analog output value has been updated.

An UNSUPPORTED status will be set if the secondary analog output is not supported on this instrument or is not configured for this operation.

An INVALID_ARGUMENT will be set if the requested value is out of range.

Appendices

Appendix A: Statistics

Statistics are readings and measurements that instruments provide. Use the following values in commands as indicated to apply the desired statistic to the command.

Some statistics are restricted to specific instruments and configurations. Please contact Alicat support (page 2) with any questions about a statistic and how it may work with your instrument.

Appendix A-1: Flow Statistics

Flow Statistic	Value	Notes
Mass flow	5	Current mass flow
Mass flow setpoint	37	Setpoint for mass flow
Total mass	9	Totalized mass
Total volume	8	Totalized volume, referenced to flow conditions.
Volumetric flow	4	Volumetric flow, referenced to flow conditions.
Volumetric flow setpoint	36	The setpoint for volumetric flow referenced to flow conditions.

Appendix A-2: Pressure Statistics

Pressure Statistic	Value	Notes
Pressure, absolute	2	Current absolute pressure
Pressure, absolute setpoint	34	Setpoint for absolute pressure
Pressure, barometric	15	Barometer reading
Pressure, differential	7	Current differential pressure reading
Pressure, differential setpoint	39	Setpoint for differential pressure
Pressure, gauge	6	Current gauge pressure reading
Pressure, gauge setpoint	38	Setpoint for gauge pressure
Pressure, second absolute	344	For instruments with a second pressure sensor, the absolute pressure of the second sensor
Pressure, second absolute setpoint	345	Setpoint for the second absolute pressure.
Pressure, second differential	360	For instruments with a second pressure sensor, the differential pressure of the second sensor
Pressure, second differential setpoint	361	Setpoint for the second differential pressure
Pressure, second gauge	352	For instruments with a pressure sensor in a second location, the gauge pressure of the second sensor. For DILO, this is the pressure upstream of the orifice.

Pressure Statistic	Value	Notes
Pressure, second gauge setpoint	353	Setpoint for the second gauge pressure

Appendix A-3: Other Statistics

Other Statistic	Value	Notes
None	1	No statistic: usually implies an empty location.
Relative humidity	25	The fraction of complete saturation the instrument is currently using. When the vapor used is water, this is relative humidity.
Temperature, stream	3	Current stream temperature
Valve drive	13	Valve drive signal
Valve drive setpoint	45	The setpoint directly drives the currently selected valve.

Appendix B: Engineering Units

The following tables provide the values for engineering units for use in commands. The table to refer to is dependent on the statistic that is being modified. For example, use Appendix B-3 when modifying the engineering units for a totalizer measuring standard or normal volumes.

Appendix B-1: Standard and Normal Flow Units

Unit Label	Value	Notes
	0	Unit not specified: use default values.
---	1	Unknown unit: no conversions are performed to other units. If calibrated with different units, the value is displayed in those
SµL/m	2	Standard microliter per minute
SmL/s	3	Standard milliliter per second
SmL/m	4	Standard milliliter per minute
SmL/h	5	Standard milliliter per hour
SL/s	6	Standard liter per second
SLPM	7	Standard liter per minute
SL/h	8	Standard liter per hour
SCCS	11	Standard cubic centimeter per second
SCCM	12	Standard cubic centimeter per minute
Scm³/h	13	Standard cubic centimeter per hour
Sm³/m	14	Standard cubic meter per minute
Sm³/h	15	Standard cubic meter per hour
Sm³/d	16	Standard cubic meter per day
Sin³/m	17	Standard cubic inch per minute
SCFM	18	Standard cubic foot per minute
SCFH	19	Standard cubic foot per hour
SCFD	21	Standard cubic foot per day
kSCFM	20	1000 standard cubic feet per minute
NµL/m	32	Normal microliter per minute
NmL/s	33	Normal milliliter per second
NmL/m	34	Normal milliliter per minute
NmL/h	35	Normal milliliter per hour
NL/s	36	Normal liter per second
NLPM	37	Normal liter per minute
NL/h	38	Normal liter per hour
NCCS	41	Normal cubic centimeter per second

Unit Label	Value	Notes
NCCM	42	Normal cubic centimeter per minute
Ncm³/h	43	Normal cubic centimeter per hour
Nm³/m	44	Normal cubic meter per minute
Nm³/h	45	Normal cubic meter per hour
Nm³/d	46	Normal cubic meter per day
Count	62	Setpoint count, 0–64000
%	63	Percent of the full scale

Appendix B-2: True Mass Flow Units

Unit Label	Value	Notes
mg/s	64	Milligram per second
mg/m	65	Milligram per minute
g/s	66	Gram per second
g/m	67	Gram per minute
g/h	68	Gram per hour
kg/m	69	Kilogram per minute
kg/h	70	Kilogram per hour
oz/s	71	Ounce per second
oz/m	72	Ounce per minute
lb/m	73	Pound per minute
lb/h	74	Pound per hour

Appendix B-3: Total Standard and Normal Volume Units

Unit Label	Value	Notes
	0	Unit not specified: use default values.
---	1	Unknown unit: no conversions are performed to other units. If calibrated with different units, the value is displayed in those units.
SµL	2	Standard microliter
SmL	3	Standard milliliter
SL	4	Standard liter
Scm³	6	Standard cubic centimeter
Sm³	7	Standard cubic meter
Sin³	8	Standard cubic inch
Sft³	9	Standard cubic foot
kSft³	10	1000 standard cubic feet
NµL	32	Normal microliter

Unit Label	Value	Notes
NmL	33	Normal milliliter
NL	34	Normal liter
Ncm³	36	Normal cubic centimeter
Nm³	37	Normal cubic meter

Appendix B-4: Volumetric Flow Units

Unit Label	Value	Notes
	0	Unit not specified: use default values.
---	1	Unknown unit: no conversions are performed to other units. If calibrated with different units, the value is displayed in those units.
µL/m	2	Microliter per minute
mL/s	3	Milliliter per second
mL/m	4	Milliliter per minute
mL/h	5	Milliliter per hour
L/s	6	Liter per second
LPM	7	Liter per minute
L/h	8	Liter per hour
US GPM	9	US gallon per minute
US GPH	10	US gallon per hour
CCS	11	Cubic centimeter per second
CCM	12	Cubic centimeter per minute
cm³/h	13	Cubic centimeter per hour
m³/m	14	Cubic meter per minute
m³/h	15	Cubic meter per hour
m³/d	16	Cubic meter per day
in³/m	17	Cubic inch per minute
CFM	18	Cubic foot per minute
CFH	19	Cubic foot per hour
CFD	21	Cubic foot per day
count	62	Setpoint count, 0–64000
%	63	Percent of full scale

Appendix B-5: Total Volume Units

Unit Label	Value	Notes
	0	Unit not specified: use default values.

Unit Label	Value	Notes
---	1	Unknown unit: no conversions are performed to other units. If calibrated with different units, the value is displayed in those units.
L	2	Microliter
mL	3	Milliliter
L	4	Liter
US GAL	5	US gallon
cm^3	6	Cubic centimeter
m^3	7	Cubic meter
in^3	8	Cubic inch
ft^3	9	Cubic foot
µP	61	Micropoise, a measure of viscosity: no conversions are performed to or from other units

Appendix B-6: Pressure Units

Unit Label	Value	Notes
	0	Unit not specified: use default values.
---	1	Unknown unit: no conversions are performed to other units. If calibrated with different units, the value is displayed in those units.
Pa	2	Pascal
hPa	3	Hectopascal
kPa	4	Kilopascal
MPa	5	Megapascal
mbar	6	Millibar
bar	7	Bar
g/cm^2	8	Gram force per square centimeter
kg/cm	9	Kilogram-force per square centimeter
PSI	10	Pound-force per square inch
PSF	11	Pound-force per square foot
mTorr	12	Millitorr
torr	13	Torr
mmHg	14	Millimeter of mercury at 0 °C
inHg	15	Inches of mercury at 0 °C
mmH_2O	16	Millimeter of water at 4 °C (NIST conventional)
mmH_2O	17	Millimeter of water at 60 °F
cmH_2O	18	Centimeter of water at 4 °C (NIST conventional)
cmH_2O	19	Centimeter of water at 60 °F
inH_2O	20	Inch of water at 4 °C (NIST conventional)

Unit Label	Value	Notes
inH_2O	21	Inch of water at 60 °F
atm	22	Atmosphere (absolute pressure only)
V	61	Volt: no conversions are performed to or from other units (intended only for log-linear absolute pressure sensors)
count	62	Setpoint count, 0–64000
%	63	Percent of full scale

Appendix B-7: Temperature Units

Unit Label	Value	Notes
	0	Unit not specified: use default values.
---	1	Unknown unit: no conversions are performed to other units. If calibrated with different units, the value is displayed in those units.
°C	2	Degree Celsius
°F	3	Degree Fahrenheit
°K	4	Kelvin
°Ra	5	Degree Rankine

Appendix C: Instrument Data Sources

Use the following table to determine the data source of a given value.

Data Source	Value	Notes
Instant display	1	Current readings, smoothed for front panel display.
Instant serial	2	Current readings, smoothed for serial values.
Totalizer 2	25	Readings taken from the second totalizer.

Appendix D: Gas Numbers

#	Short Name	Long Name
0	Air	Air (Clean Dry)
1	Ar	Argon
2	CH_4	Methane
3	CO	Carbon Monoxide
4	CO_2	Carbon Dioxide
5	C_2H_6	Ethane
6	H_2	Hydrogen
7	He	Helium
8	N_2	Nitrogen
9	N_2O	Nitrous Oxide
10	Ne	Neon
11	O_2	Oxygen
12	C_3H_8	Propane
13	nC_4H_{10}	Normal Butane
14	C_2H_2	Acetylene
15	C_2H_4	Ethylene (Ethene)
16	iC_4H_{10}	Isobutane
17	Kr	Krypton
18	Xe	Xenon
19	SF_6	Sulfur Hexafluoride[1]
20	C-25	25% CO_2, 75% Ar
21	C-10	10% CO_2, 90% Ar
22	C-8	8% CO_2, 92% Ar
23	C-2	2% CO_2, 98% Ar
24	C-75	75% CO_2, 25% Ar
25	He-25	25% He, 75% Ar
26	He-75	75% He, 25% Ar
27	A1025	90% He, 7.5% Ar, 2.5% CO_2
28	Star29	Stargon CS (90% Ar, 8% CO_2, 2% O_2)
29	P-5	5% CH_4, 95% Ar
30	NO	Nitric Oxide[2]
31	NF_3	Nitrogen Trifluoride[2]
32	NH_3	Ammonia[2]
33	Cl_2	Chlorine[2]
34	H_2S	Hydrogen Sulfide[2]
35	SO_2	Sulfur Dioxide[2]
36	C_3H_6	Propylene[2]
80	1Buten	1-Butylene[2]
81	cButen	Cis-Butene (cis-2-Butene)[2]
82	iButen	Isobutene[2]
83	tButen	Trans-2-Butene[2]
84	COS	Carbonyl Sulfide[2]
85	DME	Dimethylether (C_2H_6O)[2]
86	SiH_4	Silane[2]
100	R-11	Trichlorofluoromethane (CCl_3F)[2,3]
101	R-115	Chloropentafluoroethane (C_2ClF_5)[2,3]
102	R-116	Hexafluoroethane (C_2F_6)[2]
103	R-124	Chlorotetrafluoroethane (C_2HClF_4)[2,3]
104	R-125	Pentafluoroethane (CF_3CHF_2)[2,3]
105	R-134A	Tetrafluoroethane (CH_2FCF_3)[2,3]
106	R-14	Tetrafluoromethane (CF_4)[2]
107	R-142b	Chlorodifluoroethane (CH_3CClF_2)[2,3]
108	R-143a	Trifluoroethane ($C_2H_3F_3$)[2,3]
109	R-152a	Difluoroethane ($C_2H_4F_2$)[2]
110	R-22	Difluoromonochloromethane ($CHClF_2$)[2,3]
111	R-23	Trifluoromethane (CHF_3)[2,3]
112	R-32	Difluoromethane (CH_2F_2)[2,3]
113	R-318	Octafluorocyclobutane (C_4F_8)[2]
114	R-404A	44% R-125, 4% R-134A, 52% R-143A[2,3]
115	R-407C	23% R-32, 25% R-125, 52% R-143A[2,3]
116	R-410A	50% R-32, 50% R-125[2,3]
117	R-507A	50% R-125, 50% R-143A[2,3]
140	C-15	15% CO_2, 85% Ar
141	C-20	20% CO_2, 80% Ar
142	C-50	50% CO_2, 50% Ar
143	He-50	50% He, 50% Ar
144	He-90	90% He, 10% Ar
145	Bio5M	5% CH_4, 95% CO_2
146	Bio10M	10% CH_4, 90% CO_2
147	Bio15M	15% CH_4, 85% CO_2
148	Bio20M	20% CH_4, 80% CO_2
149	Bio25M	25% CH_4, 75% CO_2
150	Bio30M	30% CH_4, 70% CO_2
151	Bio35M	35% CH_4, 65% CO_2
152	Bio40M	40% CH_4, 60% CO_2
153	Bio45M	45% CH_4, 55% CO_2
154	Bio50M	50% CH_4, 50% CO_2
155	Bio55M	55% CH_4, 45% CO_2
156	Bio60M	60% CH_4, 40% CO_2
157	Bio65M	65% CH_4, 35% CO_2
158	Bio70M	70% CH_4, 30% CO_2
159	Bio75M	75% CH_4, 25% CO_2
160	Bio80M	80% CH_4, 20% CO_2
161	Bio85M	85% CH_4, 15% CO_2
162	Bio90M	90% CH_4, 10% CO_2
163	Bio95M	95% CH_4, 5% CO_2
164	EAN-32	32% O_2, 68% N_2
165	EAN-36	36% O_2, 64% N_2
166	EAN-40	40% O_2, 60% N_2
167	HeOx20	20% O_2, 80% He
168	HeOx21	21% O_2, 79% He
169	HeOx30	30% O_2, 70% He
170	HeOx40	40% O_2, 60% He
171	HeOx50	50% O_2, 50% He
172	HeOx60	60% O_2, 40% He
173	HeOx80	80% O_2, 20% He
174	HeOx99	99% O_2, 1% He
175	EA-40	Enriched Air-40% O_2
176	EA-60	Enriched Air-60% O_2
177	EA-80	Enriched Air-80% O_2
178	Metab	Metabolic Exhalant (16% O_2, 78.04% N_2, 5% CO_2, 0.96% Ar)
179	LG-4.5	4.5% CO_2, 13.5% N_2, 82% He
180	LG-6	6% CO_2, 14% N_2, 80% He
181	LG-7	7% CO_2, 14% N_2, 79% He
182	LG-9	9% CO_2, 15% N_2, 76% He
183	HeNe-9	9% Ne, 91% He
184	LG-9.4	9.4% CO_2, 19.25% N_2, 71.35% He
185	SynG-1	40% H_2, 29% CO, 20% CO_2, 11% CH_4
186	SynG-2	64% H_2, 28% CO, 1% CO_2, 7% CH_4
187	SynG-3	70% H_2, 4% CO, 25% CO_2, 1% CH_4
188	SynG-4	83% H_2, 14% CO, 3% CH_4
189	NatG-1	93% CH_4, 3% C_2H_6, 1% C_3H_8, 2% N_2, 1% CO_2
190	NatG-2	95% CH_4, 3% C_2H_6, 1% N_2, 1% CO_2
191	NatG-3	95.2% CH_4, 2.5% C_2H_6, 0.2% C_3H_8, 0.1% C_4H_{10} 0.7% CO_2
192	CoalG	50% H_2, 35% CH_4, 10% CO, 5% C_2H_4
193	Endo	75% H_2, 25% N_2
194	HHO	66.67% H_2, 33.33% O_2
195	HD-5	LPG: 96.1% C_3H_8, 1.5% C_2H_6, 0.4% C_3H_6, 1.9% n-C_4H_{10}
196	HD-10	LPG: 85% C_3H_8, 10% C_3H_6, 5% n-C_4H_{10}
197	OCG-89	89% O_2, 7% N_2, 4% Ar
198	OCG-93	93% O_2, 3% N_2, 4% Ar
199	OCG-95	95% O_2, 1% N_2, 4% Ar
200	FG-1	2.5% O_2, 10.8% CO_2, 85.7% N_2, 1% Ar
201	FG-2	2.9% O_2, 14% CO_2, 82.1% N_2, 1% Ar
202	FG-3	3.7% O_2, 15% CO_2, 80.3% N_2, 1% Ar
203	FG-4	7% O_2, 12% CO_2, 80% N_2, 1% Ar
204	FG-5	10% O_2, 9.5% CO_2, 79.5% N_2, 1% Ar
205	FG-6	13% O_2, 7% CO_2, 79% N_2, 1% Ar
206	P-10	10% CH_4 90% Ar
210	D-2	Deuterium

1 Sulfur hexafluoride is a highly potent greenhouse gas monitored under the Kyoto Protocol.

2 Corrosive-resistant units only

3 Under the Montreal Protocol and Kigali Amendment, the production and consumption of these ozone-depleting substances (ODS) is being or has been phased out. It is recommended you ensure compliance with this universally ratified treaty before using these gases, in addition to R113, R-123, and R-141b.