ANALYT MESSTECHNIK




COMMUNICATION GUIDE

# 358 Series EtherNet/IP

Network set-up, assemblies, CIP objects, commands and PLC examples for reading and controlling 358 Series flow and pressure devices over EtherNet/IP.

AMTC-CG-0004

# About this guide

This guide helps you connect, configure and use a 358 Series device ordered with EtherNet/IP. It covers flow meters and controllers, pressure gauges and controllers that carry the EtherNet/IP interface. It contains the network set-up, the CIP objects and assemblies the device supports, every command with its arguments and results, the set-up of a PLC, and the status and command error codes.

Each command states the firmware version that introduced it; a device with older firmware does not have it. The firmware version of a device is shown in its embedded web server (Home page (p. 10)) and is returned in assembly 111 (p. 17).

If you are connecting a device for the first time, start with Interface and wiring (p. 7). If you already know EtherNet/IP and only need the data layout, go straight to the Command and register reference (p. 14).

The serial commands and the Modbus registers of the 358 Series are described in two companion guides: *358 Series serial interface (RS-232/RS-485) — Communication guide* (AMTC-CG-0002) and *358 Series Modbus RTU and Modbus TCP/IP — Communication guide* (AMTC-CG-0001). If your application needs something that is not covered here, the Analyt-MTC application engineers will be glad to help.

Conventions in this guide:

- Byte offsets, assembly numbers, command IDs and argument values are decimal. Hexadecimal values carry the prefix `0x` (for example class `0x37`), or `16#` where a PLC uses that notation.
- Data types are the CIP elementary types: `UINT`, `UDINT`, `DINT`, `REAL` and the others listed in Data types (p. 12).
- Status names, PLC tags and values you enter exactly as shown are set in monospace type, for example `SUCCESS`.
- Menu paths on the device display are written *MENU* ▯ *Setup* ▯ *Advanced* ▯ *Comm Status*.

## Abbreviations

| Abbreviation | Meaning |
|---|---|
| **ADC** | Analogue-to-digital converter |
| **AOI** | Add-On Instruction (a reusable instruction in Logix PLC software) |
| **BOOTP** | Bootstrap Protocol |
| **CIP** | Common Industrial Protocol, the application layer of EtherNet/IP |
| **DHCP** | Dynamic Host Configuration Protocol |
| **EDS** | Electronic data sheet, the device description file of an EtherNet/IP device |
| **EIP** | EtherNet/IP (Ethernet Industrial Protocol) |
| **ID** | Identifier |
| **IEEE 754** | Standard for floating-point numbers; `REAL` values use its 32-bit single-precision format |
| **I/O** | Input/output |
| **IP** | Internet Protocol |
| **LED** | Light-emitting diode |
| **MAC** | Media access control (the physical address of a network port) |

| Abbreviation | Meaning |
|---|---|
| **ODVA** | The organisation that manages the EtherNet/IP specification and certifies devices |
| **PD2I, PDF** | The two closed-loop control algorithms of the 358 Series controllers |
| **PID** | Proportional-integral-derivative (closed-loop control) |
| **PLC** | Programmable logic controller |
| **UDT** | User-defined data type (Logix PLC software) |

## Safety notices in this guide

**NOTICE**

Practices not related to personal injury — damage to the product, the process or the sample.

## Contact and support


**ANALYT-MTC Messtechnik GmbH**

**German Headquarters**

Klosterrunsstraße 18, 79379 Müllheim
Germany

+49 7631 5545

info@analyt-mtc.de

**French Subsidiary**

175 Avenue d'Alsace, 68000 Colmar
France

+33 3 89 41 47 78

info@analyt-mtc.fr

www.analyt-mtc.com


**Before you call**

Have the following at hand so that Analyt-MTC can help you quickly:

- the model and serial number of the device (on its label);
- the device firmware and adapter firmware versions shown on the *Home* page of its web server;
- the assembly or command you used, with the command ID, the argument and the status you received.

## Document information

| | |
|---|---|
| Title | 358 Series EtherNet/IP — Communication guide |
| Document number | AMTC-CG-0004 |
| Revision | Rev. 1 — October 2026 |
| Applies to | 358 Series flow meters and controllers, pressure gauges and controllers with EtherNet/IP |
| Language | English |
| Related documents | 358 Series Modbus RTU and Modbus TCP/IP — Communication guide (AMTC-CG-0001); 358 Series serial interface (RS-232/RS-485) — Communication guide (AMTC-CG-0002); 358 Series M, MB, MLW and MLWB Mass Flow Meters — Operating manual (AMTC-MN-0004); 358 Series MC Mass Flow Controllers — Operating manual (AMTC-MN-0005); 358 Series L Liquid Flow Meters — Operating manual (AMTC-MN-0002); 358 Series LC Liquid Flow Controllers — Operating manual (AMTC-MN-0003); 358 Series P Pressure Transducers — Operating manual (AMTC-MN-0006); 358 Series PC Pressure Controllers — Operating manual (AMTC-MN-0007); 358 Series Dual-Valve Pressure Controllers — Operating manual (AMTC-MN-0001) |

## Revision history

| Rev. | Date | Change |
|---|---|---|
| 1 | October 2026 | First issue. |

Information in this guide is subject to change without notice. © 2026 ANALYT-MTC Messtechnik GmbH. All rights reserved.

# Contents


1 **Interface and wiring** 7

Devices with EtherNet/IP 7

Ethernet ports and LEDs 7

IP66/67 version: M12 Ethernet and M8 power ports 8

Network topology 8

Setting the IP address 9

Comm Status screen 9

Assigning a static IP address with the web server 9

Static IP address by other methods 9

Embedded web server 10

Home page 10

Data I/O page 10

Device Config page 11

2 **Protocol conventions** 12

How the device communicates 12

Real and integer formats 12

Data types 12

Byte offsets and the size of assembly 101 13

Readings, statistics and units 13

Network access and settings 13

3 **Command and register reference** 14

Supported objects 14

Assemblies 14

Real format assemblies 15

Integer format assemblies 16

Information assemblies 17

Command assemblies 21

Commands 22

Sending a command 22

Command status 22

Command list 23

Command function 24

Data readings 24

Control 27

Administrative 32

4 **Examples** 37

Rockwell PLC set-up 37

Formatting the I/O data 38

Changing the set point 40

Example: sending a command 40
Worked argument values 40

5 **Error handling** 42
Symptoms and remedies 42
Command status codes 42

A **Statistics** 44

B **Engineering units** 47

C **Data sources** 50

D **Gas numbers** 51

CHAPTER 1

# Interface and wiring

## Devices with EtherNet/IP

A 358 Series device with EtherNet/IP has two RJ45 Ethernet ports on the top of its body. In an EtherNet/IP system it works as a communications adapter (device type 12), and it is certified by ODVA. It works with any PLC that supports EtherNet/IP, and it interoperates with other EtherNet/IP devices on the same network.

To connect the device to your PLC, you need its electronic data sheet (EDS): use the device description file supplied with the instrument. The device also holds this file itself in its CIP File object (Supported objects (p. 14)).

## Ethernet ports and LEDs

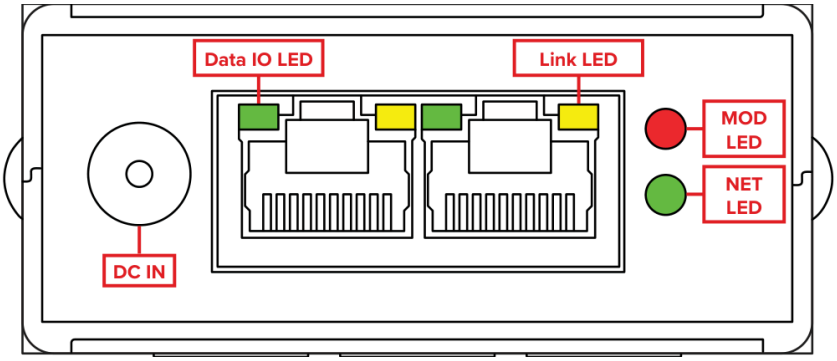



**Figure 1.** Connector panel: two RJ45 ports with their Data IO LED (green) and Link LED (yellow), the MOD and NET LEDs, and the DC IN power socket.

Both RJ45 ports are 10/100 Mbit/s Ethernet ports with a switch built into the device. You can use either port to communicate with the device, or to chain it to further devices.

Each port has two LEDs. The yellow Link LED is lit steadily while the port has an active connection and goes out only when there is none. The green Data IO LED flashes as the device receives or sends data.

To the right of the ports, with the display or label and the flow direction arrow facing you, are two status LEDs. The far one, nearer the back of the device, is the MOD LED: it shows the status of the device itself and any fault within it. The near one, nearer the front, is the NET LED: it shows the state of the EtherNet/IP connection. Both can light red or green.

### 1-1 MOD and NET LEDs

| LED state | MOD LED | NET LED |
|---|---|---|
| Off | Device not powered | No IP address assigned |
| Steady green | Device operational | EtherNet/IP connection established |
| Steady red | Major system fault | Duplicate IP address (conflict) |
| Flashing red | Minor system fault | Connection timed out |
| Flashing red and green | Self-test | Self-test |

## IP66/67 version: M12 Ethernet and M8 power ports

On the IP66/67 version, the device has two 4-pin M12 Ethernet sockets on the top and a 4-pin M8 power plug on the side. Always identify the right wires by a continuity check and by colour when you use multi-strand cables. The calibration certificate of the device is the definitive reference for custom wiring.

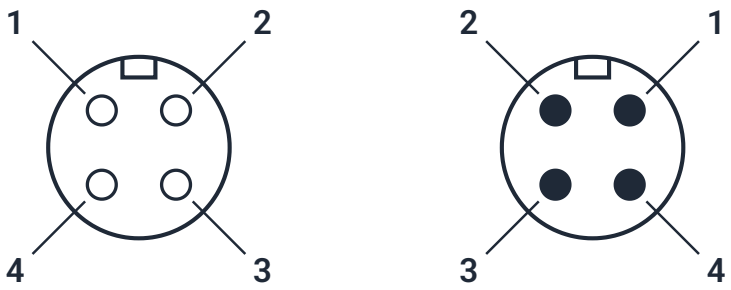



**Figure 2.** 4-pin M12 Ethernet, on the top of the device (two ports): female socket on the device (left), male plug on the cable (right).

| Pin | Signal | Function |
|---|---|---|
| **1** | Tx+ | Send data + |
| **2** | Rx+ | Receive data + |
| **3** | Tx− | Send data − |
| **4** | Rx− | Receive data − |

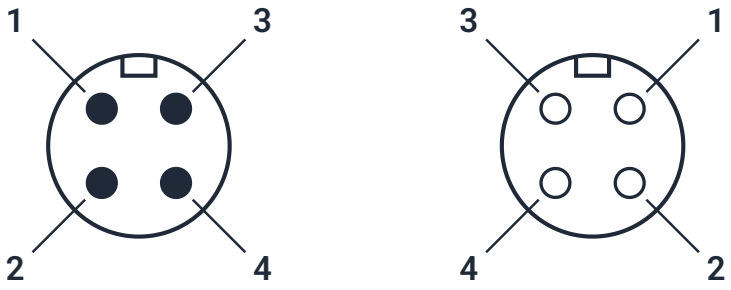



**Figure 3.** 4-pin M8 power, on the side of the device: male plug on the device (left), female socket on the cable (right).

| Pin | Function |
|---|---|
| **1** | Reserved. |
| **2** | Power in: powers the device; see the spec sheet for the supply values. |
| **3** | Reserved. |
| **4** | Ground: common ground for power, digital communication, analogue signals and alarms. |

## Network topology

The built-in switch lets you build the network in the way that suits your plant:

- **Star:** connect either port to a central switch.
- **Line (daisy chain) or ring:** use one port as the input and the other as the output, and chain the devices one after another.

## Setting the IP address

The device addresses on the network are its IP settings; an EtherNet/IP device has no baud rate or bus address to set. By default the device obtains its IP address from a DHCP server. If the device loses power, the address may change when it is powered on again.

To find the address, choose *MENU* → *Setup* → *Advanced* → *Comm Status* on a device with a display. For a device without a display, ask your network administrator for the address the DHCP server assigned.

You can give the device a static IP address through its embedded web server (below) or through the TCP/IP object of CIP. Before you change any IP address, ask your network administrator which addresses are free.

### Comm Status screen

Devices with a display have a communication status screen, which the device operating manual does not describe. Open it with *MENU* → *Setup* → *Advanced* → *Comm Status*. It shows whether the link is up, the current IP address of the device and whether a PLC is connected, which helps you diagnose communication problems.

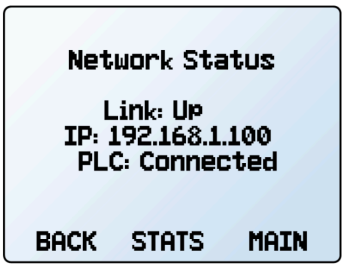



**Figure 4.** Comm Status screen: link up, IP address 192.168.1.100, PLC connected.

### Assigning a static IP address with the web server

You need the address the device has now, and a computer on the same network with a web browser.

1. In the browser, go to the current IP address of the device. The web server opens on its *Home* page.
2. Click *Network Config* on the left of the page. By default *Use DHCP* is ticked and the fields *IP Address*, *Subnet Mask* and *Gateway* are locked.
3. Clear the *Use DHCP* check box. DHCP is switched off and the fields are unlocked.
4. Enter the IP address, subnet mask and gateway you want, and click *Update*.
5. The connection to the web server is lost when the network settings change. Go to the new IP address in the browser to reconnect.

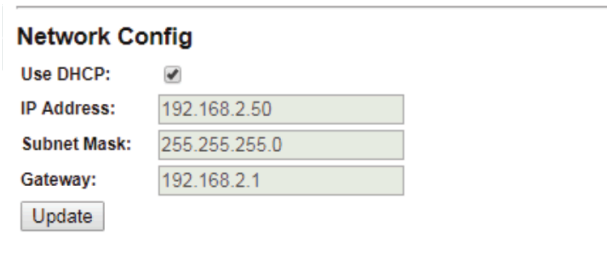



**Figure 5.** *Network Config* page of the web server, with DHCP in use.

### Static IP address by other methods

Other software and automation tools can also assign the IP address, for example the BOOTP/DHCP server of Rockwell or the EIP Tools of Molex. Refer to the documentation of the program you choose.

# Embedded web server

Every 358 Series device with EtherNet/IP has a web server that shows device information and lets you change settings. Open it by going to the IP address of the device in the web browser of your computer. Its pages are listed on the left: *Home*, *Data I/O*, *Network Config* and *Device Config*.

## Home page

The first page is for information only. It shows the device and the state of its network connection in two sections.

**Device Information** shows the protocol, the firmware and the serial number:

- **Protocol:** the communication protocol the device is configured for. It should read *EtherNet/IP*. If it does not, contact Analyt-MTC support.
- **Device FW Version:** the firmware installed in the device. It determines how the device works, its display menu and the commands available.
- **Adapter FW Version:** the firmware of the Ethernet adapter, used for troubleshooting.
- **Device Serial Num:** the serial number of the device. It never changes and is the number on the label on the back of the device.

**Network Status** shows the network information the device has now. You change these settings on the *Network Config* page (Assigning a static IP address with the web server (p. 9)).

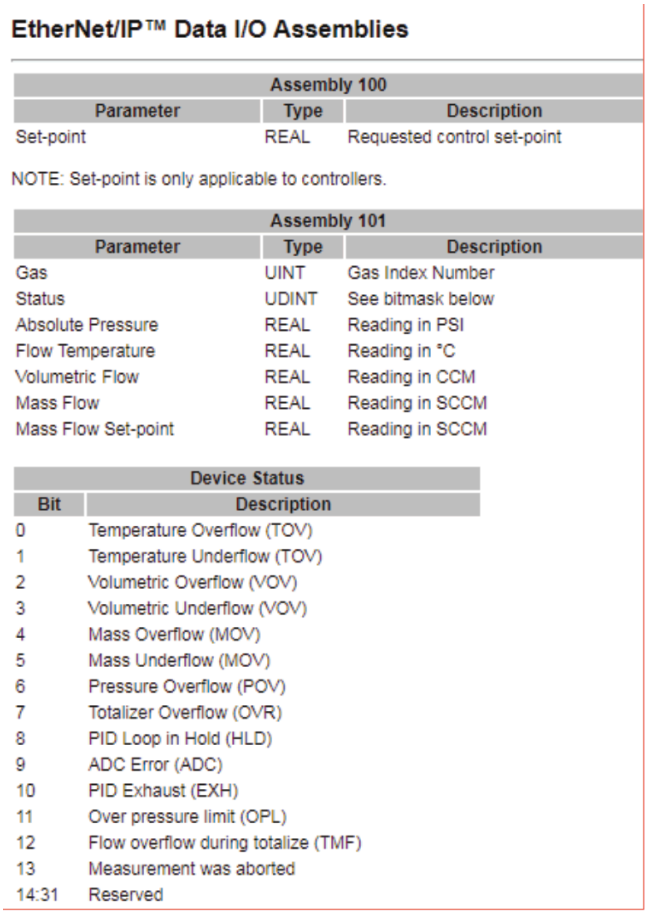
- **MAC Address:** the physical address of the device, also printed on a label on the device.
- **Address Mode:** how the device obtains its IP address, *DHCP* or *Static*.
- **IP Address:** the IP address of the device. It changes only if the address mode is *DHCP*.
- **Subnet Mask:** the subnet mask of the network the device is connected to, set by the network.
- **Gateway:** the gateway address the device is connected to, set by the network.

## Data I/O page

The *Data I/O* page lists the structure of the I/O assemblies and the device status bits. It reflects the configuration of this device, so it differs from device to device.

When you set up a PLC or other program to read the device, use this page to find the size and structure of assembly 101 (Determining the input size of assembly 101 (p. 37)). Parameters of type `REAL` or `UDINT` take 4 bytes, parameters of type `UINT` 2 bytes. Add up the sizes of all parameters of assembly 101 to obtain its size.

The device status table gives the bit of each error or status the device can report. The letters in brackets can appear on the display of the device, if it has one, to show that the condition is present. The bits are listed in Device status (p. 17).



**EtherNet/IP™ Data I/O Assemblies**

| Assembly 100 | | |
|---|---|---|
| **Parameter** | **Type** | **Description** |
| Set-point | REAL | Requested control set-point |

NOTE: Set-point is only applicable to controllers.

| Assembly 101 | | |
|---|---|---|
| **Parameter** | **Type** | **Description** |
| Gas | UINT | Gas Index Number |
| Status | UDINT | See bitmask below |
| Absolute Pressure | REAL | Reading in PSI |
| Flow Temperature | REAL | Reading in °C |
| Volumetric Flow | REAL | Reading in CCM |
| Mass Flow | REAL | Reading in SCCM |
| Mass Flow Set-point | REAL | Reading in SCCM |

| Device Status | |
|---|---|
| **Bit** | **Description** |
| 0 | Temperature Overflow (TOV) |
| 1 | Temperature Underflow (TOV) |
| 2 | Volumetric Overflow (VOV) |
| 3 | Volumetric Underflow (VOV) |
| 4 | Mass Overflow (MOV) |
| 5 | Mass Underflow (MOV) |
| 6 | Pressure Overflow (POV) |
| 7 | Totalizer Overflow (OVR) |
| 8 | PID Loop in Hold (HLD) |
| 9 | ADC Error (ADC) |
| 10 | PID Exhaust (EXH) |
| 11 | Over pressure limit (OPL) |
| 12 | Flow overflow during totalize (TMF) |
| 13 | Measurement was aborted |
| 14:31 | Reserved |

**Figure 6.** *Data I/O* page of a mass flow controller: assembly 100, assembly 101 and the device status bits.

## Device Config page

The *Device Config* page is used mainly for troubleshooting. It reads and writes the device registers that hold its configuration settings.

**NOTICE**

Writing a configuration register can invalidate the calibration of the device, disable its communication or damage it permanently. Do not write a register unless you know exactly how that register works.




**Figure 7.** *Device Config* page: register number, value, *Read* and *Write*.

CHAPTER 2

# Protocol conventions

## How the device communicates

In an EtherNet/IP network the 358 Series device is a communications adapter (device type 12). Your PLC or other scanner exchanges data with it in two ways, and you can use both:

- **Cyclic I/O data** (implicit messaging): the PLC and the device exchange assemblies cyclically, for example the set point and the readings.
- **CIP explicit messages:** the PLC reads or writes one attribute of an object when it needs to, for example to send a command or read the reading information.

Every object supports the services `Get_Attribute_Single` and `Set_Attribute_Single`, in addition to any other services it offers. The objects and assemblies are listed in Chapter 3 (p. 14).

## Real and integer formats

The set point and the readings are offered in two formats, in separate assemblies:

- **Real format** (assemblies 100, 101 and 107): each value is a `REAL`, a 32-bit IEEE 754 floating-point number in the engineering units of the reading. Send the set point in this format to assembly 100.
- **Integer format** (assemblies 105, 106 and 108): each value is a `DINT` with an implied decimal point. Multiply it by $10^{-n}$, where *n* is the number of decimal places of that reading given in the information assembly 111 or 112, to obtain the value in engineering units.

Both formats can be used in an implicit or an explicit connection. The information assemblies 111 and 112 can only be read with explicit messages.

### Example: scaling an integer reading

| Value read | Decimal places | Value in engineering units |
|---|---|---|
| 12345 | 2 | 123.45 |

## Data types

The tables of this guide use these CIP data types.

| Type | Size | Contents |
|---|---|---|
| `BOOL` | — | Flag, in a bit overlay |
| `SINT` | — | Signed 8-bit integer |
| `INT` | — | Signed 16-bit integer |
| `UINT` | 2 bytes | Unsigned 16-bit integer |
| `WORD` | — | 16-bit bit string |
| `DINT` | 4 bytes | Signed 32-bit integer (`Integer32`) |
| `UDINT` | 4 bytes | Unsigned 32-bit integer (`Unsigned32`) |
| `REAL` | 4 bytes | 32-bit IEEE 754 single-precision floating-point number |
| `STRING`, `STRUCT`, `ARRAY` | — | Text, structure and array, as given for each attribute |

## Byte offsets and the size of assembly 101

The assembly tables give each value by its byte offset from the start of the assembly. A value of type `UINT` takes 2 bytes; `UDINT`, `DINT` and `REAL` take 4 bytes.

Assemblies 101 and 106 hold up to 20 readings, depending on the configuration of the device, so their size differs from device to device. A PLC that opens an I/O connection must use the exact input size of assembly 101; see Determining the input size of assembly 101 (p. 37).

## Readings, statistics and units

A device can measure up to 20 data statistics, depending on its configuration. The information assemblies 111 and 112, and the data reading commands, return numeric codes: the statistic of a reading (Appendix A (p. 44)), its engineering units (Appendix B (p. 47)) and its data source (Appendix C (p. 50)). Gas numbers are in Appendix D (p. 51). A value of 0 means that the reading is not available on the device.

## Network access and settings

The device answers on the network at its IP address. Its embedded web server opens in any web browser that reaches that address, and its *Network Config* and *Device Config* pages change the IP settings and write configuration registers (Embedded web server (p. 10)). Before you change any IP address, ask your network administrator which addresses are free.

Settings that the device keeps across power cycles, such as a saved maximum ramp rate or the power-up set point, are listed with the commands in Chapter 3 (p. 14); Restore factory settings (p. 36) returns all settings and the configuration to their state when the device left the factory.

CHAPTER 3

# Command and register reference

The CIP objects and assemblies a 358 Series device supports over EtherNet/IP, and every command it accepts with the firmware version that introduced it.

## Supported objects

All objects support the services `Get_Attribute_Single` and `Set_Attribute_Single`, in addition to any other services. The device description file (EDS) can be read from the File object, class `0x37`, instance 200.

### 3-1 Identity object, class 1, instance 1

| Attribute | Name | Type | Access | Value or comment |
|---|---|---|---|---|
| **1** | Vendor ID | `UINT` | Get | As in the device description file supplied with the instrument |
| **2** | Device Type | `UINT` | Get | 12 (communications adapter) |
| **3** | Product Code | `UINT` | Get | As in the device description file supplied with the instrument |
| **4** | Revision | `STRUCT` | Get | Major.Minor |
| **5** | Status | `WORD` | Get | — |
| **6** | Serial Number | `UDINT` | Get | — |
| **7** | Product Name | `STRING` | Get | As in the device description file supplied with the instrument |

### 3-2 Assembly object, class 4

| Attribute | Name | Type | Access | Value or comment |
|---|---|---|---|---|
| **3** | Data | `ARRAY` | Get/Set | The data of the instance; see the assembly descriptions below. |
| **4** | Number Available | `UINT` | Get | Size of attribute 3. |

## Assemblies

The assemblies are instances of the Assembly object. The real and integer format assemblies can be used in an implicit or an explicit connection; the information assemblies only with explicit messages.

### 3-3 Assembly instances

| Instance | Contents | Data types |
|---|---|---|
| Assembly 100 (p. 15) | Set point request | `REAL` |
| Assembly 101 (p. 15) | Optimized readings | `REAL` |
| Assembly 102 (p. 21) | Limited command request | `UINT` |
| Assembly 103 (p. 21) | Limited command result | `UINT` |
| Assembly 104 (p. 21) | Gas mix configuration | `UINT` |
| Assembly 105 (p. 16) | Set point request | `DINT` |
| Assembly 106 (p. 16) | Optimized readings | `DINT` |
| Assembly 107 (p. 15) | Standard readings | `REAL` |
| Assembly 108 (p. 16) | Standard readings | `DINT` |
| Assembly 109 (p. 21) | Command request | `UDINT, DINT` |

| Instance | Contents | Data types |
|---|---|---|
| Assembly 110 (p. 21) | Command result | UDINT, DINT |
| Assembly 111 (p. 17) | Standard readings information | UINT, DINT |
| Assembly 112 (p. 20) | Optimized readings information | UINT, DINT |

# Real format assemblies

## Assembly 100: set point request (real)

Assembly 100 sets the set point of a controller: the flow or pressure the controller tries to hold while the process is flowing. Send the set point as a 32-bit IEEE floating-point value. Meters and gauges have no valve to control the process and ignore assembly 100.

| Byte offset | Type | Description |
|---|---|---|
| 0 | REAL | Requested set point |

## Assembly 107: standard readings (real)

Assembly 107 gives the readings of the device as REAL values, always at the same byte offsets.

| Byte offset | Type | Description |
|---|---|---|
| 0 | UINT | Gas number |
| 2 | UINT | Alarm outputs state |
| 4 | UDINT | Device status (table 3-4) |
| 8 | REAL | Current set point. |
| 12 | REAL | Current valve drive. |
| 16 | REAL | Current pressure. |
| 20 | REAL | Current secondary pressure, if any. |
| 24 | REAL | Current barometric pressure. |
| 28 | REAL | Current temperature. |
| 32 | REAL | Current volumetric flow. |
| 36 | REAL | Current mass flow. |
| 40 | REAL | Current totaliser 1. |
| 44 | REAL | Current totaliser 2. |
| 48 | REAL | Current humidity. |

## Assembly 101: optimized readings (real)

A device can measure up to 20 data statistics, depending on its configuration. Assembly 101 packs the readings this device has, as REAL values; the *Data I/O* page of the web server lists them (Data I/O page (p. 10)). Assembly 112 tells you what each reading is.

| Byte offset | Type | Description |
|---|---|---|
| 0 | UINT | Gas number |
| 2 | UDINT | Device status (table 3-4) |
| 6 | REAL | Reading 1 |
| 10 | REAL | Reading 2 |
| 14 | REAL | Reading 3 |
| … | … | Readings 4 to 19 follow the same pattern, 4 bytes each. |

| Byte offset | Type | Description |
|---|---|---|
| **82** | REAL | Reading 20 |

# Integer format assemblies

## Assembly 105: set point request (integer)

Assembly 105 holds the requested set point of a controller in integer format. Meters and gauges ignore it.

| Byte offset | Type | Description |
|---|---|---|
| **0** | DINT | Requested set point. Multiply by $10^{-n}$, n = value at offset 22 of assembly 111, for the value in engineering units. |

## Assembly 108: standard readings (integer)

Assembly 108 gives the readings of the device in integer format, always at the same byte offsets, with the scaling of each value.

| Byte offset | Type | Description |
|---|---|---|
| **0** | UINT | Gas number |
| **2** | UINT | Alarm outputs state |
| **4** | UDINT | Device status (table 3-4) |
| **8** | DINT | Current set point. Multiply by $10^{-n}$, n = value at offset 22 of assembly 111, for the value in engineering units. |
| **12** | DINT | Current valve drive. Multiply by $10^{-n}$, n = value at offset 38 of assembly 111, for the value in engineering units. |
| **16** | DINT | Current pressure. Multiply by $10^{-n}$, n = value at offset 54 of assembly 111, for the value in engineering units. |
| **20** | DINT | Current secondary pressure, if any. It may come from a second pressure sensor or be calculated with a barometer. Multiply by $10^{-n}$, n = value at offset 70 of assembly 111, for the value in engineering units. |
| **24** | DINT | Current barometric pressure. Multiply by $10^{-n}$, n = value at offset 86 of assembly 111, for the value in engineering units. |
| **28** | DINT | Current temperature. Multiply by $10^{-n}$, n = value at offset 102 of assembly 111, for the value in engineering units. |
| **32** | DINT | Current volumetric flow. Multiply by $10^{-n}$, n = value at offset 118 of assembly 111, for the value in engineering units. |
| **36** | DINT | Current mass flow. Multiply by $10^{-n}$, n = value at offset 134 of assembly 111, for the value in engineering units. |
| **40** | DINT | Current totaliser 1. Multiply by $10^{-n}$, n = value at offset 150 of assembly 111, for the value in engineering units. |
| **44** | DINT | Current totaliser 2. Multiply by $10^{-n}$, n = value at offset 166 of assembly 111, for the value in engineering units. |
| **48** | DINT | Current humidity. Multiply by $10^{-n}$, n = value at offset 182 of assembly 111, for the value in engineering units. |

## Assembly 106: optimized readings (integer)

Assembly 106 packs the up to 20 readings of the device in integer format. Assembly 112 tells you what each reading is and gives its decimal places.

| Byte offset | Type | Description |
|---|---|---|
| **0** | UINT | Gas number |

| Byte offset | Type | Description |
|---|---|---|
| **2** | `UINT` | Alarm outputs state |
| **4** | `UDINT` | Device status (table 3-4) |
| **8** | `DINT` | Reading 1. Multiply by $10^{-n}$, n = value at offset 14 of assembly 112, for the value in engineering units. |
| **12** | `DINT` | Reading 2. Multiply by $10^{-n}$, n = value at offset 30 of assembly 112, for the value in engineering units. |
| **16** | `DINT` | Reading 3. Multiply by $10^{-n}$, n = value at offset 46 of assembly 112, for the value in engineering units. |
| … | … | Readings 4 to 19 follow the same pattern, 4 bytes each. |
| **84** | `DINT` | Reading 20. Multiply by $10^{-n}$, n = value at offset 318 of assembly 112, for the value in engineering units. |

### Device status

The device status in assemblies 101, 106, 107 and 108 is a bit field. The letters in brackets can appear on the display of the device. The *Data I/O* page of the web server shows the same table.

#### 3-4 Device status bits

| Bit | Description |
|---|---|
| **0** | Temperature overflow (TOV) |
| **1** | Temperature underflow (TOV) |
| **2** | Volumetric overflow (VOV) |
| **3** | Volumetric underflow (VOV) |
| **4** | Mass overflow (MOV) |
| **5** | Mass underflow (MOV) |
| **6** | Pressure overflow (POV) |
| **7** | Totaliser overflow (OVR) |
| **8** | PID loop in hold (HLD) |
| **9** | ADC error (ADC) |
| **10** | PID exhaust (EXH) |
| **11** | Over pressure limit (OPL) |
| **12** | Flow overflow during totalising (TMF) |
| **13** | Measurement was aborted |
| **14–31** | Reserved |

## Information assemblies

The information assemblies give the firmware version and, for each reading, its statistic, source, minimum and maximum, engineering units and decimal places. Read them with explicit messages only.

### Assembly 111: standard readings information

Assembly 111 describes the readings of assemblies 107 and 108.

| Byte offset | Type | Description |
|---|---|---|
| **0** | `UINT` | Major firmware release installed. Example: the 10 in 10v05.0. |
| **2** | `UINT` | Minor firmware release installed. Example: the 05 in 10v05.0. |
| **4** | `UINT` | Custom firmware installed, if any. Example: the .0 in 10v05.0. |
| **6** | `UINT` | Reserved for the manufacturer. Always 0. |

| Byte offset | Type | Description |
|---|---|---|
| 8 | UINT | Set point type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| 10 | UINT | Set point source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| 12 | DINT | Minimum value of the set point, without overrange. Multiply by $10^{-n}$, n = value at offset 22, for the value in engineering units. |
| 16 | DINT | Maximum value of the set point, without overrange. Multiply by $10^{-n}$, n = value at offset 22, for the value in engineering units. |
| 20 | UINT | Set point units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| 22 | UINT | Set point decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| 24 | UINT | Valve drive type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| 26 | UINT | Valve drive source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| 28 | DINT | Minimum value of the valve drive, without overrange. Multiply by $10^{-n}$, n = value at offset 38, for the value in engineering units. |
| 32 | DINT | Maximum value of the valve drive, without overrange. Multiply by $10^{-n}$, n = value at offset 38, for the value in engineering units. |
| 36 | UINT | Valve drive units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| 38 | UINT | Valve drive decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| 40 | UINT | Pressure type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| 42 | UINT | Pressure source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| 44 | DINT | Minimum value of the pressure, without overrange. Multiply by $10^{-n}$, n = value at offset 54, for the value in engineering units. |
| 48 | DINT | Maximum value of the pressure, without overrange. Multiply by $10^{-n}$, n = value at offset 54, for the value in engineering units. |
| 52 | UINT | Pressure units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| 54 | UINT | Pressure decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| 56 | UINT | Secondary pressure type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| 58 | UINT | Secondary pressure source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| 60 | DINT | Minimum value of the secondary pressure, without overrange. Multiply by $10^{-n}$, n = value at offset 70, for the value in engineering units. |
| 64 | DINT | Maximum value of the secondary pressure, without overrange. Multiply by $10^{-n}$, n = value at offset 70, for the value in engineering units. |
| 68 | UINT | Secondary pressure units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| 70 | UINT | Secondary pressure decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| 72 | UINT | Barometric pressure type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| 74 | UINT | Barometric pressure source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| 76 | DINT | Minimum value of the barometric pressure, without overrange. Multiply by $10^{-n}$, n = value at offset 86, for the value in engineering units. |
| 80 | DINT | Maximum value of the barometric pressure, without overrange. Multiply by $10^{-n}$, n = value at offset 86, for the value in engineering units. |
| 84 | UINT | Barometric pressure units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| 86 | UINT | Barometric pressure decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| 88 | UINT | Temperature type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| 90 | UINT | Temperature source: a value from Appendix C (p. 50), or 0 if the reading is not available. |

| Byte offset | Type | Description |
|---|---|---|
| **92** | DINT | Minimum value of the temperature, without overrange. Multiply by $10^{-n}$, n = value at offset 102, for the value in engineering units. |
| **96** | DINT | Maximum value of the temperature, without overrange. Multiply by $10^{-n}$, n = value at offset 102, for the value in engineering units. |
| **100** | UINT | Temperature units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **102** | UINT | Temperature decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **104** | UINT | Volumetric flow type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **106** | UINT | Volumetric flow source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **108** | DINT | Minimum value of the volumetric flow, without overrange. Multiply by $10^{-n}$, n = value at offset 118, for the value in engineering units. |
| **112** | DINT | Maximum value of the volumetric flow, without overrange. Multiply by $10^{-n}$, n = value at offset 118, for the value in engineering units. |
| **116** | UINT | Volumetric flow units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **118** | UINT | Volumetric flow decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **120** | UINT | Mass flow type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **122** | UINT | Mass flow source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **124** | DINT | Minimum value of the mass flow, without overrange. Multiply by $10^{-n}$, n = value at offset 134, for the value in engineering units. |
| **128** | DINT | Maximum value of the mass flow, without overrange. Multiply by $10^{-n}$, n = value at offset 134, for the value in engineering units. |
| **132** | UINT | Mass flow units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **134** | UINT | Mass flow decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **136** | UINT | Totaliser 1 type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **138** | UINT | Totaliser 1 source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **140** | DINT | Minimum value of the totaliser 1, without overrange. Multiply by $10^{-n}$, n = value at offset 150, for the value in engineering units. |
| **144** | DINT | Maximum value of the totaliser 1, without overrange. Multiply by $10^{-n}$, n = value at offset 150, for the value in engineering units. |
| **148** | UINT | Totaliser 1 units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **150** | UINT | Totaliser 1 decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **152** | UINT | Totaliser 2 type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **154** | UINT | Totaliser 2 source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **156** | DINT | Minimum value of the totaliser 2, without overrange. Multiply by $10^{-n}$, n = value at offset 166, for the value in engineering units. |
| **160** | DINT | Maximum value of the totaliser 2, without overrange. Multiply by $10^{-n}$, n = value at offset 166, for the value in engineering units. |
| **164** | UINT | Totaliser 2 units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **166** | UINT | Totaliser 2 decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **168** | UINT | Humidity type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **170** | UINT | Humidity source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **172** | DINT | Minimum value of the humidity, without overrange. Multiply by $10^{-n}$, n = value at offset 182, for the value in engineering units. |

| Byte offset | Type | Description |
|---|---|---|
| **176** | DINT | Maximum value of the humidity, without overrange. Multiply by $10^{-n}$, n = value at offset 182, for the value in engineering units. |
| **180** | UINT | Humidity units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **182** | UINT | Humidity decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |

## Assembly 112: optimized readings information

Assembly 112 describes the readings of assemblies 101 and 106: 16 bytes per reading, for up to 20 readings.

| Byte offset | Type | Description |
|---|---|---|
| **0** | UINT | Reading 1 type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **2** | UINT | Reading 1 source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **4** | DINT | Minimum value of the reading 1, without overrange. Multiply by $10^{-n}$, n = value at offset 14, for the value in engineering units. |
| **8** | DINT | Maximum value of the reading 1, without overrange. Multiply by $10^{-n}$, n = value at offset 14, for the value in engineering units. |
| **12** | UINT | Reading 1 units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **14** | UINT | Reading 1 decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **16** | UINT | Reading 2 type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **18** | UINT | Reading 2 source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **20** | DINT | Minimum value of the reading 2, without overrange. Multiply by $10^{-n}$, n = value at offset 30, for the value in engineering units. |
| **24** | DINT | Maximum value of the reading 2, without overrange. Multiply by $10^{-n}$, n = value at offset 30, for the value in engineering units. |
| **28** | UINT | Reading 2 units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **30** | UINT | Reading 2 decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| **32** | UINT | Reading 3 type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **34** | UINT | Reading 3 source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **36** | DINT | Minimum value of the reading 3, without overrange. Multiply by $10^{-n}$, n = value at offset 46, for the value in engineering units. |
| **40** | DINT | Maximum value of the reading 3, without overrange. Multiply by $10^{-n}$, n = value at offset 46, for the value in engineering units. |
| **44** | UINT | Reading 3 units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **46** | UINT | Reading 3 decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |
| … | … | Readings 4 to 19 follow the same pattern, 16 bytes each. |
| **304** | UINT | Reading 20 type: a statistic value from Appendix A (p. 44), or 0 if the reading is not available. |
| **306** | UINT | Reading 20 source: a value from Appendix C (p. 50), or 0 if the reading is not available. |
| **308** | DINT | Minimum value of the reading 20, without overrange. Multiply by $10^{-n}$, n = value at offset 318, for the value in engineering units. |
| **312** | DINT | Maximum value of the reading 20, without overrange. Multiply by $10^{-n}$, n = value at offset 318, for the value in engineering units. |
| **316** | UINT | Reading 20 units: a value from Appendix B (p. 47), or 0 if the reading is not available. |
| **318** | UINT | Reading 20 decimal places: the number of digits after the implied decimal point in the current, minimum and maximum values. |

## Command assemblies

Send a command by writing its command ID and argument to assembly 109; read its status and result in assembly 110. Assemblies 102 and 103 are limited command assemblies, kept for systems built for earlier EtherNet/IP commands of the device. The procedure is in Sending a command (p. 22).

### Assembly 109: command request

| Byte offset | Type | Description |
|---|---|---|
| **0** | `UDINT` | Command ID |
| **4** | `DINT` | Command argument |

### Assembly 110: command result

| Byte offset | Type | Description |
|---|---|---|
| **0** | `UDINT` | ID of the last command written to 109 |
| **4** | `DINT` | Argument of the last command written to 109 |
| **8** | `UDINT` | Status of the last command written (table 3-6) |
| **12** | `DINT` | Return value of the last command written to 109 |

### Assembly 102: limited command request

Assembly 102 accepts any command of this chapter with two conditions: the command ID is 65535 or lower, and the firmware of the device is the same as or later than the firmware that introduced the command. The result is read in assembly 103.

| Parameter | Type | Description |
|---|---|---|
| Command ID | `UINT` | See the command list (table 3-7). |
| Command argument | `UINT` | See the description of the command. |

### Assembly 103: limited command result

Assembly 103 holds the result of the most recent command written to assembly 102 only.

| Parameter | Type | Description |
|---|---|---|
| Command ID | `UINT` | ID of the last command |
| Command status | `UINT` | Status of the last command (table 3-6) |

### Assembly 104: gas mix configuration

Assembly 104 holds the gas mix that create/update gas mix (p. 34) (command 2) stores. Write the mix to assembly 104 before you send the command. The 20 bytes hold up to 5 gases, each gas number followed by its share of the mix:

- Every gas number must be a gas available on the device; otherwise the command fails. The gases installed are listed in the device operating manual and in the gas list on the display.
- Write each share in hundredths of a percent. The shares must add up to 100 %, that is 10000; otherwise the command fails.

| Byte offset | Type | Description |
|---|---|---|
| **0** | `UINT` | Mix gas number 1 |
| **2** | `UINT` | Share of gas 1, in 0.01 % |
| **4** | `UINT` | Mix gas number 2 |

| Byte offset | Type | Description |
|---|---|---|
| 6 | UINT | Share of gas 2, in 0.01 % |
| 8 | UINT | Mix gas number 3 |
| 10 | UINT | Share of gas 3, in 0.01 % |
| 12 | UINT | Mix gas number 4 |
| 14 | UINT | Share of gas 4, in 0.01 % |
| 16 | UINT | Mix gas number 5 |
| 18 | UINT | Share of gas 5, in 0.01 % |

# Commands

## Sending a command

A command needs a command ID and an argument, written to the command request assembly 109 or to the limited command request assembly 102:

### 3-5 Command request

| Byte offset | Type | Description |
|---|---|---|
| 0 | Unsigned32 | ID of the command |
| 4 | Integer32 | Argument of the command |

1. Write the command ID to byte offset 0 of assembly 109, and the argument to byte offset 4. The ID selects the command; the argument selects one of its options.
2. Read assembly 110. It returns the command ID, the argument, the status of the command (table 3-6) and its return value.
3. Before the next command, send the no-operation command (ID 0). It clears the previous command and makes sure the next one is written to the device as intended.

The device runs a command only when the command ID or the argument changes, so the same command cannot run twice in a row. If you need to repeat it, send command 0 in between. Sending command 0 before or after every command prevents a command from repeating by accident. Command 0 is already built into the Add-On Instructions described in Example: sending a command (p. 40).

Some commands give a return value, described with each command. A command that defines no return value returns 0.

## Command status

### 3-6 Command status values

| Value | Status | Description |
|---|---|---|
| 0 | SUCCESS | The last command completed successfully. |
| 1 | IN_PROGRESS | A command is running. |
| 2 | INVALID_ID | The ID of the last command is not valid. |
| 3 | INVALID_ARGUMENT | The argument of the last command is not valid. |
| 4 | UNSUPPORTED | The device does not support the last command. |
| 5 | INVALID_MIX_IDX | The requested gas mix number is not valid. |

| Value | Status | Description |
|---|---|---|
| 6 | INVALID_MIX_GAS | A gas of the mix does not exist on the device. |
| 7 | INVALID_MIX_PCT | The shares of the gas mix do not add up to 100 %. |

## Command list

Each command gives the firmware version that introduced it, its command ID, what it does, its argument and the response to a successful command.

### 3-7 Commands

| Command ID | Command | Firmware | Devices |
|---|---|---|---|
| **COMMAND FUNCTION** | | | |
| **0** | No operation (p. 24) | 1v00 | All devices |
| **DATA READINGS** | | | |
| **32** | Query reading type (p. 25) | 10v07 | All devices |
| **27** | Query reading source (p. 25) | 10v07 | All devices |
| **65538** | Query reading minimum (integer) (p. 25) | 10v07 | All devices |
| **65536** | Query reading minimum (float) (p. 26) | 10v07 | All devices |
| **65539** | Query reading maximum (integer) (p. 26) | 10v07 | All devices |
| **65537** | Query reading maximum (float) (p. 26) | 10v07 | All devices |
| **29** | Query reading engineering units (p. 26) | 10v07 | All devices |
| **65300 + offset** | Set reading engineering units (p. 27) | 10v07 | All devices |
| **30** | Query reading decimal places (p. 27) | 10v07 | All devices |
| **CONTROL** | | | |
| **12** | Save power-up set point (p. 27) | 7v05 | Controllers only |
| **65546** | Set point maximum ramp (saved) (p. 28) | 10v07 | Controllers only |
| **65547** | Set point maximum ramp (temporary) (p. 28) | 10v07 | Controllers only |
| **6** | Hold valve(s) (p. 29) | 7v05 | Controllers only |
| **15** | Set active valve (p. 29) | 7v16 | Controllers only |
| **11** | Set loop control variable (p. 29) | 7v05 | Controllers only |
| **13** | Set loop control algorithm (p. 30) | 7v08 | Controllers only |
| **14** | Read closed-loop gain (p. 30) | 7v08 | Controllers only |
| **8** | Set proportional gain (p. 30) | 7v05 | Controllers only |
| **9** | Set derivative gain (p. 30) | 7v05 | Controllers only |
| **10** | Set integral gain (p. 30) | 7v05 | Controllers only |
| **16** | Set inverse pressure control (p. 31) | 7v18 | Controllers only |
| **65543** | Query totaliser batch (integer) (p. 31) | 10v07 | Controllers only |
| **65540** | Query totaliser batch (float) (p. 31) | 10v07 | Controllers only |
| **65544** | Set totaliser batch 1 (integer) (p. 31) | 10v07 | Controllers only |
| **65541** | Set totaliser batch 1 (float) (p. 32) | 10v07 | Controllers only |
| **65545** | Set totaliser batch 2 (integer) (p. 32) | 10v07 | Controllers only |
| **65542** | Set totaliser batch 2 (float) (p. 32) | 10v07 | Controllers only |
| **ADMINISTRATIVE** | | | |
| **4** | Perform tare (p. 33) | 7v05 | All devices |
| **31** | Perform pressure sensor tare (p. 33) | 10v07 | All devices |

| Command ID | Command | Firmware | Devices |
|---|---|---|---|
| **32** | Perform secondary pressure sensor tare (p. 33) | 10v07 | All devices |
| **33** | Perform flow tare (p. 34) | 10v07 | Flow devices |
| **5** | Reset totaliser (p. 34) | 7v05 | Flow devices |
| **1** | Set gas (p. 34) | 7v05 | Mass flow devices |
| **2** | Create/update gas mix (p. 34) | 7v05 | Mass flow devices |
| **3** | Delete gas mix (p. 35) | 7v05 | Mass flow devices |
| **24** | Set relative humidity percentage (p. 35) | 10v07 | Mass flow devices with humidity sensor |
| **25** | Set relative humidity reference temperature (p. 35) | 10v07 | Mass flow devices with humidity sensor |
| **7** | Lock/unlock display (p. 35) | 7v05 | All devices |
| **20** | Flash display (p. 35) | 8v28 | All devices |
| **26** | Restore factory settings (p. 36) | 10v07 | All devices |
| **17** | Read configuration checksum (p. 36) | 8v24 | All devices |

## Command function

This command makes sure that commands run as intended.

### No operation

`1v00` All devices

> `Command ID` **`0`**

Does nothing. It separates two identical commands (same ID and argument) and may be sent before every command or at any time.

`Argument` Ignored.

**Response:** None.

## Data readings

These commands give information on the readings of the device: the statistic, source, minimum and maximum, engineering units and decimal places. Most return a code; the appendices give its meaning.

### Query reading type

10v07 All devices

`Command ID` **32**

Returns the statistic of a reading, a value from Appendix A (p. 44).

`Argument` The reading, from the table below.

| Value | Reading |
|---|---|
| 0 | Set point |
| 1 | Valve drive |
| 2 | Pressure |
| 3 | Secondary pressure |
| 4 | Barometric pressure |
| 5 | Temperature |
| 6 | Volumetric flow |
| 7 | Mass flow |
| 8 | Totaliser 1 |
| 9 | Totaliser 2 |
| 10 | Humidity |

**Response:** `SUCCESS` with the statistic as the return value; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Query reading source

10v07 All devices

`Command ID` **27**

Returns the source of a reading, a value from Appendix C (p. 50).

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the source as the return value; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Query reading minimum (integer)

10v07 All devices

`Command ID` **65538**

Returns the lowest value a reading can have, without overrange, as an integer.

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the minimum as an integer; multiply it by $10^{-n}$, n = decimal places of the reading, for the minimum in engineering units; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Query reading minimum (float)

10v07 All devices

`Command ID` **65536**

Returns the lowest value a reading can have, without overrange, as a floating-point value.

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the minimum as an IEEE 754 single-precision floating-point value in the engineering units of the reading; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Query reading maximum (integer)

10v07 All devices

`Command ID` **65539**

Returns the highest value a reading can have, without overrange, as an integer.

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the maximum as an integer; multiply it by $10^{-n}$, n = decimal places of the reading, for the maximum in engineering units; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Query reading maximum (float)

10v07 All devices

`Command ID` **65537**

Returns the highest value a reading can have, without overrange, as a floating-point value.

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the maximum as an IEEE 754 single-precision floating-point value in the engineering units of the reading; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Query reading engineering units

10v07 All devices

`Command ID` **29**

Returns the current engineering units of a reading, a value from Appendix B (p. 47).

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the units as the return value; `INVALID_ARGUMENT` if the device has no reading for the argument.

### Set reading engineering units

10v07 All devices

```
Command ID 65300 + offset
```

Changes the engineering units of a reading. Add the offset of the reading (table below) to the command ID 65300: for the pressure reading, for example, use command ID 65302. The readings are not all independent: a change of the pressure units also changes the units of the secondary and barometric pressure.

`Argument` The value of the new units from Appendix B (p. 47).

| Offset | Reading |
|---|---|
| 0 | Set point |
| 1 | Valve drive |
| 2 | Pressure |
| 3 | Secondary pressure |
| 4 | Barometric pressure |
| 5 | Temperature |
| 6 | Volumetric flow |
| 7 | Mass flow |
| 8 | Totaliser 1 |
| 9 | Totaliser 2 |
| 10 | Humidity |

**Response:** `SUCCESS` if the units changed, with the new units as the return value; `UNSUPPORTED` if the offset in the command ID does not apply to the device; `INVALID_ARGUMENT` if the units are not valid for the reading.

### Query reading decimal places

10v07 All devices

```
Command ID 30
```

Returns the number of digits after the implied decimal point in the current, minimum and maximum values of a reading.

`Argument` The reading, as for query reading type (p. 25).

**Response:** `SUCCESS` with the number of decimal places as the return value; `INVALID_ARGUMENT` if the device has no reading for the argument.

## Control

These commands are for controllers only. They manage the set point, the valves and the totaliser batches.

### Save power-up set point

7v05 Controllers only

```
Command ID 12
```

Stores the current set point, to be used at once the next time the controller is powered on.

`Argument` Ignored.

**Response:** `SUCCESS`; the current set point is saved as the power-up set point.

### Set point maximum ramp (saved)

10v07 Controllers only

`Command ID` **65546**

Sets the maximum ramp rate of the set point and keeps it across power cycles. Do not send this command more often than every few minutes.

`Argument` A negative value queries the current maximum ramp rate; 0 switches the limit off. To set a limit, take the change in percent of full scale per millisecond and multiply it by 10 000 000. For example, 1 % of full scale every second is 0.001 % every millisecond, which gives 10000. More values are in the table below.

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

**Response:** `SUCCESS` when the command is done, with the current maximum ramp rate as the return value. Multiply it by 0.0000001 for the rate in percent of full scale per millisecond.

### Set point maximum ramp (temporary)

10v07 Controllers only

`Command ID` **65547**

Sets the maximum ramp rate of the set point without saving it: the value is lost when the device loses power. The values are the same as for set point maximum ramp (saved) (p. 28).

`Argument` A negative value queries the current maximum ramp rate; 0 switches the limit off. To set a limit, take the change in percent of full scale per millisecond and multiply it by 10 000 000. For example, 1 % of full scale every second is 0.001 % every millisecond, which gives 10000. More values are in the table below.

**Response:** `SUCCESS` when the command is done, with the current maximum ramp rate as the return value. Multiply it by 0.0000001 for the rate in percent of full scale per millisecond.

### Hold valve(s)

7v05 Controllers only

`Command ID` **6**

Holds the valve or valves of the controller and stops any further control of the process, or cancels a hold.

`Argument` A value from the table below.

| Value | Meaning |
|---|---|
| 0 | Cancel the hold and resume normal closed-loop control. |
| 1 | Hold all valves closed. |
| 2 | Hold the valves at their current positions. |
| 3 | Exhaust: close the upstream valve and open the downstream valve fully. Dual-valve controllers only. |

**Response:** `SUCCESS` if the valves are set to the requested mode; `UNSUPPORTED` if the device does not support the mode; `INVALID_ARGUMENT` if the argument is not valid.

### Set active valve

7v16 Controllers only

`Command ID` **15**

Selects the valve that actively controls the process. Available on stream-switching controllers only.

`Argument` A value from the table below.

| Value | Meaning |
|---|---|
| 0 | Upstream valve, or the only valve |
| 1 | Downstream valve, or auxiliary valve if dual-valve control is enabled |

**Response:** `SUCCESS` if the active valve changes; `INVALID_ARGUMENT` if the value is not valid; `UNSUPPORTED` if the controller is not a stream-switching controller.

### Set loop control variable

7v05 Controllers only

`Command ID` **11**

Changes the statistic the controller controls: a mass flow controller can, for example, be set to control pressure or volumetric flow.

`Argument` A statistic from the table below.

| Value | Meaning |
|---|---|
| 0 | Mass flow |
| 1 | Volumetric flow |
| 2 | Differential pressure |
| 3 | Absolute pressure |
| 4 | Gauge pressure |
| … | Any set point statistic of Appendix A (p. 44) |

**Response:** `SUCCESS` if the loop control variable changes; `INVALID_ARGUMENT` if the statistic is not available on the device (for example mass flow on a pressure controller).

### Set loop control algorithm

`7v08` Controllers only

`Command ID 13`

Selects the PDF or the PD2I closed-loop control algorithm. Only one algorithm can be active at a time.

`Argument` A value from the table below.

| Value | Meaning |
|---|---|
| 1 | PDF closed-loop control algorithm |
| 2 | PD2I closed-loop control algorithm |

**Response:** `SUCCESS` if the algorithm is selected; `INVALID_ARGUMENT` if the value is not a valid algorithm.

### Read closed-loop gain

`7v08` Controllers only

`Command ID 14`

Returns the current value of a gain of the control algorithm.

`Argument` A gain from the table below.

| Value | Meaning |
|---|---|
| 0 | Proportional gain (P gain), PDF or PD2I |
| 1 | Derivative gain (D gain), PDF or PD2I |
| 2 | Integral gain (I gain), PD2I |

**Response:** `SUCCESS` with the gain, 0–65535, as the return value; `INVALID_ARGUMENT` if the gain is not part of the active algorithm.

### Set proportional gain

`7v05` Controllers only

`Command ID 8`

Sets the proportional gain (P gain) of the control algorithm.

`Argument` The gain, 0–65535.

**Response:** `SUCCESS` when the gain is set.

### Set derivative gain

`7v05` Controllers only

`Command ID 9`

Sets the derivative gain (D gain) of the control algorithm.

`Argument` The gain, 0–65535.

**Response:** `SUCCESS` when the gain is set.

### Set integral gain

`7v05` Controllers only

`Command ID 10`

Sets the integral gain (I gain) of the control algorithm.

`Argument` The gain, 0–65535.

**Response:** `SUCCESS` when the gain is set.

### Set inverse pressure control

7v18 · Controllers only

`Command ID 16`

Sets the controller to control pressure normally or inversely (usually back pressure), and whether this mode is kept across power cycles.

`Argument` A value from the table below.

| Value | Meaning |
|---|---|
| 0 | Normal pressure control, not saved across power cycles |
| 1 | Inverse pressure control (usually back pressure), not saved across power cycles |
| 3 | Normal pressure control, saved across power cycles |
| 4 | Inverse pressure control (usually back pressure), saved across power cycles |

**Response:** `SUCCESS` if the pressure control mode changed; `INVALID_ARGUMENT` if the argument is not available on the device or has no pressure control mode.

### Query totaliser batch (integer)

10v07 · Controllers only

`Command ID 65543`

Returns the batch size of one of the two totalisers as an integer.

`Argument` 1 for totaliser 1, 2 for totaliser 2.

**Response:** `SUCCESS` if the totaliser is enabled, with the batch size as the return value; multiply it by $10^{-n}$, n = decimal places of the totaliser, for the volume in its engineering units. The return value is 0 if batching is off. `INVALID_ARGUMENT` if the totaliser is not enabled.

### Query totaliser batch (float)

10v07 · Controllers only

`Command ID 65540`

Returns the batch size of one of the two totalisers as a floating-point value.

`Argument` 1 for totaliser 1, 2 for totaliser 2.

**Response:** `SUCCESS` if the totaliser is enabled, with the batch size as an IEEE 754 single-precision floating-point value in the engineering units of the totaliser. The return value is 0 if batching is off. `INVALID_ARGUMENT` if the totaliser is not enabled.

### Set totaliser batch 1 (integer)

10v07 · Controllers only

`Command ID 65544`

Sets the batch size of totaliser 1.

`Argument` The batch size multiplied by $10^{n}$, n = decimal places of the totaliser. 0 switches batching off.

**Response:** `SUCCESS` if totaliser 1 is enabled and the batch size is valid, with the batch size as the return value; `UNSUPPORTED` if totaliser 1 is not enabled; `INVALID_ARGUMENT` if the batch is larger than totaliser 1 can hold.

### Set totaliser batch 1 (float)

10v07 Controllers only

> Command ID **65541**

Sets the batch size of totaliser 1.

`Argument` The batch size as an IEEE 754 single-precision floating-point value, in the engineering units of totaliser 1. 0 switches batching off.

**Response:** `SUCCESS` if totaliser 1 is enabled and the batch size is valid, with the requested batch volume as the return value; `UNSUPPORTED` if totaliser 1 is not enabled; `INVALID_ARGUMENT` if the batch is larger than totaliser 1 can hold.

### Set totaliser batch 2 (integer)

10v07 Controllers only

> Command ID **65545**

Sets the batch size of totaliser 2.

`Argument` The batch size multiplied by $10^n$, n = decimal places of the totaliser. 0 switches batching off.

**Response:** `SUCCESS` if totaliser 2 is enabled and the batch size is valid, with the batch size as the return value; `UNSUPPORTED` if totaliser 2 is not enabled; `INVALID_ARGUMENT` if the batch is larger than totaliser 2 can hold.

### Set totaliser batch 2 (float)

10v07 Controllers only

> Command ID **65542**

Sets the batch size of totaliser 2.

`Argument` The batch size as an IEEE 754 single-precision floating-point value, in the engineering units of totaliser 2. 0 switches batching off.

**Response:** `SUCCESS` if totaliser 2 is enabled and the batch size is valid, with the requested batch volume as the return value; `UNSUPPORTED` if totaliser 2 is not enabled; `INVALID_ARGUMENT` if the batch is larger than totaliser 2 can hold.

## Administrative

Taring the sensors, totaliser reset, gas selection and gas mixes, display control and factory settings.

### Perform tare

`7v05` All devices

> `Command ID` **4**

Takes the current reading as the zero reading. A gauge pressure tare needs the sensor open to atmosphere; a differential pressure tare needs a common pressure to measure from; an absolute pressure tare needs a barometer in the device. A flow tare needs no flow through the process, and is best done at, or as close as possible to, process pressure.

`Argument` A value from the table below.

| Value | Meaning |
|---|---|
| 0 | Tare gauge or differential pressure |
| 1 | Tare absolute pressure |
| 2 | Tare mass and/or volumetric flow |

**Response:** `SUCCESS` if the tare is done; `UNSUPPORTED` if the tare does not apply to the device (for example a mass flow tare on a pressure device).

### Perform pressure sensor tare

`10v07` All devices

> `Command ID` **31**

Takes the current gauge or differential pressure reading as the zero reading; an absolute pressure tare uses the zero reading of the barometer. A gauge pressure tare needs the sensor open to atmosphere; a differential pressure tare needs a common pressure to measure from; an absolute pressure tare needs a barometer in the device.

`Argument` The tare time in milliseconds, 0–32767; 0 uses the default of 256 milliseconds.

**Response:** `SUCCESS` if the tare is done; `UNSUPPORTED` if the sensor cannot be tared, for example an absolute pressure tare without a barometer; `INVALID_ARGUMENT` if the argument is below 0 or above 32767.

### Perform secondary pressure sensor tare

`10v07` All devices

> `Command ID` **32**

Takes the current reading of the secondary pressure sensor as the zero reading; an absolute pressure tare uses the zero reading of the barometer. Not every device has a secondary pressure sensor. A gauge pressure tare needs the sensor open to atmosphere; a differential pressure tare needs a common pressure to measure from; an absolute pressure tare needs a barometer in the device.

`Argument` The tare time in milliseconds, 0–32767; 0 uses the default of 256 milliseconds.

**Response:** `SUCCESS` if the tare is done; `UNSUPPORTED` if the sensor cannot be tared, usually an absolute pressure tare without a barometer, or no secondary pressure sensor; `INVALID_ARGUMENT` if the argument is below 0 or above 32767.

### Perform flow tare

10v07 · Flow devices

```
Command ID 33
```

Takes the current flow reading as the zero reading, for volumetric and mass flow. A flow tare needs no flow through the process, and is best done at, or as close as possible to, process pressure.

`Argument` The tare time in milliseconds, 0–32767; 0 uses the default of 256 milliseconds.

**Response:** `SUCCESS` if the tare is done; `UNSUPPORTED` if the flow cannot be tared, usually because the device does not read volumetric or mass flow; `INVALID_ARGUMENT` if the argument is below 0 or above 32767.

---

### Reset totaliser

7v05 · Flow devices

```
Command ID 5
```

Clears the current totaliser count.

`Argument` None.

**Response:** `SUCCESS` when the reset is complete.

---

### Set gas

7v05 · Mass flow devices

```
Command ID 1
```

Selects the gas the mass flow device is set to read. Set the process gas to read its mass flow accurately.

`Argument` The gas number from Appendix D (p. 51).

**Response:** `SUCCESS` if the gas changed; `INVALID_ARGUMENT` if the gas number does not exist.

---

### Create/update gas mix

7v05 · Mass flow devices

```
Command ID 2
```

Writes a new gas mix to the gas list of the device, from the mix written beforehand to assembly 104 (p. 21). Using the number of an existing mix overwrites that mix.

`Argument` 0 or 236–255: the gas number of the mix. 0 writes the mix to the first free number, starting at 255 and counting down; if no number is free, the command fails.

**Response:** `SUCCESS` with the gas number of the mix as the return value; `INVALID_MIX_IDX` if the number is outside 236–255; `INVALID_MIX_GAS` if a gas of the mix does not exist on the device; `INVALID_MIX_PCT` if the shares do not add up to 100 %.

### Delete gas mix

7v05 · Mass flow devices

`Command ID 3`

Removes a gas mix from the device.

`Argument` The gas number of the mix to delete.

**Response:** `SUCCESS` if the mix was deleted; `INVALID_MIX_IDX` if the mix does not exist.

### Set relative humidity percentage

10v07 · With humidity sensor

`Command ID 24`

Sets the relative humidity used for gas corrections.

`Argument` 0–10000, in 0.01 % steps: 100 is 1 %, 10000 is 100 %.

**Response:** `SUCCESS` if the relative humidity changed; `UNSUPPORTED` if the relative humidity cannot be set on the device; `INVALID_ARGUMENT` if the argument is outside 0–10000.

### Set relative humidity reference temperature

10v07 · With humidity sensor

`Command ID 25`

Sets the reference temperature of the relative humidity used for gas corrections, in degrees Celsius. It should match the relative humidity percentage.

`Argument` −3000 to 10000, in 0.01 °C steps: −3000 is −30 °C, 10000 is 100 °C.

**Response:** `SUCCESS` if the reference temperature changed; `UNSUPPORTED` if the relative humidity cannot be set on the device; `INVALID_ARGUMENT` if the argument is outside −3000 to 10000.

### Lock/unlock display

7v05 · All devices

`Command ID 7`

Locks or unlocks the display of the device. While it is locked, the device still responds to button presses, but settings cannot be changed on the display.

`Argument` 0 unlocks the display; any other value locks it.

**Response:** `SUCCESS` if the display is locked or unlocked as requested.

### Flash display

8v28 · All devices

`Command ID 20`

Makes the backlight of the display flash, for a set time or without end.

`Argument` 1–65534: flash for this number of seconds. 0 stops the flashing; 65535 flashes without end.

**Response:** `SUCCESS` if the backlight flashes or stops as requested; `UNSUPPORTED` if no display is connected to the device.

### Restore factory settings

`10v07` All devices

`Command ID` **26**

Returns all settings and the configuration of the device to their state when the device last left the factory. Power-cycle the device after the restore.

**NOTICE**

Restoring factory settings removes all third-party calibrations. Use this command only while troubleshooting with Analyt-MTC support.

`Argument` 49374, to confirm the restore.

**Response:** `SUCCESS` if the restore completed (it may be readable only briefly before the restore clears the status); `UNSUPPORTED` if an error occurred during the restore; `INVALID_ARGUMENT` if the argument is not 49374.

### Read configuration checksum

`8v24` All devices

`Command ID` **17**

Computes a checksum of the calibration and configuration of the device when the command starts. This can take 300 milliseconds; use a timeout of more than 500 milliseconds. Every setting kept across power cycles is included, and values that may or may not be kept as well. Because the calibration is included, two devices with identical configurations can have different checksums. A recalibration can change the checksum. A firmware change may or may not change it. Set any parameter that changes in routine operation, such as the set point, to a known value before you read the checksum.

`Argument` Always 0.

**Response:** `SUCCESS` with the checksum of the whole configuration, 0–65535, as the return value; `INVALID_ARGUMENT` if the argument is not 0.

CHAPTER 4

# Examples

Setting up a Rockwell Logix PLC for a 358 Series device, formatting its I/O data, changing the set point and sending commands.

## Rockwell PLC set-up

For the Rockwell Logix Designer software, an electronic data sheet (EDS) and Logix XML files with Add-On Instructions (AOI) are available: use the device description file supplied with the instrument, and ask Analyt-MTC for the Add-On Instruction files.

### Determining the input size of assembly 101

The size and configuration of assembly 101 depend on the device. To connect to the device, the input size of the connection parameters must be defined. If the size is wrong, the PLC faults with an *invalid input size* message.

You can read the size of assembly 101 with RSNetWorx: use service code `0x0E`, Get Attribute Single, on class 4, instance 101, attribute 4. The size is returned in hexadecimal. Enter the instance in hexadecimal as well: instance 101 is `65`.

The *Data I/O* page of the embedded web server also gives the size of assembly 101 (Data I/O page (p. 10)).

### Creating a module for the device

There are two ways to create a module for the device in Logix: import the device description file (EDS) and create an EtherNet/IP adapter, or create the device as a generic Ethernet module.

#### With the device description file (EDS)

Import the device description file supplied with the instrument with the *EDS Hardware Installation Tool* in the *Tools* menu of RSLogix.

1. Follow the prompts to register a single EDS device.
2. When prompted, browse to the device description file.
3. Continue through the installation tool.
4. Register the file as a single file.
5. When the import is complete, right-click the Ethernet network in the I/O configuration tree and select *New Module*.
6. Select the EtherNet/IP adapter of the device and click *Create*.
7. Name the module and enter the static IP address assigned to the device.
8. Optional: to change the input size, click *Change* in the *Module Definition* part of the window.
9. When the input size is set, click *OK*.
10. Click *OK*, and repeat these steps for every other device you need to connect.

### As a generic Ethernet module

If the device description file does not meet the needs of your process, set the device up as a generic Ethernet module.

1. In the I/O configuration tree, right-click the Ethernet network and select *New Module*.
2. Select *Generic Ethernet Module*, then *Create*.
3. Name the module and enter the static IP address assigned to the device.
4. Enter the input size of assembly 101.
5. Enter the output size of assembly 100: 4 for a controller, 0 for a meter or gauge.
6. For the configuration, enter assembly instance 1 and size 0. The device has no configuration assembly; the configuration assembly is ignored.
7. As the communication format, select `SINT`. This creates a controller tag holding the input and output data as an array of bytes.
8. Click *OK*, and repeat these steps for every other device you need to connect.

## Formatting the I/O data

When the module is added in RSLogix, controller tags are created with the module name followed by `:I` for the input and `:O` for the output. For a module named `Flow_1`, for example, the tags are `Flow_1:I` and `Flow_1:O`. By default both are arrays of `SINT` values, although the input and output data are of mixed types.

To format these data, import the Add-On Instruction that updates the input (ask Analyt-MTC for the file):

1. In the main project window, select *File* → *Import Component* → *Add-On Instruction*.
2. Browse to the file and select *Import*.

When the import is complete, an AOI named `UpdateInput`, a UDT named `IoData` and an Add-On-defined data type appear in the controller organiser.

### UDT format

#### 4-1 IoData UDT

| Name | Type | Description |
|---|---|---|
| `Status` | `DINT` | Bit mask of the device status flags |
| `Pressure` | `REAL` | Pressure reading |
| `Temperature` | `REAL` | Temperature reading |
| `VolFlow` | `REAL` | Volumetric flow reading |
| `MassFlow` | `REAL` | Mass flow reading |
| `ActualSetpoint` | `REAL` | Set point reading |
| `ValveDrivePct` | `REAL` | Valve drive reading, in percent |
| `Totalizer` | `REAL` | Totaliser reading |
| `RequestedSetpoint` | `REAL` | Requested set point |
| `GasIndex` | `INT` | Number of the selected gas |
| `Config` | `SINT` | Validity flags of the device |

### 4-2 IoData UDT: Config overlay

| Name | Type | Description |
|---|---|---|
| `Config[0] – PRESS` | `BOOL` | The pressure reading is valid. |
| `Config[1] – TEMP` | `BOOL` | The temperature reading is valid. |
| `Config[2] – VFLOW` | `BOOL` | The volumetric flow reading is valid. |
| `Config[3] – MFLOW` | `BOOL` | The mass flow reading is valid. |
| `Config[4] – SP` | `BOOL` | The set point reading is valid. |
| `Config[5] – VDRIVE` | `BOOL` | The valve drive reading is valid. |
| `Config[6] – TOTAL` | `BOOL` | The totaliser reading is valid. |

The status flags have a bit overlay that decodes the individual flags:

### 4-3 IoData UDT: Status overlay

| Name | Type | Description |
|---|---|---|
| `Status[0] – T_OVER` | `BOOL` | Temperature overflow. |
| `Status[1] – T_UNDER` | `BOOL` | Temperature underflow. |
| `Status[2] – V_OVER` | `BOOL` | Volumetric overflow. |
| `Status[3] – V_UNDER` | `BOOL` | Volumetric underflow. |
| `Status[4] – M_OVER` | `BOOL` | Mass flow overflow. |
| `Status[5] – M_UNDER` | `BOOL` | Mass flow underflow. |
| `Status[6] – P_OVER` | `BOOL` | Pressure overflow. |
| `Status[7] – TOT_OVER` | `BOOL` | Totaliser overflow. |
| `Status[8] – HLD` | `BOOL` | The PID loop is holding the valve positions. |
| `Status[9] – ADC` | `BOOL` | ADC error. |
| `Status[10] – EXH` | `BOOL` | The controller is exhausting the process. |
| `Status[11] – OPL` | `BOOL` | Over pressure limit. |
| `Status[12] – TMF` | `BOOL` | The totaliser missed flow during an overflow. |

The config flag is stored as a hexadecimal number. A `MOV` instruction also accepts binary arguments and converts the type itself: write a binary value with the prefix `2#`.

### 4-4 IoData UDT: Config flag values

| Device | Config | Flags |
|---|---|---|
| Pressure controller | `16#11` (`2#10001`) | `PRESS`, `SP` |
| Mass flow meter | `16#0F` (`2#1111`) | `PRESS`, `TEMP`, `VFLOW`, `MFLOW` |
| Mass flow controller | `16#1F` (`2#11111`) | `PRESS`, `TEMP`, `VFLOW`, `MFLOW`, `SP` |

## Using the AOI

Create a tag of type `IoData` for each device. The `UpdateInput` instruction also needs a tag of its own.

A ControlLogix PLC updates the I/O data asynchronously, so create an array of type `SINT` to buffer the input data before it passes to the AOI. One buffer can serve several devices, provided it is as large as the largest input size. If the buffer is too small, the I/O data is truncated.

## Changing the set point

To change the set point of a controller, copy a 4-byte floating-point number (`REAL`) to the output of the device. The `IoData` data type has a read/write element of type `REAL` for the requested set point. A copy instruction from this element into the output tag lets you control the set point by editing the element.

## Example: sending a command

Commands can be sent to the device asynchronously, as explicit CIP messages to assemblies 102, 103 and 104. An Add-On Instruction with the logic for this is available; ask Analyt-MTC for the file. It sends well-formed commands to instance 102 and reads the result from instance 103.

Import the command Add-On Instruction. It needs four controller tags:

- two message tags, which define the path to the device and the configuration of the explicit message;
- a 4-byte array, used as a buffer for the command and the response on their way to the I/O stream;
- a UDT of type `Command`, which prepares the command to be copied to the buffer and is tied to the logic that manages the explicit messages.

A fifth tag, of type Boolean, can serve as a switch to start the command.

**NOTICE**

These tags must be controller tags. Otherwise they are not scoped to communicate with the I/O stream.

Configure each message tag: click the tag name and select *Configure*. Set the path to the device on the *Communication* tab of both messages.

### 4-5 Message tag settings

| Setting | Instance 102 (command) | Instance 103 (result) |
|---|---|---|
| Message type | CIP Generic | CIP Generic |
| Service type | Set Attribute Single | Get Attribute Single |
| Service code | `0x10` | `0x0E` |
| Instance | 102 | 103 |
| Class | 4 | 4 |
| Attribute | 3 | 3 |
| Source element | `CmdData` | — |
| Source length | 4 | — |
| Destination element | — | `CmdData` |

## Worked argument values

Argument values taken from the command descriptions in Chapter 3 (p. 14):

### 4-6 Worked argument values

| Task | Command ID | Argument | Why |
|---|---|---|---|
| Limit the set point ramp to 1 % of full scale per second, kept after power-off | 65546 | 10000 | 0.001 % per millisecond × 10 000 000 (set point maximum ramp (saved) (p. 28)) |

| Task | Command ID | Argument | Why |
|---|---|---|---|
| Change the engineering units of the pressure reading | 65302 | — | 65300 + offset 2 of the pressure reading; the argument is the units value from Appendix B (p. 47) (set reading engineering units (p. 27)) |
| Set the relative humidity to 1 % | 24 | 100 | 1 count = 0.01 % (set relative humidity percentage (p. 35)) |
| Restore factory settings | 26 | 49374 | Confirmation value (restore factory settings (p. 36)) |

**NOTICE**

Restoring factory settings removes all third-party calibrations. Use this command only while troubleshooting with Analyt-MTC support.

CHAPTER 5

# Error handling

Common symptoms, their likely causes and what to do. If none of these fits, contact Analyt-MTC support with the information listed in About this guide (p. 2).

## Symptoms and remedies

| Symptom | Cause and remedy |
|---|---|
| **CONNECTION** | |
| NET LED off | No IP address is assigned. By default the device obtains its address from a DHCP server; you can also assign a static address (Setting the IP address (p. 9)). |
| NET LED steady red | Another device on the network has the same IP address. Give the device a free address; ask your network administrator which addresses are free. |
| NET LED flashing red | The EtherNet/IP connection timed out. The *Comm Status* screen shows whether the link is up and whether a PLC is connected (Comm Status screen (p. 9)). |
| MOD LED steady or flashing red | A major (steady) or minor (flashing) system fault in the device. Contact Analyt-MTC support. |
| The device is no longer found at its IP address after a power loss | With DHCP, the IP address may change when the device is powered on again. Read the address on the *Comm Status* screen (Comm Status screen (p. 9)) or assign a static address. |
| The connection to the web server is lost after *Update* | The network settings changed. Go to the new IP address in the browser (Assigning a static IP address with the web server (p. 9)). |
| The *Home* page does not show *EtherNet/IP* as the protocol | The device is not configured for EtherNet/IP. Contact Analyt-MTC support. |
| The PLC faults with *invalid input size* | The input size of the connection does not match assembly 101 of this device. Determine the size (Determining the input size of assembly 101 (p. 37)) and enter it in the module. |
| **DATA** | |
| The input data is cut off | The `SINT` buffer for the AOI is smaller than the input size. Make it as large as the largest input size (Using the AOI (p. 39)). |
| The message tags do not reach the I/O stream | The tags are not controller tags. Create them as controller tags (Example: sending a command (p. 40)). |
| A reading information field reads 0 | The reading is not available on this device (assembly 111 (p. 17), assembly 112 (p. 20)). |
| The set point has no effect | The device is a meter or gauge: it has no valve and ignores assemblies 100 and 105. |
| **COMMANDS** | |
| A repeated command does nothing | The device runs a command only when the ID or the argument changes. Send command 0 between two identical commands (Sending a command (p. 22)). |
| A command sent to assembly 102 is rejected | Assembly 102 accepts only command IDs up to 65535, and only commands introduced in the firmware of the device or earlier. Use assembly 109 (assembly 102 (p. 21)). |
| A configuration register written on the *Device Config* page made the device misbehave | Writing a configuration register can invalidate the calibration, disable communication or damage the device permanently. Contact Analyt-MTC support (Device Config page (p. 11)). |

## Command status codes

Assembly 110 (or 103 for the limited command assemblies) returns one of these status values after each command. The command descriptions in Commands (p. 22) give the cause of each status for each command.

| Value | Status | Meaning and remedy |
|---|---|---|
| 0 | `SUCCESS` | The last command completed successfully. |

| Value | Status | Meaning and remedy |
|---|---|---|
| 1 | `IN_PROGRESS` | A command is running. |
| 2 | `INVALID_ID` | The command ID is not valid. Check it against the command list (p. 23). |
| 3 | `INVALID_ARGUMENT` | The argument is not valid for this command. Check the allowed values in the command description. |
| 4 | `UNSUPPORTED` | The device does not support the command, for example a flow tare on a pressure device, or the exhaust mode on a single-valve controller. |
| 5 | `INVALID_MIX_IDX` | The gas mix number is not valid: use 0 or 236–255 (create/update gas mix (p. 34)), or a mix that exists (delete gas mix (p. 35)). |
| 6 | `INVALID_MIX_GAS` | A gas of the mix does not exist on the device. Check the gas numbers in assembly 104 (p. 21). |
| 7 | `INVALID_MIX_PCT` | The shares of the mix do not add up to 100 %, that is 10000 (assembly 104 (p. 21)). |

APPENDIX A

# Statistics

Statistics are the readings and measurements a device provides. Their values are returned by the information assemblies (assembly 111 (p. 17) and assembly 112 (p. 20)) and by query reading type (p. 25), and are used by set loop control variable (p. 29). Some devices and configurations restrict the statistics available. A firmware version in brackets is the version that added the statistic; without a version, the statistic is available on all devices. Ask Analyt-MTC about a statistic and how it works with your device.

## A-1 Flow statistics

| Flow statistic | Value | Notes |
|---|---|---|
| Batch mass remaining | **12** | Remaining mass in the totaliser batch |
| Batch volume remaining | **11** | Remaining volume, referenced to flow conditions, in the totaliser batch |
| Mass flow | **5** | Current mass flow |
| Mass flow, average | **69** | Average mass flow over the time of totalising |
| Mass flow, maximum (8v32) | **175** | Highest mass flow since reset |
| Mass flow, minimum (8v32) | **174** | Lowest mass flow since reset |
| Mass flow, peak | **101** | Peak mass flow during the time of totalising |
| Mass flow set point | **37** | Set point for mass flow |
| Mass flow set point error (8v00) | **173** | Mass flow minus the ramp-limited set point |
| Time, totalising | **10** | Time for which the totaliser has been running |
| Total mass | **9** | Totalised mass |
| Total volume | **8** | Totalised volume, referenced to flow conditions |
| Volumetric flow | **4** | Volumetric flow, referenced to flow conditions |
| Volumetric flow, average | **68** | Average volumetric flow, referenced to flow conditions, over the time of totalising |
| Volumetric flow, maximum (8v32) | **167** | Highest volumetric flow since reset |
| Volumetric flow, minimum (8v32) | **166** | Lowest volumetric flow since reset |
| Volumetric flow, peak | **100** | Peak volumetric flow, referenced to flow conditions, during the time of totalising |
| Volumetric flow set point | **36** | Set point for volumetric flow, referenced to flow conditions |
| Volumetric flow set point error (8v00) | **165** | Volumetric flow minus the ramp-limited set point |

## A-2 Pressure statistics

| Pressure statistic | Value | Notes |
|---|---|---|
| Pressure, absolute | **2** | Current absolute pressure |
| Pressure, absolute maximum (8v32) | **151** | Highest absolute pressure since reset |
| Pressure, absolute minimum (8v32) | **150** | Lowest absolute pressure since reset |
| Pressure, absolute set point | **34** | Set point for absolute pressure |
| Pressure, absolute set point error (8v00) | **149** | Absolute pressure minus the ramp-limited set point |
| Pressure, barometric | **15** | Barometer reading |
| Pressure, barometric maximum (8v32) | **255** | Highest barometric pressure since reset |
| Pressure, barometric minimum (8v32) | **254** | Lowest barometric pressure since reset |

| Pressure statistic | Value | Notes |
|---|---|---|
| Pressure, differential | **7** | Current differential pressure |
| Pressure, differential maximum (8v32) | **191** | Highest differential pressure since reset |
| Pressure, differential minimum (8v32) | **190** | Lowest differential pressure since reset |
| Pressure, differential set point | **39** | Set point for differential pressure |
| Pressure, differential set point error (8v00) | **189** | Differential pressure minus the ramp-limited set point |
| Pressure, gauge | **6** | Current gauge pressure |
| Pressure, gauge maximum (8v32) | **183** | Highest gauge pressure since reset |
| Pressure, gauge minimum (8v32) | **182** | Lowest gauge pressure since reset |
| Pressure, gauge set point | **38** | Set point for gauge pressure |
| Pressure, gauge set point error (8v00) | **181** | Gauge pressure minus the ramp-limited set point |
| Pressure, second absolute (7v01) | **344** | For devices with a pressure sensor in a second location, the absolute pressure of the second sensor |
| Pressure, second absolute maximum (8v32) | **351** | Highest second absolute pressure since reset |
| Pressure, second absolute minimum (8v32) | **350** | Lowest second absolute pressure since reset |
| Pressure, second absolute set point (7v01) | **345** | Set point for second absolute pressure |
| Pressure, second absolute set point error (8v00) | **349** | Second absolute pressure minus the ramp-limited set point |
| Pressure, second differential (7v01) | **360** | For devices with a pressure sensor in a second location, the differential pressure of the second sensor |
| Pressure, second differential maximum (8v32) | **367** | Highest second differential pressure since reset |
| Pressure, second differential minimum (8v32) | **366** | Lowest second differential pressure since reset |
| Pressure, second differential set point (7v01) | **361** | Set point for second differential pressure |
| Pressure, second differential set point error (8v00) | **365** | Second differential pressure minus the ramp-limited set point |
| Pressure, second gauge (7v01) | **352** | For devices with a pressure sensor in a second location, the gauge pressure of the second sensor |
| Pressure, second gauge maximum (8v32) | **359** | Highest second gauge pressure since reset |
| Pressure, second gauge minimum (8v32) | **358** | Lowest second gauge pressure since reset |
| Pressure, second gauge set point (7v01) | **353** | Set point for second gauge pressure |
| Pressure, second gauge set point error (8v00) | **357** | Second gauge pressure minus the ramp-limited set point |

## A-3 Other statistics

| Other statistic | Value | Notes |
|---|---|---|
| None | **1** | No statistic; usually an empty location |
| Set point | **32** | The current ramp-limited set point. Where this statistic is specified, the active set point statistic replaces it |
| Set point error (8v00) | **133** | Current process value minus the ramp-limited set point |
| Status | **26** | The status of the device; usually used internally only |
| Temperature, stream | **3** | Current stream temperature |
| Temperature, stream maximum (8v32) | **159** | Highest stream temperature since reset |
| Temperature, stream minimum (8v32) | **158** | Lowest stream temperature since reset |
| Valve drive | **13** | Valve drive signal |
| Valve drive set point (8v00) | **45** | The set point drives the currently selected valve directly |

| Other statistic | Value | Notes |
|---|---|---|
| Vapour fraction (percent water vapour) | **488** | The molecular fraction of vapour in the gas. When the vapour is water, this is the percentage of water vapour |
| Vapour fraction maximum | **495** | Highest vapour fraction since reset |
| Vapour fraction minimum | **494** | Lowest vapour fraction since reset |
| Vapour saturation (relative humidity) | **25** | The fraction of complete saturation the device is using. When the vapour is water, this is the relative humidity |
| Vapour saturation, average | **89** | Average vapour saturation over the time of totalising |
| Vapour saturation, maximum (8v32) | **335** | Highest vapour saturation since reset |
| Vapour saturation, minimum (8v32) | **334** | Lowest vapour saturation since reset |
| Vapour saturation, peak | **121** | Peak vapour saturation over the time of totalising |
| Vapour saturation temperature (dew point) | **496** | Temperature that would give complete vapour saturation. When the vapour is water, this is the dew point |
| Vapour saturation temperature, maximum | **503** | Highest vapour saturation temperature since reset |
| Vapour saturation temperature, minimum | **502** | Lowest vapour saturation temperature since reset |

APPENDIX B

# Engineering units

Commands that take or return a unit use the values below. Which table applies depends on the statistic: for example, use table B-3 for a totaliser that counts standard or normal volume. To change the units of a reading, use set reading engineering units (p. 27).

## B-1 Standard and normal flow units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | `SµL/m` | Standard microlitre per minute |
| **3** | `SmL/s` | Standard millilitre per second |
| **4** | `SmL/m` | Standard millilitre per minute |
| **5** | `SmL/h` | Standard millilitre per hour |
| **6** | `SL/s` | Standard litre per second |
| **7** | `SLPM` | Standard litre per minute |
| **8** | `SL/h` | Standard litre per hour |
| **11** | `SCCS` | Standard cubic centimetre per second |
| **12** | `SCCM` | Standard cubic centimetre per minute |
| **13** | $Scm^3/h$ | Standard cubic centimetre per hour |
| **14** | $Sm^3/m$ | Standard cubic metre per minute |
| **15** | $Sm^3/h$ | Standard cubic metre per hour |
| **16** | $Sm^3/d$ | Standard cubic metre per day |
| **17** | $Sin^3/m$ | Standard cubic inch per minute |
| **18** | `SCFM` | Standard cubic foot per minute |
| **19** | `SCFH` | Standard cubic foot per hour |
| **21** | `SCFD` | Standard cubic foot per day. Added in 6v08.0. |
| **20** | `kSCFM` | 1000 standard cubic feet per minute |
| **32** | `NµL/m` | Normal microlitre per minute |
| **33** | `NmL/s` | Normal millilitre per second |
| **34** | `NmL/m` | Normal millilitre per minute |
| **35** | `NmL/h` | Normal millilitre per hour |
| **36** | `NL/s` | Normal litre per second |
| **37** | `NLPM` | Normal litre per minute |
| **38** | `NL/h` | Normal litre per hour |
| **41** | `NCCS` | Normal cubic centimetre per second |
| **42** | `NCCM` | Normal cubic centimetre per minute |
| **43** | $Ncm^3/h$ | Normal cubic centimetre per hour |
| **44** | $Nm^3/m$ | Normal cubic metre per minute |
| **45** | $Nm^3/h$ | Normal cubic metre per hour |
| **46** | $Nm^3/d$ | Normal cubic metre per day |
| **62** | `Count` | Set point count, 0–64000 |
| **63** | `%` | Percent of the full scale |

## B-2 True mass flow units

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

## B-3 Total standard and normal volume units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | SµL | Standard microlitre |
| **3** | SmL | Standard millilitre |
| **4** | SL | Standard litre |
| **6** | $Scm^3$ | Standard cubic centimetre |
| **7** | $Sm^3$ | Standard cubic metre |
| **8** | $Sin^3$ | Standard cubic inch |

| Value | Label | Description |
|---|---|---|
| **9** | $Sft^3$ | Standard cubic foot |
| **10** | $kSft^3$ | 1000 standard cubic feet |
| **32** | NµL | Normal microlitre |
| **33** | NmL | Normal millilitre |
| **34** | NL | Normal litre |
| **36** | $Ncm^3$ | Normal cubic centimetre |
| **37** | $Nm^3$ | Normal cubic metre |

## B-4 Volumetric flow units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | `µL/m` | Microlitre per minute |
| **3** | `mL/s` | Millilitre per second |
| **4** | `mL/m` | Millilitre per minute |
| **5** | `mL/h` | Millilitre per hour |
| **6** | `L/s` | Litre per second |
| **7** | `LPM` | Litre per minute |
| **8** | `L/h` | Litre per hour |
| **9** | `US GPM` | US gallon per minute |
| **10** | `US GPH` | US gallon per hour |
| **11** | `CCS` | Cubic centimetre per second |
| **12** | `CCM` | Cubic centimetre per minute |
| **13** | $cm^3/h$ | Cubic centimetre per hour |
| **14** | $m^3/m$ | Cubic metre per minute |
| **15** | $m^3/h$ | Cubic metre per hour |
| **16** | $m^3/d$ | Cubic metre per day |
| **17** | $in^3/m$ | Cubic inch per minute |
| **18** | `CFM` | Cubic foot per minute |
| **19** | `CFH` | Cubic foot per hour |
| **21** | `CFD` | Cubic foot per day. Added in 6v08.0. |
| **62** | `count` | Set point count, 0–64000 |
| **63** | `%` | Percent of full scale |

## B-5 Total volume units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | µL | Microlitre |
| **3** | mL | Millilitre |
| **4** | L | Litre |
| **5** | US GAL | US gallon |
| **6** | $cm^3$ | Cubic centimetre |
| **7** | $m^3$ | Cubic metre |
| **8** | $in^3$ | Cubic inch |
| **9** | $ft^3$ | Cubic foot |
| **61** | µP | Micropoise, a measure of viscosity; no conversions are performed to or from other units |

## B-6 Pressure units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | Pa | Pascal |
| **3** | hPa | Hectopascal |
| **4** | kPa | Kilopascal |
| **5** | MPa | Megapascal |
| **6** | mbar | Millibar |
| **7** | bar | Bar |
| **8** | $g/cm^2$ | Gram force per square centimetre |
| **9** | $kg/cm^2$ | Kilogram-force per square centimetre |
| **10** | PSI | Pound-force per square inch |
| **11** | PSF | Pound-force per square foot |
| **12** | mTorr | Millitorr |
| **13** | torr | Torr |
| **14** | mmHg | Millimetre of mercury at 0 °C |
| **15** | inHg | Inches of mercury at 0 °C |
| **16** | $mmH_2O$ | Millimetre of water at 4 °C (NIST conventional) |
| **17** | $mmH_2O$ | Millimetre of water at 60 °F |
| **18** | $cmH_2O$ | Centimetre of water at 4 °C (NIST conventional) |
| **19** | $cmH_2O$ | Centimetre of water at 60 °F |
| **20** | $inH_2O$ | Inch of water at 4 °C (NIST conventional) |
| **21** | $inH_2O$ | Inch of water at 60 °F |

| Value | Label | Description |
|---|---|---|
| **22** | atm | Atmosphere (absolute pressure only, no A/G/D suffix is applied) |
| **61** | V | Volt; no conversions are performed to or from other units (intended only for log-linear absolute pressure sensors, no A/G/D suffix is applied) |
| **62** | count | Set point count, 0–64000 |
| **63** | % | Percent of full scale (no A/G/D suffix is applied) |

## B-7 Temperature units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | °C | Degree Celsius |
| **3** | °F | Degree Fahrenheit |
| **4** | °K | Kelvin |
| **5** | °Ra | Degree Rankine |

## B-8 Time interval units

| Value | Label | Description |
|---|---|---|
| **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. |
| **2** | h:m:s | The value is displayed as hours:minutes:seconds. |
| **3** | ms | Millisecond |
| **4** | s | Second |
| **5** | m | Minute |
| **6** | hour | Hour |
| **7** | day | Day |

APPENDIX C

# Data sources

The source fields of the information assemblies (assembly 111 (p. 17) and assembly 112 (p. 20)) and query reading source (p. 25) return one of these values.

## C-1 Data sources

| Data source | Value | Notes |
|---|---|---|
| Instant display | **1** | Current readings, smoothed for the display |
| Instant serial | **2** | Current readings, smoothed for serial values |
| Totaliser 2 | **25** | Readings taken from the second totaliser |

APPENDIX D

# Gas numbers

Gas numbers are used by set gas (p. 34), by the gas mix configuration assembly 104 (p. 21) and by the gas mix commands, and are returned at byte offset 0 of assemblies 101, 106, 107 and 108. Numbers 236 to 255 are reserved for your own gas mixes (create/update gas mix (p. 34)).

| # | Short name | Long name |
|---|---|---|
| 0 | Air | Air (Clean Dry) |
| 1 | Ar | Argon |
| 2 | $CH_4$ | Methane |
| 3 | CO | Carbon Monoxide |
| 4 | CO2 | 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 | Trichlorofluoro-methane ($CCl_3F$)[2,3] |
| 101 | R-115 | Chloropentafluoro-ethane ($C_2ClF_5$)[2,3] |
| 102 | R-116 | Hexafluoroethane ($C_2F_6$)[2] |
| 103 | R-124 | Chlorotetrafluoro-ethane ($C_2HClF_4$)[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 | Difluoromonochloro-methane ($CHClF_2$)[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-134A[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 |

| # | Short name | Long name |
|---|---|---|
| **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 % $C_4H_{10}$, 1.3 % $N_2$, 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] Corrosion-resistant devices only.

[3] Under the Montreal Protocol and its Kigali Amendment, the production and consumption of these ozone-depleting substances is being, or has been, phased out. Make sure you comply with these regulations before using these gases; the same applies to R-113, R-123 and R-141b.

**ANALYT-MTC Messtechnik GmbH**

Sales, service and calibration

**German Headquarters**
Klosterrunsstraße 18, 79379 Müllheim
Germany
+49 7631 5545
info@analyt-mtc.de

**French Subsidiary**
175 Avenue d'Alsace, 68000 Colmar
France
+33 3 89 41 47 78
info@analyt-mtc.fr

**www.analyt-mtc.com**

AMTC-CG-0004 · Rev. 1 · 2026-10

© 2026 ANALYT-MTC Messtechnik GmbH. All rights reserved. Subject to technical changes without notice.