



edwardsvacuum.com

Polycold Cryochiller Profibus Interface

INSTRUCTION MANUAL

Copyright notice

©Edwards Limited 2023. All rights reserved.

Published: 9/1/2023

Trademark credit

Edwards and the Edwards logo are trademarks of Edwards Limited, Innovation Drive, Burgess Hill, West Sussex RH15 9TW.

Disclaimer

The content of this manual may change from time to time without notice. We accept no liability for any errors that may appear in this manual nor do we make any expressed or implied warranties regarding the content. As far as practical we have ensured that the products have been designed and constructed to be safe and without risks when properly installed and used in accordance with their operating instructions.

We accept no liability for loss of profit, loss of market or any other indirect or consequential loss whatsoever.

Product warranty and limit of liability are dealt with in our standard terms and conditions of sale or negotiated contract under which this document is supplied.

You must use this product as described in this manual. Read the manual before you install, operate, or maintain the product.

Contents

1. Introduction.	4
1.1 Overview.	4
1.2 Polycold implementation.	4
1.3 GSD files and module configuration.	5
2. Data definition.	6
2.1 Polycold data.	6
2.1.1 Data I/O perspective.	6
2.1.2 Reserved values.	6
2.1.3 Outputs.	6
2.1.4 Inputs.	6
2.1.5 Data table (inputs and outputs).	7
2.2 GSD File.	9
2.3 Modules.	9
2.3.1 Module table.	10
2.3.2 Module configuration.	10
2.4 Identification number.	11
2.5 Controller.	11
2.6 Field device (Polycold application).	11
2.7 Baud rate detection.	13
2.8 Handshake.	13
2.8.1 Diagnostics.	13
2.8.2 Parameter.	13
2.8.3 Data exchange.	13

1. Introduction

This document describes the interface between the Profibus field device operating on the Cryochiller and a Profibus controller.

1.1 Overview

Process Field Bus (Profibus), is a standard field bus communication protocol that was developed in Germany. It is most commonly used in Europe.

Field bus is a family of industrial network protocols used for real-time distributed control. A field bus is typically used to link a Programmable Logic Controller (PLC) with individual sensors and valves where the PLC is the controller and the sensors and valves are the field device devices.

A detailed description of Profibus is beyond the scope of this document so a brief description is provided.

- Profibus Distributed Processing (DP), is used to monitor and operate sensors and actuators on field device devices from a controller controller in a factory automation environment.
- There are three versions of Profibus DP: V0, V1 and V2.
- DP V0 provides cyclical exchange of factory data.
- The Polycold Cryochiller field device interface uses DP V0 exclusively.

This protocol differs from the serial interfaces in many ways but in general it is intended to send a stream of data very rapidly between the machines on a factory floor. A data stream containing outputs is sent from the controller to a field device. The field device processes the outputs and returns a data stream consisting of inputs back to the controller.

- Profibus is most suitable for monitoring data such as temperatures.
- Profibus is least suited for configuring equipment such as equipment configuration parameters that are very seldom changed.

1.2 Polycold implementation

The Polycold application contains a Profibus manager that maintains a Profibus field device application. This application has two threads of execution:

- One thread constantly updates and caches data required for the Profibus modules.
- The other thread monitors the Profibus driver waiting for a controller to interrogate the field device for this information.

When there is no controller, the data is updated once per second. When a controller is present, the data is updated frequently but not exceeding the DSP's update rate (once per 100 ms).

1.3 GSD files and module configuration

The controller and field device share a common General Station Description (GSD) file that defines the data that is transferred between the two nodes along with equipment description, protocol configuration and limitations.

2. Data definition

2.1 Polycold data

2.1.1 Data I/O perspective

Data I/O in Profibus is from the perspective of the controller.

Inputs are values being sent from the Polycold field device to a Profibus controller (i.e. a get command).

Outputs are values being sent from the Profibus controller to the Profibus field device (i.e. set command).

The data exchanged between a Profibus controller and the Polycold Profibus field device is listed in [Table: Data \(inputs and outputs\)](#) on page 7.

2.1.2 Reserved values

There are several reserved values that are not used at present but may be added in the future.

Reserving the values allows future software upgrades to include the data in the data exchange without modifying the module configuration. Any GSD file updates will be naming only, which means Polycold updates to the data would be backwards compatible with the original GSD file.

2.1.3 Outputs

Output values (sets) are only processed by Polycold field device if the Polycold SystemControlMode Input is set to Profibus (6). SystemControlMode can only be set from the Polycold unit itself, using the "SYSTEM_CONTROL_MODE" host command, or by making the Polycold unit default to remote control mode (6) from the HMI (TOP MENU->CONFIGURATION->CONTROL CONFIG ->LOCAL/REMOTE->DEF REMOTE MODE) or using "SYSTEM_CONTROL_MODE_DEF" host command.

If the SystemControlMode is not set to Profibus (6), the other outputs are ignored. The Profibus controller commands the Profibus field device to set outputs that Profibus does not have permission to set on the Polycold field device.

The Profibus field device checks the status of outputs before implementing them and only implements the command if it is not already commanded. Example: If the Polycold field device is already in STANDBY mode the Polycold field device will not continuously command the unit to transfer to that mode.

2.1.4 Inputs

Input values are constantly updated and cached by the Profibus field device. These values are immediately returned when the Profibus controller ends a telegram.

2.1.5 Data table (inputs and outputs)

Table: Data (inputs and outputs)

Table 1 Data (inputs and outputs)

I/O type	Data name	Data size	Description or comparable command
Input	ActiveAlarmStatus	4 bytes	Equivalent to: \$ACTIVE_ALARMS,2
Input	SensorValueValidStatus	4 bytes	Equivalent to: \$SENSOR_SNAPSHOT,? Param #1
Input	SensorFaultStatus	4 bytes	Equivalent to: \$SENSOR_SNAPSHOT,? Param #2
Input	TC1	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC2	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC3	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC4	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC5	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC6	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC7	2 bytes	Not Used
Input	TC8	2 bytes	Not Used
Input	TC9	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC10	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC11	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC12	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC13	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC14	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC15	2 bytes	Not Used
Input	TC16	2 bytes	TC in Celsius including 1/10 (ex. 301 = 30.1)
Input	TC17	2 bytes	Not Used
Input	TC18	2 bytes	Not Used
Input	TC19	2 bytes	Not Used
Input	TC20	2 bytes	Not Used
Input	Reserved	2 bytes	n/a
Input	Reserved	2 bytes	n/a
Input	Reserved	2 bytes	n/a
Input	Reserved	2 bytes	n/a
Input	Reserved	2 bytes	n/a
Input	PT1	2 bytes	Pressure (PSI) Suction
Input	PT2	2 bytes	Pressure (PSI) Discharge
Input	PT3	2 bytes	Pressure (PSI) Not Used
Input	Line Voltage	2 bytes	Volts
Input	Reserved	2 bytes	n/a

I/O type	Data name	Data size	Description or comparable command
Input	ValveStatusMask	4 bytes	Equivalent to: \$SENSOR_SNAPSHOT,? Param #36
Input	SystemControlMode	1 byte	Equivalent to: \$SYSTEM_CONTROL_MODE,? Local (0), Ethernet (1), RS232 (2), Legacy(4), DeviceNet (5), Profibus (6), Web Gui (7), Default Control Mode (8), EtherCAT (9), Profinet (10)
Input	System Status	1 byte	0x1 = Compressor Off, 0x3 = Compressor ON 1 st bit is unit state (Always 1), 2 nd bit is compressor state and remaining bits are 0. The values represent ON(1) or OFF(0) for each bit.
Input	Circuit1OperatingStatus	1 byte	Equivalent to: \$OPERATING_MODE,?
Input	Circuit2OperatingStatus	1 byte	Equivalent to: \$OPERATING_MODE,?
Input	24 Volt System Status	1 byte	1 st bit UNIT_OK Unit OK indicator, 1 = okay 2 nd bit PWR_OK Power OK indicator, 1 = ON 3 rd bit RMTCTRL_ACT 1 = remote control is active 4 th bit LCLCTRL_ACT 1 = local control is active 5 th bit STPT_RELAY1 1= Setpoint relay #1 is active 6 th bit STPT_RELAY2 1= Setpoint relay #2 is active 7 th bit Not Used 8 th bit Not Used
Input	24 Volt Legacy Remote	1 byte	1 st bit CKT1_COOL_ACT 1 = cool CKT1 is active 2 nd bit CKT1_DEFR_ACT 1= defrost CKT1 is active 3 rd bit CKT1_DEFCPLT 1= defrost CKT1 is complete 4 th bit CKT2_COOL_ACT 1 = cool CKT2 is active 5 th bit CKT2_DEFR_ACT 1= defrost CKT2 is active 6 th bit CKT2_DEFCPLT 1= defrost CKT2 is complete 7 th bit Not Used 8 th bit Not Used
Input	Reserved	1 byte	n/a
Input	Reserved	1 byte	n/a
Output	Reserved	1 byte	n/a

I/O type	Data name	Data size	Description or comparable command
Output	System Status	1 byte	0x1 = Turn Compressor OFF 0x3 Turn Compressor ON 1 st bit is unit state (Always 1, ignored), 2 nd bit is compressor state and remaining bits must be 0. The only valid settings are ON(1) or OFF(0) for each bit.
Output	Circuit1 Operating Status	1 byte	Equivalent to: \$OPERATING_MODE,x
Output	Circuit2 Operating Status	1 byte	Equivalent to: \$OPERATING_MODE,x
Output	Reserved	1 byte	n/a
Output	Reserved	1 byte	n/a
Output	Reserved	1 byte	n/a
Output	Reserved	1 byte	n/a

2.2 GSD File

The GSD file defines the equipment and the data that is exchanged. The GSD file is created for the field device device and provided to the controller device. It allows the controller to know how to communicate with the field device.

The key values for Polycold that are registered with Profibus international organisation are listed in the [Table: GSD file](#). These are defined in the GSD file and must not be changed.

Table 2 GSD file

GSD File Attribute	Value	Description
Vendor_Name	-	Company name
Model_Name	MaxCool	Equipment model name
Revision	0.0.3	Registered with Profibus International revision corresponds to GSD
Ident_Number	-	Permanently assigned identification for all Polycold units

2.3 Modules

The GSD file data is grouped in modules. A primary goal is to eliminate or greatly minimise the requirement for updating the GSD file and application software.

Each module has a configuration value that indicates the size of the data it contains and if it is input or output. Some modules are reserved and could be populated in later versions of the Polycold field device. The updates would be to add descriptive names in place of the reserved names. The original GSD will still function correctly.

There are nine modules of data in the MaxCool1.GSD file that must be loaded in order on the controller. These modules correspond to the GSD file and are show below taken directly from the GSD file.

All 9 MaxCool1.GSD modules must be configured in the order shown in the [Table: Module](#). MaxCool Cryochiller Profibus implementation is not a modular station.

2.3.1 Module table

Ordered list of modules in the MaxCool1.GSD.

Table 3 Module

Order	Module Name	Values
1	Get_BitStatus	AlarmStatus, SensorValueValidStatus, Sensor-FaultStatus
2	Get_TC1-TC8	TC1 - TC8
3	Get_TC9-TC16	TC9 - TC16
4	Get_TC17_TC20	TC17 - TC 20 and Reserved1
5	Reserved_1	Reserved2-5
6	Get_Pressure	PT1 - PT3
7	Get_Voltage_Valves	LineVoltage, Reserved6, ValveStatusMask
8	GetSystemStatus	SysControlMode, SysStatus, Circ1OpStatus, Circ2OpStatus, 24 V_System, 24 V_Circuit, Reserved 9-10
9	SetSystemStatus	ReservedOut1, SetSysStatus, SetCirc1OpStatus, SetCirc2OpStatus, ReservedOut2-5

2.3.2 Module configuration

Each module has a configuration value that defines the size and type of data and if the data is output or input. The configuration is defined in the GSD File with each module.

Example from GSD file where 0x27 is the configuration for module 9:

Module="SetSystemStatus" 0x27

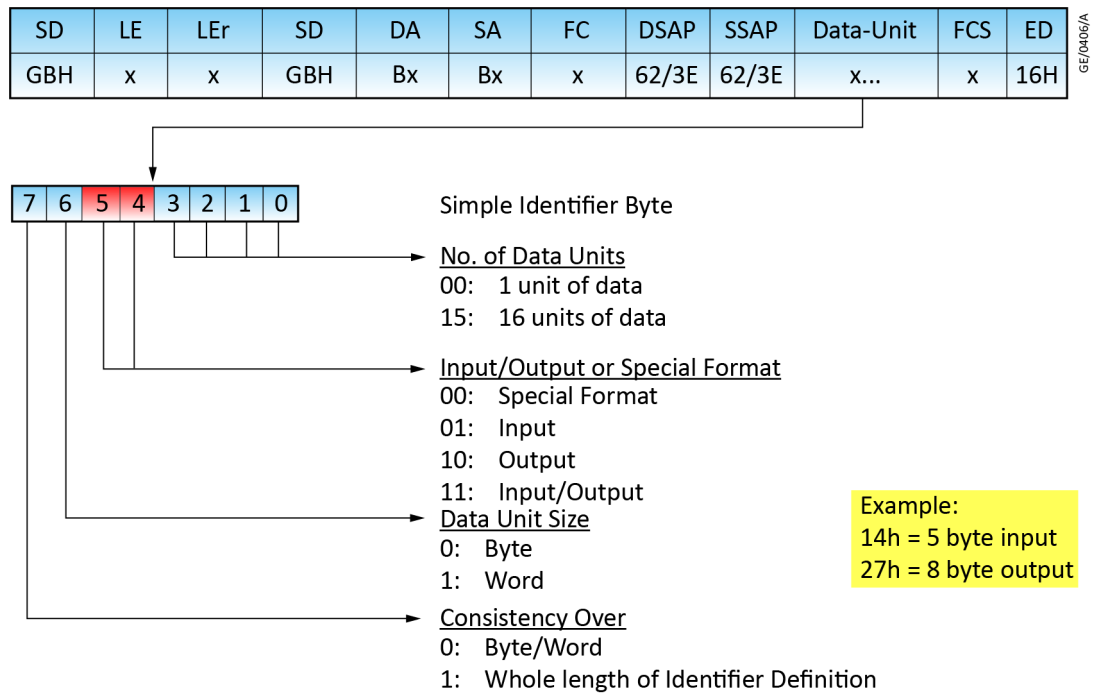
The complete configuration for all 9 modules shown below is:

"0x1B, 0x57, 0x57, 0x54, 0x17, 0x52, 0x53, 0x17, 0x27"

The individual module configuration value is referred to as the data unit in a Profibus telegram and is calculated based on the consistency, data unit size, I/O, and size as shown in [Figure: Module configuration](#).

If the values change, it will require a modification to Profibus field device software and the GSD file. This can be avoided by using reserved values in the GSD with data types that are most likely to expand.

The Profibus field device application has the capability to modify the module configuration without a software update and it saves the configuration in nonvolatile memory. This allows issuing a modified GSD file without changing the field device application image but will not happen as the field device application need to provide the data that matches the GSD file. This is a safety mechanism for some unforeseen failure.

Figure 1 Module configuration

2.4 Identification number

Polycold has a unique identifier called Ident_Number in the GSD file assigned by the Profibus International organisation that identifies the equipment and the manufacturer. The identification number is known by the system's Profibus driver and by the controller from the GSD file. The number must match otherwise the communication will fail.

2.5 Controller

A Profibus controller can communicate with the Polycold field device by importing the Polycold GSD file and configuring the 9 modules identified in this document and in the GSD file in the order they are listed.

All 9 MaxCool1.GSD modules must be configured in the order shown in the [Table: Module](#). MaxCool Cryochiller Profibus implementation is not a modular station.

The Profibus controller must set the field device address to a value that matches the field device it wishes to communicate with. The default Profibus field device application address is 7.

The Profibus controller may set the speed to any value up to a maximum of 12 Mbps.

The controller may then send telegrams using the configuration from the GSD file to monitor the Polycold field device. The outputs and inputs are described in [Polycold data](#) on page 6.

2.6 Field device (Polycold application)

The Polycold field device is part of the Polycold application and is always running if the Polycold unit has been configured for the Profibus interface.

The Polycold field device's address is configurable by the PROFIBUS_PARAMS, ADDRESS,n host command. The 'n' parameter is the desired address or a '?' to query the current address. The Profibus Field device driver is reset if the address is changed. The address must be unique on the Profibus network.

The PolyCold field device's address is also configurable via the HMI menu. Follow the path of TOP MENU -> CONFIGURATION SET COMM PARAMS -> PROFIBUS -> ADDRESS. This is available on software versions 1.3.0.2 and above. Available options are dependent on software version.

 Note:

The PolyCold field device's address, should only be changed while there is no active controller connection/data transfer to the PolyCold field device.

To ensure a correct address change, it is strongly recommended that the Profibus cable connection be removed from the PolyCold field device unit while the PolyCold field device Address is being changed. Once the PolyCold field device Address has been changed, the Profibus cable may be reconnected to the PolyCold unit.

The current Profibus network speed that the Profibus field device is using can be queried via the PROFIBUS_PARAMS,SPEED,? host command. The Response to PROFIBUS_PARAMS,SPEED,? is always A for Auto.

The Profibus field device uses a Profibus driver and is in a state of waiting to receive a command from a controller. When no Profibus controller is present, the Profibus field device updates its data infrequently and uses very few system resources.

When a Profibus controller is present on the network, the Profibus field device reads the controller's request and handles the hand shake at the driver level. The Profibus field device application responds when it receives the Profibus controller's request by processing the outputs and returning the inputs.

Command 'PROFIBUS_PARAMS' Profibus

Purpose	Get the Profibus Speed and Get/Set the Profibus Field device Address	
Parameter(s)	,SPEED,? or ,ADDRESS,? or ,ADDRESS,[address] [speed] is READ ONLY: A = Auto [address] is 1 - 125	
Published	Yes	
Return Value	[A] or [A][speed] or [A][address]	
Default Value	[speed] = A [address] = 7	
Security Level	Get: 1	Set: 3
Example	"PROFIBUS_PARAMS,SPEED,?" Get the Profibus Field device Speed	
Interface	RS-232, Ethernet	
Min: [speed] = A Read Only [address] = 1	Max: [speed] = A Read Only [address] = 125	

2.7 Baud rate detection

The Profibus field device provides automatic baud rate detection through Anybus module and supports baud rates up to 12 Mbps. The Profibus controller may use any baud rate up to maximum 12 Mbps.

2.8 Handshake

2.8.1 Diagnostics

The Profibus field device uses a 6 byte diagnostic telegram that is handled by Anybus module and has no application specific data. An exchange between Profibus controller and Profibus field device is done to determine if the Anybus module is ready to communicate.

2.8.2 Parameter

After diagnostics are complete, the Profibus controller sends the module configuration telegram. It contains the module configuration of the data that it wants to exchange based on the GSD file definitions. The Profibus field device validates the data it expects to progress to the next step.

2.8.3 Data exchange

After the parameters have been validated, the Profibus controller send the outputs (refer to the [Table: Data \(inputs and outputs\)](#)) to the Profibus field device. The Profibus field device read the outputs and return the inputs from the data table that also align with the outputs in the GSD file.

This page has been intentionally left blank.

This page has been intentionally left blank.

