吉思特仪器
GAS TOOL

# GT230 Series

## Pressure Controllers User Manual

# Table of contents

# 1. Overview

## Introduction

Tianjin GASTOOL Instrument Co., Ltd. is a manufacturer specializing in flowmeters/controllers and pressure controllers. Providing reliable solutions for various applications, widely used in gas chromatography, analytical equipment, material equipment, biomedicine, semiconductors and other fields.

GT230 series pressure controller can be customized according to customer needs and is more suitable for the customer's use environment.

## Features

- High-precision, high-resolution control, control accuracy **0.001PSI (0-150 PSI range control accuracy is 0.01 PSI)**
- Power less than 2W
- 50×24×55mm (L × W × H) package size
- Digital control, RS-485 Modbus RTU communication mode
- Control Range -15PSIG...150PSIG
- Provides various controller models such as pressure reducing type and back pressure type
- High repeatability and reliability
- Using electromagnetic proportional valve, i, long service life and a fast response speed
- Built-in temperature compensation
- Internal closed-loop control with high stability

## Range and Accuracy

- Built-in high-precision sensor, accuracy up to ± 0.25% FS
- GT230 series control accuracy up to 0.001 PSI
- Provide various ranges: -1...1PSIG/ -15...15PSIG/0...60PSIG/0...150PSIG

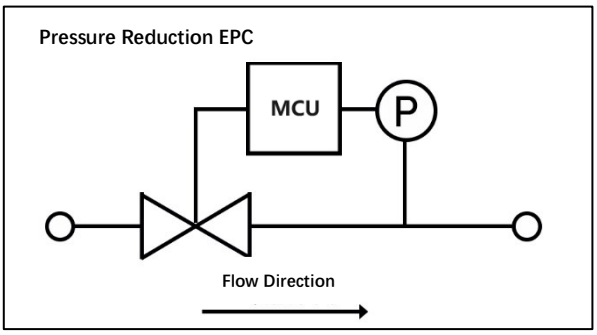
## Digital Design

GT230 series pressure controllers are equipped with high-precision pressure sensors and digital circuits as standard . The digital circuits can complete all basic functions required for measurement and control. GT230 series uses standard RS -485 Modbus RTU communication mode. RS-485 bus can be expanded to up to 255 devices at the same time.

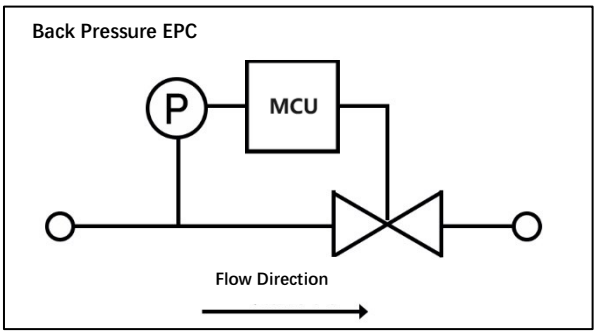
## Application Areas

- Gas and liquid chromatographs
- Semiconductor Process
- Environmental monitoring equipment (such as VOCS, etc.)
- Micro- reactor and material preparation experimental device (CVD)
- Reactor pressure control
- Gas generator
- Gas flow control

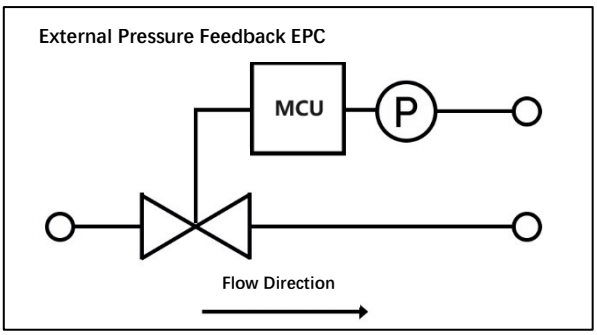
## 2. Control Principle



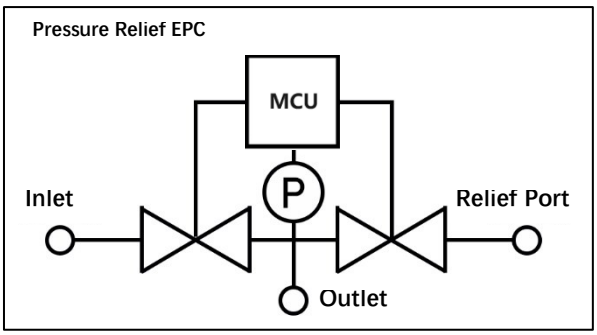

GT230 pressure reducing type pressure controller




GT230B back pressure controller




GT232 external pressure type pressure controller

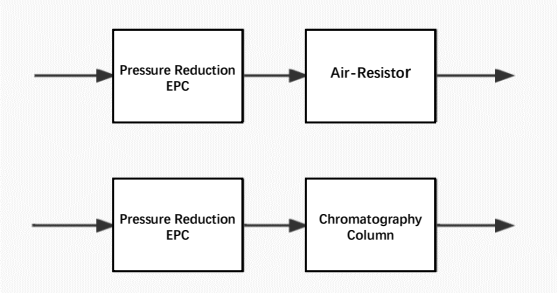



GT233 process control (pressure relief) pressure controller

# 3. Application Examples

There are many types of gas pressure control solutions, which cannot be listed one by one. The following are some common control solutions. Since the actual application environment is different, the following cases are for reference only. At the same time, you are welcome to call for consultation, and our staff will provide you with better ideas and solutions.

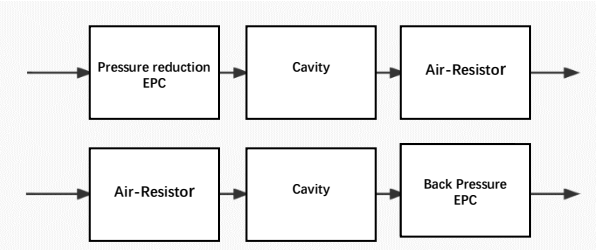
## 1. Pressure Controller for Constant Flow Control




In the fields of gas chromatography, analytical testing equipment, biomedicine, etc., where micro-flow or constant flow control is required, a pressure controller can be used to control the pressure at the front end of the flow-limiting device (gas resistance, chromatographic column, etc.) to obtain a constant flow through the flow-limiting device (gas resistance, chromatographic column, etc.).

✱ The recommended gas resistance is the GT550R series two-way gas resistance, which has a wide variety and is easy to use. For details, please visit www.gastool.cn or call for consultation.

## 2. Cavity Constant Pressure Control




The cavity pressure control can choose pressure reducing type and back pressure type pressure controller according to the application scenario. During use, it is necessary to add a flow limiting device, which can be a gas resistance or a flow controller.

✱ The current limiting device can be selected from GT300 series flow controller or GT550R series two-way gas resistor. Please visit gastool.cn or call for consultation.

# 4. Technical Parameters

## Physical Parameters

| | |
|---|---|
| **Valve** | Proportional valve |
| **Through the Medium** | Non-corrosive gas |
| **Operating Temperature** | 0 – 50 ℃ |
| **Temperature Compensation Range** | 0 - 50 ℃ |
| **Storage Temperature** | -40 ~ 85 ℃ |
| **Leakage Rate** | < 0.020SCCM He (pressure difference 150PSI) |
| **Length × Width × Height** | 50 × 24 × 55 mm |

## Electrical Parameters

| | |
|---|---|
| **Operating Voltage** | DC 12V±10% / 24V±10% (depending on the model) |
| **Maximum Operating Current** | < 200mA |
| **Power** | < 2W |
| **Digital Control Signal** | RS-485 MODBUS RTU |
| **Analog Control Signal** | Analog voltage signal 0 - 5V |

## Run Parameters

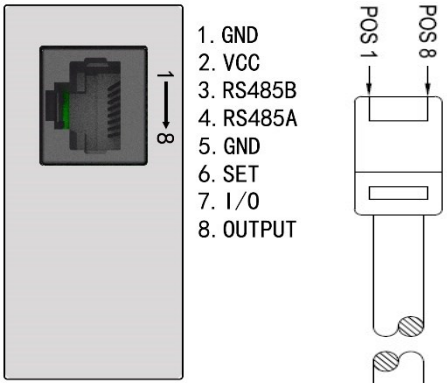
| | |
|---|---|
| **Pressure Range** | - 1 ~ 1PSIG / -15~15 PSIG / 0~60 PSIG / 0 ~ 150 PSIG |
| **Pressure Accuracy** | < ±0.25 % FS |
| **Repeatability** | < ±0.25 % FS |
| **Control Accuracy** | 0.001PSI (150PSI range control accuracy 0.01PSI) |
| **Response Time** | < 100ms (depending on inlet pressure) |
| **Pressure Resistance** | 150 PSID |

* PSIG stands for gauge pressure, which is relative to ambient air pressure; PSID stands for differential pressure, which is the pressure difference between the inlet and outlet of the equipment.

## 5. Electrical Configuration

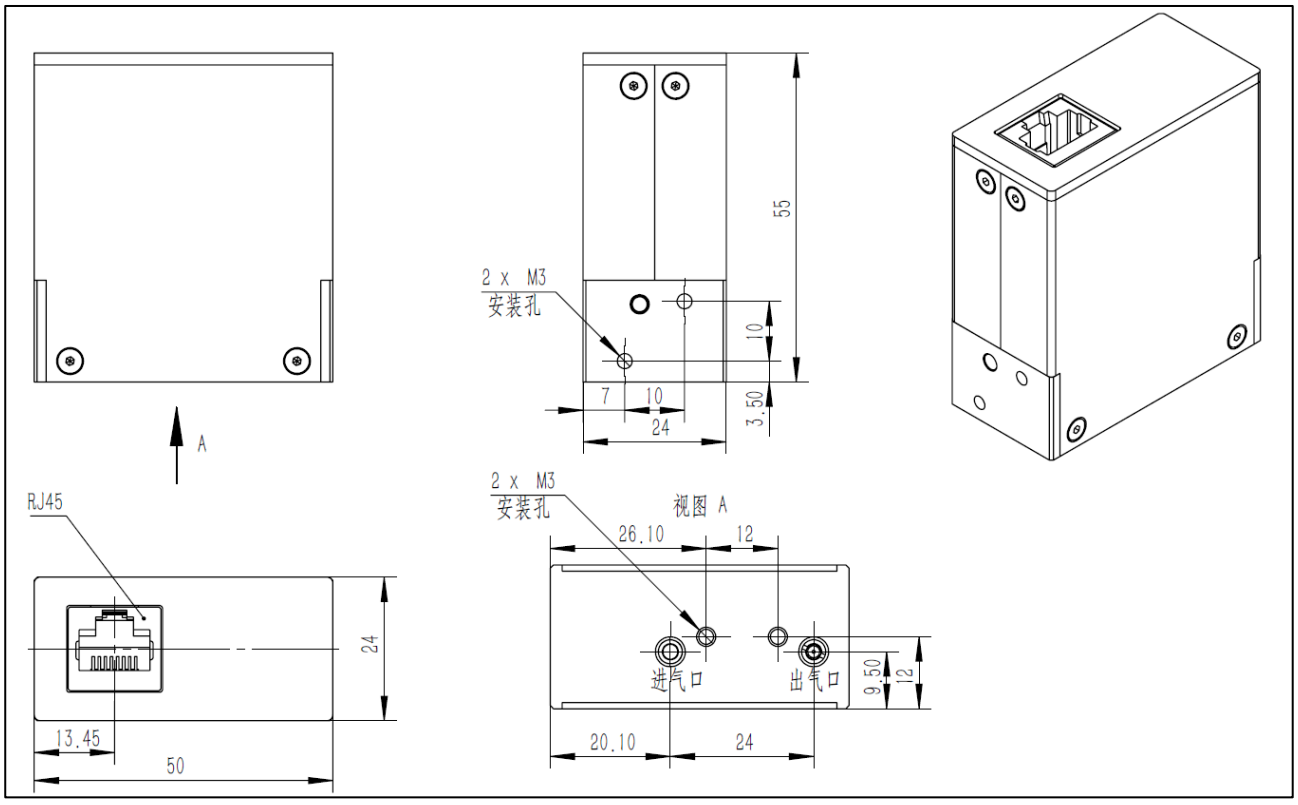
| Dot Number | Line Color | Line Number Definition | Parameter |
|---|---|---|---|
| 1 | White Orange | GND | Power Ground |
| 2 | Orange | VCC | Power supply DC +12V/+24V |
| 3 | White Green | RS485B | Digital signal RS485B/- |
| 4 | Blue | RS485A | Digital signal RS485A/+ |
| 5 | White and blue | GND | Analog signal reference ground |
| 6 | Green | SET | Analog signal setting ( 0-5V ) |
| 7 | White Brown | I/O | Digital signal selection (floating or DC3.3V )<br>Analog signal selection (connected to pin 5) |
| 8 | Brown | OUTPUT | Analog signal output ( 0-5V ) |

Note: Switching between analog and digital signals requires power off.




## 6. Size Structure

Unit: mm




GT230 Dimensions

# 7. Modbus Communication

## 7.1 Quick Start

The Modbus - RTU standard communication protocol is used , including CRC 16 checksum. This section is a quick start communication protocol. For detailed communication protocol, please refer to 7.2 .
Baud rate: Depends on the device settings, the product default baud rate is 9600
Parity bit: none
Data bits: 8 bit
Stop bit: 1 bit
Data communication format: hexadecimal

Commonly used communication protocols are as follows:

- **Reading Pressure**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | | CRC-16 |
|---|---|---|---|---|---|---|
| | 0xXX | 0x04 | 0x0001 | 0x0002 | | 0xXXXX |
| Return Data | Address | Function Code | Number of Bytes | Low Byte | High Byte | CRC-16 |
| | 0xXX | 0x04 | 0x04 | 0xXXXX | 0xXXXX | 0xXXXX |

Description: This command can be used to obtain the current EPC actual pressure. The pressure value is a hexadecimal floating point number.
Example: Read the current pressure value of EPC at address 1,20 (hexadecimal floating-point number: 41 A0 00 00):
Send data: 01 04 00 01 00 02 20 0B
Return data: 01 04 04 00 00 41 A0 CB AC

- **Setting Pressure**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Low Byte Data | High Byte Data | CRC-16 |
|---|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x000B | 0x0002 | 0x04 | 0xXXXX | 0xXXXX | 0xXXXX |
| Return Data | Address | Function Code | Starting Register Address | Number of Registers | | | | CRC-16 |
| | 0xXX | 0x10 | 0x000B | 0x0002 | | | | 0xXXXX |

Description: This command can be used to set the pressure setting value, which is a hexadecimal floating point number.

✱ When the memory function is turned on, a configuration information burning command needs to be issued, and the EPC will burn and save the configuration information.

Example: Set the EPC pressure value at address 1 to 30 ( hexadecimal floating point number: 41 F0 00 00 ):
Send data: 01 10 00 0B 00 02 04 00 00 41 F0 82 08
Return data: 01 10 00 0B 00 02 30 0A

- **Read the Set Pressure Value**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | | Number Of Data Bytes |
|---|---|---|---|---|---|---|
| | 0xXX | 0x03 | 0x000B | 0x0002 | | 0xXXXX |
| Return Data | Address | Function Code | Number of Bytes | Number of Registers | CRC-16 | Return Data |
| | 0xXX | 0x03 | 0x04 | 0xXXXX | 0xXXXX | 0xXXXX |

Description: This command can be used to obtain the current EPC pressure setting value, which is a hexadecimal floating point number.

Example: Read the current pressure setting value of the EPC at address 1 as 30 (hexadecimal floating point number: 41 F0 00 00 ):

Send data: 01 03 00 0B 00 02 B5 C9

Return data: 01 03 04 00 00 41 F0 CA 27

- **Zero Setting**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | Number Of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x0006 | 0x0001 | 0x02 | 0x0001 | 0xXXXX |
| Return Data | Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x0006 | 0x0001 | | | 0xXXXX |

Description: Issue this command and the device will adjust the zero point by itself. When using this command, you first need to confirm that no gas passes through the product.

Example: reset the EPC address to 01 :

Send data: 01 10 00 06 00 01 02 00 01 67 F6

Return data: 01 10 00 06 00 01 E1 C8

- **Changing the Pressure Units**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | Number Of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x0005 | 0x0001 | 0x02 | 0xXXXX | 0xXXXX |
| Return Data | Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x0005 | 0x0002 | | | 0xXXXX |

Description: This command can be used to change the pressure unit used by EPC. 0x 0000 represents PSI and 0x 0001 represents kPa .

※ When the changes are completed and need to be saved, a configuration information burning command needs to be issued, and EPC will burn and save the configuration information.

EPC address 01 to kPa unit :

Send data: 01 10 00 05 00 01 02 00 01 67 C5

Return data: 01 10 00 05 00 01 11 C8

- **Read the Current Pressure Unit**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | CRC-16 |
|---|---|---|---|---|---|
| | 0xXX | 0x 03 | 0x0005 | 0x0001 | 0xXXXX |
| Return Data | Address | Function Code | Number Of Bytes | Data | CRC-16 |
| | 0xXX | 0x03 | 0x02 | 0xXXXX | 0xXXXX |

Description: This command can be used to obtain the pressure unit currently used by the EPC. The data is a hexadecimal integer. 0x 0000 represents PSI and 0x 0001 represents kPa .

Example: Read the current pressure unit of EPC at address 1 in kPa (hexadecimal floating point number: 00 01 ):

Send data: 01 03 00 05 00 01 94 0B

Return data: 01 03 02 00 01 79 84

- **Change the Pressure Control Mode (Gauge Pressure Control/Absolute Pressure Control)**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x000F | 0x0001 | 0x02 | 0xXXXX | 0xXXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x000F | 0x0002 | | | 0xXXXX |

Description: This command can be used to change the pressure type used by EPC. 0x 0000 represents gauge pressure control, and 0x 0001 represents absolute pressure control .

✱ When the changes are completed and need to be saved, a configuration information burning command needs to be issued, and EPC will burn and save the configuration information.

Example: Change the EPC with address 01 to absolute pressure control mode :

Send data: 01 10 00 0F 00 01 02 00 01 67 6F

Return data: 01 10 00 0F 00 01 31 CA

- **Read the Current Pressure Control Mode**

| Send Data | Address | Function Code | Starting Register Address | Number of Registers | CRC-16 |
|---|---|---|---|---|---|
| | 0xXX | 0x03 | 0x000F | 0x0001 | 0xXXXX |
| Return Data | Address | Function Code | Number Of Bytes | Low Byte Data | High Byte Data |
| | 0xXX | 0x03 | 0x02 | 0xXXXX | 0xXXXX |

Description: This command can be used to obtain the current EPC pressure type. The data is a hexadecimal integer. 0x 0000 represents gauge pressure control, and 0x0001 represents absolute pressure control .

Example: Read the current pressure type of the EPC at address 1 as absolute pressure control (hexadecimal floating point number: 00 00 ):

Send data: 01 03 00 0F 00 01 B4 09

Return data: 01 03 02 00 00 B8 44

- **Change of Address**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x0003 | 0x0001 | 0x02 | 0xXXXX | 0xXXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x0003 | 0x0001 | | | 0xXXXX |

Note: The factory default address is 01. Use this command if you need to change the address. After receiving the reply, power off and restart the device to complete the address change. If the power is not turned off and restarted, the communication uses the previous address. The address range is 1-255 .

✱ When the changes are completed and need to be saved, a configuration information burning command needs to be issued, and EPC will burn and save the configuration information.

Example: Change the EPC address from 01 to 05 :

Send data: 01 10 00 03 00 01 02 00 05 66 60

Return data: 01 10 00 03 00 01 F1 C9

- **Changing the Baud rate**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x0004 | 0x0001 | 0x02 | 0xXXXX | 0xXXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x0004 | 0x0001 | | | 0xXXXX |

Note: The factory default baud rate is 9600. If you need to change the baud rate, use this command. After receiving the reply, power off and restart the device to complete the baud rate change. If the power is not turned off and restarted, the communication uses the previous baud rate. The data is a hexadecimal integer. The sent data is the actual baud rate reduced by 100 times , that is, the sent data = actual baud rate/100. The baud rate can be customized. The commonly used baud rates are shown in the table:

| Serial number | Baud rate | Sending Data |
|---|---|---|
| **1** | 9600 | 96 |
| **2** | 14400 | 144 |
| **3** | 19200 | 192 |
| **4** | 38400 | 384 |
| **5** | 57600 | 576 |
| **6** | 115200 | 1152 |

✱ When the changes are completed and need to be saved, a configuration information burning command needs to be issued, and EPC will burn and save the configuration information.

Example: Set the baud rate of the EPC with address 1 from 9600 to 115200 :

Send data: 01 10 00 04 00 01 02 04 80 A4 B4

Return data: 01 10 00 04 00 01 40 08

- **Adjust Valve Control Switch Status**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x000D | 0x0001 | 0x02 | 0x000X | 0xXXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x000D | 0x0001 | | | 0xXXXX |

Description: This command can be used to realize valve control switch operation. The data is a hexadecimal integer. The relationship between data and function is shown in the table:

| Function | data |
|---|---|
| Valve fully closed | 0x0000 |
| Valve fully open | 0x0001 |
| Closed loop automatic control | 0x0002 |

Example: Fully close the EPC valve with address 01 :

Send data: 01 10 00 0D 00 01 02 00 00 A7 4D

Return data: 01 10 00 0D 00 01 90 0A

- **Memory Function On and Off**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x000E | 0x0001 | 0x02 | 0x000X | 0xXXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x000E | 0x0001 | | | 0xXXXX |

Description: This command can be used to turn the memory function on and off. The data is a hexadecimal integer. The relationship between data and function is shown in the table:

| Function | data |
|---|---|
| Memory function off | 0x0000 |
| Memory function on | 0x0001 |

✱ When the changes are completed and need to be saved, a configuration information burning command needs to be issued, and EPC will burn and save the configuration information.

Example: EPC memory function at address 01 :

Send data: 01 10 00 0E 00 01 02 00 01 66 BE

Return data: 01 10 00 0E 00 01 60 0A

- **Configuration Information Burning**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0xXX | 0x10 | 0x0006 | 0x0001 | 0x02 | 0x0004 | 0xXXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0xXX | 0x10 | 0x0006 | 0x0001 | | | 0xXXXX |

Description: After issuing this command, EPC will burn and save the configuration information. The configuration information that needs to be burned and saved is as follows:

| Configuration Information |
|---|
| Set pressure (with memory function) |
| Address |
| Baud rate |
| Pressure Unit |
| Valve controlled switch |
| Memory function |
| Pressure control method |

Example: Save the configuration parameters of the EPC with address 01 :
Send data: 01 10 00 06 00 01 02 00 04 A7 F5
Return data: 01 10 00 06 00 01 E1 C8

- **Restore Factory Settings**

| Send Data | Device Address | Function Code | Starting Register Address | Number of Registers | Number of Data Bytes | Data | CRC-16 |
|---|---|---|---|---|---|---|---|
| | 0x XX | 0x 10 | 0x0006 | 0x0001 | 0x02 | 0x 0005 | 0x XXXX |
| Return Data | Device Address | Function Code | Starting Register Address | Number of Registers | | | CRC-16 |
| | 0x XX | 0x 10 | 0x0006 | 0x0001 | | | 0x XXXX |

Description: After this command is issued, the EPC is restored to factory settings.
Example: EPC address 01 to factory settings :
Send data: 01 10 00 06 00 01 02 00 05 66 35
Return data: 01 10 00 06 00 01 E1 C8

## 7.2 Communication Protocol Detail

### 7.2.1 Data Format

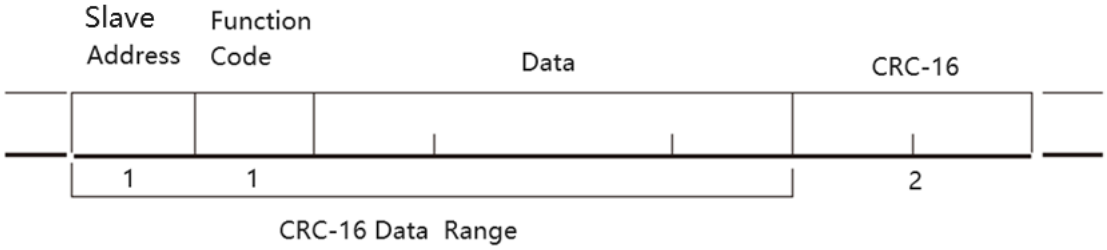
According to the Modbus (RTU) communication procedure, commands from the host computer and responses from the EPC are in units of data sets called frames.

The structure of the command frame and response frame is as follows.

In the following descriptions, if a value is preceded by " 0x ", it means a hexadecimal number, such as " 0x02 ".

**1. Command Frame**

In RTU mode, the signal must start with a quiet interval of at least 3.5 characters and end with a quiet interval of at least 3.5 characters.




| minimum quiet interval of 3.5 character times | |
|---|---|
| Slave Address | Please specify a Unit Number.<br>0x00 to 0x63 (0 to 99) can be set in hexadecimal format .<br>For unified broadcast, specify 0x00 .<br>But no response is returned during broadcast. |
| Function Code | The function code is a code indicating the command type of the host device and is set using 1 byte in hexadecimal format . |
| Data | The text body corresponding to the function code.<br>Used to specify variable addresses, parameter values, etc. ( set in hexadecimal format ) |
| CRC-16 | Cyclic Redundancy Check<br>The checksum is calculated from the value from the slave address to the end of the data.<br>Use 2 bytes in hexadecimal format . |
| minimum quiet interval of 3.5 character times | |

- **CRC Check**

Information is processed byte by byte in a calculation workpiece (16 -bit register: hereinafter referred to as CRC register ) .

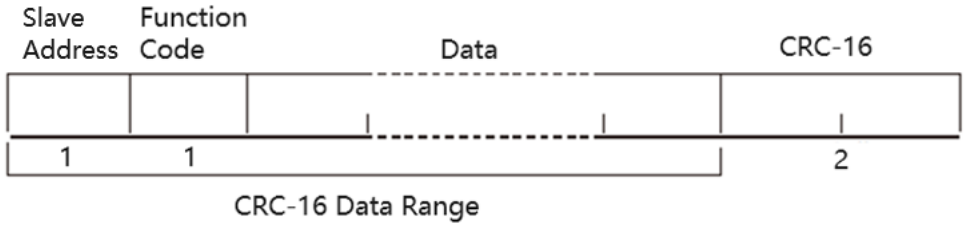
(1) Set the initial value of the CRC register to 0xFFFF .
(2) XOR operation on the CRC register and the first byte of the information , and return the calculation result to the CRC register.
(3) Fill the MSB with " 0 " and shift the CRC register right by 1 bit.
(4) If the bit shifted from the LSB is " 0 ", step (3) is repeated ( processing the next shift ) .
If the bit shifted from the LSB is " 1 ", an XOR operation is performed on the CRC register and 0xA001 , and the result is returned to the CRC register.
(5) Repeat steps (3) and (4) until 8 bits are shifted.
(6) If the information processing has not been completed, an XOR operation is performed on the CRC register and the next byte of the information , and the result is returned to the CRC register, and the process is repeated from step (3) .
(7) Append the calculated result ( the value of the CRC register ) to the message from the low-order byte.

**Calculation Example:**

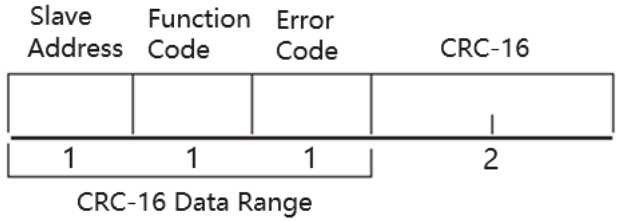
Data to be calculated: 01 03 04 00 00 41 A0
Calculation result: 1B CA
Send data: 01 03 04 00 00 41 A0 CA 1B

## 2. Response Frame

- **Normal Response Frame**




- **Response Frame When Exception Occurs**




| | |
|---|---|
| **Slave Address** | Use the number specified via the command frame directly.<br>The unit number of the response was returned. |
| **Function Code** | The function code of the received signal.<br>However, the response frame in case of exception is the value of " 0x80 " added to the function code that received the signal, indicating an abnormal response.<br>example: Receive function code = 0x03<br>Function code in the response frame when responding to exception = 0x83 |
| **Error Code** | An end code indicating the abnormal contents. |
| **CRC-16** | Cyclic Redundancy Check<br>The checksum is calculated from the value from the slave address to the end of the data.<br>Use hexadecimal format and 2 bytes. |

## 3. Error Code

| Error Code | Name | Error Level | Description |
|---|---|---|---|
| 0x01 | Configuration Data Exception | Generally | Configuration data is abnormal |
| 0x02 | Configuration Data Exception | Serious | Configuration data is abnormal |
| 0x07 | Pressure Setting Value Exceeded the Limit | Generally | If the pressure setting value exceeds the maximum range, the value will be refreshed to the maximum range value. |
| 0x08 | Pressure Exceeding Limit | Generally | The current pressure exceeds the maximum range of the device hardware |
| 0x09 | Wrong Direction | Generally | Pressure less than -5%* maximum range |
| 0x0C | Configuration Register Programming Error | Generally | Configuration register programming error |
| 0x0D | Cache Register Programming Error | Generally | Cache register programming error |
| 0x10 | Sensor Reading Error | Serious | Sensor communication abnormality |

- **No Response**

In the following cases, the received command is not processed and no response is returned. Therefore, the host device is in timeout state.

· When the slave address receiving the instruction is inconsistent with the communication address number;

· When parity error or frame error occurs due to transmission error, etc.

· When a CRC-16 code error occurs in the received command frame ;

· When the reception time interval of each data constituting the command frame is vacant for more than 3.5 character times.

In addition, in the following cases, there is no response even though the process ( target function ) is executed .

· When broadcast ( slave address: 0x00) is specified .

## 7.2.2 Function Code

The function codes are shown in the following table.

| Function code | Name | Description |
|---|---|---|
| **0x03** | Read Configuration Registers | Read the configuration register area.<br>Multiple contiguous configuration register regions can be read. |
| **0x04** | Read Status Register | Read the status register area.<br>Multiple consecutive status register areas can be read |
| **0x06** | Writing to a Single Configuration Register | Writes to a single configuration register area. |
| **0x10** | Writing to Multiple Configuration Registers | Write to the configuration register area.<br>Multiple contiguous configuration register areas can be written. |

### 1. Read Configuration Register Operation (0x03)

- **Command Frame**

| Slave Address | Function Code | Starting Register Address | Number of Registers | CRC-16 |
|---|---|---|---|---|

| Data name | Length (Bytes) | Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function Code** | 1 | 0x03 | Read configuration register function code. |
| **Starting Register Address** | 2 | 0x0000-0xFFFF | The first configuration register address among the configuration registers to be read. |
| **Number of Registers** | 2 | 0x0000-0x00FF | The number of configuration registers to read. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Response Frame**

| Slave Address | Function Code | Number of Bytes | Register Data Value | CRC-16 |
|---|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function Code** | 1 | 0x03 | Read configuration register function code. |
| **Number of Bytes** | 1 | 0x00-0xFF | The total number of bytes of configuration register data read. |
| **Register Data** | Number of | - | The configuration register data to be read. |

| | | | |
|---|---|---|---|
| **Value** | bytes | | |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Error response Frame**

| Slave Address | Error Function Code | Error Code | CRC-16 |
|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Error Function Code** | 1 | 0x83 | Read configuration register error function code. |
| **Error Code** | 1 | 0x00-0xFF | Query the error condition based on the error code. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

## 2. Read Status Register Operation (0x04)

- **Command Frame**

| Slave Address | Function Code | Starting Register Address | Number of Registers | CRC-16 |
|---|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function Code** | 1 | 0x04 | Read the status register function code. |
| **Starting Register Address** | 2 | 0x0000-0xFFFF | The first status register address in the status registers to be read. |
| **Number of Registers** | 2 | 0x0000-0x00FF | The number of status registers to read. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Response Frame**

| Slave Address | Function Code | Number of Bytes | Register Data Value | CRC-16 |
|---|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function Code** | 1 | 0x04 | Read the status register function code. |

| | | | |
|---|---|---|---|
| **Number of Bytes** | 1 | 0x00-0xFF | The total number of bytes of status register data read. |
| **Register Data Value** | Number of bytes | - | The status register data to be read. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Error Response Frame**

| | | | |
|---|---|---|---|
| Slave Address | Error Function code | Error Code | CRC-16 |

| Data name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Error Function Code** | 1 | 0x84 | Read status register error function code. |
| **Error Code** | 1 | 0x00-0xFF | Query the error condition based on the error code. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

### 3. Write Single Register Operation (0x06)

- **Command Frame**

| | | | | |
|---|---|---|---|---|
| Slave Address | Function Code | Register Address | Register Data Value | CRC-16 |

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function code** | 1 | 0x06 | Write a single configuration register function code. |
| **Register Address** | 2 | 0x0000-0xFFFF | The configuration register address to be written. |
| **Register Data Value** | 2 | 0x0000-0xFFFF | The data value to be written to the configuration register. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Response Frame**

| | | | | |
|---|---|---|---|---|
| Slave Address | Function Code | Register Address | Register Data Value | CRC-16 |

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function code** | 1 | 0x06 | Write a single configuration register function code. |
| **Number of Bytes** | 2 | 0x0000-0xFFFF | The configuration register address to be written. |
| **Register Data Value** | 2 | 0x0000-0xFFFF | The data value to be written to the configuration register. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Error Response Frame**

| Slave Address | Error Function Code | Error Code | CRC-16 |
|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Error Function Code** | 1 | 0x86 | Write configuration register error function code. |
| **Error Code** | 1 | 0x00-0xFF | Query the error condition based on the error code. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

## 4. Write Multiple Register Operations (0x10)

- **Command Frame**

| Slave Address | Function Code | Starting Register Address | Number of Registers | Number of Bytes | Register Data Value | CRC-16 |
|---|---|---|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function code** | 1 | 0x10 | Write multiple configuration register function codes. |
| **Starting Register Address** | 2 | 0x0000-0xFFFF | The first register address of the configuration register to be written. |
| **Number of Registers** | 2 | 0x0000-0xFFFF | The number of configuration registers that need to be written. |
| **Number of Bytes** | 1 | 0x00-0xFF | The total Number of bytes of register data to be written. |
| **Register Data Value** | 2 | 0x0000-0xFFFF | The data value to be written to the configuration register. |

| | | | |
|---|---|---|---|
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Response Frame**

| Slave Address | Function code | Starting Register Address | Number of Registers | CRC-16 |
|---|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Function Code** | 1 | 0x10 | Write multiple configuration register function codes. |
| **Starting Register Address** | 2 | 0x0000-0xFFFF | The first register address of the configuration register to be written. |
| **Number of Registers** | 2 | 0x0000-0xFFFF | The number of configuration registers that need to be written. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

- **Error Response Frame**

| Slave Address | Error Function Code | Error Code | CRC-16 |
|---|---|---|---|

| Data Name | Length (Bytes) | Data Range | Description |
|---|---|---|---|
| **Slave Address** | 1 | 0x01-0xFF | EPC communication address. |
| **Error Function Code** | 1 | 0x90 | Write configuration register error function code. |
| **Error Code** | 1 | 0x00-0xFF | Query the error condition based on the error code. |
| **CRC-16** | 2 | 0x0000-0xFFFF | The checksum is calculated from the value from the slave address to the end of the data. |

## 7.2.3 Register

### 7.2.3.1 Status Register (read only)

**Status Register Table**

| Register Address | Name | Data Types | Description |
|---|---|---|---|
| **0x0001-0x0002** | Real-time pressure | 32 -bit floating point | Real-time pressure data. |
| **0x0007-0x0008** | temperature | 32-bit floating point | Temperature. |
| **0x0009-0x000A** | Ambient air pressure | 32-bit floating point | Ambient air pressure |

### 7.2.3.2 Configuration Register (read/write)

**The data in the configuration register is executed immediately after being sent. If you need to save the sent configuration data, you need to send the configuration register burn command after sending the data. If the configuration register burn command is not sent, the original data will be restored after the device is powered off. (Except for 0x0 002 gas type.)**

**※ If the register address is marked and needs to be saved, the configuration register burning command needs to be executed!**

**Configuration Register Table：**

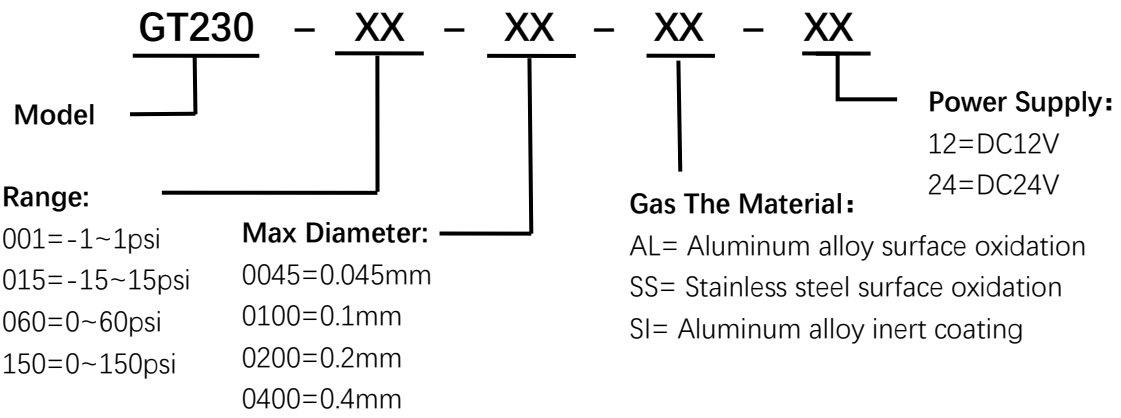
| Register Address | Name | Data Types | Description |
|---|---|---|---|
| **※ 0x0003** | Mailing address | 16-bit integer | Communication address, address range 1 - 255 .<br>After changing the communication address, you need to power off and restart the device to complete the change. The communication address before power off will remain the same as before the change. |
| **※ 0x0004** | Baud rate | 16-bit integer | Communication baud rate, baud rate range is 9600 - 614400 .<br>After changing the baud rate, you need to power off and restart the device to complete the change. The communication baud rate before power off will remain the same as before the change.<br>Send data = actual baud rate / 100.<br>Example: Set the baud rate to 115200 and send data to 1152 . |
| **※ 0x0005** | Unit | 16-bit integer | EPC pressure unit switch.<br>Sending data:<br>0x0000 — PSI<br>0x0001 — kPa |
| **0x0006** | Function Commands | 16-bit integer | EPC needs a certain amount of running time to execute the following functions. During the execution, the device cannot communicate. After the execution is completed, |

| Register Address | Name | Data Types | Description |
|---|---|---|---|
| | | | it can be used normally and the register data is restored to the normal working state (0x 0000 ). You can check whether the function command is executed by reading the register data.<br><br>0x0000 —Normal working status<br>0x0001 —Device Zeroing<br>0x0004 —Configuration register burn<br>0x0005 —Factory Reset |
| **0x000B-0x000C** | Pressure setting | 32-bit floating point | Pressure setting value. After setting, EPC will control the pressure to this setting value.<br>✱ When the memory function is turned on, the configuration register burn command must be executed, otherwise the set value data will not be saved. |
| **0x000D** | Valve controlled switch | 16-bit integer | The valve fully closed, valve fully opened, and automatic control functions can be achieved by configuring this register.<br>Sending data:<br>0x0000 —Valve fully closed<br>0x0001 —Valve fully open<br>0x0002 —Automatic control |
| ✱ **0x000E** | Memory function | 16-bit integer | The memory function can be turned on or off by configuring this register.<br>Sending data:<br>0x0000 —Memory function disabled<br>0x0001 —Memory function enabled |
| ✱ **0x000F** | Pressure Type | 16-bit integer | gauge pressure control mode and absolute pressure control mode can be realized by configuring this register .<br>Sending data:<br>0x0000 —Gauge pressure control mode<br>0x0001 —Absolute pressure control mode |
| **0x0010** | Error Code | 16-bit integer | 0x 0000 indicates no error.<br>For other data, see the error code table. |
| ✱ **0x0016-0x0017** | Proportional coefficient | 32-bit floating point | It is not recommended to adjust this data.<br>If changes are required, it is recommended to use the product supporting software or call for consultation.<br>Software download website www.gastool.cn . |
| ✱ **0x0018-0x0019** | Integration coefficient | 32-bit floating point | It is not recommended to adjust this data.<br>If changes are required, it is recommended to use the product supporting software or call for consultation. |

| Register Address | Name | Data Types | Description |
|---|---|---|---|
| | | | Software download website www.gastool.cn . |
| * 0x001A-0x001B | Integral limit | 32-bit floating point | It is not recommended to adjust this data. If changes are required, it is recommended to use the product supporting software or call for consultation. Software download website www.gastool.cn . |
| * 0x001C-0x001D | Differential coefficient | 32-bit floating point | It is not recommended to adjust this data. If changes are required, it is recommended to use the product supporting software or call for consultation. Software download website www.gastool.cn . |
| * 0x001E-0x001F | Differential Filter | 32-bit floating point | It is not recommended to adjust this data. If changes are required, it is recommended to use the product supporting software or call for consultation. Software download website www.gastool.cn . |
| * 0x0020 | Control cycle | 16-bit integer | It is not recommended to adjust this data. If changes are required, it is recommended to use the product supporting software or call for consultation. Software download website www.gastool.cn . |

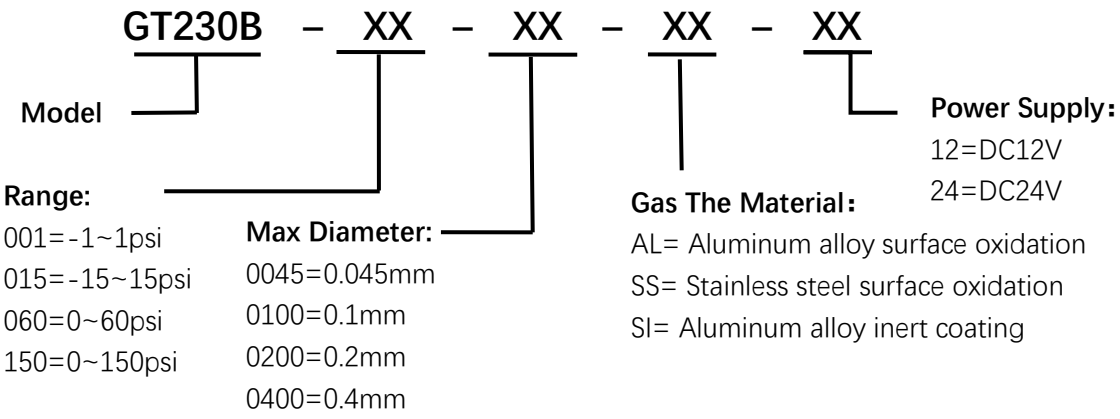
# 8. Nomenclature and Order Guide

## 1. Pressure Relief Type



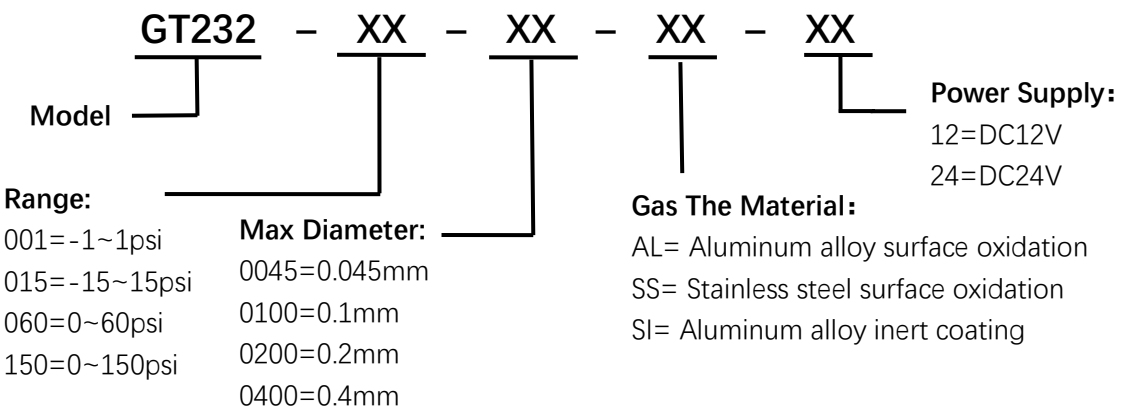

**The above are convertional models,which can be customized according to usage.**

## 2. Back Pressure Type




**The above are convertional models,which can be customized according to usage.**

3. External Pressure Type




**The above are convertional models,which can be customized according to usage.**

# 9. Other

Thank you for purchasing the GST Gas Pressure Controller ( EPC ).

This manual describes the product functions, performance and application methods to achieve the best use of the product.

Please pay attention to the following when using this product:

- This manual should be read and understood thoroughly before using this product to ensure correct use.
- Personnel using this product should have basic knowledge of electrical systems.
- Please keep this manual in a safe place so that you can refer to it whenever you need it.

## Safety Precautions

- Do not touch the terminals while power is on.
- Do not allow metal objects, wires, liquids, etc. to enter the controller, otherwise it may cause dangerous events such as equipment short circuit, electric shock or fire.
- Do not place this product in flammable or explosive places.
- Never disassemble, modify, or repair this product or remove any internal components.
- Please set the product parameters suitable for system control. Improper settings may cause property damage or accidents due to unexpected operations.
- Please calibrate the device within the specified time to ensure the accuracy of the device.
- Before powering on the device, please confirm whether the wiring is correct and whether the power supply voltage meets the requirements of the user manual.
- The gas used must be purified and dust, liquid and oil must be avoided. If necessary, a filter must be installed in the gas line.
- Before use, please confirm whether the gas used is consistent with the calibration gas to avoid incorrect pressure data.
- Do not use corrosive gases to avoid damage to the EPC gas circuit.

## Warranty & Service

- Our EPC products will be repaired free of charge if there are any problems within one year after delivery, provided that the user uses them normally according to the instruction manual and the products have not been physically damaged, contaminated, modified or refurbished.
- The free repair scope does not include gas circuit joints and gas circuit joint sealing rings.
- Please inspect the product promptly after receiving it. If any problem occurs, please promptly report it to the salesperson.
- During the warranty period, the product must be repaired by our company or an authorized service center.

- If the user has used the product with toxic, polluting or corrosive gas, the company will not be responsible for repair or warranty.
- The input gas pressure must meet the pressure resistance standard of the product and cannot exceed the maximum pressure required by the product.
- The gas used in the product must be compatible with the sealing material selected by the user . The user is responsible for using each gas in accordance with the applicable safety regulations. Improper use of the product will invalidate the warranty, and the damage caused by improper use cannot be attributed to the company.
- It is forbidden to disassemble the EPC by yourself . If damage is caused by disassembly by yourself, the warranty promised by our company will be invalid.

GASTOOL

Focus on gas control products Professional docking application service

Tianjin GAST Instrument Co.,Ltd

www.gastool.cn