

Subscribe to DeepL Pro to edit this document.  
Visit [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document) for more information.

**1. Detailed description of the basic instruction set**

Detailed instruction set description is as described in the following sections.Modbus RTU is hexadecimal and must be filled in by converting decimal to hexadecimal when filling in the instructions, including register addresses etc. The instruction address in parentheses in the protocol is in hexadecimal. **The instruction address in parentheses in the protocol is in decimal, and the user must convert it to hexadecimal when writing the instruction, for example, the address in parentheses is 036, which is converted to hexadecimal as 24.**

**1.1 Module address (000)**

When 2 or more transmitters/meter are connected to the upper unit, each transmitter/meter must be set to a different address.

Instruction format: 01 10 00 00 00 01 02 00 02 27 91 (unlocked before use) Code format when address is changed from 01 to 02

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 00 | 00 | 01 | 02 | 00 | 02 | 27 | 91 |

Return format: 01 10 00 00 00 01 01 C9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 00 | 00 | 01 | 01 | C9 |

**1.2 Baud rate setting (001)**

The default baud rate of the transmitter is 0x03:9600 when it is shipped from the factory, and it is changed to 0x07:115200 with the following input format.

Command Format: 01 10 00 01 00 01 02 00 07 E6 43, selects the system baud rate to 115200 after manually sending the command (unlocked before use)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 01 | 00 | 01 | 02 | 00 | 07 | E6 | 43 |

Return format: 01 10 00 01 00 01 50 09 (the answered data is returned after the transmitter/meter is switched to the new baud rate, if the host computer is not switched to the new baud rate in time, the data cannot be received)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 01 | 00 | 01 | 50 | 09 |

**1.3 Data frame format (002)**

The transmitter defaults to the 05 (8 data bits, no parity, 1 stop bit) option format when shipped from the factory, and when modified to the 6 (8 data bits, no parity, 2 stop bits) option

Command format: 01 10 00 02 00 01 02 00 06 27 B0, manually send the command after the parity bit, data bit, stop bit in the upper computer set to the contents of the 4 (need to be unlocked before use)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 02 | 00 | 01 | 02 | 00 | 06 | 27 | B0 |

Return Format: 01 10 00 02 00 01 A0 09 (the answered data is returned after the transmitter is switched to the new data frame format; if the host computer does not switch to the new data frame format in time, the data cannot be received)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 02 | 00 | 01 | A0 | 09 |

**1.4 Protocol type setting (003)**

Transmitter/instrument default protocol is Modbus RTU, if the protocol is changed to free protocol (unlocked before use)

Instruction format: 01 10 00 03 00 01 02 00 00 A6 63

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 03 | 00 | 01 | 02 | 00 | 00 | A6 | 63 |

Return format: 01 10 00 03 00 01 F1 C9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 03 | 00 | 01 | F1 | C9 |

00 (Free Protocol), 01 (Modbus RTU), 02 (ASCII), **after the protocol type is switched, the previously modified calibration parameters and other modified parameters are retained, but the digital frame format will be restored to the default value.**

**1.5 Command response delay setting (004)**

When the delay is 10ms, it is converted to 0A in hexadecimal.

Instruction format: 01 10 00 04 00 01 02 00 0A 27 D3

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 04 | 00 | 01 | 02 | 00 | 0A | 27 | D3 |

Return format: 01 10 00 04 00 01 40 08

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 04 | 00 | 01 | 40 | 08 |

The unit is ms, the answer delay is used for RS485 communication, because RS485 is half-duplex, can only send or receive, can not send and receive at the same time. Some hosts are slow to switch between sending and receiving, resulting in the loss of the answer command, so the answer delay time can be reasonably set to avoid the loss of the command.

**1.6 Locking/Unlocking System Configuration (005)**

Instruction format: 01 10 00 05 00 01 02 5A A5 5C DE

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 05 | 00 | 01 | 02 | 5A | A5 | 5C | DE |

Return format: 01 10 00 05 00 01 11 C8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 05 | 00 | 01 | 11 | C8 |

Prevents accidental modification of the system configuration due to erroneous commands received during module operation. Once the configuration is locked, the module will not be able to receive external serial port commands for modification until the lock is lifted.  
 Including: module address, baud rate, digital frame format, protocol type, restore factory settings and other registers. Write

0x5AA5 Unlocks the system configuration; writing any other value locks the system configuration; reading this register will return 0.

※The transmitter/instrument is locked by default after powering up.

**Firmware version 1.7 (006)**

Returns the module's internal program version number to the host computer, which varies for each transmitter/instrument depending on the model and when it left the factory.

Instruction format: 01 03 00 06 00 01 64 0B

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 06 | 00 | 01 | 64 | 0B |

Return format: 01 03 02 00 FA 38 07

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| module address | function code | byte count | High 16-bit register data | | CRC16 checksum | |
| 01 | 03 | 02 | 00 | FA | 38 | 07 |

**1.8 Restoration of factory settings (007)**

Instruction format: 01 10 00 07 00 01 02 00 55 67 D8

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 07 | 00 | 01 | 02 | 00 | 55 | 67 | D8 |

Return format: 01 10 00 07 00 01 B0 08

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 07 | 00 | 01 | B0 | 08 |

**Note that this operation will delete all user setup parameters and calibration results inside the transmitter and is not recoverable, so please use with caution!**

**1.9 Module status (008)**

Bit15---Bit12:All 0 Bit11:0 Peak Not Detected/1 Detected

Bit10:0 valley not detected/1 detected Bit9:0 normal/overload (V1.3)

Bit8:0 General/1 Smart Sensor Bit7:0 Non-Zero/1 Zero

Bit6:0 normal/1 overflow Bit5:0 stable/1 unstable

Bit4:0 power on not cleared/1 power on cleared

Bit3:0 plus sign/1 minus sign Bit2-0:Decimal point position

(Supported by V1.1 firmware and above)

Instruction format: 01 03 00 08 00 01 05 C8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 08 | 00 | 01 | 05 | C8 |

Return format: 01 03 02 08 02 3E 45

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| module address | function code | bytes | digital | | CRC16 checksum | |
| 01 | 03 | 02 | 08 | 02 | 3E | 45 |

The return data is 0802, 0802 is hexadecimal data, convert 0802 to binary, the data obtained is 0000100000000010.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bit15-Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2-0 |
| binary data | 0000 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 010 |
| decimal system | 0000 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| corresponding state |  | sensing | Trough not detected | normalcy | routine | non-zero | normalcy | stabilise | Power on not cleared | positive value sign + (math.) | 2 decimal places |

**1.10 Reading the measured value (030)**

Command Format: 01 03 00 1E 00 02 A4 0D

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 1E | 00 | 02 | A4 | 0D |

Return format: 01 03 04 00 00 01 62 7A 4A (data varies according to the actual situation)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 03 | 04 | 00 | 00 | 01 | 62 | 7A | 4A |

The measured value is the AD internal code value calibrated and converted by zero and gain.

**1.11 AD conversion speed (032)**

The default AD conversion speeds of different versions of transmitters are different, the high speed version AD conversion speed is 0x07:800, the medium speed version is 0x04:120, and the low speed version is 0x02:640, take the low speed version as an example, when the default speed 0x02:640 is changed to 0x03:1280

Instruction format: 01 10 00 20 00 01 02 00 02 20 F1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 20 | 00 | 01 | 02 | 00 | 02 | 20 | F1 |

Return format: 01 10 00 20 00 01 00 03

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 20 | 00 | 01 | 00 | 03 |

The conversion of analog signals to digital signals, referred to as AD conversion, the faster the AD conversion, the lower the sampling accuracy.

The AD sampling rate is the detection speed of the weighing equipment for the weight of the items on the platform, usually between a few times per second to several hundred times, high-speed weighing applications, up to several thousand times, for an established weighing equipment, the faster the AD rate, the worse the accuracy of the data detected by the AD will be, and the slower the rate of the AD, the higher the accuracy of the AD detection will be relative. Therefore, according to the real weighing needs on the rate, reasonable choice can meet the needs of the lowest gear rate for AD sampling, can maximize the detection accuracy, so as to achieve the best balance on the speed and accuracy.

**1.13 Filter types (034)**

Default is 09: Sliding Average Filtering+ First Order Filtering, change to 08: Median Filtering+ First Order Filtering when command format: 01 10 00 22 00 01 02 00 08 A1 14

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 22 | 00 | 01 | 02 | 00 | 08 | A1 | 14 |

Return format: 01 10 00 22 00 01 A1 C3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 22 | 00 | 01 | A1 | C3 |

Select the appropriate filtering method according to different applications  
 0x00:Not used 0x01:Mean value filtering  
 0x02:Median Filter 0x03:First Order Filter  
 0x04:Sliding average filter 0x05:Median value average filter  
 0x06:Sliding Median Filter  
 0x07:Mean filter + First order filter  
 0x08:Median Filter + First Order Filter  
 0x09:Sliding Average Filter + First Order Filter  
 0x0A:median mean filter + first order filter

**1.14 baud intensity (035)**

When baud strength is changed to 10

Instruction format: 01 10 00 23 00 01 02 00 10 A0 CF

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 23 | 00 | 01 | 02 | 00 | 10 | A0 | CF |

Return format: 01 10 00 23 00 01 F0 03

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 23 | 00 | 01 | F0 | 03 |

Range: 0~50, the larger the number, the stronger the filtering

Filter strength: AD sampling data, due to a variety of reasons, often mixed with a variety of noise from different reasons in which, in order to get a weighing data as close as possible to the real weighing data, weighing equipment will be used in the way of digital filtering data signal processing, and the AD filter strength is an important parameter of the data processing, in general, the smaller the filter strength, the faster the response speed of the data output signal, but the effect of noise filtering is also worse; while the larger the filter strength, the slower the response speed of the output signal, but the effect of noise filtering will be better, between the response speed and filtering effect is reasonable. Generally speaking, the smaller the filter strength, the faster the response of the data output signal, but the worse the effect of noise filtering; and the larger the filter strength, the slower the response of the output signal, but for the effect of noise filtering will be the better, between the response speed and filtering effect, reasonable trade-offs, looking for the optimal balance point, is to use a good weighing equipment is a key step, there is no definite standard, the need for users to do a trade-off based on the site, whether it is the speed priority, or stability priority, according to the customer's actual situation, or the stability priority, according to the customer's actual situation. There is no definite standard, users need to make a trade-off according to the site conditions, whether it is speed priority, or stability priority, according to the actual needs of customers.

**1.15 Code value within zero point (036)**

Instruction format: 01 10 00 24 00 02 04 7F FF FF FF FF 10 D8

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 24 | 00 | 02 | 04 | 7F | FF | FF | FF | 10 | D8 |

Return format: 01 10 00 24 00 02 01 C3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 24 | 00 | 02 | 01 | C3 |

AD internal code value corresponding to zero point; range:-8000000~8000000  
 Write:0x7fffffff to set the current internal code to the zero internal code

The zero point is the reference point for weighing, and the weight added or subtracted from this reference point is the actual weighed weight. Zero point calibration, as the name suggests, is a zero point recorded as a benchmark during calibration, and then the weight calibration done on this basis.

**1.16 Zero point calibration (038)**

Instruction format: 01 10 00 26 00 02 04 00 00 00 00 71 9D

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 26 | 00 | 02 | 04 | 00 | 00 | 00 | 00 | 71 | 9D |

Return format: 01 10 00 26 00 02 A0 03

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 26 | 00 | 02 | A0 | 03 |

**1.18 Gain (weights) calibration (042)**

Instruction format: 01 10 00 2A 00 02 04 4E 20 27 10 16 7D

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 2A | 00 | 02 | 04 | 4E | 20 | 27 | 10 | 16 | 7D |

Return format: 01 10 00 2A 00 02 60 00

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 2A | 00 | 02 | 60 | 00 |

Place a weight and enter the value you want to calibrate, for example, place a 1KG weight on a 10KG pressure sensor and set the weight weight value to 1000. when the calibration is done, place a 2KG weight on the pressure sensor and the measured value read is 2000.

**1.19 Reading AD internal code (044)**

Instruction format: 01 03 00 2C 00 02 05 C2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 2C | 00 | 02 | 05 | C2 |

Return format: 01 03 04 00 19 3B 67 79 2E (data vary according to the actual situation)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 03 | 04 | 00 | 19 | 3B | 67 | 79 | 2E |

The module returns the current AD internal code value to the host.

**1.20 Sensor sensitivity (046)**

If the sensitivity of the sensor is 2.000mv/V, write 20000 (keep 4 digits after the decimal point), 20000 is converted to hexadecimal as 4E20.

Instruction format: 01 10 00 2E 00 02 04 00 00 4E 20 44 43

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 2E | 00 | 02 | 04 | 00 | 00 | 4E | 20 | 44 | 43 |

Return format: 01 10 00 2E 00 02 21 C1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 2E | 00 | 02 | 21 | C1 |

**1.21 Sensor range (048)**

If the sensor range is 100kg, to be accurate to 1g, enter 100000, 100000 into hexadecimal as 186A0

Instruction format: 01 10 00 30 00 02 04 00 01 86 A0 C3 63

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 30 | 00 | 02 | 04 | 00 | 01 | 86 | A0 | C3 | 63 |

Return format: 01 10 00 30 00 02 41 C7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 30 | 00 | 02 | 41 | C7 |

**1.22 Multi-point correction closure (060)**

Instruction format: 01 10 00 3C 00 01 02 00 01 62 AC

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 3C | 00 | 01 | 02 | 00 | 01 | 62 | AC |

Return format: 01 10 00 3C 00 01 C1 C5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| module address | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 00 | 3C | 00 | 01 | C1 | C5 |

This register is write-only, writing any non-zero value turns off multipoint correction, reading this register will return 0

**1.23 Number of multipoint amendments (061)**

Instruction format: 01 03 00 3D 00 01 15 C6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 3D | 00 | 01 | 15 | C6 |

Return format: 01 03 02 00 00 B8 44

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| module address | function code | byte count | register data | | CRC16 checksum | |
| 01 | 03 | 02 | 00 | 00 | B4 | 44 |

This register is read-only. Reading this register returns the number of internal multipoint corrections; writing this register is invalid.

**1.24 Nth point internal code value (062)**

Instruction format: 01 10 00 3E 00 02 04 7F FF FF FF FF 59 63

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 3E | 00 | 02 | 04 | 7F | FF | FF | FF | 59 | 63 |

Return format: 01 10 00 3E 00 02 20 04

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 3E | 00 | 02 | 20 | 04 |

The value of AD internal code corresponding to the Nth point; Range:-8000000~8000000; If 0x7fffffff is written to this register, it will be replaced by the current AD internal code value;

**1.25 Point N weight value (064)**

Instruction format: 01 10 00 40 00 02 04 00 01 00 00 A6 5F

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 40 | 00 | 02 | 04 | 00 | 01 | 00 | 02 | A6 | 5F |

Return format: 01 10 00 40 00 02 40 1C

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 40 | 00 | 02 | 40 | 1C |

Measured value corresponding to the Nth point; Range: -8000000~8000000.

**1.26 Insert correction (066)**

Instruction format: 01 10 00 42 00 01 02 00 01 68 B2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 42 | 00 | 01 | 02 | 00 | 01 | 68 | B2 |

Return format: 01 10 00 42 00 01 A1 DD

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 42 | 00 | 01 | A1 | DD |

First write the AD internal code into the Nth point internal code value register; then write the Nth point weight value to the register; then write 0x01 to this register, the module will insert the data into the internal multi-point correction data table; the data table supports a maximum of 50 points (economy type is 10 points), the register is write-only; read return 0

**1.27 Reading of weights/measurements (080)**

Instruction format: 01 03 00 50 00 02 C4 1A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 50 | 00 | 02 | C4 | 1A |

Return format: 01 03 04 00 00 00 00 84 FA 50 (data varies according to the actual situation)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 03 | 04 | 00 | 00 | 00 | 84 | FA | 50 |

**1.28 Net weight reading (082)**

Instruction format: 01 03 00 52 00 02 65 DA

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 03 | 00 | 52 | 00 | 02 | 65 | DA |

Return format: 01 03 04 FF FF C1 EF EA 0B (data vary according to the actual situation)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 03 | 04 | FF | FF | C1 | EF | EA | OB |

Net Weight= Weight-Tare

**1.29 Tare (084)**

Instruction format: 01 10 00 54 00 02 04 00 00 00 00 64 F6 8B (assuming a tare weight of 100)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 54 | 00 | 02 | 04 | 00 | 00 | 00 | 64 | F6 | 8B |

Return format: 01 10 00 54 00 02 00 18

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 54 | 00 | 02 | 00 | 18 |

Tare Value;Range:-8000000~8000000;Write 0x7fffffff to execute auto tare

When the item weighed by the device has a package, if we only need to weigh the item itself, we have to pre-remove the package as a tare. It is possible to physically place the package directly on the weighing platform and tare it, writing 0x7fffffff to perform the automatic tare removal. If the package is inconveniently separated and the weight of the package is known, the tare weight can be entered into the weighing device by sending a command, this is called digital tare.

**1.30 Setting maximum range (086)**

Instruction format: 01 10 00 56 00 02 04 00 00 27 10 6C 85 (assuming 10000 is entered)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 56 | 00 | 02 | 04 | 00 | 00 | 27 | 10 | 6C | 85 |

Return format: 01 10 00 56 00 02 A1 D8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 56 | 00 | 02 | A1 | D8 |

For example: there is a weighing equipment, it can weigh the maximum weight is 100.00KG, weighing the minimum number of digital jump change is 0.02KG, then the maximum weighing of this weighing is 100.00KG, that is to say, 100.00KG is the maximum weighing of this weighing can weigh, the index value is 0.02KG, the calibration needs to be set up before the maximum weighing and indexing.

**1.31 Setting the index value (088)**

Command format: 01 10 00 58 00 01 02 00 09 6B 4E (set to 0x09:0.1)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 58 | 00 | 01 | 02 | 00 | 09 | 6B | 4E |

Return format: 01 10 00 58 00 01 80 1A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 58 | 00 | 01 | 80 | 1A |

The index value of the weighing platform; you need to set this value before using the weighing platform function.

0x00:0.0001 0x01:0.0002 0x02:0x0005  
 0x03:0.001 0x04:0.002 0x05:0.005  
 0x06:0.01 0x07:0.02 0x08;0.05  
 0x09:0.1 0x0A:0.2 0x0B:0.5  
 0x0C:1 0x0D:2 0x0E:5  
 0x0F:10 0x10:20 0x11:50

**1.32 Manual zeroing range (093)**

Command Format: 01 10 00 5D 00 01 02 00 0A 2B 1A (set 10%)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 5D | 00 | 01 | 02 | 00 | 0A | 2B | 1A |

Return format: 01 10 00 5D 00 01 90 1B

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 5D | 00 | 01 | 90 | 1B |

Manual zero setting is to allow the weighing equipment to take the current weighing data as the current zero point through external keys or commands. As long as the current weighing value does not exceed the range of manual zero setting, the weighing equipment will show the zero reset immediately when the manual zero setting is executed.

**1.33 Perform manual zeroing (094)**

Instruction format: 01 10 00 5E 00 01 02 00 01 6A EE

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 5E | 00 | 01 | 02 | 00 | 01 | 6A | EE |

Return format: 01 10 00 5E 00 01 60 1B

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 5E | 00 | 01 | 60 | 1B |

When multiple channels of transmitters are zeroed simultaneously, the command is 01 10 00 5E 00 01 02 00 FF EB 6E

**1.34 Power-on zero setting range (095)**

Command Format: 01 10 00 5F 00 01 02 00 0A 2A F8 (set 10%)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 5F | 00 | 01 | 02 | 00 | 0A | 2A | F8 |

Return format: 01 10 00 5F 00 01 31 DB

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 5F | 00 | 01 | 31 | DB |

**1.35 Setting the automatic zero tracking range (096)**

Command format: 01 10 00 60 00 01 02 00 64 AE 1B (when 10d is set)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 60 | 00 | 01 | 02 | 00 | 64 | AE | 1B |

Return format: 01 10 00 60 00 01 01 D7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 60 | 00 | 01 | 01 | D7 |

Parameter range: 0~10000; unit: 0.1d; zero tracking function is turned off when setting 0

When the weighing equipment is turned on and in use, the AD signal output will drift because of various reasons such as AD temperature drift, sensor temperature drift creep, etc. The zero tracking calibration program in the equipment will automatically track this very slow drift to offset it, but the zero tracking method has a speed and range.

**1.36 Setting the automatic zero tracking time (097)**

Command format: 01 10 00 61 00 01 02 00 0A 2E 26 (when setting 1s)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 61 | 00 | 01 | 02 | 00 | 0A | 2E | 26 |

Return format: 01 10 00 61 00 01 50 17

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 61 | 00 | 01 | 50 | 17 |

**1.37 Stabilization range (098)**

Command format: 01 10 00 62 00 01 02 00 0A 2E 15 (when 10d is set)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 62 | 00 | 01 | 02 | 00 | 0A | 2E | 15 |

Return format: 01 10 00 62 00 01 A0 17

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 62 | 00 | 01 | A0 | 17 |

**1.38 Stabilization time (099)**

Instruction format: 01 10 00 63 00 01 02 00 0A 2F C4 (when setting 1s)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 63 | 00 | 01 | 02 | 00 | 0A | 2F | C4 |

Return format: 01 10 00 63 00 01 F1 D7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 63 | 00 | 01 | F1 | D7 |

**1.39 Zero Point Range (100)**

Instruction format: 01 10 00 64 00 02 04 00 00 00 0A 74 73 (when 10 is set)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 64 | 00 | 02 | 04 | 00 | 0A | 74 | 73 |

Return format: 01 10 00 64 00 02 00 17

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 64 | 00 | 02 | 00 | 17 |

**1.40 Creep tracking range (102)**

Command format: 01 10 00 66 00 01 02 00 64 AE 7D (when 10d is set)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 66 | 00 | 01 | 02 | 00 | 64 | AE | 7D |

Return format: 01 10 00 66 00 01 E1 D6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 66 | 00 | 01 | E1 | D6 |

**1.41 Creep tracking time (103)**

Command format: 01 10 00 67 00 01 02 00 0A 2E 40 (when setting 1s)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 67 | 00 | 01 | 02 | 00 | 0A | 2E | 40 |

Return format: 01 10 00 67 00 01 B0 16

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 67 | 00 | 01 | B0 | 16 |

**1.42 Weight units (104)**

Command format: 01 10 00 68 00 01 02 00 01 6F 78 (when setting 1-g)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 68 | 00 | 01 | 02 | 00 | 01 | 6F | 78 |

Return format: 01 10 00 68 00 01 80 15

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 68 | 00 | 01 | 80 | 15 |

**1.43 Analog types (130)**

Command Format: 01 10 00 82 00 01 02 00 00 B8 72 (when setting 4~20mA)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 82 | 00 | 01 | 02 | 00 | 00 | B8 | 72 |

Return format: 01 10 00 82 00 01 A1 E1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 82 | 00 | 01 | A1 | E1 |

**1.44 Output data types (131)**

Command format: 01 10 00 83 00 01 02 00 01 78 63 (when setting gross weight value)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 83 | 00 | 01 | 02 | 00 | 01 | 78 | 63 |

Return format: 01 10 00 83 00 01 F0 21

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 83 | 00 | 01 | F0 | 21 |

**1.45 First point analog (132)**

Command Format: 01 10 00 84 00 01 02 0F A0 BD 9C (4000 for 4mA setting)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 84 | 00 | 01 | 02 | 0F | A0 | BD | 9C |

Return format: 01 10 00 84 00 01 41 E0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 84 | 00 | 01 | 41 | E0 |

**1.46 First point analog correction (133)**

Command Format: 01 10 00 85 00 01 02 00 64 B8 2E (fill in 100 when setting 0.1mA)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 85 | 00 | 01 | 02 | 00 | 64 | B8 | 2E |

Return format: 01 10 00 85 00 01 10 20

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 85 | 00 | 01 | 10 | 20 |

**1.47 First point weight value (134)**

Command format: 01 10 00 86 00 02 04 00 00 00 00 7B E5 (when setting full scale 0g)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 86 | 00 | 02 | 04 | 00 | 00 | 00 | 00 | 7B | E5 |

Return format: 01 10 00 86 00 02 A0 21

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 86 | 00 | 02 | A0 | 21 |

**1.48 Second point analog (136)**

Command Format: 01 10 00 88 00 01 02 4E 20 8C A0 (fill in 20000 when setting full scale 20mA)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 88 | 00 | 01 | 02 | 4E | 20 | 8C | A0 |

Return format: 01 10 00 88 00 01 81 E3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 88 | 00 | 01 | 81 | E3 |

**1.49 Second point analog correction (137)**

Command Format: 01 10 00 89 00 01 02 00 64 B8 E2 (fill in 100 when setting full scale 0.1mA)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | register data | | CRC16 checksum | |
| 01 | 10 | 00 | 89 | 00 | 01 | 02 | 00 | 64 | B8 | E2 |

Return format: 01 10 00 89 00 01 D0 23

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | CRC16 checksum | |
| 01 | 10 | 00 | 89 | 00 | 01 | D0 | 23 |

**1.50 Second point weight value (138)**

Command format: 01 10 00 8A 00 02 04 00 00 27 10 61 8C (when setting full scale 10000g)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | | Number of registers | | byte count | High 16-bit register data | | Low 16-bit register data | | CRC16 checksum | |
| 01 | 10 | 00 | 8A | 00 | 02 | 04 | 00 | 00 | 27 | 10 | 61 | 8C |

Return format: 01 10 00 8A 00 02 60 22

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| module address | function code | Register Starting Address | Number of registers | CRC16 checksum |

## Return format: (data varies according to actual situation)