

STEPPERONLINE®

User Manual

Modbus RS485 Stepper Drive



©2022 All Rights Reserved

Address: 15-4, #799 Hushan Road, Jiangning, Nanjing, China

Tel: 0086-2587156578

Web: www.oem-stepperonline.com

Sales: sales@stepperonline.com

Support: technical@stepperonline.com

Notice

Read this manual carefully before any assembling and using. Incorrect handling of products in this manual can result in injury and damage to persons and machinery. Strictly adhere to the technical information regarding installation requirements.

This manual is not for use or disclosure outside of STEPPERONLINE except under permission. All rights are reserved. No part of this manual shall be reproduced, stored in retrieval form, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise without approval from STEPPERONLINE. While every precaution has been taken in the preparation of the book, STEPPERONLINE assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

This document is proprietary information of STEPPERONLINE that is furnished for customer use ONLY. Information in this document is subject to change without notice and does not represent a commitment on the part of STEPPERONLINE. Therefore, information contained in this manual may be updated from time-to-time due to product improvements, etc., and may not conform in every respect to former issues.

- ◆ **Thank you for purchasing STEPPERONLINE DMRS Series Products**
- ◆ **Please read this manual carefully before operating**
- ◆ **Please keep this manual appropriately**

Safety Precautions

Overall Notes



- Do not remove the housing with the drive powered on. Cables. Connectors and optional equipment.
- Please disconnect the power supply for at least 2 minutes and make sure the power indicator is off before wiring and checking. Even if the power is disconnected, voltage may remain inside the drive. Therefore, do not touch the power terminals while the power indicator is on.



- Please use the power supply specifications (number of phases.) that match the product. Voltage. Frequency. AC/DC).
- Be sure to connect the ground terminal of the driver (mounting surface) and motor to the ground pole.
- Do not damage or drag the cable, do not overstress the cable, do not hang heavy objects on the cable, or get caught in the cabinet door.
- Please do not disassemble the product yourself. Repair or modification.
- When the machine is connected to the machine and starts to operate, make sure that the machine is ready for emergency stop.
- Do not touch the inside of the drive.



- The heat sink of the driver may be hot when the power is on or when the power is just cut off. The motor, etc. may be in a high temperature. Take safety measures such as installing a cover to prevent accidental touching by hands and parts (cables, etc.).
- Use double-insulated or reinforced insulation for control power.
- Do not use in places where water can be splashed. Corrosive environments. Do not use the product in the vicinity of flammable gases and combustible materials.
- Do not use damaged. Drivers and motors with missing parts.
- Please set up an emergency stop circuit externally to ensure that the power can be cut off and the operation can be stopped immediately in case of an abnormality.
- If the product is used under poor power conditions, install protection equipment (AC reactor, etc.) to ensure that the input power is supplied within the specified voltage variation range.
- Please use a noise filter to reduce the influence of electromagnetic interference.
- The driver and motor should be used in the specified combination.

Precautions for Storage and Transportation



- Please follow the Commands on the packaging for storage and do not overload the product.
- Please place this product in the following environment:
 - No direct sunlight in the place.
 - Ambient temperature does not exceed the product specification.
 - Humidity does not exceed product specifications. Without condensation.
 - No corrosive gases. Place of flammable gas.
 - Dust. Dust. The place where there is less salt and metal powder.
 - No water. Oil. The place where the splash of medicine, etc. occurs.
 - Vibration or shock does not exceed product specifications.
 - No equipment generating strong magnetic fields in the vicinity.

Precautions for Installation



- Please install the drive in a cabinet that provides fire protection. Electrical protection in the control cabinet.
- Please install the driver and motor in a position with sufficient weight resistance.
- Please install this product in the following environment:
 - No direct sunlight in the place.
 - Ambient temperature does not exceed the product specification.
 - Humidity does not exceed product specifications. Without condensation.
 - No corrosive gases. Place of flammable gas.
 - Dust. Dust. The place where there is less salt and metal powder.
 - No water. Oil. The place where the splash of medicine, etc. occurs.
 - Vibration or shock does not exceed product specifications.
 - No equipment generating strong magnetic fields in the vicinity.
- Do not block the air inlet and exhaust ports, and do not allow foreign objects to enter the drive and motor.
- Do not step on the product or place heavy objects on the drive.
- Please install the driver in the specified direction.
- Make sure to keep the specified intervals between the inner surfaces of the drive control cabinet and other machines.

Precautions for Wiring



- Do not pass the magnetic contactor in the wiring between the drive and the motor.
- Please connect the power terminal and motor terminal firmly.
- Keep a minimum distance of 10mm between the drive and the control cabinet or other equipment.
- Allow at least 30mm of wiring space above and below the driver.
- Signal cable. The encoder cable should be a twisted shielded cable with the shield grounded at both ends.
- The wiring length of the encoder is up to 20m.
- Reduce the frequency of power on/off as much as possible.

Precautions during operation



- To prevent accidents, perform a test run of the servo motor at no load (without the driver connected).
- When you install the machine and start operation, please set the user parameters in advance to match the machine.
- Positive limit (POT) during JOG operation and zero return operation. The signal of negative limit (NOT) is not valid.
- When using the motor on a vertical axis, please provide a safety device to avoid dropping the work-piece in case of alarm or over travel.
- When an alarm occurs, please reset it after investigating the cause and making sure it is safe.
- Do not use the brake of the holding motor for normal braking.

Contents

| | |
|---|----------|
| 1 Introduction | 1 |
| 1.1 Product Introduction | 1 |
| 1.2 Features | 1 |
| 1.3 Application Scenarios | 1 |
| 1.3.1 Hands-on Tuning | 1 |
| 1.3.2 Practical Application Scenarios | 1 |
| 1.4 Check of Product | 3 |
| 1.4.1 Arrival inspection | 3 |
| 1.4.2 Nameplate information | 3 |
| 1.4.3 Part number | 3 |
| 1.4.4 Parts description | 4 |
| 1.4.5 Accessories Information | 4 |
| 2 Installation | 4 |
| 2.1 Storage and Installation Conditions | 4 |
| 2.1.1 Storage condition | 4 |
| 2.1.2 Operating ambience conditions | 4 |
| 2.2 Dimensions | 5 |
| 2.3 Installation Direction and Space | 5 |
| 3 Product Specifications | 6 |
| 3.1 Electrical and Operating Specifications | 6 |
| 3.1.1 Electrical and Operating Specifications | 6 |
| 3.2 Wiring Instructions | 7 |
| 3.2.1 Power Supply Cable & Motor Cable | 8 |
| 3.2.2 I/O Signal Cable | 8 |
| 3.2.3 RS485 Communication Cable | 8 |
| 3.3 Interface Specifications | 8 |
| 3.3.1 Connectors Definition | 8 |
| 3.3.2 CN1 &CN2 Input Power Connector | 9 |
| 3.3.3 CN3-I/O Signals Connector | 9 |
| 3.3.4 CN4-RS485 Communication Connector | 10 |
| 3.3.6 CN5-RS232 Tuning Port | 10 |
| 3.3.8 DIP Switches | 10 |
| 3.4 I/O Connection | 12 |
| 3.4.1 Digital Inputs | 12 |
| 3.4.2 Digital Output | 12 |
| 3.4.3 Brake Output | 12 |

| | |
|---|-----------|
| 4 Modbus RTU | 14 |
| 4.1 Communication Specifications | 14 |
| 4.2 Modbus Function Codes (FC) | 14 |
| 4.2.1 Read Holding Registers FC= 03 | 15 |
| 4.2.2 Preset Single Register FC= 06 | 16 |
| 4.2.3 Preset Multiple Registers FC= 10 | 17 |
| 4.3 Modbus & PR Parameters | 18 |
| 4.3.1 Basic Parameters | 18 |
| 4.3.2 Input and Output Parameters | 20 |
| 4.3.3 Smooth Filter Time Setting for Digital Inputs | 21 |
| 4.3.4 Status Monitoring Parameters | 22 |
| 4.3.5 Control Word and Status Word | 23 |
| 4.4 Error Codes and Troubleshooting | 23 |
| 4.4.1 Communication Error Codes | 23 |
| 4.4.2 Drive Alarm Codes and Troubleshooting | 24 |
| 4.4.3 Error Clear | 25 |
| 4.5 Register Mapping Continuous Read/Write Function | 25 |
| 4.6 S-code Application | 27 |
| 4.7 Enable Drive | 28 |
| 5 PR Mode (Indexer Table) | 29 |
| 5.1 PR Main Features | 29 |
| 5.2 Homing / Return to Zero Position | 29 |
| 5.2.1 Homing Parameters | 30 |
| 5.2.2. Homing by Home Switch | 31 |
| 5.2.3. Homing by Limit Switch | 33 |
| 5.3 Soft Limit & JOG & Quick Stop | 34 |
| 5.3.1 Soft Limit | 34 |
| 5.3.2 JOG | 34 |
| 5.3.3 Quick Stop | 36 |
| 5.4 PR Path | 36 |
| 5.4.1 PR Parameters | 36 |
| 5.4.2 PR Path Configuration | 37 |
| 5.4.3 Other functions of PR | 38 |
| 5.5 Trigger Methods | 40 |
| 5.5.1 CTRG Trigger | 40 |
| 5.5.2 IO Combination Trigger | 41 |
| 5.5.3 Fixed Trigger | 41 |
| 5.5.4 Immediate Trigger | 42 |
| 6 Tuning Operations | 44 |

| | |
|---|-----------|
| 6.1 Basic operation of STEPPERONLINE MotionStudio | 44 |
| 6.1.1 Preparation and Steps | 44 |
| 6.1.2 Operation of Trial Run | 46 |
| 6.1.3 Operation of PR Function | 47 |
| 6.2 Basic Operation of Serial Port Tools Software | 49 |
| 6.2.1 Preparation and Steps | 49 |
| 6.2.2 Operation Instruction Format..... | 50 |
| 6.2.3 Command Cases of Modbus RTU..... | 50 |
| Appendix A Parameters List | 52 |

1 Introduction

1.1 Product Introduction

DMRS Series are stepper drive based on standard Modbus RTU protocol, using RS485 communication can network up to 31 axes. They built-in PR feature with 16-segment position table (PR Mode) can save additional controllers in most of point-to-point applications, to greatly enhance system reliability and reduce the cost. DMRS Series also support the feature of teaching, the operation modes of Profile Position, Profile Velocity and Homing. They can power 2-phase NEMA 8, 11, 14, 17, 23, 24, 34 stepper motors.

The DMRS series is highly reliable and affordable and performs excellently in many industrial applications such as solar equipment, textile, civil, robotics, power generation equipment, 3C, packaging...

1.2 Features

- Low noise and vibration, smooth motion
- Support Modbus RTU protocol, Internal 16-segment position Commands
- Motion can be started by External IO or RS485 or HMI
- Support operation modes: Profile Position, Profile Velocity, Homing
- 7 configurable digital inputs, 3 optically isolated digital outputs
- Limit +, Limit -, Origin, Quick stop, Enable, JOG +, JOG - and Position Table inputs
- Alarm, Brake, Homing complete, In Position complete, Commands complete, Path complete outputs
- 20-50VDC supply voltage for DM556RS, max output current 5.6A
18-80VAC or 30-110VDC supply voltage for DM882RS, max output current 8.2A
- RS232 communication for parameters configuration
- Protections for over voltage, over current, motor cable error, etc.

Compare with Step/Direction

- Built-in single-axis control can save the PLC in most of point-to-point applications to reduce cost;
- Built-in rich diagnostic functions and input and output signals to setup easily;
- Modbus brings more expansion possibility to add value;

1.3 Application Scenarios

Divided into simple hands-on tuning and practical application scenarios

1.3.1 Hands-on Tuning

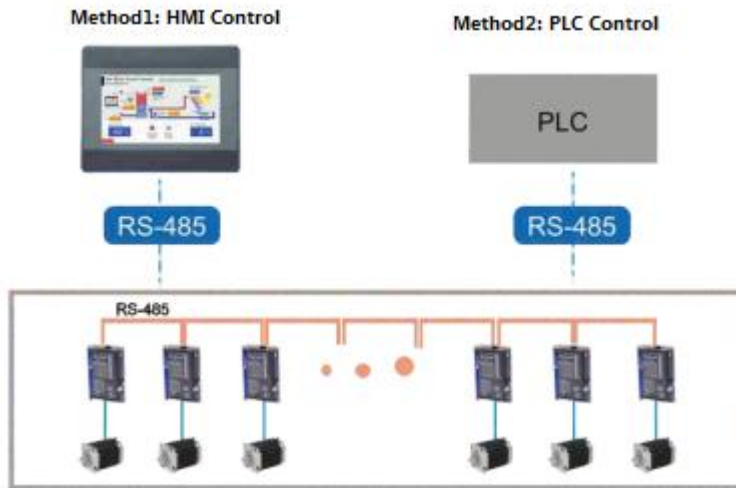
- (1) The trial run can be performed with STEPPERONLINE's MS tuning software, which requires [STEPPERONLINE's tuning software](#), [RS232 tuning cable \(provided by STEPPERONLINE\)](#), [RS232toUSB converter](#), [drive and motor](#), as described in [Section 6.1](#)
- (2) Simple motion can also be performed with serial port tools, which requires [General serial port tools software](#), [RS485 tuning cable \(User-provided\)](#), [RS485toUSB converter](#), [drive and motor](#), as described in [Section 6.2](#).

1.3.2 Practical Application Scenarios

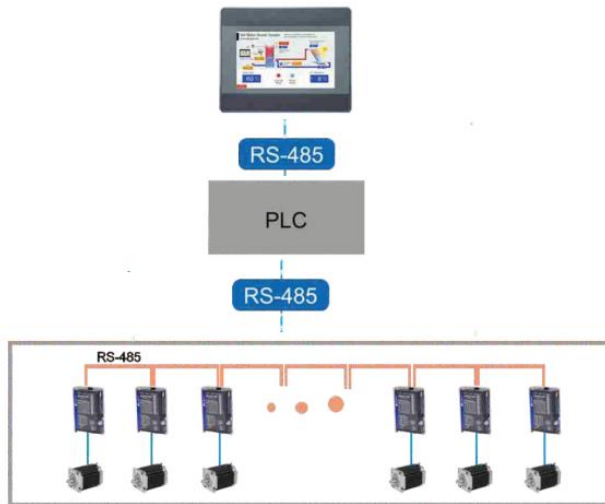
(1) Controlled by Modbus RS485 (PLC or HMI)

The RS485 (Modbus-RTU protocol) communication triggers the drive's PR motion register, which can realize the drive's path motion. [Both PLC and HMI have RS485 communication, so user can choose one or both](#). Using PLC

can write more complex programs to let motion more intelligent, and the HMI can monitor and modify the drive parameters in real time.



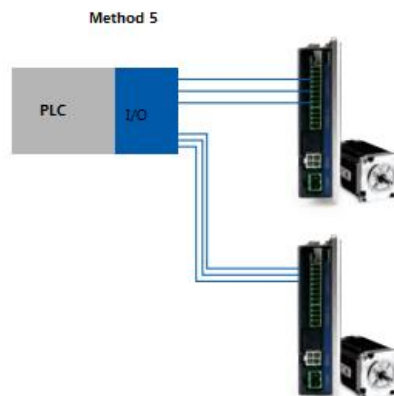
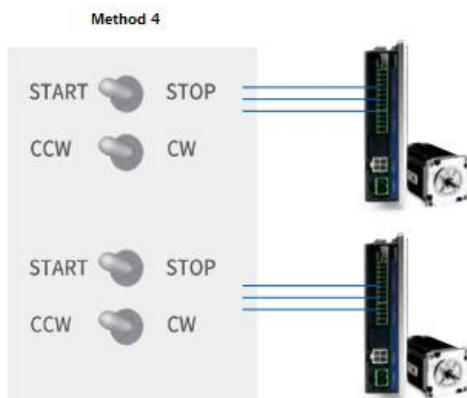
Method3: HMI + PLC Control



(2) Controlled by I/O (switch signal or PLC)

The user only needs to turn off the switch signal to realize the PR motion, which is simple to control and low-cost design.

Users can also use PLC I/O module to realize PR motion, which is more intelligent than switch signal control.



1.4 Check of Product

1.4.1 Arrival inspection

- Check whether the surface of the product is damaged or not during transportation.
- Check the nameplate models of the drive and motor are what you have ordered.
- Check if it is fully equipped with accessories. Accessories include power supply and I/O signals connector.



- Neither the damaged nor missing accessories of stepper system is allowed to install.
- Contact Leadshine or local distributor if any failure was found.

1.4.2 Nameplate information



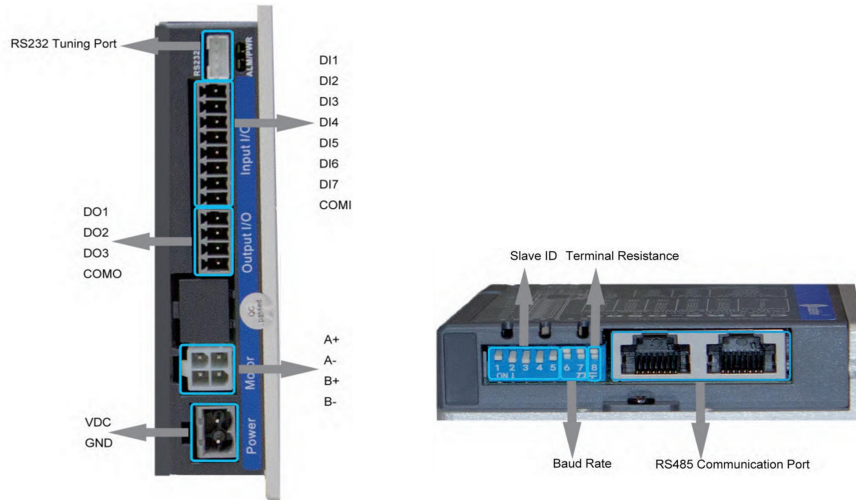
1.4.3 Part number

DM 5 56 RS



- Series Name
- Maximum Operating Voltage
5: 50V
- Maximum Output Current
56: 5.6A
- Communication Mode
RS: RS485

1.4.4 Parts description



1.4.5 Accessories Information

| Name | Necessary | Picture | Description | Need to cost extra |
|------------------------------|-----------|---------|---|--------------------|
| Tuning Cable ⁽¹⁾ | No | | CABLE-PC-1 cable | Yes |
| Network cable ⁽²⁾ | Yes | | (CABLE-TX*M*-BUS) Optional length: 0.1m, 0.2m, 0.3m, 0.4m, 1m, 1.5m, 2m, 3m, 5m, 7m, 10m, 15m, 20m | Yes |

Note:

- (1) Tuning cable is not necessary, but it is recommended to order.
- (2) Network cable is necessary, but user can also buy shielded network cable through 3rd party.
- (3) Motor connector model: [39012040](#), [39000038](#), [Molex](#)

2 Installation

2.1 Storage and Installation Conditions

2.1.1 Storage condition

- Correctly packaged and store in a clean and dry environment where direct sunlight is avoided.
- Store within an ambient temperature ranging from -20°C to +65°C.
- Store within a relative humidity ranging from 40% to 90% and non-condensed.
- Avoid any type of exposure to corrosive gases.

2.1.2 Operating ambience conditions

- Temperature ranging from 0°C to 50°C. The ambient temperature of drive for long-term reliability should be under 40°C. Please install the drive in a well-ventilated area.
- Operation within a relative humidity ranging from 40% to 90% and non-condensed.
- Vibration lower than 0.15mm at a frequency of 10Hz-55Hz.



- DO NOT mount the drive and motor in a location subjected to corrosive or flammable gases, and combustibles.
- Please mount the drive and motor in an indoor electric control cabinet without liquid where direct sunlight is avoided.
- DO NOT mount the drive and motor in a location subjected to airborne dust.
- Please ensure grounding wires are securely connected

2.2 Dimensions

Unit: mm (1inch=25.4mm)

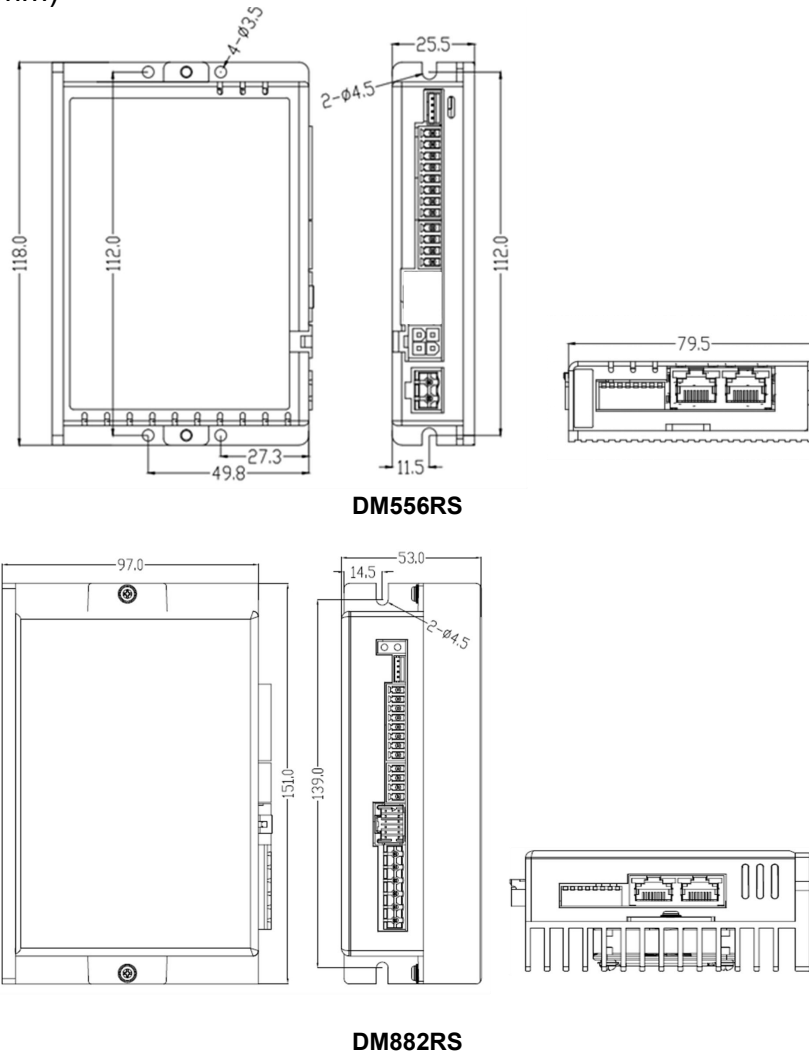


Figure 2.1: DMRS series mechanical drawing

2.3 Installation Direction and Space

- The mounting of drive, wiring and motor should be under the regulations of EN 61800-5-1.
- Incorrect installation may result in a drive malfunction or premature failure of the drive and /or motor. Please follow the guidelines in this manual when installing
- The drive should be mounted perpendicular to the wall or in the control panel.
- In order to ensure the drive is well ventilated, ensure that the all ventilation holes are not obstructed and sufficient free space is given to the drive, and a cooling fan is mounted in the control panel.
- Please ensure grounding wires are securely connected.

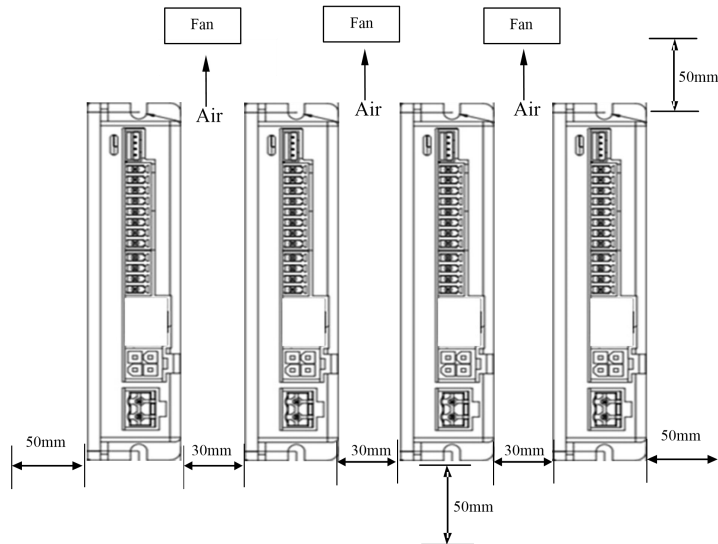



Figure 2.2: DMRS series installation drawing

3 Product Specifications



CAUTION

- Don't hot plug the motor wiring, encoder wiring and RS232 communication wiring during power on.
- Be sure to check the connections and make sure the power lead polarity is correct, or there could result in injury or fire.
- Be sure wait for 5 minutes or longer to touch drives after turning off power
- Be sure to give the power supply voltage dose not exceed the drive's input range.
- If using a motor with small phase current, be sure to modify the driver output

3.1 Electrical and Operating Specifications

3.1.1 Electrical and Operating Specifications

| Name | DM556RS | DM882RS |
|-----------------------|---|-----------------------|
| Supply Voltage | 20-50VDC | 18-80VAC or 30-110VDC |
| Output Current (Peak) | 1.0-5.6A | 0.1-8.2A |
| Size (H*W*L mm) | 118*79.5*25.5 | 151*91*53 |
| Weight (kg) | 0.23 | 0.58 |
| Matched Motor | NEMA 17, 23, 24 | NEMA34 |
| Input Signals | Limit +, Limit -, Origin, Quick stop, Enable, JOG +, JOG - and Position Table | |
| Output Signals | Brake, Alarm, In Position, GPIOs | |
| Protection Functions | Over Current, Over Voltage, Motor Cable Error, etc. | |
| PC Software | STEPPERONLINE MotionStudio | |

| | | |
|-----------------------|-----------------------|--|
| Operating Environment | Environment | Avoid dust, oil, fog and corrosive gases |
| | Operating Temperature | 0-50°C (32 F – 122 F) |
| | Storage Temperature | -20°C-65°C (-4 F – 149 F) |
| | Humidity | 40-90%RH |
| | Vibration | 10-55Hz/0.15mm |
| | Mount | Vertical or horizontal mounting |

3.2 Wiring Instructions

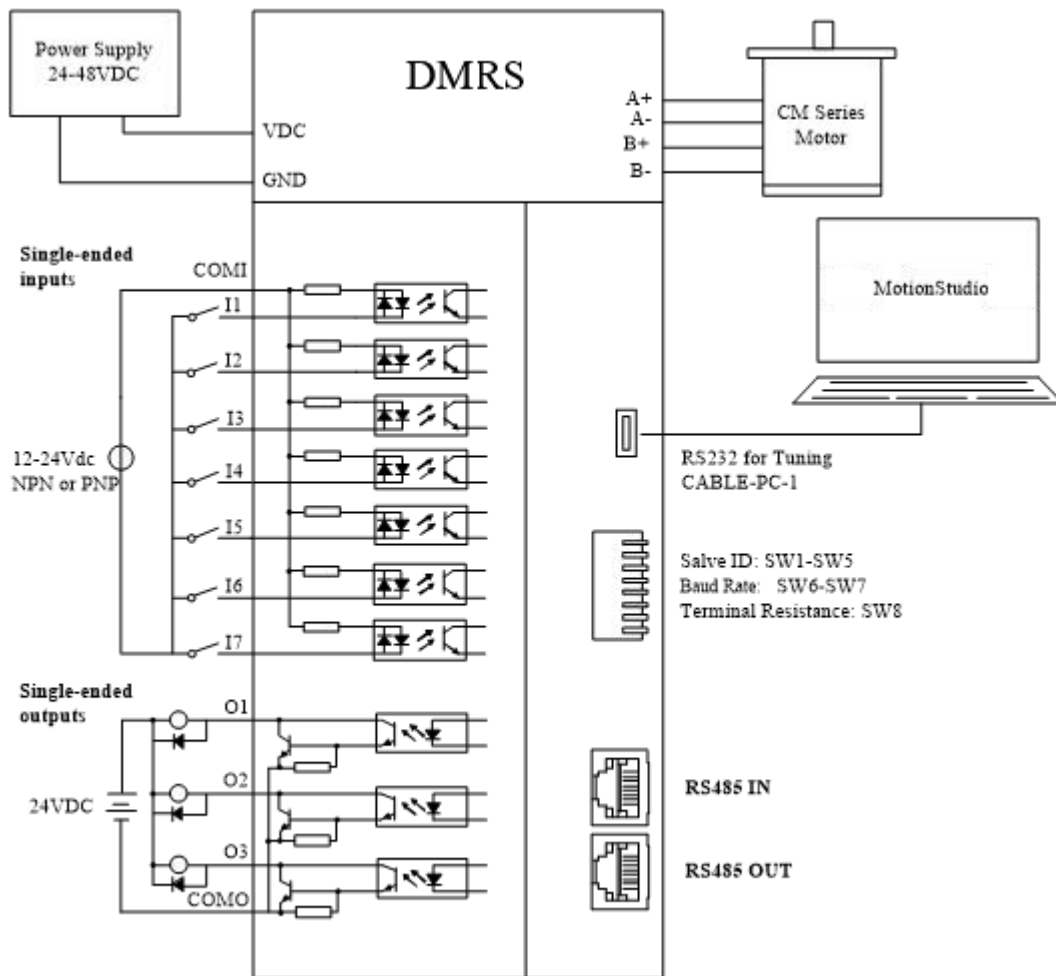


Figure 3.1 Wiring Instructions

- Note:
- (1) There are two RS485 communication ports above, one of them is input port which connects with master station or previous slave, and the other is output port which connects with the following slave.
 - (2) Single-ended inputs I1, I2, I3, I4, I5, I6 and I7 connection types can be common-cathode and common-anode.
 - (3) Single-ended outputs connection type is common-anode.

3.2.1 Power Supply Cable & Motor Cable

- Wire diameter: +VDC, GND, A+, A-, B+, B- terminal wire diameter $\geq 0.3\text{mm}^2$ (AWG15-22)
- A noise filter which can improve anti-interference performance is recommended to be connected between power supply and drive.

3.2.2 I/O Signal Cable

- Wire diameter: I1 - I7, O1 - O3, COM wires diameter $\geq 0.12\text{mm}^2$ (AWG24-26)
- Recommend to adopt shielded twisted pair cable with a length of less than 3 meters (the shorter the better).
- Wiring: As far as possible away from the power line wiring, in order to prevent interference
- Please connect surge absorber to inductive device, such as anti-parallel diode for DC coil, parallel RC-snubbers circuit for AC coil.

3.2.3 RS485 Communication Cable

It is recommended to use shielded Ethernet network cables that do not exceed 100 meters.

3.3 Interface Specifications

3.3.1 Connectors Definition

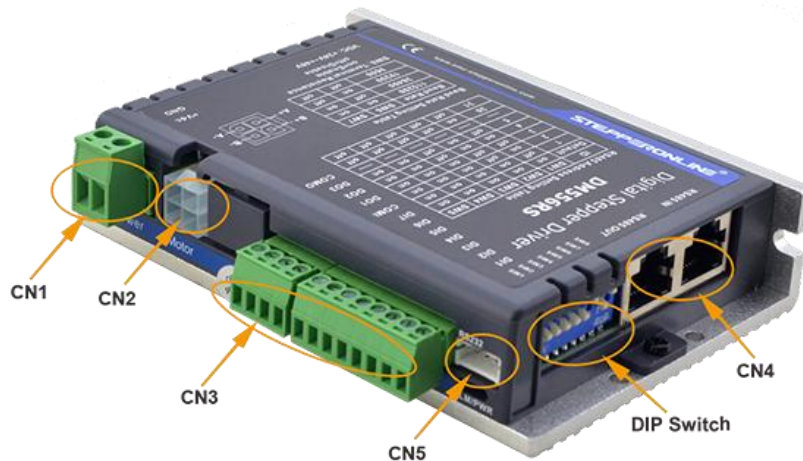

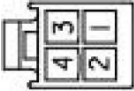


Figure 3.2: DMRS series connectors

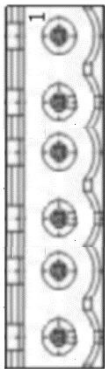
| Name | Description |
|------------|---|
| CN1 | Input power connector |
| CN2 | Motor connector |
| CN3 | Encoder input signals connector |
| CN4 | Digital inputs and outputs connector |
| CN5 | RS485 communication connector |
| CN6 | RS232 tuning connector |
| DIP Switch | Salve ID: SW1-SW5 Baud Rate: SW6-SW7 Terminal Resistance: SW8 |

3.3.2 CN1 & CN2 Input Power Connector

■ **DM556RS**

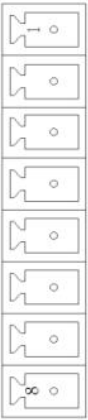
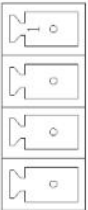
| Name | Pic | PIN | Signal | Description |
|------|---|-----|--------|----------------|
| CN1 |  | 1 | VDC | 24V- 48V |
| | | 2 | GND | GND |
| CN2 |  | 4 | A+ | Motor phase A+ |
| | | 3 | B+ | Motor phase B+ |
| | | 2 | A- | Motor phase A- |
| | | 1 | B- | Motor phase B- |

■ **DM882RS**

| Name | Pic | PIN | Signal | Description |
|---------|--|-----|--------|---------------------------------------|
| CN1&CN2 |  | 1 | A+ | Motor phase A+ |
| | | 2 | B+ | Motor phase B+ |
| | | 3 | A- | Motor phase A- |
| | | 4 | B- | Motor phase B- |
| | | 5 | AC | 18-80VAC or 24-100VDC; No polarity |
| | | 6 | AC | |

Note: When the user uses an AC transformer to supply power, be sure to use an isolation transformer to prevent electric shock or burn out the computer

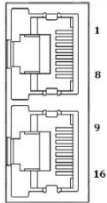
3.3.3 CN3-I/O Signals Connector

| Name | Pic | PIN | Signal | I/O | Description |
|------|---|-----|--------|-----|---|
| CN4 |  | 1 | DI1 | I | Configurable Single-ended Digital Inputs DI1-DI7, 12V - 24V. DI1 is enabling signal default, DI2-DI7 are GPIOs |
| | | 2 | DI2 | I | |
| | | 3 | DI3 | I | |
| | | 4 | DI4 | I | |
| | | 5 | DI5 | I | |
| | | 6 | DI6 | I | |
| | | 7 | DI7 | I | |
| | | 8 | COMI | I | |
| |  | 9 | DO1 | O | Configurable Single-ended Outputs Signals DO1-DO3 (common-cathode or common-anode), Max. 24V/100mA, GPIOs. |
| | | 10 | DO2 | O | |
| | | 11 | DO3 | O | |
| | | 12 | COMO | O | |


Note:(1) DI1 is normally closed, default by Enable signal. It means the motor is locked shaft after the driver powered on.

(2) When using Brake output signals, you need to connect a relay and a diode

3.3.4 CN4-RS485 Communication Connector

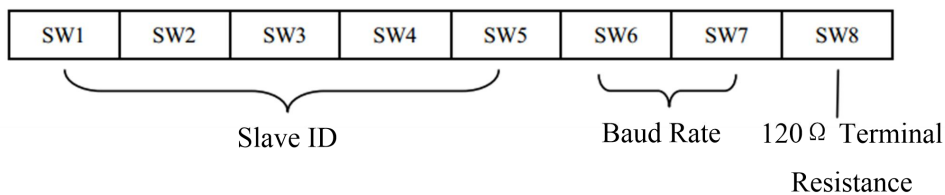
| Name | Pic | PIN | Signal | Description |
|------|---|----------------------------|--------|-------------|
| CN5 |  | 1 | RS485+ | RS485 TxD+ |
| | | 9 | | RS485 RxD+ |
| | | 2 | RS485- | RS485 TxD- |
| | | 10 | | RS485 RxD- |
| | | 5, 6, 13, 14 | GND | GND |
| | | 7, 8, 15, 16, 3, 4, 11, 12 | NC | Reserved |
| | | Connector cover | PE | Shield GND |

3.3.6 CN5-RS232 Tuning Port

| Name | Pic | PIN | Signal |
|------|--|-----|--------|
| CN6 |  | 1 | NC |
| | | 2 | TxD |
| | | 3 | GND |
| | | 4 | RxD |

3.3.8 DIP Switches

The DMRS series drives use an 8-bit DIP switched to set Slave ID (also called Site Alias), Baud Rate and Terminal Resistance, they are shown as below:



(1) Slave ID: SW1-SW5 (off=1, on=0)

| Slave ID | SW1 | SW2 | SW3 | SW4 | SW5 |
|-------------|-----|-----|-----|-----|-----|
| 1 (default) | on | on | on | on | on |
| 1 (factory) | off | on | on | on | on |
| 2 | on | off | on | on | on |
| 3 | off | off | on | on | on |
| 4 | on | on | off | on | on |
| 5 | off | on | off | on | on |
| 6 | on | off | off | on | on |
| 7 | off | off | off | on | on |

| | | | | | |
|----|-----|-----|-----|-----|-----|
| 8 | on | on | on | off | on |
| 9 | off | on | on | off | on |
| 10 | on | off | on | off | on |
| 11 | off | off | on | off | on |
| 12 | on | on | off | off | on |
| 13 | off | on | off | off | on |
| 14 | on | off | off | off | on |
| 15 | off | off | off | off | on |
| 16 | on | on | on | on | off |
| 17 | off | on | on | on | off |
| 18 | on | off | on | on | off |
| 19 | off | off | on | on | off |
| 20 | on | on | off | on | off |
| 21 | off | on | off | on | off |
| 22 | on | off | off | on | off |
| 23 | off | off | off | on | off |
| 24 | on | on | on | off | off |
| 25 | off | on | on | off | off |
| 26 | on | off | on | off | off |
| 27 | off | off | on | off | off |
| 28 | on | on | off | off | off |
| 29 | off | on | off | off | off |
| 30 | on | off | off | off | off |
| 31 | off | off | off | off | off |

Note: (1) When the SW1-SW5 is default (all are on), the Slave ID can be configured by the PC software

(2) Baud Rate: SW6 - SW7

| Baud Rate | SW6 | SW7 |
|------------------|-----|-----|
| 115200 (Default) | on | on |
| 38400 (Factory) | off | on |
| 19200 | on | off |
| 9600 | off | off |

Note: (1) When the SW6-SW7 is default (all are off), the Baud Rate can be configured by the PC software

(3) Terminal Resistance Selection: SW8

SW8=ON: terminal resistance is valid;

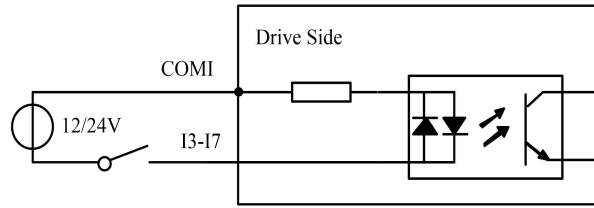
SW8=OFF: terminal resistance is invalid (Factory setting)

Note: (1) The last slave in the network needs to connect a 120Ω terminal resistance, it means set the SW8 to on

3.4 I/O Connection

3.4.1 Digital Inputs

The connections of input signals are as below:



COMI: can be connected to 12 / 24 V (common-anode), or 0 V (common-cath)

Figure 3.3: Input Interface Connection

3.4.2 Digital Output

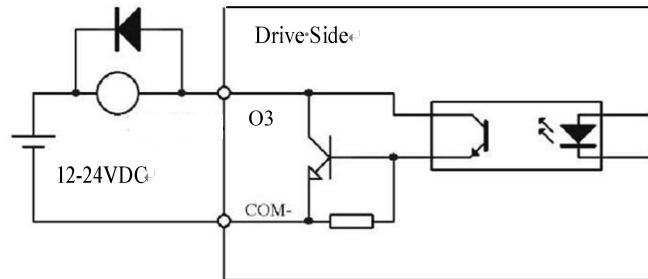


Figure 3.4: Output Interface Wiring

Note: (1) The power supply (12-24VDC) above is provided by user, and if the polarity of power supply is reversed, it will damage the drive.

(2) Digital output is OC output with the maximum capacity of 100mA/24V (recommended 50mA/24V), the provided power supply should be under 30V (recommended 24V), otherwise it will cause damage to the drive.

3.4.3 Brake Output

Use PC software (from STEPPERONLINE or Controller or PLC vendor) to configure this output as a BRAKE CONTROL output. In this case, this signal can be used for automatic brake control while system power failure. It is recommended to connect a fly-wheel diode in parallel to a 24VDC relay and brake coil connection. Refer to the following figure for brake connection.

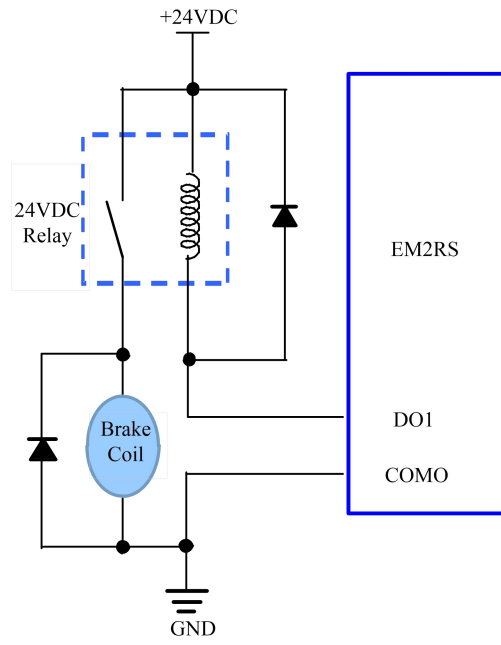


Figure 3.5: Brake output connection

4 Modbus RTU

4.1 Communication Specifications

| Items | | Specifications | | Remarks | |
|----------------|-------------------------------------|--|------------------------------|--|---|
| Communication | Communication Port | RS485 and RS232 | | RS232 only for fine tuning RS485 for motion control | |
| | Baud Rate | 9600/19200/38400/115200[bps] | | Parameter setting | |
| | Synchronous Mode | Start / Stop Synchronization | | - | |
| | Communication Mode | Half-duplex, Master-slave Mode | | Slave/slave communication is prohibited | |
| | Character Composition | Starting bit:1bit Data length:8bit Check bit:even/ odd/ no Stop bit:0/1/2 | | Parameter setting | |
| Protocol | Communicating Protocol | Modbus RTU | | ASCII is not supported | |
| | Slave ID | 0:broadcast 1-31:valid sub-devices Numbers | | Parameter setting | |
| | Function code (FC) | Function code (FC) | Function | | - |
| | | 0x03 | Read single or multiple data | | |
| | | 0x06 | Write value to single data | | |
| | | 0x10 | Write value to multiple data | | |
| | Check Mode | CRC-16 | | Left is Low-bit, Right is high-bit | |
| Message Length | Variable, the max length is 200byte | | - | | |

Single message communication rate of RS485: (Unit: ms)

| Baud rate | Start receiving to send completion time | Receive wait time | Time from transmission completion to recovery of receiving state | Total |
|-----------|---|-------------------|--|-------|
| 115200 | 2.44 | 0.64 | 0.6 | 3.08 |
| 38400 | 5.5 | 1.16 | 0.46 | 7.12 |
| 19200 | 10.76 | 2.2 | 0.38 | 13.34 |
| 9600 | 20.5 | 3.8 | 0.6 | 24.9 |

4.2 Modbus Function Codes (FC)

The current supported function codes as below:

| Function Code (FC) | Type | Description |
|--------------------|---------------------------|---------------------------------------|
| 03 | Read Holding Registers | Requests content of holding registers |
| 06 | Preset Single Register | Writes to single holding register |
| 10 | Preset Multiple Registers | Writes to multiple holding register |

4.2.1 Read Holding Registers FC= 03

| Read Holding Registers Query (Master to Slave) | | Read Holding Registers Response (Slave to Master) | |
|--|---------|---|---------|
| Slave ID | 00 - 1F | Slaver ID | 00 - 1F |
| FC | 03 | FC | 03 |
| Address of beginning register to be read | High | Number of data bytes in message | High |
| | Low | | Low |
| Total number of registers to be read | High | Data from registers (2 bytes per register) | High |
| | Low | | Low |
| CRC | Low | CRC | Low |
| | High | | High |

Example A: Read the value of a single register--peak current

Send message: 01 03 01 91 00 01 D3 1B

Receive message: 01 03 02 00 0A 38 43

Details as following:

Master->slave data:

| | | | | | |
|---------------------|----------|----|------------------|--------------------------|-------|
| Message: | 01 | 03 | 01 91 | 00 01 | D3 1B |
| Description: | Slave ID | FC | Register address | Number of registers read | CRC |

Slave->master data:

| | | | | | |
|---------------------|----------|----|--------------------------|------------------|-------|
| Message: | 01 | 03 | 02 | 00 0A | 38 43 |
| Description: | Slave ID | FC | Number of bytes returned | Value of 0x01 91 | CRC |

Note: 0x0191-- output peak current, 000A(Hexadecimal)=10(decimal, unit: 0.1A), it means the current 1A.

Example B: Read multiple register values

Send message: 01 03 01 BC 00 06 05 D0

Receive message: 01 03 0C 00 00 00 02 00 00 00 01 00 00 00 04 B6 13

Details as following:

Master->slave data:

| | | | | | |
|-------------|----------|----|------------------|--------------------------|-------|
| Message | 01 | 03 | 01 BC | 00 06 | 05 D0 |
| Description | Slave ID | FC | Register address | Number of registers read | CRC |

Slave>master data:

| | | | | | | | | | | |
|-------------|----------|----|--------------------------|-----------------|----------------|-----------------|-----------------|-----------------|-----------------|-------|
| Message | 01 | 03 | 0C | 00 00 | 00 02 | 00 00 | 00 01 | 00 00 | 00 04 | 9D B3 |
| Description | Slave ID | FC | Number of bytes returned | Value of 0x01BC | Address 0x01BD | Value of 0x01BE | Value of 0x01BF | Value of 0x01C0 | Value of 0x01C1 | CRC |

Note:

(1) The above example shows reading the value of Pr5.22, Pr5.23, Pr5.24, their corresponding address are 0x01BD, 0x01BF, 0x01C1.

(2) The data type of parameter is 32bit ,which include high 16bit register and low 16bit register. Usually, we use low 16bits only, but it needs to take the high 16 bits as beginning when we read/write multiple parameters continuously.

4.2.2 Preset Single Register FC= 06

| Preset Single Register Query (Master to Slave) | | Preset Single Register Response (Slave to Master) | |
|--|----------|---|----------|
| Slave ID | 00 -- 1F | Slaver ID | 00 -- 1F |
| FC | 06 | FC | 06 |
| Address of register to write to | High | Address of register written to | High |
| | Low | | Low |
| Value to write | High | Value written to register | High |
| | Low | | Low |
| CRC | Low | CRC | Low |
| | High | | High |

Example C: Write the value of a single register--peak current

Send message: 01 06 01 91 00 20 DD 7B

Receive message: 01 06 01 91 00 20 DD 7B

Details as following:

Master->slave data:

| | | | | | | |
|-------------|----------|----|--|------------------|------------|-------|
| Message | 01 | 06 | | 01 91 | 00 20 | DD 7B |
| Description | Slave ID | FC | | Register address | Write data | CRC |

Slave>master data:

| | | | | | | |
|-------------|----------|----|--|------------------|------------|-------|
| Message | 01 | 06 | | 01 91 | 00 20 | DD 7B |
| Description | Slave ID | FC | | Register address | Write data | CRC |

Note: 0x0191-- output peak current, write data 0x0020=32(decimal, unit: 0.1A), it means the current 3.2A.

Example D: Save the written value to EEPROM

Send message: 01 06 18 01 22 11 06 06

Receive message:01 06 18 01 22 11 06 06

Details as following:

Master->slave data

| | | | | | |
|-------------|---------|---------------|------------------|------------|----------------|
| Message: | 01 | 06 | 18 01 | 22 11 | 06 06 |
| Description | Address | Function code | Register address | Write data | CRC check code |

Slave>master data:

| | | | | | |
|-------------|---------|---------------|------------------|------------|----------------|
| Message: | 01 | 06 | 18 01 | 22 11 | 06 06 |
| Description | Address | Function code | Register address | Write data | CRC check code |

Note: 0x1801-- Auxiliary control word, and 0x2211 is to save the value to EEPROM. This step is required after the parameter has been modified, to prevent losing the written value after power-off.

4.2.3 Preset Multiple Registers FC= 10

| Preset Multiple Registers Query (Master to Slave) | | Preset Multiple Registers Response (Slave to Master) | |
|---|----------------------|--|---------|
| Slave ID | 00 - 1F | Slaver ID | 00 - 1F |
| FC | 10 | FC | 10 |
| Address of first register to write to | High | Address of first register written to | High |
| | Low | | Low |
| Total number of registers to write to | High | Total number of registers written to | High |
| | Low | | Low |
| Number of data bytes in message | 2 bytes per register | CRC | Low |
| Value1 to write | High | | High |
| | Low | | |
| Value2 to write | High | | |
| | Low | | |
| CRC | Low | | |
| | High | | |

Example E: Write multiple registers-- configure input port functions

Send message:01 10 01 46 00 04 08 00 00 00 28 00 00 00 29 1C 14

Receive message:01 10 01 46 00 04 21 E3

Details as following:

Master->slave data:

| | | | | | | | |
|-------------|----------|----|---------------------|-------------------------|-----------------|----------------------------|-------|
| Message | 01 | 10 | 01 46 | 00 04 | 08 | 00 00, 00 28, 00 00, 00 29 | 1C 14 |
| Description | Slave ID | FC | First address write | Number of address write | Number of bytes | Value to write | CRC |

Slave>master data:

| | | | | | |
|-------------|----------|----|---------------------|-------------------------|-------|
| Message | 01 | 10 | 01 46 | 00 04 | 21 E3 |
| Description | Slave ID | FC | First address write | Number of address write | CRC |

Note: (1) In above example, modify the function of DI2 / DI3, and write the value as DI2=0x28 (means path 0), DI2=0x29 (means path 1);

(2) The data type of parameter is 32bit, which include high 16bit register and low 16bit register.usually , we use low

16bits only, but it needs to take the high 16 bits as beginning when we read/write multiple parameters continuously.

4.3 Modbus & PR Parameters

4.3.1 Basic Parameters

STEPPERONLINE RS485 parameter data type is 32 bits, a parameter contains two registers of high 16 bits and low 16 bits, only the lower 16 bits are used in practice. However, when reading or writing multiple parameters in succession, the high 16 bits of the parameter need to be used as the start, usually 00.

| Register Address | Software Par. # | Name | Description | Range | Default Value | Unit |
|------------------|-----------------|---|---|-----------|---------------|---------|
| 0x0001 | Pr0.00 | Pulse/revolution | 10000 | 200-51200 | 10000 | P/R |
| 0x0005 | Pr0.02 | Control mode source | Invalid | 0-10 | 1 | -- |
| 0x0007 | Pr0.03 | Motor direction | 0: CW 1: CCW | 0-1 | 0 | -- |
| 0x0009 | Pr0.04 | Motor inductance | Invalid | 0-10000 | 1499 | 0.001mH |
| 0x00F | Pr0.07 | Forced enable by software | Software forced enable has a higher priority than IO enable, and when this value is 0, the enable status of the drive only depends on the IO signal. When this value is 1, the motor is enabled regardless of the IO signal status. | 0-1 | 0 | -- |
| 0x00A1 | Pr2.00 | Command filter time | To configure the time for internal command filtering | 0-512 | 15 | 0.1ms |
| 0x0145 | Pr4.02 | DI1(input 1) | Default is normal-open (N.O) type, it can be set to normal-closed (N.C) type by setting the corresponding port +0x80 DI1 is enable(default), N.C type input. 0: invalid; 7: alarm clearing; 8: enable (also can be set by 0x00F); 0x20: Trigger command (CTRG); 0x21: Trigger homing; 0x22: EMG (quick stop); 0x23: JOG+; 0x24: JOG-; 0x25: POT (positive limit); 0x26: NOT (negative limit); 0x27: ORG (home switch); 0x28: ADD0 (path address 0); 0x29: ADD1 (path address 1); 0x2A: ADD2 (path address 2); 0x2B: ADD3 (path address 3); 0x2C: JOG velocity 2 | 0-65535 | 136 (0x88) | -- |
| 0x0147 | Pr4.03 | DI2(input 2) | | 0-65535 | 0 | -- |
| 0x0149 | Pr4.04 | DI3(input 3) | | 0-65535 | 0 | -- |
| 0x014B | Pr4.05 | DI4(input 4) | | 0-65535 | 0 | -- |
| 0x014D | Pr4.06 | DI5(input 5) | | 0-65535 | 0 | -- |
| 0x014F | Pr4.07 | DI6(input 6) | | 0-65535 | 0 | -- |
| 0x0151 | Pr4.08 | DI7(input 7) | | 0-65535 | 0 | -- |
| 0x0157 | Pr4.11 | DO1(output 1) | Default is normal-open(N.O) type, it can be set to normal-closed(N.C) type by setting the corresponding port +0x80 0: invalid; 0x20: command completed; 0x21: path completed; 0x22: homing completed; 0x23: in-position completed; 0x24: brake output; 0x25: alarm output; | 0-65535 | 0 | -- |
| 0x0159 | Pr4.12 | DO2(output 2) | | 0-65535 | 0 | -- |
| 0x015B | Pr4.13 | DO3(output 3) | | 0-65535 | 0 | -- |
| 0x0167 | Pr4.19 | Delay of brake released | Keep default normally | 0-1500 | 250 | ms |
| 0x0169 | Pr4.20 | Delay of brake locked | Keep default normally | 0-1500 | 250 | ms |
| 0x016B | Pr4.21 | Threshold value of brake locking velocity | Keep default normally | 0-500 | 10 | -- |
| 0x016D | Pr4.22 | Alarm selection detection | Bit setting: =1: Yes; =0: No bit0: over-current (Cannot be changed) bit1: over-voltage bit3: ADC sampling failure | 0-65535 | 0 | -- |

| | | | bit4: Locked shaft alarm bit5: EEPROM alarm bit6: Auto-tuning alarm | | | | | | | | | | | | | | | | | |
|---------|---------------------|--|--|--|---------------------|-------|-------|------|--------|------|---------|------|-------------------|------|----------------|------|--------|--|---|----|
| 0x0177 | Pr4.27 | Bus voltage | | 0-65535 | 0 | 0.1V | | | | | | | | | | | | | | |
| 0x0179 | Pr4.28 | Digital input statue | Bit0-Bit6: DI1-DI7 | 0-65535 | 0 | -- | | | | | | | | | | | | | | |
| 0x017B | Pr4.29 | Digital output statue | Bit0-Bit2: DO1-DO3 | 0-65535 | 0 | -- | | | | | | | | | | | | | | |
| 0x0187 | Pr4.35 | DIP switches statue | | 0-65535 | 0 | -- | | | | | | | | | | | | | | |
| 0x0191 | Pr5.00 | Peak current | Approximately equal to motor phase current value multiplied by 1.4 | 522: 0-22 556: 0-56 870: 0-70 882: 0-82 | 10 | 0.1A | | | | | | | | | | | | | | |
| 0x0197 | Pr5.03 | Percentage of shaft locked current(power on) | Keep default normally | 0-100 | 100 | -- | | | | | | | | | | | | | | |
| 0x0199 | Pr5.04 | Shaft locked duration | Keep default normally | 0-1500 | 200 | 1ms | | | | | | | | | | | | | | |
| 0x019F | Pr5.07 | Rising time of shaft locked current (power on) | Keep default normally | 1-60 | 1 | 100ms | | | | | | | | | | | | | | |
| 0x01A5 | Pr5.10 | The max stop time | Keep default normally | 100-1000 | 1000 | ms | | | | | | | | | | | | | | |
| 0x01AB | Pr5.13 | Auto-tuning at power on | 0: disable, 1: enable | 0-1 | 1 | -- | | | | | | | | | | | | | | |
| 0x01BD | Pr5.22 | RS485 baud rate | 0: 2400 1: 4800 2: 9600 (default) 3: 19200 4: 38400(factory) 5: 57600 6: 115200 When SW6 and SW7 are all OFF, it can be set by PC software. | 0-6 | 4 | -- | | | | | | | | | | | | | | |
| 0x01BF | Pr5.23 | RS485 ID | When SW1 - SW5 are all ON, it can be set by PC software | 0-127 | 1 | -- | | | | | | | | | | | | | | |
| 0x01C1 | Pr5.24 | RS485 data type selection | 0: 8-bit data, even check, 2 stop bits; 1: 8-bit data, odd check, 2 stop bits 2: 8-bit data, even check, 1 stop bit; 3: 8-bit data, odd check, 1 stop bit: 4: 8-bit data, no check,1 stop bit: 5: 8-bit data, no check,2 stop bits; | 0-11 | 4 | -- | | | | | | | | | | | | | | |
| 0x01C3 | Pr5.25 | RS485 control word | | 0-32767 | 0 | -- | | | | | | | | | | | | | | |
| 0x01C4 | Pr5.26 | Communication bit delay | | 0-100 | 35 | bit | | | | | | | | | | | | | | |
| 0x01D1 | Pr5.32 | Switching time to standby | | 10-65535 | 200 | ms | | | | | | | | | | | | | | |
| 0x01D3 | Pr5.33 | Standby current percentage | | 0-100 | 50 | | | | | | | | | | | | | | | |
| 0x01E1 | Pr6.00 | JOG velocity | This JOG is triggered by RS485. For JOG triggered by IO, please use Pr8.40/8.41 | 0-5000 | 60 | r/min | | | | | | | | | | | | | | |
| 0x01E3 | Pr6.01 | Interval | | 0-10000 | 100 | ms | | | | | | | | | | | | | | |
| 0x01E5 | Pr6.02 | Running times | | 0-30000 | 1 | -- | | | | | | | | | | | | | | |
| 0x01E7 | Pr6.03 | Acc. / Dec. time | | 0-10000 | 200 | -- | | | | | | | | | | | | | | |
| 0x01FF | Pr6.15 | Version information | Read only | 0-65535 | 0 | -- | | | | | | | | | | | | | | |
| 0x0201 | Pr6.16 | Firmware information | Read only | 0-65535 | 0 | -- | | | | | | | | | | | | | | |
| 0x0231 | Pr7.00 | Motor model | Invalid | 0-100 | 0 | -- | | | | | | | | | | | | | | |
| 0x0235 | Pr7.02 | Back EMF coefficient | Invalid | 0-32767 | 100 | 1ms | | | | | | | | | | | | | | |
| 0x0237 | Pr7.03 | Current loop proportional KP | Invalid | 0-3000 | 1500 | -- | | | | | | | | | | | | | | |
| 0x0239 | Pr7.04 | Current loop KI | It is recommended to keep the factory value, when disable auto-tuning, can be modified | 0-1500 | 300 | -- | | | | | | | | | | | | | | |
| 0x023B | Pr7.05 | Current loop KP | | 0-1024 | 100 | -- | | | | | | | | | | | | | | |
| 0x023D | Pr7.06 | Current loop Kc | Invalid | 0-32767 | 300 | -- | | | | | | | | | | | | | | |
| 0x0243 | Pr7.09 | Over-voltage threshold | Invalid | 0-1000 | 90 | V | | | | | | | | | | | | | | |
| 0x1003 | - | Motion status | Read only: <table border="1" style="margin-left: 20px;"> <tr> <th>Bit NO.</th> <th>Read value =1 means</th> </tr> <tr> <td>Bit0</td> <td>Fault</td> </tr> <tr> <td>Bit1</td> <td>Enable</td> </tr> <tr> <td>Bit2</td> <td>Running</td> </tr> <tr> <td>Bit4</td> <td>Command completed</td> </tr> <tr> <td>Bit5</td> <td>Path completed</td> </tr> <tr> <td>Bit6</td> <td>Homing</td> </tr> </table> | Bit NO. | Read value =1 means | Bit0 | Fault | Bit1 | Enable | Bit2 | Running | Bit4 | Command completed | Bit5 | Path completed | Bit6 | Homing | | - | -- |
| Bit NO. | Read value =1 means | | | | | | | | | | | | | | | | | | | |
| Bit0 | Fault | | | | | | | | | | | | | | | | | | | |
| Bit1 | Enable | | | | | | | | | | | | | | | | | | | |
| Bit2 | Running | | | | | | | | | | | | | | | | | | | |
| Bit4 | Command completed | | | | | | | | | | | | | | | | | | | |
| Bit5 | Path completed | | | | | | | | | | | | | | | | | | | |
| Bit6 | Homing | | | | | | | | | | | | | | | | | | | |

| | | | completed | | | | | | | | | | | | | | | | | | | | | |
|-------------|--|----------------------------|--|-------------|----------|------------|---------------------|--------|---------------------|--------|-------------------------------|--------|---|--------|-------------------------------------|--------|-------------------------------|--------|---|--------|--|----|---|----|
| 0x1801 | - | Control word | <table border="1"> <thead> <tr> <th>Write value</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0x1111</td> <td>Reset current alarm</td> </tr> <tr> <td>0x1122</td> <td>Reset history alarm</td> </tr> <tr> <td>0x2211</td> <td>Save all parameters to EEPROM</td> </tr> <tr> <td>0x2222</td> <td>Parameter reset(exclude motor parameters)</td> </tr> <tr> <td>0x2233</td> <td>All parameters are reset to factory</td> </tr> <tr> <td>0x2244</td> <td>Save all mappings into EEPROM</td> </tr> <tr> <td>0x4001</td> <td>JOG CW (Need to write once at least 50ms)</td> </tr> <tr> <td>0x4002</td> <td>JOG CCW (Need to write once at least 50ms)</td> </tr> </tbody> </table> | Write value | Function | 0x1111 | Reset current alarm | 0x1122 | Reset history alarm | 0x2211 | Save all parameters to EEPROM | 0x2222 | Parameter reset(exclude motor parameters) | 0x2233 | All parameters are reset to factory | 0x2244 | Save all mappings into EEPROM | 0x4001 | JOG CW (Need to write once at least 50ms) | 0x4002 | JOG CCW (Need to write once at least 50ms) | - | - | -- |
| Write value | Function | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1111 | Reset current alarm | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1122 | Reset history alarm | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2211 | Save all parameters to EEPROM | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2222 | Parameter reset(exclude motor parameters) | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2233 | All parameters are reset to factory | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2244 | Save all mappings into EEPROM | | | | | | | | | | | | | | | | | | | | | | | |
| 0x4001 | JOG CW (Need to write once at least 50ms) | | | | | | | | | | | | | | | | | | | | | | | |
| 0x4002 | JOG CCW (Need to write once at least 50ms) | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1901 | - | Save parameter status word | <table border="1"> <thead> <tr> <th colspan="2">Read only</th> </tr> <tr> <th>Read Value</th> <th>Means</th> </tr> </thead> <tbody> <tr> <td>0x5555</td> <td>Save successfully</td> </tr> <tr> <td>0xAAAA</td> <td>Failed to save</td> </tr> </tbody> </table> | Read only | | Read Value | Means | 0x5555 | Save successfully | 0xAAAA | Failed to save | - | - | -- | | | | | | | | | | |
| Read only | | | | | | | | | | | | | | | | | | | | | | | | |
| Read Value | Means | | | | | | | | | | | | | | | | | | | | | | | |
| 0x5555 | Save successfully | | | | | | | | | | | | | | | | | | | | | | | |
| 0xAAAA | Failed to save | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2203 | - | Current alarm | <table border="1"> <thead> <tr> <th colspan="2">Read only</th> </tr> <tr> <th>Error code</th> <th>Means</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>Over- current</td> </tr> <tr> <td>0x02</td> <td>Over- voltage</td> </tr> <tr> <td>0x40</td> <td>Current sampling fault</td> </tr> <tr> <td>0x80</td> <td>Failed to lock shaft</td> </tr> <tr> <td>0x200</td> <td>EEPROM fault</td> </tr> <tr> <td>0x100</td> <td>Auto-tuning fault</td> </tr> </tbody> </table> | Read only | | Error code | Means | 0x01 | Over- current | 0x02 | Over- voltage | 0x40 | Current sampling fault | 0x80 | Failed to lock shaft | 0x200 | EEPROM fault | 0x100 | Auto-tuning fault | - | - | -- | | |
| Read only | | | | | | | | | | | | | | | | | | | | | | | | |
| Error code | Means | | | | | | | | | | | | | | | | | | | | | | | |
| 0x01 | Over- current | | | | | | | | | | | | | | | | | | | | | | | |
| 0x02 | Over- voltage | | | | | | | | | | | | | | | | | | | | | | | |
| 0x40 | Current sampling fault | | | | | | | | | | | | | | | | | | | | | | | |
| 0x80 | Failed to lock shaft | | | | | | | | | | | | | | | | | | | | | | | |
| 0x200 | EEPROM fault | | | | | | | | | | | | | | | | | | | | | | | |
| 0x100 | Auto-tuning fault | | | | | | | | | | | | | | | | | | | | | | | |

4.3.2 Input and Output Parameters

| Register Address | Par. # in software | Definition | Description | Range | Default | Unit |
|------------------|--------------------|------------|---|---------|------------|------|
| 0x0145 | Pr4.02 | SI1 (DI1) | (1) SI1 is set to enable by default, N.C (normally closed). (2) Other inputs are N.O (normally open) by default. (3) The value of bit7 of each input register is set normally closed or normally open, bit7=1 is normally closed, bit7=0 is normally closed | 0-65535 | 136 (0x88) | -- |
| 0x0147 | Pr4.03 | SI2 (DI2) | | 0-65535 | 0 | -- |
| 0x0149 | Pr4.04 | SI3 (DI3) | | 0-65535 | 0 | -- |
| 0x014B | Pr4.05 | SI4 (DI4) | | 0-65535 | 0 | -- |
| 0x014D | Pr4.06 | SI5 (DI5) | | 0-65535 | 0 | -- |
| 0x014F | Pr4.07 | SI6 (DI6) | | 0-65535 | 0 | -- |
| 0x0151 | Pr4.08 | SI7 (DI7) | | 0-65535 | 0 | -- |
| 0x0157 | Pr4.11 | SO1 (DO1) | (1) All outputs are N.O (normally open) by default, (2) The value of bit7 of each input register is set normally closed or normally open, bit7=1 is normally closed, bit7=0 is normally closed. | 0-65535 | 0 | -- |
| 0x0159 | Pr4.12 | SO2 (DO2) | | 0-65535 | 0 | -- |
| 0x015B | Pr4.13 | SO3 (DO3) | | 0-65535 | 0 | -- |

Note: (1) If the input or output function is set repeatedly, only after the restart drive can detect it.
(2)When input functions are configured, it is effective after clicking save and restart drive.

Digital Input port function assignment:

Normal-open (N.O); Normal-closed (N.C).

| Digital Inputs | | | | Digital Outputs | | | |
|-----------------|---------|------|------|------------------|---------|------|------|
| Functions | Symbols | Type | | Functions | Symbols | Type | |
| | | N.O. | N.C. | | | N.O. | N.C. |
| Trigger Command | CTRG | 0x20 | 0xA0 | Command Complete | CMD_OK | 0x20 | 0xA0 |
| Homing Trigger | HOME | 0x21 | 0xA1 | Path Complete | MC_OK | 0x21 | 0xA1 |
| Quick Stop | STP | 0x22 | 0xA2 | Homing Complete | HOME_OK | 0x22 | 0xA2 |
| JOG+ | JOG+ | 0x23 | 0xA3 | Alarm | ALM | 0x25 | 0xA5 |
| JOG- | JOG- | 0x24 | 0xA4 | Brake | BRK | 0x24 | 0xA4 |
| Positive Limit | PL | 0x25 | 0xA5 | In-position | PEND | 0x23 | 0xA3 |
| Negative Limit | NL | 0x26 | 0xA6 | | | | |
| Home switch | ORG | 0x27 | 0xA7 | | | | |
| Path Address 0 | ADD0 | 0x28 | 0xA8 | | | | |
| Path Address 1 | ADD1 | 0x29 | 0xA9 | | | | |
| Path Address 2 | ADD2 | 0x2A | 0xAA | | | | |
| Path Address 3 | ADD3 | 0x2B | 0xAB | | | | |
| Clear Error | CLR | 0x07 | 0x87 | | | | |
| Enable | SRV-ON | 0x08 | 0x88 | | | | |

4.3.3 Smooth Filter Time Setting for Digital Inputs

This section describes the setting of the value of each bit of the registers of the 7 inputs, where the filtering time is set by the high 8 bits.

Set value of low 8 bits:

| Normally open. Normally closed setting | Digital input function setting | | | | | | |
|--|--|------|------|------|------|------|------|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Normally open 1: Normally closed | (000 0000) 0x00: Invalid input. (000 1000) 0x08: Enable. (010 0000) 0x20: CTRG. (010 0001) 0x21: Homing trigger. (010 0010) 0x22: Quick stop. (010 0011) 0x23: JOG+. (010 0100) 0x24: JOG-. (010 0101) 0x25: Positive limit. (010 0110) 0x26: Negative limit. (010 0111) 0x27: Home switch. (010 1000) 0x28: Path address 0. (010 1001) 0x29: Path address 1. (010 1010) 0x2A: Path address 2. (010 1011) 0x2B: Path address 3. (010 1100) 0x2C: JOG2 velocity | | | | | | |

Set value of high 8 bits:

| Reserved | | | | Filter time setting | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------|-------|-------|---|---------------------------|----------------|---------------------------|------|----|------|---|------|---|------|---|------|---|------|---|------|---|------|---|------|----|------|----|------|----|------|----|------|----|------|-----|------|-----|------|-----|--|--|
| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved bit, write 0 by default | | | | <table border="1"> <thead> <tr> <th>Register Value</th> <th>Filtering time (unit: ms)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>10</td></tr> <tr><td>0001</td><td>1</td></tr> <tr><td>0010</td><td>2</td></tr> <tr><td>0011</td><td>3</td></tr> <tr><td>0100</td><td>4</td></tr> <tr><td>0101</td><td>5</td></tr> <tr><td>0110</td><td>6</td></tr> <tr><td>0111</td><td>8</td></tr> <tr><td>1000</td><td>15</td></tr> <tr><td>1001</td><td>20</td></tr> <tr><td>1010</td><td>30</td></tr> <tr><td>1011</td><td>40</td></tr> <tr><td>1100</td><td>50</td></tr> <tr><td>1101</td><td>100</td></tr> <tr><td>1110</td><td>200</td></tr> <tr><td>1111</td><td>500</td></tr> </tbody> </table> | | Register Value | Filtering time (unit: ms) | 0000 | 10 | 0001 | 1 | 0010 | 2 | 0011 | 3 | 0100 | 4 | 0101 | 5 | 0110 | 6 | 0111 | 8 | 1000 | 15 | 1001 | 20 | 1010 | 30 | 1011 | 40 | 1100 | 50 | 1101 | 100 | 1110 | 200 | 1111 | 500 | | |
| | | | | Register Value | Filtering time (unit: ms) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0000 | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0001 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0010 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0011 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0100 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0101 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0110 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0111 | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 1000 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 1001 | 20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 1010 | 30 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 1011 | 40 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 1100 | 50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 1101 | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 | 200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 | 500 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Note:</p> <p>(1) The values of the registers correspond to the individual filter times. The default state is 0000, which correspond to 10ms</p> <p>(2) If you need to set other filter times for the IO port, you can fill in the value manually in the parameter manage table of STEPPERONLINE MotionStudio.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Example 1: IO input port 1 is set to enable function. Normally closed. Filter time 50ms
 The register value is configured as 0000 1100 1000 1000, which is converted to 3208 in decimal, i.e. write 3208 to Pr4.02 to achieve the above configuration.

Example 2: IO input port 1 is set to enable function. Normally closed. Filter time 10ms (default)
 The register value is configured as 0000 0000 1000 1000, which translates to 136 in decimal, i.e. write 136 to Pr4.02 to achieve the above configuration.

Routine 3: IO input port 7 is set to JOG2 function. Normally open. Filter time 500ms
 The register configuration is 0000 1111 0010 1100, which is converted to 3884 in decimal, i.e. write 3884 to Pr4.08 to achieve the above configuration.

4.3.4 Status Monitoring Parameters

| Register address | Definition | Attributes | Unit | Description |
|---|--------------------------|------------|--------|---|
| 0x1001 | Control Mode | R | / | Invalid, always "0" |
| 0x1003 | Motion state | R | / | Bit0: 0--normally, 1--faulty; Bit1: 0--drive disable, 1--enable; Bit2: 0--not running, 1--running; Bit4: 0-- Command not completed, 1-- Command completed; Bit5: 0-- Path not completed, 1-- Path completed; Bit6: 0--Homing not completed, 1--Homing completed. |
| 0x1010 (high 16-bit) 0x1011 (low 16-bit) | Position following error | R | pulses | - |
| 0x1012 (high 16-bit) | Profile position | R | pulses | - |

| | | | | |
|---|-------------------|---|--------|---|
| 0x1013 (low 16-bit) | | | | |
| 0x1014 (high 16-bit) 0x1015 (low 16-bit) | Feedback position | R | pluses | - |
| 0x1044 (high 16-bit) 0x1045 (low 16-bit) | Profile velocity | R | rpm | - |
| 0x1046 (high 16-bit) 0x1047 (low 16-bit) | Feedback velocity | R | rpm | - |

4.3.5 Control Word and Status Word

- (1) The related function is started by sending the control word,
- (2) The completion is judged by checking the status word. The status word is automatically returned to its initial state after it is read.

■ **Control Word**

| Register address | Definition | Attributes | Description |
|------------------|--------------|------------|---|
| 0x1801 | Control word | W/S | Write 0x1111: Reset current alarm Write 0x1122: Reset history alarm Write 0x2211: Save all parameters to EEPROM Write 0x2222: Parameter reset (exclude motor parameters) Write 0x2233: All parameters are reset to factory Write 0x2244: Save all mapping parameters into EEPROM Write 0x4001: JOG CW (Need to write once at least 50ms) Write 0x4002: JOG CCW |

■ **Status Word of Saving Parameter**

| Register address | Definition | Attributes | Description |
|------------------|-------------|------------|---|
| 0x1901 | status word | R | Show 0x5555: Saving parameter OK Show 0xAAAA: Saving parameter fault |

- Note: (1) The read value is 0x1111 when no save instruction has been executed after the first power-up;
 (2) The first read value is 0x5555 after a save instruction is executed, and then it changes back to 0x1111.

4.4 Error Codes and Troubleshooting

4.4.1 Communication Error Codes

When the master station receives a message from the slave about a communication error, you can follow the table below for analysis

| No. | Return Commands (slave->master) | | |
|-----|---------------------------------|---------------|---------|
| | Symbols | Description | Content |
| 1 | ID | Slave ID | 0-31 |
| 2 | FC | Function code | FC+0x80 |
| 3 | EC | Error Code | - |
| 4 | CRC | check code | Low |
| | | | High |

■ **Error Code**

| Error Code (EC) | Description |
|-----------------|--|
| 0x01 | Wrong FC (This supports FC beside of 03h/ 06h/ 10h) |
| 0x02 | Wrong access address |
| 0x03 | Wrong data, for example, write data over-limit value, etc. |
| 0x08 | Wrong CRC check code |

Example F: CRC check code error

Master-> slave data:

| | | | | | |
|-------------|----------|---------------|------------------|--------------------------|----------------|
| Message | 01 | 03 | 00 01 | 00 01 | D5 C1 |
| Description | Slave ID | Function code | Register address | Read Number of registers | CRC check code |

slave-> Master data:

| | | | | |
|-------------|----------|---------|------------|-------|
| Message | 01 | 83 | 08 | 40 F6 |
| Description | Slave ID | FC+0x80 | Error code | CRC |

Example F: Function code error

Master-> slave data:

| | | | | | |
|-------------|----------|---------------|------------------|--------------------------|----------------|
| Message | 01 | 02 | 00 01 | 00 01 | E8 0A |
| Description | Slave ID | Function code | Register address | Read Number of registers | CRC check code |

slave-> Master data:








| | | | | |
|-------------|----------|----------------------|------------|----------------|
| Message | 01 | 82 | 01 | 81 60 |
| Description | Slave ID | Function code + 0x80 | Error code | CRC check code |

4.4.2 Drive Alarm Codes and Troubleshooting

| Register Address | definition | operation | unit | description |
|------------------|---------------|-----------|------|-------------|
| 0x2203 | Current alarm | R | / | Below table |

Error code and cause:

The green light is always on after the drive power on. When the error is occurred, the drive will stop working and red light will be flashed indicates the current error code. Whatever error occurs, the user need to power off the drive and restart it after removing the error. The user can read the corresponding error code through the PC software. The latest errors will be saved to EEPROM which supports 10 historical errors in the list.

| Error code | Content | Red Light Flash Times | Sequence wave of RED LED | Trouble Shooting |
|------------|-------------------------------------|-----------------------|--|--|
| 0x01 | Over current | 1 |  | 1. Restart the drive; 2. If it still exists, check whether the motor is short-circuited or not connected to the motor; |
| 0x02 | Over voltage | 2 |  | 1. Restart the drive; 2. If it still exists, check the voltage of power supply; |
| 0x40 | Current sampling circuit error | 3 |  | 1. Restart the drive; 2. If it still exists, the hardware failure |
| 0x80 | Shaft locking error | 4 |  | 1. Check whether the motor wire is broken |
| 0x200 | EEPROM error | 5 |  | 1. Connect the drive to STEPPERONLINE software to reset parameters to the factory 2. If it still exists, the hardware failure |
| 0x100 | Auto tuning error | 6 |  | 1. Restart the drive; 2. If it still exists, disable auto-tuning by 0x01AB (Pr5.13). |
| - | Repeated settings of input function | 9 |  | |

4.4.3 Error Clear

Check the error via STEPPERONLINE MotionStudio

| | | |
|---------------------|--------------------------|---|
| Current Error | Current happened error | Over current,over voltage,etc |
| History error | History happened error . | Over current,over voltage,etc |
| Read error | Read Historical error . | Check the Historical happened error . |
| Clear current error | Current error | (1) It can clear over voltage error, but can not clear over current error; (2) There are two methods to clear the current alarm: one is via STEPPERONLINE MotionStudio, and the second is via external I/O, refer to Section 4.3.2 . (3) If the current error cannot be cleared, please check the drive |
| Clear history error | History error | All history error records can be cleared by STEPPERONLINE MotionStudio |

4.5 Register Mapping Continuous Read/Write Function

Address description 0x0F10-0x0F19. By writing the "address to be mapped" to 0x0F10-0x0F19, it is possible to set the address mapping. The 10 consecutive mapped addresses are available for mapping discontinuous parameter addresses.

Examples

| Mapping target address | Write to | Parameter original address |
|------------------------|----------|--|
| 0x0F10 | ← | 0x0001 (Pr0.00: Microstep) |
| 0x0F11 | ← | 0x0009 (Pr0.04: inductance value) |
| 0x0F12 | ← | 0x00A1 (Pr2.00: instruction filter time) |
| 0x0F13 | ← | 0x0191 (Pr5.00: peak current) |
| 0x0F14 | ← | 0x0167 (Pr4.19: delay of releasing brake) |
| 0x0F15 | ← | 0x0173 (Pr4.25: software de-jitter delay of position when in position) |
| 0x0F16 | ← | 0x0233 (Pr7.01: encoder resolution) |
| 0x0F17 | ← | 0x0243 (Pr7.09: over-voltage threshold) |
| 0x0F18 | ← | 0x602E (Pr8.46: digital inputs) |
| 0x0F19 | ← | 0x6203 (Pr9.03: PR0 velocity) |

At this point, the mapping of the 10 parameter addresses to the mapped area is complete, and the mapped addresses can be used for reading and writing instead of the original parameter addresses. For example, if 0x0001 is written to 0x0F10, the operation of reading and writing to 0x0001 can be replaced by "read and write to 0x0F00". The mapped address is equivalent to a "stand-in" or "proxy" for the original address.

| Mapping target address | Reading and writing | Host |
|--|---------------------|------|
| 0x0F00 (Pr0.00: Microstep) | ←----→ | Host |
| 0x0F01 (Pr0.04: inductance value) | ←----→ | |
| 0x0F02 (Pr2.00: instruction filter time) | ←----→ | |
| 0x0F03 (Pr5.00: peak current) | ←----→ | |
| 0x0F04 (Pr4.19: delay of releasing brake) | ←----→ | |
| 0x0F05 (Pr4.25: software de-jitter delay of position when in position) | ←----→ | |
| 0x0F06 (Pr7.01: encoder resolution) | ←----→ | |
| 0x0F07 (Pr7.09: over-voltage threshold) | ←----→ | |
| 0x0F08 (Pr8.46: digital inputs) | ←----→ | |
| 0x0F09 (Pr9.03: PR0 velocity) | ←----→ | |

Message Example

Conditions: Drive ID is 1, motor is stationary.

(1) Mapping

Master→Slave:

01 10 0F 10 00 0A 14 00 01 00 09 00 A1 01 91 01 67 01 73 02 33 02 43 60 2E 62 03 1C 56

Slave → Master:

01 10 0F 10 00 0A 42 DF

(2) Read and write

Master→Slave:

01 03 0F 00 00 0A C6 D9

Slave → Master:

01 03 14 27 10 05 87 00 0F 00 3C 00 FA 00 03 0F A0 00 5A 00 01 00 00 00 56 F4

(3) Mapping parameter save instruction 0x2244

Master→Slave:

01 06 18 01 22 44 C6 39 (Note: Write 0x2244 to 0x1801, the function is to save the mapped address into EEPROM)

Slave → Master:

01 06 18 01 22 44 C6 39

4.6 S-code Application

S-code (status code) is the status indicator of the currently executed PR path, whether it is running, or the path is completed.

In a normal path program, no matter which path is completed, the drive will output the run status uniformly, but it is difficult to identify which path. The S-code function can help to identify the specific path number and running status, or read the value of Pr8.38(0x601C). Each PR path can set an S-code, i.e. each path has a dedicated S-code register, the exact S-code is user-defined according to the table format below.

| Size S | High 8 bits | | Lower 8 bits | |
|-------------|---|----------------------|--|---------------------|
| # Bit | bit15 | bit8-10 | bit7 | bit0-2 |
| Description | Whether the S code is valid at the time of completion: 0: Invalid, keep last value 1: Effective (Recommended) | S-code at completion | Whether the start S code is valid or not: 0: Invalid 1:Effective (Recommended) | S-code at start -up |
| | bit7 and bit15 are set to 1. The rest of the unused bits are 0, such as bits 11-14, bits 3-6. | | | |

DMRS series drivers have only 3 outputs, and the S-code can only use 3 bits, each bit corresponds to an output. Therefore, there are only 8 output combinations (000, 001, 010, 011, 100, 101, 110, 111), and these 8 states can be set freely, depending on the requirements.

Example:

Here is an example of paths 1-4:

- All output ports are set to normally open.
- “1” indicates that the output optocoupler is switched on.
- “0” indicates that the output optocoupler is switched off.
- “001” means that output port 1 is on, and outputs 2 and 3 are not on.
- Pr8.28: S-code current output value; register address: 0x601C.

| Path number | S-code at completion bit8-10 | S-code at start-up bit0-2 | S code setting value | Remarks |
|-------------|------------------------------|---------------------------|--|---|
| Path 1 | 001 | 000 | Binary: 1000 0001 1000 0000 Hexadecimal: 0x8180 | At start-up: No level at all three outputs Completion: Output port 1 has level, PR8.28=1 |
| Path 2 | 011 | 010 | Binary: 1000 0011 1000 0010 Hexadecimal: 0x8382 | At start-up: Output port 2 has level On completion: Outputs 1 and 2 have levels, PR8.28 = 3 |
| Path 3 | 101 | 100 | Binary: 1000 0101 1000 0100 Hexadecimal: 0x8584 | At start-up: Output port 3 has level On completion: Outputs 1 and 3 have levels, PR8.28 = 5 |
| Path 4 | 111 | 110 | Binary: 1000 0111 1000 0110 Hexadecimal: 0x8786 | At start-up: Outputs 2 and 3 are level When finished: output ports 1. 2. 3 are level, PR8.28=7 |

4.7 Enable Drive

There are two methods to enable DMRS series drives:

- I/O enable:
DI1 of DMRS series drive is the enable input by default, normally closed, so DMRS drive immediately enters the enable state after power on.
- RS485 communication:
Pr0.07 of STEPPERONLINE MotionStudio (Register address 0x00F) set to value "1" indicates enable the drive by RS485

5 PR Mode (Indexer Table)

PR mode is a single-axis motion control function with 16-segment position table, also called indexer table. It can save the motion control function of the controller.

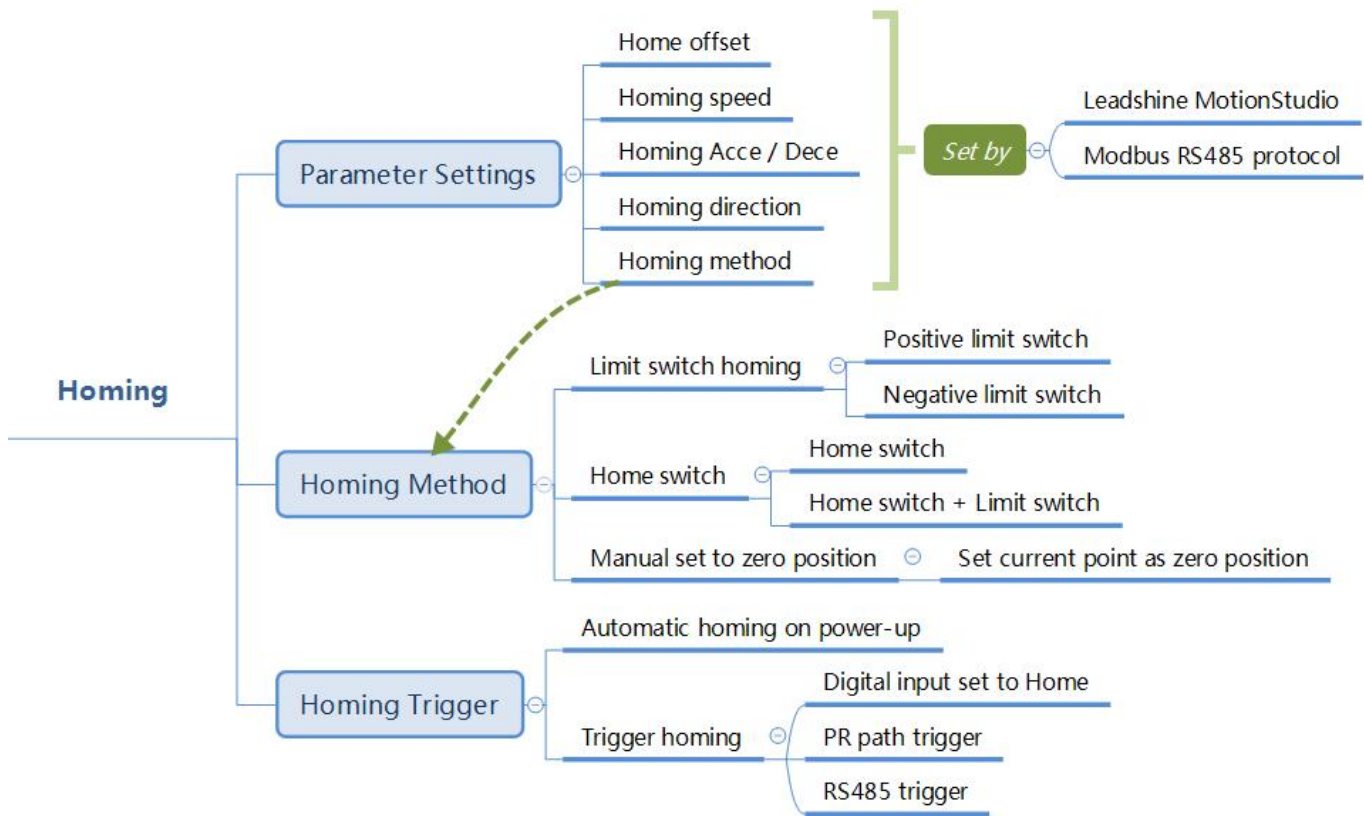
5.1 PR Main Features

PR mode can mainly set the following functions:

| Features | Description |
|--------------|---|
| Homing | <p>The drive can detect zero position by homing processing, and confirm the zero point of mechanical coordinate system</p> <ul style="list-style-type: none"> ● Select homing method: by limit switch signal, by home switch signal, auto homing after power up, set homing point by manual; ● Homing direction can be set; ● Homing offset can be set; ● Move to the specified position after homing; ● Acc/Dec of homing velocity can be set; <p><i>Note: Control instruction can not be sent to drive during homing</i></p> |
| JOG | <p>JOG can be realized by external I/O or RS485 communication.</p> <ul style="list-style-type: none"> ● CW JOG; ● CCW JOG; ● JOG teaching function; ● The JOG velocity and JOG Acc/Dec can be set; |
| Limit | <p>Protect the machine by limiting the working area range.</p> <ul style="list-style-type: none"> ● Positive/negative limit switch by digital input; ● Soft limit setting; ● Acc/Dec limit can be set <p><i>Note: Soft limit is valid after homing completed;</i></p> |
| Quick stop | <p>The movement stop immediately when the quick stop input is on, which is valid at PR mode only</p> |
| PR Path | <p>Select the PR path number by 4 digital inputs (ADD0, ADD1, ADD2, ADD3), and then start the PR path movement by external I/O (CTRG) or RS485 communication.</p> <ul style="list-style-type: none"> ● Include PP mode, PV mode and HM mode ● Trigger methods include: double edge trigger (CTGR) or RS485 trigger ● Support up to 16-segment PR positioning path ● Position, velocity and Acc/Dec values can be set ● Pause time or timing time can be set ● Support interrupt, Overlapping, jump, etc. |
| Path Trigger | <p>The above PR functions can be controlled by RS485</p> |

5.2 Homing / Return to Zero Position

Homing includes homing by detecting limit switch signal, homing by detecting home switch signal, set zero position by manual, auto homing after powering on.



Trigger method of homing:

- Automatic homing on power-up: After the drive is powered on, the motor will automatically search the zero position.
- Trigger to homing: when IO port set to Home function triggered by external level, or trigger via Modbus RS485.

Homing method:

- Limit switch homing: Set by register address 0x600A, or STEPPERONLINE software. If the homing direction is positive, then it is positive limit switch homing. Conversely, the negative limit homing.
- Home switch: Set by register address 0x600A, or STEPPERONLINE software. If the motor turns toward the home switch when zeroing, it will use the home switch it touches as the home point; if the motor turns in the opposite direction when zeroing, it will automatically reverse to find the home point after it touches the limit switch.
- Manual set to zero position: Set by register address 0x600A, or by tuning software. After triggering, the current value of the motor is cleared to zero and the current point is used as the origin.

Note: In the normal homing, because the motor to find the origin process is slowed down and stopped, so after finding the origin will move a certain distance, the actual read position value may not be 0, in this case, you can check the "return to the origin to move to the specified position", that is, move to the 0 position.

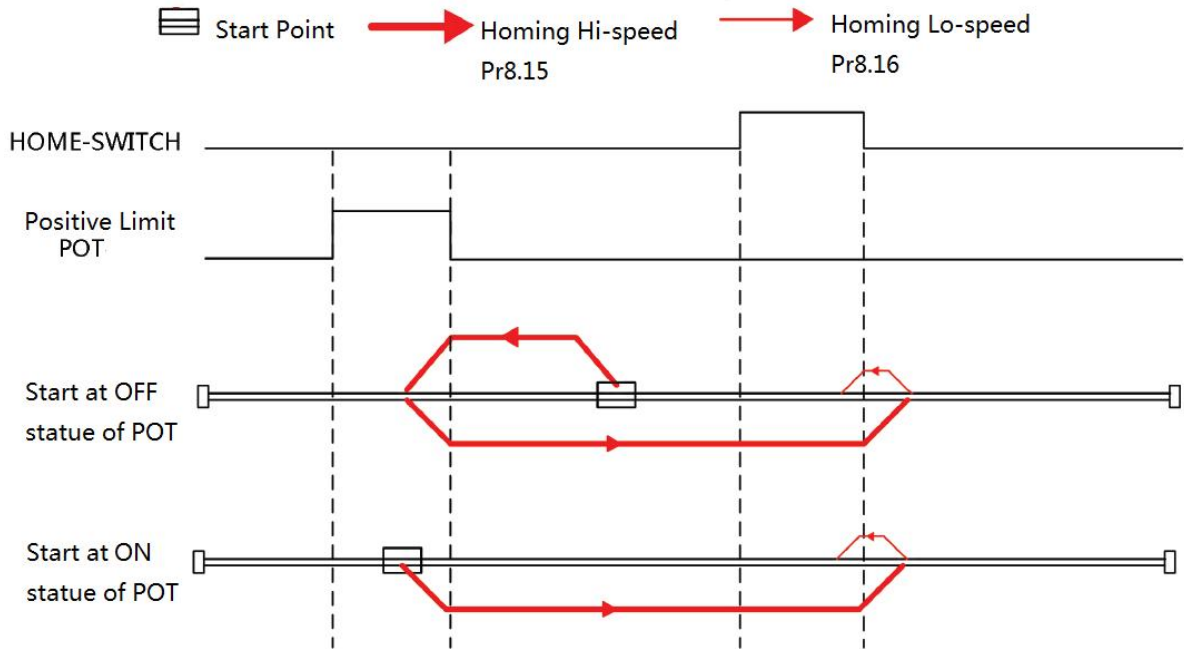
5.2.1 Homing Parameters

| Parameters | Register address | Definition | Description |
|--|------------------|--------------------------------|---|
| Pr8.10 | 0x600A | Homing mode | Bit0: homing direction =0:CCW; =1: CW. Bit1: move to the Specified point after homing? =0: No; =1: Yes. Bit2: homing type =0: Homing by detecting limit switch signal =1: Homing by detecting Home Switch signal Note: (1) Write 0x21 to the address 0x6002 to set the current point to zero position; (2) Write 1 to the bit2 of address 0x6000 can set to auto-homing after powering on; |
| Pr8.11 | 0x600B | Home Switch position high bits | The Home Switch position on the coordinate system . Pr8.11 is high 16 bits, and Pr8.12 is low 16 bits |
| Pr8.12 | 0x600C | Home Switch low bits | |
| Pr8.13 | 0x600D | Homing stop position high bits | The motor moves to the Specified position after homing. If bit1 of Pr8.10=1, the motor will move to Specified absolute position. Pr8.13 is high 16 bits , and Pr8.14 is low 16 bits. |
| Pr8.14 | 0x600E | Homing stop position low bits | |
| Pr8.15 | 0x600F | Homing high velocity | The 1 st segment velocity of homing, unit: rpm |
| Pr8.16 | 0x6010 | Homing low velocity | The 2 nd segment velocity of homing, unit: rpm |
| Pr8.17 | 0x6011 | Homing Acc | Acc of homing, unit: ms/1000rpm |
| Pr8.18 | 0x6012 | Homing Dec | Dec of homing, unit: ms/1000rpm |
| Note: Generally homing to find the zero position, the motion is decelerating to stop, so after finding the zero position will also move a distance, resulting in the actual read position value may not be mechanical zero point, but will still output homing completion signal. If the motion is an absolute position mode, it does not matter. But if the user does care about the zero position, you can set the value of home offset to reach the mechanical zero point | | | |

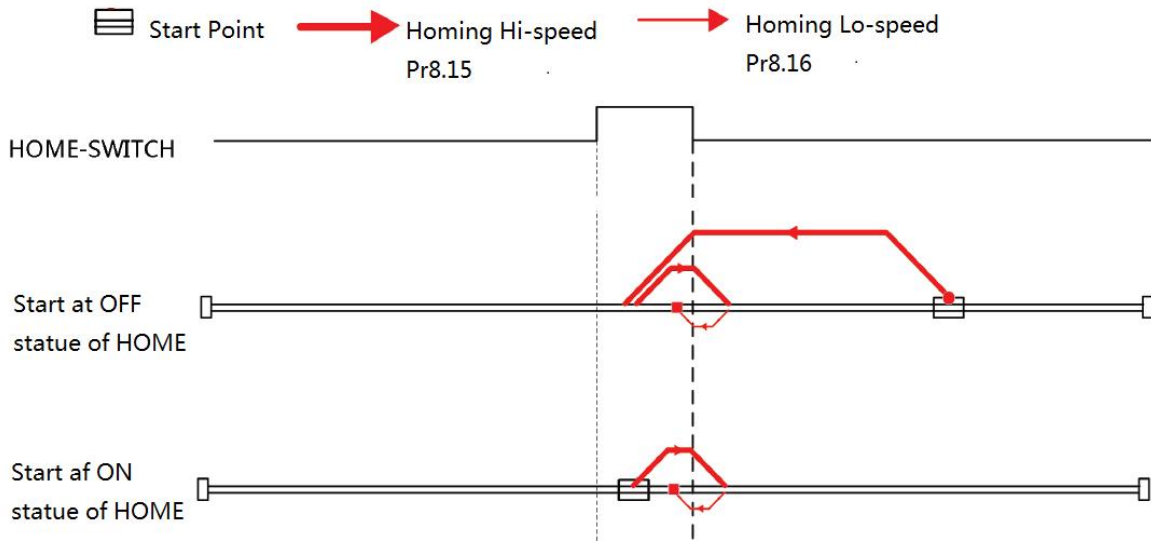
5.2.2. Homing by Home Switch

Set value 1 to the bit2 of 0x600A, and then according to the homing direction, positive or negative limit switch and location of home switch, it can be combined into the following four cases.

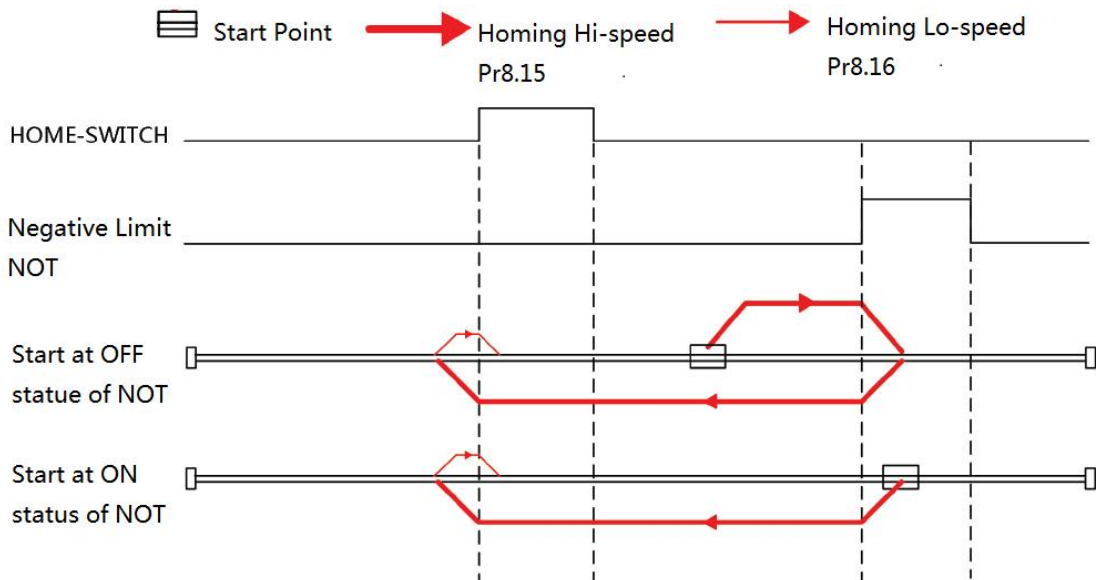
(1) Home Switch & Positive Limit Switch



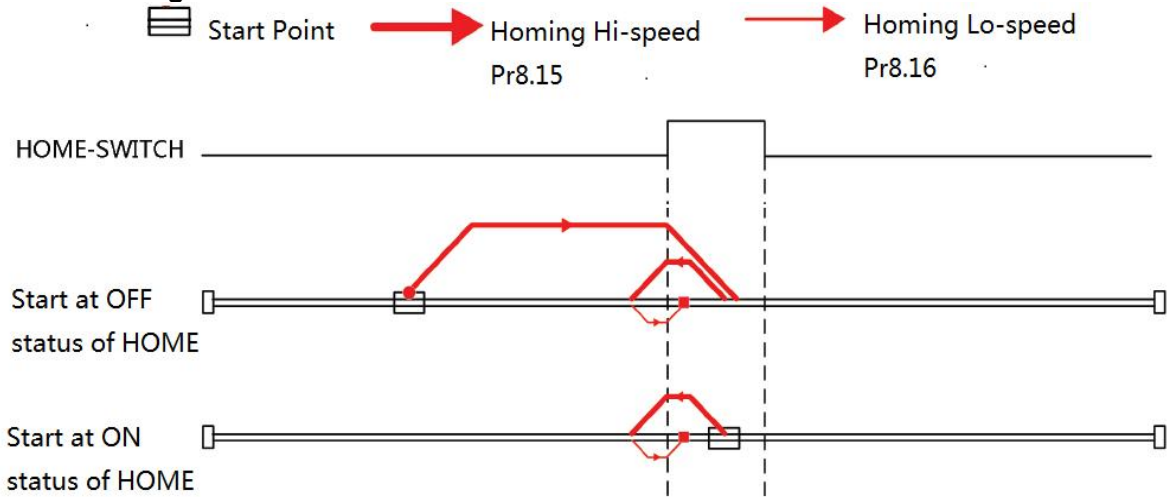
(2) Home Switch at Positive Direction



(3) Home Switch & Negative Limit Switch

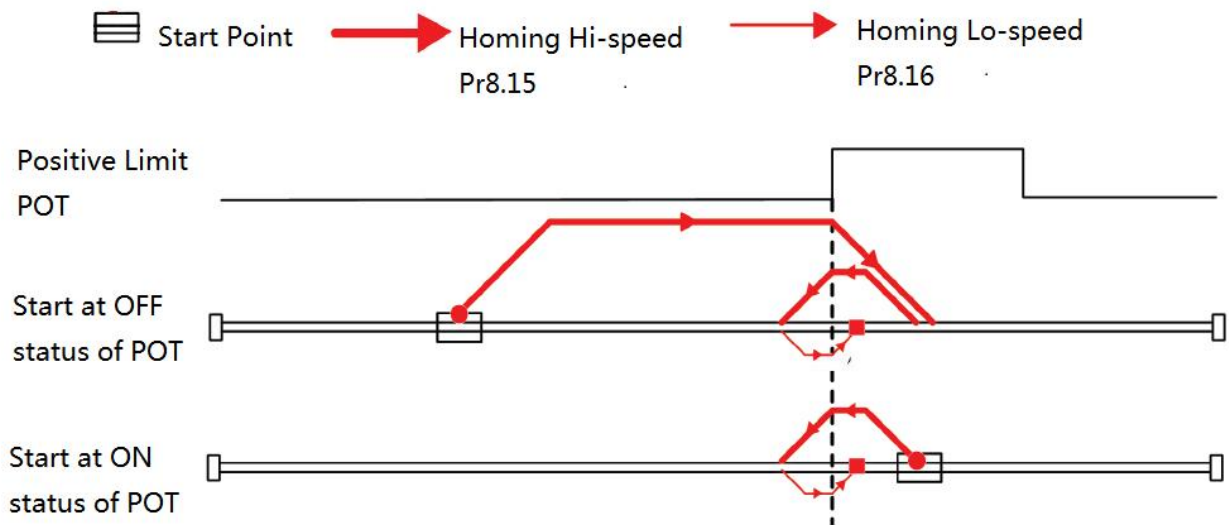


(4) Home Switch at Negative Direction

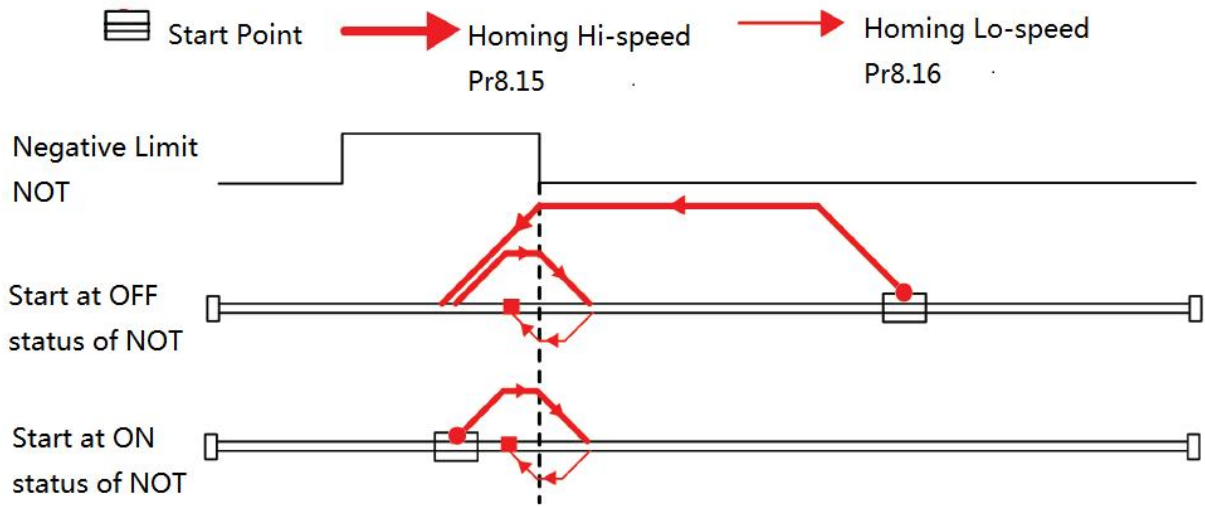


5.2.3. Homing by Limit Switch

(1) Positive Limit Switch



(2) Negative Limit Switch



5.3 Soft Limit & JOG & Quick Stop

5.3.1 Soft Limit

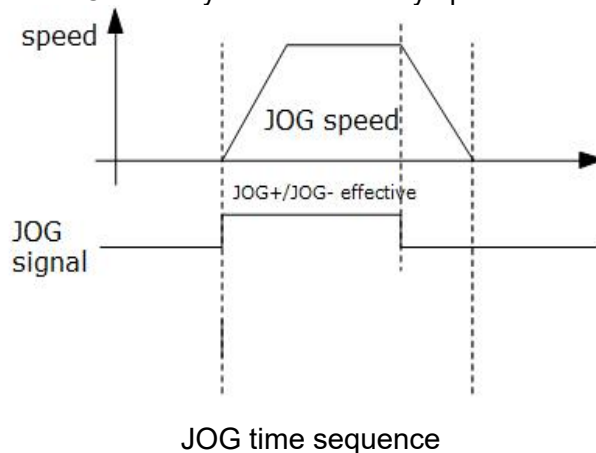
The soft limit function means that the internal position feedback of the drive is compared with the limit position, an alarm and stop when determining that the motor exceeds the limit position. Soft limit function is universal for linear movement and rotating movement. It is not requiring hardware, eliminating malfunction due to poor wiring contact, and it can prevent mechanical slip and abnormal action with internal position comparison. And the DMRS series drives carry out homing to find the mechanical home before the soft limit function can be activated.

Relevant objects:

| Register address | Par. # in software | Definition | Range | Default Value | Description |
|------------------|--------------------|----------------------------|-------|---------------|---|
| 0x6000 | Pr8.00 | PR control setting | 32767 | - | Bit1: Soft limit.. 0--disable, 1--enable |
| 0x6006 | Pr8.06 | Soft limit+ H | 32767 | 0x7FFF | Soft limit positive high bits |
| 0x6007 | Pr8.07 | Soft limit+ L | 32767 | 0xFFFF | Soft limit positive low bits |
| 0x6008 | Pr8.08 | Soft limit- H | 32767 | 0x8000 | Soft limit negative high bits |
| 0x6009 | Pr8.09 | Soft limit- L | 32767 | 0 | Soft limit negative low bits |
| 0x6016 | Pr8.22 | Soft limit quick stop time | - | - | Deceleration time after triggering the soft limit, unit: ms |

5.3.2 JOG

The JOG function allows fine tuning of the drive, or fine adjustment of the position of the device, or forward and reverse velocity operation of the motor. Commonly used for velocity operation.



Relevant objects:

| Register address | Par. # in software | Definition | Default Value | Description |
|------------------|--------------------|------------------|---------------|---|
| 0x6026 | Pr8.38 | JOG 2 velocity | 0 | JOG triggered by external I/O signal, Unit: rpm |
| 0x6027 | Pr8.39 | JOG 1 velocity | 0 | |
| 0x6028 | Pr8.40 | JOG Acceleration | 0 | Unit: ms/1000rpm |
| 0x6029 | Pr8.41 | JOG Deceleration | 0 | Unit: ms/1000rpm |
| 0x10E1 | Pr6.00 | JOG velocity | 60 | Unit: rpm |
| 0x01E3 | Pr6.01 | Interval | 100 | Unit: ms |
| 0x01E5 | Pr6.02 | Running times | 1 | Unit: ms/1000rpm |
| 0x01E7 | Pr6.03 | Acc. / Dec. time | 200 | Unit: ms/1000rpm |

Note:(1) Pr8.39 - Pr8.41(JOG) are available when triggered by I/O; (2) Pr6.00 - Pr6.03 (JOG) are available when triggered by RS485.

The trigger mode of JOG is divided into two types: RS485 communication and IO, which correspond to different parameter addresses:

RS485 communication trigger JOG:

- Write value 0x4001 to 0x1801, JOG+.
- Write value 0x4002 to 0x1801, JOG-.
- JOG velocity: Pr6.00 (0x01E1).
- JOG acceleration and deceleration time: Pr6.03 (0x01E7).
- JOG quick stop: write value 0x40 to 0x6002.

Note: JOG triggered by RS485 communication will run continuously only if the trigger interval is less than 50ms, otherwise it can only point operation.

IO trigger JOG:

- Assign JOG+ and JOG- to IO port. Then through input level to trigger JOG function.
- JOG velocity: Pr8.39 (0x6027).
- JOG acceleration time: Pr8.40 (0x6028).
- JOG deceleration time: Pr8.41(0x6029)
- JOG quick stop: write value 0x40 to 0x6002; or trigger forced quick stop input to IO port; or stop giving IO port level if using level trigger method.

Example for JOG+ /- and IO Trigger

DMRS uses JOG+/JOG- to achieve forward and reverse rotation operation; the running velocity can be set in JOG velocity and JOG velocity 2, then through the external IO signal to realize the two velocity switching. During the is running, the value of velocity can be modified and effect in real time. The path motion can also be triggered by external IO.

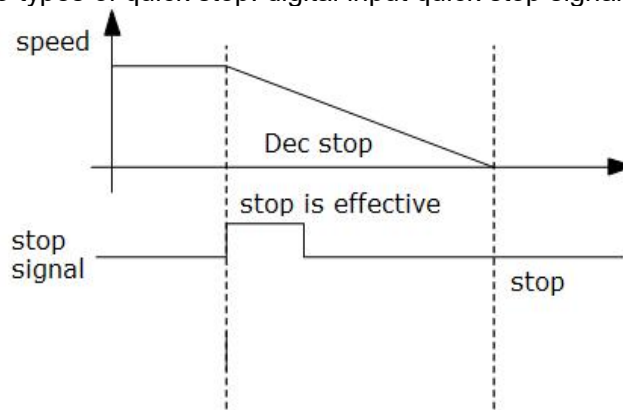
- SI1 is set to enable input by default, normally closed.
- SI2 is set to JOG+, used for positive directional velocity operation.
- SI3 is set to JOG-, used for negative directional velocity operation.
- SI6 is set to JOG velocity 2, used for JOG high and low velocity switching.
- SI4 is set to trigger signal CTRG, which is used to trigger path motion.
- SI5 is set to ADD0, used to select path number.

Note: the value of Pr8.38 is as JOG velocity 2, the value of Pr8.39 is as JOG velocity.

- When SI2 is switched on, the motor runs in the positive direction of JOG velocity.
- When SI3 is switched on, the motor runs in negative JOG velocity.
- When SI6 is switched on, the motor switches to JOG velocity 2 operation.
- If SI2and SI3are not switched on, the motor stops.
- When SI4 is switched on, start running on Path 0, [refer to Section 5.4.2](#)
- When SI5 is switched on, then SI4 on, and start running on Path 1, [refer to Section 5.4.2](#)

5.3.3 Quick Stop

The DMRS series drives have two types of quick stop: digital input quick stop signal and register quick stop.



Quick stop time sequence

Relevant objects:

| Register address | Par. # in software | Definition | Range | Default Value | Description |
|------------------|--------------------|------------------|-------|---------------|--|
| 0x6017 | Pr8.23 | Quick stop time | | | Deceleration time after quick stop, unit: ms |
| 0x6002 | Pr8.02 | Trigger register | | | Write value 0x040---- E-stop; |

5.4 PR Path

The PR path can be run with single segment movement or continuous movement, which includes three operation modes: position mode, velocity mode and homing mode. There are 16 PR paths, and each path sets the operation modes, the target position, the target velocity, the acceleration and deceleration and the pause time, etc.

5.4.1 PR Parameters

Usually it is recommended using the PTP window of the STEPPERONLINE tuning software to configure the PR path parameters, but it can also use the following objects:

| Par. # in software | Register Address | Definition | Description |
|--------------------|------------------|------------|--|
| Pr9.00 | 0x6200 | PR path 0 | The corresponding functions can be selected for different bit Bit0-3: Operation mode =0---- no action =1---- position mode =2---- velocity mode =3---- homing mode; Bit4: INS, =0---- No interrupt =1---- interrupt(all the current ones are 1.); Bit5: OVLP, =0---- Non overlapping =1---- Overlapping Bit6: =0----absolute position =1----relative position Bit8-13: Jump to the corresponding PR path 0-15; bit14: JUMP, =0---- No jump =1---- jump |
| Pr9.01 | 0x6201 | Position H | High 16 bit, |

| | | | |
|----------------|------------------------|------------------------|--|
| Pr9.02 | 0x6202 | Position L | Low 16 bit |
| Pr9.03 | 0x6203 | velocity | Unit: rpm |
| Pr9.04 | 0x6204 | Acc | Unit: ms/1000rpm |
| Pr9.05 | 0x6205 | Dec | Unit: ms/1000rpm |
| Pr9.06 | 0x6206 | Pause time | Pause time after the command is stopped |
| Pr9.07 | 0x6207 | Special parameter | PR Path 0 maps directly to Pr8.02, Others are reserved |
| Pr9.08 | 0x6208 | PR path 1 | --- |
| Pr9.09 | 0x6209 | Position | --- |
| Pr9.10 | 0x620A | Position | --- |
| Pr9.11 | 0x620B | velocity | --- |
| Pr9.12 | 0x620C | Acc | --- |
| Pr9.13 | 0x620D | Dec | --- |
| Pr9.14 | 0x620E | Pause time | --- |
| Pr9.15 | 0x620F | Special parameter | --- |
| Pr9.16 | 0x6210 | PR path 2 | --- |
| Pr9.17 | 0x6211 | Position | --- |
| Pr9.18 | 0x6212 | Position | --- |
| Pr9.19 | 0x6213 | velocity | --- |
| Pr9.20 | 0x6214 | Acc | --- |
| Pr9.21 | 0x6215 | Dec | --- |
| Pr9.22 | 0x6216 | Pause time | --- |
| Pr9.23 | 0x6217 | Special parameter | --- |
| Pr9.24- Pr9.31 | Similar as above paths | Similar as above paths | PR path 3 |
| Pr9.32- Pr9.39 | Similar as above paths | Similar as above paths | PR path 4 |
| Pr9.40- Pr9.47 | Similar as above paths | Similar as above paths | PR path 5 |
| Pr9.48- Pr9.55 | Similar as above paths | Similar as above paths | PR path 6 |
| Pr9.56- Pr9.63 | Similar as above paths | Similar as above paths | PR path 7 |
| Pr9.64- Pr9.71 | Similar as above paths | Similar as above paths | PR path 8 |

5.4.2 PR Path Configuration

If use the digital input ports to configure the PR path, they can be set to ADD0, ADD1, ADD2 and ADD3, thus forming 16-segment PR path, and then trigger the path number to complete the PR motion.

Each digital input of the drive can be configured as any one of ADD0 - ADD3, then combine the path numbers, as shown in the table below, "on" means the digital input is configured as ADDx (x=1-3), and the input signal optocoupler on, "off" means signal input off.

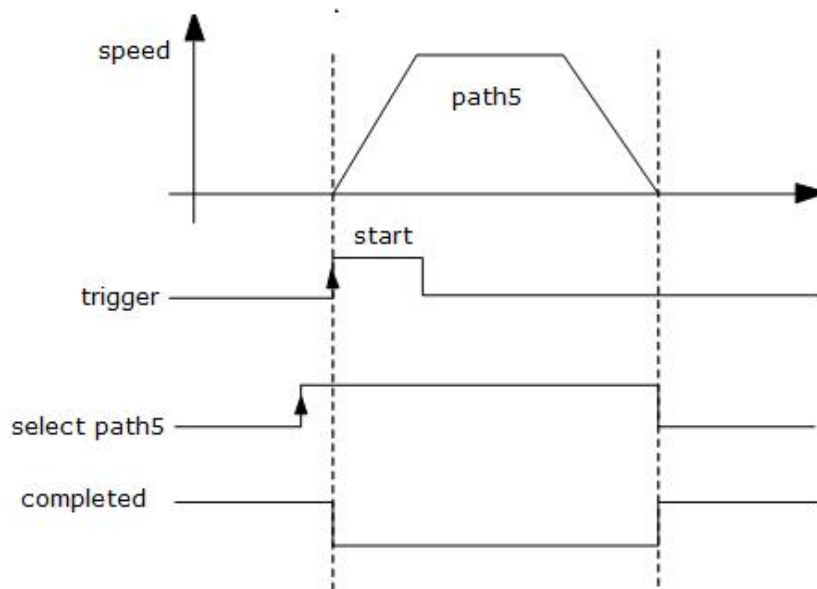
| PR path | DI1 (ADD0) | DI2 (ADD1) | DI3 (ADD2) | DI4 (ADD3) | DI5 (CTRG) |
|---------|---------------|---------------|---------------|---------------|---------------|
| Path 0 | off | off | off | off | on |
| Path 1 | on | off | off | off | on |
| Path 2 | off | on | off | off | on |
| Path 3 | on | on | off | off | on |
| Path 4 | off | off | on | off | on |
| Path 5 | on | off | on | off | on |
| Path 6 | off | on | on | off | on |
| Path 7 | on | on | on | off | on |
| Path 8 | off | off | off | on | on |
| Path 9 | on | off | off | on | on |

| | | | | | |
|---------|-----|-----|-----|----|----|
| Path 10 | off | on | off | on | on |
| Path 11 | on | on | off | on | on |
| Path 12 | off | off | on | on | on |
| Path 13 | on | off | on | on | on |
| Path 14 | off | on | on | on | on |
| Path 15 | on | on | on | on | on |

Note: The 4 digital inputs do not have to be configured as ADD0 - ADD3, it is configured according to the actual required path motions. For example, if only Path0 is needed, refer to the above table, DI1-4 are not required to be configured as ADD0 - ADD3, because they are "off", only one input needs to be configured as CTRG.

5.4.3 Other functions of PR

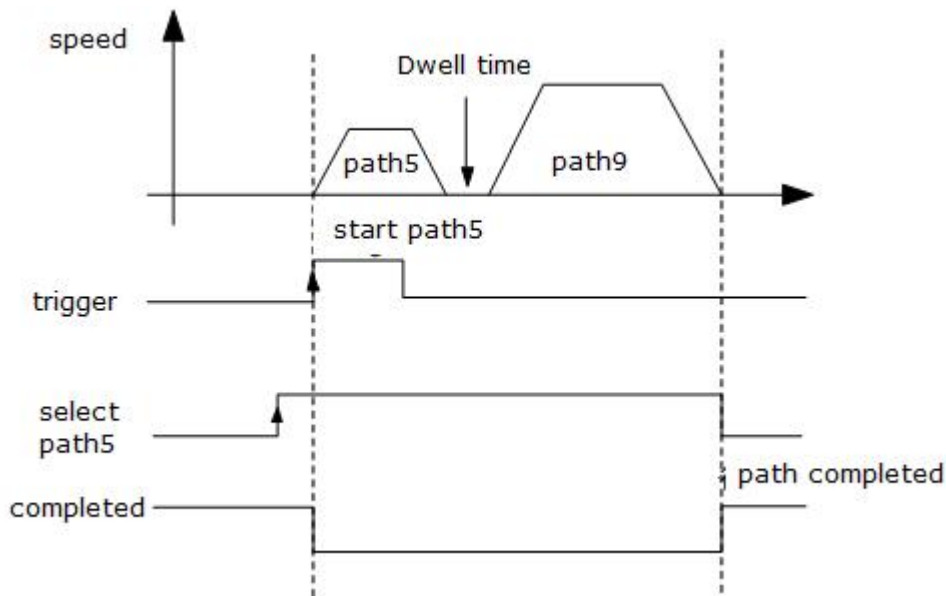
● **Timing sequence of signal path**



Single path sequence diagram

● **5.5.3 Multi-segment Jump**

For example: set paths 5 and 9, set path 5 to jump to path 9.

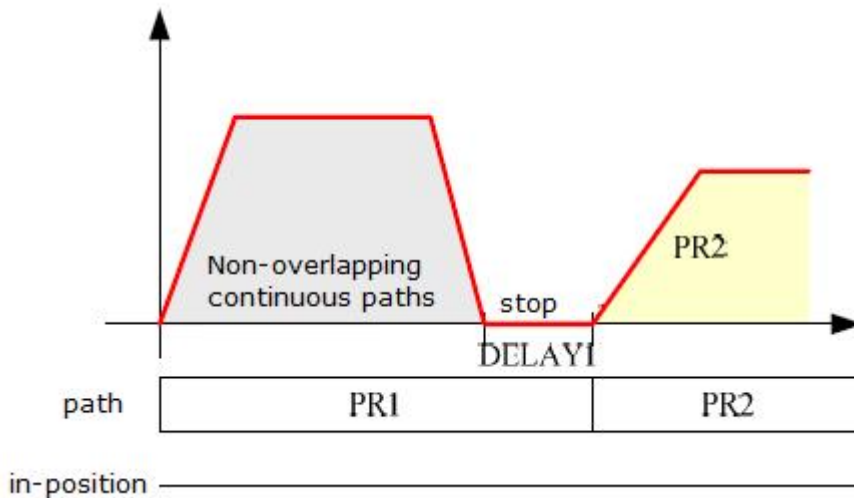


Multi-segment jump path sequence diagram

● **Continuous movement**

The bit5 of Pr9.00 is 0 , which does not overlap the continuous path.

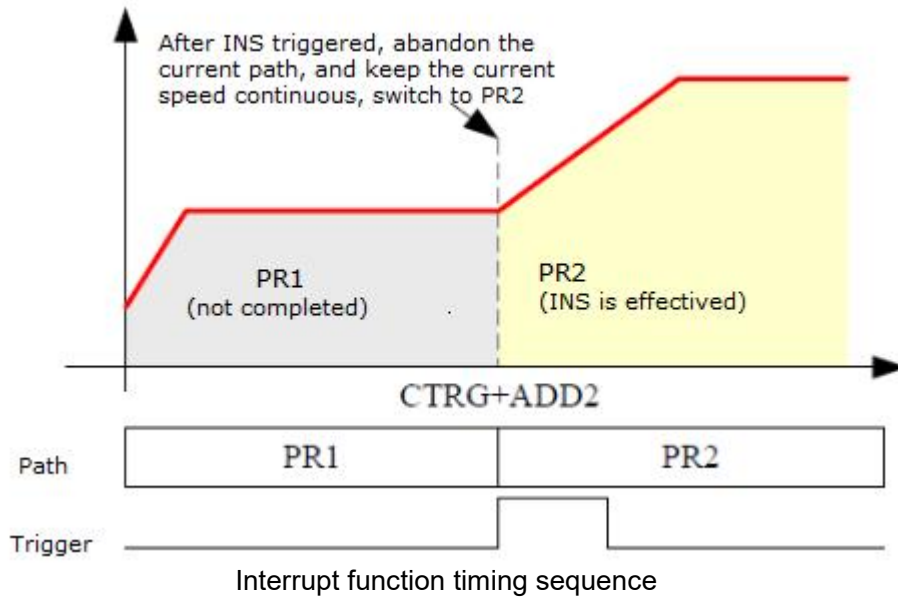
As shown in the figure below, set PR1 (PR path1) and PR2 (PR path2) to run continuously and PR1 jump. There is no in-position signal in the intermediate delay from PR1 jump to PR2.



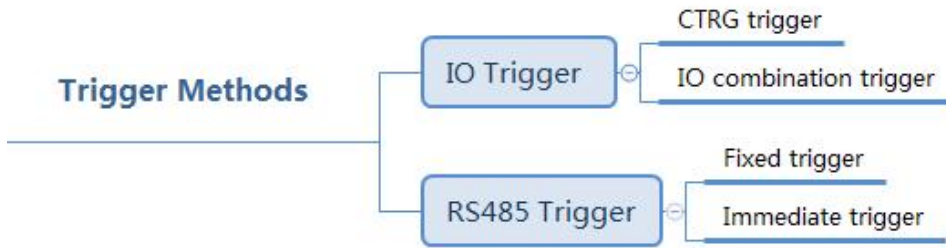
Continuous movement timing sequence (no overlap).

● **Interrupt function**

The interrupt function is the priority of a PR path. Interrupts a valid path means that interrupting and abandoning the current path under trigger, and runs another path directly, which is similar as Interrupt priority of function.. As below example, interrupt the PR1 (PR path1) to run PR2 (PR oath2)



5.5 Trigger Methods



5.5.1 CTRG Trigger

When the path triggered by CTRG, it can be configured for rising edge triggering or double edge triggering. (No falling edge trigger function).

| Par. # | Register address | Definition | Description |
|--------|------------------|--------------------|---|
| Pr8.00 | 0x6000 | PR control setting | Global Control function of PR: Bit0: CTRG =0: Rising edge trigger =1: Double edge trigger; Bit1: =0: Soft limit is invalid =1: Soft limit is valid; Bit2: =0: homing after power up is invalid =1: homing after power up is valid; |

For example: 4-segment PR path

The CTRG trigger method requires not only the configuration of the required number of paths, but also the setting of an additional input to the CTRG signal

- SI4 is set to enable input by default, normally closed.
- SI1 is set to ADD0, normally open,
- SI2 is set to ADD1, normally open,
- Set SI5 as trigger (CTRG).

- When CTRG is on, SI1 and SI2 are off, the path 0 is triggered;
- When CTRG and SI1 are on, and SI2 is off, the path 1 is triggered;
- When CTRG and SI2 are on, and SI1 is off, the path 2 is triggered;
- When CTRG, SI1 and SI2 are all on, the path 3 is triggered.

As shown in the table below, on means signal input on, off means signal input off

| IO/running path | SI1 (ADD0) | SI2 (ADD1) | SI5 (CTRG) |
|-----------------|---------------|---------------|---------------|
| Path 0 | off | off | on |
| Path 1 | on | off | on |
| Path 2 | off | on | on |
| Path 3 | on | on | on |

5.5.2 IO Combination Trigger

The configuration is done by the parameter Pr8.26 Combination trigger mode. With the IO combination trigger, there is no need to set one digital input as the trigger signal (CTRG), which saves IO port and makes the control simpler. After programming the path number, the path can be triggered directly by ADD0-ADD3. (In this mode, PR0 is not available.) The parameter Pr8.26=2 enables this function.

| Par. # | Register address | Definition | Description |
|--------|------------------|-------------------------------|--|
| Pr8.26 | 0x601A | IO combination trigger method | 0: Turn off IO combination trigger 1: Enable IO combination trigger, return to zero OK to be effective 2: Enable IO combination trigger, no need to return to zero |

For example: 3-Segment PR Path

Path configuration example:

- SI1 is set to enable input by default, normally closed.
- SI2 is set to ADD0, normally closed, means ADD0 is "on".
- After triggering SI3, then SI2, SI3, SI4 are "on, on, off", the path 3 is triggered.
- After triggering SI3 and SI4, then SI2, SI3, SI4 are "on, on, on", the path 7 is triggered.
- If SI2 and SI3 are "off", and only SI1 is "on", the path 1 is triggered. If the velocity of path 1 is 0, i.e. the motor stops immediately.

| Path number | SI2(ADD0) | SI3(ADD1) | SI4(ADD2) |
|-------------|-----------|-----------|-----------|
| Path 1 | ON | OFF | OFF |
| Path 2 | OFF | ON | OFF |
| Path 3 | ON | ON | OFF |
| Path 4 | OFF | OFF | ON |
| Path 5 | ON | OFF | ON |
| Path 6 | OFF | ON | ON |
| Path 7 | ON | ON | ON |

Note: "on" the input signal optocoupler on, "off" means signal optocoupler off.

5.5.3 Fixed Trigger

Fixed trigger mode is after configuring homing and path (less than 16-segment), then use Pr8.02(0x6002) to start the operation path which replace IO functions of CTRG and HOME. This method is suitable for systems with fixed and simple motion. (Recommended).

Steps as below:

1. Firstly, configure required homing and path, it can be set through controller/PLC software after power-on , or through STEPPERONLINE PC software;
2. Enable drive;
3. Write corresponding command to the 0x6002 to realize the selection and startup of each action.

| Par. # | Register address | Definition | Description |
|--------|------------------|------------------|--|
| Pr8.02 | 0x6002 | Trigger register | <p>Write corresponding command to the 0x6002 to realize the selection and startup of each action.</p> <p>Write value 0x01P----P-segment positioning, (P indicates path #0-15);</p> <p>Write value 0x020---- Homing; (edge triggered)</p> <p>Write value 0x021---- Set the current position as origin by manual;</p> <p>Write value 0x040---- E-stop;</p> <p>Read value 0x000---- Positioning is completed and new data can be received;</p> <p>Read value 0x01P, 0x020, 0x040---- Not responding to the command;</p> <p>Read value 0x10P---- Path is running;</p> <p>Read value 0x200---- Command is completed and waiting for positioning.</p> <p>Note: (P indicates path NO. 0-15);</p> |

5.5.4 Immediate Trigger

The immediate trigger method means that each time the current path 0(PR0) is written, the run of PR0 is triggered in real time. The position & velocity& homing and so on are achieved through one data frame. This method uses PR0 to implement, which has 8 data, the last parameter Pr9.07 is mapped to Pr8.02, writing value 0x10 to it will trigger PR0 motion immediately, thus realizing the immediate trigger operation.

| Order | | Sending message (Master->Slave) | | Return message (Slave->Master) | | |
|-------|---------------|---------------------------------|------|--------------------------------|-------------------------------|------|
| 1 | ID | Slave ID. | 0-31 | ID | Sub-station No. | 0-31 |
| 2 | FC | Function code | 0x10 | FC | Function code | 0x10 |
| 3 | ADDR | Address | 0x62 | ADDR | Address | 0x62 |
| 4 | | | 0x00 | | | 0x00 |
| 5 | NUM1 | Number of Word | 0x00 | NUM | Actually written Number of | 0x00 |
| 6 | | | 0x08 | | | 0x08 |
| 7 | NUM2 | Number of Byte | 0x10 | CRC | check code | Lo |
| 8-9 | Pr9.00 | Mode | XXXX | | | Hi |

| | | | |
|-------|---------------|-----------------|--------|
| 10-11 | Pr9.01 | High position | XXXX |
| 12-13 | Pr9.02 | Low position | XXXX |
| 14-15 | Pr9.03 | velocity | XXXX |
| 16-17 | Pr9.04 | Acceleration | XXXX |
| 18-19 | Pr9.05 | Deceleration | XXXX |
| 20-21 | Pr9.06 | Delay time | XXXX |
| 22-23 | Pr9.07 | Trigger control | 0x0010 |
| 24 | CRC | Check code | Lo |
| 25 | | | Hi |

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

For example:

PR0: velocity 200 rpm, distance 10000 p

Master sending: 07 10 62 00 00 08 10 00 01 00 00 27 10 00 00 27 10 27 10 00 00 00 10 8D 50

Drive return: 07 10 62 00 00 08 DE 11

6 Tuning Operations

There are three kinds of tuning operations for DMRS series:

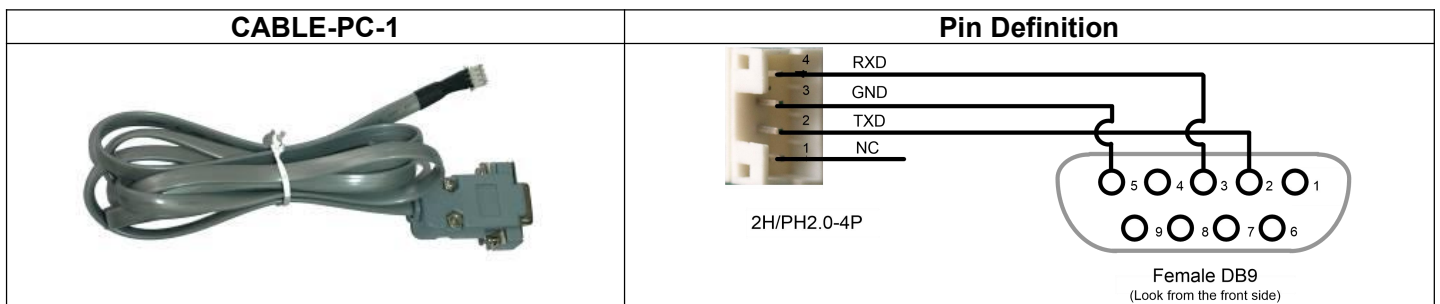
- (1) Through the trial run function of STEPPERONLINE's MotionStudio software,
- (2) Through the PR function of STEPPERONLINE's MotionStudio software,
- (3) Through the general serial port tool software.

6.1 Basic operation of STEPPERONLINE MotionStudio

This uses RS232 communication to control the motor, The STEPPERONLINE MotionStudio software currently available is version 1.3.10, which supports Windows 7 and Windows 10 systems.

6.1.1 Preparation and Steps

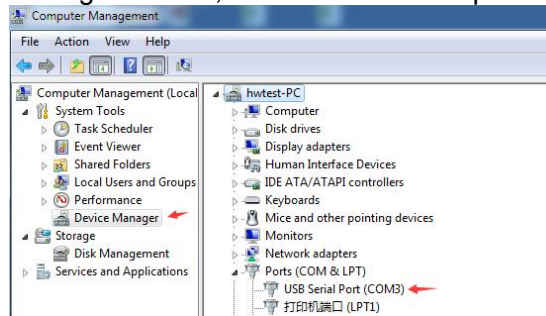
(1) RS232 Tuning cable



It is recommended that users order this cable directly from STEPPERONLINE, not to make it yourself

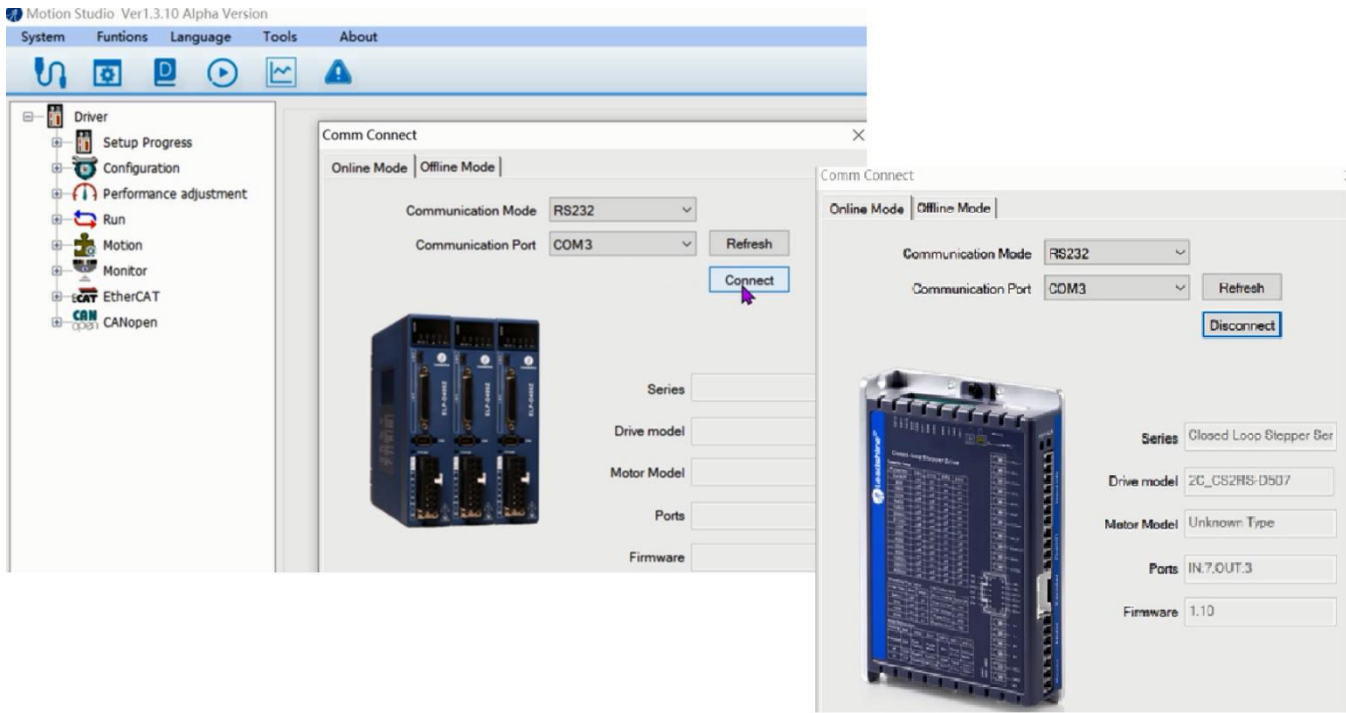
(2) USB to RS232 converter, sometimes it needs to manually install the drive program.

(3) COM port selection, as shown in the figure below, the communication port is COM3:

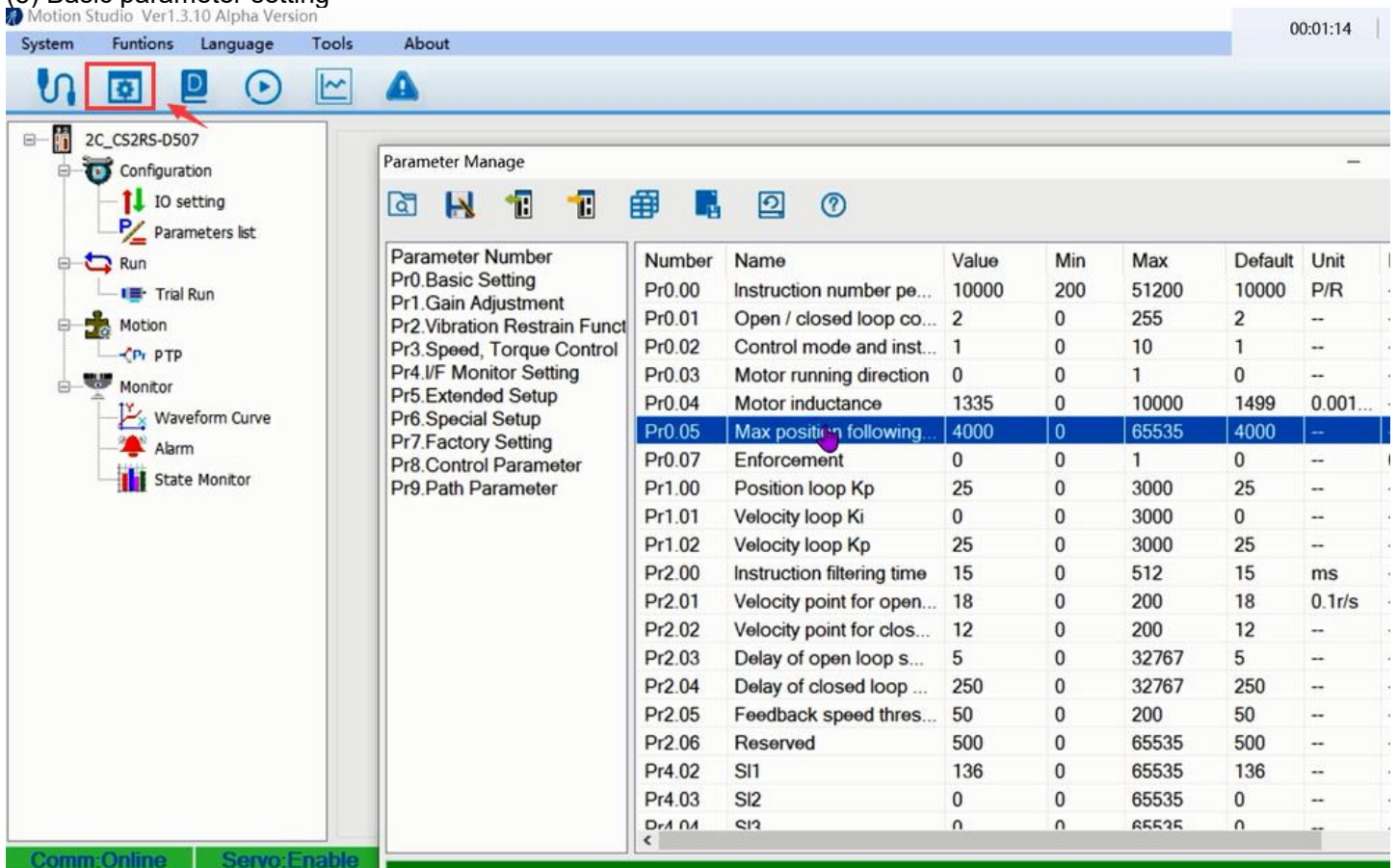


(4) Connect tuning software

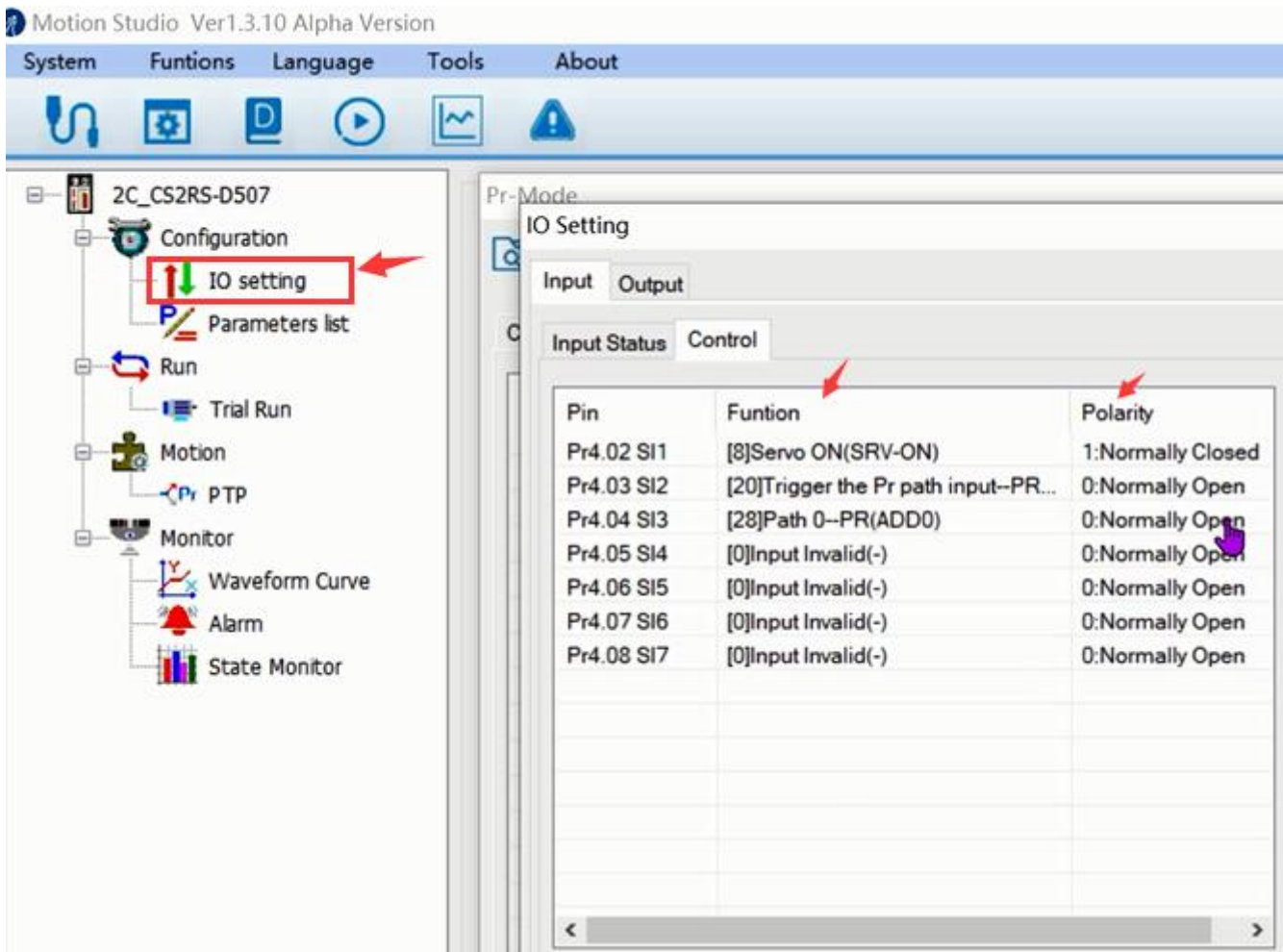
Select COM3, do not select baud rate and device number, keep the default settings. After clicking connect, it will display the driver model. The number of ports and other information, and the lower left corner of the color block turns green, indicating that the connection has been successful.



(5) Basic parameter setting



(6) Input and output function and polarity setting



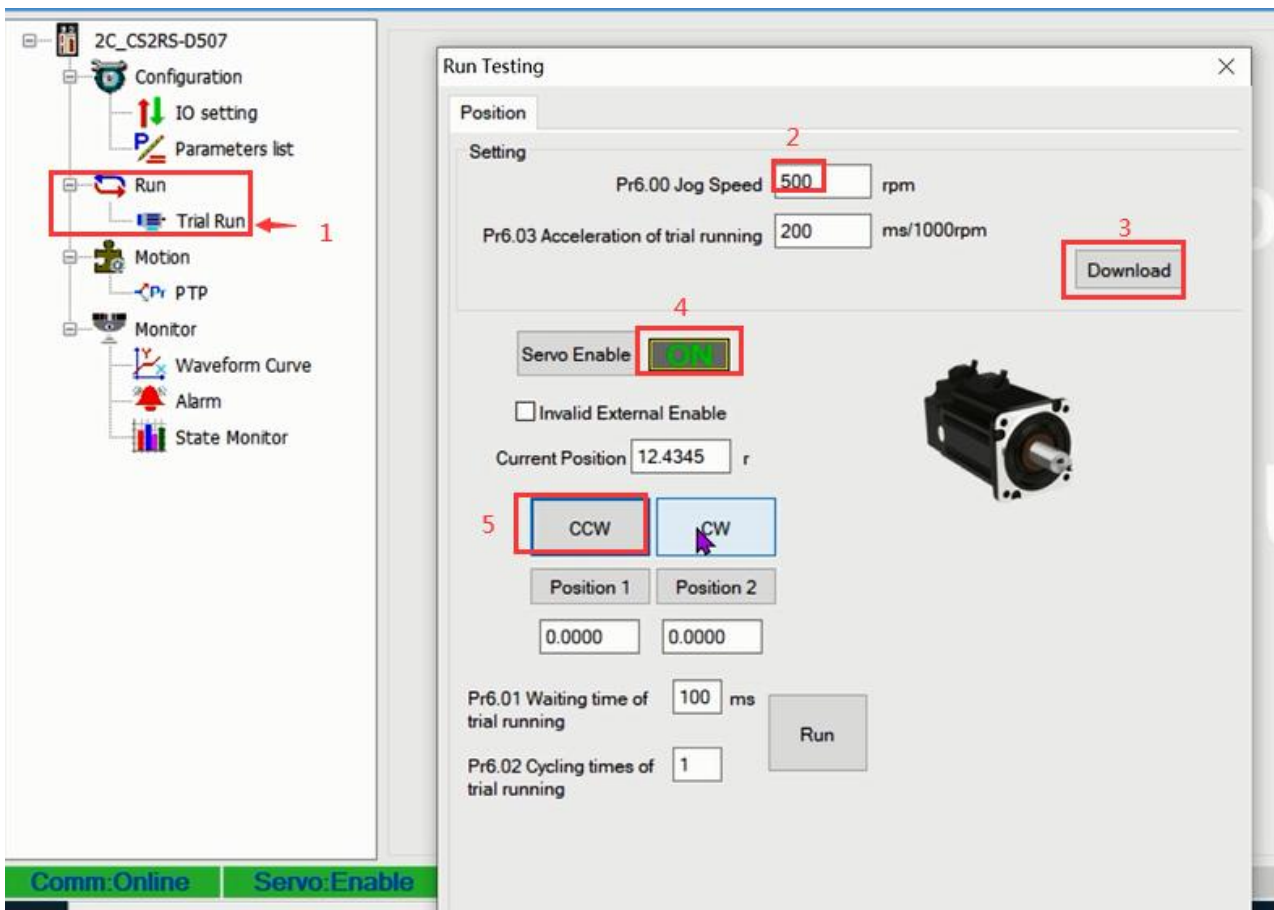
Note: After setting the parameters, click "OK". Then, in the parameter management window, click the Save button to prevent the parameter values from being lost after the drive is powered off.

Parameter Manage

| Parameter Number | Number | Name | Value | Min | Max | Default | Unit |
|------------------------------|--------|---------------------------|-------|--------|-------|---------|------|
| Pr0.Basic Setting | Pr6.15 | Version information (...) | 110 | 0 | 0 | 0 | -- |
| Pr1.Gain Adjustment | Pr6.16 | Version information (s... | 12810 | 0 | 0 | 0 | -- |
| Pr2.Vibration Restrain Funct | Pr7.01 | Encoder resolution | 4000 | 0 | 20000 | 4000 | -- |
| Pr3.Speed, Torque Control | Pr7.02 | Back EMF coefficient | 100 | 0 | 32767 | 100 | 1ms |
| Pr4.I/F Monitor Setting | Pr7.03 | Current loop Kp | 1500 | 0 | 32767 | 1500 | -- |
| Pr5.Extended Setup | Pr7.04 | Current loop Ki | 300 | 0 | 32767 | 300 | -- |
| Pr6.Special Setup | Pr7.05 | Adjustment ratio of cu... | 100 | 0 | 1024 | 100 | -- |
| Pr7.Factory Setting | Pr7.06 | Current loop Kc | 300 | 0 | 32767 | 300 | -- |
| Pr8.Control Parameter | Pr7.07 | Weak magnetic coeffi... | 0 | 0 | 255 | 0 | -- |
| Pr9.Path Parameter | Pr7.08 | Weak magnetic coeffi... | 0 | 0 | 255 | 0 | -- |
| | Pr7.09 | Over voltage limit | 90 | 0 | 1000 | 90 | V |
| | Pr8.00 | PR control setting | 0 | -32768 | 32767 | 0 | -- |

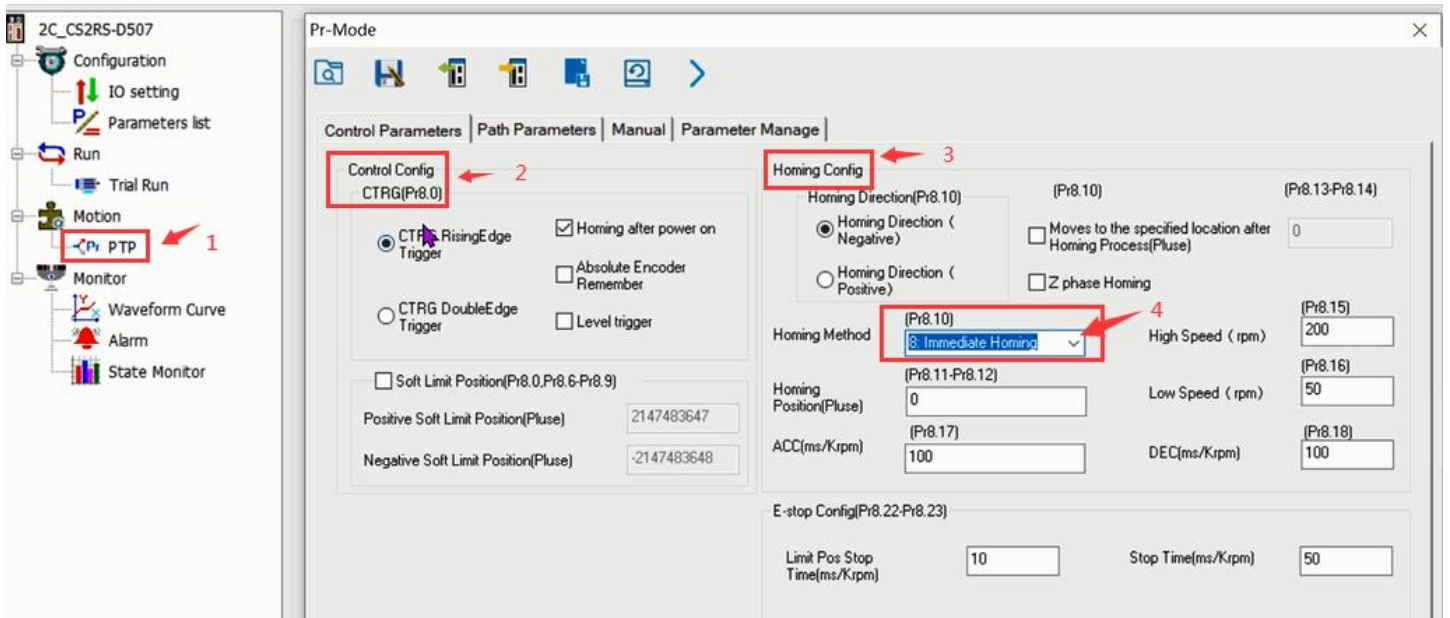
6.1.2 Operation of Trial Run

Trial run lets the motor to achieve forward and reverse rotation, or repeat motion. The operation steps are as follows

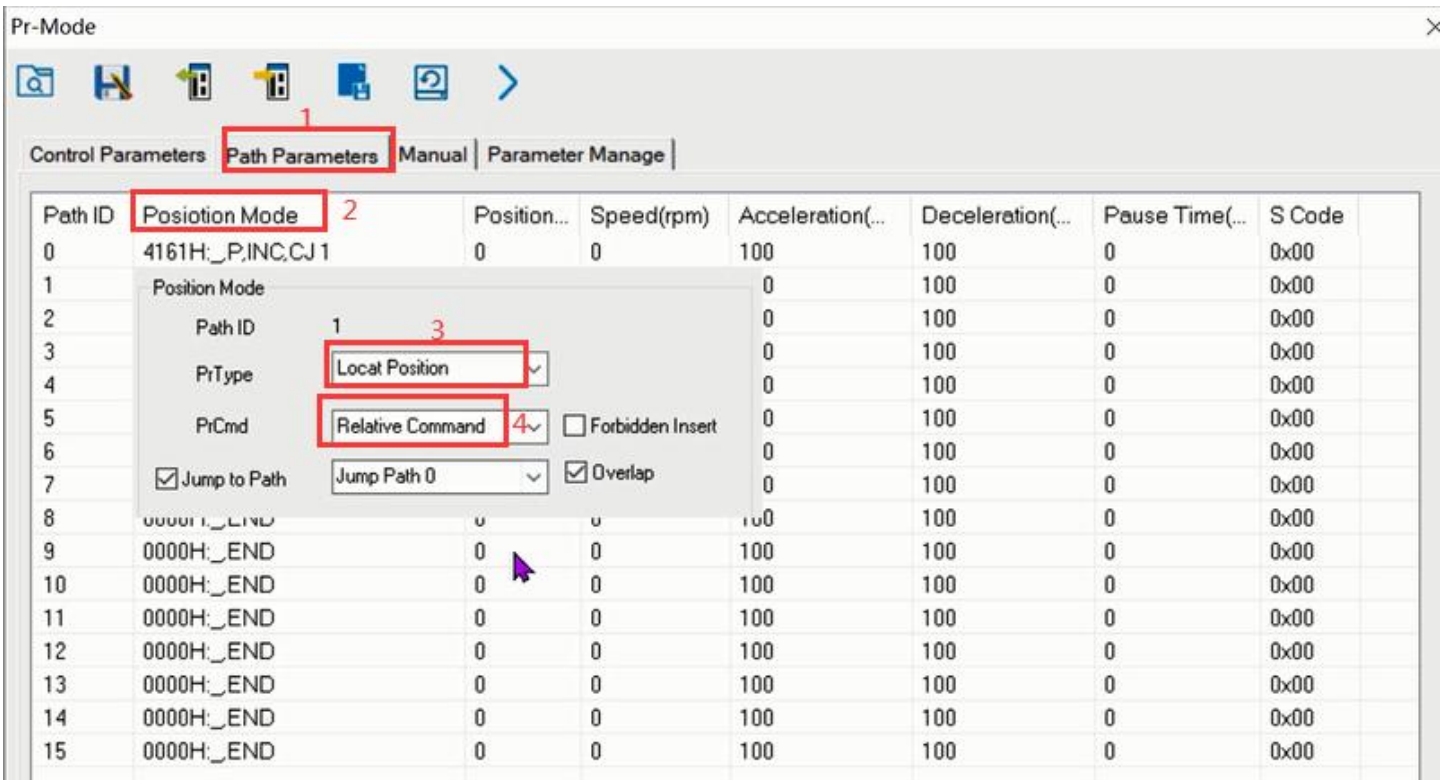


6.1.3 Operation of PR Function

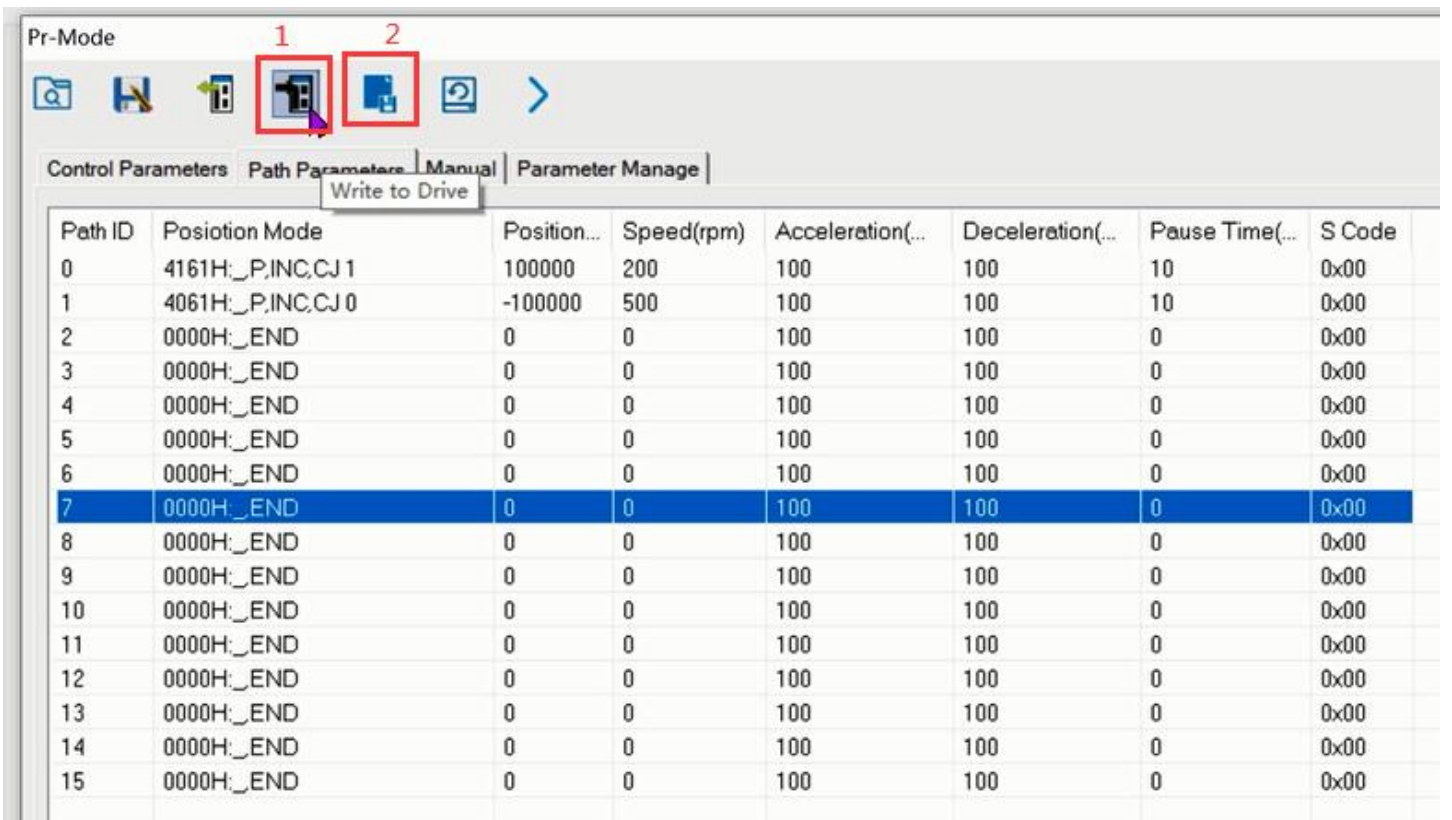
(1) This window can set the CTGR trigger and Homing parameters of PR motion:



(2) This window is the PR path parameter setting, including operation mode, target position, speed value, etc. Double click to modify parameters.

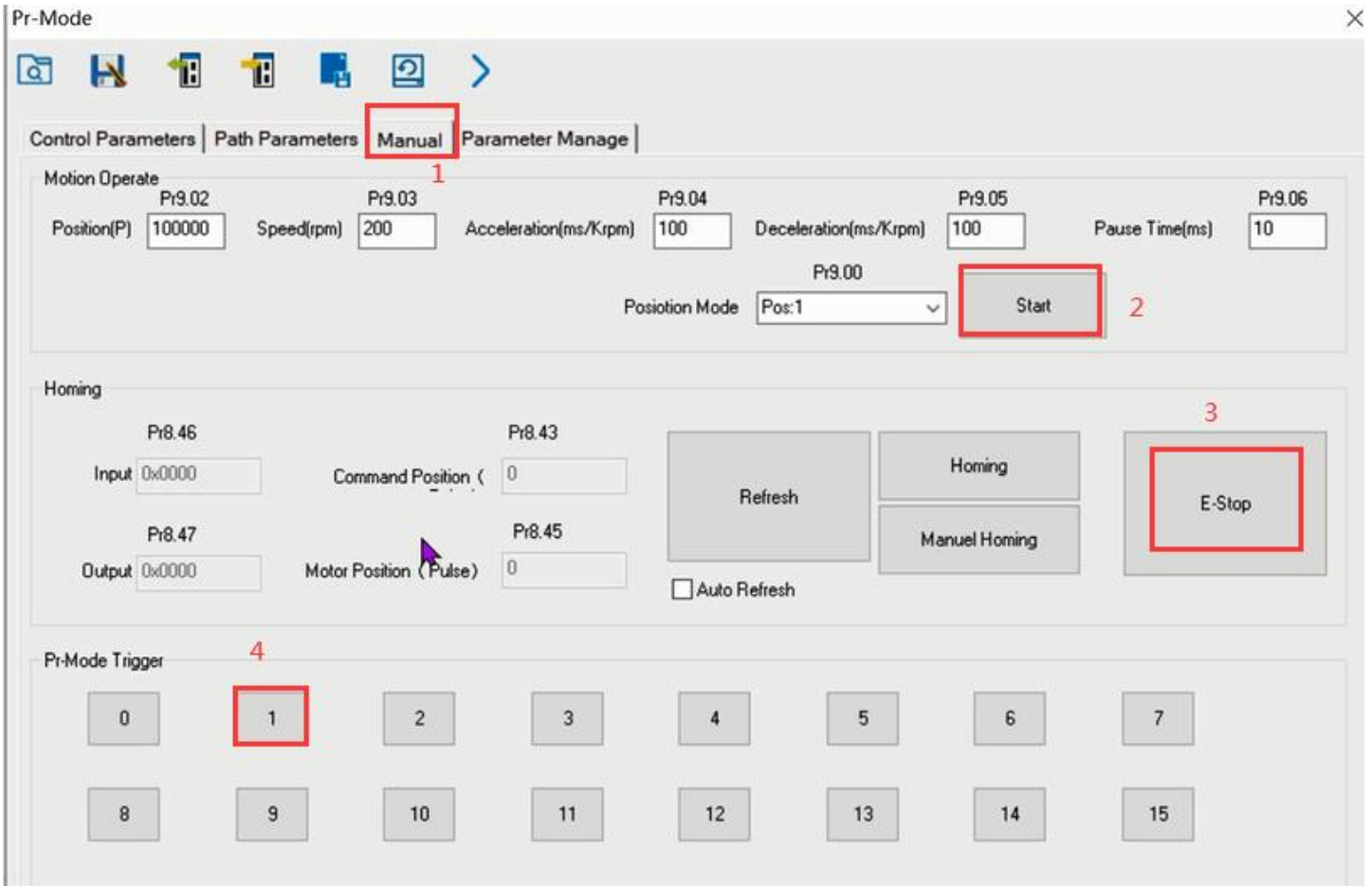


After the setting is completed, please click to download and save, as follows



(3) Manually run the PR path

As shown in the figure below, the default is the motion parameter of PR0. As long as click Start, the motor will run according to the path of PR0. If click PR1 in step 4, the motor will be forced to switch to the path of PR1. Click step 3 to stop motor.

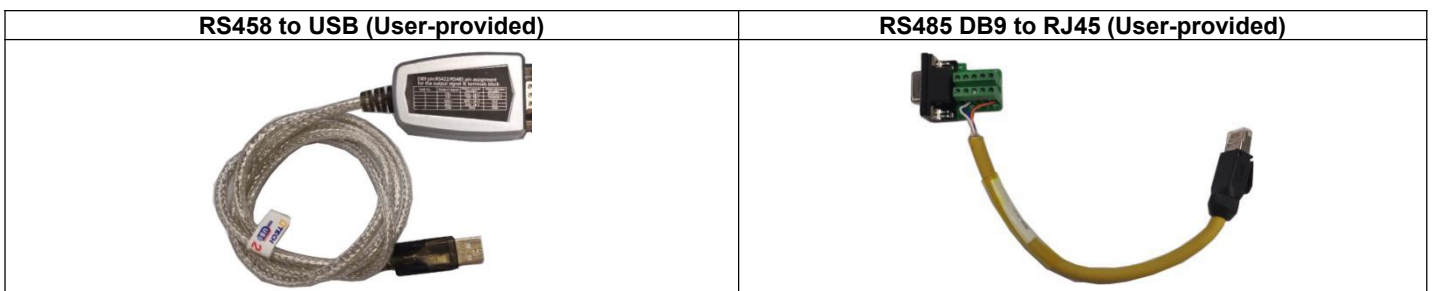


6.2 Basic Operation of Serial Port Tools Software

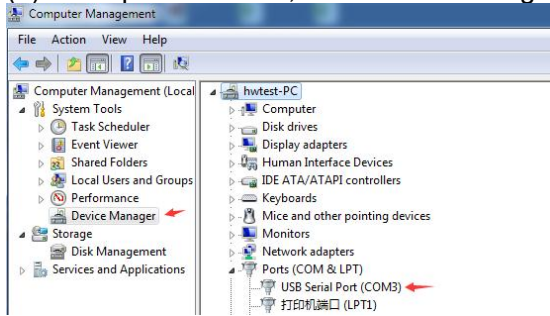
This is to control the motor through RS485 communication, user can realize the movement of the motor by sending commands to the corresponding registers.

6.2.1 Preparation and Steps

(1) RS485 tuning cable



(2) COM port selection, as shown in the figure below, the communication port is COM3:



(4) Connect tuning software

Select COM3, select the same baud rate as the drive settings. After clicking connect.



6.2.2 Operation Instruction Format

Data format: Here is an example of setting the PR0 path: (Data is in hexadecimal)

| Slave ID | Function Code | Register Address | Value to Write | CRC |
|--|--------------------------------|---------------------------------------|--|---|
| 01 | 06 (Preset single register) | 62 00 (PR0 operation mode setting) | 00 41 (Set PR mode to relative position mode) | 56 42 (Automatically generated by the serial port tool software) |
| A complete command 01 06 62 00 00 41 56 42 | | | | |

6.2.3 Command Cases of Modbus RTU

(1) Set PR0 to go absolute position mode, running distance 200000p (microstep 10000)

Commands:

- 01 06 62 00 00 00 01 57 B2 Set PR0 mode to absolute position mode
- 01 06 62 01 00 03 87 B3 Set PR0 position high
- 01 06 62 02 0D 40 32 D2 Set PR0 position low

01 06 62 03 02 58 66 E8 Set PR0 speed value
 01 06 62 04 00 32 56 66 Set PR0 acceleration
 01 06 62 05 00 32 07 A6 Set PR0 deceleration velocity
 01 06 60 02 00 10 37 C6 Trigger PR0 motion
 Send when you need to stop:
 01 06 60 02 00 40 37 FA Emergency stop

(2) Set PR0 to go relative position mode, running distance 10000p (microstep 10000)

Commands:
 01 06 62 00 00 41 56 42 Set PR mode to relative position mode
 01 06 62 01 00 00 C7 B2 Set PR0 position high
 01 06 62 02 27 10 2D 8E Set PR0 position low
 01 06 62 03 02 58 66 E8 Set PR0 velocity
 01 06 62 04 00 32 56 66 Set PR0 acceleration
 01 06 62 05 00 32 07 A6 Set PR0 deceleration velocity
 01 06 60 02 00 10 37 C6 Trigger PR0 motion
 Send when you need to stop:
 01 06 60 02 00 40 37 FA Emergency stop

(3) Set PR0 to go in velocity mode, velocity is 300rpm

Commands:
 01 06 62 00 00 00 02 17 B3 Set PR0 to velocity mode
 01 06 62 03 01 2C 66 3F Set PR0 speed value
 01 06 60 02 00 10 37 C6 Trigger PR0 motion
 Send when you need to stop:
 01 06 60 02 00 40 37 FA Emergency stop

(4) Set PR1 to go absolute position mode, running distance -200000p (microstep is 10000)

Commands:
 01 06 62 08 00 01 D6 70 Set PR1 mode
 01 06 62 09 FF FC 07 C1 Set PR1 position high
 01 06 62 0A F2 C0 F3 40 Set PR1 position low
 01 06 62 0B 02 58 E7 2A Set PR1 speed value
 01 06 62 0C 00 32 D7 A4 Set PR1 acceleration
 01 06 62 0D 00 32 86 64 Set PR1 deceleration
 01 06 60 02 00 11 F6 06 Trigger PR1 motion
 Send when you need to stop:
 01 06 60 02 00 40 37 FA Emergency stop

(Note: -200000 is converted to hexadecimal as FFFCF2C0. You can enter the minus sign first, then 200000, and then enter the minus sign again to display FFFCF2C0)

(5) Set PR1 go velocity mode, speed value is 200rpm (microstep is 10000)

Commands:
 01 06 62 08 00 02 96 71 Sets PR1 to velocity mode
 01 06 62 0B 01 2C E7 FD Set PR1 speed value
 01 06 60 02 00 11 F6 06 Trigger PR1 motion
 Send when you need to stop:
 01 06 60 02 00 40 37 FA Emergency stop

(6) Homing

Commands:
 01 06 60 0A 00 00 B7 C8 Set homing method
 01 06 60 0F 00 64 A6 22 Set homing speed high
 01 06 60 10 00 1E 16 07 Set homing speed low
 01 06 60 02 00 20 37 D2 Trigger Homing

Appendix A Parameters List

1. Modbus RTU Parameters

The STEPPERONLINE RS485 parameter data type is 32-bit data, and a parameter contains two registers, high 16 bits and low 16 bits, but in practice most parameters only need to use the low 16 bits. When reading and writing multiple parameters in succession, the high 16 bits of the parameter need to be used as the start.

| Register Address | Software Parameter No. | Name | Description | Range | Default Value | Unit |
|------------------|------------------------|---|--|---|---------------|---------|
| 0x0001 | Pr0.00 | Pulse/revolution | 10000 | 200-51200 | 10000 | P/R |
| 0x0005 | Pr0.02 | Control mode source | Invalid | 0-10 | 1 | -- |
| 0x0007 | Pr0.03 | Motor direction | 0: CW 1: CCW | 0-1 | 0 | -- |
| 0x0009 | Pr0.04 | Motor inductance | Invalid | 0-10000 | 1499 | 0.001mH |
| 0x000F | Pr0.07 | Forced enable by software | Software forced enable has a higher priority than IO enable, and when this value is 0, the enable status of the drive only depends on the IO signal. When this value is 1, the motor is enabled regardless of the IO signal status. | 0-1 | 0 | -- |
| 0x00A1 | Pr2.00 | Command filter time | To configure the time for internal command filtering | 0-512 | 15 | 0.1ms |
| 0x0145 | Pr4.02 | DI1(input 1) | Default is normal-open(N.O) type, it can be set to normal-closed(N.C) type by setting the corresponding port +0x80 DI1 is enable(default), N.C type input. 0: invalid; 7: alarm clearing; 8: enable (also can be set by 0x00F); 0x20: Trigger command; 0x21: Trigger homing; 0x22: EMG (quick stop); 0x23: JOG+; 0x24: JOG-; 0x25: POT (positive limit); 0x26: NOT (negative limit); 0x27: ORG (home switch); 0x28: ADD0 (path address 0); 0x29: ADD1 (path address 1); 0x2A: ADD2 (path address 2); 0x2B: ADD3 (path address 3); 0x2C: JOG velocity 2 | 0-65535 | 136 (0x88) | -- |
| 0x0147 | Pr4.03 | DI2(input 2) | | 0-65535 | 0 | -- |
| 0x0149 | Pr4.04 | DI3(input 3) | | 0-65535 | 0 | -- |
| 0x014B | Pr4.05 | DI4(input 4) | | 0-65535 | 0 | -- |
| 0x014D | Pr4.06 | DI5(input 5) | | 0-65535 | 0 | -- |
| 0x014F | Pr4.07 | DI6(input 6) | | 0-65535 | 0 | -- |
| 0x0151 | Pr4.08 | DI7(input 7) | | 0-65535 | 0 | -- |
| 0x0157 | Pr4.11 | DO1(output 1) | | Default is normal-open(N.O) type, it can be set to normal-closed(N.C) type by setting the corresponding port +0x80 0: invalid; 0x20: command completed; 0x21: path completed; 0x22: homing completed; 0x23: in-position completed; 0x24: brake output; 0x25: alarm output; | 0-65535 | 0 |
| 0x0159 | Pr4.12 | DO2(output 2) | 0-65535 | | 0 | -- |
| 0x015B | Pr4.13 | DO3(output 3) | 0-65535 | | 0 | -- |
| 0x0167 | Pr4.19 | Delay of brake released | Keep default normally | 0-1500 | 250 | ms |
| 0x0169 | Pr4.20 | Delay of brake locked | Keep default normally | 0-1500 | 250 | ms |
| 0x016B | Pr4.21 | Threshold value of brake locking velocity | Keep default normally | 0-500 | 10 | -- |
| 0x016D | Pr4.22 | Alarm detection selection | Bit setting: =1: Yes; =0: No bit0: over-current (Cannot be changed) bit1: over-voltage bit3: ADC sampling failure bit4: Locked shaft alarm bit5: EEPROM alarm bit6: Auto-tuning alarm | 0-65535 | 0 | -- |
| 0x0177 | Pr4.27 | Bus voltage | | 0-65535 | 0 | 0.1V |
| 0x0179 | Pr4.28 | Digital input statue | Bit0-Bit6: DI1-DI7 | 0-65535 | 0 | -- |
| 0x017B | Pr4.29 | Digital output statue | Bit0-Bit2: DO1-DO3 | 0-65535 | 0 | -- |
| 0x0187 | Pr4.35 | DIP switches statue | | 0-65535 | 0 | -- |
| 0x0191 | Pr5.00 | Peak current | Approximately equal to motor phase current value multiplied by 1.4 | 522: 0-22 556: 0-56 870: 0-70 882: 0-82 | 10 | 0.1A |

| 0x0197 | Pr5.03 | Percentage of shaft locked current(power on) | Keep default normally | 0-100 | 100 | -- | | | | | | | | | | | | | | | | | | |
|-------------|--|--|--|-------------|---------------------|--------|---------------------|--------|---------------------|--------|-------------------------------|--------|---|--------|-------------------------------------|--------|-------------------------------|--------|---|--------|--|--|--|----|
| 0x0199 | Pr5.04 | Shaft locked duration | Keep default normally | 0-1500 | 200 | 1ms | | | | | | | | | | | | | | | | | | |
| 0x019F | Pr5.07 | Rising time of shaft locked current (power on) | Keep default normally | 1-60 | 1 | 100ms | | | | | | | | | | | | | | | | | | |
| 0x01A5 | Pr5.10 | The max stop time | Keep default normally | 100-1000 | 1000 | ms | | | | | | | | | | | | | | | | | | |
| 0x01AB | Pr5.13 | Auto-tuning at power on | 0: disable, 1: enable | 0-1 | 1 | -- | | | | | | | | | | | | | | | | | | |
| 0x01BD | Pr5.22 | RS485 baud rate | 0: 2400 1: 4800 2: 9600 (default) 3: 19200 4: 38400(factory) 5: 57600 6: 115200 When SW6 and SW7 are all OFF, it can be set by PC software. | 0-6 | 4 | -- | | | | | | | | | | | | | | | | | | |
| 0x01BF | Pr5.23 | RS485 ID | When SW1 - SW5 are all ON, it can be set by PC software | 0-127 | 1 | -- | | | | | | | | | | | | | | | | | | |
| 0x01C1 | Pr5.24 | RS485 data type selection | 0: 8-bit data, even check, 2 stop bits; 1: 8-bit data, odd check, 2 stop bits 2: 8-bit data, even check, 1 stop bit; 3: 8-bit data, odd check, 1 stop bit: 4: 8-bit data, no check, 1 stop bit: 5: 8-bit data, no check, 2 stop bits; | 0-11 | 4 | -- | | | | | | | | | | | | | | | | | | |
| 0x01C3 | Pr5.25 | RS485 control word | | 0-32767 | 0 | -- | | | | | | | | | | | | | | | | | | |
| 0x01C4 | Pr5.26 | Communication bit delay | | 0-100 | 35 | bit | | | | | | | | | | | | | | | | | | |
| 0x01D1 | Pr5.32 | Switching time to standby | | 10-65535 | 200 | ms | | | | | | | | | | | | | | | | | | |
| 0x01D3 | Pr5.33 | Standby current percentage | | 0-100 | 50 | | | | | | | | | | | | | | | | | | | |
| 0x01E1 | Pr6.00 | JOG velocity | This JOG is triggered by RS485. For JOG triggered by IO, please use Pr8.40/8.41 | 0-5000 | 60 | r/min | | | | | | | | | | | | | | | | | | |
| 0x01E3 | Pr6.01 | Interval | | 0-10000 | 100 | ms | | | | | | | | | | | | | | | | | | |
| 0x01E5 | Pr6.02 | Running times | | 0-30000 | 1 | -- | | | | | | | | | | | | | | | | | | |
| 0x01E7 | Pr6.03 | Acc. / Dec. time | | 0-10000 | 200 | -- | | | | | | | | | | | | | | | | | | |
| 0x01FF | Pr6.15 | Version information | Read only | 0-65535 | 0 | -- | | | | | | | | | | | | | | | | | | |
| 0x0201 | Pr6.16 | Firmware information | Read only | 0-65535 | 0 | -- | | | | | | | | | | | | | | | | | | |
| 0x0231 | Pr7.00 | Motor model | Invalid | 0-100 | 0 | -- | | | | | | | | | | | | | | | | | | |
| 0x0235 | Pr7.02 | Back EMF coefficient | Invalid | 0-32767 | 100 | 1ms | | | | | | | | | | | | | | | | | | |
| 0x0237 | Pr7.03 | Current loop proportional KP | Invalid | 0-3000 | 1500 | -- | | | | | | | | | | | | | | | | | | |
| 0x0239 | Pr7.04 | Current loop KI | It is recommended to keep the factory value, when disable auto-tuning, can be modified | 0-1500 | 300 | -- | | | | | | | | | | | | | | | | | | |
| 0x023B | Pr7.05 | Current loop KP | | 0-1024 | 100 | -- | | | | | | | | | | | | | | | | | | |
| 0x023D | Pr7.06 | Current loop Kc | Invalid | 0-32767 | 300 | -- | | | | | | | | | | | | | | | | | | |
| 0x0243 | Pr7.09 | Over-voltage threshold | Invalid | 0-1000 | 90 | V | | | | | | | | | | | | | | | | | | |
| 0x1003 | - | Motion status | Read only: <table border="1"> <thead> <tr> <th>Bit NO.</th> <th>Read value =1 means</th> </tr> </thead> <tbody> <tr> <td>Bit0</td> <td>Fault</td> </tr> <tr> <td>Bit1</td> <td>Enable</td> </tr> <tr> <td>Bit2</td> <td>Running</td> </tr> <tr> <td>Bit4</td> <td>Command completed</td> </tr> <tr> <td>Bit5</td> <td>Path completed</td> </tr> <tr> <td>Bit6</td> <td>Homing completed</td> </tr> </tbody> </table> | Bit NO. | Read value =1 means | Bit0 | Fault | Bit1 | Enable | Bit2 | Running | Bit4 | Command completed | Bit5 | Path completed | Bit6 | Homing completed | | | -- | | | | |
| Bit NO. | Read value =1 means | | | | | | | | | | | | | | | | | | | | | | | |
| Bit0 | Fault | | | | | | | | | | | | | | | | | | | | | | | |
| Bit1 | Enable | | | | | | | | | | | | | | | | | | | | | | | |
| Bit2 | Running | | | | | | | | | | | | | | | | | | | | | | | |
| Bit4 | Command completed | | | | | | | | | | | | | | | | | | | | | | | |
| Bit5 | Path completed | | | | | | | | | | | | | | | | | | | | | | | |
| Bit6 | Homing completed | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1801 | - | Control word | <table border="1"> <thead> <tr> <th>Write value</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0x1111</td> <td>Reset current alarm</td> </tr> <tr> <td>0x1122</td> <td>Reset history alarm</td> </tr> <tr> <td>0x2211</td> <td>Save all parameters to EEPROM</td> </tr> <tr> <td>0x2222</td> <td>Parameter reset(exclude motor parameters)</td> </tr> <tr> <td>0x2233</td> <td>All parameters are reset to factory</td> </tr> <tr> <td>0x2244</td> <td>Save all mappings into EEPROM</td> </tr> <tr> <td>0x4001</td> <td>JOG CW (Need to write once at least 50ms)</td> </tr> <tr> <td>0x4002</td> <td>JOG CCW (Need to write once at least 50ms)</td> </tr> </tbody> </table> | Write value | Function | 0x1111 | Reset current alarm | 0x1122 | Reset history alarm | 0x2211 | Save all parameters to EEPROM | 0x2222 | Parameter reset(exclude motor parameters) | 0x2233 | All parameters are reset to factory | 0x2244 | Save all mappings into EEPROM | 0x4001 | JOG CW (Need to write once at least 50ms) | 0x4002 | JOG CCW (Need to write once at least 50ms) | | | -- |
| Write value | Function | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1111 | Reset current alarm | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1122 | Reset history alarm | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2211 | Save all parameters to EEPROM | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2222 | Parameter reset(exclude motor parameters) | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2233 | All parameters are reset to factory | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2244 | Save all mappings into EEPROM | | | | | | | | | | | | | | | | | | | | | | | |
| 0x4001 | JOG CW (Need to write once at least 50ms) | | | | | | | | | | | | | | | | | | | | | | | |
| 0x4002 | JOG CCW (Need to write once at least 50ms) | | | | | | | | | | | | | | | | | | | | | | | |
| 0x1901 | - | Save parameter status word | Read only <table border="1"> <thead> <tr> <th>Read Value</th> <th>Means</th> </tr> </thead> <tbody> <tr> <td>0x5555</td> <td>Save successfully</td> </tr> <tr> <td>0xAAAA</td> <td>Failed to save</td> </tr> </tbody> </table> | Read Value | Means | 0x5555 | Save successfully | 0xAAAA | Failed to save | | | -- | | | | | | | | | | | | |
| Read Value | Means | | | | | | | | | | | | | | | | | | | | | | | |
| 0x5555 | Save successfully | | | | | | | | | | | | | | | | | | | | | | | |
| 0xAAAA | Failed to save | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | |
|--------|-------------------|---------------|------------|------------------------|---|---|----|
| 0x2203 | - | Current alarm | Read only | | - | - | -- |
| | | | Error code | Means | | | |
| | | | 0x01 | Over- current | | | |
| | | | 0x02 | Over- voltage | | | |
| | | | 0x40 | Current sampling fault | | | |
| | | | 0x80 | Failed to lock shaft | | | |
| | | | 0x200 | EEPROM fault | | | |
| 0x100 | Auto-tuning fault | | | | | | |

2. PR Parameters:

PR parameter data type is 16-bit data, one parameter occupies one register (one 16-bit register = 2 8-bit bytes).

| Parameter NO. in PC software | Register Address | Name | Description |
|------------------------------|------------------|---------------------------------------|--|
| Pr8.00 | 0x6000 | PR global control function | Bit0: CTRG effective edge. 0--rising edge, 1--double edges Bit1: Soft limit.. 0--disable, 1--enable Bit2: Homing after power on. 0--no, 1--yes Bit4: CTRG trigger type. 0--refer to bit0, 1--level trigger |
| Pr8.02 | 0x6002 | PR control register | (1) Write commands to 0x6002 to select and trigger each action: Write 0x1P ("P" = 0-15), run the path P motion; Write 0x20, homing; Write 0x21, manually set to zero position; Write 0x40, quick stop; (2) Read 0x6002 can check the running path NO. and status: Return "0x000P", path P motion complete, can receive new motion command; Return "0x001P or 0020 or 0040", not yet responding to commands; Return 0x010P, path P motion is running; Return 0x0200, command completed and waiting for positioning; Return 0x020P, a position following error during path P; |
| Pr8.06 | 0x6006 | Limit+ H | Soft limit positive high bits |
| Pr8.07 | 0x6007 | Limit+ L | Soft limit positive low bits |
| Pr8.08 | 0x6008 | Limit- H | Soft limit negative high bits |
| Pr8.09 | 0x6009 | Limit- L | Soft limit negative low bits |
| Pr8.10 | 0x600A | Homing mode | Bit0: homing direction, 0--CW, 1--CCW; Bit1: Whether to move to the specified position after homing, 0--no, 1--yes; Bit2: homing method, 0--limit switch homing, 1--home switch homing, Note: (1) Write 0x21 to 0x6002, manually set the current point to zero position) (2) Other methods are not valid |
| Pr8.15 | 0x600F | Homing high velocity | The 1 st segment velocity of homing, unit: rpm |
| Pr8.16 | 0x6010 | Homing low velocity | The 2 nd segment velocity of homing, unit: rpm |
| Pr8.17 | 0x6011 | Homing Acc | Acc of homing, unit: ms/1000rpm |
| Pr8.18 | 0x6012 | Homing Dec | Dec of homing, unit: ms/1000rpm |
| Pr8.19 | 0x6015 | Homing over-travel | If setting a value, the motor stops when reaches this set distance even if homing is not completed. |
| Pr8.22 | 0x6016 | Limit switch Dec. time | Deceleration time after reaching limit switch, unit: ms |
| Pr8.23 | 0x6017 | STOP time | Deceleration time after quick stop, unit: ms |
| Pr8.26 | 0x601A | IO combination trigger mode selection | 0: Disable IO combination triggering (factory) 1: Enable IO combination triggering, activated after homing is completed 2: Enable IO combination triggering, nothing to do with homing (recommended) |
| Pr8.27 | 0x601B | IO combination filter time | Unit: ms |
| Pr8.28 | 0x601C | S code current output value | -- |
| Pr8.29 | 0x601D | PR alarm | 0: Automatic zeroing of new commands 0x100: Limit switch error during homing 0x102: Homing over-travel error 0x20P: Limit switch error in Path P (P=1-15) 0x300: Touch limit switch during JOG movement |
| Pr8.38 | 0x6026 | JOG velocity 2 | JOG triggered by external I/O signal, Unit: rpm |
| Pr8.39 | 0x6027 | JOG velocity | Unit: rpm |

| | | | |
|--------|--------|-----------------------------------|--|
| Pr8.40 | 0x6028 | JOG Acceleration | Unit: ms/1000rpm |
| Pr8.41 | 0x6029 | JOG Deceleration | Unit: ms/1000rpm |
| Pr8.42 | 0x602A | Profile position H | Read only. High 16-bit (0-65535), it will be cleared after successful homing |
| Pr8.43 | 0x602B | Profile position L | Read only. Low 16-bit (0-65535), it will be cleared after successful homing |
| Pr8.44 | 0x602C | Actual position H | Read only. High 16-bit (0-65535), it will be cleared after successful homing |
| Pr8.45 | 0x602D | Actual position H | Read only. Low 16-bit (0-65535), it will be cleared after successful homing |
| Pr8.46 | 0x602E | Digital Input | |
| Pr8.48 | 0x6030 | S-code output setting for Path 0 | |
| Pr8.49 | 0x6031 | S-code output setting for Path 1 | |
| Pr8.50 | 0x6032 | S-code output setting for Path 2 | |
| Pr8.51 | 0x6033 | S-code output setting for Path 3 | |
| Pr8.52 | 0x6034 | S-code output setting for Path 4 | |
| Pr8.53 | 0x6035 | S-code output setting for Path 5 | |
| Pr8.54 | 0x6036 | S-code output setting for Path 6 | |
| Pr8.55 | 0x6037 | S-code output setting for Path 7 | |
| Pr8.56 | 0x6038 | S-code output setting for Path 8 | |
| Pr8.57 | 0x6039 | S-code output setting for Path 9 | |
| Pr8.58 | 0x603A | S-code output setting for Path 10 | |
| Pr8.59 | 0x603B | S-code output setting for Path 11 | |
| Pr8.60 | 0x603C | S-code output setting for Path 12 | |
| Pr8.61 | 0x603D | S-code output setting for Path 13 | |
| Pr8.62 | 0x603E | S-code output setting for Path 14 | |
| Pr8.63 | 0x603F | S-code output setting for Path 15 | |
| Pr9.00 | 0x6200 | Motion of Path 0 | <p>The corresponding functions can be selected for different bit</p> <p>Bit0-3: TYPE, =0---- no action =1---- position positioning =2---- velocity movement =3---- homing;</p> <p>Bit4: INS, =0---- No interrupt =1---- interrupt(all the current ones are 1.);</p> <p>Bit5: OVLP, =0---- Non overlapping =1---- Overlapping</p> <p>Bit6: =0----absolute position =1----relative Commands</p> <p>Bit8-13: Jump to the corresponding PR path 0-15; bit14: JUMP, =0---- No jump =1---- jump</p> |
| Pr9.01 | 0x6201 | Position H | High 16 bit, |
| Pr9.02 | 0x6202 | Position L | Low 16 bit |
| Pr9.03 | 0x6203 | velocity | Unit: rpm |
| Pr9.04 | 0x6204 | Acc | Unit: ms/1000rpm |
| Pr9.05 | 0x6205 | Dec | Unit: ms/1000rpm |
| Pr9.06 | 0x6206 | Pause time | Pause time after the command is stopped |
| Pr9.07 | 0x6207 | Special parameter | PR Path 0 maps directly to Pr8.02, Others are reserved |
| Pr9.08 | 0x6208 | Motion of Path 1 | --- |
| Pr9.09 | 0x6209 | Position H | --- |
| Pr9.10 | 0x620A | Position L | --- |
| Pr9.11 | 0x620B | velocity | --- |

| | | | |
|----------------|------------------------|------------------------|----------------------|
| Pr9.12 | 0x620C | Acc | --- |
| Pr9.13 | 0x620D | Dec | --- |
| Pr9.14 | 0x620E | Pause time | --- |
| Pr9.15 | 0x620F | Special parameter | --- |
| Pr9.16 | 0x6210 | Motion of Path 2 | --- |
| Pr9.17 | 0x6211 | Position H | --- |
| Pr9.18 | 0x6212 | Position L | --- |
| Pr9.19 | 0x6213 | velocity | --- |
| Pr9.20 | 0x6214 | Acc | --- |
| Pr9.21 | 0x6215 | Dec | --- |
| Pr9.22 | 0x6216 | Pause time | --- |
| Pr9.23 | 0x6217 | Special parameter | --- |
| Pr9.24- Pr9.31 | Similar as above paths | Similar as above paths | Each path has 8 data |
| Pr9.32- Pr9.39 | Similar as above paths | Similar as above paths | Each path has 8 data |
| Pr9.40- Pr9.47 | Similar as above paths | Similar as above paths | Each path has 8 data |
| Pr9.48- Pr9.55 | Similar as above paths | Similar as above paths | Each path has 8 data |
| Pr9.56- Pr9.63 | Similar as above paths | Similar as above paths | Each path has 8 data |
| Pr9.64- Pr9.71 | Similar as above paths | Similar as above paths | Each path has 8 data |

Zero Position: a fixed position on the machine can correspond to a definite digital input signal, or to a Z signal

Zero Point of Machine: mechanical absolute zero position

Home offset: difference between zero position and zero point of machine, the value of Object 607Ch (default = 0), Zero position= zero point + home offset

Home Switch: homing switch input signal

Negative Limit: negative limit switch input signal

Positive Limit: positive limit switch input signal