

AG Gripper

Short Manual (Modbus-RTU)

Content

Content	1
Revisions.....	2
1 Specifications.....	3
1.1 Indicator light definition	3
1.2 Pinout Description.....	4
2 Modbus-RTU Control	4
2.1 Debugging software description	5
2.1.1 Installation and wiring of debugging software.....	5
2.1.2 Debugging software instructions.....	6
2.2 Modbus-RTU default allocation.....	9
2.3 Command description	10
2.3.1 RTU Framing	10
2.3.2 Register Mapping.....	10
2.3.3 Register Description.....	12
2.3.3.1 Initialization	12
2.3.3.2 Force.....	13
2.3.3.3 Position.....	13
2.3.3.4 Initialization State	14
2.3.3.5 Gripper State	14
2.3.3.6 Current Position	15
2.3.3.7 Save Parameter.....	15
2.3.3.8 Initialization Direction	16
2.3.3.9 Slave Address	16
2.3.3.10 Baud Rate.....	17
2.3.3.11 Stop Bits	17
2.3.3.12 Parity	18
2.3.3.13 Test I/O Parameters	18
2.3.3.14 I/O Mode Switch	19
2.3.3.15 I/O Parameter Configuration.....	20
3 I/O Control.....	21
3.1 I/O Configuration.....	21
3.2 IO use.....	23
4 Maintain	24
4.1 Daily cleaning	24
4.2 Fingertip replacement	25
4.3 Periodic inspection and nursing	26

Revisions

Date	Version	Revised content
20200426	V1.0	First edition, write wiring instructions and command instructions
20200904	V2.0	Change some instructions , Update the description of IO mode
20210401	V2.2	<ol style="list-style-type: none">1. Added functions: Add drop feedback in IO mode, add 04 function code, and further adapt to robots such as AUBO and JICA.2. Add product list, add hardware connection description example, add 04 and 10 function code description;3. Add robot connection part;4. Add communication protocol converter (optional) description;5. Add debugging software description;6. Modification of IO part: modify the description of the IO settings part of the document, add IO type hardware configuration examples; VI. Modify the layout to make the layout beautiful.
20210727	V2.3	Modify the high resistance description as disconnection, batch update, line sequence modification
20211008	V2.4	Delete the data that has nothing to do with the operation part, and re-typesetting

1 Specifications

AG series are adaptive electric gripper, The number(AG-number) represents the maximum gripping stroke of the gripper. The gripper is equipped with a pair of parallel fingertips, which runs symmetrically during the movement. The main structure of the gripper is a smooth rectangular structure. It is equipped with an 8-core communication interface, as shown in Figure 1.1. It has the following characteristics:

Controllable force/position: The gripper can program and adjust the grip position and grip force. In the process of gripper movement, the running speed is related to the clamping force. The greater the clamping force is, the faster the running speed is.

Multiple communication modes: The gripper supports Modbus RTU protocol and IO mode control. Other communication protocols such as USB and ETHERNET can be transferred through protocol converter.

Gripping detection: The combination of force control and position control is adopted in the gripping process.

Gripping feedback: The state of the gripper can be read by programming, and can also be judged according to the indicator of the gripper.

Fingertips can be customized: Fingertips can be replaced according to situation, which is suitable for precision machining, parts assembly, and other fields.

1.1 Indicator light definition

The gripper can feed back the state of the gripper in real time. In addition to the command reading, it can also be judged on the color of the indic.

Color description of indicator

- **Uninitialized state:** Red light blinks, other lights are off.
- **Initialized State:** the blue light is always on, indicating that it is in the operable state.
- **Received command state:** the red light blink once quickly (because the blue light is always on at this time, the gripper indicator light will looks like a purple light).
- **Object caught state:** green light is always on, other lights are off.
- **Object dropped state:** green light blinking.

1.2 Pinout Description

The line sequence definition on the gripper body is shown in Figure 1.1(a) and 1.1(b)

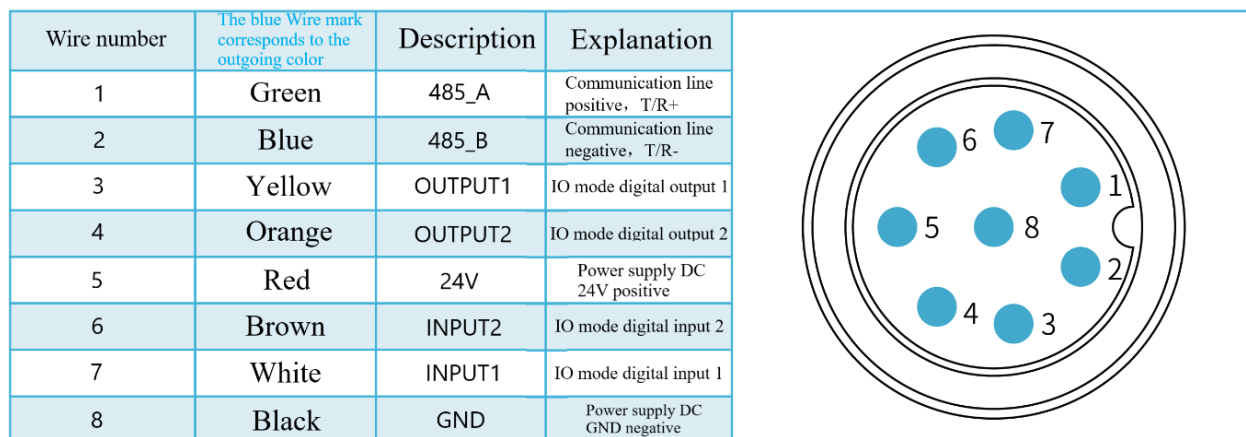


Figure 1.1 (a) The blue line marked diagram

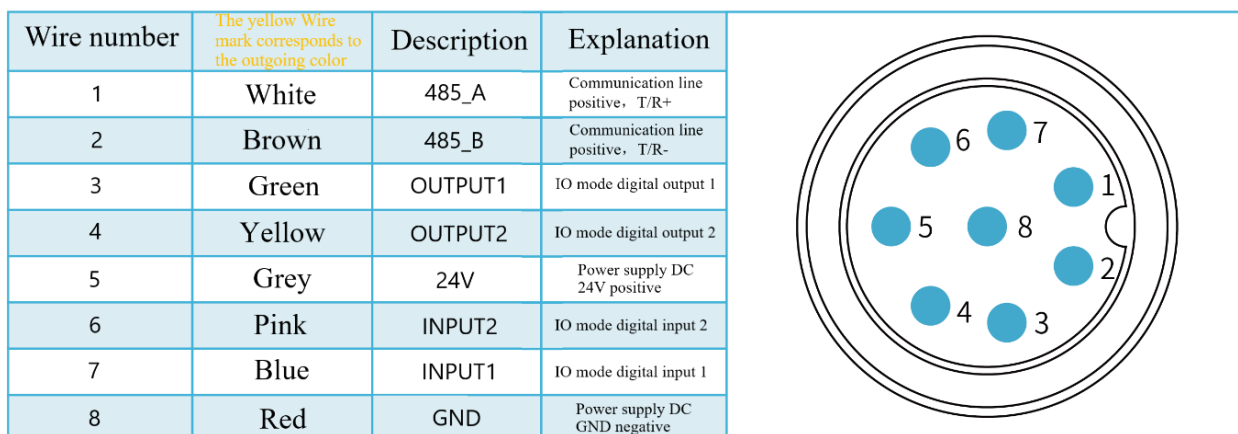


Figure 1.1 (b) The yellow line marked diagram

ote: Please distinguish the wire sequence according to the wire mark. If the wire mark is lost, dropped, or forgotten, please contact our staff to cooperate in determining the wire sequence. If you do not contact our staff, the clamping jaws will be damaged due to the wrong wiring sequence, and you will be responsible for the consequences.

2 Modbus-RTU Control

The gripper command is controlled by standard Modbus-RTU. For part of the description of the Modbus-RTU command, please refer to [2.3.1](#) command format (Modbus-RTU is a standard communication format on the market, widely used in the industrial field, please refer to the network for the specific format); please refer to [2.1.1](#) for the specific wiring method Debugging software installation and wiring; please refer to [2.3.3](#) for detailed instructions of commands for specific communication register address descriptions.

2.1 Debugging software description

The debugging software is specially used to control the gripper and set debugging parameters on the computer. Because there is no RS485 interface in the computer, the USB to 485 module is needed to convert the interface to USB interface, which is convenient for the debugging and control of the gripper in the computer.

2.1.1 Installation and wiring of debugging software

Connecting by debugging software is essentially controlled by RS485 interface. The specific connection needs to be connected to the 24 V, GND, 485_ A(T/R+,485+) ,485_ B (T / R -, 485 -) 4 wires in total. The power supply is a 24 V DC regulated power supply. Plug the USB port of the module into the USB port of the computer. The wiring definitions of different series are different. Please connect according to the instruction of specific clamping claw, as shown below:

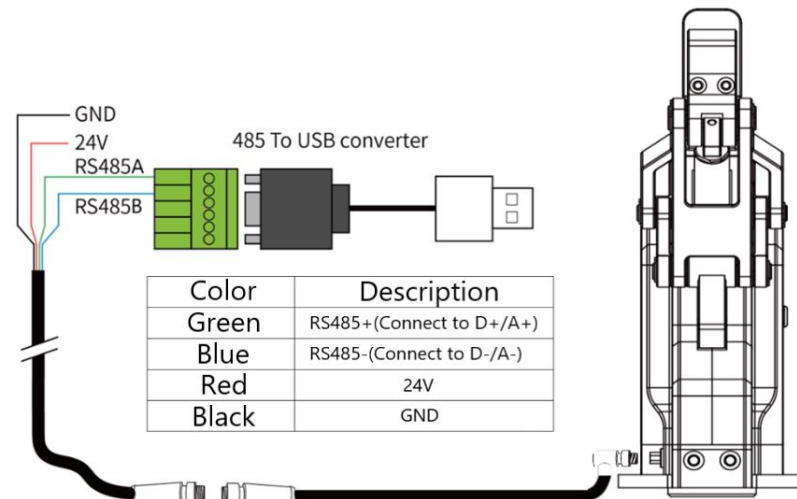


Figure 2.1 RS485 Connection

Wiring instructions

- ① : when the device (computer) has RS485 interface, the communication can be directly connected to RS485_A and RS485_B communication lines without transferring to 485 module through USB
- ② : in this way, other serial port debugging software (such as MODBUS poll) can be used for debugging

Software can be downloaded on the official website. Software and driver are integrated in the process of software installation, and both are installed together. It is recommended to check the create shortcut during installation.

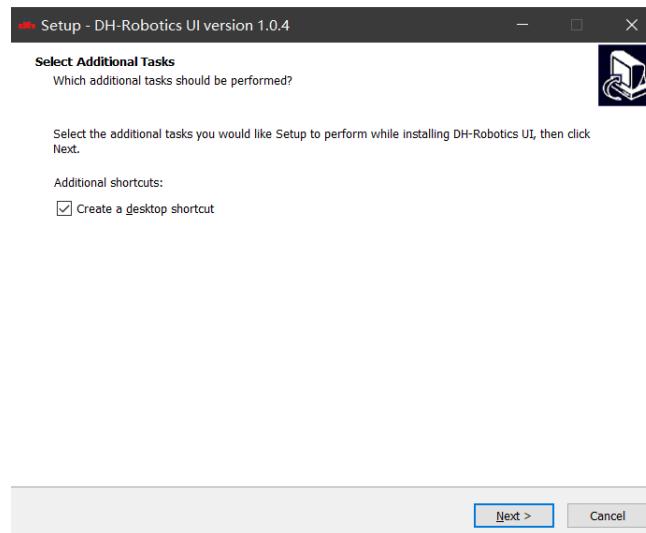


Figure 2.2 (a) installation interface 1

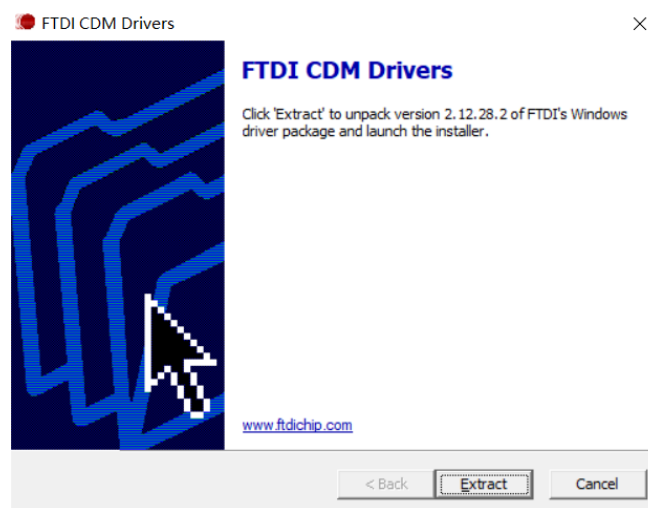


Figure 2.2 (b) driver installation interface

2.1.2 Debugging software instructions

Before use, it is necessary to connect the corresponding wiring according to the instructions (see [2.1.1](#) Installation and wiring of debugging software).

Open the software, the software will automatically identify the serial port, baud rate, ID number and other information of the gripper for automatic connection. As shown in the figure below:

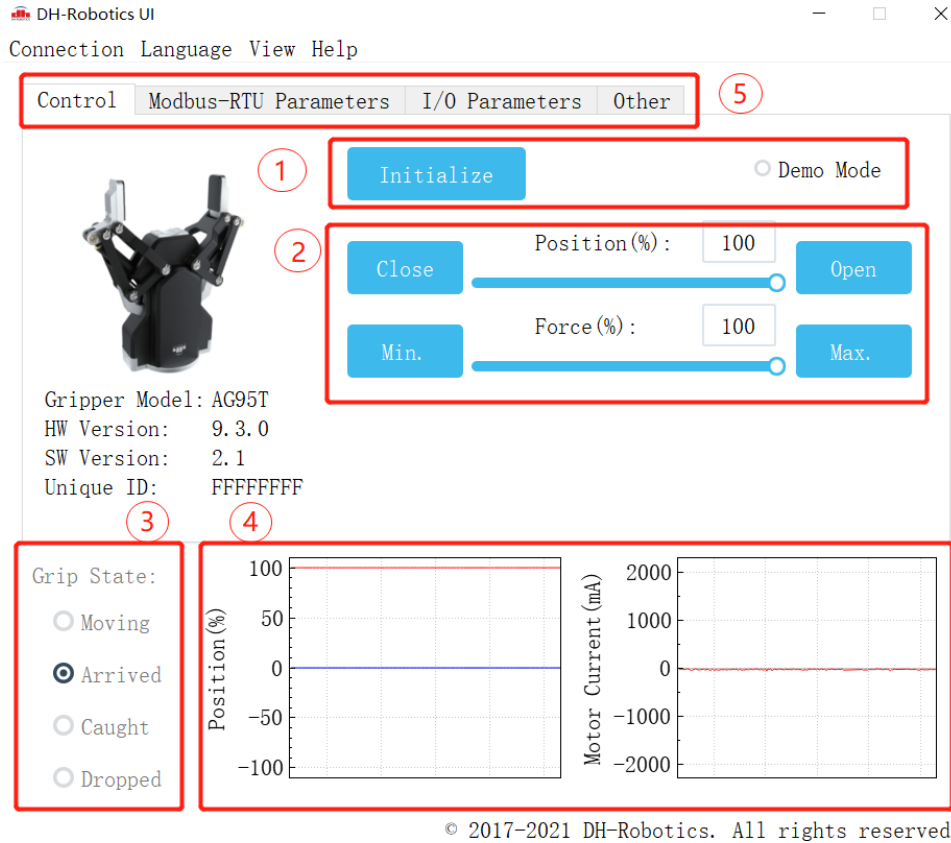


Figure 2.3 main control interface

The specific interface description is as follows:

Interface description

- ① **Initialization and demonstration mode:** the gripper needs to be initialized before operation to calibrate the zero point. The demonstration mode is a cyclic program.
- ② **Control interface:** it can control the position, force and speed of the gripper.
- ③ **Clamping status:** real time display of clamping status of clamping claw.
- ④ **Position current real time graph:** real time display position and current. The current represents the current of the internal motor, not the current actually consumed by the gripper. The current real-time graph can reflect the stability of clamping force.
- ⑤ **Parameter setting:** the configuration parameters of Modbus RTU, such as baud rate and check bit, can be configured; IO mode is to configure the parameters related to IO mode;

The gripper body uses Modbus RTU for communication, you can read and write data into the register. The data can be read and written at the view register. The data includes control, feedback, user parameters and I/O parameters as shown in the following figure:

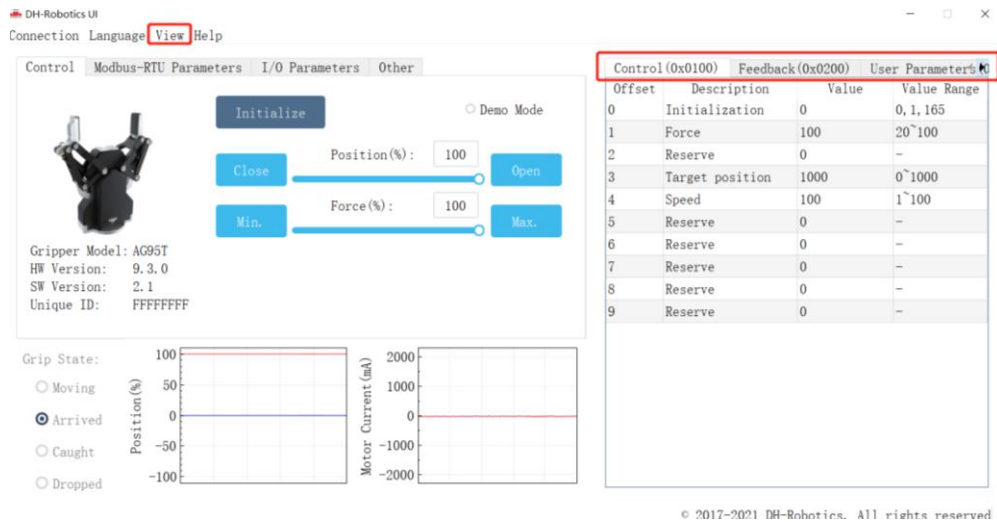


Figure 2.4 View

If there are multiple 485 devices, sometimes the baud rate and ID number of the gripper need to be modified, the parameters can be modified in Modbus RTU parameters

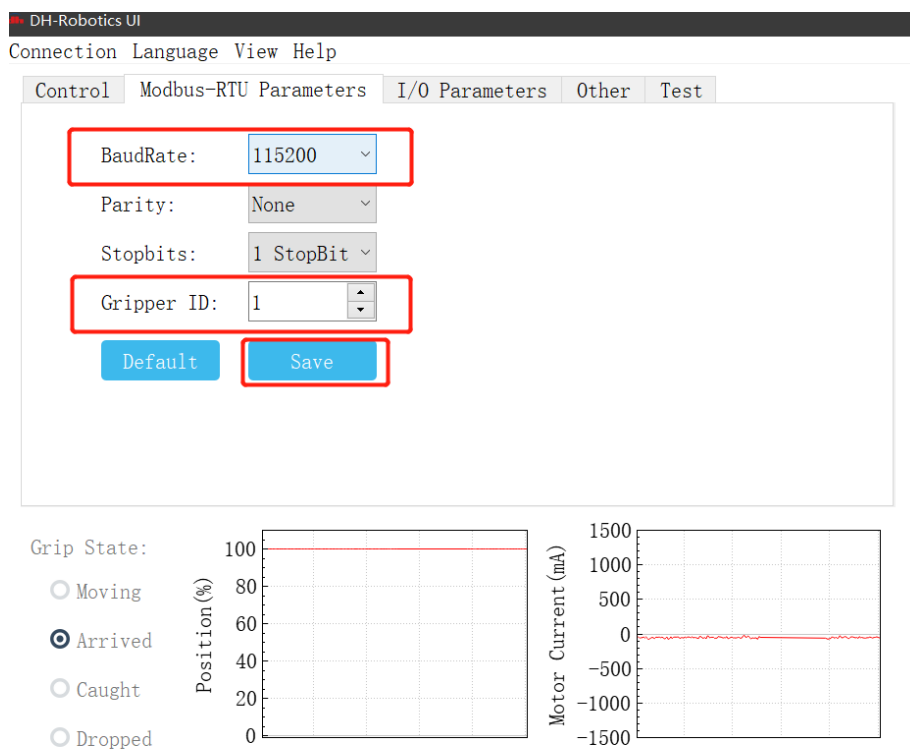


Figure 2.5 Modbus RTU parameters

You can set and configure the gripper I / O parameters in [I / O parameters]. After modifying the parameters, please click Save button to save. The following figure shows the operation of opening IO mode:

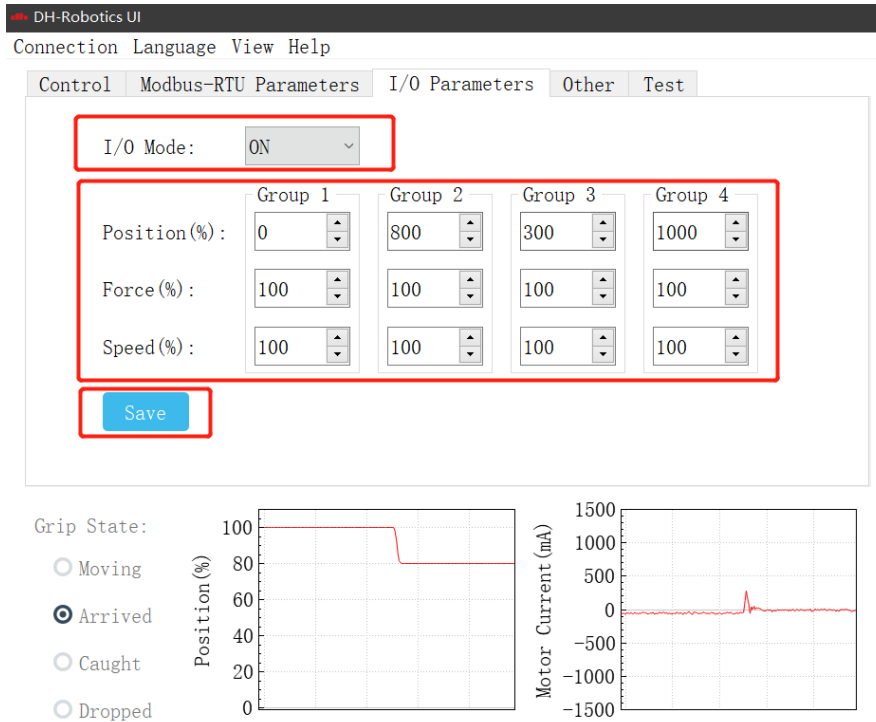


Figure 2.6 Modbus RTU parameters

The steps of switching IO are as follows:

Steps to switch IO mode

- ① **Open IO mode:** open IO mode first.
- ② **Configure four groups of IO parameters:** set the four groups of parameters of gripper, including position, force and speed.
- ③ **Save:** click the Save button to write the parameters to the internal register of flash, and restart to control.
- ④ **Restart:** after the restart, the switch to IO mode is successful. You can control the gripper according to the input signal, and the running status will be fed back through output.

2.2 Modbus-RTU default allocation

Slave Address : 1
 Baud Rate : 115200
 Data Bits : 8 bits
 Stop Bits : 1 stop bit
 Parity : None

2.3 Command description

2.3.1 RTU Framing

This gripper uses the standard Modbus-RTU protocol.

In RTU mode, the first field is the device address. The allowable characters transmitted for all fields are hexadecimal 0 ... 9, A ... F. Networked devices monitor the network bus continuously, including during the silent intervals. When the first field (the address field) is received, each device decodes it to find out if it is the addressed device.

A typical message frame is shown in Table 2.1.

Table 2.1 RTU Framing (Function Code:0x06)

Slave Address	Function	Register address	Register data	CRC
01	06	01 00	00 01	49 F6

Slave Address: The Slave address of the gripper. The default is 1, you can also modify it through write different value to Slave Address register.

Function: The Function Code field tells the addressed slave what function to perform. Includes read or write registers function.

Register address: Specifies which registers reference to be written.

Register data: Specifies which value to be written. Each register (word - 16 bits) of the Modbus RTU protocol is composed of 2 bytes (8 bits) from the Gripper.

CRC: the CRC error-checking field contains a 16-bit value implemented as two eight-bit bytes. The CRC field is appended to the message as the last field in the frame. The low-order byte of the field is appended first, followed by the high-order byte. The CRC high-order byte is the last byte to be sent in the message.

Supported Modbus Function Code:03 (HEX): Read Holding Registers;04 (HEX): Read Multiple Registers;06 (HEX): Write Single Register;10 (HEX): Write Multiple Registers

2.3.2 Register Mapping

The gripper's Modbus-RTU registers consist of two types of registers: **the basic control registers** and **the configuration registers**.

Basic control registers: initialization, force setting, reference position, speed, and some states.

Configuration registers: gripper's parameter configuration. Includes Modbus communication parameters and I/O parameters.

Table 2.2 Basic Control register map

Function	High bytes	Low bytes	Description	Write	Read
Initialization	0x01	0x00	Initialize the gripper	0x01: initialization; 0xA5: Fully initialization	Current setting
force		0x01	Gripper's force	20-100 (%)	Closing-force currently set
reserve		0x02	-	-	-
Position		0x03	Position	0-1000 (‰)	Reference position currently set
Initialization state	0x02	0x00	Initialization state of the gripper	Read Only	0: Not initialized; 1: Initialized
Gripper state		0x01	Gripper state	Read Only	0: In motion; 1: Reach position; 2 Object caught; 3: Object dropped
Position		0x02	gripper position	Read Only	Current actual position

Table 2.3 Configuration register map

Function	High bytes	Low bytes	Description	Write	Read
Save Parameter	0x03	0x00	Save all the parameters	0: default, 1: Write all parameters to save	0
Initialization direction		0x01	Configure initialization direction	0: Open, 1:Close (default: 0)	Current setting
Slave Address		0x02	Configure gripper Modbus address	0-255 (default: 1)	Current setting
Baud Rate		0x03	Configure gripper Modbus Baud rate	0-5: 115200, 57600, 38400, 19200, 9600, 4800 (default :0)	Current setting
Stop Bits		0x04	Configure gripper Modbus stop bits	0: 1 stop bit; 1: 2 stop bits (default: 0)	Current setting
Parity		0x05	Configure gripper Modbus Parity	0: None parity; 1: Odd parity; 2: Even parity (default: 0)	Current setting
I/O Parameters Test	0x04	0x00	Test I/O parameters	1; 2; 3; 4	Current setting
I/O Mode Switch		0x02	I/O control switch	0: OFF, 1: ON	Current setting
I/O Parameter Configuration		0x05-0x10	Four groups of I/O parameters	position 1, force 1 to position 4, force 4	Current setting

2.3.3 Register Description

2.3.3.1 Initialization

This register is used to initialize the gripper.

Write: If write 1 (0x01 hex) to this register, the gripper will be initialized (fingers move to the minimal or maximum position. The initialization direction depends on the value of initialization direction register). If write 165 (0xA5 hex) to this register will fully initialize the gripper(find the minimal and maximum position).

Read: if gripper need to be initialized or have initialized, this register value is 0; and if gripper is in initializing process, this register value is 1.

The register address is 0x0100. The description of this register is shown in Table 2.4.

Table 2.4 Initialization

Function	Address	Description	Write	Read
Initialization	0x0100	Initialize the gripper	0x01: initialize; 0xA5: Fully initialize	Current setting

The gripper needs to be initialized before control.

The sample command is as follows:

Initialize (write):

Send: 01 06 01 00 01 49 F6

Receive: 01 06 01 00 01 49 F6

Reinitialize(write):

Send:01 06 01 00 00 A5 48 4D

Receive: 01 06 01 00 00 A5 48 4D

2.3.3.2 Force

This register is used to set Force. It defines the current for the Gripper. If the current limit is exceeded, the fingers stop and trigger an object detection.

The address is 0x0101. The description of this register is shown in Table 2.5.

Table 2.5 Force

Function	Address	Description	Write	Read
Force	0x0101	Gripper's closing force	20-100 (%)	Force currently set

The force value range is 20-100, the corresponding value is 00 14–00 64(Hexadecimal).

Example:

Set 30% closing force (write):

Send: 01 06 01 01 1E 59 FE

Return: 01 06 01 01 1E 59 FE

Read the closing force currently set (read):

Send: 01 03 01 01 00 01 D4 36

Return: 01 03 02 xx xx crc1 crc2

2.3.3.3 Position

This register is used to set the reference position of gripper's fingers, then the fingers will move to the position immediately.

The address is 0x0103. The description of this register is shown in Table 2.6.

Table 2.6 Position

Function	Address	Description	Write	Read
Position	0x0103	Reference Position	0-1000 (‰)	Reference position currently set

The reference position value range is 0-1000 (‰), the corresponding value is 00 00 – 03 E8(Hexadecimal).

Example:

Set 500‰ position (write):

Send: 01 06 01 03 01 F4 78 21

Return: 01 06 01 03 01 F4 78 21

Read the reference position currently set(read):

Send: 01 03 01 03 00 01 75 F6

Return: 01 03 02 xx xx crc1 crc2

Read the current real-time position (read operation):

Send: 01 03 02 02 00 01 24 72

Return: 01 03 02 xx xx crc1 crc2

2.3.3.4 Initialization State

This register is used to store current initialization state of gripper, you can get the initialization state by reading this register.

The address is 0x0200. The description of this register is shown in Table 2.8.

Table 2.8 Initialization State

Function	Address	Description	Write	Read
Initialization State	0x0200	Initialization state of the gripper	Read Only	0: Not initialized; 1: Initialized

Example:

Read initialization state (read):

Send: 01 03 02 00 00 01 85 B2

Return: 01 03 02 00 00 B8 44

2.3.3.5 Gripper State

This register is used to store the Gripper state, you can get the state of gripper by reading this register.

And the address is 0x0201. The description of this register is shown in Table 2.9.

Table 2.9 Gripper State

Function	Address	Description	Write	Read
Gripper State	0x0201	the gripper state	Read Only	0: In motion; 1: Reached position; 2: Object caught; 3: Object dropped

States Description

Different values indicate different states of the gripper. The descriptions of states are as follows:

- 00: Fingers are in motion .
- 01: Fingers are at reference position. No object detected or object has been dropped.
- 02: Fingers have stopped due to an object detection.
- 03: Fingers are at reference position due to object has been dropped after the gripper caught object.

Example:

Read gripper state (read):

Send: 01 03 02 01 00 01 D4 72

Return: 01 03 02 00 02 39 85(02: object caught)

2.3.3.6 Current Position

This register is used to store the Actual position of the Gripper.

The address is 0x0202. The description of this register is shown in Table 2.10.

Table 2.10 Current Position

Function	Address	Description	Write	Read
Current Position	0x0202	Gripper actual position	Read Only	Current actual position

Example:

Read actual position (read):

Send: 01 03 02 02 00 01 24 72

Return: 01 03 02 xx xx crc1 crc2

2.3.3.7 Save Parameter

This register is used to Save Parameter.

Write 1 to this register to save all parameter, If you modified the I/O or communication parameters.

The address is 0x0300. The description of this register is shown in Table 2.11.

Table 2.11 Save Parameter

Function	Address	Description	Write	Read
Save Parameter	0x0300	Save register's value to Flash	0: default, 1: Save all parameters	0

Example:

Save Parameter (Write):

Send: 01 06 03 00 00 01 48 4E

Return: 01 06 03 00 00 01 48 4E

NOTE

- The Saving process will take 1-2 seconds, and the gripper won't response to other command during this process. The gripper will response this command after saving process finished.

2.3.3.8 Initialization Direction

This register is used to set Initialization Direction of gripper.

The address is 0x0301. The description of this register is shown in Table 2.12.

Table 2.12 Baud Rate

Function	Address	Description	Write	Read
Baud Rate	0x0301	Configure initialization direction	0: Open, 1:Close (default: 0)	Current setting

The value of this register is 0 by default.

If the register value is 0, when you send the initialization command, the gripper finger will open and find the maximum position.

If the register value is 1, when you send the initialization command, the gripper finger will close and find the minimal position.

Example:

Write 0 to initialization direction register:

Send: 01 06 03 01 00 00 D8 4E

Return: 01 06 03 01 00 00 D8 4E

2.3.3.9 Slave Address

This register is used to set Slave Address of gripper.

The address is 0x0302. The description of this register is shown in Table 2.13.

Table 2.13 Slave Address

Function	Address	Description	Write	Read
Slave Address	0x0302	Configure gripper Slave Address	1-247 (default: 1)	Current setting

The value of this register is 1 by default.

Example:

Set the Slave Address to 1 (write):

Send: 01 06 03 02 00 01 E9 8E

Return: 01 06 03 02 00 01 E9 8E

NOTE

- Please make sure that no other networked device has the same slave address as the gripper.

2.3.3.10 Baud Rate

This register is used to set Baud Rate of gripper.

The address is 0x0303. The description of this register is shown in Table 2.14.

Table 2.14 Baud Rate

Function	Address	Description	Write	Read
Baud Rate	0x0303	Configure gripper Modbus Baud rate	0-5: 115200, 57600, 38400, 19200, 9600, 4800 (default: 0)	Current setting

The value of this register is 0 by default, corresponding to a baud rate of 115200.

Example:

Set gripper baud rate to 115200 (write):

Send: 01 06 03 03 00 00 79 8E

Return: 01 06 03 03 00 00 79 8E

2.3.3.11 Stop Bits

This register is used to set Stop Bits of gripper.

The address is 0x0302. The description of this register is shown in Table 2.15.

Table 2.15 Stop bits settings

Function	Address	Description	Write	Read
Stop Bits	0x0304	Configure gripper Modbus stop bits	0: 1 stop bit 1: 2 stop bits (default: 0)	Current setting

The value of this register is 0 by default, corresponding to 1 stop bit.

Example:

Set the gripper stop bit to 1 stop bit (write):

Send: 01 06 03 04 00 00 C8 4F

Return: 01 06 03 04 00 00 C8 4F

2.3.3.12 Parity

This register is used to set Parity of gripper.

The address is 0x0305. The description of this register is shown in Table 2.16.

Table 2.16 Parity

Function	Address	Description	Write	Read
Parity	0x0305	Configure gripper Modbus Parity	0: None Parity 1: Odd Parity 2: Even Parity (default : 0)	Current setting

The value of this register is 0 by default, corresponding to None Parity.

Example:

Set the gripper's Parity to None Parity (write):

Send: 01 06 03 05 00 00 99 8F

Return: 01 06 03 05 00 00 99 8F

2.3.3.13 Test I/O Parameters

This register is used to test the I/O Parameters.

The address is 0x0400. The description of this register is shown in Table 2.17.

Table 2.17 I/O Control

Function	Address	Description	Write	Read
Test I/O Parameters	0x0400	Test I/O Parameters	1; 2; 3; 4	Current setting

This register can be used to directly test 4 groups of I/O parameters through Modbus-RTU to ensure that the I/O parameters are appropriate. For example, Write 1 to this register, the gripper will execute action with the first group of I/O parameter.

Example:

Control gripper by using first group of I/O parameter (write):

Send: 01 06 04 00 00 01 49 3A

Return: 01 06 04 00 00 01 49 3A

2.3.3.14 I/O Mode Switch

This register is used to turn I/O Control Mode ON or OFF.

The address is 0x0402. The description of this register is shown in Table 2.18.

Table 2.18 I/O Mode Switch

Function	Address	Description	Write	Read
I/O Mode Switch	0x0402	I/O Control Switch	0: OFF, 1: ON	Current setting

If you have written 1 to this register and have saved all parameters, the gripper will be initialized automatically after power on.

When the I/O Control Mode is turned on, the gripper can respond to Modbus-RTU commands and I/O, but I/O has priority.

The control method in different mode is shown in Table 2.19.

Table 2.19 Control method

Switch State	Description	Modbus-RTU	I/O
0	I/O control mode off	YES	No
1	I/O control mode on	YES	YES

Example:

Set the I/O control mode switch off (write):

Send: 01 06 04 02 00 00 29 3A

Return: 01 06 04 02 00 00 29 3A

NOTE

- If you just need to control the gripper through Modbus RTU, you should write 0 to this register and save all parameters to turn off the I/O control mode.

2.3.3.15 I/O Parameter Configuration

Those registers are used to Set the I/O Parameters.

The address is 0x0405-0x0410. The description of this register is shown in Table 2.20.

Table 2.20 I/O Parameter Configuration

Function	High-byte	Low bytes	Description	Write	Read
I/O Group 1	0x04	0x05	position 1	0-1000‰	Current setting
		0x06	force 1	20-100 %	
I/O Group 2		0x08	position 2	0-1000‰	
		0x09	force 2	20-100 %	
I/O Group 3		0x0B	position 3	0-1000‰	
		0x0C	force 3	20-100 %	
I/O Group 4		0x0E	position 4	0-1000‰	
		0x0F	force 4	20-100 %	

Example:

Set the first group of I/O parameter (write) :

Send: 01 06 04 05 01 2C 98 B6 (Reference position: 300‰)

Return: 01 06 04 05 01 2C 98 B6

Send: 01 06 04 06 00 1E E8 F3 (Force: 30‰)

Return: 01 06 04 06 00 1E E8 F3

IO parameter address is continuous address, and four groups of IO parameters can be configured at one time by using the function code of 0x10, as follows:

Continuous multiple address write(write)

[Group 1:1000‰position;20%force;

Group 2:100‰position;20%force;

Group 3:0‰position;100%force;

Group 4:592‰position;100%force]:

Send: 01 10 04 05 00 0C 18 03 E8 00 14 00 00 01 00 00 14 00 00 00 00 00 64 00 00 02 50 00 64 00 00 D6 D8

Return: 01 10 04 05 00 0C D1 3D

3 I/O Control

IO mode is a common control method in the industry, which controls the clamping jaws in the form of hardware wiring. When using IO control, you need to set the gripper jaws to IO mode in advance, and set the 4 groups of gripper IO states.

3.1 I/O Configuration

The I/O mode is a common control method in industry.

The grippers will monitor the pin states of Input 1 and Input 2 (0V and high resistance states). For these two pins, there will be four logic states:00,01,10,11. You can control this gripper through changing the states of Input 1 and Input 2. As shown in Table 3.1. The specific configuration is shown in the figure below:

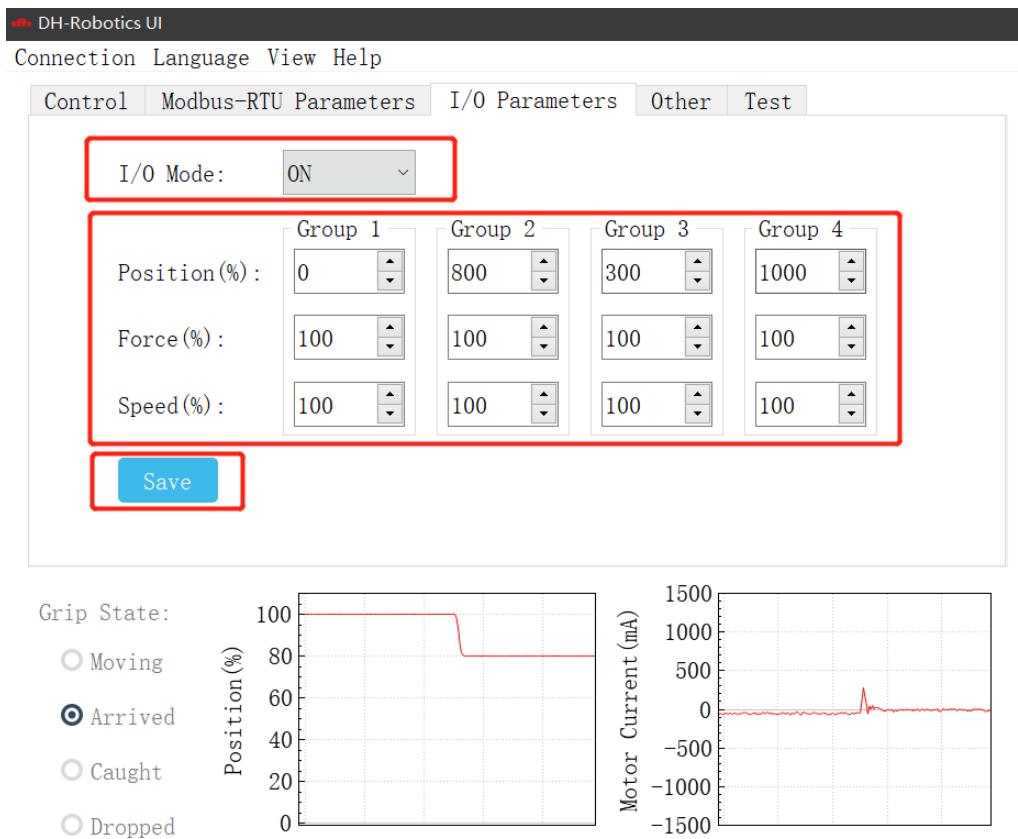


Figure 3.1 I/O mode

Steps to switch IO mode

- ① **Open IO mode:** open IO mode first.
- ② **Configure four groups of IO parameters:** set the four groups of parameters of gripper, including position, force and speed.
- ③ **Save:** click the Save button to write the parameters to the internal register of flash, and restart to control.
- ④ **Restart:** after the restart, the switch to IO mode is successful. You can control the gripper according to the input signal, and the running status will be fed back through output.

Table 3.1 Input State

INPUT 1	INPUT 2	Pin state	I/O state	Perform action
No wiring	No wiring	0 0	Group 1	Target position 1,target force 1,target speed 1
0V	No wiring	1 0	Group 2	Target position 2,Target Force 2,Target Speed 2
No wiring	0V	0 1	Group 3	Target position 3,Target Force 3,Target Speed 3
0V	0V	1 1	Group 4	Target position 4,Target force 4,target speed 4

You can also get the gripper state by detecting the states of Output1 and Output 2(0V and high resistance states). as shown in Table 3.2.

Table 3.2 Output1 Output2 State

OUTPUT 1	OUTPUT 2	I/O State	State description
No wiring	No wiring	0 0	Fingers are in motion
0V	No wiring	1 0	Fingers are at reference position, No object detected or object has been dropped
No wiring	0V	0 1	Fingers have stopped due to an object detection
0V	0V	1 1	The gripper detects an object falling

NOTE

- Digital IO default input and output are NPN type, input and output are 0V valid (low level valid).
- Input and output can be configured. If it needs to be changed to PNP type, the input and output of 24 V is effective (high level is effective), it is necessary to communicate with our company in advance.

3.2 IO use

After configuring the parameters, 24V, GND, INPUT 1, INPUT 2, OUTPUT 1, and OUTPUT 2 need to be connected to the hardware.

Connect INPUT and OUPUT to the corresponding equipment, confirm that the wiring is correct and restart, the gripper will automatically initialize. Then control the gripper according to the INPUT signal. The running status will be fed back through OUTPUT.

4 Maintain

4.1 Daily cleaning

The daily cleaning frequency is shown in Table 4.1.

Table 4.1 Daily cleaning

Suggest the best cleaning frequency	Tools needed	Parts needed
Once a week (If the environment is dirty, once a day)	1. Circlip pliers 2. Dry towels or paper towels	without

Note: The surface of the paw is not waterproof, just clean it with a dry towel or paper towel.

Notice

- Please turn off the robot and the gripper power supply before performing any operation with the hand gripper.
- Cleaning personnel must wear electrostatic wristbands and other electrostatic measures to avoid damaging electronic components.

The cleaning steps are as follows, as shown in Figure 4.1:

1. Use circlip pliers to remove the $\phi 6$ retaining ring (located at both ends of each joint of the connecting rod finger).
2. Remove the $\phi 6$ shaft.
3. Use a dry towel or paper towel to clean up the debris, dirt and dust on the surface of the claw; use a dry towel or paper towel to clean the $\phi 6$ shaft.
4. Apply grease to the $\phi 6$ shaft.
5. Reinstall the $\phi 6$ shaft and install the $\phi 6$ retaining ring with circlip pliers.

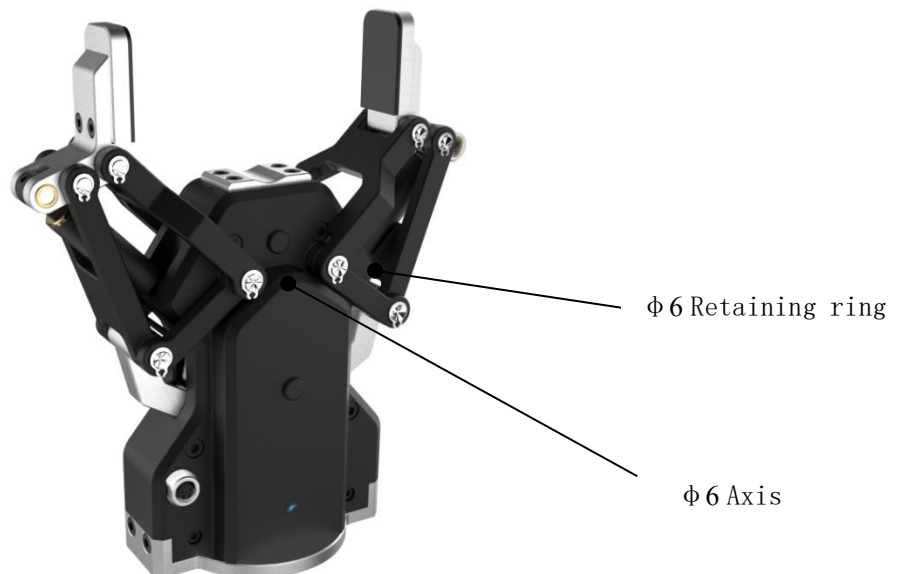


Figure 4.1 AG- series Gripper diagram

4.2 Fingertip replacement

The frequency of fingertip replacement is shown in Table 4.2.

Table 4.2 Fingertip replacement

Check frequency	Tools needed	Parts needed
When gripping 1,000,000 times or the fingertips are severely damaged	M3 hexagon socket screwdriver	A pair of (2) Big Huan two-finger adaptive robot fingertips (with rubber)

Note: Contact Dahuan Robot Technical Support to replace fingertip parts

Notice

- Please turn off the robot and the gripper power supply before performing any operation with the hand gripper.

The fingertip replacement steps are as follows, as shown in Figure 4.2:

1. Use an M3 Allen screwdriver to remove the fingertip screws and remove the worn fingertips.
2. Clean the fingers and dry them thoroughly.
3. Take out the $\phi 3 \times 6$ mm positioning pin on the fingertip.
4. Insert the $\phi 3 \times 6$ mm positioning pin into the new fingertip.
5. Attach the new fingertips to the claws and fix them with screws.
6. Repeat the above operation with the other fingertip to replace it.

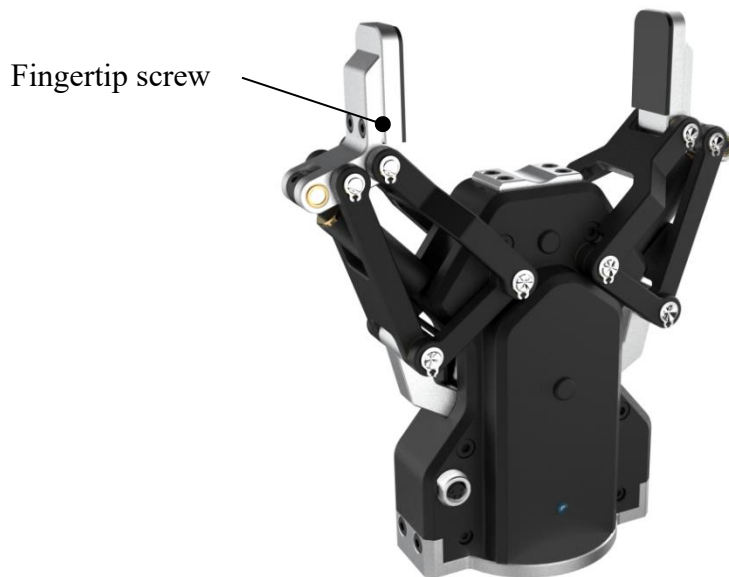


Figure AG- series Gripper diagram

4.3 Periodic inspection and nursing

The frequency of regular inspection and testing is shown in Table 4.3.

Table 4.3 Periodic inspection

Check frequency	Tools needed	Parts needed
Once a month	1. M3 hexagon socket screwdriver 2. M4 hexagon socket screwdriver 3. Circlip pliers	without(unless it is damaged and needs to be replaced)

Notice

- Please turn off the robot and the gripper power supply before performing any operation with the hand gripper.
- The cleaning staff must wear electrostatic wristbands and other electrostatic measures to avoid damaging the electronic components.

The steps to check the gripper are as follows:

1. The movement of the fingers must be symmetrical and smooth: open the fingers to test the stroke, and the fingers must return to the initial starting position by themselves; check whether the claws are stuck or shaken during operation.
2. The wear of the finger rubber should not affect the clamping. If the wear is severe and affects the operation of the claw, please replace the finger pad.
3. Check for collision damage. If there is damage, please contact Dahuan Robotics technical support.
4. Check whether there is abrasion at the bottom of the gripper. If the abrasion is severe and affects the gripper work, please contact the technical support of Dahuan Robotics.
5. Check all screws for rust or damage, and replace them if they are.