ELECTRIC GRIPPER

EG2-4X2

USER MANUAL

Beijing Inspire-Robots Technology Co., Ltd. | January 2024
ID: PRJ-03-TS-U-001 | Page Number: 25 pages | Version: V1.0.7

Table of Contents

Electric Gripper EG2-4X2

User Manual

1 Product Overview

1.1 Product Features

The EG2 Series Electric Gripper is an electric gripper designed to integrate the linear servo actuator with small volume and large torque (hereinafter referred to as "Electric Gripper"). It is equipped with a linear servo actuator. For the user interface, the RS485 communication interface is used. It supports the MODBUS RTU protocol. There is a sensitive built-in pressure sensor. By setting different pressure threshold values, the user can easily grip objects with different hardness. Concise and efficient interface control instructions enables the user to quickly manipulate and control the gripper. The excellent performance makes this gripper suitable for applications such as service robot, teaching aids, etc. The gripper has the following features:

- Grip force: EG2-4B2 can realize the grip force of 1.5 kg, and EG2-4C2 can realize the grip force of 2 kg;
- Supply voltage: DC 7 V to 9 V power supply in a wide voltage range is available, and 8 V is recommended (24 V power supply is allowed for the products with special designation);
- Positioning repeatability: ±0.5 mm;
- Maximum opening: 70 mm;
- Communication interface: EG2-4X2 model uses the RS485 serial port and supports the MODBUS RTU protocol.

1.2 Electric connection

1.2.1 Pin definition

The electrical interface of the gripper is shown in Figure 2. The interface is a standard micro aviation connector XS6-5P. The pins are defined as follows:

	RS232 interface	RS485 interface	CAN bus interface

1	GND	GND	GND
2	VCC	VCC	VCC
3	RX (RX pin of the gripper)	A+	CANH
4	TX (TX pin of the gripper)	B-	CANL
5	GND pin	GND pin	GND pin

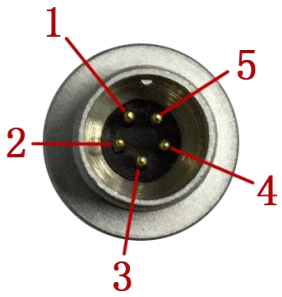

Figure 2: Electrical Interface Diagram of the Gripper

1.2.2 Communication mode

EG2-4X2 uses the RS485 standard interface to realize parallel connection and control of 254 grippers to the same bus. For the purpose of communication control, the grippers connected to the same bus should be configured with different ID.

RS232 communication gripper and CAN communication gripper are customized.

2 Communication Protocol of Common Serial Ports

2.1 Summary of Communication Protocol

Q&A communication is used between the main control unit (MCU) and the gripper. The MCU actively sends the instruction frame; after receiving the instruction frame, the gripper will return the response frame after resolution.

In one control network, a MCU is allowed to connect and control multiple grippers, so each gripper needs to be configured by the user with a different ID as unique identifier (the factory default ID of any gripper is 1). The data volume of the instruction frame sent by the MCU includes the ID information. The gripper can completely receive the instruction frame only if the ID matches, and will return the corresponding response frame after processing the instruction.

2.2 Instruction frame

Format of basic instruction frame:

Frame header (FH) (2 bytes)		ID (1 byte)	Length of data volume (1 byte)	Instruction type of Data Volume 1/2 (1 byte)	Data of Data Volume 2/2 (Len-1 bytes)	Checksum (1 byte)
0xEB	0x90	ID	Len	CMD	Data	Check_Sum

Frame header (FH): Continuous receipt of 0xEB and 0x90 indicates arrival of the instruction frame.

ID: Each gripper has an ID. The range of ID is from 0x01 to 0xFE (hexadecimal number), i.e., 1 to 254. The ID for broadcast is 255 (0xFF). If the ID sent by the controller is 255 (0xFF), all grippers will receive the instruction frame, without returning any response message.

Length of data volume: Length of the data segment to be sent, including the instruction type and data, i.e., "Len"; the total instruction frame length is "Len+5".

Data Volume 1/2 - instruction type: This specifies the type of the instruction frame.

CMD_MC_PARA_SAVE (0x01): Parameters will be saved in the internal flash memory and will not be lost after power failure.

CMD_MC_PARA_ID_SET (0x04): Set the gripper ID.

CMD_MC_MOVE_CATCH_XG (0x10): Grasp an object based on the set speed and force control threshold.

CMD_MC_MOVE_CATCH2_XG (0x18): Continuously grasp objects based on the set speed and force control threshold.

CMD_MC_MOVE_RELEASE (0x11): Unclench the gripper at the set speed.

CMD_MC_SEEKPOS (0x54): Specify the gripper opening.

CMD_MC_MOVE_STOPHERE (0x16): Emergency stop

CMD_MC_SET_EG_PARA (0x12): Set the maximum/minimum value of gripper opening.

MD_MC_READ_EG_PARA (0x13): Read the maximum/minimum value of gripper opening.

CMD_MC_READ_ACTPOS (0xD9): Read the gripper opening.

CMD_MC_READ_EG_STATE (0x14): Read the status of the current gripper.

CMD_MC_READ_EG_RUNSTATE (0x41): Read the running status of the current gripper.

Data Volume 2/2 - data: Data content sent with the instruction frame. For a read instruction, this data volume has no byte; for a write instruction, it is the content to be written.

Checksum (Check_Sum): Low-order bytes of the sum of all data before checksum in the frame except the frame header (2 bytes)

2.3 Response frame

2.3.1 Format of response frame

Format of response frame:

Frame header (FH) (2 bytes)		ID (1 byte)	Length of data volume (1 byte)	Instruction type of Data Volume 1/2 (1 byte)	Data of Data Volume 2/2 (Len-1 bytes)	Checksum (1 byte)
0xEE	0x16	ID	Len	CMD	Data	Check_Sum

Frame header (FH): Continuous receipt of 0xEE and 0x16 indicates arrival of the response frame.

Length of data volume: Instruction type (1 Byte) and data (Len-1 Bytes); the total length of the response frame is "Len+5".

ID: Each gripper has an ID. The range of ID is from 0x01 to 0xFE (hexadecimal number), i.e., 1 to 254. The ID for broadcast is 255 (0xFF). If the ID sent by the controller is 255 (0xFF), all grippers will receive the instruction frame, without returning any response message.

Data Volume 1/2 - instruction type: This specifies the type of the instruction frame corresponding to this response frame.

Data Volume 2/2 - data: Data content sent with the response frame. For a read instruction, it is the data content to be read by the main control unit (MCU); for a write instruction, it is one byte; 0x01 indicates successful writing; 0x55 indicates writing failure.

Checksum (Check_Sum): Low-order bytes of the sum of all data before checksum in the frame except the frame header (2 bytes)

2.3.2 Response time

The minimum response time corresponding to the response frame format of the gripper is 165us. The response time of parameter fixing and restoring factory defaults is about 850ms (including the time of Flash erasing and writing). It is recommended to control the instruction interval above 5ms during use; parameter fixing needs more than 1s.

2.4 Instruction type

The type of instruction frame is shown below:

Instruction type	Function description	Values Hex	Length of data content Bytes
CMD_MC_PARA_SAVE	Parameters will be saved in the internal flash memory and will not be lost after power failure.	0x01	0
CMD_MC_PARA_ID_SET	Set the gripper ID.	0x04	1
CMD_MC_MOVE_CATCH_XG	Grasp an object based on the set speed and force control threshold.	0x10	3

Instruction type	Function description	Values Hex	Length of data content Bytes
CMD_MC_MOVE_CATCH2_XG	Continuously grasp objects based on the set speed and force control threshold.	0x18	3
CMD_MC_MOVE_RELEASE	Unclench the gripper at the set speed.	0x11	1
CMD_MC_MOVE_STOPHERE	Emergency stop	0x16	0
CMD_MC_SEEKPOS	Specify the gripper opening.	0x54	2
CMD_MC_SET_EG_PARA	Set the maximum/minimum value of gripper opening.	0x12	4
CMD_MC_READ_EG_PARA	Read the maximum/minimum value of gripper opening.	0x13	0
CMD_MC_READ_ACTPOS	Read the gripper opening.	0xD9	0
CMD_MC_READ_EG_STATE	Read the opening of the current gripper, as well as the current value and set threshold of the pressure sensor.	0x14	0
CMD_MC_READ_EG_RUNSTATE	Read the gripper running status.	0x41	0

2.4.1 Parameter fixing

Function: The main control unit (MCU) saves the parameter of maximum/minimum opening used by the current gripper in the internal flash memory. As a result, the parameter will not be lost after power failure.

Length of instruction frame: 6 bytes

Instruction type: 0x01 (CMD_MC_PARA_SAVE)

Data content: None

For example, when the MCU fixes the parameter of a gripper (ID=1), the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content (0 bytes)	Checksum B5 (1 byte)
0xEB	0x90	0x01	0x01	0x01		0x03

The instruction frame sent is "EB 90 01 01 01 03", where "EB 90" is the frame header (FH); the first "01" is the ID; the second "01" is the length of data volume; the third "01" is the parameter fixing instruction (CMD_MC_PARA_SAVE); "03" is the checksum, i.e., low-order bytes ((B2+B3+...+B4) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (1 byte)	Checksum B6 (1 byte)
0xEE	0x16	0x01	0x 02	0x 01	0x 01	0x 05

The specific response frame received is "EE 16 01 02 01 01 05", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; the second "01" is the instruction type; the third "01" is the successful instruction reception flag ("55" indicates an exception, and re-sending is required); "05" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.2 ID setting

Function: The MCU sets the ID of a gripper.

Length of instruction frame: 7 bytes

Instruction type: 0x04 (CMD_MC_PARA_ID_SET)

Data content: new ID

For example, when the MCU sets the new ID of a gripper (ID=1) to 3, the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (1 byte)	Checksum B6 (1 byte)
0xEB	0x90	0x01	0x02	0x04	0x03	0x0A

The instruction frame sent is "EB 90 01 02 04 03 0A", where "EB 90" is the frame

header (FH); "01" is the ID; "02" is the length of data volume; "04" is the ID setting instruction (CMD_MC_PARA_ID_SET); "03" is the new ID; "0A" is the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (12 bytes)	Checksum B6 (1 byte)
0xEE	0x16	0x01	0x02	0x04	0x01	0x08

The specific response frame received is "EE 16 01 02 04 01 08", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "04" is the instruction type; the second "01" is the successful instruction reception flag ("55" indicates an exception, and re-sending is required); "08" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.3 Force control for grasping objects

Function: The MCU enables gripper setting, so that a gripper can grasp objects based on the preset speed and force control threshold. Once the grip force exceeds the preset force control threshold, the gripper will stop moving.

Length of instruction frame: 9 bytes

Instruction type: 0x10 (CMD_MC_MOVE_CATCH_XG)

Data content: B5B6 indicates speed (max.=1000; min.=1; dimensionless unit); B7B8 indicates the force control threshold from 50 to 1000 (in order of low-order bytes followed by high-order bytes; dimensionless unit). For example, when the force control threshold is set to 300, the gripper will grasp an objects with the force of 300 g.

For example, when the MCU sets the parameters (speed=500, grasping threshold=100 g) of a gripper (ID=1) for grasping objects, the following instruction frame should be sent.

Frame header (FH) B0-B1	ID B2	Length of data volume B3	Instruction type B4	Data content B5-B8	Checksum B9

(2 bytes)		(1 byte)	(1 byte)	(1 byte)	(4 bytes)	(1 byte)
0xEB	0x90	0x01	0x05	0x10	0xF4 0x01 0x64 0x00	0x6F

The instruction frame sent is "EB 90 01 05 10 F4 01 64 00 6F", where "EB 90" is the frame header (FH); the first "01" is the ID; "04" is the length of data volume; "10" is the force control instruction for grasping (CMD_MC_MOVE_CATCH_XG); "01F4" indicates the speed of 500 (dimensionless); "0064" is the force control threshold of 100 g; "6F" is the checksum, i.e., low-order bytes ((B2+B3+...+B8) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. After this instruction frame is sent to the gripper, it starts to grasp an object. Once the grip force exceeds 100 g, it will stop moving. The response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (1 byte)	Checksum B6 (1 byte)
0xEE	0x16	0x01	0x02	0x10	0x01	0x14

The specific response frame received is "EE 16 01 02 10 01 14", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "10" is the instruction type; the second "01" is the successful instruction identification flag; "14" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.4 Force control for continuously grasping objects

Function: The MCU enables gripper setting, so that a gripper can grasp objects based on the preset speed and force control threshold. Once the grip force exceeds the preset force control threshold, the gripper will stop moving. If it is detected that the grip force is lower than the force control threshold after the gripper stops moving, the gripper will continue to grasp objects until the grip force exceeds the preset force control threshold.

Length of instruction frame: 9 bytes

Instruction type: 0x18 (CMD_MC_MOVE_CATCH2_XG)

Data content: B5B6 indicates speed (max.=1000; min.=1; dimensionless unit); B7B8 indicates the force control threshold from 50 to 1000 (in order of low-order bytes followed by high-order bytes; dimensionless unit). For example, when the force control threshold is

set to 300, the gripper will grasp an object with the force of 300 g.

For example, when the MCU sets the parameters (speed=500, grasping threshold=100 g) of a gripper (ID=1) for grasping objects, the following instruction frame should be sent.

Frame header (FH) B0-B1(2 bytes)		ID B2 (1 byte)	Length of data volume B3(1 byte)	Instruction type B4(1 byte)	Data content B5-B8 (4 bytes)	Checksum B9 (1 byte)
0xEB	0x90	0x01	0x05	0x18	0xF4 0x01 0x64 0x00	0x77

The instruction frame sent is "EB 90 01 05 18 F4 01 64 00 77", where "EB 90" is the frame header (FH); "01" is the ID; "0x04" is the length of data volume; "0x18" is the force control instruction for continuously grasping objects; "01F4" indicates the speed of 500 (dimensionless); "0064" is the force control threshold of 100 g; "77" is the checksum, i.e., low-order bytes ((B2+B3+...+B8) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. After this instruction frame is sent to the gripper, it starts to grasp an object. Once the grip force exceeds 100 g, it will stop moving. The response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (1 byte)	Checksum B6 (1 byte)
0xEE	0x16	0x01	0x02	0x18	0x01	0x1C

The specific response frame received is " EE 16 01 02 18 01 1C", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "18" is the instruction type; the second "01" is the successful instruction identification flag; "1C" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.5 Unclenching

Function: The MCU enables gripper setting, so that the gripper can be unclenched to its maximum opening based on the entered speed parameter.

Length of instruction frame: 7 bytes

Instruction type: 0x11 (CMD_MC_MOVE_RELEASE)

Data content: B5B6 indicates speed (max.=1000; min.=1; dimensionless unit).

For example, when the MCU sets the parameter (speed=500) of a gripper (ID=1) to unclench it to the maximum opening, the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5B6 (1 byte)	Checksum B7 (1 byte)
0xEB	0x90	0x01	0x03	0x11	0xF4 0x01	0x0A

The instruction frame sent is "EB 90 01 03 11 F4 01 0A", where "EB 90" is the frame header (FH); "01" is the ID; "03" is the length of data volume; "11" is the instruction for unclenching the gripper (CMD_MC_MOVE_RELEASE); "01F4" indicates the speed of 500 (dimensionless); "0A" is the checksum, i.e., low-order bytes ((B2+B3+...+B6) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, it will move to the maximum opening at the speed (500). The response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (12 bytes)	Checksum B6 (1 byte)
0xEE	0x16	0x01	0x02	0x11	0x01	0x15

The specific response frame received is "EE 16 01 02 11 01 15", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "11" is the instruction type; the second "01" is the successful instruction identification flag; "15" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.6 Specify the opening.

Function: The MCU specifies the target opening of a gripper. After the gripper receives this instruction, if the current opening is less than the set value, the gripper will be unclenched at the set speed until its actual opening reaches the target opening; and then the gripper will stop moving. If the current opening is larger than the set value, the gripper will grasp an object based on the set speed and force control threshold. Once the grip force exceeds the set force control threshold, or the actual opening reaches the target opening, the

gripper will stop moving.

Length of instruction frame: 7 bytes

Instruction type: 0x54 (CMD_MC_SEEKPOS)

Data content: B5B6 indicates opening (max.=1000 (70 mm); min.=1 (0 mm); dimensionless unit).

For example, when the MCU sets the opening (500 (35 mm)) of a gripper (ID=1), the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5B6 (1 byte)	Checksum B7 (1 byte)
0xEB	0x90	0x01	0x03	0x54	0xF4 0x01	0x4D

The instruction frame sent is "EB 90 01 03 54 F4 01 4D", where "EB 90" is the frame header (FH); "01" is the ID; "03" is the length of data volume; "54" is the instruction for unclenching the gripper (CMD_MC_MOVE_RELEASE); "01F4" indicates the speed of 500 (dimensionless); "4D" is the checksum, i.e., low-order bytes ((B2+B3+...+B6) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, it will move to the maximum opening at the speed (500). The response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (12 bytes)	Checksum B6 (1 byte)
0xEE	0x16	0x01	2 (0x02)	0x54	0x01	0x58

The specific response frame received is "EE 16 01 02 54 01 58", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "54" is the instruction type; the second "01" is the successful instruction identification flag; "58" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.7 Emergency stop

Function: The MCU makes a gripper in motion stop moving and stay in the position through this instruction.

Length of instruction frame: 6 bytes

Instruction type: 0x16 (CMD_MC_MOVE_STOPHERE)

Data content: None

For example, when the MCU sets a gripper (ID=1) to enter the emergency stop state, the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content (0 byte)	Checksum B5 (1 byte)
0xEB	0x90	0x01	1 (0x01)	0x16	None	0x18

The instruction frame sent is "EB 90 01 01 16 18", where "EB 90" is the frame header (FH); the first "01" is the ID; the second "01" is the length of data volume; "16" is the emergency stop instruction (CMD_MC_MOVE_STOPHERE); "0A" is the checksum, i.e., low-order bytes ((B2+B3+...+B4) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (12 bytes)	Checksum B6 (1 byte)
0xEE	0x16	0x01	2 (0x02)	0x16	0x01	0x1A

The specific response frame received is "EE 16 01 02 16 01 1A", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "16" is the instruction type; the second "01" is the successful instruction reception flag ("55" indicates an exception, and resending is required); "1A" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.8 Opening setting

Function: The MCU sets the maximum and minimum opening of a gripper.

Length of instruction frame: 10 bytes

Instruction type: 0x12 (CMD_MC_SET_EG_PARA)

Data content: B5B6 indicates the maximum opening (data range: 0-1000; dimensionless unit). B7B8 indicates the minimum opening (data range: 0-1000; dimensionless unit).

For example, when the MCU sets the maximum opening (1000) and the minimum opening (112) of a gripper (ID=1), the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5-B8 (4 bytes)	Checksum B9 (1 byte)
0xEB	0x90	0x01	5 (0x05)	0x12	0xE8 0x03 0x70 0x00	0x73

The instruction frame sent is "EB 90 01 05 12 E8 03 70 00 73", where "EB 90" is the frame header (FH); "01" is the ID; "05" is the length of data volume; "12" is the gripper opening setting instruction (CMD_MC_SET_EG_PARA); "73" is the checksum, i.e., low-order bytes ((B2+B3+...+B8) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (12 bytes)	Checksum B6 (1 byte)
0xEE	0x16	0x01	2 (0x02)	0x12	0x01	0x16

The specific response frame received is "EE 16 01 02 12 01 16", where "EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "12" is the instruction type; the second "01" is the successful instruction identification flag (if the set data is incorrect, 0x55 will be returned); "16" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.9 Opening reading

Function: The MCU reads the opening of a gripper.

Length of instruction frame: 6 bytes

Instruction type: 0x13 (MD_MC_READ_EG_PARA)

Data content: None

For example, when the MCU reads the opening of a gripper (ID=1), including the maximum and minimum opening values, their data range is the same (0-1000; dimensionless unit). The following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content -- (0 bytes)	Checksum B5 (1 byte)
0xEB	0x90	0x01	1 (0x01)	0x13		0x15

The instruction frame sent is "EB 90 01 01 13 15", where "EB 90" is the frame header (FH); the first "01" is the ID; the second "01" is the length of data volume; "13" is the gripper opening reading instruction (MD_MC_READ_EG_PARA); "15" is the checksum, i.e., low-order bytes ((B2+B3+...+B4) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5—B8 (4 bytes)	Checksum B15 (1 byte)
0xEE	0x16	0x01	5 (0x05)	0x13	0xE8 0x03 0x70 0x00	0x74

The specific response frame received is "EE 16 01 05 13 E8 03 70 00 74", where "EE 16" is the response frame header; "01" is the ID; "05" is the length of data volume; "13" is the instruction type; "E8--00" is the gripper opening read by the MCU (0x03E8=1000 indicates the maximum opening (1000); 0x0070=112 indicates the minimum opening (112)); "74" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B8) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.10 Current opening reading

Function: The MCU reads the current opening of a gripper.

Length of instruction frame: 6 bytes

Instruction type: 0xD9 (CMD_MC_READ_ACTPOS)

Data content: None

For example, when the MCU reads the current opening (data range: 0-1000; dimensionless unit) of a gripper (ID=1), the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content -- (0 bytes)	Checksum B5 (1 byte)
0xEB	0x90	0x01	0x01	0xD9		0xDB

The instruction frame sent is "EB 90 01 01 D9 DB", where "EB 90" is the frame header (FH); the first "01" is the ID; the second "01" is the length of data volume; "D9" is the gripper opening reading instruction; "DB" is the checksum, i.e., low-order bytes ((B2+B3+...+B4) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5—B6 (4 bytes)	Checksum B7 (1 byte)
0xEE	0x16	0x01	0x03	0xD9	0xE8 0x03	0x74

The specific response frame received is "EE 16 01 03 D9 F1 01 CF", where "EE 16" is the response frame header; "01" is the ID; "05" is the length of data volume; "D9" is the instruction type; "F1-01" is the current gripper opening read by the MCU (0x01F1; integer: 497 converted from 0x01F1); "CF" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B7) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.11 Read the gripper running status.

Function: By reading the gripper running status, the MCU can acquire the current opening, set value of grip force, running status, error code, and driver temperature value. This instruction can be used to periodically acquire the running status messages of grippers,

and is helpful for the user to quickly identify the gripper status.

Length of instruction frame: 6 bytes

Instruction type: 0x41 (CMD_MC_READ_EG_RUNSTATE)

Data content: None

For example, when the MCU reads the current status of a gripper (ID=1), the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content -- (0 byte)	Checksum B5 (1 byte)
0xEB	0x90	0x01	1 (0x01)	0x41		0x43

The instruction frame sent is "EB 90 01 01 41 43", where "EB 90" is the frame header (FH); the first "01" is the ID; the second "01" is the length of data volume; "0x41" is the gripper running status reading instruction (CMD_MC_READ_EG_RUNSTATE); "0x43" is the checksum, i.e., low-order bytes ((B2+B3+...+B4) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 B)		ID B2 (1 B)	Length of data volume B3 (1 B)	Instruction type B4 (1 B)	Data content B5-B11					Checksum B12 (1 B)
					Running status code B5 (1 B)	Error code B6 (1 B)	Temperature B7 (1 B)	Opening B8-9 (2 B)	Grip force setting B10-11 (2 B)	
0xEE	0x16	0x01	8 (0x07)	0x41	0x01	0x00	0x23	0xE9 0x03	0x64 0x00	0xBD

Bytes of the returned frame received: EE 16 01 08 41 01 00 23 E8 03 64 00 BD

Meaning of running status codes:

0x01	The gripper has unclenched to its maximum opening and is in the idle state.
0x02	The gripper has clenched to its minimum opening and is in the idle state.
0x03	The gripper stops and is in the idle state.
0x04	The gripper is clenching.

0x05	The gripper is unclenching.
0x06	The gripper stops due to force control during clenching.

Meaning of each error code:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Preserved	Preserved	Preserved	Internal communication failure	Driver running fault	Over-current failure	Over temperature failure	Locked-rotor failure

The temperature unit is ℃. For example, 0x23 indicates 35℃.

Opening: 0x03E8 indicates that the current gripper opening is 1000 (dimensionless unit).

The grip force unit is g. For example, 0x0064 indicates that the current set value of grip force is 100 g.

2.4.12 Fault clearance

Function: For the faults indicated by bit0, bit2, bit3 and bit4 among the gripper error codes, the fault clearance command can be used to restore the gripper to normal operation. For the over temperature fault (temperature above 80℃) indicated by bit1, the gripper will not resume normal operation until the temperature falls (below 60℃). If a fault appears again after the fault clearance command has been executed, it is implied that the product fault cannot be cleared. This fault persists and requires repair and troubleshooting by an engineer.

Length of instruction frame: 6 bytes

Instruction type: 0x17 (CMD_MC_ERROR_CLR)

Data content: None

For example, when the MCU sets a gripper (ID=1) for fault clearance, the following instruction frame should be sent.

Frame header (FH) B0-B1 (2 bytes)	ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content (0 byte)	Checksum B5 (1 byte)

0xEB	0x90	0x01	1 (0x01)	0x17	None	0x19

"EB 90" is the frame header (FH); the first "01" is the ID; the second "01" is the length of data volume; "17" is the reference point calibration instruction (CMD_MC_ERROR_CLR); "0x19" is the checksum, i.e., low-order bytes ((B2+B3+...+B4) &0xFF) of the sum of all bytes before checksum in the instruction frame except the frame header. When this instruction frame is sent to the gripper, the response frame returned is shown below.

Frame header (FH) B0-B1 (2 bytes)		ID B2 (1 byte)	Length of data volume B3 (1 byte)	Instruction type B4 (1 byte)	Data content B5 (12 bytes)	Checksum B6 (1 byte)
0xEE	0x16	0x01	2 (0x02)	0x17	0x01	0x1B

"EE 16" is the response frame header; the first "01" is the ID; "02" is the length of data volume; "17" is the instruction type; the second "01" is the successful instruction reception flag ("55" indicates an exception, and re-sending is required); "0x1B" is the last byte and indicates the checksum, i.e., low-order bytes ((B2+B3+...+B5) &0xFF) of the sum of all data before checksum except the response frame header.

2.4.13 Instruction for compatibility with the dexterous hand

Function: The control instruction protocol for the electric gripper is also compatible with some control instruction protocols for our dexterous hand. The user can use the dexterous hand control instructions to manipulate and control the gripper. This enables the user to quickly experience and apply our different products.

S/N.	Instruction type	Function description	Values Hex	Length of data content Bytes
1	CMD_MC_SET_DRVALL_SEEKANGLE_GYH	Set the normalized target position.	0x54	12
3	CMD_MC_SET_DRVALL_SPEED	Set the speed.	0x51	12
4	CMD_MC_SET_DRVALL_YBP_THRESHOLD	Set the grip force threshold.	0x52	10
5	CMD_MC_READ_DRVALL_SEEKGYHANGLE	Read the normalized target position.	0xD7	0
7	CMD_MC_READ_DRVALL_CURA	Read the current	0xD9	0

S/N.	Instruction type	Function description	Values Hex	Length of data content Bytes
	NGLE_GYH	normalized position.		
9	CMD_MC_READ_DRVALL_SPEED	Read the current speed.	0xD2	0
10	CMD_MC_READ_DRVALL_YBP_THRESHOLD	Read the grip force threshold.	0xD3	0
13	CMD_MC_READ_HAND_STATUS	Read the status information on the dexterous hand.	0xDC	0

3 MODBUS RTU Communication Protocol

16-bit reverse cyclic redundancy check (CRC16-REV = 0xA001)

Supported function codes: 0x03 (read register); 0x06 (write to single register); 0x10 (write to multiple registers)

Definition of registers:

Address	Name	Range	Description		
01	CMD_SAVE	0/1	1: Parameter saving 0: Null	Read and write	With immediate effect
02	CMD_SETDEFAULT	0/1	1: Parameter reset to the default parameter 0:	Read and write	With immediate effect
03	CMD_ID	0-254	Device address (ID) of the gripper	Read and write	With immediate effect
04	CMD_BAUD	0,1, 2	Baud rate setting 0: 115200 1: 57600 2: 19200	Read and write	After it is saved, the setting will become effective after re-power on.
05	CMD_CATCH_MOD	0/1	0: One grasping action (the gripper will stop moving after it reaches the target opening or grasps an object.) 1: Continuous grasping actions (after the gripper grasps an object and stops moving, if the force applied to the gripper decreases, grasping will continue to maintain the grip force.)	Read and write	With immediate effect
06	CMD_STOP	0/1	1: Emergency stop 0:	Read and write	With immediate effect
07	CMD_FAULTACK	0/1	1: Fault clearance 0:	Read and write	With immediate effect
10	CMD_OPENLEN_SET	0-1000	Gripper opening setting: 1000 (maximum opening), 0 (minimum opening); after this register is set, the	Read and write	With immediate effect

			gripper will take an immediate action.		
11	CMD_SPEED_SET	10-1000	Gripper speed setting: 1000 (maximum running speed); after this register is set, the gripper will take no action.	Read and write	With immediate effect
12	CMD_FORCE_SET	100-1000	Grip force setting of the gripper: 1000 (maximum grip force); after this register is set, the gripper will take no action.	Read and write	With immediate effect
16	CMD_MAX_OPENLEN	0-1000	Maximum opening	Read and write	With immediate effect
17	CMD_MIN_OPENLEN	0-1000	Minimum opening	Read and write	With immediate effect
61	CMD_OPENLEN_ACT	0-1000	Actual opening of the gripper	Read only	
62	CMD_CURRENT	0-2000	Actual current of the gripper	Read only	
63	CMD_TEMP	0-100	Gripper driver temperature	Read only	
64	CMD_ERRORCODE		Error codes of the gripper bit0: Locked-rotor error bit1: Over temperature error bit2: Over-current error bit3: Running fault bit4: Internal communication failure	Read only	
65	CMD_STATUS		Gripper status code 1: The gripper has unclenched to its maximum opening and has stopped. 2: The gripper has clenched to its minimum opening and has stopped. 3: The gripper stops. 4: The gripper is grasping. 5: The gripper is unclenching. 6: The gripper has grasped an object during grasping and has stopped.	Read only	

Example 1: The following instruction frames are used to read the opening (61), current

(62), temperature (63), and error codes (64) of a gripper.

		Example	
		Hexadecimal number	Decimal number
byte[0]	Gripper ID	01	1
byte[1]	Function code	03	3
byte[2]	High-order 8 bits in the register address	00	61
byte[3]	Low-order 8 bits in the register address	3D	
byte[4]	High-order 8 bits in the number of registers	00	4
byte[5]	Low-order 8 bits in the number of registers	04	
byte[6]	CRC code	D5	
byte[7]	CRC code	C5	

Gripper response:

		Example	
		Hexadecimal number	Decimal number
byte[0]	Gripper ID	01	1
byte[1]	Function code	03	3
byte[2]	Byte	08	8
byte[3]	High-order 8 bits in the register (61) content	00	0
byte[4]	Low-order 8 bits in the register (61) content	00	
byte[5]	High-order 8 bits in the register (62) content	FF	-2
byte[6]	Low-order 8 bits in the register (62) content	FE	
byte[7]	High-order 8 bits in the register (63) content	00	37
byte[8]	Low-order 8 bits in the register (63) content	25	
byte[9]	High-order 8 bits in the register (64) content	00	0
byte[10]	Low-order 8 bits in the register (64) content	00	
byte[11]	CRC code	C7	
byte[12]	CRC code	B9	

Example 2: The following instruction frames can be used to set the gripper opening (1010) to 0.

		Example

		Hexadecimal number	Decimal number
byte[0]	Gripper ID	01	1
byte[1]	Function code	06	6
byte[2]	High-order 8 bits in the register address	00	10
byte[3]	Low-order 8 bits in the register address	0A	
byte[4]	High-order 8 bits in the set value	00	0
byte[5]	Low-order 8 bits in the set value	00	
byte[6]	CRC code	A9	
byte[7]	CRC code	C8	

Gripper response:

		Example	
		Hexadecimal number	Decimal number
byte[0]	Gripper ID	01	1
byte[1]	Function code	06	6
byte[2]	High-order 8 bits in the register address	00	10
byte[3]	Low-order 8 bits in the register address	0A	
byte[4]	High-order 8 bits in the set value	00	0
byte[5]	Low-order 8 bits in the set value	00	
byte[6]	CRC code	A9	
byte[7]	CRC code	C8	

Example 3: The following instruction frames can be used to set the gripper opening (10) to 0, the speed (11) to 1000, and the grip force (12) to 1000.

		Example	
		Hexadecimal number	Decimal number
byte[0]	Gripper ID	01	1
byte[1]	Function code	10	16
byte[2]	High-order 8 bits in the register address	00	10
byte[3]	Low-order 8 bits in the register address	0A	
byte[4]	High-order 8 bits in the number of registers	00	3

byte[5]	Low-order 8 bits in the number of registers	03	
byte[6]	Byte	06	6
byte[7]	High-order 8 bits in the value to be set by the register (10)	00	0
byte[8]	Low-order 8 bits in the value to be set by the register (10)	00	
byte[9]	High-order 8 bits in the value to be set by the register (11)	03	1000
byte[10]	Low-order 8 bits in the value to be set by the register (11)	E8	
byte[11]	High-order 8 bits in the value to be set by the register (12)	03	1000
byte[12]	Low-order 8 bits in the value to be set by the register (12)	E8	
byte[13]	CRC code	46	
byte[14]	CRC code	6E	

Gripper response:

		Example	
		Hexadecimal number	Decimal number
byte[0]	Gripper ID	01	1
byte[1]	Function code	10	16
byte[2]	High-order 8 bits in the register address	00	10
byte[3]	Low-order 8 bits in the register address	0A	
byte[4]	High-order 8 bits in the number of registers	00	3
byte[5]	Low-order 8 bits in the number of registers	03	
byte[6]	CRC code	A0	
byte[7]	CRC code	0A	