HIWIN®

Electric Gripper

Modbus Operation Manual

Original Instruction

GOOD DESIGN AWARD 2017 TAIWAN EXCELLENCE 2017 CE RoHS

www.hiwin.tw

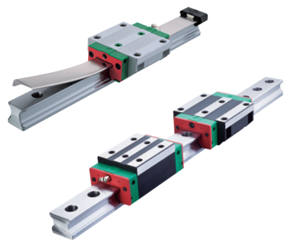
HIWIN® INDUSTRIE 4.0 Best Partner

Multi-Axis Robot
Pick-and-Place / Assembly /
Array and Packaging / Semiconductor /
Electro-Optical Industry /
Automotive Industry / Food Industry
- Articulated Robot
- Delta Robot
- SCARA Robot
- Wafer Robot
- Electric Gripper
- Integrated Electric Gripper
- Rotary Joint

Single-Axis Robot
Precision / Semiconductor /
Medical / FPD
- KK, SK
- KS, KA
- KU, KE, KC

Torque Motor Rotary Table
Medical / Automotive Industry /
Machine Tools / Machinery Industry
- RAB Series
- RAS Series
- RCV Series
- RCH Series

Ballscrew
Precision Ground / Rolled
- Super S Series
- Super T Series
- Mini Roller
- Ecological & Economical Lubrication Module E2
- Rotating Nut (R1)
- Energy-Saving & Thermal-Controlling (Cool Type)
- Heavy Load Series (RD)
- Ball Spline

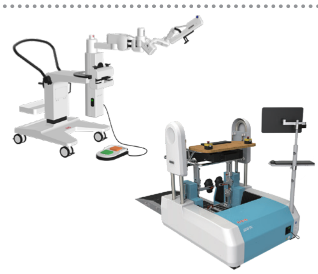


Linear Guideway
Automation / Semiconductor / Medical
- Ball Type--HG, EG, WE, MG, CG
- Quiet Type--QH, QE, QW, QR
- Other--RG, E2, PG, SE, RC

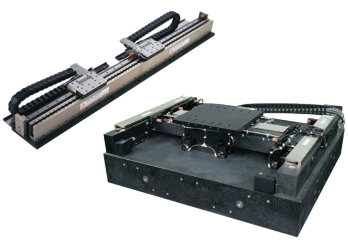
Bearing
Machine Tools / Robot
- Crossed Roller Bearing
- Ballscrew Bearing
- Linear Bearing
- Support Unit

DATORKER® Robot Reducer
Robot / Automation Equipment /
Semiconductor Equipment / Machine Tools
- WUT-PO Type
- WUI-CO Type
- WTI-PH Type
- WTI-AH Type

AC Servo Motor & Drive
Semiconductor / Packaging Machine
/ SMT / Food Industry / LCD
- Drives--D1, D2T/D2T-LM, E1
- Motors--50W~2000W



Medical Equipment
Hospital / Rehabilitation Centers /
Nursing Homes
- Robotic Gait Training System
- Robotic Endoscope Holder



Linear Motor
Automated Transport /
AOI Application / Precision /
Semiconductor
- Iron-core Linear Motor
- Coreless Linear Motor
- Linear Turbo Motor LMT
- Planar Servo Motor
- Air Bearing Platform
- X-Y Stage
- Gantry Systems

Torque Motor & Direct Drive Motor
Machine Tools
- Torque Motor--
  TM-2/IM-2, TMRW Series

Inspection / Testing Equipment / Robot
- Direct Drive Motor--
  DMS, DMY, DMN Series

Contents

1. Introduction 3
2. Safety Statement 3
3. System Configuration 4
3.1. External Wiring 4
3.2. Hardware Configuration 5
3.3. Communication Specification 7
4. Data Allocation Table 8
4.1. Read IO 9
4.2. Read Information 9
4.3. Read/Write Type 10
4.4. Read/Write Command 10
4.4.1. Indirect Command 10
4.4.2. Direct Command 11
5. Message Structure 13
5.1. Message Structure 13
5.2. Function Code 15
5.2.1. Read Discrete Inputs (0x02) 15
5.2.2. Read Holding Registers (0x03) 16
5.2.3. Read Input Registers (0x04) 17
5.2.4. Read Exception Status (0x07) 18
5.2.5. Write Multiple Registers (0x10) 19
5.3. Response 20
5.3.1. Normal Response 20
5.3.2. No Response 20
5.3.3. Exception Response 20
6. Error Message 22
Appendix 1: Quick Start 24
Appendix 2: CRC-16 Calculation 27

Introduction

HIWIN's XEG Series enable users to flexibly operate the electric gripper for all kinds of automation application via RS485 interface and Modbus-RTU protocol. These protocols are especially used with a Programmable Logic Controller (PLC) host for automation control through the corresponding interface. A user can refer to the application methods specified in the operation manual, simultaneously connecting up to 15 controllers and electric grippers. Operations like read/write gripper's model number, initialize gripper, move gripper, grip object, identify object and monitor gripper, are carried out by sending a command to the respective controller number (Unicast Mode) or by broadcasting (Broadcast Mode).

1. Safety Statement

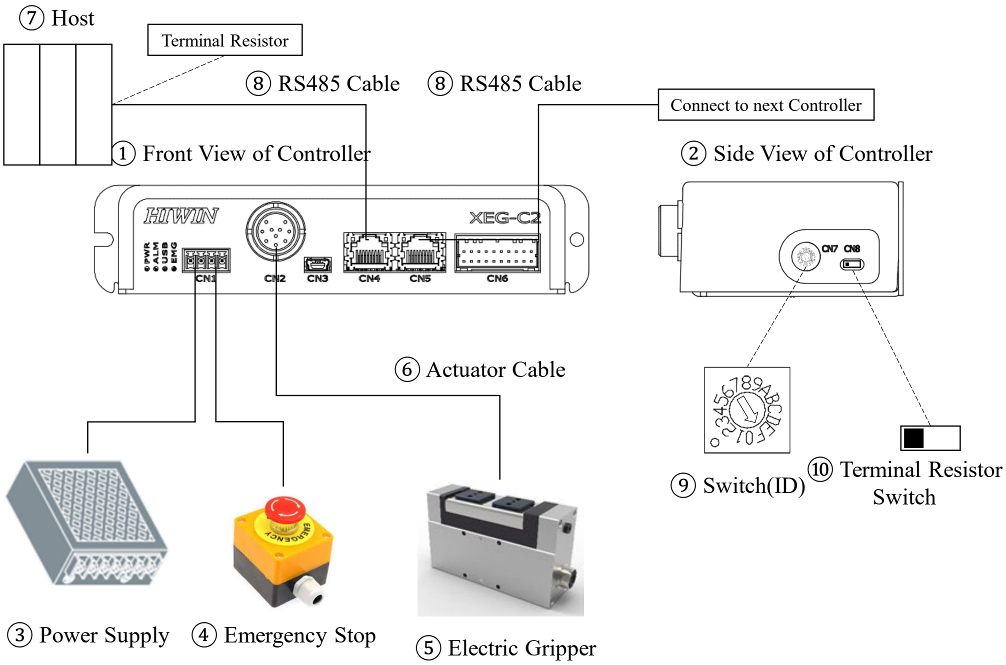
The safety statement in the manual allows a user to correctly operate the product and avoid injury and damage. Please read the manual prior to operation, and strictly obey the relevant regulations to ensure your safety.

2. System Configuration

2.1. External Wiring

- The hardware wiring map comprises the following parts:

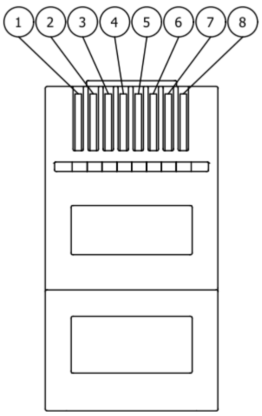
No.	Item Name	Description
1	Controller's front	XEG-C2, including the required connection components.
2	Controller's side	XEG-C2, including the required switches.
3	Power Supply	DC24V, 0.5A
4	Emergency Stop Switch	Closed when operation is safe, the controller won't work when the emergency circuit is opened.
5	Electric Gripper	HIWIN XEG-Series Electric Gripper.
6	Actuator Cable	Connecting the electric gripper to the controller (CN2).
7	Host System	Host with Modbus-RTU protocol and interface, such as: PC, PLC, HMI.
8	RS485 Cable	The cable connect to the Host and the controller (CN4); The RJ45 interface (CN5) can be used to cascade the next controller without limiting the connection.
9	Controller ID Switch	Set controller's Modbus ID, 0 can only be used to broadcast mode (default as 1).Cascaded controllers can't be set as the same ID.
10	Terminal Resistor Switch	Terminal switch, which is Off by default. When the total cascade length is greater than 10 m, the last controller's Terminal Resistor Switch must be set as On.



External Wiring and Setting of Modbus Hardware

2.2. Hardware Configuration

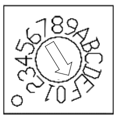
- CN4 and CN5 sockets are those for RJ-45. However, they can't be connected to the network interface because the electric characteristics are different from the protocol. The pins are defined as follows:



Pin	Signal Name	Color
1	N/A	White/Orange
2	GND	Orange
3	D+	White/Green
4	N/A	Blue
5	GND	White/Blue
6	D-	Green
7	N/A	White/Brown
8	N/A	Brown

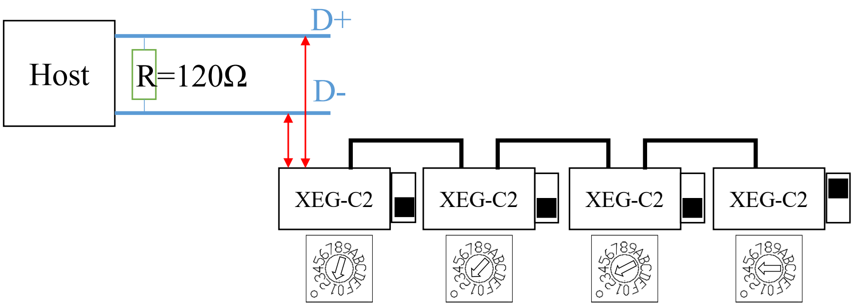
Definition of RS485 Pin

- 15 IDs (1~F) can be set, 0 is reserved for Broadcast. The IDs can't be repeated in a single cascade network. The power must be reset after the ID is set. The ID is set as 15 (F) in the following example:



Example of setting ID

- The example of cascading the cables is as follows:
  (The total cascade length must be less than 50 m. The last one is used to start the terminal resistor.)



Slave No.	1	2	3	4
Terminal Resistor	Off	Off	Off	On

2.3. Communication Specification

Electric Characteristics	Based on EIA-485, straight cable, using twisted-pair cable (recommended TIA/EIA CAT5e or more.)
Communication Interface	Half-duplex RS-485
Starting Bit	1 bit
Data Length	8 bits
Parity Check	None
Stop Bit	1 bit
Protocol	Modbus-RTU
Baud Rate	115200 bps
Error Check	CRC-16
Quantity Limit	15 sets
Length Limit	50 m

3. Data Allocation Table

The Controller refers to the Modbus specification, memory is divided into four types:

No.	Register Type	Pattern	Read/Write Limit	Support by Controller
1	Discrete Input	Single bit	Only Read	Supported
2	Coils	Single bit	Read/Write	Not supported
3	Input Registers	16-bit word	Only Read	Supported
4	Holding Registers	16-bit word	Read/Write	Supported

You can refer to the following tables when reading/writing all data names, including the code commands, the starting address and value range. For reading/writing the single data name, please refer to the following tables.

Register Type	Data Name	Code Command	Starting Address	Range	Description
Discrete Input	Input bits	02h	0000h	0-1	DI status (see 4.1)
	Output bits		0010h	0-1	DO status (see 4.1)
Input Registers	Position	04h	0300h	0-6400	Gripper position (unit: 0.01mm)
	State		0301h	0-7	Gripper status (see 4.2)
	Firmware Version		0303h	0-99	Firmware version (see 4.2)
Holding Registers	EG Type	03h 10h	0600h	0-2660	Gripper Model (see 4.3)
	Trigger Motion Data		0601h	1-63	Indirect Control - trigger to control motion data (see 4.4.1)
	Reset		0610h	0-1	Direct Control - way to directly control (see 4.4.2)
	Stop		0620h	0-1	Direct Control - way to directly control (see 4.4.2)
	Absolute Move		0630h	0-10000	Direct Control - way to directly control (see 4.4.2)
	Relative Grip		0640h	0-10000	Direct Control - way to directly control (see 4.4.2)

3.1. Read IO

The following are the inputs address for the electric gripper:

Input Bits (0000h~0007h)								
Address	0000h	0001h	0002h	0003h	0004h	0005h	0006h	0007h
DI	IN1	IN2	IN3	IN4	IN5	IN6	START	-

The following are the outputs address for the electric gripper:

Output Bits (0000h~0007h)								
Address	0010h	0011h	0012h	0013h	0014h	0015h	0016h	0017h
DO	POS	HOLD	BUSY	ALM1	ALM2	CHK1	CHK2	CHK3

3.2. Read Information

The following are the meanings for address 0301h (Gripper Status).

Gripper Status (0301h)			
Value	**Description**	**Value**	**Description**
0	Idle	4	Position Failure (Alarm)
1	Working (Busy)	5	Move (Alarm)
2	Position (Pos)	6	Home Reset Failure (Alarm)
3	Hold (Hold)	7	Emergency Stop (Alarm)

The following are the meanings for the firmware version of address 0303h to 0306h.

Example: V2.0.20.535

Firmware Version (0303h~0306h)	
Address	**Value**
0303h	2
0304h	0
0305h	20
0306h	535

3.3. Read/Write Type

The following are the meanings for the Gripper Model of address 0600h, where 0A represents XEG Series.

Gripper Model (0600h)	
Value	**Gripper Model**
0x0A10	XEG-16
0x0A20	XEG-32
0x0A21	XEG-32-PR
0x0A30	XEG-48
0x0A40	XEG-64

3.4. Read/Write Command

The Read/Write Commands include: Indirect Command and Direct Command. Indirect Commands have the advantages of less communication data and quicker speed while Direct Commands have more motion flexibility. When the Controller receives the direct command from the Host, some registers will reset their value. The following tables list the items that will automatically reset.

3.4.1. Indirect Command

Indirect Commands trigger movements which have already been set-up in the Gripper's Data using the gripper software. For example, if you want to trigger the indirect motion for Data10, 10 must be written to address 0601h.

Motion Data Trigger (0601h)		
Value Range	**Initial Value**	**Auto Reset**
0-63	0	O

3.4.2. Direct Command

Before carrying out Move and Expert commands, user should firstly reset the gripper. This command needs to be carried out when resetting the gripper only.

Reset (0610h)				
Address	**Name**	**Value Range**	**Initial Value**	**Auto Reset**
0610h	Home Reset	0-1	0	O

When carrying out any motion, you can end it by sending the Stop command.

Stop (0620h)				
Address	**Name**	**Vale Range**	**Initial Value**	**Auto Reset**
0620h	Stop	0-1	0	O

Users can make the gripper move quickly via the Move command. The Gripper will move the absolute position without holding, when positioned, the Gripper status will be changed accordingly, if there are obstacles in the process, an alarm will be output.

Move (0630h~0632h)				
Address	**Name**	**Value Range**	**Initial Value**	**Auto Reset**
0630h	Move	0~64.00 mm (by type)	0	X
0631h	Speed	0~100.00 mm/s (by type)	0	X
0632h	Flag	0-1	0	O

Users can quickly move and slowly hold using the Expert command. The stroke input is referred as relative value. Gripper's status and outputs will change accordingly when holding an object; if the stroke exceeds the gripping boundary, an alarm will be output. The minimum unit of the stroke and speed is 0.01. **Example: if moving 8 mm, users must send 800.**

Expert (0640h~0646h)				
Address	**Name**	**Value Range**	**Initial Value**	**Auto Reset**
0640h	Direction	0-1 (OFF/ON)	0	X
0641h	Move	0~64.00 mm (by type)	0	X

0642h	Speed	0~100.00 mm/s (by type)	0	X
0643h	Holding stroke	0~64.00 mm (by type)	0	X
0644h	Holding speed	0~20.00 mm/s (by type)	0	X
0645h	Holding force	40~100% (by type)	0	X
0646h	Flag	0-1	0	O

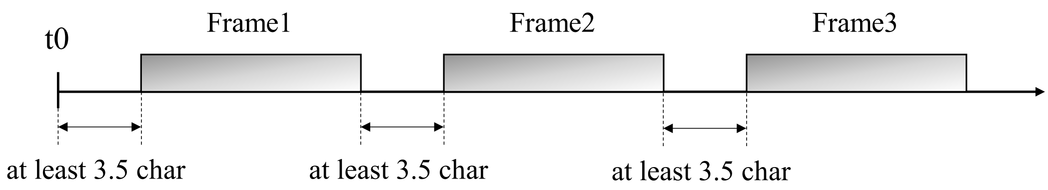
All electric grippers moving range is as follows:

Gripper Model	Move	Speed	Holding Stroke	Holding Speed	Holding Force
FXEG-16	0~16 (mm)	0~60 (mm/s)	0~16 (mm)	0~10 (mm/s)	50~100%
XEG-32	0~32 (mm)	0~80 (mm/s)	0~32 (mm)	0~20 (mm/s)	40~100%
XEG-32-PR	0~32 (mm)	0~60 (mm/s)	0~32 (mm)	0~10 (mm/s)	50~100%
XEG-48	0~48 (mm)	0~80 (mm/s)	0~48 (mm)	0~20 (mm/s)	50~100%
XEG-64	0~64 (mm)	0~100 (mm/s)	0~64 (mm)	0~20 (mm/s)	40~100%

4. Message Structure

4.1. Message Structure

According to the Modbus specification, there should be more than 3.5 characters between the Message Structure and the Message Structure, equivalent to (0.3 ms) or more:



In the RTU mode, the Message Structure in a complete Master Inquiry is as follows:

No.	ID	Function Code (FC)	Data	Cyclic Redundancy Check (CRC-16)
Byte Count	1	1	N	2

When the ID is not 0, the ID controller should exist in the cascaded controllers, responding to the Message Structure of Slave Response as follows. There are some differences according to Read or Write:

No.	ID	Function Code (FC)	Data	Cyclic Redundancy Check (CRC-16)
Byte Count	1	1	N	2

- ID and Inquiry Mode
  - Unicast Mode

    The Host sends an inquiry to one controller; the controller processes, and responds; the ID is set as 1~F without repeating in the same ID in any two controllers.

  - Broadcast Mode

The slave address appointed by the Host is set as 0, which can send an inquiry to all controllers; the controller processes, but doesn't respond.

- Function Code (FC)

Function Code	Description	Applicable Register Type
02H	Read multiple bit data	Discrete Input
04H	Read multiple word data	Input Registers
03H	Read multiple word data	Holding Registers
10H	Write multiple word data	
07H	Read failure status	N/A

- Data
  Set the data related to the function code, where the data length will be different according to the respective function code.

- Cyclic Redundancy Check (CRC-16)
  The Controller will calculate the received messages, and compare them with CRC; if the calculation value is consistent with cyclic redundancy check, it will be determined as correct message. For CRC-16 calculation, please refer to appendix.

4.2. Function Code

4.2.1. Read Discrete Inputs (0x02)

- Description
  Read multiple BIT data depending on the value WORD starting from the Starting Address. Broadcast mode isn't supported.

- Master Inquiry
  - Message Structure

Slave Address	Function Code	Starting Address High	Starting Address Low	Number Inputs High	Number Inputs Low	CRC	CRC
Byte	Byte	Word		Word		Word	

  - Example

From the Controller ID as 2 and the starting address as 0000h, read 8 values of Discrete Inputs. 02 02 00 00 00 08 CRC CRC

- Slave Response
  - Message Structure

Slave Address	Function Code	Byte Count	Data	…	CRC	CRC
Byte	Byte	Byte	Byte	Byte	Word	

  - Example

The return value 22 hex is 0010 0010 binary. The addresses 0001h (HOLD) and 0005h (CHK1) are On, and the rest are Off; From left to right, 0000h to 0007h is: Off-On-Off-Off-Off-On-Off-Off. 02 02 01 22 CRC CRC

4.2.2. Read Holding Registers (0x03)

- Description

  Read the WORD data in WORD from reading the starting address. Broadcast mode isn't supported.

- Master Inquiry
  - Message Structure

Slave Address	Function Code	Starting Address High	Starting Address Low	Number Inputs High	Number Inputs Low	CRC	CRC
Byte	Byte	Word		Word		Word	

  - Example

From the Controller with ID 2, and starting address 0600h, read one value of Holding Register.
02 03 06 00 00 01 CRC CRC

- Slave Response
  - Message Structure

Slave Address	Function Code	Byte Count	Data High	Data Low	…	CRC	CRC
Byte	Byte	Byte	Word		Word	Word	

  - Example

The return value 0A 20 hex represents type 32 Electric Gripper in XEG Series.
02 03 02 0A 20 CRC CRC

4.2.3. Read Input Registers (0x04)

- Description
  The format is the same as 0x03. Read the WORD data in WORD from reading the starting address. The Broadcast mode isn't supported.

- Master Inquiry
  - Message Structure

Slave Address	Function Code	Starting Address High	Starting Address Low	Number Inputs High	Number Inputs Low	CRC	CRC
Byte	Byte	Word		Word		Word	

  - Example

From the Controller with ID 2 and starting address 0300h, read two values of Input Register. 02 04 03 00 00 02 CRC CRC

- Slave Response
  - Message Structure

Slave Address	Function Code	Byte Count	Data High	Data Low	…	CRC	CRC
Byte	Byte	Byte	Word		Word	Word	

  - Example

The return value includes 07 D0 hex, as well as 00 02 hex is 2000 Decimal and 2 Decimal, representing the gripper position as 2000 (the minimum unit of 0.01mm) and the gripper status as "Pos", respectively. 02 04 04 07 D0 00 02 CRC CRC

4.2.4. Read Exception Status (0x07)

- Description
  When exception code 04H occurs, further read the exception status code.

- Master Inquiry
  - Message Structure

Slave Address	Function Code	CRC	CRC
Byte	Byte	Word	

  - Example

From the controller ID as 2, read the exception status code.
02 07 CRC CRC

- Slave Response
  - Message Structure

Slave Address	Function Code	Exception Status	CRC	CRC
Byte	Byte	Byte	Word	

  - Example

The return value 02 hex represents the detailed reason for the exception code. The exception status is emergency stop.
02 07 02 CRC CRC

Exception Status Code	Value	Remark
_EXCEPTION_STATUS_IDLE	00H	No error
_EXCEPTION_STATUS_ENN_POWER	02H	Emergency stop
_EXCEPTION_STATUS_ADDRESS_FAIL	03H	No command for this address
_EXCEPTION_STATUS_EG_TYPE_NULL	04H	Incorrect gripper type
_EXCEPTION_STATUS_RESET_FAIL	05H	Incorrect Reset
_EXCEPTION_STATUS_STOP_FAIL	06H	Incorrect Stop
_EXCEPTION_STATUS_ABS_MOVE_NULL	07H	Incorrect Move
_EXCEPTION_STATUS_RELATIVE_GRIP_NULL	08H	Incorrect Expert
_EXCEPTION_STATUS_MOVE_ERROR	10H	Move error
_EXCEPTION_STATUS_OVER_ERROR	11H	Move error

_EXCEPTION_STATUS_RESET_ERROR	12H	Reset error

4.2.5. Write Multiple Registers (0x10)

- Description
  Write the WORD data in WORD at the appointed address. Broadcast mode is supported.

- Master Inquiry
  - Message Structure

Slave Address	Function Code	Starting Address High	Starting Address Low	Number Address High	Number Address Low	Byte Count	Data	...	CRC	CRC
Byte	Byte	Word		Word		Byte	Word	Word	Word	

  - Example

Write the multiple motion values to the register address 0640h~0646h for the controller with ID 2.
The values are Direction (0)- Move (1000)- Speed (6000)- ff (500)- Holding Speed (1000)- Holding Force (100)- Flag (1).

- Except for the direction, holding force and flag, the rest of parameter unit is 0.01 (unit).

02 10 06 40 00 07 0E
00 00 03 E8 17 70 01 F4 03 E8 00 64 00 01 CRC CRC

- Slave Response
  - Message Structure

Slave Address	Function Code	Starting Address High	Starting Address Low	Number Address High	Number Address Low	CRC	CRC
Byte	Byte	Word		Word		Word	

  - Example

Returning the Message Structure only includes Starting Address and the sent WORD quantity. Please note the gripper status is “not working” and “inwardly holding in the move” while sending. Otherwise, the move failure will take place. At this time, the gripper will start to inwardly hold and automatically reset the value in the register address 0646h as 0, waiting for next trigger.
02 10 06 40 00 07 CRC CRC

4.3. Response

Controller's responses include: Normal Response, No Response and Exception Response.

4.3.1. Normal Response

When an inquiry is received from the Host, the Controller will process and reply with the response corresponding to the respective function code.

4.3.2. No Response

Although the Host sends an inquiry, the Controller will not sometimes reply to the response. This status is referred as No Response. There are two reasons No Response happens:

- Sending Failure
  When the Controller detects the sending failure below, it will abandon the inquiry and show no response.

	Reason	Content
1	Bit frame error	Stop bit detected
2	Parity error	Parity setting error
3	CRC not consistent	CRC-16 calculation different from CRC
4	Excessive message length	Message length over 256 bytes

- Non-sending Failure
  Even though a sending failure isn't detected, No Response might also sometimes occur in the following situations.

	Reason	Content
1	Broadcast mode	The ID for the Host appointed as 0 is in the Broadcast mode. The Controller will process, but show no response.
2	Slave ID not existed	The ID appointed for the Host doesn't exist in the controller cascade. No one responds.

4.3.3. Exception Response

When the Controller can't process the inquiry, the Exception Response will be replied back. The reasons why it can't be processed will be additionally displayed in Response, which is the exception code. This Error Message will be introduced in details in the next chapter.

5. Error Message

- Description

  When an error state is detected, exception made for the response condition, an exception code corresponding to the error type will be sent back.

  The message structure of a complete Master Inquiry for the Host is as follows:

No.	ID	Function Code (FC)	Data	Cyclic Redundancy Check (CRC-16)
Byte Count	1	1	N	2

When an error takes place, the message structure in the Slave Response is as follows:

No.	ID	Function Code (FC+80H)	Exception Code	Cyclic Redundancy Check (CRC-16)
Byte Count	1	1	1	2

- Exception Code

  Display the reason that can't be processed.

Exception Code	Reason	Content
01H	Incorrect function	Function code can't be identified (please double check Function Code.)
02H	Incorrect data address	Register address can't be identified (please refer to the Data Allocation Table.) ● Read/write an address that doesn't exist.
03H	Incorrect data	Data can't be executed (please refer to the specification of C2 Manual.) ● The register quantity is 0. ● The bytes are not the value for the register x2. ● The motion value is not in the specification of electric gripper (by type). ● Read/write the data length over the range

04H	Incorrect controller	An error that can't be recovered takes place when processing. (Please inquire the cyclic redundancy check via 0x07.)

- Example of Inquiry and Response
  - Master Inquiry

Write the value 1 to the register address 0611h for the controller ID as 2.
02 06 06 11 00 01 CRC CRC

  - Slave Response

The response code is 06h+80h due to an incorrect exception. Since 0611h is an address interval that doesn't exist, the response exception code is 02h.
02 86 02 CRC CRC

Appendix 1: Quick Start

In order to accelerate user familiarity with Modbus in the initial phase of operation or testing, this appendix provides a basic scenario for the application of the Electric Gripper and compares the protocol handshake, allowing users to verify the quality of the system communication and quickly controlling the operation know-how.

Basic Scenario of Electric Gripper (taking XEG-32 as example):

- Please pay attention and refer to the System Configuration. The Controller ID is set as 2, cascading a single controller:

  For the protocol, please refer to the Data Allocation Table and the Message Structure.
- Write gripper model: *XEG-32*
- Read gripper model
- Read firmware version: *V3.0.1.884*
- Write reset
- Read gripper status: *Working*
- Wait for 30 seconds
- Read gripper status and position: *Idle, 32.00 mm*
- Write Expert: *inwardly move 10mm, 80mm/s, hold 5mm, 20mm/s, 100%*
- Read gripper status: *Working*
- Wait for three seconds
- Read gripper status and position: *Idle, 17.00 mm*
- Broadcast to write Move: *32mm, 80mm/s*
- Read gripper status: *Working*
- Wait for three seconds
- Read gripper status and position: *positioned, 32.00 mm*

Basic Scenario of Gripper and Protocol:

Definition of Communication Direction:

- Host inquires Controller: →
- Control responds to Host: ←

Scenario	No.	Communication Direction	ID	Function Code	Date	CRC
Write gripper model	1	→	02	10	06 00 00 01 02 0A 20	D3 D8
	2	←	02	10	06 00 00 01	01 72
Read gripper model	3	→	02	03	06 00 00 01	84 B1
	4	←	02	03	02 0A 20	FB 3C
Read firmware version	5	→	02	04	03 03 00 04	01 BE
	6	←	02	04	08 00 03 00 00 00 01 03 74	49 5E
Write reset	7	→	02	10	06 10 00 01 02 00 01	17 F0
	8	←	02	10	06 10 00 01	00 B7
Read gripper status	9	→	02	04	03 01 00 01	60 7D
	10	←	02	04	02 00 01	3C F0
Wait for 30 seconds	11	None				
Read status and position	12	→	02	04	03 01 00 01	60 7D
	13	←	02	04	02 00 00	FD 30
Write Expert	14	→	02	10	06 40 00 07 0E 00 00 03 E8 1F 40 01 F4 07 D0 00 64 00 01	E6 B8
	15	←	02	10	06 40 00 07	80 A4
Read gripper status	16	→	02	04	03 01 00 01	60 7D
	17	←	02	04	02 00 01	3C F0

Wait for three seconds	18	None				
Read status and position	19	→	02	04	03 01 00 01	60 7D
	20	←	02	04	02 00 00	FD 30
Broadcast to write Move	21	→	00	10	06 30 00 03 06 0C 80 1F 40 00 01	3C A4
	22	None, no response form Controller when broadcasting				
Read gripper status	23	→	02	04	03 01 00 01	60 7D
	24	←	02	04	02 00 01	3C F0
Wait for three seconds	25	None				
Read status and position	26	→	02	04	03 01 00 01	60 7D
	27	←	02	04	02 00 02	7C F1

Appendix 2: CRC-16 Calculation

The Controller uses CRC-16 as cyclic redundancy check. The following provide the C example functions from the official document of Modbus, used to generate CRC16. The function input must include the Message Structure and Message Length. All bytes from ID to data are calculated in the Message Structure. Two sets of table array are referred in the calculation. The generating functions for CRC are as follows:

```
/* The function returns the CRC as a unsigned short type    */
unsigned short CRC16 (unsigned char *puchMsg, unsigned short usDataLen)
{
    unsigned char uchCRCHi = 0xFF ;    /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ;    /* low byte of CRC initialized */
    unsigned uIndex ;    /* will index into CRC lookup table */
    while (usDataLen--)    /* pass through message buffer */
    {
        uIndex = uchCRCLo ^ *puchMsg++ ;      /* calculate the CRC      */
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex] ;
        uchCRCHi = auchCRCLo[uIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}
```

High-Order Byte Table /* Table of CRC values for high–order byte */

static unsigned char auchCRCHi[] = {

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,

0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40

} ;

Low-Order Byte Table /* Table of CRC values for low–order byte */

static char auchCRCLo[] = {

0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,

0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,

0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,

0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,

0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,

0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,

0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,

0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,

0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,

0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,

0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,

0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,

0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,

0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,

0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,

0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,

0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,

0x40

};

Electric Gripper Modbus Operation Manual (Original Instruction)

Publication Date：September 2021

1. HIWIN is a registered trademark of HIWIN Technologies Corp.. For your protection, avoid buying counterfeit products from unknown sources.
2. Actual products may differ from specifications and photos provided in this catalog. These differences may be the result of various factors including product improvements.
3. HIWIN website for patented product directory: http://www.hiwin.tw/Products/Products_patents.aspx
4. HIWIN will not sell or export products or processes restricted under the "Foreign Trade Act" or related regulations. Export of restricted products should be approved by proper authorities in accordance with relevant laws and shall not be used to manufacture or develop nuclear, biochemical, missiles or other weapons.

HIWIN®

Global Sales And Customer Service Site

HIWIN GmbH
OFFENBURG, GERMANY
www.hiwin.de
www.hiwin.eu
info@hiwin.de

HIWIN Schweiz GmbH
JONA, SWITZERLAND
www.hiwin.ch
info@hiwin.ch

HIWIN KOREA
SUWON · CHANGWON, KOREA
www.hiwin.kr
info@hiwin.kr

HIWIN JAPAN
KOBE · TOKYO · NAGOYA · NAGANO · TOHOKU · SHIZUOKA · HOKURIKU · HIROSHIMA · FUKUOKA · KUMAMOTO, JAPAN
www.hiwin.co.jp
info@hiwin.co.jp

HIWIN s.r.o.
BRNO, CZECH REPUBLIC
www.hiwin.cz
info@hiwin.cz

HIWIN CHINA
SUZHOU, CHINA
www.hiwin.cn
info@hiwin.cn

HIWIN USA
CHICAGO, U.S.A.
www.hiwin.us
info@hiwin.com

HIWIN FRANCE
STRASBOURG, FRANCE
www.hiwin.fr
info@hiwin.de

Mega-Fabs Motion Systems, Ltd.
HAIFA, ISRAEL
www.mega-fabs.com
info@mega-fabs.com

HIWIN Srl
BRUGHERIO, ITALY
www.hiwin.it
info@hiwin.it

HIWIN SINGAPORE
SINGAPORE
www.hiwin.sg
info@hiwin.sg

HIWIN TECHNOLOGIES CORP.
No. 7, Jingke Road,
Taichung Precision Machinery Park,
Taichung 40852, Taiwan
Tel: +886-4-23594510
Fax: +886-4-23594420
www.hiwin.tw
business@hiwin.tw (Sales)
robotservice@hiwin.tw (Customer Service)

The specifications in this catalog are subject to change without notification.

©2021 FORM C02UE801-2109 (PRINTED IN TAIWAN)