UFACTORY

UFACTORY xARM

BIO GRIPPER USER MANUAL

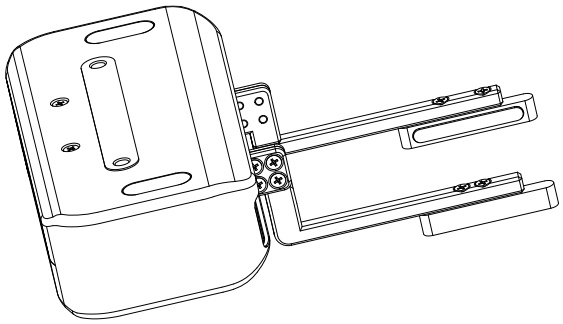
SHENZHEN UFACTORY CO., LTD

V 1.6.1

Table

1. General Presentation

1.1. Gripper Introduction



BIO Gripper

The xArm BIO Gripper is a gripper designed for liquid handling. It provides fast deployment paired with simple customization and easy programming. The gripper is a multifunctional tool, boasting customized fingertips to provide gripping flexibility.

Main Features of BIO Gripper:

(1) Designed for liquid handling

The gripper stroke is 70-150mm with special designed fingertips which perfectly fit the liquid plate.

(2) Customizable fingertips

Easily customized fingertips enable all sorts of fitting for tubes and plates corresponding to their shapes.

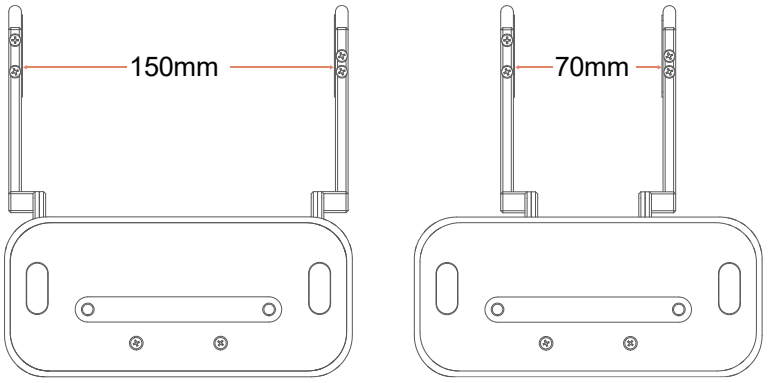
(3) Highly integrated with xArm

As a safe and stable integration, the xArm BIO Gripper is highly compatible with xArm, controlled directly by the IO port at the end of the machine without external cables and connectors.

1.2. Object Picking

The fingers of the BIO Gripper adopt a parallel grasp, Figures are shown below.

The width of the object that the gripper can grasp is: 70-150mm



(a) Maximum Stroke (b) Minimum Stroke

1.3. Setup and Control

The gripper is powered and controlled directly via a single gripper connection cable that carries a 24V DC supply and Modbus RTU communication over RS-485.

1.4. Safety

Warning

The operator must have read and understood all of the instructions in the following manual before handling the BIO Gripper.

Caution

The term "operator" refers to anyone responsible for any of the following operations on the BIO Gripper:

- Installation
- Control
- Maintenance
- Inspection
- Calibration
- Programming
- Decommissioning

This documentation explains the various components of the BIO Gripper and general operations regarding the whole life-cycle of the product from installation to operation and decommissioning.

The drawings and photos in this documentation are representative examples and differences may exist between them and the delivered product.

1.4.1. Warning

Caution

Any use of the Gripper in noncompliance of these warnings is inappropriate and may cause injury or damage.

Warning

- The Gripper needs to be properly secured before operating the robot.
- Do not install or operate a Gripper that is damaged or lacking parts.
- Never supply the Gripper with an alternative current (AC) source.
- Make sure all cord sets are always secured at both ends,Gripper end & Robot end
- Always satisfy the recommended keying for electrical connections.
- Be sure no one is in the robot and/or gripper path before initializing the robot's routine.
- Always satisfy the gripper payload.
- Set the gripper speed accordingly, based on your application.
- Keep fingers and clothes away from the gripper while the power is on.
- Do not use the gripper on people or animals.

1.4.2. Risk Assessment and Final Application

The BIO Gripper is meant to be used on an industrial robot. The robot, gripper and any other equipment used in the final application must be evaluated with a risk assessment. The robot integrator must ensure that all local safety measures and regulations are respected. Depending on the application, there may be risks that need additional protection/safety measures, for example, the work-piece the gripper is manipulating may be inherently dangerous to the operator.

1.4.3. Intended Use

The gripper is designed for grasping and temporarily securing or holding objects.

Caution

The Gripper is NOT intended for applying force against objects or surfaces.

The product is intended for installation on a robot or other automated machinery and equipment.

Info

Always comply with local and/or national laws, regulations and directives on automation safety and general machine safety.

The unit may be used only within the range of its technical data. Any other use of the product is deemed improper and unintended use.

UFACTORY will not be liable for any damages resulting from any improper or unintended use.

2. Installation

The following subsections will guide you through the installation and general setup of BIO Gripper.

(1) The Scope of Delivery section

(2) The Mechanical Installation section

(3) The Electrical Setup section

Warning

Before installing:

Read and understand the safety instructions related to the BIO Gripper.

Verify your package according to the Scope of delivery and your order info.

Have the required parts, equipment and tools listed in the requirements readily available.

Installing:

Satisfy the environmental conditions.

Do not operate the Gripper, or even turn on the power supply, before it is firmly anchored and the danger zone is cleared.

Caution the fingers of the gripper which may move and cause injury or damage.

2.1. Scope of Delivery

2.1.1. General Kit

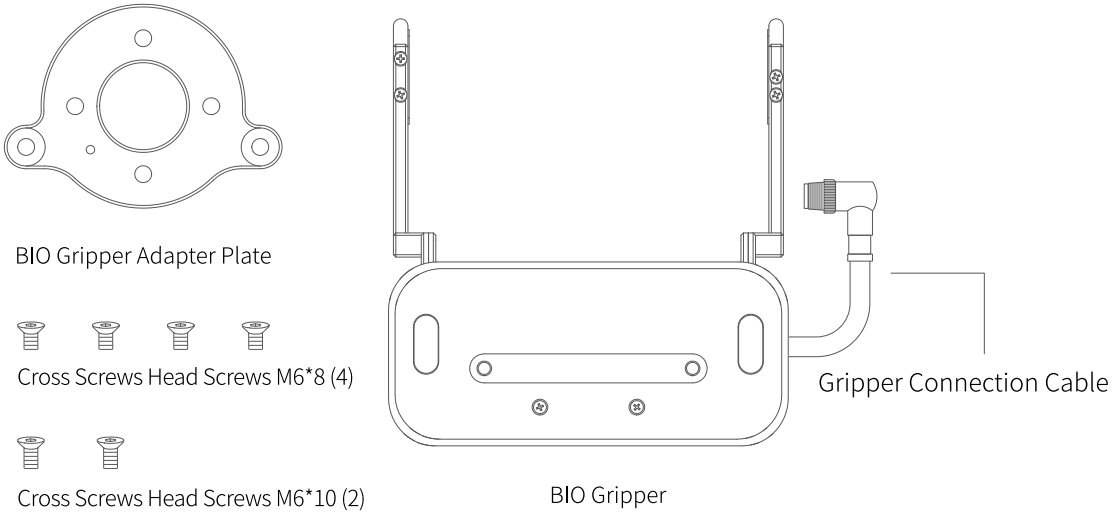
A Gripper Kit generally includes these items:

BIO Gripper

BIO Gripper adapter plate

Cross countersunk head screws M6*8 (4)

Cross countersunk head screws M6*10 (2)



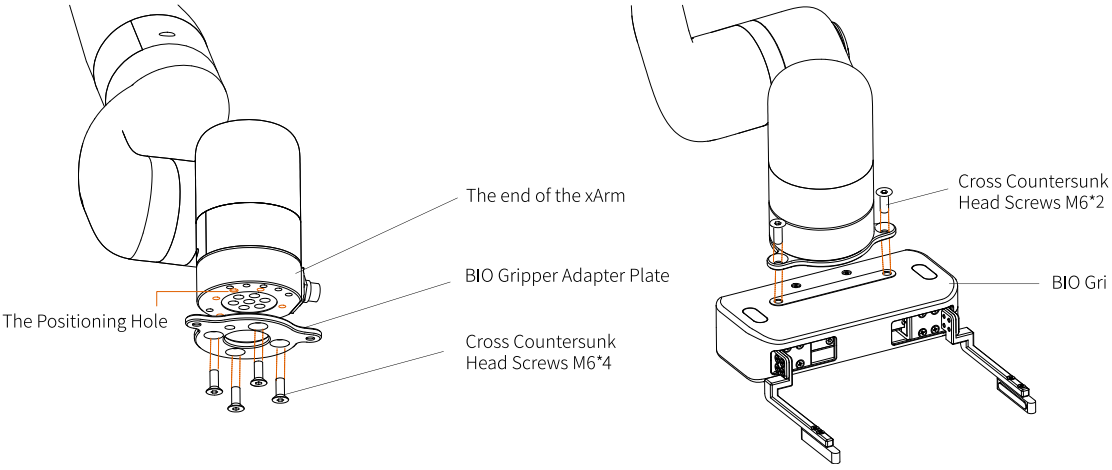
2.2. Mechanical Installation

BIO Gripper installation steps (as shown below):

(1) Fix the BIO Gripper adapter plate to the tool flange of the robotic arm with screws. (Note: The positioning posts on the adapter plate should be aligned with the positioning hole at the end of the robotic arm.)

(2) Fix the BIO Gripper to the BIO Gripper adapter plate with screws.

(3) Connect the robotic arm and the Gripper with the gripper connection cable.



Note:

1. When wiring the gripper connection cable, be sure to power off the robotic arm, the emergency stop button is in the pressed state and the power indicator of the robotic arm is off, so as to avoid robotic arm failure caused by hot plugging;

2. When connecting the gripper and the robotic arm, be sure to align the positioning holes at the ends of the gripper and the robotic arm. Since the male pins of the gripper connection cable are relatively thin, avoid bending the male pins during plugging.

2.3. Electrical Setup

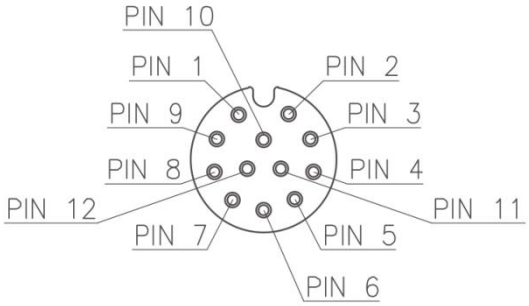
Power and communication are established with the BIO Gripper via a single gripper connection cable. The gripper connection cable provides a 24V power supply to the Gripper and enables serial RS485 communication to the robot control box.

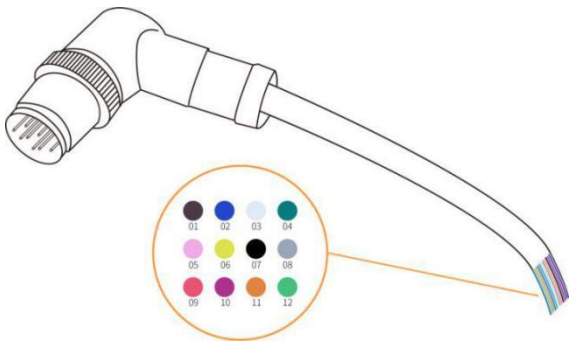
Warning

Power must be off before connecting the Gripper and the robotic arm via the gripper connection cable.

2.3.1. Pinout Interface

The BIO Gripper is connected to the tool end of the robotic arm via a 12 pin connector.





Pin sequence	Function
1	24V
2	24V
3	GND
4	GND
5	485-A
6	485-B
7	Digital Input 0
8	Digital Input 1
9	Digital Output 0
10	Digital Output 1
11	No Connect
12	No Connect

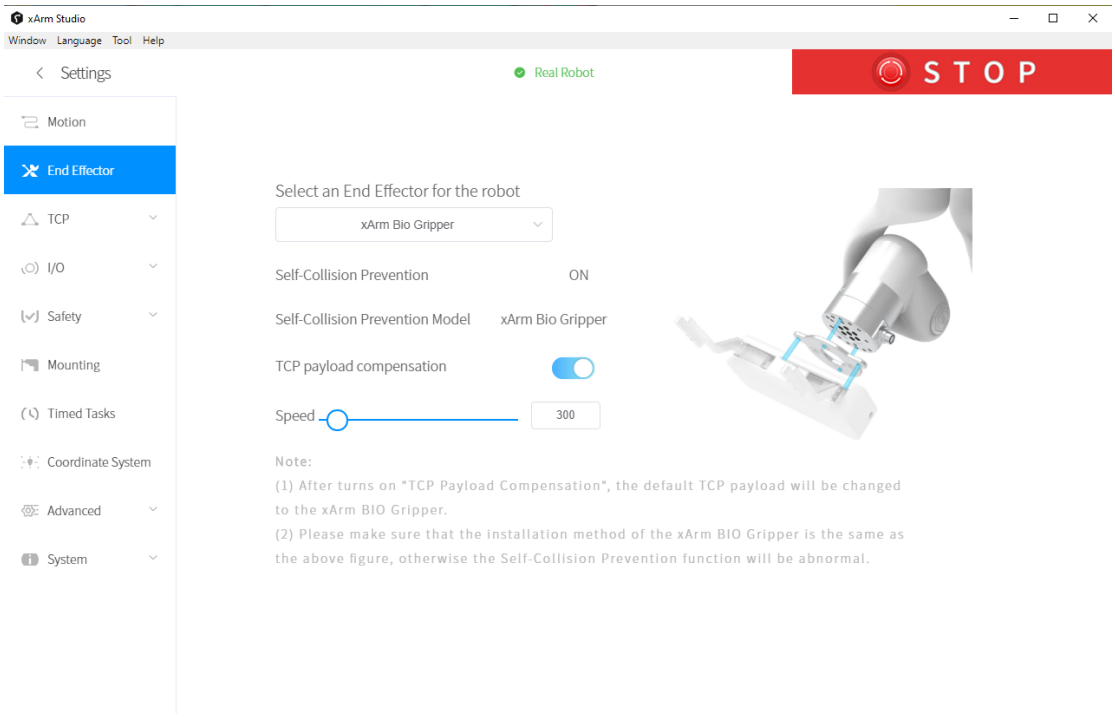
3. Control

3. 1. Use xArm Studio to Control BIO Gripper

1. Set up BIO Gripper

- Enter [Settings]-[End Effector]

Select the end effector: xArm BIO Gripper



1. The opening and closing speed of the gripper can be adjusted.

2. The self-collision prevention model of the gripper can be turned on by clicking the button.

3. When "TCP payload compensation" is turned on, the default TCP payload will be changed to the TCP payload parameter of the gripper.

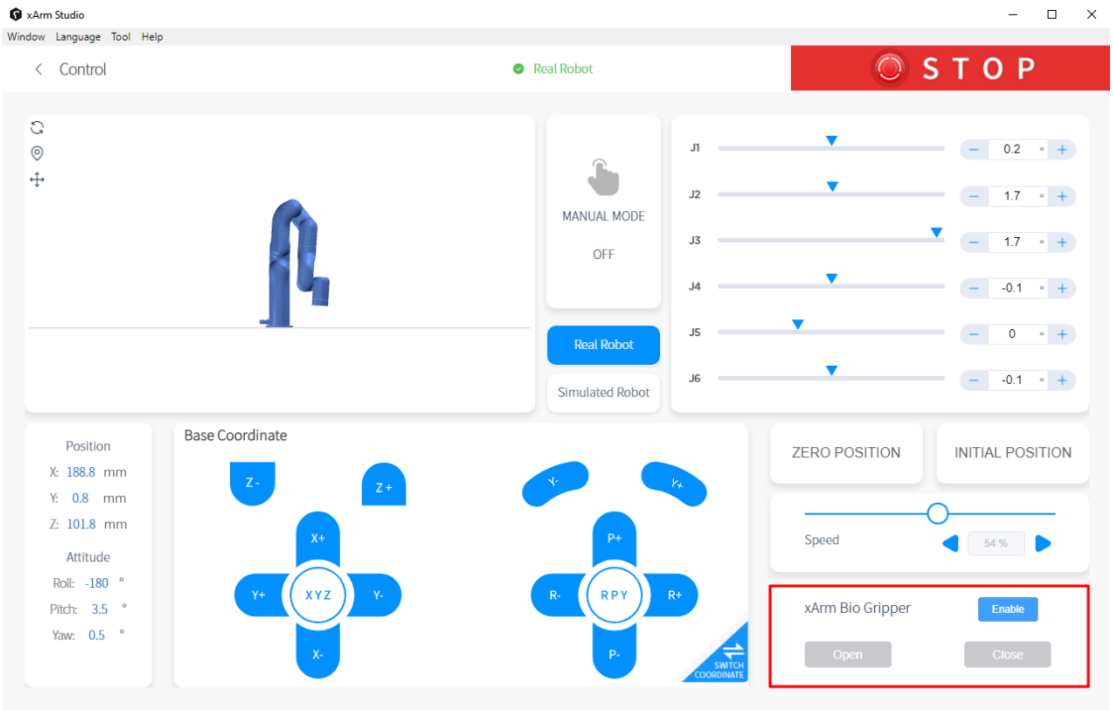
2. Control BIO Gripper

- Control the BIO gripper in the live control

Control Method:

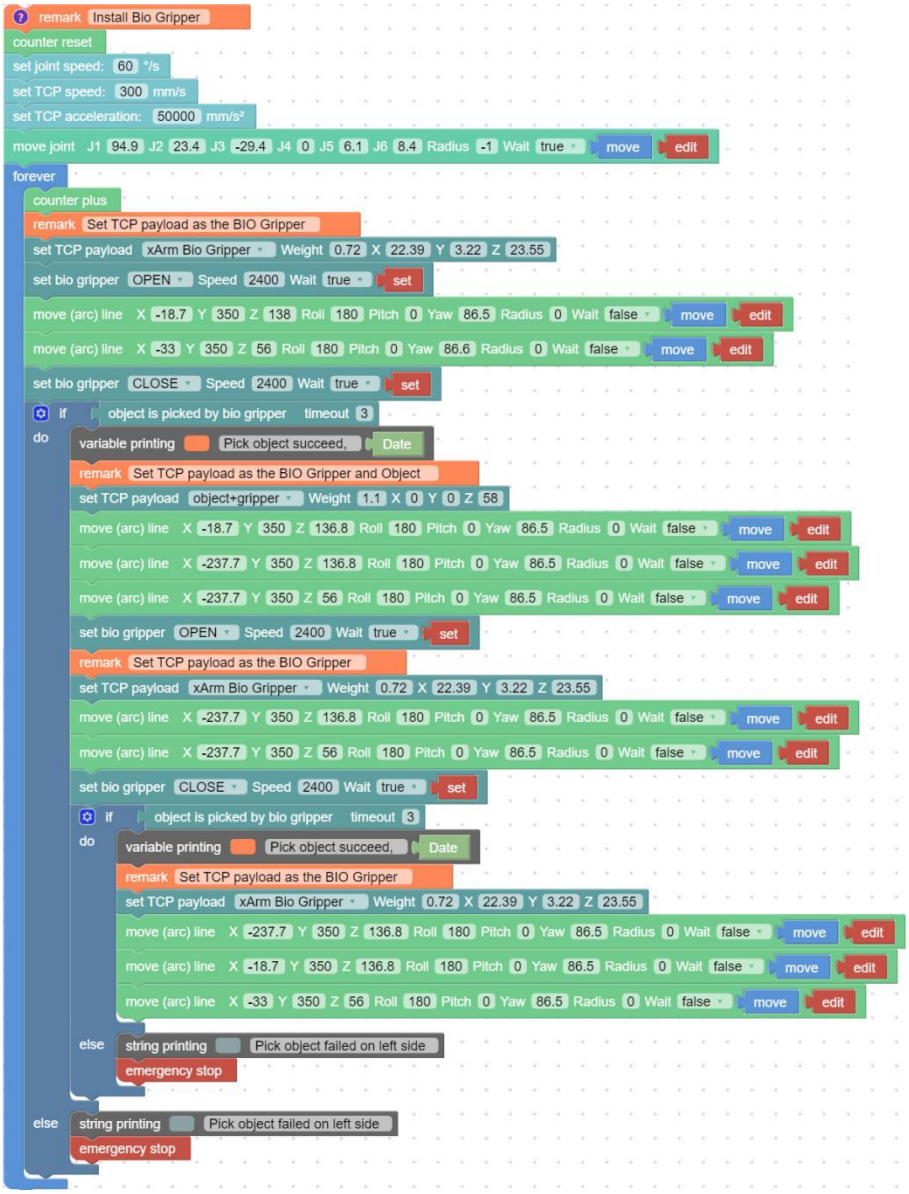
1) Click the [Enable] button to enable the BIO gripper;

2) By clicking the [Open]/[Close] button, you can control the opening and closing of the gripper.



- Control the BIO gripper through Blockly

BIO_Gripper.Blockly



The role of this program: execute this program to control the gripper to pick the target object at the specified position, and then place the target object at the target position.

Note:

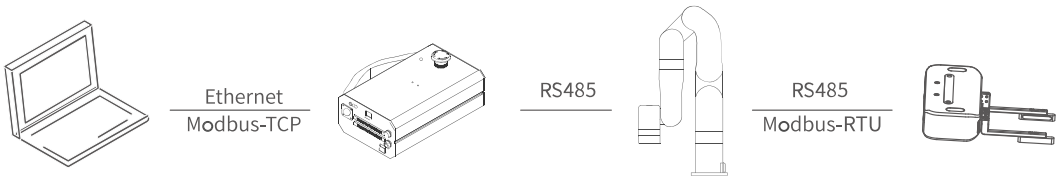
1) When the gripper is installed on the robotic arm, the TCP Payload of the gripper should be set in the Blockly program. When the total weight of the gripper changes after the object is picked, a new TCP Payload needs to be set.

3.2. Use Python-SDK to Control BIO Gripper

For details on controlling Gripper with python-SDK, please refer to the link below:

https://github.com/xArm-Developer/xArm-Python-SDK/blob/master/example/wrapper/common/5009-set_bio_gripper.py

3.3. Use Modbus-TCP Communication Protocol to Control BIO Gripper



This section mainly explains how to control the BIO Gripper by using the Modbus-TCP protocol through xArm control box.

3.3.1. Modbus-TCP Communication Format

Modbus-TCP:

Modbus protocol is an application layer message transmission protocol, including

three message types: ASCII, RTU, and TCP. The standard Modbus protocol physical layer interface includes RS232, RS422, RS485 and Ethernet interfaces, and adopts master / slave communication.

Modbus TCP Communication Process:

1. Establish a TCP connection

2. Prepare Modbus messages

3. Use the send command to send a message

4. Waiting for a response under the same connection

5. Use the recv command to read the message and complete a data exchange

6. When the communication task ends, close the TCP connection

Parameter:

Default TCP Port: 502 Protocol: 0x00 0x02

On the problem of users using communication protocols to organize data in big endian and little endian:

In this article, data analysis is big-endian analysis.

3.3.2. Read BIO Gripper Register

3.3.2.1. Register Function

Read Register			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	6+**N***x2
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	**Address**
	Quantity of Registers	**N***x2 Bytes	**N***
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	6+**N***x2
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	**N*x2**
	Registers Value	**N***x2 Bytes	**Value**

N* = Quantity of Registers

Address = Register Starting Address

Resgister:

	Resgister Starting Address	**Registers Value**	
Get Gripper status Register	0x0000	2 Bytes	**Disabled:** 0x0000 **Enabling:** 0x0004 **Enabling completed:** 0x0008 **Stop status**: 0x0008 **Motion status:** 0x0009 **Clipping status:** 0x000A

			Error status: 0x000B
Get Gripper Error Register	0x000F	2 Bytes	**An error occurs:** all other return values indicate an error(except 0) **No error occurred:** 0x0000

3.3.2.2. Example

1. Get the BIO Gripper status

Get the BIO Gripper status			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x08
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0000
	Quantity of Registers	2 Bytes	0x0001
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0008
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value (Robotic arm is in motion status)	2 Bytes	0x0009

2. Get the BIO Gripper Error

Get the BIO Gripper Error			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x08
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x000F
	Quantity of Registers	2 Bytes	0x0001
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0008
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value (No error occurred in the Gripper)	2 Bytes	0x0000

3. 3. 3. Write BIO Gripper Register

3. 3. 3. 1. Register Function

Write Register			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	9+**N***x2
	Unit Identifier	1 Byte	0x7C

Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	**Address**
	Quantity of Registers	2 Bytes	**N***
	Byte Count	1 Byte	**N*x2**
	Registers Value	**N***x2	**Value**
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0009
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	**Address**
	Quantity of Registers	2 Bytes	**N***

N* = Quantity of Registers

Address = Register Starting Address

Resgister:

	Resgister Starting Address	**Registers Value**	
Enable/Disable Gripper Register	0x0100	2 Bytes	**Enable :** 0x0001 **Disable :** 0x0000
Set Gripper Position Register	0x0700	4 Bytes	**Open the Gripper :** 0x0000 0x0082 **Close the Gripper :** 0x0000 0x0032
Set Position Speed Register	0x0303	2 Bytes	0x0100-0x0400 **Unit :** r/min
Clear Position Error Register	0x000F	2 Bytes	0x0000

3.3.3.2. Example

1. Enable/Disable BIO Gripper

Enable/Disable BIO Gripper			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x000B
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x0100
	Quantity of Registers	2 Bytes	0x0001
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x0001
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0009
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x0100
	Quantity of Registers	2 Bytes	0x0001

2. Set BIO Gripper Speed

Set BIO Gripper Speed			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x000B
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09

Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0303
	Quantity of Registers	2 Bytes	0x0001
	Byte Count	1 Byte	0x02
	Registers Value(1500r/min)	2 Bytes	0x05DC
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0009
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0303
	Quantity of Registers	2 Bytes	0x0001

3. Set BIO Gripper Position

Set BIO Gripper Position			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x000D
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x0700
	Quantity of Registers	2 Bytes	0x0002
	Byte Count	1 Byte	0x04
	Registers Value (Open the BIO Gripper)	4 Bytes	0x0000, 0x0082

Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0009
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x0700
	Quantity of Registers	2 Bytes	0x0002

4. Clear BIO Gripper Error

Clear BIO Gripper Error			
Request			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x000B
	Unit Identifier	1 Byte	0x7C
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x000F
	Quantity of Registers	2 Bytes	0x0001
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x0001
Response			
MBTP Header	Transaction Identifier	2 Bytes	0x0001
	Protocol Identifier	2 Bytes	0x0002
	Length	2 Bytes	0x0009
	Unit Identifier	1 Byte	0x7C
	Status Value	1 Byte	0x00
Internal Use	Internal Use	1 Byte	0x09
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x000F
	Quantity of Registers	2 Bytes	0x0001

3.3.4. BIO Gripper Control Process

The complete process of controlling the motion of the BIO Gripper is as follows:

(1) Enable the Gripper

0x00, 0x01, 0x00, 0x02, 0x00, 0x0B, 0x7C, 0x09, 0x08, 0x10, 0x01, 0x00, 0x00, 0x01, 0x02, 0x00, 0x01

(2) Open the Gripper

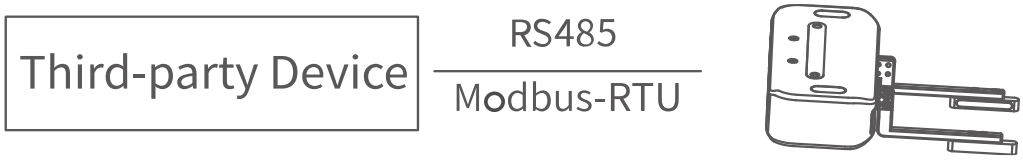
0x00, 0x01, 0x00, 0x02, 0x00, 0x0D, 0x7C, 0x09, 0x08, 0x10, 0x07, 0x00, 0x00, 0x02, 0x04, 0x00, 0x00, 0x00, 0x82

(3) Close the Gripper

0x00, 0x01, 0x00, 0x02, 0x00, 0x0D, 0x7C, 0x09, 0x08, 0x10, 0x07, 0x00, 0x00, 0x02, 0x04, 0x00, 0x00, 0x00, 0x32

3.4. Use Modbus-RTU Communication Protocol to Control BIO Gripper

3.4.1. Modbus RTU Communication Format



The gripper defaults to the standard Modbus RTU protocol at a default baud rate is 2Mbps and the slave ID is 0x08. The currently supported function codes are: 0x03 /

0x10. In this article, data analysis is big-endian analysis.

3.4.2. Read BIO Gripper Register

Read Register			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	**Address**
	Quantity of Register	2 Bytes	**N***
	Modbus CRC16	2 Bytes	**CRC***
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	**N*x2**
	Registers Value	**N***x2 Bytes	**Value**
	Modbus CRC16	2 Bytes	**CRC***

N* = Quantity of Registers

Address = Register Starting Address

CRC* = Cyclic Redundancy Check

Resgister:

	Resgister Starting Address	Register Value	
Get Gripper status Register	0x0000	2 Bytes	**Disabled:** 0x0000 **Enabling:** 0x0004 **Enabling completed:** 0x0008 **Stop status**: 0x0008 **Motion status:** 0x0009 **Clipping status:** 0x000A **Error status:** 0x000B
Get Gripper Error Register	0x000F	2 Bytes	**An error occurs:** all other return values indicate an error(except 0)

			No error occurred: 0x0000

3. 4. 3. Write BIO Gripper Register

Write Register			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	**Address**
	Quantity of Register	2 Bytes	**N***
	Byte Count	1 Byte	**N*x2**
	Registers Value	**N***x2	**Value**
	Modbus CRC16	2 Bytes	**CRC***
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	**Address**
	Quantity of Registers	2 Bytes	**N***
	Modbus CRC16	2 Bytes	**CRC***

N* = Quantity of Registers

Address = Register Starting Address

CRC* = Cyclic Redundancy Check

Resgister:

	Resgister Starting Address	**Register Value**	
Enable/Disable Gripper Register	0x0100	2 Bytes	**Enable :** 0x0001 **Disable :** 0x0000
Set Gripper Position Register	0x0700	4 Bytes	**Open the Gripper :** 0x0000 0x0082 **Close the Gripper :** 0x0000 0x0032
Set Position Speed Register	0x0303	2 Bytes	0x0100-0x0400 **Unit :** r/min

Clear Position Error Register	0x000F	2 Bytes	0x0000

3. 4. 4. Modbus RTU Example

This section demonstrates the example given in the Control Logic section when programmed using the Modbus RTU protocol.

Step1: Enable BIO Gripper

Enable BIO Gripper			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x0100
	Quantity of Registers	2 Bytes	0x0001
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x0001
	Modbus CRC16	2 Bytes	0x1D00
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting	2 Bytes	0x0100
	Quantity of Registers	2 Bytes	0x0001
	Modbus CRC16	2 Bytes	0x00AC

Step2: Set BIO Gripper Speed

Set BIO Gripper Speed			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0303
	Quantity of Registers	2 Bytes	0x0001
	Byte Count	1 Byte	0x02

	Registers Value(1500r/min)	2 Bytes	0x05DC
	Modbus CRC16	2 Bytes	0xFDFA
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0303
	Quantity of Registers	2 Bytes	0x0001
	Modbus CRC16	2 Bytes	0xF114

Step3：Open BIO Gripper

Open BIO Gripper			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0700
	Quantity of Registers	2 Bytes	0x0002
	Byte Count	1 Byte	0x04
	Registers Value (Open the BIO Gripper)	4 Bytes	0x0000, 0x0082
	Modbus CRC16	2 Bytes	0x7B62
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0700
	Quantity of Registers	2 Bytes	0x0002
	Modbus CRC16	2 Bytes	0x4025

Read the Gripper status until it is in a stopped status.

Get the BIO Gripper status			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0000
	Quantity of Registers	2 Bytes	0x0001
	Modbus CRC16	2 Bytes	0x8493
Response			

Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value (Robotic arm is in stop statu)	2 Bytes	0x0000
	Modbus CRC16	2 Bytes	0x6445

Step4：Close BIO Gripper

Close BIO Gripper			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0700
	Quantity of Registers	2 Bytes	0x0002
	Byte Count	1 Byte	0x04
	Registers Value (Close the BIO Gripper)	4 Bytes	0x0000, 0x0032
	Modbus CRC16	2 Bytes	0x7AD6
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0700
	Quantity of Registers	2 Bytes	0x0002
	Modbus CRC16	2 Bytes	0x4025

Read the Gripper status until it is in a stopped status.

Get the BIO Gripper status			
Request			
Modbus RTU Data	Slave ID (Gripper)	1 Byte	0x08
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0000
	Quantity of Registers	2 Bytes	0x0001
	Modbus CRC16	2 Bytes	0x8493
Response			
Modbus RTU Data	Slave ID	1 Byte	0x08

	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value (Robotic arm is in stop statu)	2 Bytes	0x0000
	Modbus CRC16	2 Bytes	0x6445

4. Gripper Error Code & Error Handling

The user can re-power on the robotic arm as an error handling, the steps are as follows (re-power on need to perform all the following steps):

1. Re-powering the robotic arm via the emergency stop button on the control box.

2. Enable robotic arm.

   a. xArm Studio enable mode: Click the guide button of the error pop-up window or the 'STOP' red button in the upper right corner.

   b. xArm-Python-SDK enable mode: Refer to Alarm Handling Mode.

   c. xArm_ROS library: users can view related documents at https://github.com/xArm-Developer/xarm_ros

3. Re-enable the gripper.

Error Code	Error Description	Error Handling
0x0B	Gripper overcurrent	Gripper current is too large Please click "OK" to re-enable the Gripper
0x0C	The gripping object falls off	The gripping object falls off Please place the gripping object and clear the error, set the 0x000F register to 0.
If the problem remains unsolved after power on/off for multiple times, please contact UFACTORY team for support.		

xArm-Python-SDK Error Handling:

When designing the robotic arm motion path with the Python library, if the robotic arm error (see Appendix for Alarm information) occurs, then it needs to be cleared manually. After clearing the error, the robotic arm should be motion enabled.

Python library error clearing steps: (Please check GitHub for details on the following interfaces)

a. Eerror clearing: clean_error()

b. Re-enable the robotic arm: motion_enable(true)

c. Set the motion statu: set_statu(0)

5. BIO Gripper Technical Specifications

BIO Gripper	
Rated Supply Voltage	24V DC
Absolute Maximum Supply Voltage	28V DC
Static Power Consumption (Minimum Power Consumption)	0.96W
Peak Current	1.5A
Weight	760g
Maximum Gripping Force	20N
Stroke	70-150mm
Communication Mode	RS-485
Communication Protocol	Modbus RTU
Programmable Gripping Specification	Speed Control
Status Indication	Working Status, Power
Feedback	Drop Detection, Pick-up Detection

6. After-sales Service

1. After-sales policy:

For the detailed after-sales policy of the product, see the official website:

https://store-ufactory-cc.myshopify.com/pages/warranty-returns

2. The general process of after-sales service is:

(1) Contact UFACTORY technical support (support@ufactory.cc) to confirm whether the product needs to repair and which part should be send back to UFACTORY.

(2) After bill of lading on UPS, we will send the invoice and label to you by mail. You need to make an appointment with the local UPS and then send the product to us.

(3) UFACTORY will check the product warranty status according to the after-sales policy.

(4) Generally, the process takes around 1-2 weeks except for shipment.

Note:

1. When you need to send the product back to get repaired, please pack the product with the box to protect the product during transportation.