THE DEXTEROUS HAND
RH56 SERIES USER MANUAL

Beijing Inspire-Robots Technology Co., Ltd. | January 2024
ID: PRJ-02-TS-U-001 | Page Number: 26 pages | Version: V1.0.9

Table of Contents

Dexterous Hand RH56

User Manual

1 Product Overview

1.1 Product Features

RH56 Series Dexterous Hand is a mechanical dexterous hand (hereinafter referred to as "Dexterous Hand") designed to integrate the linear servo actuator with small volume and large torque. It is equipped with six linear servo actuators. For the user interface, RS232, RS485 and CAN communication interfaces is used. There is a sensitive built-in pressure sensor. By setting different 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 dexterous hand. The excellent performance makes this dexterous hand suitable for applications such as service robot, teaching aids, prosthetics, etc.

1.2 Construction and Dimensions

The construction and dimensions of this dexterous hand are shown in Figure 1.

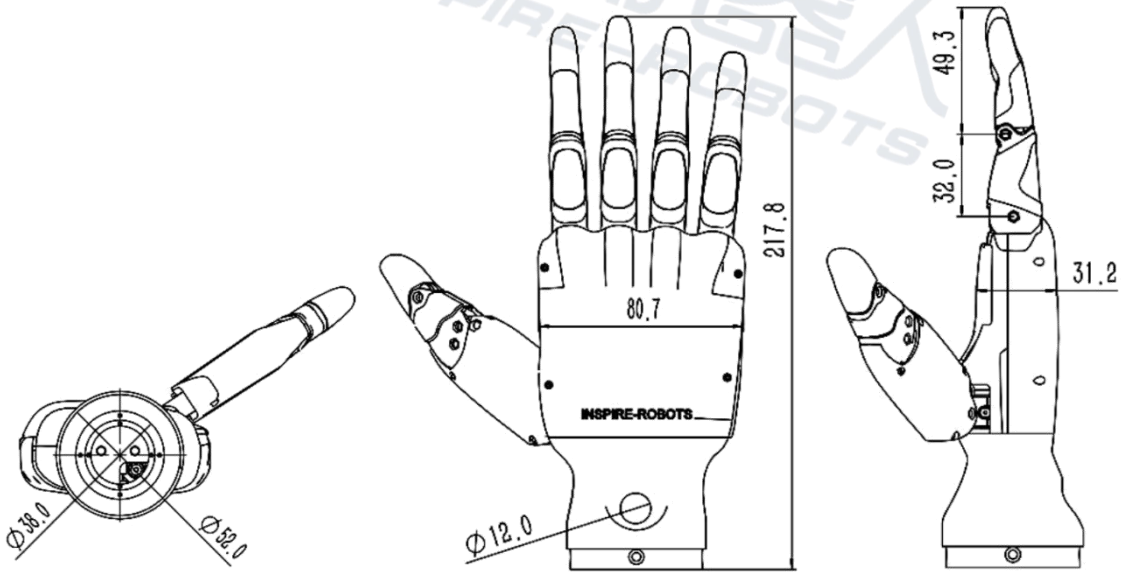


Figure 1: Construction and Dimensions

1.3 Performance parameters

Total number of joints	12
Degree of freedom (DOF)	6
Number of force sensors	6
Resolution of force sensor	0.5 N
Repeated fingertip positioning accuracy	0.2 mm
Gripping force of four fingers	DF: 10 N BF: 4 N
Gripping force of the thumb	DF: 15 N BF: 6 N
Operating voltage	24 V
Quiescent current	0.12 A @24 V
Average current for no-load operation	0.6 A @24 V
Current for maximum gripping force	2 A @24 V
Communication interface	RS485

1.4 Electric connection

1.4.1 Pin definition

The electrical interface of the dexterous hand is shown in Figure 2. The interface is a standard aviation plug with GX12 pins. The pins are defined as follows:

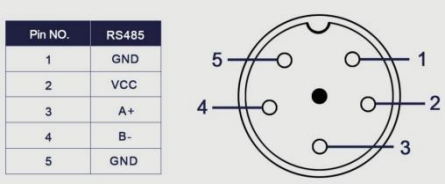
Interface type	Pin definition
RS485	

Figure 2: Interface Description

1.4.2 Communication mode

When the RS485 interface is used, a particular conversion circuit can realize parallel connection of 254 dexterous hands to the same bus. （The RS232 interface and CAN interface are customized dexterous hands.）

The communication instructions and registers of dexterous hands are open to the product users. Dexterous hands can be connected with users' PC (controller or general computer) via the aforesaid interface. The user can use a PC or an embedded controller to perform parameter configuration, motion control and other actions of dexterous hands via the aforesaid interface. Details are shown in Figure 3.

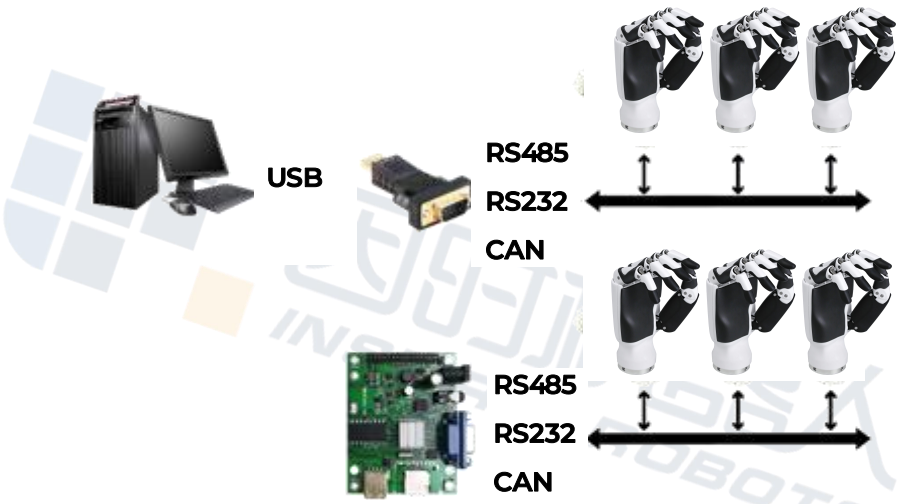


Figure 3: Communication Mode Diagram

2 Communication Protocol

2.1 Summary of Communication Protocol

The main control unit (MCU) reads and writes the internal register of a dexterous hand to achieve state acquisition and control for the dexterous hand.

Read the register: The upper system reads the internal register values of a dexterous hand (writable in groups; groups refer to several registers with adjacent addresses); the upper system sends a read instruction to the dexterous hand (including the register head address, length and other details of the register group); after the dexterous hand receives and successfully checks such data, it will return the data content of the corresponding register to the upper system.

Write the register: The upper system writes the corresponding data to the internal register of a dexterous hand (writable in groups); the upper system sends a write instruction to the dexterous hand (including the register head address of the register group and the data content to be written); after the dexterous hand receives and successfully checks such data, it will return the acknowledgment signal to the upper system.

2.2 Register reading/writing of RS485

The communication parameters of RS485 are 8 data bits, 1 stop bit, and no parity,the default baud rate is 115200bps.

2.2.1 Reading a register of the dexterous hand

The instruction frame format for reading a register of the dexterous hand is shown below. "Address" is the start address of the register to be read. "Hands_ID" is the ID of the dexterous hand. "Address_L" is the low-order 8 bits of "Address". "Address_H" is the high-order 8 bits of "Address". "Register_Length" is the length (unit: byte) of the register to be read. "Checksum" refers to the low-order bytes of the sum of all data before checksum except the response frame header.

	Value	Description
byte[0]	0xEB	Packet header
byte[1]	0x90	Packet header
byte[2]	Hands_ID	ID of the dexterous hand

byte[3]	0x04	Length of the frame data
byte[4]	0x11	Register reading flag
byte[5]	Address_L	Low-order 8 bits in the start address of the register
byte[6]	Address_H	High-order 8 bits in the start address of the register
byte[7]	Register_Length	Length of the register to be read
byte[8]	Checksum	Checksum

The dexterous hand returns the following response frame to the read register instruction:

	Value	**Description**
byte[0]	0x90	Packet header
byte[1]	0xEB	Packet header
byte[2]	Hands_ID	ID of the dexterous hand
byte[3]	Register_Length+3	Length of the frame data
byte[4]	0x11	This frame is the response to the read register instruction.
byte[5]	Address_L	Low-order 8 bits in the start address of the register
byte[6]	Address_H	High-order 8 bits in the start address of the register
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	Register value
byte[7+ Register_Length]	Checksum	Checksum

For example, the dexterous hand (ID=1) reads the actual angle for each degree of freedom (DOF). The actual angle for DOF is stored in the register group "ANGLE_ACT(m)", with 1546 (0x060A) as start address, and 12 bytes (0x0C) as length. The format of this instruction frame is shown below:

	Value	**Description**
byte[0]	0xEB	Packet header
byte[1]	0x90	Packet header
byte[2]	0x01	ID of the dexterous hand: 1
byte[3]	0x04	Length of the frame data

byte[4]	0x11	Register reading flag
byte[5]	0x0A	Low-order 8 bits in the start address of the register
byte[6]	0x06	High-order 8 bits in the start address of the register
byte[7]	0x0C	Length of the register to be read
byte[8]	0x32	Checksum

The dexterous hand returns the following response frame to this instruction:

	Value	**Description**
byte[0]	0x90	Packet header
byte[1]	0xEB	Packet header
byte[2]	0x01	ID of the dexterous hand
byte[3]	0x0F	Length of the frame data: 12 + 3
byte[4]	0x11	This frame is the response to the read register instruction.
byte[5]	0x0A	Low-order 8 bits in the start address of the register
byte[6]	0x06	High-order 8 bits in the start address of the register
byte[7] byte[8]	0x6400	Converted to the integer data 100 (0x0064)
byte[9] byte[10]	0x6400	Converted to the integer data 100 (0x0064)
byte[11] byte[12]	0x6400	Converted to the integer data 100 (0x0064)
byte[13] byte[14]	0x6400	Converted to the integer data 100 (0x0064)
byte[15] byte[16]	0xE803	Converted to the integer data 1000 (0x03E8)
byte[17] byte[18]	0x0000	Converted to the integer data 0 (0x0000)
byte[19]	0xAC	Checksum

From this response frame, we can obtain the actual values for DOF: 100, 100, 100, 100, 1000, and 0.

2.2.2 Writing to a register of the dexterous hand

The instruction frame format for writing to a register of the dexterous hand is shown below. "Data[0]- Data[Register_Length-1]" is the data to be written to the register.

	Value	**Description**
byte[0]	0xEB	Packet header
byte[1]	0x90	Packet header

byte[2]	Hands_ID	ID of the dexterous hand
byte[3]	Register_Length+3	Length of the frame data
byte[4]	0x12	Write Register instruction flag
byte[5]	Address_L	Low-order 8 bits in the start address of the register
byte[6]	Address_H	High-order 8 bits in the start address of the register
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	Data to be written to the register
byte[7+ Register_Length]	checksum	Checksum

The dexterous hand returns the following response frame to the write register instruction:

	Value	**Description**
byte[0]	0x90	Packet header
byte[1]	0xEB	Packet header
byte[2]	Hands_ID	ID of the dexterous hand
byte[3]	4	Length of the frame data
byte[4]	0x12	This frame is the response to the write register instruction.
byte[5]	Address_L	Low-order 8 bits in the start address of the register
byte[6]	Address_H	High-order 8 bits in the start address of the register
byte[7]	1	
byte[8]	checksum	Checksum

For example, the angles of a dexterous hand (ID=1) for degrees of freedom (DOF) are set to 100, 100, 100, 100, 1000, and 0. It is necessary to modify the register group "ANGLE_SET(m)". The start address of this register group is 1486 (0x05CE), and its length is 12 bytes (0x0C). The instruction to be sent is listed below:

	Value	**Description**
byte[0]	0xEB	Packet header
byte[1]	0x90	Packet header
byte[2]	0x01	ID of the dexterous hand
byte[3]	0x0F	Length of the frame data: 12 + 3

byte[4]	0x12	Write Register instruction flag
byte[5]	0xCE	Low-order 8 bits in the start address of the register
byte[6]	0x05	High-order 8 bits in the start address of the register
byte[7] byte[8]	0x6400	Converted to the integer data 100 (0x0064)
byte[9] byte[10]	0x6400	Converted to the integer data 100 (0x0064)
byte[11] byte[12]	0x6400	Converted to the integer data 100 (0x0064)
byte[13] byte[14]	0x6400	Converted to the integer data 100 (0x0064)
byte[15] byte[16]	0xE803	Converted to the integer data 1000 (0x03E8)
byte[17] byte[18]	0x0000	Converted to the integer data 0 (0x0000)
byte[19]	0x70	Checksum

The dexterous hand returns the following response frame to this instruction:

	Value	**Description**
byte[0]	0x90	Packet header
byte[1]	0xEB	Packet header
byte[2]	0x01	ID of the dexterous hand
byte[3]	0x04	Length of the frame data
byte[4]	0x12	This frame is the response to the write register instruction.
byte[5]	0xCE	Low-order 8 bits in the start address of the register
byte[6]	0x05	High-order 8 bits in the start address of the register
byte[7]	0x01	
byte[8]	0xEB	Checksum

2.3 Summary of MODBUS protocol

The MODBUS protocol adopts the master-slave request/response communication mode. The protocol frame contains function codes, data fields, and cyclic redundancy check (CRC). This actuator series supports reading holding register (function code: 0x03), preset single register (function code: 0x06), and preset multiple registers (function code: 0x10).

2.3.1 Read the holding register (function code: 0x03).

Query frame format of	Slave station address	Function code	Starting register (high-order	Starting register (low-order	Number of registers (high-order	Number of registers (low-order	CRC

master station			bytes)	bytes)	bytes)	bytes)	
	0x11	0x03	0x6B	0x00	0x00	0x02	XXXX

Explanation: Read the holding register of Slave Station 17# (0x11), with start address = 0x006B, number of registers = 0x0002, and end address = 0x006B+2-1=0x006C; that is, read the holding register of Slave Station 17#, 0x006B-0x006C; there are totally two registers.

Response frame format of slave station	Slave station address	Function code	Byte count	0x006B register (high-order bytes)	0x006B register (low-order bytes)	0x006C register (high-order bytes)	0x006C register (low-order bytes)	CRC
	0x01	0x03	0x04	0x00	0x01	0x00	0x02	XXXX

Explanation: Return to the holding register of Slave Station 17# (0x11), 0x006B-0x006C; there are totally two registers. The value of 0x006B register is 0x0001, and that of 0x0062 register is 0x0002.

2.3.2 Preset single register (function code: 0x06)

Query frame format of master station	Slave station address	Function code	Starting register (high-order bytes)	Starting register (low-order bytes)	Data content (high-order bytes)	Data content (low-order bytes)	CRC
	0x11	0x06	0x00	0x6B	0x10	0x00	XXXX

Explanation: Set the holding register of Slave Station 17# (0x11), with register address = 0x006B and data content = 0x1000.

Response frame format of slave station	Slave station address	Function code	Starting register (high-order bytes)	Starting register (low-order bytes)	Data content (high-order bytes)	Data content (low-order bytes)	CRC
	0x11	0x06	0x00	0x00	0x00	0x00	XXXX

2.3.3 Preset multiple registers (function code: 0x10)

Query frame format of master station	Slave station address	Function code	Starting register (high-order bytes)	Starting register (low-order bytes)	Number of registers (high-order bytes)	Number of registers (low-order bytes)	Byte count	Data (high-order bytes)	Data (low-order bytes)	Data (high-order bytes)	Data (low-order bytes)	CRC
	0x11	0x10	0x00	0x01	0x00	0x02	0x04	0x00	0x0A	0x01	0x02	XX XX

Explanation: Set the holding register of Slave Station 17# (0x11), with register start address = 0x0001, number of registers = 0x0002, byte count of data content = 0x04, 0x000A and 0x0102 as data content.

Response frame format of slave station	Slave station address	Function code	Starting register (high-order bytes)	Starting register (low-order bytes)	Number of registers (high-order bytes)	Number of registers (low-order bytes)	CRC
	0x11	0x10	0x00	0x01	0x00	0x02	XXXX

2.4 Register description

The register parameters of the dexterous hand which are open to the users are listed below:

Address	Meaning	Abbreviation	Length	Permission (read / write)
1000 (0x03E8)	ID	HAND_ID	1byte	W/R
1002 (0x03EA)	Baud rate setting	REDU_RATIO	1byte	W/R
1004 (0x03EC)	Error clearance	CLEAR_ERROR	1byte	W/R
1005 (0x03ED)	Saving data in Flash	SAVE	1byte	W/R
1006 (0x03EE)	Restoring factory defaults	RESET_PARA	1byte	W/R
1008 (0x03F0)	reserve	reserve	1byte	W/R
1009 (0x03F1)	Force sensor calibration	GESTURE_FORCE_CLB	1byte	W/R
1020 (0x03FC)	Actuator current protection value for each DOF	CURRENT_LIMIT(m)	6short(12byte)	W/R
1032 (0x0408)	Set value of power-on speed for each DOF	DEFAULT_SPEED_SET(m)	6short(12byte)	W/R
1044	Set value of power-on force control threshold for	DEFAULT_FORCE_SET(m)	6short(12byte)	W/R

(0x0414)	each DOF			
1472 (0x05C0)	System voltage	VLTAGE	1short	R
1474 (0x05C2)	Set value of the actuator position for each DOF	POS_SET(m)	6short(12byte)	W/R
1486 (0x05CE)	Set value of the angle for each DOF	ANGLE_SET(m)	6short(12byte)	W/R
1498 (0x05DA)	Set value of force control threshold for each DOF	FORCE_SET(m)	6short(12byte)	W/R
1522 (0x05F2)	Set value of the speed for each DOF	SPEED_SET(m)	6short(12byte)	W/R
1534 (0x05FE)	Actual value of the actuator position for each DOF	POS_ACT(m)	6short(12byte)	R
1546 (0x060A)	Actual angle for each degree of freedom (DOF)	ANGLE_ACT(m)	6short(12byte)	R
1582 (0x062E)	Actual force applied to each finger	FORCE_ACT(m)	6short(12byte)	R
1594 (0x063A)	Actuator current value for each DOF	CURRENT(m)	6short(12byte)	R
1606 (0x0646)	Error codes of the actuator for each DOF	ERROR(m)	6byte	R
1612 (0x064C)	Status information for each DOF	STATUS(m)	6byte	R
1618 (0x0652)	Actuator temperature for each DOF	TEMP(m)	6byte	R
2000 (0x07D0)	Current action sequence check code 1	ACTION_SEQ_CHECK DATA1	1byte	W/R
2001 (0x07D1)	Current action sequence check code 2	ACTION_SEQ_CHECK DATA2	1byte	W/R
2002 (0x07D2)	Total number of steps in the current action sequence	ACTION_SEQ_STEPN UM	1byte	W/R
2016 (0x07E0)	Setting of the current action sequence: Step 0	ACTION_SEQ_STEP(0)	19short(38byte)	W/R
2054 (0x0806)	Setting of the current action sequence: Step 1	ACTION_SEQ_STEP(1)	19short(38byte)	W/R

2092 (0x082C)	Setting of the current action sequence: Step 2	ACTION_SEQ_STEP(2)	19short(38byte)	W/R
2130 (0x0852)	Setting of the current action sequence: Step 3	ACTION_SEQ_STEP(3)	19short(38byte)	W/R
2168 (0x0878)	Setting of the current action sequence: Step 4	ACTION_SEQ_STEP(4)	19short(38byte)	W/R
2206 (0x089E)	Setting of the current action sequence: Step 5	ACTION_SEQ_STEP(5)	19short(38byte)	W/R
2244 (0x08C4)	Setting of the current action sequence: Step 6	ACTION_SEQ_STEP(6)	19short(38byte)	W/R
2282 (0x08EA)	Setting of the current action sequence: Step 7	ACTION_SEQ_STEP(7)	19short(38byte)	W/R
2320 (0x0910)	Current action sequence index number	ACTION_SEQ_INDEX	1byte	W/R
2321 (0x0911)	Save the current action sequence	SAVE_ACTION_SEQ	1byte	W/R
2322 (0x0912)	Run the current action sequence	ACTION_SEQ _RUN	1byte	W/R
2324 (0x0914)	Force control adjustment value of the action sequence	ACTION_ADJUST_FORCE_SET	2byte	W/R

2.4.1 HAND_ID (Dexterous Hand ID)

Default value = 1; range: 1-254; it can be saved.

When several dexterous hands are connected on one bus, each of them should be assigned a different ID.

2.4.2 REDU_RATIO (Baud rate setting)

Default value = 0; range: 0-2; it can be saved.

0: Baud rate = 115200

1: Baud rate = 57600

2: Baud rate = 19200

2.4.3 CLEAR_ERROR (error clearance)

Default value = 0; range: 0-1; it cannot be saved.

After 1 is written, clearable errors (actuator errors such as locked-rotor, over-current, abnormal operation, or communication error) in the dexterous hand will be cleared.

Note: The over temperature error of the actuator is not clearable. When the temperature falls, such error will be cleared automatically.

2.4.4 SAVE (Saving parameters in the FLASH)

Default value = 0; range: 0-1; it cannot be saved.

After 1 is written, the dexterous hand will write current parameters in the Flash. These parameters will not be lost after power failure.

2.4.5 RESET_PARA (restoring factory defaults)

Default value = 0; range: 0-1; it cannot be saved.

After 1 is written, parameters of the dexterous hand will be restored to factory default settings.

2.4.6 GESTURE_FORCE_CLB (force sensor calibration)

When the user sets this parameter to 1, the dexterous hand will start the force sensor calibration process, which will take 6 seconds. First of all, hold five fingers fully open; then bend four fingers (little finger, ring finger, middle finger, and index finger); hold the four fingers open, and bend the thumb; finally, extend the thumb.

Note: The dexterous hand must be maintained in the no-load state during calibration; that is, fingers cannot touch any object.

2.4.7 CURRENT_LIMIT(m) (actuator current protection value for each DOF)

This register group consists of six registers corresponding to the current protection value of the dexterous hand for 6 DOF. If current exceeds this value during finger movement, the finger will stop moving, and it will be indicated in the finger status register "STATUS(m)" that the finger stops due to current protection. More details are provided in the table below. These parameters can be saved after power failure.Unit: mA.

Address	Name	Description	Type of data	Range
1020-1021	CURRENT_LIMIT(0)	Current protection value of the little finger	short	0-1500
1022-1023	CURRENT_LIMIT(1)	Current protection value of the ring finger	short	0-1500
1024-1025	CURRENT_LIMIT(2)	Current protection value of the middle finger	short	0-1500
1026-1027	CURRENT_LIMIT(3)	Current protection value of the index finger	short	0-1500
1028-1029	CURRENT_LIMIT(4)	Current protection value of thumb bending	short	0-1500
1030-1031	CURRENT_LIMIT(5)	Current protection value of thumb rotation	short	0-1500

2.4.8 DEFAULT_SPEED_SET(m) (set value of power-on speed for each DOF)

This register group consists of six registers corresponding to the set value of the power-on speed of the dexterous hand for 6 DOF, with details in the table below. These parameters can be saved after power failure.Range values are dimensionless

Address	Name	Description	Type of data	Range
1032-1033	DEFAULT_SPEED_SET(0)	Initial power-on speed of the little finger	short	0-1000
1034-1035	DEFAULT_SPEED_SET(1)	Initial power-on speed of the ring finger	short	0-1000
1036-1037	DEFAULT_SPEED_SET(2)	Initial power-on speed of the middle finger	short	0-1000
1038-1039	DEFAULT_SPEED_SET(3)	Initial power-on speed of the index finger	short	0-1000
1040-1041	DEFAULT_SPEED_SET(4)	Initial power-on speed of thumb bending	short	0-1000

1042-1043	DEFAULT_SPEED_SET(5)	Initial power-on speed of thumb rotation	short	0-1000

Speed = 1000: This means that it will take 800ms for the fingers in the no-load state to move from a large angle to the minimum angle. If there is a heavy load, the actual speed will be somewhat lower than this value.

2.4.9 DEFAULT_FORCE_SET(m) (set value of power-on force control threshold for each DOF)

This register group consists of six registers corresponding to the set value of the power-on force control threshold of the dexterous hand for 6 DOF, with details in the table below. These parameters can be saved after power failure.Unit: g.

Address	Name	Description	Type of data	Range
1044-1045	DEFAULT_FORCE_SET(0)	Initial power-on force control of the little finger	short	0-1000
1046-1047	DEFAULT_FORCE_SET(1)	Initial power-on force control of the ring finger	short	0-1000
1048-1049	DEFAULT_FORCE_SET(2)	Initial power-on force control of the middle finger	short	0-1000
1050-1051	DEFAULT_FORCE_SET(3)	Initial power-on force control of the index finger	short	0-1000
1052-1053	DEFAULT_FORCE_SET(4)	Initial power-on force control of thumb bending	short	0-1500
1054-1055	DEFAULT_FORCE_SET(5)	Initial power-on force control of thumb rotation	short	0-1000

This register value indicates the gripping force provided by the fingertip of the corresponding finger. For example, when DEFAULT_FORCE_SET(1) is set to 800, it means that the fingertip of the ring finger is allowed to provide the gripping force of 800 g. (If the portion of a finger that touches an object is not the fingertip, the available gripping force will be different. Its magnitude is related to the length of the arm of force. The shorter the arm of force is, the greater the gripping force will be.)

2.4.10 POS_SET(m) (set value of the actuator position for each DOF)

This register group consists of six registers corresponding to the set value of the actuator position of the dexterous hand for 6 DOF, with details in the table below. These parameters cannot be saved..Range values are dimensionless

Address	Name	Description	Type of data	Range
1474-1475	POS_SET(0)	Actuator position setting for the little finger	short	0-2000
1476-1477	POS_SET(1)	Actuator position setting for the ring finger	short	0-2000
1478-1479	POS_SET(2)	Actuator position setting for the middle finger	short	0-2000
1480-1481	POS_SET(3)	Actuator position setting for the index finger	short	0-2000
1482-1483	POS_SET(4)	Actuator position setting for thumb bending	short	0-2000
1484-1485	POS_SET(5)	Actuator position setting for thumb rotation	short	0-2000

0: Minimum actuator stroke, corresponding to the maximum finger angle (i.e., holding fingers open);

2000: Maximum actuator stroke, corresponding to the minimum finger angle (i.e., bending fingers);

-1: The actuator does not take any action.

It is not recommended to set the finger position angle by setting this register group.

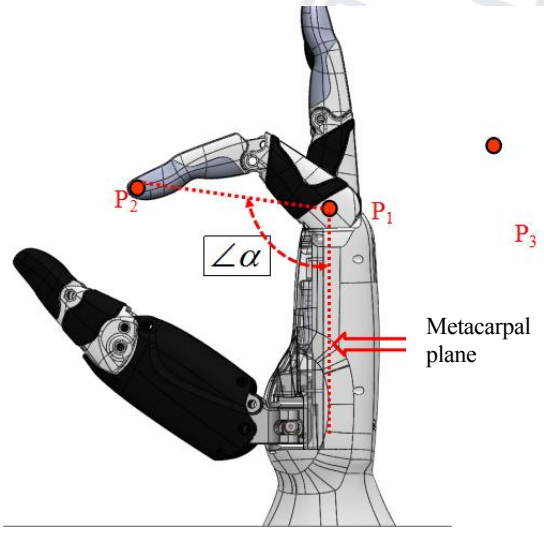
2.4.11 ANGLE_SET(m) (set value of the angle for each DOF)

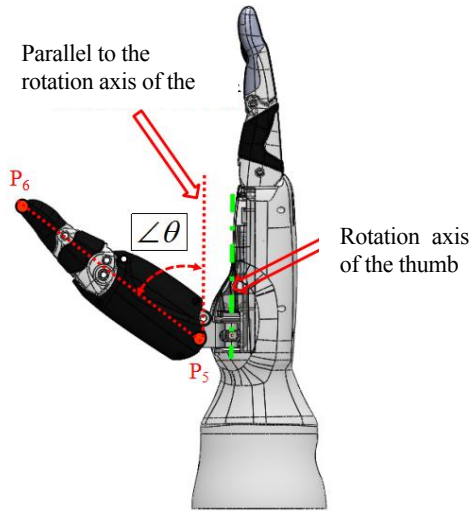
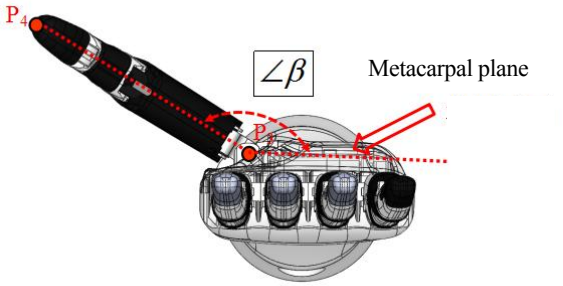
This register group consists of six registers corresponding to the set value of the angle of the dexterous hand for 6 DOF, with details in the table below. These parameters can be saved after power failure. When the angle value for a degree of freedom (DOF) is set to the data within 0-1000, the corresponding finger will take an immediate action. If the angle value is set to -1, the corresponding finger will make no response. For example, after the angle values of six registers (ANGLE_SET(0)- ANGLE_SET(5)) are set to 500, 500, -1, 0, 500 and 500 respectively, the little and ring fingers and the thumb will be bent, and the

thumb will rotate by 4 DOF and move to an angle corresponding to "500"; the index finger will move to an angle corresponding to "0"; and the middle finger will not move (i.e., it will be maintained in the current position without action).Range values are dimensionless

Address	Name	Description	Type of data	Range
1486-1487	ANGLE_SET(0)	Initial power-on angle of the little finger	short	-1, 0-1000
1488-1489	ANGLE_SET(1)	Initial power-on angle of the ring finger	short	-1, 0-1000
1490-1491	ANGLE_SET(2)	Initial power-on angle of the middle finger	short	-1, 0-1000
1492-1493	ANGLE_SET(3)	Initial power-on angle of the index finger	short	-1, 0-1000
1494-1495	ANGLE_SET(4)	Initial power-on angle of thumb bending	short	-1, 0-1000
1496-1497	ANGLE_SET(5)	Initial power-on angle of thumb rotation	short	0-1000

The definition of angle and the range of motion for each DOF are described as follows.

Angle	Legend description	Range
Little finger Ring finger Middle finger Index finger		19°to 176.7°

Bending angle of the thumb		-13°to 53.6°
Rotation angle of the thumb		90°to 165°

2.4.12 FORCE_SET (m) (set value of force control threshold for each DOF)

This register group consists of six registers corresponding to the set value of the force control threshold of the dexterous hand for 6 DOF, with details in the table below. These parameters can be saved after power failure. The user can set this register group to control the gripping force of fingers. For example, set FORCE_SET(3) to 300, and set the current actual angle of the index finger "ANGLE_ACT(3)" to 1000 (i.e., fully open). After the user sets the angle of the index finger "ANGLE_SET(3)" to 0, the index finger will bend towards the palm. Once it is detected that the actual force (i.e., FORCE_ACT(3); see 2.4.17 for more details) applied to the index finger during bending reaches 300, the index finger will stop moving; otherwise, the index finger will move to the preset angle..Unit: g.

Address	Name	Description	Type of data	Range
1498-1499	FORCE_SET(0)	Set value of force control for the little finger	short	0-1000
1500-1501	FORCE_SET(1)	Set value of force control for the ring finger	short	0-1000

1502-1503	FORCE_SET(2)	Set value of force control for the middle finger	short	0-1000
1504-1505	FORCE_SET(3)	Set value of force control for the index finger	short	0-1000
1506-1507	FORCE_SET(4)	Set value of bending force control for the thumb	short	0-1000
1508-1509	FORCE_SET(5)	Set value of rotation force control for the thumb	short	0-1000

This register value indicates the gripping force provided by the fingertip of the corresponding finger. For example, when DEFAULT_FORCE_SET(1) is set to 800, it means that the fingertip of the ring finger is allowed to provide the gripping force of 800 g. (If the portion of a finger that touches an object is not the fingertip, the available gripping force will be different. Its magnitude is related to the length of the arm of force. The shorter the arm of force is, the greater the gripping force will be.)

2.4.13 SPEED_SET(m) (set value of the speed for each DOF)

This register group consists of six registers corresponding to the set value of the speed of the dexterous hand for 6 DOF, with details in the table below. These parameters can be saved after power failure..Range values are dimensionless

Address	Name	Description	Type of data	Range
1522-1523	SPEED_SET(0)	Set value of the speed for the little finger	short	0-1000
1524-1525	SPEED_SET(1)	Set value of the speed for the ring finger	short	0-1000
1526-1527	SPEED_SET(2)	Set value of the speed for the middle finger	short	0-1000
1528-1529	SPEED_SET(3)	Set value of the speed for the index finger	short	0-1000
1530-1531	SPEED_SET(4)	Set value of the speed for thumb bending	short	0-1000
1532-1533	SPEED_SET(5)	Set value of the speed for thumb rotation	short	0-1000

Speed = 1000: This means that it will take 800ms for the fingers in the no-load state to move from a large angle to the minimum angle. If there is a heavy load, the actual speed will be somewhat lower than this value.

2.4.14 POS_ACT(m) (actual value of the actuator position for each DOF)

This register group consists of six registers corresponding to the current actual position of the actuator of the dexterous hand for 6 DOF, with details in the table below. These parameters are read only.Range values are dimensionless

Address	Name	Description	Type of data	Range
1534-1535	POS_ACT(0)	Actual actuator position value of the little finger	short	0-2000
1536-1537	POS_ACT(1)	Actual actuator position value of the ring finger	short	0-2000
1538-1539	POS_ACT(2)	Actual actuator position value of the middle finger	short	0-2000
1540-1541	POS_ACT(3)	Actual actuator position value of the index finger	short	0-2000
1542-1543	POS_ACT(4)	Actual actuator position value of thumb bending	short	0-2000
1544-1545	POS_ACT(5)	Actual actuator position value of thumb rotation	short	0-2000

0: Minimum actuator stroke, corresponding to the maximum finger angle;

2000: Maximum actuator stroke, corresponding to the minimum finger angle

2.4.15 ANGLE_ACT(m) (actual angle for each DOF)

This register group consists of six registers corresponding to the actual angle of the dexterous hand for 6 DOF, with details in the table below. For the definition of angle for each DOF, refer to the description of the register group "1.7DEFAULT_ANGLE_SET(m)". These parameters are read only.

Address	Name	Description	Type of data	Range
1546-1547	ANGLE_ACT(0)	Actual angle of the little finger	short	0-1000
1548-1549	ANGLE_ACT(1)	Actual angle of the ring finger	short	0-1000
1550-1551	ANGLE_ACT(2)	Actual angle of the middle finger	short	0-1000
1552-1553	ANGLE_ACT(3)	Actual angle of the index finger	short	0-1000
1554-1555	ANGLE_ACT(4)	Actual angle of thumb bending	short	0-1000
1556-1557	ANGLE_ACT(5)	Actual angle of thumb rotation	short	0-1000

2.4.16FORCE_ACT(m) (actual force applied to each finger)

This register group consists of six registers corresponding to the actual force of the dexterous hand for 6 DOF, with details in the table below. These parameters are read only.Unit: g.

Address	Name	Description	Type of data	Range
1582-1583	FORCE_ACT(0)	Actual force applied to the little finger	short	0-1000
1584-1585	FORCE_ACT(1)	Actual force applied to the ring finger	short	0-1000
1586-1587	FORCE_ACT(2)	Actual force applied to the middle finger	short	0-1000
1588-1589	FORCE_ACT(3)	Actual force applied to the index finger	short	0-1000
1590-1591	FORCE_ACT(4)	Actual bending force applied to the thumb	short	0-1000
1592-1593	FORCE_ACT(5)	Actual rotation force applied to the thumb	short	0-1000

2.4.17CURRENT(m) (actuator current value for each DOF)

This register group consists of six registers corresponding to the current value of six actuators of the dexterous hand, with details in the table below. These parameters are read only.

Address	Name	Description	Type of data	Range	Unit
1594-1595	CURRENT(0)	Current value of the actuator for the little finger	short	0-1000	mA
1596-1597	CURRENT(1)	Current value of the actuator for the ring finger	short	0-1000	mA
1598-1599	CURRENT(2)	Current value of the actuator for the middle finger	short	0-1000	mA
1600-1601	CURRENT(3)	Current value of the actuator for the index finger	short	0-1000	mA
1602-1603	CURRENT(4)	Current value of the actuator	short	0-1000	mA

		for thumb bending			
1604-1605	CURRENT(5)	Current value of the actuator for thumb rotation	short	0-1000	mA

2.4.18 ERROR(m) (error codes of each actuator)

This register group consists of six registers corresponding to error codes of six actuators of the dexterous hand, with details in the table below. These parameters are read only.

Address	Name	Description	Type of data
1606	ERROR(0)	Error code of the actuator for the little finger	byte
1607	ERROR (1)	Error code of the actuator for the ring finger	byte
1608	ERROR (2)	Error code of the actuator for the middle finger	byte
1609	ERROR (3)	Error code of the actuator for the thumb	byte
1610	ERROR (4)	Error code of the actuator for thumb bending	byte
1611	ERROR (5)	Error code of the actuator for thumb rotation	byte

The implication of error codes is listed below. For example, if ERROR (3) is 0x06 (00000110 in the binary system), it means that over temperature and over-current errors occur in the actuator of the index finger.

	Description
Bit0	Locked-rotor error
Bit1	Over temperature error
Bit2	Over-current error
Bit3	Abnormal operation of the motor
Bit4	Communication error

2.4.19 STATUS(m) (status information for each DOF)

This register group consists of six registers corresponding to the status information of the dexterous hand for 6 DOF, with details in the table below. These parameters are read only.

Address	Name	Description	Type of data
1612	STATUS(0)	Status information of the little finger	byte

1613	STATUS(1)	Status information of the ring finger	byte
1614	STATUS(2)	Status information of the middle finger	byte
1615	STATUS(3)	Status information of the index finger	byte
1616	STATUS(4)	Status information of thumb bending	byte
1617	STATUS(5)	Status information of thumb rotation	byte

Meaning of status codes:

Status code	Description
0	Unclenching
1	Grasping
2	Stop after reaching the target position
3	Stop after reaching the defined force control value
5	Stop due to current protection
6	Stop due to locked-rotor of the actuator
7	Stop due to actuator fault

2.4.20 TEMP(m) (temperature of each actuator)

This register group consists of six registers corresponding to the temperature value of six actuators of the dexterous hand, with details in the table below. These parameters are read only.

Address	Name	Description	Type of data	Range	Unit
1618	TEMP(0)	Temperature value of the actuator for the little finger	byte	0-100	℃
1619	TEMP(1)	Temperature value of the actuator for the ring finger	byte	0-100	℃
1620	TEMP(2)	Temperature value of the actuator for the middle finger	byte	0-100	℃
1621	TEMP(3)	Temperature value of the actuator for the index finger	byte	0-100	℃
1622	TEMP(4)	Temperature value of the actuator for thumb bending	byte	0-100	℃
1623	TEMP(5)	Temperature value of the actuator for thumb rotation	byte	0-100	℃

2.4.21 Register related to the action sequence

Address	Meaning	Abbreviation	Length	Permission (read / write)
2000	Current action sequence check code 1	ACTION_SEQ_CHECKDATA1	1byte	W/R
2001	Current action sequence check code 2	ACTION_SEQ_CHECKDATA2	1byte	W/R
2002	Total number of steps in the current action sequence	ACTION_SEQ_STEPNUM	1byte	W/R
2016	Setting of the current action sequence: Step 0	ACTION_SEQ_STEP(0)	19short(38byte)	W/R
2054	Setting of the current action sequence: Step 1	ACTION_SEQ_STEP(1)	19short(38byte)	W/R
2092	Setting of the current action sequence: Step 2	ACTION_SEQ_STEP(2)	19short(38byte)	W/R
2130	Setting of the current action sequence: Step 3	ACTION_SEQ_STEP(3)	19short(38byte)	W/R
2168	Setting of the current action sequence: Step 4	ACTION_SEQ_STEP(4)	19short(38byte)	W/R
2206	Setting of the current action sequence: Step 5	ACTION_SEQ_STEP(5)	19short(38byte)	W/R
2244	Setting of the current action sequence: Step 6	ACTION_SEQ_STEP(6)	19short(38byte)	W/R
2282	Setting of the current action sequence: Step 7	ACTION_SEQ_STEP(7)	19short(38byte)	W/R
2320	Current action sequence index number	ACTION_SEQ_INDEX	1byte	W/R
2321	Save the current action sequence	SAVE_ACTION_SEQ	1byte	W/R
2322	Run the current action sequence	ACTION_SEQ _RUN	1byte	W/R

An action sequence refers to a set of continuous gestures. A set of action sequence data includes the angle, speed, force control, and delay time (ms) settings of several gestures (up

to 8). The dexterous hand can make each gesture based on such data, sequence and delay time, and all gestures will constitute a set of continuous action sequences.

The memory of the dexterous hand can store 40 sets of action sequence data. Different action sequences are identified by the index number. Such data is read in a slightly different way.

Read the action sequence data:

1. First of all, set the current action ID register (ACTION_SEQ_INDEX);

2. Read the data in ACTION_SEQ_CHECKDATA1 and ACTION_SEQ_CHECKDATA2. If the value of the register "ACTION_SEQ_CHECKDATA1" is 0x90 and that of the register "ACTION_SEQ_CHECKDATA10" is xEB, it means that such action sequence data is valid; otherwise, such action sequence data is invalid.

3. After it is determined in the previous step that the data is valid, read the data in ACTION_SEQ_STEPNUM and ACTION_SEQ_STEP(0-7).

Write the action sequence data:

1. First of all, set the current action ID register (ACTION_SEQ_INDEX);

2. Write the desired data to the register "ACTION_SEQ_STEPNUM 、 ACTION_SEQ_STEP(0-7)".

3. Write 0x90 and 0xEB to the registers "ACTION_SEQ_CHECKDATA1" and "ACTION_SEQ_CHECKDATA2".

4. Set the action sequence library save register "SAVE_ACTION_SEQ" to 1. The dexterous hand will write the data in ACTION_SEQ_CHECKDATA1, ACTION_SEQ_CHECKDATA2, ACTION_SEQ_STEPNUM, and ACTION_SEQ_STEP (0-7) to the Flash address corresponding to the index number.

Run the action sequences:

1. First of all, set the current action ID register (ACTION_SEQ_INDEX) to the index number of the action sequence to be run;

2. Set the register "ACTION_SEQ _RUN" to 1. The dexterous hand will start to execute the action sequence. After running, the register value will be automatically reset to zero.