THE DEXTEROUS HAND
RH56DFTP USER MANUAL

Beijing Inspire-Robots Technology Co., Ltd. | December 2024
ID: PRJ-02-TS-U-010 | Page Number: 27 pages | Version: V 1.0.0

Table of Contents

RH56DFTP Dexterous Hand
User Manual

1 Product Overview

1.1 Product Features

RH56DFTP Series Dexterous Hand is a mechanical dexterous hand (hereinafter referred to as "Dexterous Hand") designed to integrate Micro linear servo actuator with small volume and large torque. These dexterous hands feature six Micro linear servo actuators equipped with sensitive pressure sensors. By setting different force control thresholds, precise gripping force control can be achieved. High-precision tactile sensors are integrated into the fingertips, finger pads, and palm areas, providing real-time feedback on finger gripping information. User interfaces include Ethernet, RS485, and CAN2.0 communication interface. Concise and efficient interface control commands enable users to quickly manipulate the dexterous hands. Their superior performance makes them suitable for applications in service robots, educational tools, industrial automation, and other fields.

1.2 Performance parameters

Total number of joints	12
Degree of freedom (DOF)	6
Number of force sensors	6
Resolution of force sensor	0.1 N
Number of tactile sensors	5-17
Repeated fingertip positioning accuracy	±0.2mm
Five-finger gripping force	30 N
Operating voltage	24V
Quiescent current	0.2A @24V
Average current for no-load operation	1.2A @24V
Current for maximum gripping force	4.5A @24V
Communication interface	Modbus TCP + CAN2.0 or Modbus TCP + RS485

1.3 Electric connection

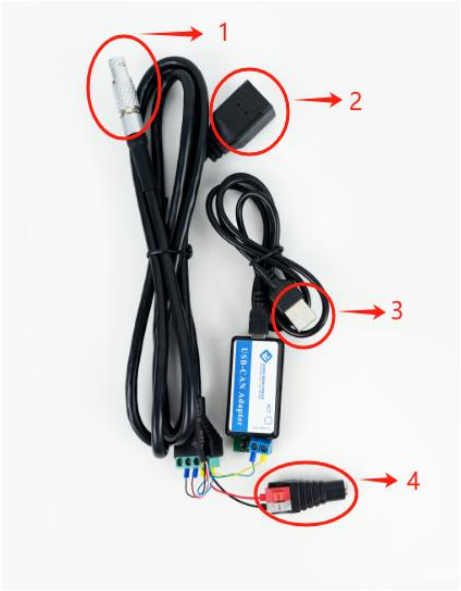
1.3.1. Pin definition

The wrist of the dexterous hands uses a 1BZ1G08CLL00000 aviation plug interface (compatible with the Lemo FGG-1B-8P aviation plug interface). The physical image is shown below, and the function definitions are as follows.

S/N.	Definition
1	GND
2	VCC (24V)
3	485_A/ CAN_H
4	485_B/ CAN_L
5	TX+ (Data Send Positive)
6	TX- (Data Send Negative)
7	RX+ (Data Receive Positive)
8	RX- (Data Receive Negative)

The external cables of the dexterous hands are shown below, with the following function definitions:

Cable Description	Function Definition
White	TX+ (Data Send Positive)
Blue	TX- (Data Send Negative)
Red (thin)	RX+ (Data Receive Positive)
Black (thin)	RX- (Data Receive Negative)
Yellow	485_A/ CAN_H
Green	485_B/ CAN_L
Red (thick)	VCC (24V)
Black (thick)	GND



Cable Interface Definitions:

1-Aviation Plug Interface: Connects to the dexterous hand's flange aviation plug

2-Network Interface: Connects to an Ethernet cable

3-CAN/RS485 Interface

4-Power Interface

1.3.2. Communication mode

When the RS485 interface is used, a particular conversion circuit can realize parallel connection of 254 dexterous hands to the same bus. When the CAN interface is used, multiple dexterous hands (theoretically up to 16383 dexterous hands) can be connected to the same bus in parallel. For the purpose of communication control, the dexterous hands connected to the same bus should be configured with different ID. When using Modbus TCP control, the dexterous hands are controlled through IP addresses and port numbers. (Default IP address: 192.168.11.210, port number: 6000)

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.

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 (readable in groups; groups refer to several registers with adjacent addresses); the upper system sends a read instruction to the dexterous hand (including the first 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 first 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 115200bps, 8 data bits, 1 stop bit, and no parity.

2.2.1 Register reading

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
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
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]	0xD007	Converted to the integer data 2000 (0x07D0)
byte[17] byte[18]	0x0000	Converted to the integer data 0 (0x0000)
byte[19]	0x98	Checksum

From this response frame, we can obtain the actual values for DOF: 100, 100, 100, 100, 2000, and 0.

2.2.2 Register writing

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 parameter saving (the register "SAVE" is set to 1), the result information frame will be returned one second after the dexterous hand returns the response frame. The content of the result information frame is shown below:

	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]	0xED	Low-order 8 bits in the start address of the register "SAVE"
byte[6]	0x03	High-order 8 bits in the start address of the register "SAVE"
byte[7]	**	0x00 saving successful; 0xFF saving failure
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, 2000, 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]	0xD007	Converted to the integer data 2000 (0x07D0)
byte[17] byte[18]	0x0000	Converted to the integer data 0 (0x0000)
byte[19]	0x5C	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 Register reading/writing of CAN

The default baud rate is 1000K. It adopts the extended identifier and data frame format. It does not use the standard identifier and remote frame. The extended identifier has 29 bits, which are defined from low-order to high-order bits as follows:

bit0-13: Hand_ID supports up to 16383 devices;

bit14-25: Start address of the register to be operated;

bit26-28: Read/write flag bits; 0 indicates reading a register of the dexterous hand; 1 indicates writing to a register of the dexterous hand; 4 indicates reading the wrist register of the dexterous hand; 5 indicates writing to the wrist register of the dexterous hand.

Bit29-31: Reserved flag bits

Identifier	bit 31-29	bit 26-28	bit14-25	bit 13-0
Meaning	Reserve	W/R 0:R - Reading a register of the dexterous hand 1:W - Writing to a register of the dexterous hand 4: R - Reading the wrist register of the dexterous hand 5:W - Writing to the wrist register of the dexterous hand	Register address	Hand_ID

2.3.1 Register reading

The identifier settings of register reading are as follows:

Identifier	bit 31-29	bit 26-28	bit14-25	bit 13-0
Meaning	Reserve	0	Address	Hand_ID

The data length is 1 byte.

Data content: Length of the register data to be read

After the dexterous hand receives and correctly analyzes the aforesaid instruction, it will return the following frame:

Identifier:

Identifier	bit 31-29	bit 26-28	bit14-25	bit 13-0
Meaning	Reserve	0	Address	Hand_ID

Data length: Length of the data to be returned to the register

Data content: Register data

For example, if we want to read the current actual angle of the index finger of the dexterous hand (ID=1), the following frame should be sent to the CAN bus:

Identifier: The binary number is 0000 0001 1000 0100 0000 0000 0000 0001.

bit 31-29	bit 26-28	bit14-25	bit 13-0

0	0	Address of ANGLE_ACT(3) = 1552; The binary number is 011000010000.	1

Data length: 1

Data content: 2; the current actual angle of the index finger is an integer data; the data length is 2 bytes.

The dexterous hand returns the following frame:

Identifier: The binary number is 0000 0001 1000 0100 0000 0000 0000 0001.

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	0	Address of ANGLE_ACT(3) = 1552; The binary number is 011000010000.	1

Data length: 2

Data content: The current actual angle of the index finger is POS_ACT(3), which is an integer. The following data should be converted to integers (low-order bytes are followed by high-order bytes). After high-order and low-order bytes are exchanged, the hexadecimal number is 0x01F4, and the decimal number is 500 (the current actual angle of the index finger is 500).

byte0	byte1
0xF4	0x01

2.3.2 Register writing

The identifier settings of register writing are as follows:

Identifier	bit 31-29	bit 26-28	bit14-25	bit 13-0
Meaning	Reserve	1	Address	Hand_ID

Data length: Length of the data to be written to the register

Data content: Data to be written to the register

After the dexterous hand receives and correctly analyzes the aforesaid instruction, it will return the following frame:

Identifier	bit 31-29	bit 26-28	bit14-25	bit 13-0
Meaning	Reserve	1	Address	Hand_ID

Data length: 1

For example, if we want to set the angle of the index finger of the dexterous hand (ID=1) to 600, the following frame should be sent to the CAN bus:

Identifier: The binary number is 0000 0101 0111 0101 0000 0000 0000 0001.

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	1	Address of ANGLE_SET(3) = 1492; The binary number is 010111010100.	1

Data length: 2

Data content: The current actual angle of the index finger is ANGLE_SET(3), which is an integer. It is necessary to split the integer data into high-order and low-order bytes and then exchange these bytes; that is, 600 (0x0258) is converted to the data content below:

byte0	byte1
0x58	0x02

The dexterous hand returns the following frame:

Identifier: The binary number is 0000 0101 0111 0101 0000 0000 0000 0001.

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	1	Address of ANGLE_SET (3) = 1492; The binary number is 01 0111 0101 00	1

Data length: 1

2.4 Summary of Modbus RTU 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). Support reading holding register (function code: 0x03), preset single register (function code: 0x06), and preset multiple registers (function code: 0x10).

2.4.1. Read Registers - Function Code 03 (0x03)

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)	CRC
	0x01	0x03	0x6B	0x00	0x00	0x02	XXXX

Explanation: Read the holding register of Slave Station 1# (0x01), 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 1# (0x01), 0x006B-0x006C; there are totally two registers. The value of 0x006B register is 0x0001, and that of 0x0062 register is 0x0002.

2.4.2. Write Single Register - Function Code 06 (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
	0x01	0x06	0x00	0x6B	0x10	0x00	XXXX

Explanation: Set the holding register of Slave Station 1# (0x01), 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
	0x01	0x06	0x00	0x00	0x00	0x00	XXXX

2.4.3. Write Multiple Registers - Function Code 16 (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	XXXX

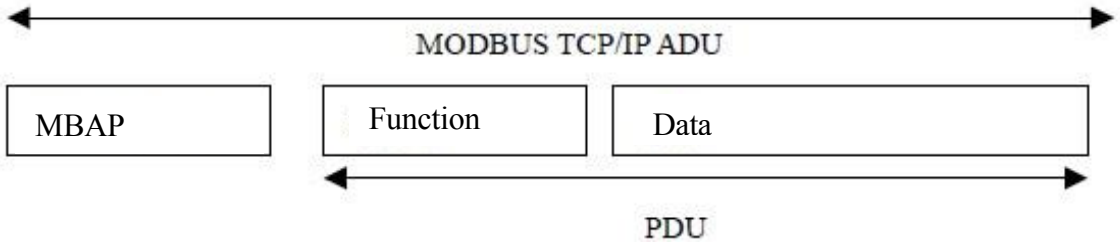
Explanation: Set the holding register of Slave Station 1# (0x01), with register start address = 0x0001, number of registers = 0x0002, byte count of data content = 0x04, 0x000A and 0x0102 as data content.

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
	0x01	0x10	0x00	0x01	0x00	0x02	XXXX

2.5 Summary of Modbus TCP Protocol

The Ethernet conversion module adopts the Modbus TCP protocol, which is a Modbus protocol based on Ethernet TCP/IP introduced by Schneider company (It is a message transmission protocol at the application layer that utilizes a master/slave communication mode).

The data frame of Modbus TCP can be divided into two parts: MBAP and PDU.



MBAP Header:

Field	Length	Description	Client	Server
Transaction Identifier	2 bytes	Identifier for a MODBUS request/response transaction	Initiated by Client	Copied by Server from received request
Protocol Identifier	2 bytes	0 = MODBUS Protocol	Initiated by Client	Copied by Server from received request
Length	2 bytes	Number of the following bytes	Initiated by Client (for requests)	Initiated by Server (for responses)
Unit Identifier	2 bytes	Remote slave station identifier on a serial link or other bus	Initiated by Client	Copied by Server from received request

Note: The Length includes the Unit Identifier, Function Code, and the number of data bytes.

The Unit Identifier is set to 0xFF. For TCP, the Unit Identifier is not utilized (TCP relies on IP addressing for destination identification);

PDU Frame Structure:

The PDU comprises a Function Code and Data. The Function Code occupies 1 byte, while the Data length varies depending on the specific function in use.

Modbus offers a suite of 8 function codes, among which we employ three: 03, 06, and 16.

Code	English Name	Bit/Word Operation	Quantity of Operations
01	READ COIL STATUS	Bit Operation	Single or Multiple
02	READ INPUT STATUS	Bit Operation	Single or Multiple

03	READ HOLDING REGISTER	Word Operation	Single or Multiple
04	READ INPUT REGISTER	Word Operation	Single or Multiple
05	WRITE SINGLE COIL	Bit Operation	Single
06	WRITE SINGLE REGISTER	Word Operation	Single
15	WRITE MULTIPLE COIL	Bit Operation	Multiple
16	WRITE MULTIPLE REGISTER	Word Operation	Multiple

2.5.1 Read Registers - Function Code 03 (0x03)

Request

Function code	1 byte	0x03
Start address	2 bytes	0x0000 to 0xFFFF
Number of registers	2 bytes	1 to 125 (0x7D)

Response

Function code	1 byte	0x03
Byte	1 byte	2*N
Number of registers	N*2 bytes	

Example: This is an example of a request to read holding registers 108 to 110. The content of register 108 is represented as two hexadecimal byte values, 02 2B, or decimal 555. The contents of registers 109 and 110 are represented as hexadecimal 00 00 and 00 64, respectively, or decimal 0 and 100.

Request		Response	
Field Name	Hexadecimal	Field Name	Hexadecimal
Function	03	Function	03
High start address	00	Byte	06
Low start address	6B	Register Value Hi (108)	02
High Register Number	00	Register Value Lo (108)	2B
Low Register Number	03	Register Value Hi (109)	00
		Register Value Lo (109)	00
		Register Value Hi (110)	00
		Register Value Lo (110)	64

2.5.2 Write Single Register - Function Code 06 (0x06)

Request

Function code	1 byte	0x06
Register address	2 bytes	0x0000 to 0xFFFF

Register Value	2 bytes	0x0000 to 0xFFFF

Response

Function code	1 byte	0x06
Register address	2 bytes	0x0000 to 0xFFFF
Register Value	N*2 bytes	0x0000 to 0xFFFF

Example: This is an example of a request to write the hexadecimal value 00 03 to register 2.

Request		Response	
Field Name	Hexadecimal	Field Name	Hexadecimal
Function	06	Function	06
Register Address Hi	00	Output Address Hi	00
Register address Lo	01	Output Address Lo	01
Register Value Hi	00	Output Value Hi	00
Register Value Lo	03	Output Value Lo	03

2.5.3 Write Multiple Registers - Function Code 16 (0x10)

Request

Function code	1 byte	0x10
Start address	2 bytes	0x0000 to 0xFFFF
Number of registers	2 bytes	0x0001 to 0x0078
Byte	1 byte	2*N
Register Value	N*2 bytes	

Response

Function code	1 byte	0x10
Start address	2 bytes	0x0000 to 0xFFFF
Number of registers	2 bytes	1 to 123 (0x7B)

Example: This is an example of a request to write the hexadecimal values 00 0A and 01 02 to the first two registers starting from register 2.

Request		Response	
Field Name	Hexadecimal	Field Name	Hexadecimal
Function	10	Function	10
High start address	00	High start address	00
Low start address	01	Low start address	01

Number of Registers Hi	00	Number of Registers Hi	00
Number of Registers Lo	02	Number of Registers Lo	02
Byte	04		
Register Value Hi	00		
Register Value Lo	0A		
Register Value Hi	01		
Register Value Lo	02		

2.6 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	ID of the dexterous hand	HAND_ID	1 byte	W/R
1002	Baud rate setting	REDU_RATIO	1 byte	W/R
1004	Error clearance	CLEAR_ERROR	1 byte	W/R
1005	Saving data in Flash	SAVE	1 byte	W/R
1006	Restoring factory defaults	RESET_PARA	1 byte	W/R
1009	Force sensor calibration	GESTURE_FORCE_CLB	1 byte	W/R
1032	Set value of power-on speed for each DOF	DEFAULT_SPEED_SET(m)	6 short (12 bytes)	W/R
1044	Set value of power-on force control threshold for each DOF	DEFAULT_FORCE_SET(m)	6 short (12bytes)	W/R
1474	Set value of the actuator position for each DOF	POS_SET(m)	6 short (12bytes)	W/R
1486	Set value of the angle for each DOF	ANGLE_SET(m)	6 short (12bytes)	W/R
1498	Set value of force control threshold for each DOF	FORCE_SET(m)	6 short (12bytes)	W/R
1522	Set value of the speed for each DOF	SPEED_SET(m)	6 short (12bytes)	W/R
1534	Actual value of the actuator position for each DOF	POS_ACT(m)	6 short (12bytes)	R
1546	Actual angle for each degree of freedom (DOF)	ANGLE_ACT(m)	6 short (12bytes)	R
1582	Actual force applied to each finger	FORCE_ACT(m)	6 short (12bytes)	R
1594	Actuator current value for each DOF	CURRENT(m)	6 short (12bytes)	R

1606	Error codes of the actuator for each DOF	ERROR(m)	6 bytes	R
1612	Status information for each DOF	STATUS(m)	6 bytes	R
1618	Actuator temperature for each DOF	TEMP(m)	6 bytes	R
1700	First field of IP address, default 192, Range 0-255, become effective after re-power on	IP_PART1	1 byte	W/R
1701	Second field of IP address, default 168, Range 0-255, become effective after re-power on	IP_PART2	1 byte	W/R
1702	Third field of IP address, default 11, Range 0-255, become effective after re-power on	IP_PART3	1 byte	W/R
1703	Fourth field of IP address, default 210, Range 0-255, become effective after re-power on	IP_PART4	1 byte	W/R
3000	Little finger tactile sensor	FINGERONE_TOUCH	370 bytes	R
3370	Ring finger tactile sensor	FINGERTWO_TOUCH	370 bytes	R
3740	Middle finger tactile sensor	FINGERTHE_TOUCH	370 bytes	R
4110	Index finger tactile sensor	FINGERFOR_TOUCH	370 bytes	R
4480	Thumb tactile sensor	FINGERFIV_TOUCH	420 bytes	R
4900	Palm tactile sensor	FINGERPALM_TOUCH	224 bytes	R

2.6.1 ID of the dexterous hand

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 ID.

2.6.2 Baud rate setting

RS485 Interface: default value = 0; range: 0-3; it cannot be saved.

0: Baud rate = 115200

1: Baud rate = 57600

2: Baud rate = 19200

3: Baud rate = 921600

CAN Interface: default value = 0; range: 0-1; it cannot be saved.

0: Baud rate = 1000K

1: Baud rate = 500K

2.6.3 Error clearance

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

After 1 is written, clearable errors (actuator errors such as locked-rotor, overcurrent, 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.

2.6.4 Save parameters to 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.

2.6.5 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.

2.6.6 Force sensor calibration

When the user sets the parameter to 1, the dexterous hand initiates the force sensor calibration process.

Note: The dexterous hand must be maintained in a palm open state during calibration and fingers cannot touch any object.

2.6.7 Target gesture sequence number

Default value = 0xFF; range: 1- 40; it cannot be saved.

When the user modifies this register value, the dexterous hand will make a gesture corresponding the serial number.

1-13 gestures are fixed (saved before product delivery), i.e., holding open, two fingers touching each other, fist clenching, etc.

14-40 gestures are user-defined (the data is stored in the register "USER_DEF_ANGLE (K, m)") and can be modified and saved by the user.

2.6.8 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.

Address	Name	Description	Data type	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 600ms for the fingers in the no-load state to move from a maximum angle to the minimum angle. If there is a heavy load, the actual speed will be somewhat lower than this value.

2.6.9 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.

Address	Name	Description	Data type	Range
1044-1045	DEFAULT_FORCE_SET(0)	Initial power-on force control of the little finger	short	0-3000
1046-1047	DEFAULT_FORCE_SET(1)	Initial power-on force control of the ring finger	short	0-3000
1048-1049	DEFAULT_FORCE_SET(2)	Initial power-on force control of the middle finger	short	0-3000
1050-1051	DEFAULT_FORCE_SET(3)	Initial power-on force control of the index finger	short	0-3000
1052-1053	DEFAULT_FORCE_SET(4)	Initial power-on force control of thumb bending	short	0-3000
1054-1055	DEFAULT_FORCE_SET(5)	Initial power-on force control of thumb rotation	short	0-3000

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.)

2.6.10 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.

Address	Name	Description	Data type	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.

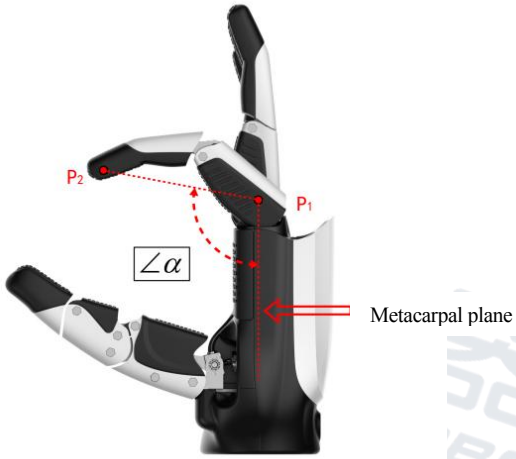
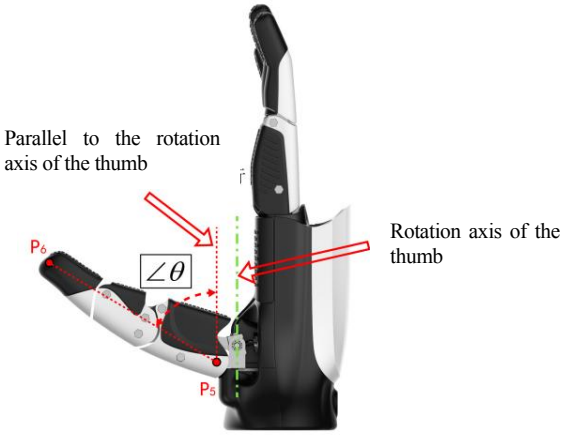
2.6.11 Set value of the angle for each DOF

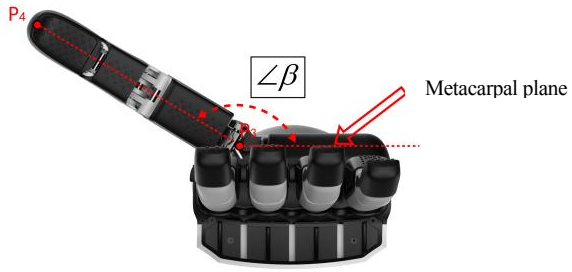
This register group consists of six registers corresponding to the set value of the angle of the dexterous hand for 6 DOF, with details in the table below. When the angle value for a degree of freedom (DOF) is set to the data within 0-1000, the corresponding finger will take an immediate action. If the 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).

Address	Name	Description	Data type	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	-1, 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		20°-176°
Bending angle of the thumb		-13°-70°

Rotation angle of the thumb		90°-165°

2.6.12 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. 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);) 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.

Address	Name	Description	Data type	Range
1498-1499	FORCE_SET (0)	Set value of force control for the little finger	short	0-3000
1500-1501	FORCE_SET (1)	Set value of force control for the ring finger	short	0-3000
1502-1503	FORCE_SET (2)	Set value of force control for the middle finger	short	0-3000
1504-1505	FORCE_SET (3)	Set value of force control for the index finger	short	0-3000
1506-1507	FORCE_SET (4)	Set value of bending force control for the thumb	short	0-3000
1508-1509	FORCE_SET (5)	Set value of rotation force control for the thumb	short	0-3000

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.)

2.6.13 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.

Address	Name	Description	Data type	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 600ms 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.

2.6.14 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 can be read only.

Address	Name	Description	Data type	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 the thumb bending	short	0-2000
1544-1545	POS_ACT(5)	Actual actuator position value of the 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.

2.6.15 Actual angle for each degree of freedom (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 can be read only.

Address	Name	Description	Data type	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

2.6.16 Actual force applied to each finger

This register group consists of six registers corresponding to the actual force applied to six fingers of the dexterous hand, unit: g, with details in the table below. These parameters can be read only.

Address	Name	Description	Data type	Range
1582-1583	FORCE_ACT(0)	Actual force applied to the little finger	short	-4000-4000
1584-1585	FORCE_ACT(1)	Actual force applied to the ring finger	short	-4000-4000
1586-1587	FORCE_ACT(2)	Actual force applied to the middle finger	short	-4000-4000
1588-1589	FORCE_ACT(3)	Actual force applied to the index finger	short	-4000-4000
1590-1591	FORCE_ACT(4)	Actual bending force applied to the thumb	short	-4000-4000
1592-1593	FORCE_ACT(5)	Actual rotation force applied to the thumb	short	-4000-4000

2.6.17 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 can be read only.

Address	Name	Description	Data type	Range	Unit
1594-1595	CURRENT(0)	Current value of the actuator for the little finger	short	0-2000	mA
1596-1597	CURRENT(1)	Current value of the actuator for the ring finger	short	0-2000	mA
1598-1599	CURRENT(2)	Current value of the actuator for the middle finger	short	0-2000	mA
1600-1601	CURRENT(3)	Current value of the actuator for the index finger	short	0-2000	mA
1602-1603	CURRENT(4)	Current value of the actuator for the thumb bending	short	0-2000	mA
1604-1605	CURRENT(5)	Current value of the actuator for the thumb rotation	short	0-2000	mA

2.6.18 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 can be read only.

Address	Name	Description	Data type
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 index finger	byte
1610	ERROR (4)	Error code of the actuator for the thumb bending	byte
1611	ERROR (5)	Error code of the actuator for the 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 overcurrent errors occur in the actuator of the index finger.

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

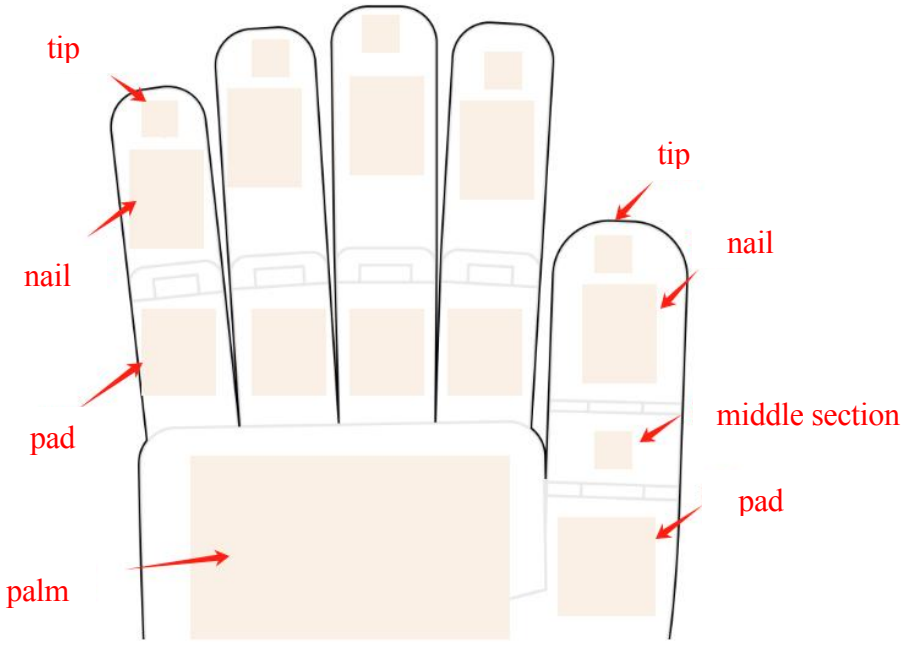
2.6.19 Temperature of each actuator

Address	Name	Description	Data type	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 the thumb bending	byte	0-100	℃
1623	TEMP(5)	Temperature value of the actuator for the thumb rotation	byte	0-100	℃

This register group consists of six temperature values of six actuators of the dexterous hand, with details in the table above. These parameters can be read only.

2.6.20 Description of Tactile Sensor Data Format

The tactile distribution of the dexterous hand is illustrated below (with the light yellow sections indicating the locations of the tactile sensors).



Address	Meaning	Row and column count	Length	Permission (read / write)
3000-3017	Tactile data for little finger tip	3*3	18 bytes	R
3018-3209	Tactile data for little finger nail	12*8	192 bytes	R
3210-3369	Tactile data for little finger pad	10*8	160 bytes	R
3370-3387	Tactile data for ring finger tip	3*3	18 bytes	R
3388-3579	Tactile data for ring finger nail	12*8	192 bytes	R
3580-3739	Tactile data for ring finger pad	10*8	160 bytes	R
3740-3757	Tactile data for middle finger tip	3*3	18 bytes	R
3758-3949	Tactile data for middle finger nail	12*8	192 bytes	R
3950-4109	Tactile data for middle finger pad	10*8	160 bytes	R
4110-4127	Tactile data for index finger tip	3*3	18 bytes	R
4128-4319	Tactile data for index finger nail	12*8	192 bytes	R
4320-4479	Tactile data for index finger pad	10*8	160 bytes	R
4480-4497	Tactile data for thumb tip	3*3	18 bytes	R
4498-4689	Tactile data for thumb nail	12*8	192 bytes	R
4690-4707	Tactile data for thumb middle section	3*3	18 bytes	R
4708-4899	Tactile data for thumb pad	12*8	192 bytes	R
4900-5123	Tactile data for palm	8*14	224 bytes	R

Each tactile point in the above data is represented by a 16-bit integer (composed of two bytes in little-endian format, with the low byte first and the high byte second), with a value range of 0-4095.

For the fingers, the data points correspond to actual positions as follows: Data Point 1 corresponds to the first row, first column; Data Point 2 corresponds to the first row, second column; Data Point 3 corresponds to the first row, third column; Data Point 4 corresponds to the first row, fourth column; Data Point 5 corresponds to the first row, fifth column; Data Point 6 corresponds to the first row, sixth column; Data Point 7 corresponds to the first row, seventh column; Data Point 8 corresponds to the first row, eighth column; Data Point 9 corresponds to the second row, first column; Data Point 10 corresponds to the second row, second column; and so on.

For the palm, the data points correspond to actual positions as follows: Data Point 1 corresponds to the eighth row, first column; Data Point 2 corresponds to the seventh row, first column; Data Point 3 corresponds to the sixth row, first column; Data Point 4 corresponds to the fifth row, first column; Data Point 5 corresponds to the fourth row, first column; Data Point 6 corresponds to the third row, first column; Data Point 7 corresponds to the second row, first column; Data Point 8 corresponds to the first row, first column; Data Point 9 corresponds to the eighth row, second column; Data Point 10 corresponds to the seventh row, second column; and so on.