X-Hand 1 Product Manual

V1.0	2024/10/15

CONTENTS

1. Product Outline

1.1 Product Description

XHAND1 is the first product-level robotic dexterous hand launched by RobotEra with high degree of freedom, high performance and high intelligence.

1.2 Product Characteristics

- Five finger joints full drive: Each finger joint is driven by gears and motors to realize independent motion control of each joint.
- High Degrees of Freedom: 12 active degrees of freedom (DoF), including 3 for the thumb and index finger and 2 for each of the other three fingers. The index finger is supported to swing sideways, and the thumb has a wide range of motion, which can reproduce a variety of high-precision gripping operations of the human hand.
- High Fractional Tactile Sensor: Each finger can be equipped with a high fractional (>100 points) tactile sensor to provide accurate 3D force tactile and temperature information; in addition to traditional pressure tactile sensing of tangential force, high spatial fractional sensing of object edges and shapes.

- Powerful Grip:4-finger grip upper limit is 8ON, single finger maximum load 50N.
- High Intelligence: Pairing robotic arm and robot through reinforcement learning, imitation learning full linkage verification.

1.3 Software support

- Development Kit (SDK): XHAND provides standard Linux system SDK and ROS-compatible applications to facilitate algorithm development.
- Remote Teleoperation: Provides VisionPro teleoperation support, compatible with gloves and vision mapping, for easy customer operation testing and training data collection.
- Embedded system development: Equipped with standard evaluation applications for the host computer, it supports customer debugging and embedded system development.

2. Version overview

releases	dates	revised record
V1.0	2024/10/15	first edition

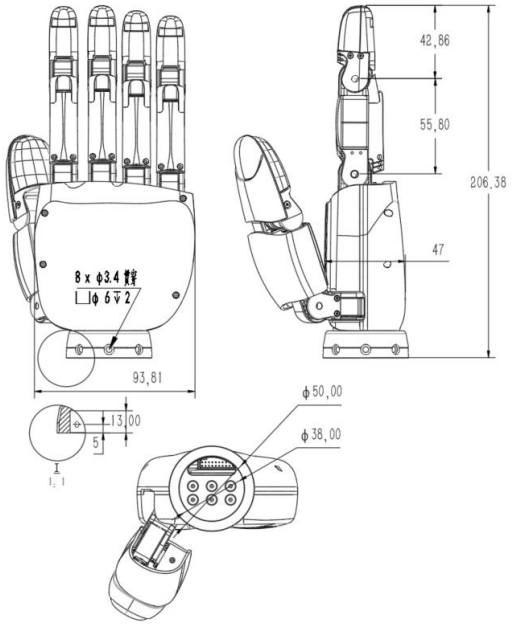
3. Technical specifications

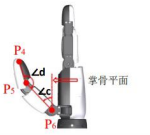
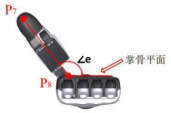
3.1 Specifications

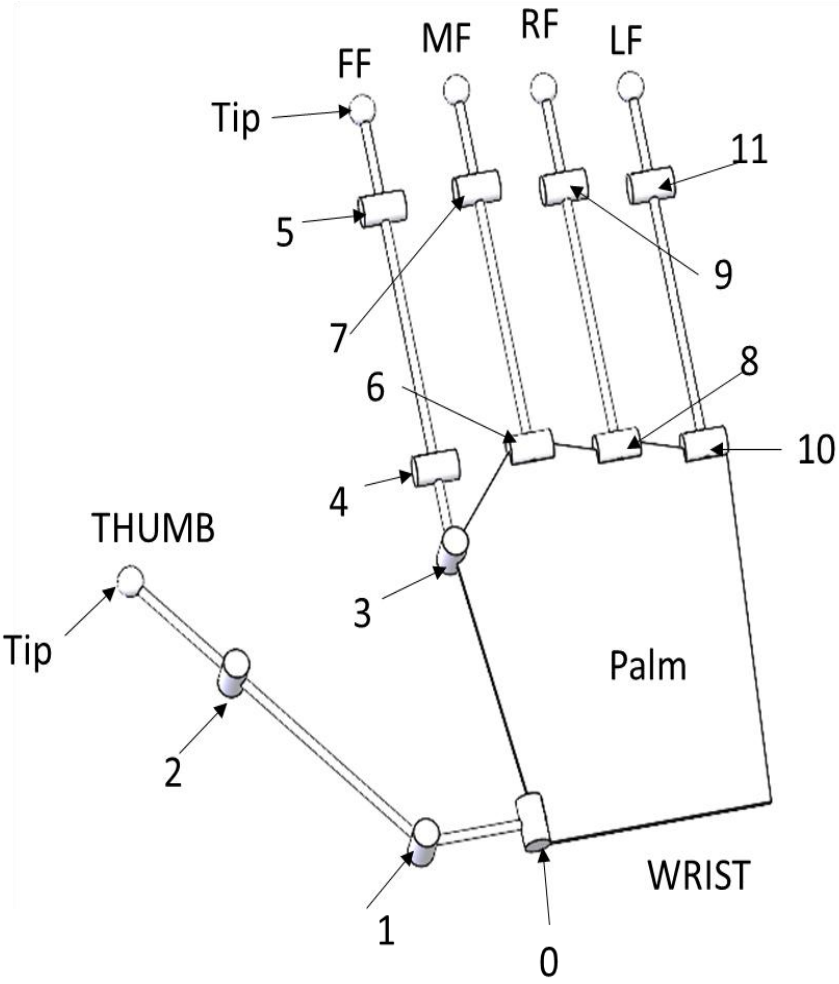
sports event	**parameters**	
Product Model	2L32 (left-handed) 2R32 (right hand)	
Active Freedom	12	
weights	1.1Kg	
sizes	190.36*94*47mm	
Rated Grip Strength	12N (single finger)	24N (full lot)
Rated load	1.2kg (single finger)	4.8kg (whole hand)
fig. sideways movement of the index finger	-5~17°	
Thumb movement angle	<110°	
Minimum Grip Diameter	16mm	
Repeat Positioning Accuracy	0.5mm	
communication interface	EtherCAT	RS485

power supply	DC 24~70V	Max 3A, 144W

3.2 Dimensions and dynamics



角度	图例说明	范围
小拇指 无名指 中指 食指		∠a:0°~110° ∠b:0°~110°
大拇指弯曲角度		∠c: -60°~90° ∠d: 0°~90°
大拇指旋转角度		∠e: 0°~90°
食指摆动角度		∠f: -5°~17°

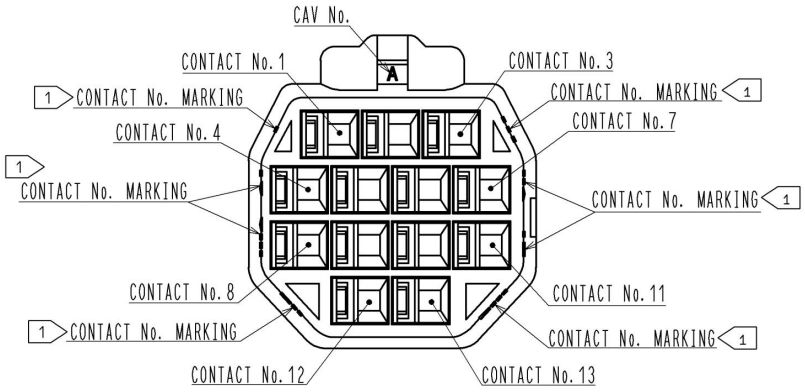


Joints	Angle/Degrees		Radians	
	Min/Min	Large value/Max	Min/Min	Large value/Max
0	0	90	0	1.57
1	-60	90	-1.05	1.57
2	0	90	0	1.57
3	-5	17	-0.087	0.297

4	0	110	0	1.92
5	0	110	0	1.92
6	0	110	0	1.92
7	0	110	0	1.92
8	0	110	0	1.92
9	0	110	0	1.92
10	0	110	0	1.92
11	0	110	0	1.92

3.3 Electrical interfaces and communications

The dexterous hand is left with an interface that uses the DF62B-13EP-2.2C connector; the pin definitions are as follows

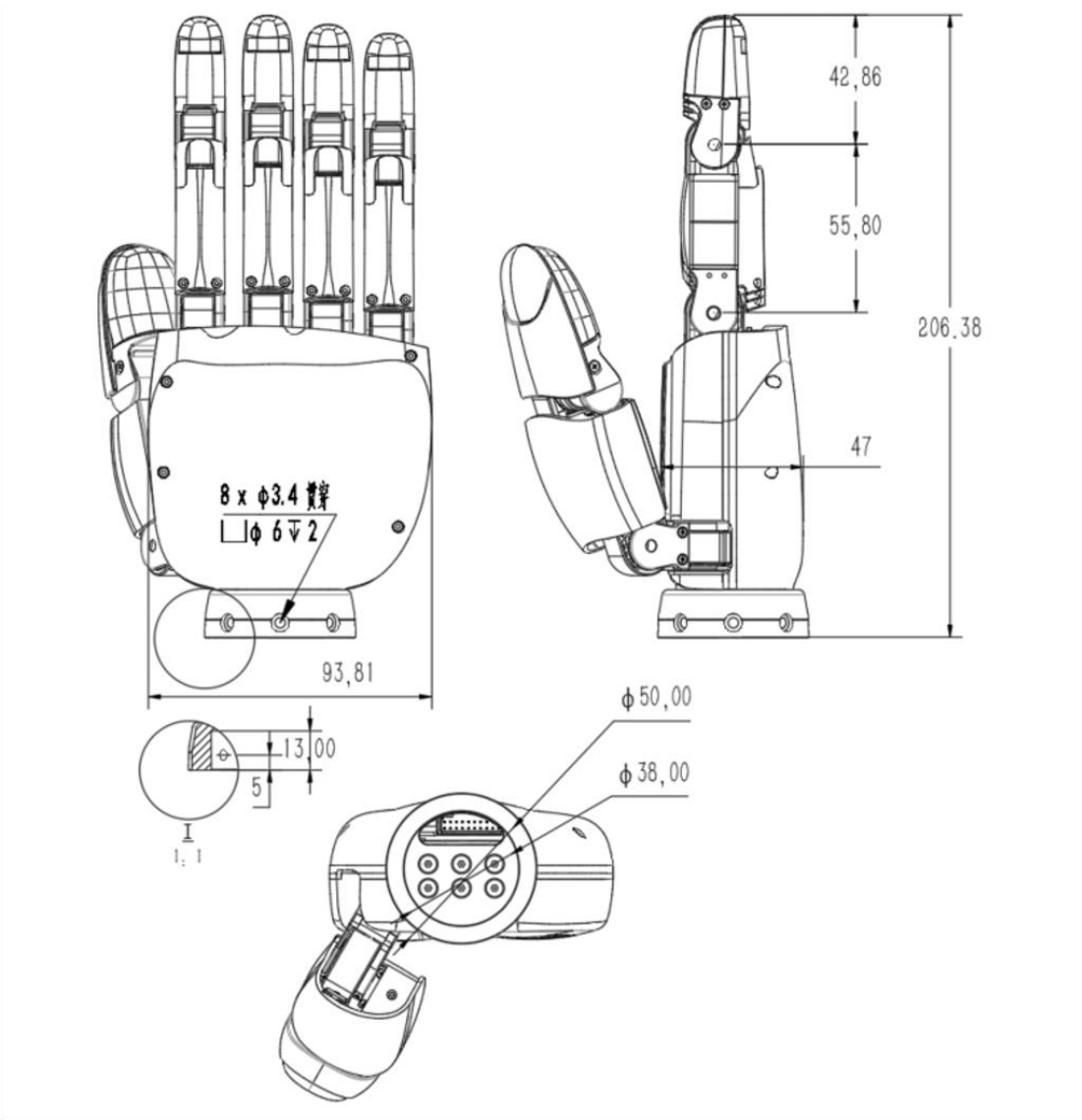


Interface Type	Pin Definitions
RS485	8 (485_SD_P)
	9 (485_SD_N)
Power supply interface	6 (POWER)

	7 (GND)
	10 (GND)
Ether CAT	4 (RA_N)
	3 (TA_P)
	5 (RA_P)
	2 (TA_N)
	1 (SHELL1)

3.4 Mechanical connections

The dexterous hand interface is attached via 8 M3 screws secured circumferentially, and the wiring harness can be routed out the side once the dexterous hand is mounted to the mount.



4. Communication protocols

4.1 Summary of communication protocols

The host computer communicates with the dexterous hand using RS485 and EtherCat. The methods of communication include real-time communication and non-real-time communication.

Real-time communication refers to the data communication of the upper computer controlling the hand movements, and the whole hand communication rate is 83 Hz. Among the control commands in RS485 communication, the downstream data includes the control data of all the knuckles, and the upstream data includes the state information of all the knuckles and the data of all the fingertip sensors, and the time of one interaction is completed within 12 ms. The control instructions in EtherCat communication use the PDO communication method, each PDO data communication in the downstream data includes the control data of 1 joint, the upstream data

includes the status data of 1 joint (the same joint as the downstream data) and the data of 1 fingertip sensor, the interaction time is 1ms, and the whole hand has 12 joints, which takes a total of 12ms.

Non-real-time communication refers to data communication in which the host computer reads and writes hand parameters. The question-and-answer method of communication is used, where the upper computer sends commands and the hand parses them and replies with an answer frame. Interactive transmission is possible via RS485 channel and EtherCat SDO channel.

4.1.1 RS485 Protocol Configuration

starting position	character length	parity check	stop bit	flow control	byte order	baud
1	8bit	not have	1	not have	LSB Priority	3Mbps

4.1.2 RS485 and ethercat SDO Frame format is as follows:

name (of a thing)	header 2byte		source address 1byte	target address 1byte	command word 1byte	lengths 2byte		digital Nbyte	CRC check 2byte	
(be) worth	0x55	0xAA	0~255	0~255	0~255	L	H		L	H

Each node in the system has an ID which is not repeated and remains constant. 0xFF is the broadcast ID

Source address: sender of the data frame

Destination address: the receiver of the data frame.

Length: is the length of the data segment

CRC checksum: All bytes from the frame header to the data segment are checksummed, using CRC16 for checksum, as in the following code:

```
C++
 static const uint16_t crc16tab[256]=
{ 0x0000,0x1021,0x2042,0x3063,0x4084,0x50a5,0x60c6,0x70e7,
0x8108,0x9129,0xa14a,0xb16b,0xc18c, 0xd1ad,0xe1ce,0xf1ef,
0x1231,0x0210,0x3273,0x2252,0x52b5,0x42b5,0x42b5,0x42c
0xd1ad,0xe1ce,0xf1ef,
```

```
0x1231,0x0210,0x3273,0x2252,0x52b5,0x4294,0x72f7,0x62d6,
0x9339,0x8318,0xb37b,0xa35a,0xd3bd,0xc39c,0xf3ff, 0xe3de,
0x2462,0x3443,0x0420,0x1401,0x64e6,0x74c7,0x44a4,0x5485,
0xa56a,0xb54b,0x8528,0x9509,0xe5ee,0xf5cf,0xc5ac,0xd58d, 0x3653,
0x2672,0x1611,0x0630,0x76d7,0x66f6,0x5695,0x46b4,
0xb75b,0xa77a,0x9719,0x8738,0xf7df,0xe7fe,0xd79d,0xc7bc,
0x48c4,0x58e5,0x6886, 0x78a7,0x0840,0x1861,0x2802,0x3823,
0xc9cc,0xd9ed,0xe98e,0xf9af,0x8948,0x9969,0xa90a,0xb92b,
0x5af5,0x4ad4,0x7ab7,0x6a96,0x1a71, 0x0a50,0x3a33,0x2a12,
0xdbfd,0xcbdc,0xfbbf,0xeb9e,0x9b79,0x8b58,0xbb3b,0xab1a,
0x6ca6,0x7c87,0x4ce4,0x5cc5,0x2c22,0x3c03,0x0c60, 0x1c41,
0xedae,0xfd8f,0xcdec,0xddcd,0xad2a,0xbd0b,0x8d68,0x9d49,
0x7e97,0x6eb6,0x5ed5,0x4ef4,0x3e13,0x2e32,0x1e51,0x0e70,
    0xff9f,0xefbe,0xdfdd,0xcffc,0xbf1b,0xaf3a,0x9f59,0x8f78,
0x9188,0x81a9,0xb1ca,0xa1eb,0xd10c,0xc12d,0xf14e,0xe16f,
0x1080,0x00a1, 0x30c2,0x20e3,0x5004,0x4025,0x7046,0x6067,
0x83b9,0x9398,0xa3fb,0xb3da,0xc33d,0xd31c,0xe37f,0xf35e,
0x02b1,0x1290,0x22f3,0x32d2, 0x4235,0x5214,0x6277,0x7256,
0xb5ea,0xa5cb,0x95a8,0x8589,0xf56e,0xe54f,0xd52c,0xc50d,
0x34e2,0x24c3,0x14a0,0x0481,0x7466,0x6447, 0x5424,0x4405,
0xa7db,0xb7fa,0x8799,0x97b8,0xe75f,0xf77e,0xc71d,0xd73c,
0x26d3,0x36f2,0x0691,0x16b0,0x6657,0x7676,0x4615,0x5634,
0xd94c,0xc96d,0xf90e,0xe92f,0x99c8,0x89e9,0xb98a,0xa9ab,
0x5844,0x4865,0x7806,0x6827,0x18c0,0x08e1,0x3882,0x28a3,
0xcb7d,0xdb5c, 0xeb3f,0xfb1e,0x8bf9,0x9bd8,0xabbb,0xbb9a,
0x4a75,0x5a54,0x6a37,0x7a16,0x0af1,0x1ad0,0x2ab3,0x3a92,
0xfd2e,0xed0f,0xdd6c,0xcd4d, 0xbdaa,0xad8b,0x9de8,0x8dc9,
0x7c26,0x6c07,0x5c64,0x4c45,0x3ca2,0x2c83,0x1ce0,0x0cc1,
0xef1f,0xff3e,0xcf5d,0xdf7c,0xaf9b,0xbfba, 0x8fd9,0x9ff8,
0x6e17,0x7e36,0x4e55,0x5e74,0x2e93,0x3eb2,0x0ed1,0x1ef0
}; uint16_t crc16_t

 uint16_t crc16(const char *buf, int len) { int counter; uint16_t
crc = 0; for (counter = 0; counter < len; counter++)
crc = (crc<<8) ^ crc16tab[(( crc>>8) ^ *buf++)&0x00FF]; return
crc;
}
```

4.1.3 ID development

The whole lot ID defaults to 0. Accessing multiple lots in the same system requires that the whole lot ID be set to a different value.

The hand consists of 1 communication board unit (responsible for data transmission), 12 knuckle units (responsible for grasping action) and 5 fingertip pressure sensors.

Each unit has a different ID as a unique identifier for communication.

Upper ID: fixed value 0xFE.

Communication Board ID: The highest bit value 1 of the Whole Hand ID is used as the communication board ID (i.e.: Whole Hand ID | 0x80). The whole hand ID is stored in the communication board.

Knuckle and pressure sensor ID:

name (of a thing)	ID	name (of a thing)	ID	name (of a thing)	ID	name (of a thing)	ID	name (of a thing)	ID
The thumb is off-set on the pendulum.	0	index finger off-set (math.)	3	Middle Finger 1 Joint	6	Ring finger 1 joint	8	Little finger joint 1	10
Thumb 1 joint	1	Index finger 1 joint	4	Middle finger 2 joints	7	Ring finger 2 joints	9	Little finger 2 joints	11
Thumb 2 joint	2	2 joints of the index finger	5						
Thumbs Up Sensor	0x11	Index Finger Sensor	0x12	Middle Finger Sensor	0x13	Ring Finger Sensor	0x14	Pinky Sensor	0x15

4.2 Real-time communication (control hand movements)

4.2.1 EtherCAT PDO Protocol

PDO is used to transmit periodic data (period 1ms), real-time data update, divided into receive PDO (RxPDO) and transmit PDO (TxPDO)

The following is the PDO data table, which needs to be 2-byte aligned

orientati	boyfriend	digital	typology	name (of a thing)	

ons					
RxPDO	index: 0x6001 Subindex: 0	Cur_id	uint8_t	Knuckle ID	
		sensor_id	uint8_t	Fingertip Pressure Sensor ID	
		Current_pos	float	Current position of the knuckle, in radians	
		Cur_torq	uint16_t	Knuckle real-time torque (current)	
		raw_position	uint16_t	Higher 12 bits store original position (0~4096), lower 4 bits store control_mode Higher bits are in front, lower bits are in back.	
		temperature	uint16_t	High 8bit Storage Palm Board Temperature Low 8bit Storage Knuckle Board Temperature	
		commboard_err	uint16_t	Communication Board Error Code	
		jonitboard_err	uint16_t	Joint Plate Error Code	
		tipboard_err	uint16_t	Fingerboard Error Code	
		default5	uint16_t	retained value	
		default6	uint16_t	retained value	
		default7	uint16_t	retained value	
		FX	int8_t	Value of the combined force in the x-direction of the fingertip sensor	

		FY	int8_t	Value of the combined force in the y-direction of the fingertip sensor	
		FZ	uint8_t	Combined force value in the z-direction of the fingertip sensor	
		forces [3*120]	uint8_t[3*120]	Force values in x, y, and z directions for 120 array points of fingertip sensors	
		temp[20]	uint8_t[20]	Temperature values for 20 temperature points of the fingertip sensor	
		temptur e	uint8_t	Fingertip Sensor Temperature Combined Value	
TxPDO	index: 0x7001 Subindex: 0	id	uint16_t	Knuckle ID	
		kp	int16_t	Proportional Gain Parameters	
		ki	int16_t	Integral coefficient	
		kd	int16_t	Differential Gain Parameters	
		position	float	target location	
		tor_max	uint16_t	Upper limit of knuckle torque	
		mode	uint16_t	Knuckle movement mode: (default 3) 1: Parallel mode 2: String level model 3: Position mode	
		res0	uint16_t	retained value	
		res1	uint16_t	retained value	

		res2	uint16_t	retained value	
		res3	uint16_t	retained value	

4.2.2 RS485 real-time control protocols

Real-time transmission, cycle time 12ms, the host computer sends the desired parameters of the knuckle, the hand returns the current real-time position and torque, command word: 0x02

The host computer sends the command frame to the hand:

frame byte		lengths	numerical value	clarification	typology	
header		2B	0x55AA	header	uint16_t	
Source ID		1B	Host ID	Source ID	uint8_t	
Destination ID		1B	Hand ID | 0x80	Destination ID	uint8_t	
command word		1B	0x02	command word	uint8_t	
Data segment length		2B		Data segment length	uint16_t	
data segment (24*12=288B)	Knuckle 0 expectation parameter (24B)	2B		Knuckle id	uint16_t	
		2B		kp proportional gain parameter	int16_t	
		2B		ki Integration factor	int16_t	
		2B		kd differential gain parameter	int16_t	

		4B		position Desired position (in radians)	float	
		2B		Upper torque limit	uint16_t	
		2B		Knuckle movement mode: (default 3) 1: Parallel mode 2: String level model 3: Position mode	uint16_t	
		2B		res0 Reserved	uint16_t	
		2B		res1 Reserved	uint16_t	
		2B		res2 Reservations	uint16_t	
		2B		res3 Reservations	uint16_t	
	Knuckle 1 Expectation Parameters (24B)	Same Knuckle 0				
	...	...				
	Knuckle 11 Desired Parameters (24B)	Same Knuckle 0				
CRC		2B		CRC of all bytes from the frame header to the data segment	uint16_t	

The hand replies to the response frame of the OP:

frame byte		lengths	numerical value	clarification	typology	
header		2B	0x55AA	header	uint16_t	
Source ID		1B	Hand ID | 0x80	Source ID	uint8_t	
Destination ID		1B	Host ID	Destination ID	uint8_t	
command word		1B	0x02	command word	uint8_t	
Data segment length		2B		Data segment length	uint16_t	
data segment (24*12+384*5=2208B)	Knuckle 0 current state (24B)	2B		Knuckle id	uint16_t	
		4B		Current position (in radians)	float	
		2B		Current torque	uint16_t	
		2B		default0	uint16_t	
		2B		default1	uint16_t	
		2B		default2	uint16_t	
		2B		default3	uint16_t	
		2B		default4	uint16_t	

		2B		default5	uint16_t	
		2B		default6	uint16_t	
		2B		default7	uint16_t	
	Current state of knuckle 1 (24B)	Same Knuckle 0				
	...	...				
	Current status of knuckle 11 (24B)	Same Knuckle 0				
	Fingertip Sensor 0x11 Data (384B)	1B		FX Value of the combined force in the x-direction	int8_t	
		1B		Value of the combined force in the FY y-direction	int8_t	
		1B		Value of the combined force in the direction of FZ z	uint8_t	
		3*120B		Force values in x, y, and z directions for 120 array points	uint8_t	
		20B		Temperature values for 20 temperature points	uint8_t	
		1B		temperature coefficient	uint8_t	

	...	...				
	Fingertip sensor 0x15 data (384B)	Same as fingertip sensor 0x11 data				
CRC		2B		CRC of all bytes from the frame header to the data segment	uint16_t	

4.3 Non-real-time transmission protocols

4.3.1 EtherCat SDO for transferring non-periodic data

Object subdictionary index : 0x8001

subindex: 0x01 sdo read index, length: 256Byte

subindex: 0x02 sdo write index, length: 4096+128Byte

The transmitted data is compatible with the RS485 protocol, i.e., the frame format in (1. Communication Protocol Summary).

4.3.2 Error Code Upload Command Word: 0x00

The reply frame that the hand sends to the host computer:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B		Source ID	uint8_t	
Destination ID	1B	Host ID	Destination ID	uint8_t	
command word	1B	0x00	command word	uint8_t	
Data segment length	2B	2	Data segment length	uint16_t	

data segment	2B		error code	uint16_t	
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

Error Code Definition:

Error Code Name	define	error code interpretation	conditions for the emergence of
ERROR_ID	301	Communication ID error code, mismatch with local ID	Send a command that is not the ID of a knuckle board, palm board, or sensor board
ERROR_CMD	302	Communication CMD error, non-existent CMD	
ERROR_COMMUNICATION	303	Communication error, protocol anomaly or checksum failure	
ERROR_DATA_LEN	304	Data segment length error	
ERROR_NOFLASHPARAM	305	Flash parameter read exception, no preset parameter exists	The inside of the joint plate can produce
ERROR_COMMUNICATION_BUSY	306	Previous frame data not processed, device busy	
ERROR_BOOT_CMD	307	Running in boot CMD error, non-existent CMD	
ERROR_DEVICE_DISCONNECT	308	Joint or sensor plate disconnection	

Joint plate error code:			
ERROR_SM_REG	100	State Machine Registration Failed	
ERROR_PARAM_INIT	101	Flash parameter initialization failure	
ERROR_SM_TRANS	102	State machine state transfer failure	
ERROR_TEMP_PROCTCED	103	Joint plate over-temperature, exceeding safe temperature limits	
ERROR_MOTION_MODE	104	Received incorrect mode of operation	
ERROR_PARAM_OUTOF_RANGE	105	Control parameters exceeding the permissible range	
ERROR_NOFLASHPARAM	106	Flash parameter read exception, no preset parameter exists	
ERROR_COMMUNICATION	107	Communication disconnection (actually detected and reported by the communication board)	
ERROR_CMD	108	Communication CMD error, received non-existent CMD	
ERROR_POSITION_RAW	109	The raw voltage of the position signal read is not in the normal range	
error_current_procted	110	Current exceeds set threshold, triggers overcurrent warning	

		(more than 500ms)	
Sensor board (fingertip) error code:			
ERROR_READ_TOTAL_FORCE	200	Sensor failed to read the combined force	
ERROR_READ_FORCES	201	Failure of sensor to read distributed force	
ERROR_READ_TEMP	202	Sensor failed to read temperature	
ERROR_CALIBRIATE	203	Sensor calibration failure	
ERROR_CMD	205	Communication CMD error, non-existent CMD	
ERROR_NOFLASHPARAM	206	Flash parameter read exception, no preset parameter exists	
ERROR_COMMUNICATION	207	Communication disconnection (actually detected and reported by the communication board)	

4.3.3 Write user parameters to flash Command word: 0x05

The host computer sends the command frame to the hand:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Host ID	Source ID	uint8_t	

Destination ID	1B	Hand ID | 0x80	Destination ID	uint8_t	
command word	1B	0x05	command word	uint8_t	
Data segment length	2B	0	Data segment length	uint16_t	
data segment	0B	not have			
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

The reply frame that the hand sends to the host computer:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Hand ID | 0x80	Source ID	uint8_t	
Destination ID	1B	Host ID	Destination ID	uint8_t	
command word	1B	0x05	command word	uint8_t	
Data segment length	2B	2	Data segment length	uint16_t	
data segment	2B		0x0010 Success 0x0000 Failed	uint16_t	
CRC	2B		CRC of all bytes from the frame header to the data	uint16_t	

			segment		

4.3.4 Fingertip Sensor Zero Command Word: 0x12

The host computer sends the command frame to the hand:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Host ID	Source ID	uint8_t	
Destination ID	1B	Fingertip Sensor ID	Destination ID	uint8_t	
command word	1B	0x12	command word	uint8_t	
Data segment length	2B		Data segment length	uint16_t	
data segment	0B	not have			
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

The reply frame that the hand sends to the host computer:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Fingertip Sensor ID	Source ID	uint8_t	

Destination ID	1B	Host ID	Destination ID	uint8_t	
command word	1B	0x12	command word	uint8_t	
Data segment length	2B	0	Data segment length	uint16_t	
data segment	0B	not have			
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

4.3.5 Getting the Version Number Command Word: 0x13

The host computer sends the command frame to the hand:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Host ID	Source ID	uint8_t	
Destination ID	1B		Destination ID	uint8_t	
command word	1B	0x13	command word	uint8_t	
Data segment length	2B		Data segment length	uint16_t	
data segment	0B	not have			

CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

The reply frame that the hand sends to the host computer:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B		Source ID	uint8_t	
Destination ID	1B	Host ID	Destination ID	uint8_t	
command word	1B	0x13	command word	uint8_t	
Data segment length	2B	8	Data segment length	uint16_t	
data segment	8B		The first 4 bytes indicate the software version number The last 4 bytes indicate the hardware version number Specific representation of software and hardware version numbers: where 24~31bit indicates major_ver 16~23bit indicates minor_ver 0~15bit Indicates release_ver Version number: major_ver.minor_ver.relea		

			se_ver		
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

4.3.6 Firmware Reset Reboot Command Word: 0x14

The host computer sends the command frame to the hand:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Host ID	Source ID	uint8_t	
Destination ID	1B		Destination ID	uint8_t	
command word	1B	0x14	command word	uint8_t	
Data segment length	2B		Data segment length	uint16_t	
data segment	0B	not have			
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

Lower unit firmware reset, no reply

4.3.7 Read Register Command Word: 0x15

The host computer sends the command frame to the hand:

frame byte	lengths	numerical value	clarification	typology	

header	2B	0x55AA	header	uint16_t	
Source ID	1B	Host ID	Source ID	uint8_t	
Destination ID	1B	Hand ID | 0x80	Destination ID	uint8_t	
command word	1B	0x15	command word	uint8_t	
Data segment length	2B		Data segment length	uint16_t	
data segment	4B		Index Starting register index	uint16_t	
			lengh Register length	uint16_t	
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

The reply frame that the hand sends to the host computer:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Hand ID | 0x80	Source ID	uint8_t	
Destination ID	1B	Host ID	Destination ID	uint8_t	
command word	1B	0x15	command word	uint8_t	
Data segment	2B	8	Data segment length	uint16_t	

length					
data segment	nB		Index Starting register index	uint16_t	
			data Register data	uint8_t[]	
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

4.3.8 Write Register Command Word: 0x16

After writing the register, you need to use the "user parameter write flash command word: 0x05" instruction to store it in flash, otherwise it will go back to the data before writing after reboot.

The host computer sends the command frame to the hand:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Host ID	Source ID	uint8_t	
Destination ID	1B	Hand ID | 0x80	Destination ID	uint8_t	
command word	1B	0x16	command word	uint8_t	
Data segment length	2B		Data segment length	uint16_t	
data segment	nB		Index Starting register index	uint16_t	
			data Register data	uint8_t[]	
CRC	2B		CRC of all bytes from the	uint16_t	

			frame header to the data segment		

The reply frame that the hand sends to the host computer:

frame byte	lengths	numerical value	clarification	typology	
header	2B	0x55AA	header	uint16_t	
Source ID	1B	Hand ID | 0x80	Source ID	uint8_t	
Destination ID	1B	Host ID	Destination ID	uint8_t	
command word	1B	0x16	command word	uint8_t	
Data segment length	2B	8	Data segment length	uint16_t	
data segment	2B		0x0010 Success 0x0000 Failed	uint16_t	
CRC	2B		CRC of all bytes from the frame header to the data segment	uint16_t	

The registers are listed below:

register (byte)	element	causality	note
0~19	reserved		
20	Left hand or right hand	read-only (computing)	Left hand: 'L' or 'l' Right hand: 'R' or 'r'

21~52	SN	read-only (computing)	
53	Hand ID	fill out or in (information on a form)	
54~85	hand name	fill out or in (information on a form)	
86~205	calibration parameters	fill out or in (information on a form)	typedef struct { int16_t position_stretching[12]; int16_t position_closing[12]; int16_t position_zero[12]; int16_t position_init[12]; int8_t angle_stretching[12]; int8_t angle_closing[12]; }__attribute__((packed)) HandParam_t;
206~255	reserved		

5. Legal statements

1. The copyright of this product manual and its introduction of our (Beijing RobotEra Technology Co., Ltd.) products, such as text, graphics, photographs and other works and supporting software works belongs to our company. Without the permission of our company, any unit or individual shall not exercise the copyright, including but not limited to, shall not exercise the following behaviors:

(1) Publishing, registering, reproducing, transmitting, or distributing this product manual or the text, graphics, photographs, or software contained therein.

(2) Intentionally deleting or altering the content or layout of this brochure.

(3) Knowing or should have known that the contents and layout of this product manual have been deleted or changed without authorization, and still providing it to the public.

2. The product implementation details, and business information not disclosed in this product manual belong to our company (Beijing RobotEra Technology Co., Ltd.) commercial secrets, without our company's permission, any unit or individual shall not be implemented to infringe on our company's commercial secrets.

(1) Theft, bribery, fraud, coercion, electronic intrusion or other improper means to obtain the above trade secrets of our company.

(2) Disclosing, using or permitting others to use the above trade secrets of our company obtained by the means described in the preceding paragraph.

(3) Disclosing, using or permitting others to use the above trade secrets of the Company in its possession in violation of the obligation of confidentiality or the requirement of the Company to keep trade secrets.

(4) Abetting, inducing, or assisting others to violate the duty of confidentiality or the requirement of our company to keep trade secrets by obtaining, disclosing, using, or allowing others to use the above trade secrets of our company.

(5) Obtaining, disclosing, using or permitting others to use the above trade secrets of our company, knowing or should have known that our employees, former employees, or other units or individuals have committed the offenses listed in (1)-(4).