EXOSKELETON MANIPULATOR

EH55BF1-BR

USER MANUAL

Beijing Inspire-Robots Technology Co., Ltd. | March 2023

ID: PRJ-04-TS-U-001 | Page Number: 18 pages | Version: V2.0.2

Table of Contents

Exoskeleton Manipulator EH55BF1-BR

User Manual

1 Product Overview

1.1 Product Features

EH55BF1-BR is an exoskeleton manipulator (hereinafter referred to as "Exoskeleton Hand") designed to integrate the Micro Linear Servo Actuators with small volume and large torque. This exoskeleton hand integrates five Micro Linear Servo Actuators. The user interface adopts Bluetooth 4.0 and RS232. There is a sensitive built-in pressure sensor. With a concise and simple structure, it is wearable like a glove and can sense and capture hand movements and realize AR/VR interaction. Each finger can provide the active force of 3 N and the passive force of 5 N. The exoskeleton hand is suitable for applications such as exoskeleton for rehabilitation, teleoperation with force feedback, etc. The exoskeleton hand has the following features:

- ➢ Supply voltage: Wide voltage range (DC 6 V to 8.4 V) for power supply; 8 V and 4 A (peak current) are recommended.
- ➢ Communication interface:
- ◆ Wireless: Bluetooth 4.0
- ◆ Wired: RS232 serial port (115200bps, 8 data bits, 1 stop bit, no parity)
- ➢ Weight: about 500 g

1.2 Appearance and structure

The appearance and structure of the exoskeleton hand are as shown in Figure 1.

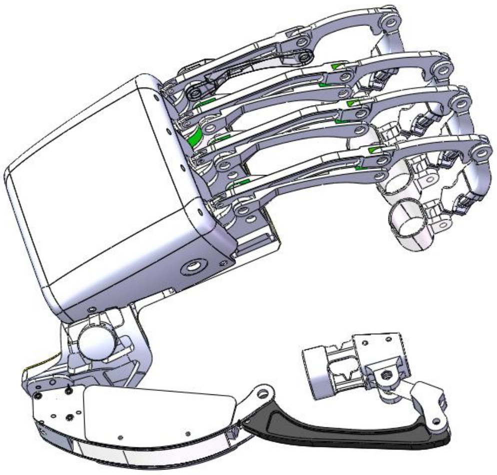
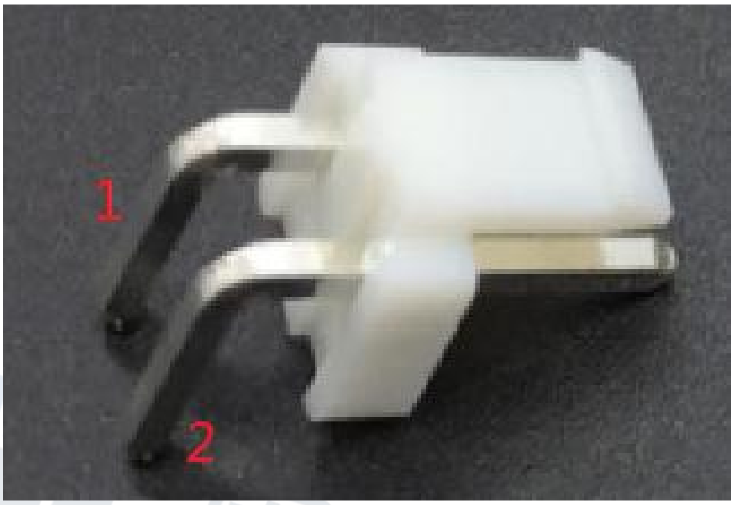


Figure 1: Appearance and Structure

1.3 Electric connection

1.3.1 Pin definition

The electrical interfaces of the exoskeleton hand include a power interface and a communication interface. The power interface is a VH3.96 connector with 90°right angle plug; it has a latch, so it is not easy to detach. The wired RS232 communication interface is a XS6-3 connector with micro aviation plug.



VH3.96	1	GND
	2	VCC

Figure 2: chematic diagram of power supply interface

The wired RS232 communication interface is Micro aviation plug XS6-5 connector

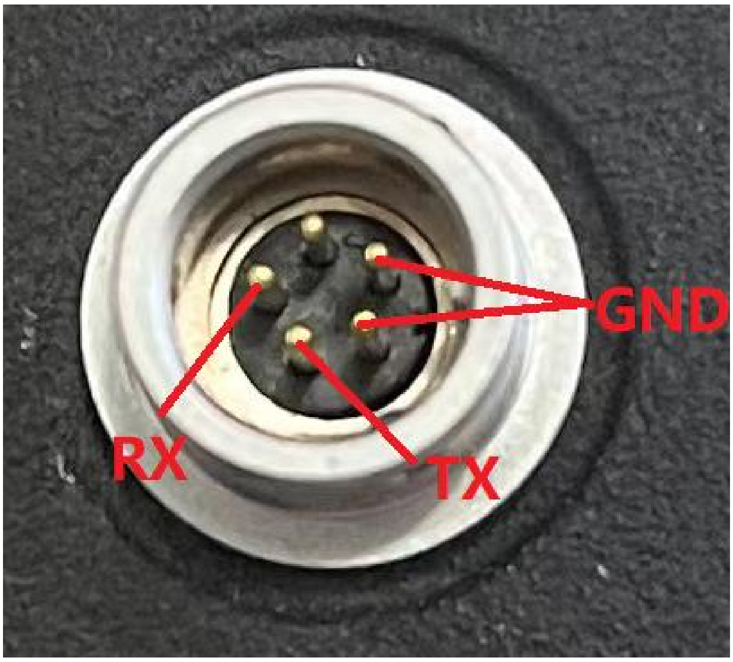


Figure 3:The pin definition of the communication interface

1.3.2 Communication mode

The exoskeleton manipulator has a standard RS232 serial port. After the user connects the serial port cable of the exoskeleton hand to the USB-serial port of the PC, the user can operate the manipulator through the PC master control software of the exoskeleton hand. This software can send the communication protocol-compliant packet to the exoskeleton hand via the serial port. After receiving a correct instruction, the exoskeleton hand can analyze it, complete the corresponding instruction requirement, and then return the response packet.

2 Communication Protocol

2.1 Summary of Communication Protocol

Q&A communication is used between the main control unit (MCU) and the exoskeleton hand. The MCU actively sends the instruction frame; after receiving the instruction frame, the exoskeleton hand will return the response frame after resolution.

The user must assign a different ID to the exoskeleton hand as unique identifier (the factory default ID of the exoskeleton hand is 1). The data volume of the instruction frame sent by the MCU includes the ID information. The exoskeleton hand can completely receive the instruction frame only if the ID matches, and will return the corresponding response frame after processing the instruction.

The communication mode is the universal asynchronous receiver/transmitter (UART) serial port. The unit of the instruction frame is byte. A single byte consists of 10 bits, including a start bit, 8 data bits and 1 stop bit, no parity.

2.2 Instruction Format

2.2.1 Force control command

This command can be used to set the magnitude of force that the exoskeleton hand applies to the user's fingers. The frame format is as follows:

S/N.	Value	Type	Description
Byte0	0xEB	Byte	Packet header
Byte1	0x90	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	0x0B	Byte	Data length of this frame
Byte4	0x31	Byte	Force control command flag
Byte5, 6	Finger0_Force_Set	Short	Force control setting for the little finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte7, 8	Finger1_Force_Set	Short	Force control setting for the ring finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte9, 10	Finger2_Force_Set	Short	Force control setting for the middle finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①

Byte11, 12	Finger3_Force_Set	Short	Force control setting for the index finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte13, 14	Finger4_Force_Set	Short	Force control setting for the thumb; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte15	Checksum	Byte	Checksum②

Notes:

1. If a positive force control value is set, the exoskeleton hand will apply force to the user's hand in such a direction that the hand is held open. If a negative force control value is set, the exoskeleton hand will apply force to the user's hand in such a direction that the fist is clenched.

2. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

After receiving this frame, the exoskeleton hand will return the response frame for the position with 6 degrees of freedom (DOF).

S/N.	Value	Type	Description
Byte0	0x90	Byte	Packet header
Byte1	0xEB	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	0x0D	Byte	Data length of this frame
Byte4	0x31	Byte	Force control command flag
Byte5, 6	Finger0_Pos_Set	Short	Little finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte7, 8	Finger1_Pos_Set	Short	Ring finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte9, 10	Finger2_Pos_Set	Short	Middle finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte11, 12	Finger3_Pos_Set	Short	Index finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte13, 14	Finger4_Pos_Set	Short	Thumb position; low-order bytes are followed by high-order bytes; range [0,

			1000]①
Byte15, 16	Finger5_Pos_Set	Short	Lateral rotation position of the thumb; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte17	Checksum	Byte	Checksum②

Notes:

1. When the position of the five fingers is 0, it means that the fingers have bent to the maximum extent; when it is 1000, it indicates that the fingers are fully held open. When the lateral rotation position of the thumb is 0, it means that the thumb has rotated to the innermost position of the palm; when it is 1000, it indicates that the thumb has rotated to the outermost position of the palm.

2. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

2.2.2 Position control command

This command can be used to set the finger positions of the exoskeleton hand. The frame format is as follows:

	Value	Type	Description
Byte0	0xEB	Byte	Packet header
Byte1	0x90	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	0x0B	Byte	Data length of this frame
Byte4	0x32	Byte	Position control command flag
Byte5, 6	Finger0_Pos_Set	Short	Set value of the little finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte7, 8	Finger1_Pos_Set	Short	Set value of the ring finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte9, 10	Finger2_Pos_Set	Short	Set value of the middle finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte11, 12	Finger3_Pos_Set	Short	Set value of the index finger position; low-order bytes are followed by high-order bytes; range [0, 1000]①

Byte13, 14	Finger4_Pos_Set	Short	Set value of the thumb position; low-order bytes are followed by high-order bytes; range [0, 1000]①
Byte15	Checksum	Byte	Checksum②

Notes:

1. When the position of the five fingers is 0, it means that the fingers have bent to the maximum extent; when it is 1000, it indicates that the fingers are fully held open. When the lateral rotation position of the thumb is 0, it means that the thumb has rotated to the innermost position of the palm; when it is 1000, it indicates that the thumb has rotated to the outermost position of the palm.

2. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

After receiving this frame, the exoskeleton hand will return the response frame for the actual force applied to the five fingers.

S/N.	Value	Type	Description
Byte0	0x90	Byte	Packet header
Byte1	0xEB	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	0x0B	Byte	Data length of this frame
Byte4	0x32	Byte	Position control command flag
Byte5, 6	Finger0_Force_Set	Short	Actual force applied to the little finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte7, 8	Finger1_Force_Set	Short	Actual force applied to the ring finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte9, 10	Finger2_Force_Set	Short	Actual force applied to the middle finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte11, 12	Finger3_Force_Set	Short	Actual force applied to the index finger; low-order bytes are followed by high-order bytes; range [-1000, 1000]①
Byte13, 14	Finger4_Force_Set	Short	Actual force applied to the thumb; low-order bytes are followed by high-order bytes; range [-1000, 1000]①

Byte15	Checksum	Byte	Checksum②

Notes:

1. If the actual force is a positive value, the user's fingers will apply force to the exoskeleton hand in such a direction that the fist is clenched. When the force control value is set to a negative value, the user's fingers will apply force to the exoskeleton hand in such a direction that the hand is held open.

2. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

2.2.3 Read Register instruction

Format as follows:

S/N.	Value	Type	Description
Byte0	0xEB	Byte	Packet header
Byte1	0x90	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	0x04	Byte	Data length of this frame
Byte4	0x11	Byte	Read Register instruction flag
Byte5, 6	Address	Short	Register address; low-order bytes are followed by high-order bytes.①
Byte7	N	Byte	Bytes of Read Register Content②
Byte8	Checksum	Byte	Checksum③

a. See the following text for a detailed register description.

b. The content of multiple registers (with adjacent addresses) can be read.

c. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

After receiving the instruction frame of Read Register, the exoskeleton hand will return the response frame containing the register content.

S/N.	Value	Type	Description
Byte0	0x90	Byte	Packet header

Byte1	0xEB	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	Length	Byte	Data length of this frame
Byte4	0x11	Byte	Read Register instruction flag
Byte5, 6	Address	Short	Register address; low-order bytes are followed by high-order bytes.①
Byte7...(7+N-1)	Data	Byte	Register content; content length (N)= Length-3②
Byte(7+N)	Checksum	Byte	Checksum③

a. See the following text for a detailed register description.

b. The data content is the register (several registers with adjacent addresses) content with "Address" as start address and "N" as byte length. If the register contains 16 bits integer data, low-order bytes will be followed by high-order bytes.

c. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

2.2.4 Write Register instruction

Format as follows:

S/N.	Value	Type	Description
Byte0	0xEB	Byte	Packet header
Byte1	0x90	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	Length	Byte	Data length of this frame
Byte4	0x12	Byte	Write Register instruction flag
Byte5, 6	Address	Short	Register address; low-order bytes are followed by high-order bytes.①
Byte7...(7+N-1)	Data	Byte	Content to be written in the register; bytes (N)= Length-3②
Byte8	Checksum	Byte	Checksum③

a. See the following text for a detailed register description.

b. The data content is the data to be written in the register, which is the register (several registers with adjacent addresses) with "Address" as start address and "N" as byte length. If the register contains 16 bits integer data, low-order bytes will be followed by high-order bytes.

c. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

After receiving the instruction frame of Read Register, the exoskeleton hand will return the response frame containing the register content.

S/N.	Value	Type	Description
Byte0	0x90	Byte	Packet header
Byte1	0xEB	Byte	Packet header
Byte2	ID	Byte	Exoskeleton ID
Byte3	0x04	Byte	Data length of this frame
Byte4	0x12	Byte	Write Register instruction flag
Byte5, 6	Address	Short	Register address; low-order bytes are followed by high-order bytes.①
Byte7	0x01	Byte	
Byte8	Checksum	Byte	Checksum②

a. See the following text for a detailed register description.

b. The checksum consists of low-order bytes of the sum of all bytes except the packet header (byte0, byte1).

2.3 Register description

Specific register addresses and their data implication are described below.

Address	**Meaning**	**Abbreviation**	**Length**	**Permission (read / write)**
1000	ID	HAND_ID	1byte	W/R
1001	Baud rate setting	REDU_RATIO	1byte	W/R
1004	Error clearance	CLEAR_ERROR	1byte	W/R
1005	Saving data in Flash	SAVE	1byte	W/R
1009	Calibration	CLIBRATION	1byte	W/R
1472	System voltage	VLTAGE	1short	R
1486	Set value of the angle for each finger	ANGLE_SET(m)	5short(10byte)	W/R

1498	Set value of force control for each finger	FORCE_SET(m)	5short(10byte)	W/R
1546	Actual angle for each degree of freedom (DOF)	ANGLE_ACT(m)	6short(12byte)	R
1582	Actual force applied to each finger	FORCE_ACT(m)	5short(10byte)	R
1594	Current value of the servo actuator for each finger	CURRENT(m)	5short(10byte)	R
1606	Error codes of the servo actuator for each finger	ERROR(m)	5byte	R
1618	Temperature of the servo actuator for each finger	TEMP(m)	5byte	R

2.3.1 HAND_ID (Exoskeleton ID)

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

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

2.3.2 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

2.3.3 CLEAR_ERROR (error clearance)

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

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

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

2.3.4 SAVE (Saving parameters in the FLASH)

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

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

2.3.5 RESET_PARA (restoring factory defaults)

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

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

2.3.6 CLIBRATION (calibration)

When the user sets this parameter to 1, the exoskeleton hand will start the calibration process. The force sensor data and positions of five fingers will be recalibrated. The user must wear the exoskeleton hand before calibration. The following five steps should be taken after calibration begins:

Step 1: The user's fingers should be relaxed to ensure no force on the exoskeleton hand. Wait for 1 second;

Step 2: Under the action of the exoskeleton hand, the user holds the hand open; next, the user can make adjustment until five fingers become naturally straight; wait for 2 seconds;

Step 3: Under the action of the exoskeleton hand, the user clenches four fingers; next, the user can make adjustment until four fingers are fully clenched; wait for 2 seconds;

Step 4: Under the action of the exoskeleton hand, the user gets four fingers straight and bends the thumb; the user can make adjustment until the thumb laterally rotates to the inner side of the palm and bends to the maximum extend; wait for 2 seconds;

Step 5: Under the action of the exoskeleton hand, the user makes the five fingers touch each other (the thumb directly touches the index finger, with the little, ring and middle fingers bending to an angle identical to that of the index finger); wait for 2 seconds; calibration will be completed.

2.3.7 ANGLE_SET(m) (set value of the position for each finger)

This register group consists of five registers corresponding to the set values of the angle for five fingers of the exoskeleton hand, 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 five registers (ANGLE_SET(0)- ANGLE_SET(4)) are set to 500, 500, -1, 0, and 500 respectively, the little and ring fingers and the thumb will be bent by 3 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	Type of data	Range
1486-1487	ANGLE_SET(0)	Set value of the position for the little finger	short	-1, 0-1000
1488-1489	ANGLE_SET(1)	Set value of the position for the ring finger	short	-1, 0-1000
1490-1491	ANGLE_SET(2)	Set value of the position for the middle finger	short	-1, 0-1000
1492-1493	ANGLE_SET(3)	Set value of the position for the index finger	short	-1, 0-1000
1494-1495	ANGLE_SET(4)	Set value of the position for the thumb	short	-1, 0-1000

2.3.8 FORCE_SET (m) (set value of force control for each finger)

This register group consists of five registers corresponding to the set values of force control for five fingers of the exoskeleton hand, with details in the table below. The user can set this register group to control the force that the exoskeleton hand will apply to the fingers. If a positive force control value is set, the exoskeleton hand will apply force to the user's hand in such a direction that the hand is held open. If a negative force control value is set, the exoskeleton hand will apply force to the user's hand in such a direction that the fist is clenched.

Address	Name	Description	Type of data	Range
1498-1499	FORCE_SET(0)	Set value of force control for the little finger	short	-1000 to 1000
1500-1501	FORCE_SET(1)	Set value of force control for the ring finger	short	-1000 to 1000
1502-1503	FORCE_SET(2)	Set value of force control for the	short	-1000 to 1000

		middle finger		
1504-1505	FORCE_SET(3)	Set value of force control for the index finger	short	-1000 to 1000
1506-1507	FORCE_SET(4)	Set value of bending force control for the thumb	short	-1000 to 1000

2.3.9 ANGLE_ACT(m) (actual position value of each finger)

This register group consists of six registers corresponding to the actual position values of the exoskeleton hand with six DOF (bending of five fingers, lateral rotation of the thumb), with details in the table below.

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

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

This register group consists of five registers corresponding to the values of actual force applied to five fingers of the exoskeleton hand, with details in the table below.

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

2.3.11 CURRENT(m) (current value of the servo actuator for each DOF)

This register group consists of five registers corresponding to the current values of six servo actuators for the exoskeleton 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 servo actuator for the little finger	short	0-1000	mA
1596-1597	CURRENT(1)	Current value of the servo actuator for the ring finger	short	0-1000	mA
1598-1599	CURRENT(2)	Current value of the servo actuator for the middle finger	short	0-1000	mA
1600-1601	CURRENT(3)	Current value of the servo actuator for the index finger	short	0-1000	mA
1602-1603	CURRENT(4)	Current value of the servo actuator for bending of the thumb	short	0-1000	mA

2.3.12 ERROR(m) (error codes of each servo actuator)

This register group consists of five registers corresponding to the error codes of six servo actuators for the exoskeleton 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 servo actuator for the little finger	byte
1607	ERROR (1)	Error code of the servo actuator for the ring finger	byte
1608	ERROR (2)	Error code of the servo actuator for the middle finger	byte
1609	ERROR (3)	Error code of the servo actuator for the thumb	byte
1610	ERROR (4)	Error code of the servo actuator for bending of the thumb	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 servo 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

2.3.13 TEMP(m) (temperature of each servo actuator)

This register group consists of five registers corresponding to the temperature values of six servo actuators for the exoskeleton 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 servo actuator for the little finger	byte	0-100	℃
1619	TEMP(1)	Temperature value of the servo actuator for the ring finger	byte	0-100	℃
1620	TEMP(2)	Temperature value of the servo actuator for the middle finger	byte	0-100	℃
1621	TEMP(3)	Temperature value of the servo actuator for the index finger	byte	0-100	℃
1622	TEMP(4)	Temperature value of the servo actuator for bending of the thumb	byte	0-100	℃