

仿人五指灵巧手

RH56 系列用户手册



目 录

1 产品概述	1
1.1 产品特性	1
1.2 结构尺寸	1
1.3 性能参数	1
1.4 电气连接	2
1.4.1 引脚定义	2
1.4.2 通信方式	2
2 通信协议	4
2.1 通信协议概要	4
2.2 RS485 读写寄存器操作	4
2.2.1 读灵巧手寄存器的操作	4
2.2.2 写灵巧手寄存器的操作	6
2.3 Modbus 协议概要	9
2.3.1 读取保持寄存器	9
2.3.2 预置单寄存器	10
2.3.3 预置多寄存器	10
2.4 寄存器说明	11
2.4.1 HAND_ID 灵巧手 ID 号	13
2.4.2 REDU_RATIO 波特率设置	13
2.4.3 CLEAR_ERROR 清除错误	13
2.4.4 SAVE 保存参数到 FLASH	13
2.4.5 RESET_PARA 恢复出厂设置	13
2.4.6 GESTURE_FORCE_CLB 力传感器校准	13
2.4.7 CURRENT_LIMIT(m) 各自由度的电缸电流保护值	13
2.4.8 DEFAULT_SPEED_SET(m) 各自由度的上电速度设置值	14
2.4.9 DEFAULT_FORCE_SET(m) 各自由度的上电力控阈值设置值	14
2.4.10 POS_SET(m) 各自由度的电缸位置设置值	15
2.4.11 ANGLE_SET(m) 各自由度的角度设置值	16

2.4.12 FORCE_SET (m) 各自由度的力控阈值设置值	18
2.4.13 SPEED_SET(m) 各自由度的速度设置值	18
2.4.14 POS_ACT(m) 各自由度的电缸位置实际值	19
2.4.15 ANGLE_ACT(m) 各自由度的角度实际值	19
2.4.16 FORCE_ACT(m) 各手指的实际受力	19
2.4.17 CURRENT(m) 各自由度的电缸的电流值	20
2.4.18 ERROR(m) 各电缸的故障信息	20
2.4.19 STATUS(m)各自由度的状态信息	21
2.4.20 TEMP(m) 各电缸的温度	22
2.4.21 动作序列相关的寄存器	22
3 常规指令示例	25



仿人五指灵巧手

RH56 系列用户手册

1. 产品概述

1.1 产品特性

RH56 系列灵巧手是一款集成小体积大扭矩微型伺服电缸而设计生产的机械灵巧手，以下简称为“灵巧手”。该灵巧手集成了 6 个微型伺服电缸，用户接口采用 RS485 通信接口，内置灵敏的压力传感器，通过设置不同的阈值方便用户进行不同硬度物体的抓取，简洁高效的接口控制指令可使用户快速实现对灵巧手的操控，优质的性能使该灵巧手应用于服务机器人、教学教具、假肢等领域。

1.2 结构尺寸

灵巧手的结构尺寸如图 1 所示。

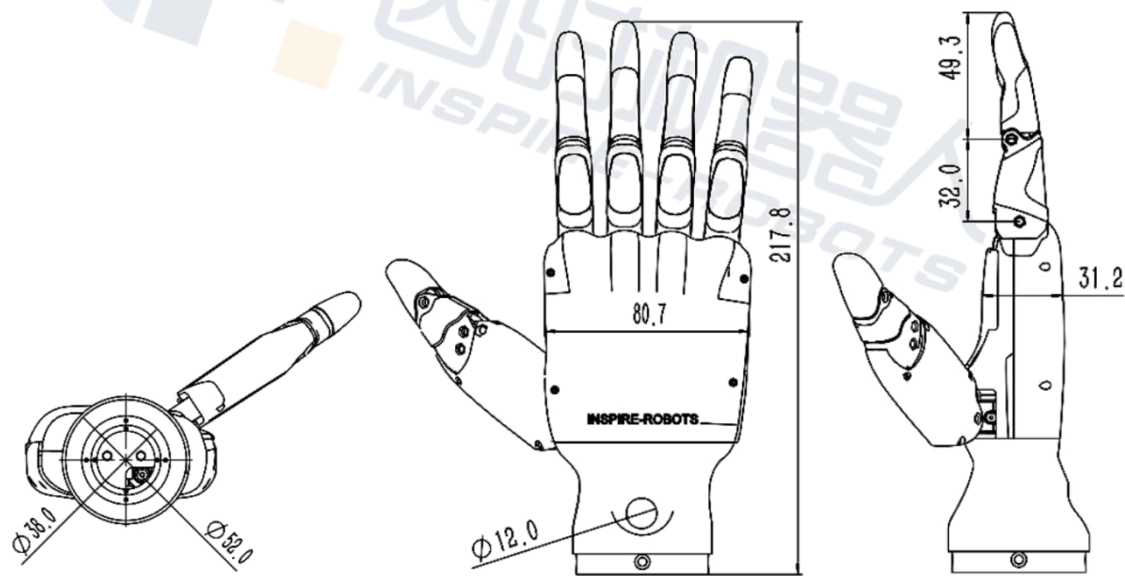


图 1 结构尺寸图

1.3 性能参数

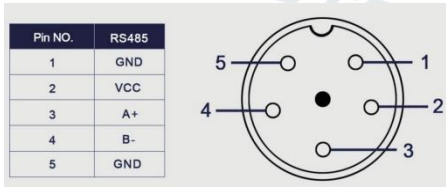
关节总数	12 个
自由度	6 个
力传感器数量	6 个

力传感器分辨率	0.5N
指尖重复定位精度	0.2mm
四指握力	DF:10N BF:4N
拇指握力	DF:15N BF:6N
工作电压	24V
静态电流	0.12A @24V
空载运动平均电流	0.6A @24V
最大握力抓取	2A @24V
通讯接口	RS485

1.4 电气连接

1.4.1. 引脚定义

灵巧手的电器接口如图 2 所示，接口为 GX12 航插端子，端子定义如下。

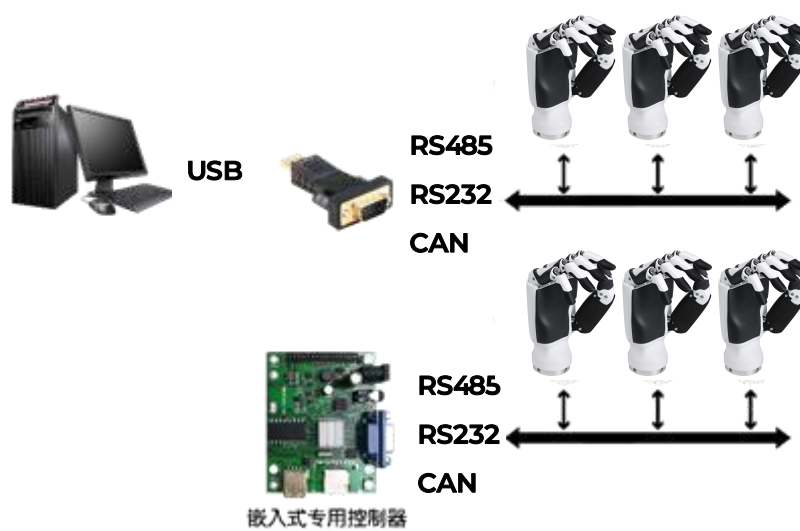
接口类型	引脚定义
RS485	

(图 2) 接口描述

1.4.2. 通信方式

灵巧手使用 RS485 接口时，经过一定的转换电路可以实现 254 只灵巧手并接在同一条总线上（RS232 接口和 CAN 接口灵巧手为定制款）。

灵巧手的通信指令和寄存器对产品用户开放，可通过以上接口与用户的上位机（控制器或通用计算机）连接。用户可以采用 PC 或者嵌入式控制器通过以上接口对灵巧手进行参数配置以及运动控制等。如图 3 所示：



(图 3) 通信方式示意图



2. 通信协议

2.1 通信协议概要

主控单元通过读写灵巧手内部寄存器来实现对灵巧手的状态获取和控制。

读寄存器是指上位系统读取灵巧手内部寄存器数值（可成组写入，成组的是指若干地址相邻寄存器），上位系统向灵巧手发送读指令（包含寄存器组首地址、寄存器组的长度等内容），灵巧手收到数据并校验成功后，灵巧手将相应的寄存器数据内容返回给上位系统。

写寄存器是指上位系统向灵巧手内部寄存器写入相应的数据（可成组写入），上位系统向灵巧手发送写命令（包含寄存器组首地址、欲写入的数据内容）发送给灵巧手，灵巧手收到数据并校验成功后，返回确认信号给上位系统。

2.2 读写寄存器操作

RS485 通讯参数为 8 数据位、1 停止位、无奇偶校验，默认 115200bps。

2.2.1. 读灵巧手寄存器的操作

读灵巧手寄存器的指令帧格式如下，其中 **Address** 为所要读取寄存器的起始地址，**Hands_ID** 是灵巧手的 ID 号，**Address_L** 为 **Address** 的低 8 位，**Address_H** 为 **Address** 的高 8 位，**Register_Length** 为所要读寄存器的长度(单位是 byte)，**checksum** 为校验和，是除应答帧头外其余数据的累加和的低字节。

	数值	说明
byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	0x04	该帧数据部分长度
byte[4]	0x11	读寄存器
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7]	Register_Length	读取寄存器的长度
byte[8]	Checksum	校验和

表示以下4个字节都是数据

命令固定码

灵巧手对读寄存器指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	Register_Length+3	该帧数据部分长度
byte[4]	0x11	说明该帧是对读寄存器命令的回复
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	寄存器数值
byte[7+ Register_Length]	Checksum	校验和

举例说明，以读取灵巧手（ID 为 1）各自由度角度实际值为例。自由度角度实际值存储在寄存器组 **ANGLE_ACT(m)** 中，起始地址为 1546（0x060A），长度为 12byte（0x0C），那么这条指令帧的格式如下：

	数值	说明
byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	0x01	灵巧手 ID 号为 1
byte[3]	0x04	该帧数据部分长度
byte[4]	0x11	读寄存器
byte[5]	0x0A	寄存器起始地址低八位
byte[6]	0x06	寄存器起始地址高八位

byte[7]	0x0C	读取寄存器的长度
byte[8]	0x32	校验和

灵巧手对这条指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	0x01	灵巧手 ID 号
byte[3]	0x0F	该帧数据部分长度 12 + 3
byte[4]	0x11	说明该帧是对读寄存器命令的回复
byte[5]	0x0A	寄存器起始地址低八位
byte[6]	0x06	寄存器起始地址高八位
byte[7] byte[8]	0x6400	转换为整形 100(0x0064)
byte[9] byte[10]	0x6400	转换为整形 100(0x0064)
byte[11] byte[12]	0x6400	转换为整形 100(0x0064)
byte[13] byte[14]	0x6400	转换为整形 100(0x0064)
byte[15] byte[16]	0xE803	转换为整形 1000(0x03E8)
byte[17] byte[18]	0x0000	转换为整形 0(0x0000)
byte[19]	0xAC	校验和

从这条回复帧，可以得出自由度角度实际值分别为 100、100、100、100、1000 和 0。

2.2.2. 写灵巧手寄存器的操作

写灵巧手寄存器的指令帧格式如下，Data[0]- Data[Register_Length-1]为所要写入的数据。

	数值	说明
byte[0]	0xEB	包头

byte[1]	0x90	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	Register_Length+3	该帧数据部分长度
byte[4]	0x12	写寄存器命令标志
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	所要写入寄存器的数据
byte[7+Register_Length]	checksum	校验和

灵巧手对写寄存器指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	4	该帧数据部分长度
byte[4]	0x12	说明该帧是对写寄存器命令的回复
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7]	1	
byte[8]	checksum	校验和

当进行保存参数操作时（将寄存器 **SAVE** 设置为 1），灵巧手在返回回复帧 1 秒后，将返回保存结果信息帧，这条帧的内容如下：

	数值	说明
--	----	----

byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	4	该帧数据部分长度
byte[4]	0x12	说明该帧是对写寄存器命令的回复
byte[5]	0xED	寄存器 SAVE 起始地址低八位
byte[6]	0x03	寄存器 SAVE 起始地址高八位
byte[7]	**	0x00 保存操作成功, 0xFF 保存操作失败
byte[8]	checksum	校验和

举例说明，以设置灵巧手（ID 为 1）各自由度的角度分别为 100、100、100、100、1000 和 0 为例。需要对寄存器组 **ANGLE_SET(m)** 进行修改，该寄存器组的起始地址为 1486（0x05CE），长度为 12byte（0x0C），那么需要发送的指令为：

	数值	说明
byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	0x01	灵巧手 ID 号
byte[3]	0x0F	该帧数据部分长度 12 + 3
byte[4]	0x12	写寄存器命令标志
byte[5]	0xCE	寄存器起始地址低八位
byte[6]	0x05	寄存器起始地址高八位
byte[7] byte[8]	0x6400	转换为整形 100(0x0064)
byte[9] byte[10]	0x6400	转换为整形 100(0x0064)
byte[11] byte[12]	0x6400	转换为整形 100(0x0064)
byte[13] byte[14]	0x6400	转换为整形 100(0x0064)
byte[15] byte[16]	0xE803	转换为整形 1000(0x03E8)

byte[17] byte[18]	0x0000	转换为整形 0(0x0000)
byte[19]	0x70	校验和

灵巧手对这条指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	0x01	灵巧手 ID 号
byte[3]	0x04	该帧数据部分长度
byte[4]	0x12	说明该帧是对写寄存器命令的回复
byte[5]	0xCE	寄存器起始地址低八位
byte[6]	0x05	寄存器起始地址高八位
byte[7]	0x01	
byte[8]	0xEB	校验和

2.3 Modbus 协议概要

Modbus 协议使用的是主从请求应答通讯方式，协议帧包括了功能码、数据域、和 CRC 校验。该系列电缸支持 读取保持寄存器（功能码 0x03）、预置单寄存器（功能码 0x06）、预置多寄存器（功能码 0x10）操作。

2.3.1. 读取保持寄存器 功能码：0x03

主站询问 帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	寄存器数量 (高位)	寄存器数量 (低位)	CRC
	0x11	0x03	0x6B	0x00	0x00	0x02	XXXX

含义：读 17 号（0x11）号从站保持寄存器，起始地址=0x006B；寄存器个数=0x0002，结束地址=0x006B+2-1=0x006C，即读 17 号从站保持寄存器 0x006B-0x006C，共两个寄存器。

从站应答帧格式	从站地址	功能码	字节计数	0x006B 寄存器 (高位)	0x006C 寄存器 (低位)	0x006C 寄存器 (高位)	0x006C 寄存器 (低位)	CRC
	0x01	0x03	0x04	0x00	0x01	0x00	0x02	XXXX

含义：返回 17（0x11）号从站保持寄存器 0x006B-0x006C，共两个寄存器，0x006B 寄存器数值是 0x0001，0x006C 寄存器数值是 0x0002。

2.3.2. 预置单寄存器 功能码：0x06

主站询问帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	数据内容 (高位)	数据内容 (低位)	CRC
	0x11	0x06	0x00	0x6B	0x10	0x00	XXXX

含义：设置 17 号（0x11）号从站保持寄存器，寄存器地址 0x006B，数据内容是 0x1000。

从站应答帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	数据内容 (高位)	数据内容 (低位)	CRC
	0x11	0x06	0x00	0x00	0x00	0x00	XXXX

2.3.3. 预置多寄存器 功能码：0x10

主站询问帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	寄存器数量 (高位)	寄存器数量 (低位)	字节计数	数据 (高位)	数据 (低位)	数据 (高位)	数据 (低位)	CRC
	0x11	0x10	0x00	0x01	0x00	0x02	0x04	0x00	0x0A	0x01	0x02	XXXX

含义：设置 17 号（0x11）号从站保持寄存器，寄存器起始地址 0x0001，寄存器个数为 0x0002，数据内容字节计数是 0x04，数据内容分别是 0x000A、0x0102。

从站应答帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	寄存器数量 (高位)	寄存器数量 (低位)	CRC
	0x11	0x10	0x00	0x01	0x00	0x02	XXXX

2.4 寄存器说明

灵巧手对用户开放的寄存器参数如下：

地址	含义	缩写	长度	读写权限
1000 (0x03E8)	ID	HAND_ID	1byte	W/R
1002 (0x03EA)	波特率设置	REDU_RATIO	1byte	W/R
1004 (0x03EC)	清除错误	CLEAR_ERROR	1byte	W/R
1005 (0x03ED)	保存数据至 Flash	SAVE	1byte	W/R
1006 (0x03EE)	恢复出厂设置	RESET_PARA	1byte	W/R
1008 (0x03F0)	保留	reserve	1byte	W/R
1009 (0x03F1)	受力传感器校准	GESTURE_FORCE_CLB	1byte	W/R
1020 (0x03FC)	各自由度的电缸电流保护值	CURRENT_LIMIT(m)	6short(12byte)	W/R
1032 (0x0408)	各自由度的上电速度设置值	DEFAULT_SPEED_SET(m)	6short(12byte)	W/R
1044 (0x0414)	各自由度的上电力控阈值设置值	DEFAULT_FORCE_SET(m)	6short(12byte)	W/R
1472 (0x05C0)	系统电压	VLTAGE	1short	R
1474 (0x05C2)	各自由度的电缸位置设置值	POS_SET(m)	6short(12byte)	W/R
1486 (0x05CE)	各自由度的角度设置值	ANGLE_SET(m)	6short(12byte)	W/R
1498 (0x05DA)	各自由度的力控阈值设置值	FORCE_SET(m)	6short(12byte)	W/R
1522 (0x05F2)	各自由度的速度设置值	SPEED_SET(m)	6short(12byte)	W/R
1534 (0x05FE)	各自由度的电缸位置实际值	POS_ACT(m)	6short(12byte)	R
1546 (0x060A)	各自由度的角度实际值	ANGLE_ACT(m)	6short(12byte)	R

1582 (0x062E)	各手指的实际受力	FORCE_ACT(m)	6short(12byte)	R
1594 (0x063A)	各自由度的电缸的电流值	CURRENT(m)	6short(12byte)	R
1606 (0x0646)	各自由度的电缸的故障信息	ERROR(m)	6byte	R
1612 (0x064C)	各自由度的状态信息	STATUS(m)	6byte	R
1618 (0x0652)	各自由度的电缸的温度	TEMP(m)	6byte	R
2000 (0x07D0)	当前动作序列检验码 1	ACTION_SEQ_CHECKDATA1	1byte	W/R
2001 (0x07D1)	当前动作序列检验码 2	ACTION_SEQ_CHECKDATA2	1byte	W/R
2002 (0x07D2)	当前动作序列总步骤数	ACTION_SEQ_STEPNUM	1byte	W/R
2016 (0x07E0)	当前动作序列第 0 步设置	ACTION_SEQ_STEP(0)	19short(38byte)	W/R
2054 (0x0806)	当前动作序列第 1 步设置	ACTION_SEQ_STEP(1)	19short(38byte)	W/R
2092 (0x082C)	当前动作序列第 2 步设置	ACTION_SEQ_STEP(2)	19short(38byte)	W/R
2130 (0x0852)	当前动作序列第 3 步设置	ACTION_SEQ_STEP(3)	19short(38byte)	W/R
2168 (0x0878)	当前动作序列第 4 步设置	ACTION_SEQ_STEP(4)	19short(38byte)	W/R
2206 (0x089E)	当前动作序列第 5 步设置	ACTION_SEQ_STEP(5)	19short(38byte)	W/R
2244 (0x08C4)	当前动作序列第 6 步设置	ACTION_SEQ_STEP(6)	19short(38byte)	W/R
2282 (0x08EA)	当前动作序列第 7 步设置	ACTION_SEQ_STEP(7)	19short(38byte)	W/R
2320 (0x0910)	当前动作序列索引号	ACTION_SEQ_INDEX	1byte	W/R
2321 (0x0911)	保存当前动作序列	SAVE_ACTION_SEQ	1byte	W/R
2322 (0x0912)	运行当前动作序列	ACTION_SEQ_RUN	1byte	W/R
2324 (0x0914)	动作序列力控调节值	ACTION_ADJUST_FORCE_SET	2byte	W/R

2.4.1. HAND_ID 灵巧手 ID 号

默认值 1,范围:1-254,可保存。

当一条总线上连接多只灵巧手时，需要给每一只灵巧手分配不同 ID 号。

2.4.2. REDU_RATIO 波特率设置

默认值 0,范围:0-2,可保存。

0:波特率 115200

1:波特率 57600

2:波特率 19200

2.4.3. CLEAR_ERROR 清除错误

默认值 0 范围 0-1 不可保存

写入 1 后，灵巧手将清除可清除的故障(电缸的堵转故障、过流故障、异常故障以及通讯故障)。

注意：电缸的过温故障是不可清除故障，当温度回落后，故障会自动清除。

2.4.4. SAVE 保存参数到 FLASH

默认值 0 范围 0-1 不可保存

写入 1 后，灵巧手将当前参数写入 flash，断电后参数不丢失。

2.4.5. RESET_PARA 恢复出厂设置

默认值 0 范围 0-1 不可保存

写入 1 后，灵巧手的参数将恢复为出厂设置参数。

2.4.6. GESTURE_FORCE_CLB 力传感器校准

当用户设置参数为 1 时，灵巧手启动力传感器校准过程，校准过程持续 6 秒，首先五指完全张开，然后四指（小拇指、无名指、中指、食指）弯曲，然后四指张开，大拇指弯曲，最后大拇指展开。

注意：校准过程必须要保证灵巧手处于空载状态，即手指不能接触任何物体。

2.4.7. CURRENT_LIMIT(m) 各自由度的电缸电流保护值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的电流保护值，当手指运动过程中电流超过该值后，手指会停止运行，并在手指状态寄存器 **STATUS(m)** 中显示手指因为电流保护停止。具体说明见下表。这些参数可断电保存，单位：mA。

地址	名称	说明	数据类型	范围
1020-1021	CURRENT_LIMIT(0)	小拇指电流保护值	short	0-1500
1022-1023	CURRENT_LIMIT(1)	无名指电流保护值	short	0-1500
1024-1025	CURRENT_LIMIT(2)	中指电流保护值	short	0-1500
1026-1027	CURRENT_LIMIT(3)	食指电流保护值	short	0-1500
1028-1029	CURRENT_LIMIT(4)	大拇指弯曲电流保护值	short	0-1500
1030-1031	CURRENT_LIMIT(5)	大拇指旋转电流保护值	short	0-1500

2.4.8. DEFAULT_SPEED_SET(m) 各自由度的上电速度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的上电速度设置值，具体说明见下表。这些参数可断电保存，无范围量纲。

地址	名称	说明	数据类型	范围
1032-1033	DEFAULT_SPEED_SET(0)	小拇指上电初始速度	short	0-1000
1034-1035	DEFAULT_SPEED_SET(1)	无名指上电初始速度	short	0-1000
1036-1037	DEFAULT_SPEED_SET(2)	中指上电初始速度	short	0-1000
1038-1039	DEFAULT_SPEED_SET(3)	食指上电初始速度	short	0-1000
1040-1041	DEFAULT_SPEED_SET(4)	大拇指弯曲上电初始速度	short	0-1000
1042-1043	DEFAULT_SPEED_SET(5)	大拇指旋转上电初始速度	short	0-1000

速度为 1000：表示指手指在空载时从大角度运动到最小角度用时 800ms，如果负载较大时，这个实际速度会有一定程度的降低。

2.4.9. DEFAULT_FORCE_SET(m) 各自由度的上电力控阈值设置值

此寄存器组由六个寄存器组成,分别对应灵巧手六个自由度的上电力控阈值设置值,具体说明见下表。这些参数可断电保存, 单位: g。

地址	名称	说明	数据类型	范围
1044-1045	DEFAULT_FORCE_SET(0)	小拇指上电初始力控	short	0-1000
1046-1047	DEFAULT_FORCE_SET(1)	无名指上电初始力控	short	0-1000
1048-1049	DEFAULT_FORCE_SET(2)	中指上电初始力控	short	0-1000
1050-1051	DEFAULT_FORCE_SET(3)	食指上电初始力控	short	0-1000
1052-1053	DEFAULT_FORCE_SET(4)	大拇指弯曲上电初始力控	short	0-1500
1054-1055	DEFAULT_FORCE_SET(5)	大拇指旋转上电初始力控	short	0-1500

寄存器数值含义是指相应手指指尖提供的握力, 例如 **DEFAULT_FORCE_SET(1)** 设置为 **800** 时, 表示允许无名指手指指尖提供 **800g** 的握力。(如果手指接触物体的部分不是指尖, 那么可提供的握力则有所不同, 具体大小与力臂长度有关, 力臂越短, 提供的握力越大)。

2.4.10. POS_SET(m) 各自由度的电缸位置设置值

此寄存器组由六个寄存器组成, 分别对应灵巧手六个自由度电缸的设置位置, 具体说明见下表。这些参数不能保存, 无单位量纲。

地址	名称	说明	数据类型	范围
1474-1475	POS_SET(0)	小拇指电缸位置设置	short	0-2000
1476-1477	POS_SET(1)	无名指电缸位置设置	short	0-2000
1478-1479	POS_SET(2)	中指电缸位置设置	short	0-2000
1480-1481	POS_SET(3)	食指电缸位置设置	short	0-2000
1482-1483	POS_SET(4)	大拇指弯曲电缸位置设置	short	0-2000
1484-1485	POS_SET(5)	大拇指旋转电缸位置设置	short	0-2000

0 表示: 电缸最短行程, 对应手指角度的最大值 (即手指张开)。

2000 表示: 电缸最大行程, 对应手指角度的最小值 (即手指弯曲)。

-1 表示：电缸不执行任何动作。

不建议用户通过设置这组寄存器来设置给手指的位置角度。

2.4.11. ANGLE_SET(m) 各自由度的角度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的角度设置值，具体说明见下表。这些参数可断电保存。当某一自由度的角度设置值被设置为 0-1000 范围内的数据时，相应的手指会立即动作，如果角度设置值被设置为-1 时，相应的手指无反应，例如将 ANGLE_SET(0)- ANGLE_SET(5)6 个寄存器设置为 500,500,-1,0,500,500, 后小拇指、无名指、大拇指弯曲和大拇指旋转 4 个自由度会运动到 500 的角度，食指运动到 0 的角度，而中指将不会动作（即保持当前实际位置不动作），无单位量纲。

地址	名称	说明	数据类型	范围
1486-1487	ANGLE_SET(0)	小拇指上电初始角度	short	-1,0-1000
1488-1489	ANGLE_SET(1)	无名指上电初始角度	short	-1,0-1000
1490-1491	ANGLE_SET(2)	中指上电初始角度	short	-1,0-1000
1492-1493	ANGLE_SET(3)	食指上电初始角度	short	-1,0-1000
1494-1495	ANGLE_SET(4)	大拇指弯曲上电初始角度	short	-1,0-1000
1496-1497	ANGLE_SET(5)	大拇指旋转上电初始角度	short	-1,0-1000

各自由度的角度定义和运动范围说明如下。

角度	图例说明	范围
小拇指 无名指 中指 食指		$19^{\circ}\sim176.7^{\circ}$
大拇指弯曲角度		$-13^{\circ}\sim53.6^{\circ}$
大拇指旋转角度		$90^{\circ}\sim165^{\circ}$

2.4.12. FORCE_SET (m) 各自由度的力控阈值设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度力控阈值设置值，具体说明见下表。这些参数可断电保存。用户可设置这组寄存器来实现对手指握力的控制，例如将 **FORCE_SET(3)** 设置为 **300**，当前食指实际角度 **ANGLE_ACT(3)** 为 **1000**（即完全张开状态），当用户将食指设置角度 **ANGLE_SET(3)** 设置为 **0** 后，食指将在向手掌弯曲运动，在弯曲过程中如果检测到食指实际受力（即 **FORCE_ACT(3)**，具体说明参见 2.4.17）到达 **300** 后，食指停止运动，如果食指实际受力没有到达 **300**，那么食指将运动到设置角度，单位：**g**。

地址	名称	说明	数据类型	范围
1498-1499	FORCE_SET(0)	小拇指力控设置值	short	0-1000
1500-1501	FORCE_SET(1)	无名指力控设置值	short	0-1000
1502-1503	FORCE_SET(2)	中指力控设置值	short	0-1000
1504-1505	FORCE_SET(3)	食指力控设置值	short	0-1000
1506-1507	FORCE_SET(4)	大拇指弯曲力控设置值	short	0-1000
1508-1509	FORCE_SET(5)	大拇指旋转力控设置值	short	0-1000

寄存器数值含义是指相应手指指尖提供的握力，例如 **DEFAULT_FORCE_SET(1)** 设置为 **800** 时，表示允许无名指手指指尖提供 **800g** 的握力。（如果手指接触物体的部分不是指尖，那么可提供的握力则有所不同，具体大小与力臂长度有关，力臂越短，提供的握力越大）。

2.4.13. SPEED_SET(m) 各自由度的速度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的速度设置值，具体说明见下表。这些参数可断电保存，无单位量纲。

地址	名称	说明	数据类型	范围
1522-1523	SPEED_SET(0)	小拇指速度设置值	short	0-1000
1524-1525	SPEED_SET(1)	无名指速度设置值	short	0-1000
1526-1527	SPEED_SET(2)	中指速度设置值	short	0-1000
1528-1529	SPEED_SET(3)	食指速度设置值	short	0-1000
1530-1531	SPEED_SET(4)	大拇指弯曲速度设置值	short	0-1000

1532-1533	SPEED_SET(5)	大拇指旋转速度设置值	short	0-1000
-----------	--------------	------------	-------	--------

速度为 1000：表示指手指在空载时从大角度运动到最小角度用时 800ms，如果负载较大时，这个实际速度会有一定程度的降低。

2.4.14. POS_ACT(m) 各自由度的电缸位置实际值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度电缸的当前实际位置，具体说明见下表。这些参数为只读参数，无单位量纲。

地址	名称	说明	数据类型	范围
1534-1535	POS_ACT(0)	小拇指电缸位置实际值	short	0-2000
1536-1537	POS_ACT(1)	无名指电缸位置实际值	short	0-2000
1538-1539	POS_ACT(2)	中指电缸位置实际值	short	0-2000
1540-1541	POS_ACT(3)	食指电缸位置实际值	short	0-2000
1542-1543	POS_ACT(4)	大拇指弯曲电缸位置实际值	short	0-2000
1544-1545	POS_ACT(5)	大拇指旋转电缸位置实际值	short	0-2000

0 表示：电缸最短行程，对应手指角度的最大值。

2000 表示：电缸最大行程，对应手指角度的最小值。

2.4.15. ANGLE_ACT(m) 各自由度的角度实际值

地址	名称	说明	数据类型	范围
1546-1547	ANGLE_ACT(0)	小拇指角度实际值	short	0-1000
1548-1549	ANGLE_ACT(1)	无名指角度实际值	short	0-1000
1550-1551	ANGLE_ACT(2)	中指角度实际值	short	0-1000
1552-1553	ANGLE_ACT(3)	食指角度实际值	short	0-1000
1554-1555	ANGLE_ACT(4)	大拇指弯曲角度实际值	short	0-1000
1556-1557	ANGLE_ACT(5)	大拇指旋转角度实际值	short	0-1000

2.4.16. FORCE_ACT(m) 各手指的实际受力

此寄存器组由六个寄存器组成，分别对应灵巧手六个手指的实际受力值，具体说明见下表。这些参数为只读参数，单位：g。

地址	名称	说明	数据类型	范围
1582-1583	FORCE_ACT(0)	小拇指实际受力值	short	0-1000
1584-1585	FORCE_ACT(1)	无名指实际受力值	short	0-1000
1586-1587	FORCE_ACT(2)	中指实际受力值	short	0-1000
1588-1589	FORCE_ACT(3)	食指实际受力值	short	0-1000
1590-1591	FORCE_ACT(4)	大拇指弯实际受力值	short	0-1000
1592-1593	FORCE_ACT(5)	大拇指旋转实际受力值	short	0-1000

2.4.17. CURRENT(m) 各自由度的电缸的电流值

此寄存器组由六个寄存器组成，分别对应灵巧手六个电缸的电流值，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型	范围	单位
1594-1595	CURRENT(0)	小拇指电缸电流值	short	0-1000	mA
1596-1597	CURRENT(1)	无名指电缸电流值	short	0-1000	mA
1598-1599	CURRENT(2)	中指电缸电流值	short	0-1000	mA
1600-1601	CURRENT(3)	食指电缸电流值	short	0-1000	mA
1602-1603	CURRENT(4)	大拇指弯曲电缸电流值	short	0-1000	mA
1604-1605	CURRENT(5)	大拇指旋转电缸电流值	short	0-1000	mA

2.4.18. ERROR(m) 各电缸的故障信息

此寄存器组由六个寄存器组成，分别对应灵巧手六个电缸的故障码，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型
1606	ERROR(0)	小拇指电缸故障码	byte
1607	ERROR (1)	无名指电缸故障码	byte
1608	ERROR (2)	中指电缸故障码	byte

1609	ERROR (3)	食指电缸故障码	byte
1610	ERROR (4)	大拇指弯曲电缸故障码	byte
1611	ERROR (5)	大拇指旋转电缸故障码	byte

故障码含义如下表所示，例如 **ERROR (3)** 为 **0x06**（转换为二进制为 **00000110**）时，说明食指电缸出现过温故障和过流故障。

	说明
Bit0	堵转故障
Bit1	过温故障
Bit2	过流故障
Bit3	电机异常
Bit4	通讯故障

2.4.19. STATUS(m)各自由度的状态信息

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的状态信息，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型
1612	STATUS(0)	小拇指状态信息	byte
1613	STATUS(1)	无名指状态信息	byte
1614	STATUS(2)	中指状态信息	byte
1615	STATUS(3)	食指状态信息	byte
1616	STATUS(4)	大拇指弯曲状态信息	byte
1617	STATUS(5)	大拇指旋转状态信息	byte

状态码含义如下：

状态码	说明
0	正在松开
1	正在抓取
2	位置到位停止

3	力控到位停止
5	电流保护停止
6	电缸堵转停止
7	电缸故障停止

2.4.20. TEMP(m) 各电缸的温度

此寄存器组由六个寄存器组成，分别对应灵巧手六个电缸温度值，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型	范围	单位
1618	TEMP(0)	小拇指电缸温度值	byte	0-100	°C
1619	TEMP(1)	无名指电缸温度值	byte	0-100	°C
1620	TEMP(2)	中指电缸温度值	byte	0-100	°C
1621	TEMP(3)	食指电缸温度值	byte	0-100	°C
1622	TEMP(4)	大拇指弯曲电缸温度值	byte	0-100	°C
1623	TEMP(5)	大拇指旋转电缸温度值	byte	0-100	°C

2.4.21. 动作序列相关的寄存器

地址	含义	缩写	长度	读写权限
2000 (0x07D0)	当前动作序列 检验码 1	ACTION_SEQ_CHECKDAT A1	1byte	W/R
2001 (0x07D1)	当前动作序列 检验码 2	ACTION_SEQ_CHECKDAT A2	1byte	W/R
2002 (0x07D2)	当前动作序列 总步骤数	ACTION_SEQ_STEPNUM	1byte	W/R
2016 (0x07E0)	当前动作序列 第 0 步设置	ACTION_SEQ_STEP(0)	19short(38byte)	W/R
2054 (0x0806)	当前动作序列 第 1 步设置	ACTION_SEQ_STEP(1)	19short(38byte)	W/R

2092 (0x082C)	当前动作序列 第 2 步设置	ACTION_SEQ_STEP(2)	19short(38byte)	W/R
2130 (0x0852)	当前动作序列 第 3 步设置	ACTION_SEQ_STEP(3)	19short(38byte)	W/R
2168 (0x0878)	当前动作序列 第 4 步设置	ACTION_SEQ_STEP(4)	19short(38byte)	W/R
2206 (0x089E)	当前动作序列 第 5 步设置	ACTION_SEQ_STEP(5)	19short(38byte)	W/R
2244 (0x08C4)	当前动作序列 第 6 步设置	ACTION_SEQ_STEP(6)	19short(38byte)	W/R
2282 (0x08EA)	当前动作序列 第 7 步设置	ACTION_SEQ_STEP(7)	19short(38byte)	W/R
2320 (0x0910)	当前动作序列 索引号	ACTION_SEQ_INDEX	1byte	W/R
2321 (0x0911)	保存当前动作 序列	SAVE_ACTION_SEQ	1byte	W/R
2322 (0x0912)	运行当前动作 序列	ACTION_SEQ_RUN	1byte	W/R

动作序列是指一组连续手势动作，一套动作序列数据包含多个手势（最多 8 个）的设定角度、速度、力控值以及延时时间（ms），灵巧手可按照这数据按顺序和延时时间做出每一步的手势，行程一套连续的动作序列。

灵巧手内存可存储 40 套动作序列数据，不同动作序列同过索引号标识，这些数据的读取稍有不同。

读取动作序列数据：

- 1) 首先设置当前动作标识号寄存器（ACTION_SEQ_INDEX）；
- 2) 读取 ACTION_SEQ_CHECKDATA1 、 ACTION_SEQ_CHECKDATA2 数据。如果 ACTION_SEQ_CHECKDATA1 寄存器数值为 0x90 且 ACTION_SEQ_CHECKDATA10 寄存器数值为 xEB,表示该动作序列数据有效，否则该动作序列数据无效。

- 3) 从上一步判断出数据有效的话，读取 ACTION_SEQ_STEPNUM 、 ACTION_SEQ_STEP(0-7)中的数据。

写动作序列数据：

- 1) 首先设置当前动作标识号寄存器 (ACTION_SEQ_INDEX) ；
- 2) 将要写入的数据写入 ACTION_SEQ_STEPNUM 、ACTION_SEQ_STEP(0-7)寄存器。
- 3) 将 0x90 和 0xEB 写入 ACTION_SEQ_CHECKDATA1 和 ACTION_SEQ_CHECKDATA2 寄存器。
- 4) 设置动作序列库保存寄存器 SAVE_ACTION_SEQ 为 1, 灵巧手将 ACTION_SEQ_CHECKDATA1 、 ACTION_SEQ_CHECKDATA2、 ACTION_SEQ_STEPNUM 、 ACTION_SEQ_STEP(0-7)中的数据烧写到与索引号相对应的 Flash 地址内。

运行动作序列：

- 1) 首先设置当前动作标识号寄存器 (ACTION_SEQ_INDEX) 为欲运行的动作序列的索引号；
- 2) 设置 ACTION_SEQ_RUN 寄存器为 1, 灵巧手开始执行动作序列，运行结束后，寄存器值自动清零。

3. 常规指令示例

示例1（修改ID）

将ID号为1的灵巧手改为ID号2：

指令帧：EB 90 01 04 12 E8 03 02 04

应答帧：90 EB 02 04 12 E8 03 01 04

指令帧中“EB 90”为包头，“01”为当前灵巧手ID，“04”为帧数据长度，“12”为写寄存器命令标志，“E8 03”为寄存器地址，“02”为写入寄存器的数据，“04”为校验和。应答帧中“90 EB”为包头，“02”为当前灵巧手ID，“04”为帧数据长度，“12”为写寄存器命令标志，“E8 03”为寄存器地址，“04”为校验和。

ID号修改完成后立即有效，之后的通信ID即为新的ID号，若需要断电保存该新ID号，则需要配合“参数保存”操作。

示例2（修改波特率）

将ID为1的灵巧手波特率由115200修改为57600：

指令帧：EB 90 01 04 12 EA 03 01 05

应答帧：90 EB 01 04 12 EA 03 01 05

波特率修改完不会立即生效，配合“参数保存”操作，断电上电可实现。

示例3（清除错误）

将ID为1的灵巧手的故障信息清除掉，当灵巧手发生堵转，过流，异常等故障可通过清除故障指令恢复正常运行：

指令帧：EB 90 01 04 12 EC 03 01 07

应答帧：90 EB 01 04 12 EC 03 01 07

示例4（保存参数）

在进行修改ID或者波特率等参数时，灵巧手将参数写入flash，实现断电后参数不丢失。

指令帧：EB 90 01 04 12 ED 03 01 08

应答帧：90 EB 01 04 12 ED 03 01 08, 90 EB 01 04 11 ED 03 00 06

保存参数指令会回复两条应答帧。

示例 5（受力传感器校准）

指令帧：EB 90 01 04 12 F1 03 01 0C

应答帧：90 EB 01 04 12 F1 03 01 0C, 90 EB 01 04 11 02 00 00 18

执行传感器校准时会收到一条应答帧，校准完成后会收到第二条应答帧，校准时需保证灵巧手处于空载状态。

示例 6（设置各自由度的力控阈值）

将 ID 为 1 的灵巧手各自由度力控阈值均设置为 1000：

指令帧：EB 90 01 0F 12 DA 05 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 83

回复帧：90 EB 01 04 12 DA 05 01 F7

设置完力控阈值后会立即生效，断电再上电不保存。

示例 7（设置各自由度的速度）

将 ID 为 1 的灵巧手各自由度速度均设置为 1000：

指令帧：EB 90 01 0F 12 F2 05 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 9B

回复帧：90 EB 01 04 12 F2 05 01 0F

设置完速度后会立即生效，断电再上电不保存。

示例 8（设置各自由度的角度）

将 ID 为 1 的灵巧手各自由度角度均设置为 1000：

指令帧：EB 90 01 0F 12 CE 05 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 77

回复帧：90 EB 01 04 12 CE 05 01 EB

示例 9（读取当前各自由度的角度位置）

读取 ID 为 1 的灵巧手当前各自由度的角度位置：

指令帧：EB 90 01 04 11 0A 06 0C 32

回复帧：见本文 2.2.1 举例说明