ZHAOWE兆威

2025 年 05 月 21 日

目录

1、 注意 1

2、 产品介绍 1

3、 产品型号、组成和尺寸 1

3.1 型号 1

3.2 材质 1

3.3 结构尺寸 2

3.4 运动空间 2

3.5 手腕连接尺寸 5

4、 技术参数 5

4.1 载荷和速度 5

4.2 重量 6

4.3 通讯接口 6

4.4 其他参数 6

5、 通信控制协议 7

5.1 通信控制协议概要 7

5.2 MODBUS 协议概述 8

5.2.1 MODBUS 读寄存器操作 8

5.2.2 MODBUS 写寄存器操作 9

5.3 灵巧手寄存器说明 10

5.3.1 灵巧手寄存器定义 10

1、注意

注意：ZWHAND 仿生人形灵巧手不具有防水能力。

ZWHAND 灵巧手内部配备了精密的电子元件、动力马达以及复杂的机械结构。一旦这些组件遭遇液体侵入，将会引发电路短路、部件生锈等严重后果，进而导致设备遭受无法逆转的损坏。因此，用户在操作过程中必须严格确保无任何液体渗透进灵巧手内部，并应避免在湿度过高或重度灰尘污染的环境中操作该设备，以防止潜在的损害发生。

ZWHAND 灵巧手采用了高强度铝合金与优质不锈钢作为主要骨架材料，以确保结构的稳固性和耐用性。然而，用户在使用时仍需特别注意，确保所施加的负载不超过其设计规格（食指 12N,其余手指 5N）。任何超出设计承载范围的负重都可能导致金属部件发生永久变形，甚至断裂，进而对内部结构造成不可恢复的破坏。此外，摔落或遭受重物撞击等意外情况也将对 ZWHAND 灵巧手的机械结构和电路系统构成严重威胁，可能导致功能失效。

2、产品介绍

ZWHAND 灵巧手配备了 17 个精密运动关节，每个关节均内置高性能电机，从而实现了 17 个独立的主动自由度。通过内嵌的智能电机控制算法，该灵巧手能够模拟人手的各种复杂抓取动作。其典型应用领域广泛，包括但不限于机器人末端执行器、教育科研装备以及工厂自动化应用。

3、产品型号、组成和尺寸

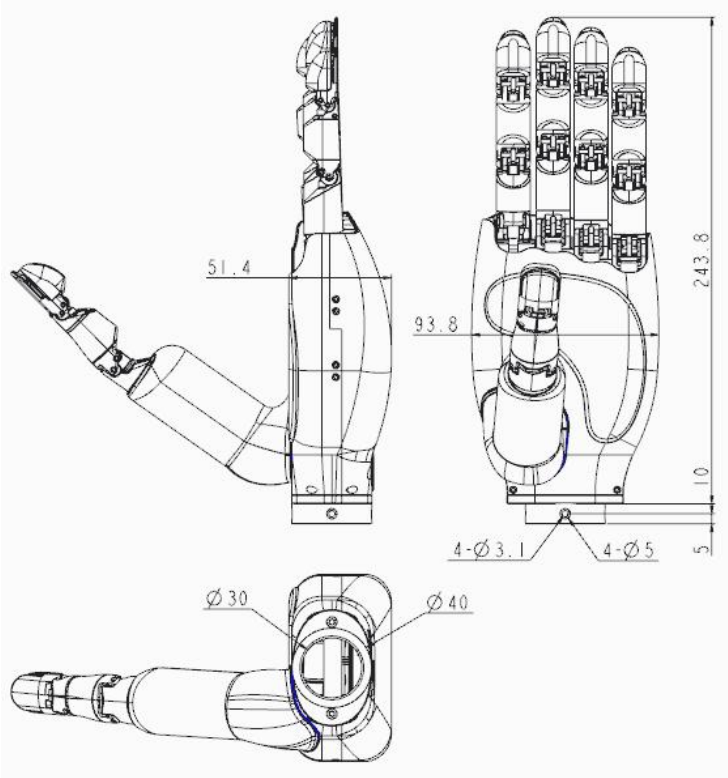
3.1 型号

ZWHAND-DM17（17 主动自由度，皮肤可选配）

3.2 材质

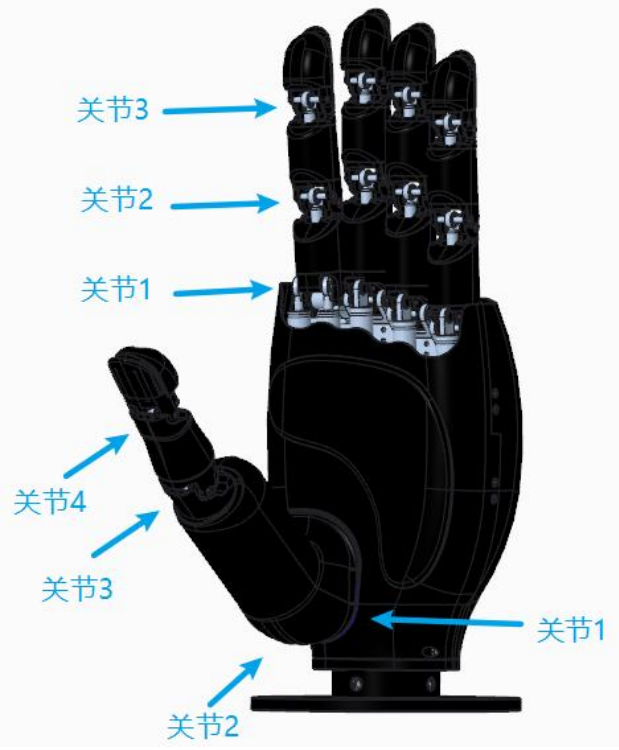
铝合金，不锈钢，硅胶，塑料。

3.3 结构尺寸

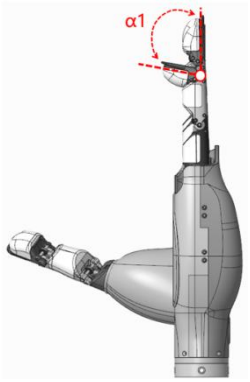
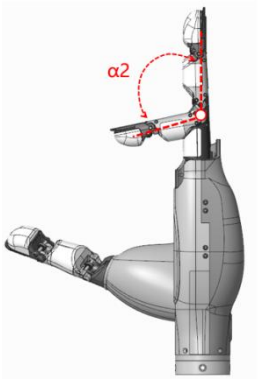


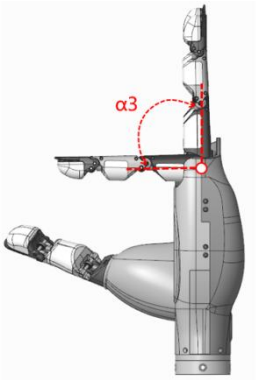
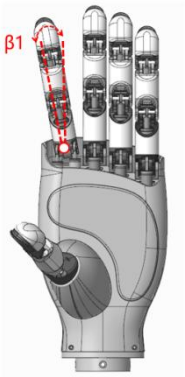
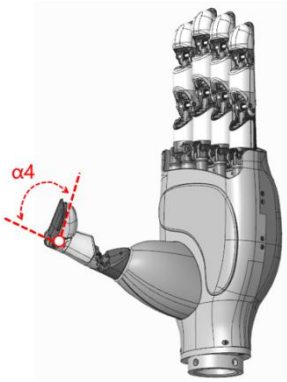
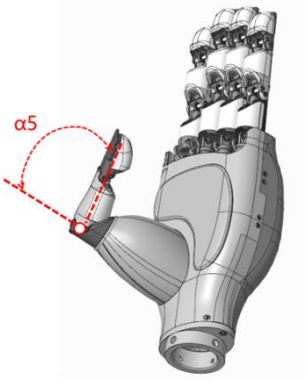
3.4 运动空间

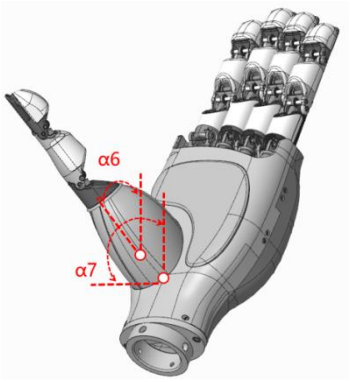
灵巧手关节运动空间如下表所示。



分类		工作空间（角度）	
大拇指	关节 1	60	30-90
	关节 2	45	45-90
	关节 3	70	10-80
	关节 4	80	0-80
食指	关节 1	90	0-90
	关节 2	90	0-90
	关节 3	80	0-80
中指	关节 1	90	0-90
	关节 2	90	0-90
	关节 3	80	0-80
无名指	关节 1	90	0-90
	关节 2	90	0-90
	关节 3	80	0-80
小指	关节 1	90	0-90
	关节 2	90	0-90
	关节 3	80	0-80

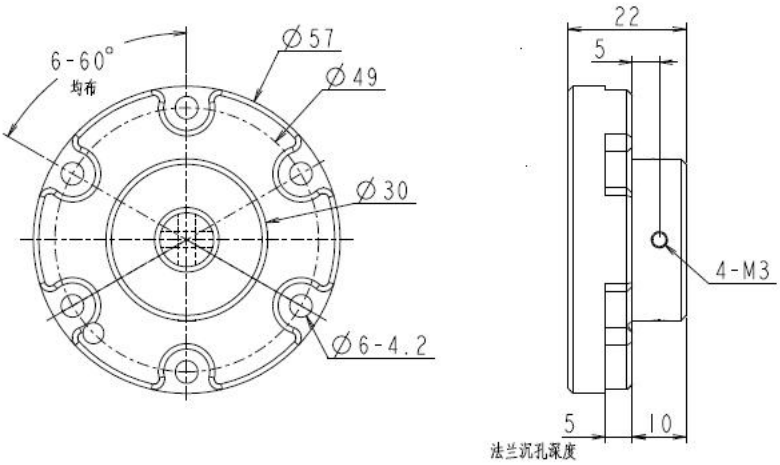
关节	图例说明	运动角度
四指远端指间关节		α1:0-80°
四指近端指间关节		α2:0-90°

四指掌指关节		α3:0-90°
食指掌指关节摆角		β1:0-10°
大拇指远端指间关节		α4:0-80°
大拇指掌指关节		α5:10-80°

大拇指指根关节		α6:30-90° α7:45-90°

3.5 手腕连接尺寸

兆威灵巧手标配的机械接口尺寸如图所示，也可按客户需求定制.



4、技术参数

4.1 载荷和速度

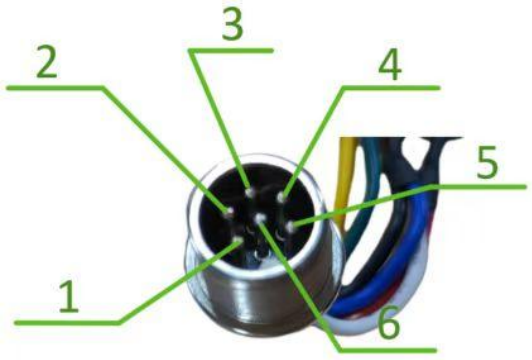
参数	值
手指全开到全闭最快时间	0.45s
手指全闭到全开最快时间	0.45s
大拇指完全闭合时间	0.38s
拇指完全张开时间	0.38s
最大提重物（重握）	3kg

4.2 重量

参数	值
重量（含手腕）	1000g

4.3 通讯接口

ZWHAND 手腕内置有 RS485/CANFD 接口的高性能驱控板，该板提供了便捷的 6 芯 M12 航空头公座转换选项。航空头公座被稳固地安装于手腕组件的背面。面对公座正面时，其顺时针方向排列的 6 根电芯信号定义具体为：



序号（PIN）	说明
1	485+/CAN_H
2	485-/CAN_L
3	GND
4	VCC
5	VCC
6	GND

4.4 其他参数

参数	数值
工作电压	DC12V
工作电流	3A
静态电流	≤300mA
运行环境温度范围	-10℃~40℃
关节总数	17
主动自由度	17
整手握力	30N
四指指尖压力	5~12N

5、通信控制协议

5.1 通信控制协议概要

本灵巧手产品使用到的通信控制协议，是基于硬件接口 RS485 的 Modbus RTU 协议。通过对 Modbus RTU 协议中定义的寄存器进行读写操作，在数据命令通过校验成功后，即可实现对灵巧手的状态读取、单个或多个手指关节动作控制。

读寄存器操作是指上位机系统读取灵巧手预先定义的相关数据参数；上位机系统通过 485 总线，发送指定寄存器地址的读取命令给到灵巧手系统，灵巧手在收到数据并校验通过后，根据当前请求读取的寄存器地址，通过 485 总线反馈相应的数据给到上位机系统；

写寄存器操作是指上位机系统对灵巧手特定的寄存器地址写入的相关数据参数控制灵巧手动作；上位机系统利用 485 总线发送写命令至灵巧手系统，指定寄存器地址。灵巧手系统在接收到数据并完成校验后，会根据请求中的寄存器地址执行相应动作，并通过 485 总线向上位机系统反馈操作状态或结果。

数据校验过程涉及对请求命令的数据执行循环冗余校验，以确保数据的有效性。校验方法采用 CRC16（多项式 0x3D65），校验范围覆盖一帧报文中除最后两字节校验值外的所有数据。校验通过则正常执行命令，否则不响应请求。

该基于硬件接口 RS485 的 Modbus RTU 协议具有以下特性：

RS485-Modbus RTU 协议		
参数	值	备注
总线波特率	9600bps 38400bps 115200bps 1000000bps 2000000bps 可选	可配置，默认 115200bps
数据位	8 位	NA
停止位	1 位	NA
奇偶校验	无	NA
Modbus RTU 从站地址	0x01	默认 0x01，可配置
校验算法	CRC-16	多项式：0x3D65
MODBUS RTU 协议支持的功能码	0x03（读保持寄存器） 0x04（读输入寄存器） 0x06（写单个保持寄存器） 0x10（写多个保持寄存器）	NA
MODBUS RTU 协议支持的错误码	0x01（非法的功能码） 0x02（非法的寄存器地址） 0x03（非法的请求值）	NA

5.2 MODBUS 协议概述

MODBUS 协议是一个主从体系结构，主设备从从设备获取信息，也可以写入从设备的寄存器；从设备响应主设备的请求，无法启动信息传输；MODBUS 协议的核心是协议数据单元（PDU），PDU 由功能码与数据组成，功能码指示主设备如何与从设备 ID 中指定的从设备进行交互。根据发送的功能代码，主设备可以读取从机的一个寄存器，或写入它们。

5.2.1 MODBUS 读寄存器操作

读寄存器操作主要包括有：读取保持寄存器（功能码 0x03）、读取输入寄存器（功能码 0x04）。

读取保持寄存器（功能码 0x03）或读取输入寄存器（功能码 0x04），字节指令操作，支持单个或多个数据（寄存器）读取，同时读取多个数据必须是连续地址的数据（寄存器）。

数据帧格式：地址+功能码（读）+数据+校验码		
请求数据帧格式	地址+功能码（0x03、0x04）+寄存器起始地址（16bit MSB ）+寄存器数量（16bit MSB ）+校验码	
字节	参数	说明
byte0	address	从站地址（灵巧手 ID）
byte1	function code	功能码，指定读或写寄存器操作
byte2	register address high byte	寄存器高地址
byte3	register address low byte	寄存器低地址
byte4	register number high byte	寄存器数量高字节
byte5	register number low byte	寄存器数量低字节
byte6	CRC16 low byte	CRC16 校验值低字节
byte7	CRC16 high byte	CRC16 校验值高字节
回复数据帧格式	地址+功能码（0x04）+寄存器数据长度+校验码	
byte0	address	从站地址（灵巧手 ID）
byte1	function code	功能码，指定读或写寄存器操作
byte2	register value length	寄存器数据长度
byte3	register value high byte	第 1 个寄存器数据高字节
byte4	register value low byte	第 1 个寄存器数据低字节
· · ·	register value high byte	第 n 个寄存器数据高字节
· · ·	register value low byte	第 n 个寄存器数据低字节
Byte(4 + 2*(n-1)+1)	CRC16 low byte	CRC16 校验值低字节
Byte(4 + 2*(n-1)+2)	CRC16 high byte	CRC16 校验值高字节

5.2.2 MODBUS 写寄存器操作

写寄存器操作主要包括有：写单个保持寄存器（功能码 0x06），写多个保持寄存器（功能码 0x10）；

写单个保持寄存器（功能码 0x06）通过字节指令操作，支持单个数据（寄存器）的写入。

数据帧格式：地址+功能码（写）+数据+校验码		
请求数据帧格式	地址+功能码（0x06）+寄存器起始地址（16bit MSB ）数据+校验码（16bit LSB）	
字节	参数	说明
byte0	address	从站地址（灵巧手 ID）
byte1	function code	功能码，指定读或写寄存器操作
byte2	register address high byte	寄存器高地址
byte3	register address low byte	寄存器低地址
byte4	register number high byte	寄存器数量高字节
byte5	register number low byte	寄存器数量低字节
byte6	total bytes	数据字节总数
byte7	register data1	寄存器数据 1
byte8		
byte9	register data2	寄存器数据 2
byte10		
byteN	register dataM	寄存器数据 M
byteN+1		
byteN+2	CRC16 low byte	CRC16 校验值低字节
byteN+3	CRC16 high byte	CRC16 校验值高字节
回复数据帧格式	地址+功能码(0x06)+寄存器数据长度+校验码	
byte0	address	从站地址（灵巧手 ID）
byte1	function code	功能码，指定读或写寄存器操作
byte2	register address high byte	寄存器高地址
byte3	register address low byte	寄存器低地址
byte4	register number high byte	寄存器数量高字节
byte5	register number low byte	寄存器数量低字节
byte6	CRC16 low byte	CRC16 校验值低字节
byte7	CRC16 high byte	CRC16 校验值高字节

写多个保持寄存器（功能码 0x10）通过字节指令操作，支持单个或多个连续地址数据（寄存器）的写入，要求所有写入的数据必须来自连续地址。

数据帧格式：地址+功能码（写）+数据+校验码		
请求数据帧格式	地址+功能码（0x10）+寄存器起始地址（16bit MSB ）+寄存器数量（16bit MSB ）+字节数+数据+校验码（16bit LSB）	
字节	参数	说明
byte0	address	从站地址（灵巧手 ID）
byte1	function code	功能码，指定读或写寄存器操作
byte2	register address high byte	寄存器高地址
byte3	register address low byte	寄存器低地址
byte4	register number high byte	寄存器数量高字节
byte5	register number low byte	寄存器数量低字节
byte6	CRC16 low byte	CRC16 校验值低字节
byte7	CRC16 high byte	CRC16 校验值高字节
回复数据帧格式	地址+功能码（0x10）+寄存器数据长度+校验码	
byte0	address	从站地址（灵巧手 ID）
byte1	function code	功能码，指定读或写寄存器操作
byte2	register address high byte	寄存器高地址
byte3	register address low byte	寄存器低地址
byte4	register number high byte	寄存器数量高字节
byte5	register number low byte	寄存器数量低字节
byte6	CRC16 low byte	CRC16 校验值低字节
byte7	CRC16 high byte	CRC16 校验值高字节

5.3 灵巧手寄存器说明

5.3.1 灵巧手寄存器定义

寄存器地址	缩写	读 写 权限	地址宽度	含义
0x0000	HAND_ID	R/W	16 位	灵巧手 ID（从站地址）
0x0001	BUS_BAUD_RATE	R/W	16 位	总线波特率设置寄存器
0x0002	CLEAR_ERROR	R/W	16 位	错误清除
0x0003	POWER_OFF_SAVE_DTAT	R/W	16 位	下电保存数据
0x0005~0x0015	MOTOR_SPEED	R/W	16 位	设置电机 1~ 17 的最大运行转速
0x0016~0x0026	MOTOR_CURRENT	R/W	16 位	设置电机 1~ 17 的最大运行电流
0x0027~0x0037	MOTOR_STOP	R/W	16 位	设置电机 1~ 17 的停止运行
0x0038~0x0048	MOTOP_ABS_POS	R/W	16 位	设置电机 1~ 17 的绝对位置
0x0049~0x0059	MOTOP_INC_POS	R/W	16 位	设置电机 1~ 17 的增量位置
0x005A~0x006A	MOTOR_ZERO	R/W	16 位	设置电机 1~ 17 单个或多个电机零点标定
0x006B	ALL_STEP_MOTOR_ZERO	R/W	16 位	设置所有步进电机零点标定
0x006C	ALL_MOTOR_ZERO	R/W	16 位	设置整手的所有电机零点标定

5.3.1.1 HAND_ID 配置灵巧手标识符（从设备 ID）

说明： 本灵巧手产品具备在线调整灵巧手标识符（从站地址）能力，目前支持的灵巧手标识符范围为 0x01~0xFE，当需要调整灵巧手标识符时，只需修改灵巧手标识符配置寄存器 HAND_ID 的数值即可，配置完新的灵巧手标识符后，重新给系统上电，系统即可按照新的灵巧书标识符工作。				
灵巧手标识符配置请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	灵巧手标识符配置寄存器高地址	uint16_t	0x0000
Byte3	0xXX	灵巧手标识符配置寄存器低地址		
Byte4	0xXX	灵巧手标识符配置寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	灵巧手标识符配置寄存器数量高字节		
Byte6	2*寄存器数量 N	字节数	uint8_t	0x02
Byte7	0xXX	灵巧手标识符高字节	uint16_t	0x01~0xFE
Byte8	0xXX	灵巧手标识符低字节		
Byte9	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte 10	0xXX	校验码高字节		
灵巧手标识符配置回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	灵巧手标识符配置寄存器高地址	uint16_t	0x0000
Byte3	0xXX	灵巧手标识符配置寄存器低地址		
Byte4	0xXX	灵巧手标识符配置寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	灵巧手标识符配置寄存器数量高字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：配置灵巧手标识符为 0x01				
请求	01 10 00 00 00 01 02 00 01 67 90			
回复	01 10 00 00 00 01 01 C9			

5.3.1.2 BUS_BAUD_RATE 配置 RS485 总线波特率

说明： 本灵巧手产品具备在线调整总线波特率能力，目前支持的总线波特率有 5 个等级 level1~level5，分别对应波特率：9600bps（level1）、38400bps（level2）、115200bps（level3）、1000000bps（level4）、2000000bps（level5）。当需要调整总线波特率时，只需修改总线波特率配置寄存器 BUS_BAUD_RATE 的波特率等级即可，配置完波特率等级后，重新给系统上电，系统即可按照新的总线波特率运行。				
总线波特率配置请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	波特率等级配置寄存器高地址	uint16_t	0x0001
Byte3	0xXX	波特率等级配置寄存器低地址		
Byte4	0xXX	波特率等级配置寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	波特率等级配置寄存器数量高字节		
Byte6	2*寄存器数量 N	字节数	uint8_t	0x02
Byte7	0xXX	波特率等级高字节	uint16_t	0x01~0x05
Byte8	0xXX	波特率等级低字节		
Byte9	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte10	0xXX	校验码高字节		
总线波特率配置回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	波特率等级配置寄存器地址高字节	uint16_t	0x0005~0x0015
Byte3	0xXX	波特率等级配置寄存器地址低字节		
Byte4	0xXX	波特率等级配置寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	波特率等级配置寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：配置总线波特率为 115200bps				
请求	01 10 00 01 00 01 02 00 03 E7 80			
回复	01 10 00 01 00 01 50 09			

5.3.1.3 CLEAR_ERROR 故障清除

说明： 本灵巧手产品具备在线清除已出现故障的能力。当需要清除故障时，只需往故障清除寄存器 CLEAR_ERROR 写入 1 即可。				
清除故障请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	清除故障寄存器高地址	uint16_t	0x0002
Byte3	0xXX	清除故障寄存器低地址		
Byte4	0xXX	清除故障寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	清除故障寄存器数量高字节		
Byte6	2*寄存器数量 N	字节数	uint8_t	0x02
Byte7	0xXX	清除故障命令高字节	uint16_t	0x01
Byte8	0xXX	清除故障命令低字节		
Byte9	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte10	0xXX	校验码高字节		
清除故障回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	清除故障寄存器地址高字节	uint16_t	0x0002
Byte3	0xXX	清除故障寄存器地址低字节		
Byte4	0xXX	清除故障寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	清除故障寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：清除故障				
请求	01 10 00 02 00 01 02 00 01 66 72			
回复	01 10 00 02 00 01 A0 09			

5.3.1.4 POWER_OFF_SAVE_DTAT 下电保存数据

说明： 本灵巧手产品需要在下电前保存当前相关参数以便在下一次的快速启动。下电前保存数据，需要往下电保存参数寄存器 POWER_OFF_SAVE_DTAT 写入命令 1 即可。				
下电保存数据请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	下电保存数据寄存器高地址	uint16_t	0x0003
Byte3	0xXX	下电保存数据寄存器低地址		
Byte4	0xXX	下电保存数据寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	下电保存数据寄存器数量高字节		
Byte6	2*寄存器数量 N	字节数	uint8_t	0x02
Byte7	0xXX	下电保存数据命令高字节	uint16_t	0x01
Byte8	0xXX	下电保存数据命令低字节		
Byte9	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte10	0xXX	校验码高字节		
下电保存数据回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	下电保存数据寄存器高地址	uint16_t	0x0003
Byte3	0xXX	下电保存数据寄存器低地址		
Byte4	0xXX	下电保存数据寄存器数量高字节	uint16_t	0x0001
Byte5	0xXX	下电保存数据寄存器数量高字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：下电前保存数据				
请求	01 10 00 03 00 01 02 00 01 67 A3			
回复	01 10 00 03 00 01 F1 C9			

5.3.1.5 MOTOR_SPEED 电机转速设置

说明： 本灵巧手产品具有 17 个自由度，每个自由度对应一个电机，支持对单个或多个电机同时进行最大转速限制，在整个电机运行过程中，电机的最大运行转速不超过转速限制阈值，每个电机转速的可调整范围为 0~100（无单位量纲），17 个电机对应的转速设置寄存器 MOTOR_SPEED 的地址分别为 0x0005~0x0015；将相应的 MOTOR_SPEED 寄存器参数进行修改，即可限制电机在运行过程中的最大运行转速。				
单个或多个电机转速设定请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	速度控制寄存器地址高字节	uint16_t	0x0005~0x0015
Byte3	0xXX	速度控制寄存器地址低字节		
Byte4	0xXX	速度控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	速度控制寄存器数量低字节		
Byte6	2* 寄存器数量 N	字节数	uint8_t	0x02~0x22
Byte7	0xXX	电机速度高字节	uint16_t	0x0000~0x0064
Byte8	0xXX	电机速度低字节		
· ·· · · · ·	0xXX	电机速度高字节	uint16_t	0x0000~0x0064
· ·· · · · ·	0xXX	电机速度低字节		
Byte(8 + 2*(N-1)+1)	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte (8 + 2*(N-1)+2)	0xXX	校验码高字节		
单个或多个电机转速设定回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	速度控制寄存器地址高字节	uint16_t	0x0005~0x0015
Byte3	0xXX	速度控制寄存器地址低字节		
Byte4	0xXX	速度控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	速度控制寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：限制所有电机转速不超过最大转速的 50%				
请求	01 10 00 05 00 11 22 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 A4 F0			
回复	01 10 00 05 00 11 10 04			

5.3.1.6 MOTOR_CURRENT 电机运行电流设置

说明： 本灵巧手产品具有 17 个自由度，每个自由度对应一个电机，支持对单个或多个电机同时进行最大运行电流限制，确保电机在运行过程中，电机的最大运行电流不超过电流限制阈值，每个电机最大运行电流的可调整范围为 0~100（无单位量纲）；17 个电机对应的电流设置寄存器 MOTOR_CURRENT 的地址分别为 0x0016~0x0026；对相应的 MOTOR_CURRENT 电流控制寄存器参数进行修改，即可限制电机在运行过程中的最大运行电流。				
电机运行电流设置请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	电流控制寄存器地址高字节	uint16_t	0x0016~0x0026
Byte3	0xXX	电流控制寄存器地址低字节		
Byte4	0xXX	电流控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	电流控制寄存器数量低字节		
Byte6	2*寄存器数量 N	字节数	uint8_t	0x02~0x22
Byte7	0xXX	电机电流高字节	uint16_t	0x0000~0xFFFF
Byte8	0xXX	电机电流低字节		
· · · · · ·	0xXX	电机电流高字节	uint16_t	0x0000~0xFFFF
· · · · · ·	0xXX	电机电流低字节		
Byte(8 + 2*(N-1)+1)	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte (8 + 2*(N-1)+2)	0xXX	校验码高字节		
电机运行电流设置回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	电流控制寄存器地址高字节	uint16_t	0x0016~0x0026
Byte3	0xXX	电流控制寄存器地址低字节		
Byte4	0xXX	电流控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	电流控制寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：限制所有电机电流不超过最大电流的 50%				
请求	01 10 00 16 00 11 22 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 00 32 15 B3			
回复	01 10 00 16 00 11 E1 C1			

5.3.1.7 MOTOR_STOP 电机停机控制

说明： 本灵巧手产品具有 17 个自由度，每个自由度对应一个电机，支持对单个或多个电机同时进行停机控制，17 个电机对应的停机控制寄存器 MOTOR_STOP 的地址分别为 0x0027~0x0037，当需要控制正在运行的电机停机时，往对应的电机停机控制寄存器写 1 即可。				
单个或多个电机停机控制请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	电机停机控制寄存器地址高字节	uint16_t	0x0027~0x0037
Byte3	0xXX	电机停机控制寄存器地址低字节		
Byte4	0xXX	电机停机控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	电机停机控制寄存器数量低字节		
Byte6	2* 寄存器数量 N	字节数	uint8_t	0x02~0x22
Byte7	0xXX	电机停机命令高字节	uint16_t	0x0000~0xFFFF
Byte8	0xXX	电机停机命令低字节		
· · · · · ·	0xXX	电机停机命令高字节	uint16_t	0x0000~0xFFFF
· · · · · ·	0xXX	电机停机命令低字节		
Byte(8 + 2*(N-1)+1)	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte (8 + 2*(N-1)+2)	0xXX	校验码高字节		
单个或多个电机停机控制回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	电机停机控制寄存器地址高字节	uint16_t	0x0027~0x0037
Byte3	0xXX	电机停机控制寄存器地址低字节		
Byte4	0xXX	电机停机控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	电机停机控制寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：控制所有电机同时停机				
请求	01 10 00 27 00 11 22 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 59 64			
回复	01 10 00 27 00 11 11 B0			

5.3.1.8 MOTOP_ABS_POS 电机绝对位置控制

说明： 本灵巧手产品具有 17 个自由度，每个自由度对应一个电机，支持单个或多个电机的绝对位置同时控制。电机在经过标定后得到整个电机最大运动量程范围为 0~1000（无单位量纲），要将电机位置控制在整个运动量程的具体位置，只需要修改相应电机的绝对位置控制寄存器 MOTOP_ABS_POS 的数值即可。				
单个或多个电机绝对位置控制请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	绝对位置控制寄存器地址高字节	uint16_t	0x0038~0x0048
Byte3	0xXX	绝对位置控制寄存器地址低字节		
Byte4	0xXX	绝对位置控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	绝对位置控制寄存器数量低字节		
Byte6	2* 寄存器数量 N	字节数	uint8_t	0x02~0x22
Byte7	0xXX	电机绝对位置高字节	uint16_t	0x0001~0x03E8
Byte8	0xXX	电机绝对位置低字节		
· · · · · ·	0xXX	电机绝对位置高字节	uint16_t	0x0001~0x03E8
· · · · · ·	0xXX	电机绝对位置低字节		
Byte(8 + 2*(N-1)+1)	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte (8 + 2*(N-1)+2)	0xXX	校验码高字节		
单个或多个电机绝对位置控制回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	绝对位置控制寄存器地址高字节	uint16_t	0x0038~0x0048
Byte3	0xXX	绝对位置控制寄存器地址低字节		
Byte4	0xXX	绝对位置控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	绝对位置控制寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：控制 15#电机运行的绝对位置在 487 的位置				
请求	01 10 00 46 00 01 02 01 F4 A8 E1			
回复	01 10 00 46 00 01 E0 1C			

5.3.1.9 MOTOP_INC_POS 电机增量位置控制

说明： 本灵巧手产品具有 17 个自由度，每个自由度对应一个电机，支持单个或多个电机的增量位置同时控制（增量位置控制：即在原本位置上进行位置自增或自减控制）。电机在经过标定后得到整个电机最大运动范围为 0~1000（无单位量纲），要控制电机位置在原本的位置上自增或自减，只需要修改对应电机相应的增量位置控制寄存器 MOTOP_INC_POS 的数值即可（正数代表在原本位置的基础上增加，负数代表在原本的位置上减少）。				
单个或多个电机增量位置控制请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	增量位置控制寄存器地址高字节	uint16_t	0x0049~0x0059
Byte3	0xXX	增量位置控制寄存器地址低字节		
Byte4	0xXX	增量位置控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	增量位置控制寄存器数量低字节		
Byte6	2* 寄存器数量 N	字节数	uint8_t	0x02~0x22
Byte7	0xXX	电机增量位置高字节	int16_t	0xFC18~0x03E8
Byte8	0xXX	电机增量位置低字节		
· · ·	0xXX	电机增量位置高字节	int16_t	0xFC18~0x03E8
· · ·	0xXX	电机增量位置低字节		
Byte(8 + 2*(N-1)+1)	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte(8 + 2*(N-1)+2)	0xXX	校验码高字节		
单个或多个电机增量位置控制回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	增量位置控制寄存器地址高字节	uint16_t	0x0049~0x0059
Byte3	0xXX	增量位置控制寄存器地址低字节		
Byte4	0xXX	增量位置控制寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	增量位置控制寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：控制 15#电机正向运动 500				
请求	01 10 00 57 00 01 02 01 F4 AA 42			
回复	01 10 00 57 00 01 11 D9			

5.3.1.10 MOTOR_ZERO 电机零点标定

说明： 本灵巧手产品具有 17 个自由度，每个自由度对应一个电机，电机首次使用需要对当前电机零点位置进行标定，17 个电机对应的标定寄存器 MOTOR_ZERO 的地址分别为 0x005A~0x006A。如需同时标定单个或多个电机零点，只需在相应的电机标定寄存器 MOTOR_ZERO 写 1 即可。				
单个电机或多个电机零点标定请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据格式	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0xXX	电机零点标定寄存器地址高字节	uint16_t	0x005A~0x006A
Byte3	0xXX	电机零点标定寄存器地址低字节		
Byte4	0xXX	电机零点标定寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0xXX	电机零点标定寄存器数量低字节		
Byte6	2* 寄存器数量 N	字节数	uint8_t	0x02~0x22
Byte7	0xXX	电机零点标定命令高字节	uint16_t	0x0001
Byte8	0xXX	电机零点标定命令低字节		
· · · · · ·	0xXX	电机零点标定命令高字节	uint16_t	0x0001
· · · · · ·	0xXX	电机零点标定命令低字节		
Byte(8 + 2*(N-1)+1)	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte(8 + 2*(N-1)+2)	0xXX	校验码高字节		
单个电机或多个电机零点标定回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0x00	电机零点标定寄存器地址高字节	uint16_t	0x005A~0x006A
Byte3	0x5A	电机零点标定寄存器地址低字节		
Byte4	0x00	电机零点标定寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0x01	电机零点标定寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：对 1#电机进行零点标定				
请求	01 10 00 5A 00 01 02 00 01 6B 6A			
回复	01 10 00 5A 00 01 21 DA			

5.3.1.11 ALL_STEP_MOTOR_ZERO 所有步进电机零点标定

说明： 本灵巧手产品具有两种类型的电机：步进电机与无刷电机。除了通过单个电机标定零点的方式标定电机零点外，还提供一种更简单的方式来标定所有的步进电机。要同时对所有步进电机的零点进行标定，只需要往所有步进电机标定寄存器 ALL_STEP_MOTOR_ZERO（0x006B）写入 1 即可。				
所有步进电机标定请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0x00	全步进电机零点标定寄存器地址高字节	uint16_t	0x006B
Byte3	0x6B	全步进电机零点标定寄存器地址低字节		
Byte4	0x00	全步进电机零点标定寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0x01	全步进电机零点标定寄存器数量低字节		
Byte6	0x02	字节数	uint8_t	0x02
Byte7	0x00	全步进电机零点标定命令高字节	uint16_t	0x0001
Byte8	0x01	全步进电机零点标定命令低字节		
Byte9	0xXX	校验码低字节	uint16_t	0x0001~0x00FF
Byte10	0xXX	校验码高字节		
所有步进电机标定回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0x00	全步进电机零点标定寄存器地址高字节	uint16_t	0x006B
Byte3	0x6B	全步进电机零点标定寄存器地址低字节		
Byte4	0x00	全步进电机零点标定寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0x01	全步进电机零点标定寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：对灵巧手的所有步进电机进行零点标定				
请求	01 10 00 6B 00 01 02 00 01 68 6A			
回复	01 10 00 6B 00 01 70 15			

5.3.1.12 ALL_MOTOR_ZERO 整手全部电机标定

说明： 本灵巧手产品具有两种类型的电机：步进电机与无刷电机。除了通过单个电机标定零点的方式标定电机零点外，还提供一种更简单的方式来标定所有的电机（步进电机与无刷电机）。要同时对所有电机的零点位置进行标定，只需要往所有电机标定寄存器 ALL_MOTOR_ZERO（0x006C）写入 1 即可。				
整手的所有电机零点标定请求格式：地址+功能码+寄存器地址+寄存器数量+字节数+数据+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x01~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0x00	全部电机零点标定寄存器地址高字节	uint16_t	0x006C
Byte3	0x6C	全部机零点标定寄存器地址低字节		
Byte4	0x00	全部电机零点标定寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0x01	全部电机零点标定寄存器数量低字节		
Byte6	0x02	字节数	uint8_t	0x02~0x22
Byte7	0x00	全部电机零点标定命令高字节	uint16_t	0x0001
Byte8	0x01	全部电机零点标定命令低字节		
Byte9	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte10	0xXX	校验码高字节		
整手的所有电机零点标定回复格式：地址+功能码+寄存器地址+寄存器数量+校验码				
字节	数据	含义	数据类型	范围
Byte0	0xXX	灵巧手地址（从站地址）	uint8_t	0x00~0xFE
Byte1	0x10	功能码（写多个保持寄存器）	uint8_t	0x10
Byte2	0x00	全部电机零点标定寄存器地址高字节	uint16_t	0x006C
Byte3	0x6C	全部电机零点标定寄存器地址低字节		
Byte4	0x00	全部电机零点标定寄存器数量高字节	uint16_t	0x0001~0x0011
Byte5	0x01	全部电机零点标定寄存器数量低字节		
Byte6	0xXX	校验码低字节	uint16_t	0x0000~0xFFFF
Byte7	0xXX	校验码高字节		
示例：对灵巧手的所有电机进行零点标定				
请求	01 10 00 6C 00 01 02 00 01 6E FC			
回复	01 10 00 6C 00 01 C1 D4			