

编号： PRJ-03-TS-U-001

页码： 共 21 页

版本： V1.06

电动夹爪

EG2-4XX

用户手册

EG2-4XX-UM

北京因时机器人科技有限公司

2019 年 11 月

## 目 录

1 产品概述.....	1
1.1 产品特性 .....	1
1.2 电气连接 .....	1
1.3.1 引脚定义.....	1
1.3.2 通信方式.....	2
2 常规串口通信协议.....	2
2.1 通信协议概要 .....	2
2.2 指令帧 .....	2
2.3 应答帧 .....	3
2.3.1 应答帧格式.....	3
2.3.2 应答时间.....	4
2.4 指令类型 .....	4
2.4.1 参数固化.....	4
2.4.2 设置 ID .....	5
2.4.3 力控夹取.....	6
2.4.4 力控持续夹取.....	7
2.4.5 松开.....	7
2.4.6 指定开口度.....	8
2.4.7 急停.....	9
2.4.8 设置开口参数.....	9
2.4.9 读取开口参数.....	10
2.4.10 读取当前开口度.....	11
2.4.11 读取夹爪运行状态.....	11
2.4.12 故障清除.....	13
2.4.13 兼容灵巧手指令.....	13
3 Modbus RTU 通讯协议 .....	14
4 CAN 总线协议 .....	18

# 电动夹爪 EG2-4XX

## 用户手册

### 1 产品概述

#### 1.1 产品特性

EG2 系列夹爪是一款利用小体积大扭矩直线伺服驱动器设计生产的电动夹爪，以下简称为“夹爪”。该夹爪内部集成了 1 个直线伺服驱动器，用户接口采用 RS232/RS485 通信接口，支持 Modbus RTU 协议，内置灵敏的压力传感器，通过设置不同的压力阈值方便用户进行不同硬度物体的夹取，简洁高效的接口控制指令可使用户快速实现对夹爪的操控，优质的性能使该夹爪应用于服务机器人、教学教具等领域。夹爪的特点如下所示：

- 夹持力：EG2-4BX 可实现 1.5Kg 的夹持力，EG2-4CX 可实现 2Kg 的夹持力；
- 供电电压：DC6V~8.4V 宽电压范围供电，建议供电 8V；
- 重复定位精度：±0.5mm；
- 最大开口度：70mm；
- 通信接口：EG2-4X1 型号采用 RS232 串口（115200bps、8 数据位、1 停止位、无奇偶校验）；EG2-4X2 型号采用 RS485 串口，支持 Modbus RTU 协议；EG2-4X0 型号采用 CAN 总线接口。

#### 1.2 电气连接

##### 1.3.1 引脚定义

夹爪的电器接口如图 2 所示，接口是标准的微型航空连接器 XS6-5P，端子定义见下表。

	RS232 接口	RS485 接口	CAN 总线接口
1	地 GND	地 GND	地 GND
2	电源 VCC	电源 VCC	电源 VCC
3	RX（夹爪接收端）	A+	CANH
4	TX（夹爪发送端）	B-	CANL
5	GND 端子	GND 端子	GND 端子

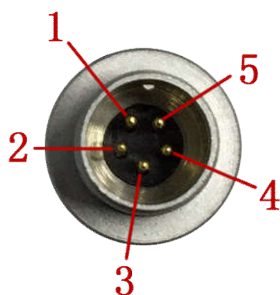


图 2 夹爪电器接口示意图

### 1.3.2 通信方式

EG2-4X1 使用 RS232 标准接口，经过一定的转换电路可以实现 254 只夹爪并接在同一条总线上，通过全双工的异步串口 RS232 接口统一控制。挂接在同一总线上的夹爪，需配置为不同的 ID 号进行通信控制。

夹爪的通信指令集对产品用户开放，可通过异步串行 RS232 接口与用户的上位机（控制器或通用计算机）连接。用户可对夹爪进行参数配置以及夹取控制等。

EG2-4X2 使用 RS485 标准接口，可以实现 254 只夹爪并接在同一条总线上控制。挂接在同一总线上的夹爪，需配置为不同的 ID 号进行通信控制。

## 2 常规串口通信协议

### 2.1 通信协议概要

主控单元与夹爪之间采用问答方式通信，主控单元主动发送指令帧，夹爪收到指令帧，解析并执行后返回应答帧。

同一控制网络中允许一个主控单元同时连接控制多个夹爪，因此每个夹爪需用户分配不同的 ID 号作为唯一标识（夹爪出厂时默认的 ID 号为 1）。主控单元发出的指令帧数据体中包括有 ID 号信息，只有与之匹配的 ID 号对应的夹爪才能完整接收指令帧信息，并在执行指令后返回相应的应答帧。

通信方式为 RS232 异步串口，每个指令帧以字节为最小单位，单一字节由 1 位起始位、8 位数据位以及 1 位停止位组成，无奇偶校验，共 10 比特。

### 2.2 指令帧

基础指令帧格式：

帧头 (2 Bytes)		ID号 (1 Byte)	数据体长度 (1 Byte)	数据体1/2指令号 (1 Byte)	数据体2/2数据 (Len-1 Bytes)	校验和 (1 Byte)
0xEB	0x90	ID	Len	CMD	Data	Check_Sum

帧头：连续收到0xEB和0x90，表示有指令帧到达。

ID号：每个夹爪都有一个ID号。ID号范围为1~254，转换为十六进制为0x01~0xFE。广播ID号255（0xFF），若控制器发出的ID号为255（0xFF），所有的夹爪均接收指令帧，但都不返回应答信息。

数据体长度：等于待发送的数据段长度，包括指令号与数据，即“Len”，总指令帧的长度为“Len+5”。

数据体1/2——指令号：指定该指令帧的类型。

CMD\_MC\_PARA\_SAVE（0x01）：参数保存到内部闪存，掉电不丢失

CMD\_MC\_PARA\_ID\_SET（0x04）：设置夹爪ID

CMD\_MC\_MOVE\_CATCH\_XG（0x10）：以设置的速度和力控阈值去夹取

CMD\_MC\_MOVE\_CATCH2\_XG（0x18）：以设置的速度和力控阈值持续夹取

CMD\_MC\_MOVE\_RELEASE（0x11）：以设置的速度松开

CMD\_MC\_SEEKPOS（0x54）：指定夹爪开口度

CMD\_MC\_MOVE\_STOPHERE（0x16）：急停

CMD\_MC\_SET\_EG\_PARA（0x12）：设置夹爪开口的最大最小值

MD\_MC\_READ\_EG\_PARA（0x13）：读取夹爪开口的最大最小值

CMD\_MC\_READ\_ACTPOS（0xD9）：读取夹爪开口度

CMD\_MC\_READ\_EG\_STATE（0x14）：读取当前夹爪的状态

CMD\_MC\_READ\_EG\_RUNSTATE（0x41）：读取当前夹爪运行状态

数据体2/2——数据：随指令帧一起发送的数据内容。其中，读指令时该数据体无字节；写指令时即为需要写入的内容。

校验和：校验和 Check\_Sum，定义为校验和之前的除帧头两字节外其余所有数据累加和的低字节。

## 2.3 应答帧

### 2.3.1 应答帧格式

应答帧格式：

帧头 (2 Bytes)		ID号 (1 Byte)	数据体长度 (1 Byte)	数据体1/2指令号 (1 Byte)	数据体2/2数据 (Len-1 Bytes)	校验和 (1 Byte)
0xEE	0x16	ID	Len	CMD	Data	Check_Sum

帧头：连续收到0xEE和0x16，表示有应答帧到达。

数据体长度：指令号（1 Byte）和数据（Len-1 Bytes），应答帧总长度“Len+5”。

ID号：每个夹爪都有一个ID号。ID号范围为1~254，转换为十六进制为0x01~0xFE。广播ID号255（0xFF），若控制器发出的ID号为255（0xFF），所有的夹爪均接收指令帧，但都不返回应答信息。

数据体1/2——指令号：指定该应答帧对应的指令帧的类型。

数据体2/2——数据：随应答帧一起发送的数据内容。其中，读指令时为主控单元需要读取的数据内容；写指令时仅一个字节，0x01表示写成功，0x55表示写失败。

校验和：校验和Check\_Sum，定义为校验和之前的除帧头两字节外其余所有数据累加和的低字节。

### 2.3.2 应答时间

夹爪的应答帧的格式对应的应答时间最快为165us，参数固化与恢复出厂的应答时间约为850ms（包括Flash的擦除与烧写），建议使用过程中控制指令间隔5ms以上，参数固化需要1s以上。

## 2.4 指令类型

指令帧的类型如下表说明：

指令类型	功能描述	数值 Hex	数据内容长度 Bytes
CMD_MC_PARA_SAVE	参数保存到内部闪存，掉电不丢失	0x01	0
CMD_MC_PARA_ID_SET	设置夹爪ID	0x04	1
CMD_MC_MOVE_CATCH_XG	以设置的速度和力控阈值去夹取	0x10	3
CMD_MC_MOVE_CATCH2_XG	以设置的速度和力控阈值持续夹取	0x18	3
CMD_MC_MOVE_RELEASE	以设置的速度松开	0x11	1
CMD_MC_MOVE_STOPHERE	急停	0x16	0
CMD_MC_SEEKPOS	指定夹爪开口度	0x54	2
CMD_MC_SET_EG_PARA	设置夹爪开口的最大最小值	0x12	4
CMD_MC_READ_EG_PARA	读取夹爪开口的最大最小值	0x13	0
CMD_MC_READ_ACTPOS	读取夹爪开口度	0xD9	0
CMD_MC_READ_EG_STATE	读取当前夹爪的开口大小以及压力传感器的当前值和设置的阈值	0x14	0
CMD_MC_READ_EG_RUNSTATE	读取夹爪运行状态	0x41	0

### 2.4.1 参数固化

功能：主控单元将当前夹爪使用的开口最大最小值参数保存到内部闪存，掉电不丢失。

指令帧长度：6Bytes

指令号：0x01（CMD\_MC\_PARA\_SAVE）

数据内容：无。

例如：主控单元将 ID 为 1 的夹爪的参数固化，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (0 Bytes)	校验和 B6 (1 Byte)
0xEB	0x90	0x01	0x01	0x01		0x03

发出的指令帧为“EB 90 01 01 01 03”，其中“EB 90”为帧头、“01”为 ID 号、“01”为数据体长度、“01”为 CMD\_MC\_PARA\_SAVE 参数固化指令、“03”为校验和，是指令帧除帧头外其余字节的累加和的低字节  $((B2+B3+\dots+B4) \&0xFF)$ 。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (1 Byte)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	0x 02	0x 01	0x 01	0x 05

具体收到的应答帧信息为“EE 16 01 02 01 01 05”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“01”为指令类型、“01”为指令成功接收标志（若为 55 则表示异常，需要重新发送）、“05”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节  $((B2+B3+\dots+B5) \&0xFF)$ 。

#### 2.4.2 设置 ID

功能：主控单元设置夹爪的 ID 号。

指令帧长度：7Bytes

指令号：0x04（CMD\_MC\_PARA\_ID\_SET）

数据内容：新的 ID 号。

例如：主控单元设置 ID 为 1 的夹爪的新 ID 号为 3，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (1 Byte)	校验和 B6 (1 Byte)
0xEB	0x90	0x01	0x02	0x04	0x03	0x0A

发出的指令帧为“EB 90 01 02 04 03 0A”，其中“EB 90”为帧头、“01”为 ID 号、“02”为数据体长度、“04”为 CMD\_MC\_PARA\_ID\_SET 设置 ID 号指令、“03”为新的 ID 号、“0A”为校验和，是指令帧除帧头外其余字节的累加和的低字节  $((B2+B3+\dots+B5) \&0xFF)$ 。当该指令帧发送给夹爪后，返回的应答帧如下表所示。



帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (12 Bytes)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	0x02	0x04	0x01	0x08

具体收到的应答帧信息为“ EE 16 01 02 04 01 08”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“04”为指令类型、“01”为指令成功接收标志（若为 55 则表示异常，需要重新发送）、“08”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节  $((B2+B3+\cdots+B5) \& 0xFF)$ 。

2.4.3 力控夹取

功能：主控单元设置夹爪以输入的速度和力控阈值去夹取，主控单元设置夹爪以输入的速度和力控阈值去夹取，当夹持力超过设定的力控阈值后，夹爪停止运动。

指令帧长度：9Bytes

指令号：0x10（CMD\_MC\_MOVE\_CATCH\_XG）

数据内容：B5B6 为速度，最大 1000，最小 1，无单位量纲；B7B8 为力控阈值，按照低字节先高字节后的顺序，从 50 到 1000，无单位量纲，例如力控阈值设置为 300g 时，夹爪将以 300g 的力夹取。

例如：主控单元设置 ID 为 1 的夹爪以速度 500、夹取阈值为 100g 的参数进行夹取，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5-B8 (4 Bytes)	校验和 B9 (1 Byte)
0xEB	0x90	0x01	0x05	0x10	0xF4 0x01 0x64 0x00	0x6F

发出的指令帧为“ EB 90 01 05 10 F4 01 64 00 6F”，其中“EB 90”为帧头、“01”为 ID 号、“04”为数据体长度、“10”为 CMD\_MC\_MOVE\_CATCH\_XG 力控夹取指令、“01F4”为速度 500，无量纲、“0064”为力控阈值 100g、“6F”为校验和，是指令帧除帧头外其余字节的累加和的低字节  $((B2+B3+\cdots+B8) \& 0xFF)$ 。当该指令帧发送给夹爪后，夹爪进行夹取，当夹持力超过 100g 后停止运动，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (1 Byte)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	0x02	0x10	0x01	0x14

具体收到的应答帧信息为“ EE 16 01 02 10 01 14”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“10”为指令类型、“01”为指令识别成功标志、“14”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节  $((B2+B3+\cdots+B5) \& 0xFF)$ 。



2.4.4 力控持续夹取

功能：主控单元设置夹爪以输入的速度和力控阈值去夹取，当夹持力超过设定的力控阈值后，夹爪停止运动；当夹爪停止运动后，如果检测到夹持力小于力控阈值时，夹爪会继续夹取直到夹持力超过设定的力控阈值。

指令帧长度：9Bytes

指令号：0x18（CMD\_MC\_MOVE\_CATCH2\_XG）

数据内容：B5B6 为速度，最大 1000，最小 1，无单位量纲；B7B8 为力控阈值，按照低字节先高字节后的顺序，从 50 到 1000，无单位量纲，例如力控阈值设置为 300g 时，夹爪将以 300g 的力夹取。

例如：主控单元设置 ID 为 1 的夹爪以速度 500、夹取阈值为 100g 的参数进行夹取，需要发送的指令帧如下表所示

帧头		ID 号	数据体长	指令号	数据内容	校验和
B0-B1		B2	度	B4	B5-B8	B9
(2 Bytes)		(1 Byte)	B3 (1 Byte)	(1 Byte)	(4 Bytes)	(1 Byte)
0xEB	0x90	0x01	0x05	0x18	0xF4 0x01 0x64 0x00	0x77

发出的指令帧为“EB 90 01 05 18 F4 01 64 00 77”，其中“EB 90”为帧头、“01”为 ID 号、“0x04”为数据体长度、“0x18”为力控持续夹取指令、“01F4”为速度 500，无量纲、“0064”为力控阈值 100g、“77”为校验和，是指令帧除帧头外其余字节的累加和的低字节（(B2+B3+...+B8) & 0xFF）。当该指令帧发送给夹爪后，夹爪进行夹取，当夹持力超过 100g 后停止运动，返回的应答帧如下表所示。

帧头		ID 号	数据体长度	指令号	数据内容	校验和
B0-B1		B2	B3	B4	B5	B6
(2 Bytes)		(1 Byte)	(1 Byte)	(1 Byte)	(1 Byte)	(1 Byte)
0xEE	0x16	0x01	0x02	0x18	0x01	0x1C

具体收到的应答帧信息为“EE 16 01 02 18 01 1C”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“18”为指令类型、“01”为指令识别成功标志、“1C”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节（(B2+B3+...+B5) & 0xFF）。

2.4.5 松开

功能：主控单元设置夹爪以输入的速度参数将夹爪松开到最大开口位置。

指令帧长度：7Bytes

指令号：0x11（CMD\_MC\_MOVE\_RELEASE）

数据内容：B5B6 为速度，最大 1000，最小 1，无单位量纲。

例如：主控单元设置 ID 为 1 的夹爪以速度 500 松开夹爪到最大开口位置，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5B6 (1 Byte)	校验和 B7 (1 Byte)
0xEB	0x90	0x01	0x03	0x11	0xF4 0x01	0x0A

发出的指令帧为“EB 90 01 03 11 F4 01 0A”，其中“EB 90”为帧头、“01”为 ID 号、“03”为数据体长度、“11”为 CMD\_MC\_MOVE\_RELEASE 松开夹爪指令、“01F4”为速度 500，无量纲、“0A”为校验和，是指令帧除帧头外其余字节的累加和的低字节  $((B2+B3+\dots+B6) \& 0xFF)$ 。当该指令帧发送给夹爪后，夹爪以速度 500 运动到最大开口位置，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (12 Bytes)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	0x02	0x11	0x01	0x15

具体收到的应答帧信息为“EE 16 01 02 11 01 15”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“11”为指令类型、“01”为指令识别成功标志、“15”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节  $((B2+B3+\dots+B5) \& 0xFF)$ 。

#### 2.4.6 指定开口度

功能：主控单元指定夹爪的目标开口度，夹爪接受到这条指令后，如果当前开口度小于设定开口度，以设定速度松开直到开口度达到目标开口度后停止运动；如果当前开口度大于设定开口度，以设定速度和力控阈值去夹取，当夹持力超过设定的力控阈值后，或者开口度达到目标开口度后停止运动。

指令帧长度：7Bytes

指令号：0x54 (CMD\_MC\_SEEKPOS)

数据内容：B5B6 为开口度，最大 1000 (70mm)，最小 1 (0mm)，无单位量纲。

例如：主控单元设置 ID 为 1 的夹爪运动到 500 (35mm) 开口位置，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5B6 (1 Byte)	校验和 B7 (1 Byte)
0xEB	0x90	0x01	0x03	0x54	0xF4 0x01	0x4D

发出的指令帧为“EB 90 01 03 54 F4 01 4D”，其中“EB 90”为帧头、“01”为 ID 号、“03”为数据体长度、“54”为 CMD\_MC\_MOVE\_RELEASE 松开夹爪指令、“01F4”为速

度 500，无量纲、“4D”为校验和，是指令帧除帧头外其余字节的累加和的低字节（ $(B2+B3+\dots+B6) \&0xFF$ ）。当该指令帧发送给夹爪后，夹爪以速度 500 运动到最大开口位置，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (12 Bytes)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	2 (0x02)	0x54	0x01	0x58

具体收到的应答帧信息为“EE 16 01 02 54 01 58”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“54”为指令类型、“01”为指令识别成功标志、“58”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节（ $(B2+B3+\dots+B5) \&0xFF$ ）。

#### 2.4.7 急停

功能：主控单元通过该指令让运动中的夹爪停止运动，保持在该位置不动。

指令帧长度：6Bytes

指令号：0x16 (CMD\_MC\_MOVE\_STOPHERE)

数据内容：无。

例如：主控单元设置 ID 为 1 的夹爪进入急停状态，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (0 Byte)	校验和 B6 (1 Byte)
0xEB	0x90	0x01	1 (0x01)	0x16	无	0x18

发出的指令帧为“EB 90 01 01 16 18”，其中“EB 90”为帧头、“01”为 ID 号、“01”为数据体长度、“16”为 CMD\_MC\_MOVE\_STOPHERE 急停指令、“0A”为校验和，是指令帧除帧头外其余字节的累加和的低字节（ $(B2+B3+\dots+B4) \&0xFF$ ）。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (12 Bytes)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	2 (0x02)	0x16	0x01	0x1A

具体收到的应答帧信息为“EE 16 01 02 16 01 1A”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“16”为指令类型、“01”为指令成功接收标志（若为 55 则表示异常，需要重新发送）、“1A”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节（ $(B2+B3+\dots+B5) \&0xFF$ ）。

#### 2.4.8 设置开口参数

功能：主控单元设置夹爪的最大和最小开口参数。

指令帧长度：10Bytes

指令号：0x12（CMD\_MC\_SET\_EG\_PARA）

数据内容：B5B6 为开口最大值，数据范围 0—1000，无单位量纲，B7B8 为开口最小值，数据范围 0—1000，无单位量纲。

例如：主控单元设置 ID 为 1 的夹爪最大开口 1000，最小开口 112，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5-B8 (4 Bytes)	校验和 B9 (1 Byte)
0xEB	0x90	0x01	5 (0x05)	0x12	0xE8 0x03 0x70 0x00	0x73

发出的指令帧为“ EB 90 01 05 12 E8 03 70 00 73”，其中“EB 90”为帧头、“01”为 ID 号、“05”为数据体长度、“12”为 CMD\_MC\_SET\_EG\_PARA 设置夹爪开口参数指令、“73”为校验和，是指令帧除帧头外其余字节的累加和的低字节 $((B2+B3+\cdots+B8)\&0xFF)$ 。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (12 Bytes)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	2 (0x02)	0x12	0x01	0x16

具体收到的应答帧信息为“ EE 16 01 02 12 01 16”，其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“12”为指令类型、“01”为指令识别成功标志（若设置数据有问题则返回 0x55）、“16”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节 $((B2+B3+\cdots+B5)\&0xFF)$ 。

2.4.9 读取开口参数

功能：主控单元读取夹爪的开口参数。

指令帧长度：6Bytes

指令号：0x13（MD\_MC\_READ\_EG\_PARA）

数据内容：无。

例如：主控单元读取 ID 为 1 的夹爪的开口参数，包括最大和最小开口，读出的最大开口数据和最小开口数据范围一样，都是 0—1000，无单位量纲，需要发送的指令帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 — (0 Bytes)	校验和 B5 (1 Byte)
0xEB	0x90	0x01	1 (0x01)	0x13		0x15

发出的指令帧为“ EB 90 01 01 13 15”，其中“EB 90”为帧头、“01”为 ID 号、

“01”为数据体长度、“13”为 MD\_MC\_READ\_EG\_PARA 读取夹爪开口参数指令、“15”为校验和，是指令帧除帧头外其余字节的累加和的低字节  $((B2+B3+\dots+B4) \& 0xFF)$ 。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5-B8 (4 Bytes)	校验和 B15 (1 Byte)
0xEE	0x16	0x01	5 (0x05)	0x13	0xE8 0x03 0x70 0x00	0x74

具体收到的应答帧信息为“EE 16 01 05 13 E8 03 70 00 74”，其中“EE 16”为应答帧头、“01”为 ID 号、“05”为数据体长度、“13”为指令类型、“E8--00”为读取到的夹爪开口参数(0x03E8=1000 即最大开口为 1000， 0x0070=112 即最小开口为 112,)、“74”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节  $((B2+B3+\dots+B8) \& 0xFF)$ 。

#### 2.4.10 读取当前开口度

功能：主控单元读取夹爪的当前开口。

指令帧长度：6Bytes

指令号：0xD9 (CMD\_MC\_READ\_ACTPOS)

数据内容：无。

例如：主控单元读取 ID 为 1 的夹爪的当前开口度，读出数据范围是 0—1000，无单位量纲，需要发送的指令帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 -- (0 Bytes)	校验和 B5 (1 Byte)
0xEB	0x90	0x01	0x01	0xD9		0xDB

发出的指令帧为“EB 90 01 01 D9 DB”，其中“EB 90”为帧头、“01”为 ID 号、“01”为数据体长度、“D9”是读取夹爪开口参数指令、“DB”为校验和，是指令帧除帧头外其余字节的累加和的低字节  $((B2+B3+\dots+B4) \& 0xFF)$ 。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5-B6 (4 Bytes)	校验和 B7 (1 Byte)
0xEE	0x16	0x01	0x03	0xD9	0xE8 0x03	0x74

具体收到的应答帧信息为“EE 16 01 03 D9 F1 01 CF”，其中“EE 16”为应答帧头、“01”为 ID 号、“05”为数据体长度、“D9”为指令类型、“F1-01”为读取到的夹爪当前开口度(0x01F1 转换为整型为 497,)、“CF”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节  $((B2+B3+\dots+B7) \& 0xFF)$ 。

#### 2.4.11 读取夹爪运行状态



功能：主控单元读取夹爪运行状态，可得到当前开口大小，夹持力设定值，运行状态，故障码、驱动器温度值，该指令可以周期性的获取到夹爪的运行状态信息，帮助用户快速定位夹爪的状态。

指令帧长度：6Bytes

指令号：0x41（CMD\_MC\_READ\_EG\_RUNSTATE）

数据内容：无。

例如：主控单元读取 ID 为 1 的夹爪当前状态，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 -- (0 Byte)	校验和 B5 (1 Byte)
0xEB	0x90	0x01	1 (0x01)	0x41		0x43

发出的指令帧为“EB 90 01 01 41 43”，其中“EB 90”为帧头、“01”为 ID 号、“01”为数据体长度、“0x41”为 CMD\_MC\_READ\_EG\_RUNSTATE 读取夹爪运行状态指令、“0x43”为校验和，是指令帧除帧头外其余字节的累加和的低字节（(B2+B3+...+B4)&0xFF）。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 B)		ID号 B2 (1B)	数据体 长度 B3 (1B)	指令号 B4 (1B)	数据内容B5-B11					校验和 B12 (1B)
					运行状态码 B5 (1B)	故障码 B6 (1B)	温度 B7 (1B)	开口度 B8-9 (2B)	夹持力设置 B10-11 (2B)	
0xEE	0x16	0x01	8 (0x07)	0x41	0x01	0x00	0x23	0xE9 0x03	0x64 0x00	0xBD

收到的返回帧字节为：EE 16 01 08 41 01 00 23 E8 03 64 00 BD。

其中运行状态码含义如下：

0x01	夹爪张开到最大且空闲
0x02	夹爪闭合到最小且空闲
0x03	夹爪停止且空闲
0x04	夹爪正在闭合
0x05	夹爪正在张开
0x06	夹爪闭合过程中遇到力控停止

故障码每一位含义如下：

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
保留	保留	保留	内部通信	驱动器	过流	过温	堵转

			故障	运行故障	故障	故障	故障
--	--	--	----	------	----	----	----

温度单位是设置度，如 0x23，代表 35 摄氏度。

开口度，0x03E8，代表当前夹爪开口度是 1000，无单位量纲。

夹持力单位是 g，如 0x0064，代表当前设置的夹持力为 100g。

2.4.12 故障清除

功能：对夹爪的故障码中的 bit0、bit2、bit3 和 bit4 产生的故障，可通过故障清除指令来恢复夹爪的正常工作，而 bit1 的过温故障（温度高于 80 摄氏度）只能等待温度降低到（低于 60 摄氏度）后自己恢复正常工作。对于经过故障清除指令后仍然又出现的故障，说明该产品的故障不可被清除，该故障一直存在，需要进行工程师维修处理。

指令帧长度：6Bytes

指令号：0x17（CMD\_MC\_ERROR\_CLR）

数据内容：无。

例如：主控单元设置 ID 为 1 的夹爪进行故障清除，需要发送的指令帧如下表所示

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 (0 Byte)	校验和 B5 (1 Byte)
0xEB	0x90	0x01	1 (0x01)	0x17	无	0x19

其中“EB 90”为帧头、“01”为 ID 号、“01”为数据体长度、“17”为 CMD\_MC\_ERROR\_CLR 零位校准指令、“0x19”为校验和，是指令帧除帧头外其余字节的累加和的低字节（(B2+B3+...+B4) &0xFF）。当该指令帧发送给夹爪后，返回的应答帧如下表所示。

帧头 B0-B1 (2 Bytes)		ID号 B2 (1 Byte)	数据体长度 B3 (1 Byte)	指令号 B4 (1 Byte)	数据内容 B5 (12 Bytes)	校验和 B6 (1 Byte)
0xEE	0x16	0x01	2 (0x02)	0x17	0x01	0x1B

其中“EE 16”为应答帧头、“01”为 ID 号、“02”为数据体长度、“17”为指令类型、“01”为指令成功接收标志（若为 55 则表示异常，需要重新发送）、“0x1B”是最后一个字节，为校验和，是除应答帧头外其余数据的累加和的低字节（(B2+B3+...+B5) &0xFF）。

2.4.13 兼容灵巧手指令

功能：对电动夹爪的控制指令协议同时兼容本公司产品灵巧手的部分控制协议指令，用户可以使用灵巧手的控制指令对夹爪进行操控，方便用户对公司不同产品进行快速的体验与应用。



序号	指令类型	功能描述	数值 Hex	数据内容长度 Bytes
1	CMD_MC_SET_DRVALL_SEEKANGLE_GYH	设置目标归一化位置	0x54	12
3	CMD_MC_SET_DRVALL_SPEED	设置速度	0x51	12
4	CMD_MC_SET_DRVALL_YBP_THRESHOLD	设置抓力阈值	0x52	10
5	CMD_MC_READ_DRVALL_SEEKGYHANGLE	读取目标归一化位置	0xD7	0
7	CMD_MC_READ_DRVALL_CURANGLE_GYH	读取当前归一化位置	0xD9	0
9	CMD_MC_READ_DRVALL_SPEED	读取当前速度	0xD2	0
10	CMD_MC_READ_DRVALL_YBP_THRESHOLD	读取抓力阈值	0xD3	0
13	CMD_MC_READ_HAND_STATUS	读取灵巧手状态信息	0xDC	0

### 3 Modbus RTU 通讯协议

采用 16 位 CRC 反向校验 (CRC16-REV = 0xA001)。

支持的功能码：0x03(读寄存器), 0x06(写入单个寄存器), 0x10(写入多个寄存器)。

寄存器定义如下：

地址	名称	范围	说明		
1001	CMD_SAVE	0/1	1: 保存参数 0: 无	读写	立即生效
1002	CMD_SETDEFAULT	0/1	1: 将参数还原为默认参数 0:	读写	立即生效
1003	CMD_ID	0-254	夹爪的设备地址 (ID 号)	读写	立即生效
1004	CMD_BAUD	0, 1, 2	波特率设置 0: 115200 1: 57600 2: 19200	读写	保存后， 重新上电 生效
1005	CMD_CATCH_MOD	0/1	0: 一次夹取 (开口度到位或者夹到物体停止运动) 1: 持续夹取 (夹到物体停止运动后，如果夹爪受力下降，夹住会继续加持，保持加持力)	读写	立即生效
1006	CMD_STOP	0/1	1: 急停 0:	读写	立即生效
1007	CMD_FAULTACK	0/1	1: 清除故障 0:	读写	立即生效
1010	CMD_OPENLEN_SET	0-1000	夹爪开口度设置，1000 对应最大开口度，0 对应最小开口度，设置此寄存器后夹爪立即动作	读写	立即生效
1011	CMD_SPEED_SET	10-1000	夹爪速度设置，1000 对应对打运行速度，设置此寄存器后夹爪不会动作	读写	立即生效
1012	CMD_FORCE_SET	100-1000	夹爪加持力设置，1000 对应最大加持力，设置此寄存器后夹爪不会	读写	立即生效

			动作		
1016	CMD_MAX_OPENLEN	0-1000	最大开口度	读写	立即生效
1017	CMD_MIN_OPENLEN	0-1000	最小开口度	读写	立即生效
1060	CMD_FORCE_ACT		夹爪实际受力值	只读	
1061	CMD_OPENLEN_ACT	0-1000	夹爪实际开口度	只读	
1062	CMD_CURRENT	0-2000	夹爪实际电流	只读	
1063	CMD_TEMP	0-100	夹爪驱动器温度	只读	
1064	CMD_ERRORCODE		夹爪故障码 bit0: 堵转故障 bit1: 过温故障 bit2: 过流故障 bit3: 运行故障 bit4: 内部通讯故障	只读	
1065	CMD_STATUS		夹爪状态码 1: 夹爪张开到最大且停止 2: 夹爪闭合到最下且停止 3: 夹爪停止 4: 夹爪正在夹取; 5: 夹爪正在张开; 6: 夹爪在夹取过程中夹到物体停止	只读	

举例 1：读取夹爪开口度（1061）、电流（1062）、温度（1063）和故障码（1064）的指令帧如下：

		举例	
		16 进制	10 进制
byte[0]	夹爪 ID	01	1
byte[1]	功能码	03	3
byte[2]	寄存器地址高 8 位	04	1061
byte[3]	寄存器地址低 8 位	25	
byte[4]	寄存器个数高 8 位	00	4
byte[5]	寄存器个数低 8 位	04	
byte[6]	CRC 校验码	86	
byte[7]	CRC 校验码	21	

夹爪回复如下：

		举例	
		16 进制	10 进制
byte[0]	夹爪 ID	01	1
byte[1]	功能码	03	3
byte[2]	字节数	08	8
byte[3]	寄存器（1061）内容高 8 位	00	0
byte[4]	寄存器（1061）内容低 8 位	00	
byte[5]	寄存器（1062）内容高 8 位	FF	-2
byte[6]	寄存器（1062）内容低 8 位	FE	
byte[7]	寄存器（1063）内容高 8 位	00	37
byte[8]	寄存器（1063）内容低 8 位	25	
byte[9]	寄存器（1064）内容高 8 位	00	0
byte[10]	寄存器（1064）内容低 8 位	00	
byte[11]	CRC 校验码	C7	
byte[12]	CRC 校验码	B9	

举例 2：设置夹住开口（1010）为 0 的指令帧如下：

		举例	
		16 进制	10 进制
byte[0]	夹爪 ID	01	1
byte[1]	功能码	06	6
byte[2]	寄存器地址高 8 位	03	1010
byte[3]	寄存器地址低 8 位	F2	
byte[4]	设置值高 8 位	00	0
byte[5]	设置值低 8 位	00	
byte[6]	CRC 校验码	7D	
byte[7]	CRC 校验码	28	

夹爪回复如下：

		举例	
		16 进制	10 进制
byte[0]	夹爪 ID	01	1
byte[1]	功能码	06	6
byte[2]	寄存器地址高 8 位	03	1010
byte[3]	寄存器地址低 8 位	F2	
byte[4]	设置值高 8 位	00	0
byte[5]	设置值低 8 位	00	
byte[6]	CRC 校验码	7D	
byte[7]	CRC 校验码	28	

举例 3：设置夹爪开口度（1010）为 0，速度（1011）为 1000，力度（1012）为 1000 的指令帧如下：

		举例	
		16 进制	10 进制
byte[0]	夹爪 ID	01	1
byte[1]	功能码	10	16
byte[2]	寄存器地址高 8 位	03	1010
byte[3]	寄存器地址低 8 位	F2	
byte[4]	寄存器个数高 8 位	00	3
byte[5]	寄存器个数低 8 位	03	
byte[6]	字节数	06	6
byte[7]	寄存器（1010）要设置的数值高 8 位	00	0
byte[8]	寄存器（1010）要设置的数值低 8 位	00	
byte[9]	寄存器（1011）要设置的数值高 8 位	03	1000
byte[10]	寄存器（1011）要设置的数值低 8 位	E8	
byte[11]	寄存器（1012）要设置的数值高 8 位	03	1000
byte[12]	寄存器（1012）要设置的数值低 8 位	E8	
byte[13]	CRC 校验码	03	
byte[14]	CRC 校验码	CD	

夹爪回复如下：

		举例	
		16 进制	10 进制
byte[0]	夹爪 ID	01	1
byte[1]	功能码	10	16
byte[2]	寄存器地址高 8 位	03	1010
byte[3]	寄存器地址低 8 位	F2	
byte[4]	寄存器个数高 8 位	00	3
byte[5]	寄存器个数低 8 位	03	
byte[6]	CRC 校验码	BF	
byte[7]	CRC 校验码	21	

#### 4 CAN 总线协议

CAN 总线波特率为 500K，采用扩展标示符、数据帧格式。不使用标准标示符及远程帧。其中扩展标示符共 29 位，从低位至高位定义如下：

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
说明	预留				操作类型 00：读寄存器 01：写寄存器		寄存器起始地址												设备ID：1-16382													
例子	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x04FF0001.操作设备的ID为1，寄存器起始地址1020（对应的二进制0b0011111100），操作类型是写寄存器																																

	含义
bit31-28	预留
bit27,26	bit27=0 bit26=0:读寄存器操作标志，读取夹爪寄存器内容 bit27=0 bit26=1:写寄存器操作标志，设置夹爪寄存器内容 bit27=1 bit26=0:手指定位运动指令，控制手指运动到指定位置，设置手指运动速度和受力值。 bit27=1 bit26=1:手指伺服随动指令，控制手指跟随运动设置角度运动。
bit14-25	所读（写）寄存器的起始地址， 范围为 2-2400
bit13-0	设备 ID。夹爪内部有两个设备 ID，分别是可设置的设备 ID（出厂默认 1，可用户修改为 1-16383）和广播 ID（16383）。

## 4.1.1 读寄存器操作

读取夹爪寄存器内容的指令帧和回复帧格式如下：

发送至夹爪					夹爪回复				
ExtId 标识符	bit 31-28	bit 27/26	bit14-25	bit 13-0	ExtId 标识符	bit 31-28	bit 27/26	bit14-25	bit 13-0
	预留	0b00（读）	寄存器地址 +1000	ID		预留	0b00（读）	寄存器地址 +1000	ID
DLC 数据长度	1				DLC 数据长度	N(读取寄存内容的长度, 不能超过 8)			
数据内容	TX[0]	N(预读取寄存内容的长度, 不能超过 8)			数据内容	RX[0]	寄存器 1 值低 8 位		
	TX[1]	无用				RX[1]	寄存器 1 值高 8 位		
	TX[2]	无用				RX[2]	寄存器 2 值低 8 位		
	TX[3]	无用				RX[3]	寄存器 2 值低 8 位		
	TX[4]	无用				RX[4]	寄存器 3 值低 8 位		
	TX[5]	无用				RX[5]	寄存器 3 值低 8 位		
	TX[6]	无用				RX[6]	寄存器 4 值低 8 位		
	TX[7]	无用				RX[7]	寄存器 4 值低 8 位		

例如欲读 ID 为 1 的夹爪当前实际受力值、开口度、电流、温度（寄存器起始地址 1120，长度为 8byte），指令帧和回复帧如下：

发送至夹爪					夹爪回复				
ExtId 标识符	bit 31-28	bit 27\26	bit14-25	bit 13-0	ExtId 标识符	bit 31-28	bit 27\26	bit14-25	bit 13-0
	预留	0b00（读）	寄存器地址 0b010001 100000	0b0000000 0000001		预留	0b00（读）	寄存器地址 0b010001 100000	0b0000000 0000001
DLC 数据长度	1				DLC 数据长度	8(读取寄存内容的长度, 不能超过 8)			
数据内容	TX[0]	8(预读取寄存内容的长度, 不能超过 8)			数据内容	RX[0]	0xF3	当前受力值为 243（0x00F3）	
	TX[1]	无用（填 0 补充）				RX[1]	0x00		
	TX[2]	无用（填 0 补充）				RX[2]	0x03	当前开口度为 1000（0x03E8）	
	TX[3]	无用（填 0 补充）				RX[3]	0xE8		
	TX[4]	无用（填 0 补充）				RX[4]	0x00	当前电流为 0（0x0000）	
	TX[5]	无用（填 0 补充）				RX[5]	0x0		
	TX[6]	无用（填 0 补充）				RX[6]	0x00	当前温度为 34（0x0022）	
	TX[7]	无用（填 0 补充）				RX[7]	0x22		

例如欲读 ID 为 1 的夹爪当前故障码、状态（寄存器起始地址 1128，长度为 4byte），指令帧和回复帧如下：

发送至夹爪					夹爪回复				
ExtId 标识符	bit 31-28	bit 27/26	bit14-25	bit 13-0	ExtId 标识符	bit 31-28	bit 27/26	bit14-25	bit 13-0
	预留	0b00（读）	寄存器地址 0b010001 101000	0b000000 00000001		预留	00（读）	寄存器地址 0b010001 101000	0b000000 00000001
DLC 数据长度	1				DLC 数据长度	4(读取寄存内容的长度, 不超过 8)			
数 据 内 容	TX[0]	4(预读取寄存内容的长度, 不能超过 8)			数 据 内 容	RX[0]	0x00	当前无故障（0x0000）	
	TX[1]	无用（填 0 补充）				RX[1]	0x00		
	TX[2]	无用（填 0 补充）				RX[2]	0x01	当前状态为夹爪张开到最大且停 止（0x0001）	
	TX[3]	无用（填 0 补充）				RX[3]	0x00		
	TX[4]	无用（填 0 补充）				RX[4]	无用（填 0 补充）		
	TX[5]	无用（填 0 补充）				RX[5]			
	TX[6]	无用（填 0 补充）				RX[6]	无用（填 0 补充）		
	TX[7]	无用（填 0 补充）				RX[7]			

#### 4.1.2 写寄存器操作

写入夹爪寄存器内容的指令帧和回复帧格式如下：

发送至夹爪					夹爪回复				
ExtId	bit 31-28	bit 27/26	bit14-25	bit 13-0	ExtId	bit 31-28	bit 27/26	bit14-25	bit 13-0
标识符	预留	0b01（写）	寄存器地址 +1000	ID	标识符	预留	0b01（写）	寄存器地址 +1000	ID
DLC 数据长度	N: 欲写入寄存器内数据的长度				DLC 数据长度	1: 返回的数据长度			
数据内容	TX[0]	数据内容		寄存器 1 低 8 位	数据内容	RX[0]	N（写入寄存器的数据长度值）		
	TX[1]	数据内容		寄存器 1 高 8 位		RX[1]	无用（填 0 补充）		
	TX[2]	数据内容		寄存器 2 低 8 位		RX[2]	无用（填 0 补充）		
	TX[3]	数据内容		寄存器 2 高 8 位		RX[3]	无用（填 0 补充）		
	TX[4]	数据内容		寄存器 3 低 8 位		RX[4]	无用（填 0 补充）		
	TX[5]	数据内容		寄存器 3 高 8 位		RX[5]	无用（填 0 补充）		
	TX[6]	数据内容		寄存器 4 低 8 位		RX[6]	无用（填 0 补充）		
	TX[7]	数据内容		寄存器 4 高 8 位		RX[7]	无用（填 0 补充）		



实例：操作 ID 为 1 的夹爪以 500g 夹持力、500 速度向 0mm 开口度夹持，指令帧和回复帧的格式如下：

发送至夹爪					夹爪回复				
ExtId 标识符	bit 31-28	bit 27/26	bit14-25	bit 13-0	ExtId 标识符	bit 31-28	bit 27/26	bit14-25	bit 13-0
	预留	0b01（写）	寄存器地址： 0b001111 111100	0b000000 00000001		预留	0b01（写）	寄存器地址 0b001111 111100	0b000000 00000001
DLC 数据长度	6：欲写入寄存器内数据的长度				DLC 数据长度	1：返回的数据长度			
数据内容	TX[0]	0x00	目标开口度设置为 0（0x0000）		数据内容	RX[0]	6：写入寄存器的数据长度值		
	TX[1]	0x00				RX[1]	无用（填 0 补充）		
	TX[2]	0xF4	运行速度设置为 500（0x01F4）			RX[2]	无用（填 0 补充）		
	TX[3]	0x01				RX[3]	无用（填 0 补充）		
	TX[4]	0xF4	夹持力设置为 500（0x01F4）			RX[4]	无用（填 0 补充）		
	TX[5]	0x01				RX[5]	无用（填 0 补充）		
	TX[6]	无用（填 0 补充）				RX[6]	无用（填 0 补充）		
	TX[7]	无用（填 0 补充）				RX[7]	无用（填 0 补充）		