

# 以太网通信协议

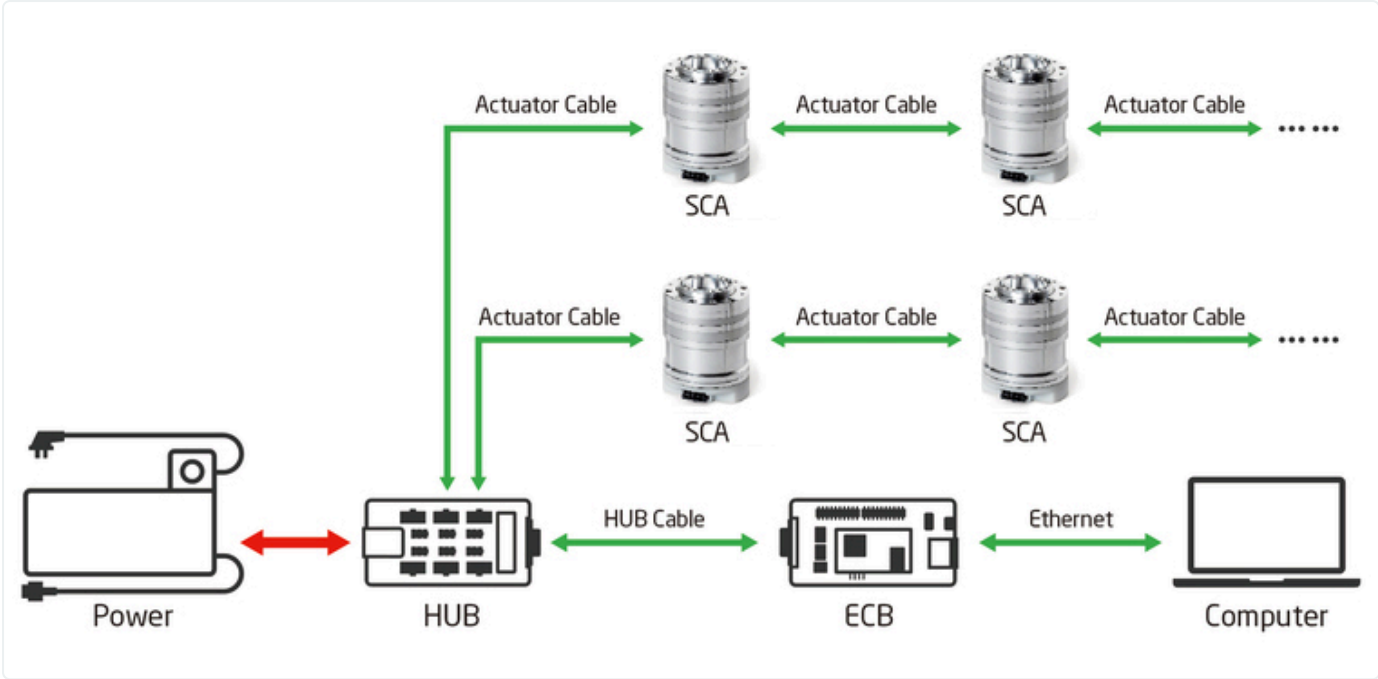
概述
<b>执行器连接</b>
通信方式
通信协议
发送流程
以太网通信协议命令参考
附录A
附录B:模式表
附录C:报警指令表
版本变更记录

## 概述

- [ ↑ ] (pages/#md-page-menu)
- 以太网是目前使用最广泛的局域网技术。由于其简单、成本低、可扩展性强、与IP网能够很好地结合等特点。
  - 本公司产品设计基于UDP协议标准，本文详细介绍了本公司产品的产品以太网通信协议格式，及产品以太网通信结构。

## 执行器连接

[ ↑ ] (pages/#md-page-menu)



(pages/../../img/connect2.png)

按上图连接好设备，再接通电源。

警告：所有线缆拔插严禁带电操作，否则容易损坏设备。

## 通信方式

[ ↑ ] (pages/#md-page-menu)

本协议分为物理层，数据链路层和用户层

- 物理层

电气接口标准为Ethernet，IEEE 802.3-2002 标准、100M 全双工通讯。

- 数据链路层

数据链路层规定了数据帧的具体格式，上位机与从机采用相同的数据格式，数据中的每个字段采用16位进制表示。

- 用户层

用户层定义了从机与主机通讯的命令接口。

## 通信协议

[ ↑ ] (pages/#md-page-menu)

### 通信协议格式

帧头	设备地址	指令符	数据位数	数据内容	CRC 校验码	帧尾
1 字节	1 字节	1 字节	2 字节	N 字节	2 字节	1 字节

#### 帧头

- 固定字节 0xEE ，标识一个传输帧的开始。

设备地址

- 一个字节标识要与之通讯的设备地址，0x01~0xFF 可用。0x00 为广播地址。

指令符

- 一个字节标识主机与从机进行的具体操作，取值范围 0x00~0xFF 。详见用户层具体描述。

数据位数

- 两个字节标识要通讯的具体数据的个数，范围 0x0000~0xFFFF ，超出范围的不作处理。

数据内容

- 某条指令的具体数据内容，其长度与之前的数据位数位保持一致。有些 指令不含具 体数据，其数 据位数位也应为 0。

CRC校验码

- 两个字节标识对通讯中数据内容的CRC16校验结果，若命令中无数据，则无 CRC校验位,CRC计算方法请参考CRC校验码计算方法

帧尾

- 固定字节 0xED ，标识一个传输帧的结束。

以太网通信协议命令应用举例

协议示例

示例1.读命令

读取执行器ID为0x01的当前速度值					
帧头	设备地址	指令符	数据长度（2字节）	参数内容	帧尾
0xEE	0x01	0x05	0x00 0x00	无	0xED

帧头 ：0xEE = 协议头

设备地址：0x01 = 读取的对象ID

指 令 符：0x05 = 读取的当前速度指令

数据长度：0x0 = 数据长度

参数内容：无 = 发送的参数内容

帧尾 ：0xED = 协议尾

应答命令						
帧头	设备地址	指令符	数据长度（2字节）	参数内容	CRC校验	帧尾
0xEE	0x01	0x05	0x00 0x04	4字节数据	2字节数据	0xED

帧头：0xEE = 协议头

设备地址：0x01 = 读取的对象ID

指令符：0x05 = 读取的当前速度指令

数据长度：0x04 = 数据长度

参数内容：四字节速度数据 = 发送的参数内容

校验位：两字节数据 = 校验位

帧尾：0xED = 协议尾

Note: 参数内容data[0~3]高位在前，低位在后。为\_IQ24格式。\_IQ(-1.0)~\_IQ(1.0)代表反转速度满量程和正转速度满量程。满量程为6000RPM。若data=\_IQ(0.5)。则为0.5\*6000=3000RPM。

## 示例2.写命令

### 设置执行器ID为0x01的电流环的比例P（设置值为5）

帧头	设备地址	指令符	数据长度	参数内容	CRC校验	帧尾
0xEE	0x01	0x0A	0x04	0x05 0x00 0x00 0x00	2字节数据	0xED

帧头：0xEE = 协议头

设备地址：0x01 = 设置对象ID

指令符：0x0A = 设置位置

数据长度：0x04 = 数据长度

参数内容：0x05 0x00 0x00 0x00 = 发送的参数内容

校验位：两字节数据 = 校验位

帧尾：0xED = 协议尾

Note: 参数内容data[0~3]高位在前，低位在后。为\_IQ24格式。\_IQ(-128.0)~\_IQ(127.999999940)代表反向位置值满量程和正向位置值满量程。若data=\_IQ(5.0),则比例为5。

### 应答命令

帧头	设备地址	指令符	数据长度	参数内容	帧尾
0xEE	0x01	0x0E	0x01	0x01	0xED

帧头：0xEE = 协议头

返回设备地址：0x01 = 应答对象ID

指令符：0x0A = 应答设置位置（与发送对应）

数据长度：0x01 = 应答的数据长度

参数内容：0x01 = 应答的参数内容（0x01表示写入成功）

帧尾：0xED = 协议尾

## 发送流程

[↑](pages/#md-page-menu)

## 通信配置

以太网通信方式采用udp通信，ECB固定了IP地址，为192.168.1.30(默认)，可以通过相关协议修改该IP地址。PC端需修改网络配置，IP地址修改为192.168.1.xxx，xxx为需大于100，以避免和ECB地址冲突，子网掩码255.255.255.0，默认网关192.168.1.1，配置成功并且连接成功通电以后，可以ping到ECB的IP地址，这样就表示连接成功，通信端口为2000，可以通过该端口与执行器通信。

## 与ECB握手

使用以太网通信需要先与ECB握手，待ECB成功回复后，发送查询执行器地址指令，成功返回执行器地址指令，才能开始发送指令控制和调节执行器。

发送指令：

EE 00 44 00 00 ED

通信成功会收到返回指令：

EE 00 44 00 01 01 ED

如果不知道ECB的IP地址，可通过广播方式发送该协议(即向192.168.1.255发送该协议)，确认ECB的IP地址后，即可通过该IP地址发送其他指令。

## 查询执行器

发送指令：

EE 00 02 00 00 ED

通信成功会返回所有执行器的ID和序列号，由于要轮询查找执行器，本条指令大约需要0.5s左右才能返回执行器id，如果长时间收不到执行器地址信息，可认为没有执行器成功连接。返回指令实例：

EE 06 02 00 04 01 64 5A DF 3B 3F ED

其中06即为执行器的ID，01 64 5A DF为执行器序列号。(当有多个执行器成功连接后，会多次返回该指令)

## 与执行器通信

成功查询到执行器后，即可根据执行器ID与执行器进行通信，例如启动ID为6的执行器的指令为：

EE 06 2A 00 01 01 7E 80 ED

其中06为执行器ID，2A为使能或失能执行器的指令，00 01是数据长度，01即为使能（00是失能），7E 80为CRC校验码。

## 控制执行器

位置控制

发送激活梯形位置模式指令:

```
EE 06 07 00 01 06 3F 42 ED
```

数据06为梯形位置模式,参见模式表， 3F 42 为CRC校验码，激活成功后会收到返回：

```
EE 06 07 00 01 01 ED
```

这时可以发送位置指令：

```
EE 06 0A 00 04 01 00 00 00 01 D8 ED
```

其中0A位设置位置值的指令， 01 00 00 00 位设置的位置值，该值是设置目标位置 1R ,通过公式：  
目标位置\*（2^24） 计算取整获得的IQ24值，关于IQ24的介绍请参考 IQmath简介。

## 其他

注意，设置速度电流位置值的命令均无返回，其他指令都有返回，更多与执行器通信指令请参考以太网通信协议命令参考。

# 以太网通信协议命令参考

[ ↑ ](pages/#md-page-menu)

## 读取命令

2.3.1.1 发送数据0字节， 返回数据1字节		
命令名称	读取命令	
说明	此命令类发送数据长度为0，返回数据长度为1	
指令符	见读取指令1	
数据长度	0	
数据内容	无	
指令符（返回值）	见读取指令1	
数据长度 (返回值)	1	
执行器返回数据	0x01：成功/使能/正常	模式查询返回数据见模式表
	0x00：失败/失能/异常	

2.3.1.2 发送数据0字节， 返回数据2字节	
命令名称	读取命令
说明	此命令类发送数据长度为0，返回数据长度为2，读取执行器参数值，高位在前。数值为真实值的2^8倍。 （ 一条特殊指令指令表内已特殊标注 ）
指令符	见读取指令2

数据长度	0	
数据内容	无	
指令符（返回值）	见读取指令2	
数据长度 (返回值)	2	
执行器返回数据	数据为IQ8格式	或见报警指令表

#### 2.3.1.3 发送数据0字节，返回数据4字节

命令名称	读取命令	
说明	此命令类发送数据长度为0，返回数据长度为4，读取执行器参数值，高位在前。数值为真实值的 $2^{24}$ 倍。 (一条特殊指令指令表内已特殊标注)	
指令符	见读取指令3	
数据长度	0	
数据内容	无	
指令符（返回值）	见读取指令3	
数据长度 (返回值)	4	
执行器返回数据	数据为IQ24格式。(一条特殊指令指令表内已特殊标注)	

#### 2.3.1.4 发送数据0字节，返回数据7字节

命令名称	读取命令	
说明	此命令类发送数据长度为0，返回数据长度为7。 (一条特殊指令指令表内已特殊标注)	
指令符	见读取指令4	
数据长度	0	
数据内容	无	
指令符（返回值）	见读取指令4	
数据长度 (返回值)	4	
执行器返回数据	数据为IQ24格式。(一条特殊指令指令表内已特殊标注)	

# 写入命令

2.3.2.1 发送数据1字节，返回数据1字节		
命令名称	写入命令	
说明	此命令类发送数据长度为1，返回数据长度为1，发送数据一个字节表示要写入参数内容。(注：上电后先发送使能指令才能使用，断电前必须先发送失能指令，否则零位可能丢失)	
指令符	见写入指令1	
数据长度	1	
数据内容	0x01：使能	模式设置见模式表
	0x00：失能	
指令 (返回值)	见写入指令1	
数据长度(返回值)	1	
执行器返回数据	0x01：成功	
	0x00：失败	

2.3.2.2 发送数据2字节，返回数据1字节		
命令名称	写入命令	
说明	此命令类发送数据长度为2字节，返回数据长度为1字节，发送数据2个字节表示要写入参数内容，高位在前。数值为真实值的2^8倍。	
指令符	见写入指令2	
数据长度	2	
数据内容	数值为IQ8格式	
	0x00：失能	
指令(返回值)	见写入指令2	
数据长度(返回值)	1	
执行器返回数据	0x01：成功	
	0x00：失败	



### 2.3.2.3 发送数据4字节，返回数据1字节或更少

命令名称	写入命令	
说明	此命令类发送数据长度为4，返回数据长度为1或0，发送数据后4个字节表示要写入的参数内容，数值为真实值的2 <sup>24</sup> 倍。 (一条特殊指令指令表内已特殊标注)	
指令符	见写入指令3	
数据长度	4	
数据内容	数据为IQ24格式。(一条特殊指令指令表内已特殊标注)	
指令(返回值)	见写入指令3	
数据长度(返回值)	1或0	
执行器返回数据	0x01：成功	(三条特殊指令无返回数据， 指令表内已标注)。
	0x00：失败	

### 2.3.2.4 发送数据0字节，返回数据1字节

名称	写入命令	
说明	此命令类发送数据长度为0，返回数据长度为1	
指令符	见写入指令4	
数据长度	0	
数据内容	无	
指令(返回值)	见写入指令4	
数据长度(返回值)	1	
执行器返回数据	0x01：成功	
	0x00：失败	

### 2.3.2.5 发送数据5字节，返回数据1字节

名称	写入命令	
说明	此命令类发送数据长度为5，返回数据长度为1，发送数据前4字节是执行器的序列号，第5字节代表发送参数	

2.3.2.5 发送数据5字节，返回数据1字节	
指令符	见写入指令5
数据长度	0
数据内容	无
指令(返回值)	见写入指令5
数据长度(返回值)	1
执行器返回数据	0x01：成功
	0x00：失败

附录A

[↑](pages/#md-page-menu)

IQmath简介

- 我们使用的处理器一般情况下，要么直接支持硬件的浮点运算，比如某些带有FPU的器件，要么就只支持定点运算，此时对浮点数的处理需要通过编译器来完成。在支持硬件浮点处理的器件上，对浮点运算的编程最快捷的方法就是直接使用浮点类型，比如单精度的float来完成。但是在很多情况下，限于成本、物料等因素，可供我们使用的只有一个定点处理器时，直接使用float类型进行浮点类型的运算会使得编译器产生大量的代码来完成一段看起来十分简单的浮点数学运算，造成的后果是程序的执行时间显著加长，且其占用的资源量也会成倍地增加，这就涉及到了如何在定点处理器上对浮点运算进行高效处理的问题。
- 既然是定点处理器，那么其对定点数，或者说字面意义上的“整数”进行处理的效率就会比它处理浮点类型的运算要高的多。所以在定点处理器上，我们使用定点的整数来代表一个浮点数，并规定整数位数和小数位数，从而方便地对定点数和浮点数进行转换。以一个32位的定点数为例，假设转换因子为Q，即32位中小数的位数为Q，整数位数则为31-Q(有符号数的情况)，则定点数与浮点数的换算关系为：

# Fractional Representation



$$-2^1 + 2^{1-1} + \dots + 2^1 + 2^0 . 2^{-1} + 2^{-2} + \dots + 2^{-Q}$$

## “IQ” – Format

“I” ⇒ INTEGER–Fraction / “Q”⇒ QUOTIENT–Fraction

*Advantage ⇒ Precision same for all numbers in an IQ format*  
*Disadvantage ⇒ Limited dynamic range compared to floating point*

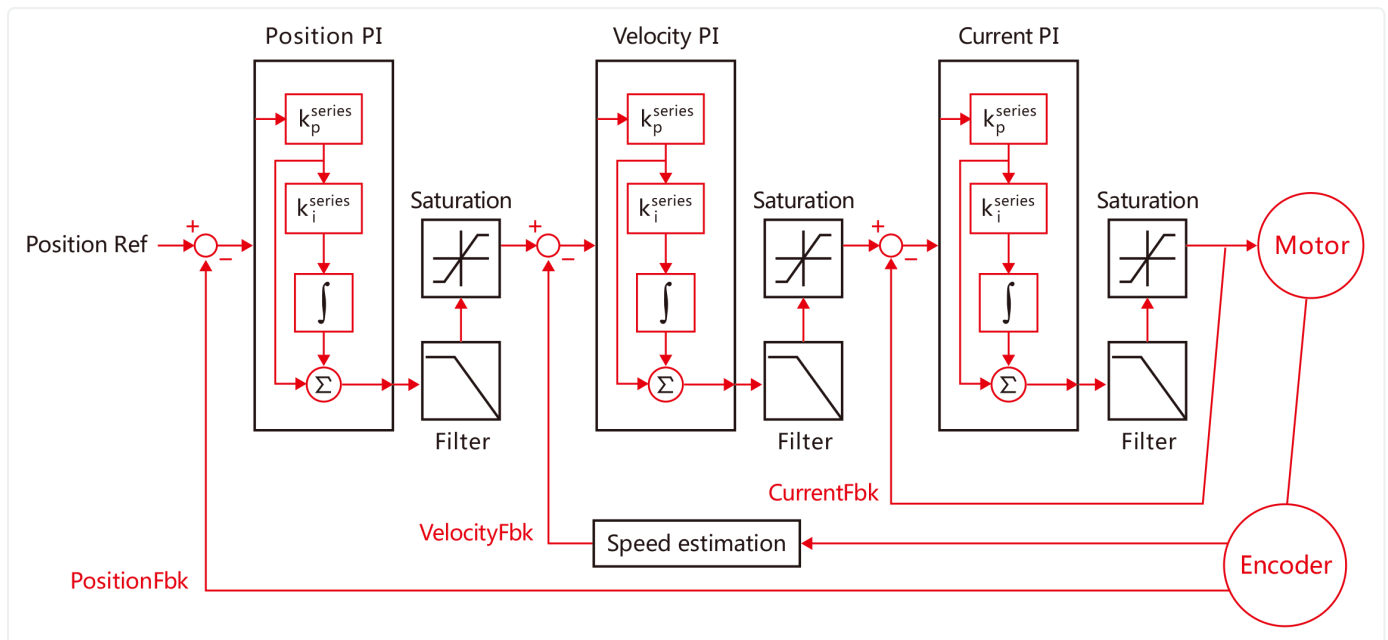
(pages/../../img/3-1通信协议.png)

IQ-MATH Library文档 (pages/../../img/C28x\_IQmath\_Library.pdf)

- 定点数=浮点数×2^Q
- 例如，浮点数-2.0转换到Q为24的定点数时，结果为：定点数=-2×2^24=-33554432
- 32位有符号数的表示范围是：-2147483648到2147483647。如果我们把有符号定点数的最大值2147483647转换为Q为24对应的浮点数，则结果为：浮点数2147483647/2^24=127.999999940

Data Type	Range		Resolution/Precision
	Min	Max	
_iq30	-2	1.999 999 999	0.000 000 001
_iq29	-4	3.999 999 998	0.000 000 002
_iq28	-8	7.999 999 996	0.000 000 004
_iq27	-16	15.999 999 993	0.000 000 007
_iq26	-32	31.999 999 985	0.000 000 015
_iq25	-64	63.999 999 970	0.000 000 030
_iq24	-128	127.999 999 940	0.000 000 060
_iq23	-256	255.999 999 981	0.000 000 119
_iq22	-512	511.999 999 762	0.000 000 238

_iq21	-1024	1023.999 999 523	0.000 000 477
_iq20	-2048	2047.999 999 046	0.000 000 954
_iq19	-4096	4095.999 998 093	0.000 001 907
_iq18	-8192	8191.999 996 185	0.000 003 815
_iq17	-16384	16383.999 992 371	0.000 007 629
_iq16	-32768	32767.999 984 741	0.000 015 259
_iq15	-65536	65535.999 969 482	0.000 030 518
_iq14	-131072	131071.999 938 965	0.000 061 035
_iq13	-262144	262143.999 877 930	0.000 122 070
_iq12	-524288	524287.999 755 859	0.000 244 141
_iq11	-1048576	1048575.999 511 719	0.000 488 281
_iq10	-2097152	2097151.999 023 437	0.000 976 563
_iq9	-4194304	4194303.998 046 875	0.001 953 125
_iq8	-8388608	8388607.996 093 750	0.003 906 250
_iq7	-16777216	16777215.992 187 500	0.007 812 500
_iq6	-33554432	33554431.984 375 000	0.015 625 000
_iq5	-67108864	67108863.968 750 000	0.031 250 000
_iq4	-134217728	134217727.937 500 000	0.062 500 000
_iq3	-268435456	268435455.875 000 000	0.125 000 000
_iq2	-536870912	536870911.750 000 000	0.250 000 000
_iq1	-1073741824	1 073741823.500 000 000	0.500 000 000



(pages/../../img/3-2通信协议-1.png)

- 注释：
- 上位机所设置的上限幅值（Maximum）最高为\_IQ(1.0)，下限幅值（Minimal）最低为\_IQ(-1.0)，起到限幅作用（原理图如图3-2）。例：（如图:3-2用位置环的输出经过限幅模块为速度环的输入，假设\_IQ (0.5)，\_IQ (-0.5) 则位置环输出最大速度应为 $\pm 0.5 \times 6000 = \pm 3000\text{RPM}$ ）
- 比例积分设置值上限为\_IQ(128.0)，设定值下限为\_IQ(-128.0)，但设置值根据实际操作情况调节（原理图: 3-2）
- 电流环设置电流值范围为\_IQ(-1.0)~\_IQ(1.0)之间，电流实际值为IQ值乘以满量程，例：（QDD-6010 型号执行器满量程电流值为33A，\_IQ(0.5) 实际电流值则为 $0.5 \times 33\text{A} = 16.5\text{A}$ ）
- 速度环设置速度值范围为\_IQ(-1.0)~\_IQ(1.0)之间，速度实际值为IQ值乘以满量程（见）。例：（\_IQ(0.5)则实际速度为 $0.5 \times 6000 = 3000\text{RPM}$ ）
- 位置环因为是\_IQ24格式，所以正向满量程为\_IQ(127.999999940)，反向满量程为\_IQ(-128.0)，IQ值即实际值，例：（\_IQ (60.0) 则实际位置为60R，即零位置正向转60转的位置。）
- 速度环曲线模式和位置环曲线模式，可以通过设置加速度，减速度的大小，相对平滑的达到自己预设的速度值和位置，可以避免操作时瞬间电流过大，触发执行器过流保护或者供电电源过流保护。

## CRC校验码计算方法

CRC校验只计算数据内容，即数据位数之后到crc校验之前的所有内容，计算方法采用查表法。

CRC校验码计算方法（c++）：

```

const uint8_t chCRCHTable[] = // CRC 高位字节值表
{
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40
};

```

```

const uint8_t chCRCLTable[] = // CRC 低位字节值表
{
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
    0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
    0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9,
    0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
    0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,
    0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF,
    0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
    0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
    0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB,
    0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
    0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
    0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
    0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
    0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89,
    0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
    0x41, 0x81, 0x80, 0x40
};

```

```
static uint16_t CRC16_1(uint8_t* pchMsg, int16_t wDataLen)
{
    uint8_t chCRCHi = 0xFF; // 高CRC字节初始化
    uint8_t chCRCLo = 0xFF; // 低CRC字节初始化
    int16_t wIndex;          // CRC循环中的索引
    while (wDataLen--)
    {
        // 计算CRC
        wIndex = chCRCHi ^ *pchMsg++;
        chCRCHi = chCRCLo ^ chCRCHTable[wIndex];
        chCRCLo = chCRCLTable[wIndex];
    }
    return ((chCRCHi << 8) | chCRCLo);
}
```

CRC16\_1方法中，pchMsg指向要校验内容的地址，wDataLen为要校验内容的字节长度，该方法会返回两字节 crc 校验码，将校验码加入到发送的指令中的对应位置即可。收到返回带 crc 校验码的指令，也可以通过该方法计算出校验码并与返回指令中的校验码比对，验证数据的有效性。

示例

执行器使能指令:

```
0xEE 0x06 0x2A 0x00 0x01 0x01 0x7E 0x80 0xED
```

其中 0xEE 为帧头，0x06 是执行器ID，0x2A 为使能指令符，0x00 0x01 为数据长度，0x01 为数据内容，0xED 为帧尾，其中数据内容 0x01 通过上述方法计算出的校验码即为 0x7E 0x80。

A.1读取指令编码定义表

读取指令1:

A.1.1 读取指令1		
指令符	定义	说明
0x00	握手	上位机发送本指令，执行器响应返回，说明执行器已经准备好与上位机通信也可作为心跳协议，实时查询从机的状态。
0x55	查询执行器当前模式	读取执行器所管理的执行器的当前的模式。
0xB0	查询执行器上次失能状态	读取执行器的上次失能状态，正常/异常
0x71	电流环滤波器状态	读取指定ID执行器的电流环滤波器使能/失能
0x75	速度环滤波器状态	读取指定ID执行器的速度环滤波器使能/失能
0x79	位置环滤波器状态	读取指定ID执行器的位置环滤波器使能/失能
0x8b	位置限位状态	读取指定ID执行器的位置限位使能/失能
0x2B	电流环滤波器状态	读取指定ID执行器使能/失能状态。

## 读取指令2:

A.1.2 读取指令2		
指令符	定义	说明
0x45	执行器电压	读取指定ID执行器的电压 (数据值为真实值的 $2^{10}$ 倍)
0x53	执行器电流量程	读取指定ID执行器的电流量程
0x73	电流环滤波器的带宽	读取指定ID执行器电流环滤波器的带宽(Hz)
0x77	速度环滤波器的带宽	读取指定ID执行器速度环滤波器的带宽(Hz)
0x7B	位置环滤波器的带宽	读取指定ID执行器速度环滤波器的带宽(Hz)
0x5F	执行器电机的温度	读取指定ID执行器电机温度°C
0x60	逆变器温度	读取指定ID执行器的逆变器温度°C
0x62	逆变器保护温度	读取指定ID执行器的逆变器保护温度°C
0x64	逆变器恢复温度	读取指定ID执行器的逆变器恢复温度°C
0x6C	电机保护温度	读取指定ID执行器的电机保护温度°C
0x6E	电机恢复温度	读取指定ID执行器的电机恢复温度°C
0xFF	报警指令(特殊指令)	执行器的报警信息

## 读取指令3:

A.1.3 读取指令3		
指令符	定义	说明
0x04	当前电流值	读取指定ID执行器的当前电流值，电流真实值需要乘以电流满量程，单位为A
0x05	当前速度值	读取指定ID执行器的当前速度值，速度真实值需要乘以速度满量程，单位为RPM
0x06	当前位置值	读取指定ID执行器的当前位置值，单位为R
0x17	速度环的P	读取指定ID执行器的当前速度环的P
0x18	速度环的I	读取指定ID执行器的当前速度环的I
0x19	位置环的P	读取指定ID执行器的当前位置环的P



0x1A	位置环的I	读取指定ID执行器的当前位置环的I
0x1C	位置梯形曲线的最大速度	读取指定ID执行器的当前位置梯形曲线的最大速度
0x1D	位置梯形曲线的加速度	读取指定ID执行器的当前位置梯形曲线的最大加速度
0x1E	位置梯形曲线的减速度	读取指定ID执行器的当前位置梯形曲线的最大减速度
0x22	速度梯形曲线的最大速度	读取指定ID执行器的当前速度梯形曲线的最大速度
0x23	速度梯形曲线的加速度	读取指定ID执行器的当前速度梯形曲线的最大加速度
0x24	速度梯形曲线的减速度	读取指定ID执行器的当前速度梯形曲线的最大减速度
0x34	电流环输出的下限	读取指定ID执行器的当前电流环输出的下限
0x35	电流环输出的上限	读取指定ID执行器的当前电流环输出的上限
0x36	速度环输出的下限	读取指定ID执行器的当前速度环输出的下限
0x37	速度环输出的上限	读取指定ID执行器的当前速度环输出的上限
0x38	位置环输出的下限	读取指定ID执行器的当前位置环输出的下限
0x39	位置环输出的上限	读取指定ID执行器的当前位置环输出的上限
0x59	电流输入的限幅	读取指定ID执行器的电流输入的限幅
0x5B	速度输入的限幅	读取指定ID执行器的速度输入的限幅
0x7D	惯量	读取指定ID执行器的惯量
0x85	执行器位置的上限	读取指定ID执行器的位置上限值
0x86	执行器位置的下限	读取指定ID执行器的位置下限值
0x8A	执行器位置偏置位置	读取指定ID执行器的位置偏置值
0x92	执行器自动归零时电流的下限	读取指定ID执行器自动归零时电流的下限
0x93	执行器自动归零时电流的上限	读取指定ID执行器自动归零是电流的上限

0x7F	堵转能量	读取指定ID执行器的堵转能量。（数值为真实值的75.225倍）堵转后发热能量，单位为J。
------	------	--

#### 读取指令4:

A.1.4 读取指令4		
指令符	定义	说明
0x94	执行器电流速度位置	读取指定ID执行器的电流速度位置。 前3字节表示位置值,数值为真实值的 $2^{16}$ 倍 中间2字节表示速度值,数值为真实值的 $2^{14}$ 倍 最后2字节表示电流值,数值为真实值的 $2^{14}$ 倍

## A.2 写入命令编码值定义表

#### 写入指令1:

A.2.1写入指令1		
指令符	定义	说明
0x07	设置指定ID执行器的模式	设置指定ID执行器的当前的模式。
0x70	电流环滤波器状态	设置指定ID执行器的电流环滤波器使能/失能
0x74	速度环滤波器状态	设置指定ID执行器的速度环滤波器使能/失能
0x78	位置环滤波器状态	设置指定ID执行器的位置环滤波器使能/失能
0x8C	位置限位状态	设置指定ID执行器的位置限位使能/失能
0x2A	执行器的开失能状态	设置指定ID执行器使能/失能

#### 写入指令2:

A.2.2写入指令2		
指令符	定义	说明
0x72	电流环滤波器的带宽	设置指定ID执行器电流环滤波器的带宽(Hz)
0x76	速度环滤波器的带宽	设置指定ID执行器速度环滤波器的带宽(Hz)
0x7A	位置环滤波器的带宽	设置指定ID执行器位置环滤波器的带宽(Hz)
0x61	逆变器保护温度	设置指定ID执行器的逆变器保护温度 $^{\circ}\text{C}$

**A.2.2写入指令2**

0x63	逆变器恢复温度	设置指定ID执行器的逆变器恢复温度℃
0x6B	电机保护温度	设置指定ID执行器的电机保护温度℃
0x6D	电机恢复温度	设置指定ID执行器的电机恢复温度℃

**写入指令3:****A.2.3写入指令3**

指令符	定义	说明
0x08	当前电流值	设置指定ID执行器的当前电流值。(注：无返回数据)
0x09	当前速度值	设置指定ID执行器的当前速度值。(注：无返回数据)
0x0A	当前位置值	设置指定ID执行器的当前位置值。(注：无返回数据)
0x10	速度环的P	设置指定ID执行器速度环的P值
0x11	速度环的I	设置指定ID执行器速度环的I值
0x12	位置环的P	设置指定ID执行器位置环的P值
0x13	位置环的I	设置指定ID执行器位置环的I值
0x1F	位置梯形曲线的最大速度	更改指定ID执行器位置梯形曲线的最大速度
0x20	位置梯形曲线的加速度	更改指定ID执行器位置梯形曲线的最大加速度
0x21	位置梯形曲线的减速速度	更改指定ID执行器位置梯形曲线的最大减速速度
0x25	速度梯形曲线的最大速度	更改指定ID执行器速度梯形曲线的最大速度
0x26	速度梯形曲线的加速度	更改指定ID执行器速度梯形曲线的加速度
0x27	速度梯形曲线的减速速度	更改指定ID执行器速度梯形曲线的减速速度
0x2E	电流环输出的下限	更改指定ID执行器电流环输出的下限
0x2F	电流环输出的上限	更改指定ID执行器电流环输出的上限
0x30	速度环输出的下限	更改指定ID执行器速度环输出的下限
0x31	速度环输出的上限	更改指定ID执行器速度环输出的上限

**A.2.3写入指令3**

0x32	位置环输出的下限	更改指定ID执行器位置环输出的下限
0x33	位置环输出的上限	更改指定ID执行器位置环输出的上限
0x58	电流输入的限幅	设置指定ID执行器的电流输入的限幅
0x5A	速度输入的限幅	设置指定ID执行器的速度输入的限幅
0x83	执行器位置的上限	更改指定ID执行器的位置上限值
0x84	执行器位置的下限	更改指定ID执行器的位置下限值
0x87	执行器的Home值	设置指定ID执行器的Home值
0x89	执行器的位置偏置	设置指定ID执行器的位置偏置值
0x90	执行器自动归零时电流的下限	设置指定ID执行器自动归零时电流的下限
0x91	执行器自动归零时电流的上限	设置指定ID执行器自动归零时电流的上限
0x7E	堵转能量	设置指定ID执行器的堵转能量。(数值为真实值的75.225倍)堵转后发热能量，单位为J

**写入指令4:****A.2.4写入指令4**

指令符	定义	说明
0xFE	消除执行器的报警	消除执行器的报警动作，接收到命令后，执行器停止报警，否则执行器不可操作
0x88	清除Homing数据	清除Homing数据
0x0D	存储参数	存储参数到EEPROM

**写入指令5:****A.2.4写入指令5**

指令符	定义	说明
0x3D	修改执行器ID	修改执行器ID,修改成功后需要使用新的ID进行通信。

# 附录B:模式表

[ ↑ ] (pages/#md-page-menu)

指令符	指令符
01	电流模式
02	速度模式
03	位置模式
06	梯形位置模式
07	梯形速度模式
08	homing模式

# 附录C:报警指令表

[ ↑ ] (pages/#md-page-menu)

指令符	错误提示	原因	检查与解决
0x0001	过压异常	执行器电压超过45V	检查电源电压/检查HUB上是否插入大电容
0x0002	欠压异常	执行器电压低于15V	检查电源电压/检查电源功率是否不足
0x0004	堵转异常	位置不动的情况下 电流长时间过大	检查执行器输出端/检查相电流
0x0008	过热异常	电机温度达到保护 温度	检查电流是否过大/等待执行器降温至恢复温 度以下即可
0x0020	多圈计数异常	多圈计数功能异常	内置电池电量不足，联系售后更换内置电池即 可
0x0200	阶跃过大	阶跃过大	位置模式阶跃大约1R/速度模式阶跃大于 600RPM 清除错误，设置正常阶跃即可
0x0010	读写参数异常	读写参数异常	联系售后维修
0x0100	电机温度传感 器异常	电机温度传感器异 常	联系售后维修
0x0040	逆变器温度传 感器异常	逆变器温度传感器 异常	联系售后维修
0x0080	CAN通信异常	CAN通信异常	联系售后维修
0x0400	DRV保护	DRV保护	联系售后维修

指令符	错误提示	原因	检查与解决
0x0800	编码器故障	编码器故障	联系售后维修
其他	设备异常		
注释	可同时报警多个错误，如返回数据为0005，则错误为0001过压异常与0004堵转异常		

# 版本变更记录

[ ↑ ] (pages/#md-page-menu)

版本号	更新时间	更改类型	位置	更新内容
V1.1.0	2019.05.17	增加	附录A	增加协议0x3D,0x45,0x53,0x6B-0x6E,0x8B,0x8C,0x94的协议说明
		修改	全文	专有名词大写，关键字变色
		增加	以太网通信协议命令参考	增加读取命令4和写入命令5类型
V1.0.0	2019.04.28	增加	全文	第一版