

目录

1.1	通信协议.....	3
1.2	数据包	4
1.2.1	发送命令和接收命令格式.....	6
1.2.2	发送数据和接收数据格式.....	6
2.1	系统命令类型数据包和系统功能.....	7
2.2	系统命令的使用方式.....	7
2.2.1	应答数据包格式（AA 和 AB）	7
2.2.2	检测从机是否在线命令标志 AC	8
2.2.3	读取电机位置状态命令 F1（E1）	8
2.2.4	读取电机速度命令 F1（E2）	10
2.2.5	读取电机和驱动器参数命令 F1（E3）	11
2.2.6	读取输入输出端口状态命令 F1（E4）	13
2.2.7	读取系统其他参数命令 F1（E5）	14
3.1	手动模式命令的使用.....	16
3.1.1	手动命令的标准格式.....	16
4.1	程序模式命令的使用.....	17
4.1.1	程序模式命令的标准格式.....	17
5.1	扩展端口命令的使用.....	21
6.1	系统数据类型的数据包.....	21
6.2	系统数据类型数据包修改系统参数.....	23
6.2.1	修改从机的电机行程值 F1（E1）	23
6.2.2	修改从机的电机速度值 F1（E2）	25
6.2.3	修改从机的电机参数 F1（E3）	26
6.2.4	修改从机的输出端口状态 F1（E4）	28
6.2.5	修改从机的其他系统参数 F1（E5）	29
7.1	产品目录.....	32

版本记录:

1) V1.00 2016-10-20

V1.00 2016-10-20

使用我司的产品进行通信的时候，目前有两种的通信方式：（1）485 单工通行；（2）串口 232 通信。但无论使用哪种通信方式，整个控制方案中都只能有一个主机，485 通信可以有多个从机（分机），232 通信就只能有一个从机，而在整个系统的通信过程中，主机是主动发出命令，从机是被动接受命令，然后根据命令作出相应的数据反馈，正常的通信，总是“一问一答”。

我司的产品中，既有可以做主机的，也有做从机的，在系统开发的工程中，要根据使用场合的特点，选择合适的产品。

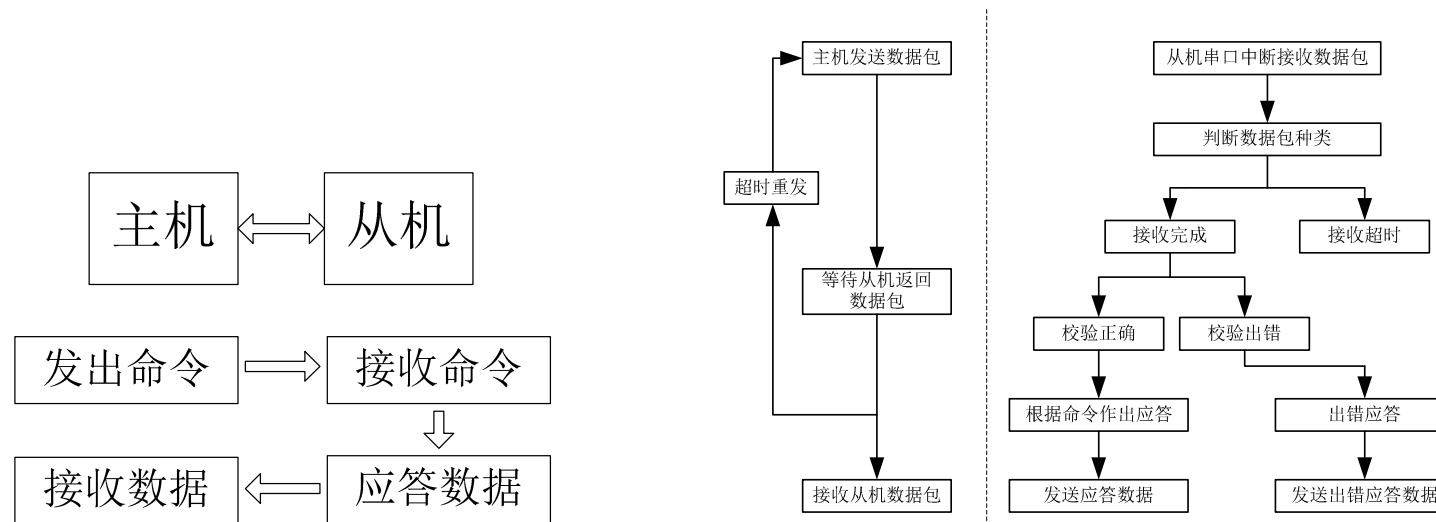


图 1 通信连接

1.1 通信协议

所谓通信协议，就是在使用产品进行通信的时候，主机和从机之间，相互约定俗成的“对话格式”。无论是作为主机还是从机，都必须严格按照协议的规格发送命令或者数据包，否则通信对方将会收到无效的数据而不作出应答或者相应的动作。

通信协议或者通信相关概念：

- 1) 主机：在整个通信网络中等级最高，负责控制所有从机，主动发送命令或者数据。
- 2) 从机：在整个通信网络中负责接收主机的命令和数据，然后做出相关的动作和数据反馈。如果主机没有给它发数据，从机是不能主动给主机发数据。
- 3) 总线：无论是 232 还是 485，我们都统一把连接主机和从机之间的通信线路定义为总线，方便下面的介绍。

4) 数据包：数据包顾名思义就是一段数据被打包，封装成为一个有规律的整体，其中的规律就是数据有一定的排列规则、固定的数据值、约定的数据长度等。数据包包括这几个类别：1.系统命令类型数据包；2.应答类型数据包；3.系统数据类型数据包；4.程序数据类型数据包；5.程序编辑类型数据包；6.动作命令类型数据包

1.2 数据包

由于数据包的种类比较多，而且不同类型的数据包需要承载的数据量不同，长度有差异，但通过统一数据包的格式，能够让总线上的设备都能接收和理解。

数据包主要包含以下几部分：

- 1) 数据头：代表一个数据包的起始，也是提示从机这是一个数据包，可以开始接收。
- 2) 数据包标识：记录了这个数据包的类别，是系统命令类型数据包还是应答类型数据包或者是其他。
- 3) 数据内容：承载了这个数据包的“意义”，让接收方（主机或者从机）知道对方的目的。

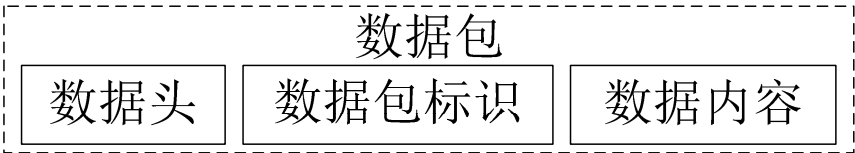


图 2 数据格式

在进行系统开发的时候，用户可以根据开发平台的不同，使用不同的开发语言，但即使语言的不同，但思维和方式都是一致的，所以在开发的时候，我们可以为数据设定一些的<结构体>，让数据包名副其实的成为一个整体。以下是我司的一个例子，方便以下介绍的时候使用。

命令结构体（系统命令、程序编辑、应答数据）	数据结构体（系统数据、程序数据、动作命令）
<div>//串口发送命令和接收命令的数据结构</div> <div>struct UART_ORDER</div> <div>{</div> <div>u8 ORDER; //命令</div> <div>u8 LEVEL; //发送命令的机器处于什么等级， ==2 系统主机，</div> <div>==1 主机， ==0 从机</div> <div>u8 OBJECT; //命令指向的对象，也就是需要接收命令的机器</div>	<div>//串口接收到的应答数据结构或更新数据的命令结构</div> <div>struct UART_DATA</div> <div>{</div> <div>u8 ORDER; //命令</div> <div>u8 LEVEL; //发送命令的机器处于什么等级， ==2 系统主机，</div> <div>==1 主机， ==0 从机</div> <div>u8 OBJECT; //命令面向的对象，如果是主机发出，那么 OBJECT</div>

<pre> u8 CLASS; //命令需要操作的种类，如修改轴速度，读取长度 u8 TARGET; //命令需要操作的项目，如修改 X 轴的最高速度， 读取 X 的行程 u8 OCR; //校验码，等于上面 5 个字节的总和，溢出部分不处 理 }; </pre>	<p>代表的就是从机序号，如果是从机发出，代表的就是从机序号</p> <pre> u8 CLASS; //命令需要操作的种类，如修改轴速度，读取长度 u8 TARGET; //命令需要操作的项目，如修改 X 轴的最高速度， 读取 X 的行程 u8 OCR; //校验码，等于上面字节的总和，溢出部分不处理 u8 PARAMETER[8]; //命令需要操作的具体内容，如修改那个轴的速 度值，读取那个轴的长度值 //PARAMETER[0-3]通用寄存器，PARAMETER[4-7] 用于串口读写程序时缓存程序内容 }; u8 LABEL[2]; //串口读写程序时缓存程序标签号 </pre>
--	---

数据包格式：数据都是 HEX 格式（HEX 就是 16 进制），每个数据包都会有校验码，但校验码只等于<数据内容>部分除去校验码的其他数据的和，溢出部分舍弃。如 UO.OCR=UO.ORDER+UO.LEVEL+UO.OBJECT+UO.CLASS+UO.TARGET。

数据头	数据包标识	数据内容															
0-1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
5A,A5	0X0A（系统命令）	命令标识	等级	从机序号	种类	项目	校验码										
5A,A5	0X0B（系统数据）	命令标识	等级	从机序号	种类	项目	参数0	参数1	参数2	参数3	校验码						
5A,A5	0X0C（应答）	命令标识	等级	从机序号	种类	项目	校验码										
5A,A5	0X0D（程序数据）	命令标识	等级	从机序号	种类	程序	命令号0	命令号1	程序D0	程序D1	程序D2	程序D3	程序D4	程序D5	程序D6	程序D7	校验码
5A,A5	0X0E（程序	命令	等	从机	种	程	命令	命令	校验								

	命令)	标识	级	序号	类	序	号 0	号 1									
5A,A5	0X0F (动作命令)	命令标识	等级	从机序号	种类	空白	空白	空白	数据 0	数据 1	数据 2	数据 3	数据 4	数据 5	数据 6	数据 7	校验码

1.2.1 发送命令和接收命令格式

适用于：系统命令类型数据包、程序编辑类型数据包、应答类型数据包

Byte 0	Byte1	Byte 2	Byte 3	Byte 4	Byte 5
命令	等级	对象	种类	项目 (目标)	校验
UO.ORDER	UO.LEVEL	UO.OBJECT	UO.CLASS	UO.TARGET	UO.OCR

1.2.2 发送数据和接收数据格式

适用于：系统数据类型数据包 (包括由主机发出的和由从机发出的)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
命令	等级	对象	种类	项目	参数内容 0	参数内容 1	参数内容 2	参数内容 3	校验
UD.ORDER	UD.LEVEL	UD.OBJECT	UD.CLASS	UD.TARGET	UD.PARAMETER[0]	UD.PARAMETER[1]	UD.PARAMETER[2]	UD.PARAMETER[3]	UD.OCR

另外还有一种数据包，适用于：程序数据类型数据包、动作命令类型数据包

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7~Byte 14	Byte 15
命令	等级	对象	种类	项目	命令号 0	命令号 1	参数内容 0~参数内容 7	校验
UD.ORDER	UD.LEVEL	UD.OBJECT	UD.CLASS	UD.TARGET	UD.TABLET[0]	UD.TABEL[1]	UD.PARAMETER[0]~UD.PARAMETER[7]	UD.OCR

2.1 系统命令类型数据包和系统功能

系统命令类型数据包只能由主机发出，一般用于查询从机的各项系统参数和修改系统功能。

系统命令发出后，只有 4 种可能性：

- 1) 如果命令是查询数据的，那么从机就会返回对应的系统数据类型数据包；
- 2) 如果命令是需要应答而已，就会返回应答数据包；
- 3) 如果命令发出后，本来是返回数据包的情况，但从机接收到数据判断数据有问题，就会返回出错应答，并不是系统数据类型数据包；
- 4) 命令发出后，没有任何响应。原因：1.总线连接有问题；2.查询的从机不存在或者从机号不匹配

命令	作用	应答数据	作用	应答类型	
0XF9	查询运动系统的相关数据	0XF9	返回相关项目的数据包	数据	
0XAC	主机检测从机是否有连接	0XAA(0XBB)	返回成功应答或者失败应答	应答	
0XFB	读取和修改扩展端口的状态	0XFC	返回端口的状态数据包	数据	
0XF3	发送手动模式时候的动作命令	0XAA(0XBB)	返回成功应答或者失败应答	应答	
0XF4	程序模式下的运动控制命令	0XF4	返回成功应答或者失败应答	应答、数据	

2.2 系统命令的使用方式

2.2.1 应答数据包格式（AA 和 AB）

在接收到主机命令后，如果没有数据返回，从机都会发送一个应答数据类型的数据包

	Byte 0	Byte1	Byte2	Byte 3	Byte 4	Byte 5
接收正确	AA	UO.LEVEL	UO.OBJECT	0X00	UO.ORDER	UO.OCR
接收出错	AB	UO.LEVEL	UO.OBJECT	0X00	UO.ORDER	UO.OCR

2.2.2 检测从机是否在线命令标志 AC

主机命令	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
	AC	UO.LEVEL	UO.OBJECT	AC	AC	UO.OCR
从机应答	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
接收正确	AA	UO.LEVEL	UO.OBJECT	0X00	AC	UO.OCR
接收出错	AB	UO.LEVEL	UO.OBJECT	0X00	AC	UO.OCR

举例：主机寻找序号为 3 的从机，通信过程如下：

主机：0xFF 0xA5 0x0A 0xAC 0x01 0x03 0xAC 0xAC 0x08

从机：0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xAC 0x59

检测从机是否在线命令一般不会出现错误应答，因为从机要不不存在，没有应答；要不总线线路有问题，从机收不到命令。

2.2.3 读取电机位置状态命令 F1 (E1)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UO.LEVEL	UO.OBJEC T	UO.CLASS	UO.TARGE T	UO.OCR				
	0xF1	0x01	0x03	0xE1	0xA1	0x77				
从机应答	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x00	0x03	0xE1	0xA1	0x01	0x01	0x00	0x00	0x78
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机位置包含几个关键参数，如电机处于正半轴还是负半轴，也就是说行程值是正还是负，还有电机现在是在移动还是停止，所以通过 UO.TARGET 这个值可以读取不同的参数。

UO. TARGET	行程值	行程方向	电机状态（运动还是停止）
X 轴电机	0xA1	0xA2（=0 正半轴，=1 负半轴）	0xA3（=1 停止，=0 运动）
Y 轴电机	0xB1	0xB2	0xB3
Z 轴电机	0xC1	0xC2	0xC3
A 轴电机	0xD1	0xD2	0xD3

举例：主机读取序号为 3 的从机设备上 X 轴的位置信息，通信过程如下：

主机命令 1：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE1 0xA1 0x77

从机 : 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE1 0xA1 0x01 0x01 0x00 0x00 0x78

主机命令 2：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE1 0xA2 0x77

从机 : 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE1 0xA2 0x01 0x00 0x00 0x00 0x78

主机命令 3：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE1 0xA3 0x77

从机 : 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE1 0xA4 0x01 0x00 0x00 0x00 0x7A

主机查询从机 3 的 X 轴位置，命令 1 读取回来的数据是 UD.PARAMETER[3]-UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00000101，十进制的数值为 257，意思是电机走了 257 个脉冲量，具体长度值需要根据整个系统的参数计算出来；命令 2 读取回来的数据是 UD.PARAMETER[3]-UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00000001，电机处于负半轴，行程值符号为负；命令 3 读取回来的数据是 UD.PARAMETER[3]-UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00000001，电机处于停止状态。

2.2.4 读取电机速度命令 F1 (E2)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UO.LEVEL	UO.OBJEC T	UO.CLASS	UO.TARGE T	UO.OCR				
	0xF1	0x01	0x03	0xE2	0xA1	0x78				
从机应答	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x00	0x03	0xE2	0xA1	0xC8	0x00	0x00	0x00	0x3F
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机速度包含几个关键参数，如电机的最低速度，最高速度，加速时间，手动速度等，所以通过 UO.TARGET 这个值可以读取不同的参数。

UO. TARGET	最低速度	最高速度	加速时间	手动速度	
X 轴电机	0xA1	0xA2	0xA3	0xA4	
Y 轴电机	0xB1	0xB2	0xB3	0xB4	
Z 轴电机	0xC1	0xC2	0xC3	0xC4	
A 轴电机	0xD1	0xD2	0xD3	0xD4	

举例 1：主机读取序号为 3 的从机设备上 X 轴的速度信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE2 0xA1 0x78

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE2 0xA1 0xC8 0x00 0x00 0x00 0x3F

主机查询从机 3 的 X 轴的最低速度，命令读取回来的数据是 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x000000C8，十进制的数值为 200，意思是电机的最低速度为 200Hz（赫兹）。

举例 2：主机读取序号为 3 的从机设备上 X 轴的速度信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE2 0xA3 0x7A

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE2 0xA3 0x08 0x00 0x00 0x00 0x81

主机查询从机 3 的 X 轴的加速时间，命令读取回来的数据是 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00000008，十进制的数值为 8，加速时间是以 0.1 秒为最小单位，所以读取回来的电机的加速时间为 0.8 秒，理论上来说，电机从最低速度到最高速度的时间是 0.8 秒。

2.2.5 读取电机和驱动器参数命令 F1（E3）

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UO.LEVEL	UO.OBJECT	UO.CLASS	UO.TARGET	UO.OCR				
	0xF1	0x01	0x03	0xE3	0xA1	0x79				
从机应答	F1	UD.LEVEL	UD.OBJECT	UD.CLASS	UD.TARGET	UD.PARAMETER[0]	UD.PARAMETER[1]	UD.PARAMETER[2]	UD.PARAMETER[3]	UD.OCR
	0xF1	0x00	0x03	0xE3	0xA1	0xC8	0x00	0x00	0x00	0x40
出错应答	AB	UO.LEVEL	UO.OBJECT	0X00	UO.ORDER	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机和驱动器参数包含几个关键参数，如电机的步进角，驱动器细分，导程，所以通过 UO.TARGET 这个值可以读取不同的参数。

UO. TARGET	电机步进角	驱动器细分	导程		
X 轴电机	0xA1	0xA2	0xA3		

Y 轴电机	0xB1	0xB2	0xB3		
Z 轴电机	0xC1	0xC2	0xC3		
A 轴电机	0xD1	0xD2	0xD3		

举例 1：主机读取序号为 3 的从机设备上 X 轴的电机步进角信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE3 0xA1 0x79

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE3 0xA1 0xC8 0x00 0x00 0x00 0x40

主机查询从机 3 的 X 轴的电机步进角信息，命令读取回来的数据是 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x000000C8，十进制的数值为 200，意思是电机转一圈需要 200 个脉冲，也就是说这是一个 1.8° 的电机。

举例 2：主机读取序号为 3 的从机设备上 X 轴的驱动器细分信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE3 0xA2 0x7B

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE3 0x10 0x00 0x07 0x00 0x00 0x81

主机查询从机 3 的 X 轴驱动器细分，命令读取回来的数据是 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00000010，十进制的数值为 16，所以电机的驱动器细分为 16。

举例 3：主机读取序号为 3 的从机设备上 X 轴的导程信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE3 0xA3 0x7B

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE3 0xA3 0xD0 0x07 0x00 0x00 0x81

主机查询从机 3 的 X 轴导程，命令读取回来的数据是 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x000007D0，十进制的数值为 2000，导程的最小单位是 0.001 毫米，所以实际的导程是 0.001*2000=2（毫米）。

综合上面读取回来的数据，电机是 1.8°，驱动器细分为 16，所以电机转一圈需要 3200 个脉冲；导程是 2 毫米，如果是丝杆带动的滑台，也就是说电机转一圈，滑台移动了 2 毫米。

2.2.6 读取输入输出端口状态命令 F1 (E4)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UO.LEVEL	UO.OBJEC T	UO.CLASS	UO.TARGE T	UO.OCR				
	0xF1	0x01	0x03	0xE4	0xA1	0x7A				
从机应答	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x00	0x03	0xE4	0xC1	0x01	0x00	0x03	0x00	0x9D
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x92				

同时，由于从机拥有多个输入或者输出端口，所以通过 UD.PARAMETER[0]--UD.PARAMETER[3]，分别代表不同的端口状态

数据	UD.PARAMETER[0]	UD.PARAMETER[1]	UD.PARAMETER[2]	UD.PARAMETER[3]	
端口	输入端 K1-K8	输入端 K9-K16	输出端 P1-P8	输出端 K9-K16	

举例 1：主机读取序号为 3 的从机设备上的端口信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE4 0x00 0xD8

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE4 0xC1 0x01 0x00 0x03 0x00 0x9D

主机查询从机 3 的端口状态信息，命令读取回来的数据是 UD.PARAMETER[0]，16 进制数据为 0x01，二进制数据值为 00000001，8 位的二进制数每一位分别对应一个端口，所以输入端 K1 为触发状态；UD.PARAMETER[2]，16 进制数据为 0x03，二进制数据值为 00000011，表示输出 P1 和 P2 正处于工作（闭合）状态。

2.2.7 读取系统其他参数命令 F1 (E5)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UO.LEVEL	UO.OBJEC T	UO.CLASS	UO.TARGE T	UO.OCR				
	0xF1	0x01	0x03	0xE5	0xC1	0x7B				
从机应答	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x00	0x03	0xE5	0xC1	0x01	0x00	0x00	0x00	0x9B
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x92				

系统参数还包含其他多个项目，如电机使能、输入设置、刹车设置等，所以通过 UO.TARGET 这个值可以读取不同的参数。

UO. TARGET	0xC1	0xC2	0xC3	0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xCB
参数	电机使能	输入设置	刹车设置	声音设置	读取密码	密码设置	启动设置	系统信息	脉冲方式		

举例 1：主机读取序号为 3 的从机设备上的电机使能信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC1 0x9B

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC1 0x01 0x00 0x00 0x00 0x9B

主机查询从机 3 的电机使能信息，命令读取回来的数据 UD.PARAMETER[0]，16 进制数据为 0x01，二进制的数值为 00000001，从低位向高位分别代表 XYZA 轴的使能状态，现在最低位是 1，也就是 X 轴是一直使能，YZA 都是没有使能，电机处于可以自由旋转的状态。

举例 2：主机读取序号为 3 的从机设备的输入设置信息，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC2 0x9C

从机: 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC2 0x06 0x07 0x08 0x00 0xB0

主机查询从机 3 的输入设置, 命令读取回来的数据 UD.PARAMETER[0]=0x06, 意思是系统的启动键是 K6, UD.PARAMETER[1]=0x07, 意思是系统的停止键是 K7, UD.PARAMETER[2]=0x08, 意思是系统的暂停键是 K8, 如果 UD.PARAMETER[2]=0x00, 也就是没有设置外部的暂停键功能, 暂停键只能使用 K8, 启动和停止键可以选用外部输入端口的 K1—K7。

举例 3: 主机读取序号为 3 的从机的电机刹车设置信息, 通信过程如下:

主机: 0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC3 0x9D

从机: 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC3 0x03 0x00 0x00 0x00 0x9F

主机查询从机 3 的电机刹车信息, 命令读取回来的数据 UD.PARAMETER[0], 16 进制数据为 0x03, 二进制的数值为 00000011, 从低位向高位分别代表 XYZA 轴的刹车状态, 现在最低位是 11, 也就是 X 轴和 Y 轴是带刹车功能, ZA 都是不带刹车。(刹车功能是在手动模式或者直线命令的时候, 电机在停止前带有减速过程, 如果不带刹车, 停止的时候, 电机是立刻停止, 机器会有一定的抖动)

举例 4: 主机读取序号为 3 的从机系统声音设置信息, 通信过程如下:

主机: 0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC4 0x9E

从机: 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC4 0x01 0x00 0x00 0x00 0x9E

主机查询从机 3 的声音设置, 命令读取回来的数据 UD.PARAMETER[0]=0x01, 意思是系统的声音功能打开, 电机转动的时候有模拟电机转动的声音输出, 如果 UD.PARAMETER[0]=0x00, 系统的声音功能关闭。

举例 5: 主机读取序号为 3 的从机启动设置信息, 通信过程如下:

主机: 0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC7 0xA1

从机: 0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC7 0x01 0x00 0x00 0x00 0xA1

主机查询从机 3 的启动设置, 命令读取回来的数据 UD.PARAMETER[0]=0x01, 意思是系统在上电的时候自动运行, UD.PARAMETER[0]=0x00, 系统上电的时候是不会自动运行。

举例 6: 主机读取序号为 3 的从机系统信息, 通信过程如下:

主机: 0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC8 0xA1

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC8 0x01 0x00 0x00 0x00 0xA1

主机查询从机 3 的系统信息，详细的代号信息可以与技术支持联系。

举例 7：主机读取序号为 3 的从机的电机脉冲方式，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF1 0x01 0x03 0xE5 0xC9 0xA1

从机：0x5A 0xA5 0x0B 0xF1 0x00 0x03 0xE5 0xC9 0x01 0x00 0x00 0x00 0xA1

主机查询从机 3 的电机脉冲方式，命令读取回来的数据 UD.PARAMETER[0]=0x01，16 进制数据为 0x01，二进制的数值为 00000001，从低位向高位分别代表 XYZA 轴的脉冲方式，现在最低位是 1，也就是 X 轴的脉冲输出方式是双脉冲，YZA 轴的脉冲输出方式是单脉冲（脉冲+方向）。

注意：单脉冲（脉冲+方向）；双脉冲（正转脉冲+反转脉冲）

3.1 手动模式命令的使用

手动命令是用于手动模式下控制电机的自由运动，方便进行位置的微调。手动命令的发送格式是属于系统命令类型的数据包。

3.1.1 手动命令的标准格式

主机命令	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
	F3	UO.LEVEL	UO.OBJECT	UO.CLASS	UO.TARGET	UO.OCR
从机应答	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
接收正确	AA	UO.LEVEL	UO.OBJECT	0X00	F3	UO.OCR
接收出错	AB	UO.LEVEL	UO.OBJECT	0X00	F3	UO.OCR

举例：主机控制序号为 3 的从机的 X 轴正方向运动，通信过程如下：

主机：0x5A 0xA5 0x0A 0xF3 0x01 0x03 0x03 0xA1 0x9B

从机：0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xF3 0xA0

主机的手动命令一旦发出，从机会有应答信号返回，如果是正确的应答信号，那么对应的电机会一直运动，直到收到下一条指令才会改变。目前的分机在手动模式下，同时只能有一个电机处于运动状态，所以 X 轴电机在运动的时候，必须先用手动命令让 X 轴停止，然后再启动 Y 轴运动。

具体的手动模式下的命令规则：

手动模式动作	UO.CLASS	UO.TARGET	说明
急停	0x01	0x00	
停止	0x02	0x00	
运动	0x03	0xA1	X 轴正转运动
	0x03	0xA2	X 轴反转运动
	0x03	0xA3	Y 轴正转运动
	0x03	0xA4	Y 轴反转运动（ZA 轴分别是 0xA5—0xA8）
清零运动行程	0x04	0xA1、0xA2	清零 X 轴的行程
	0x04	0xA3、0xA4	清零 Y 轴的行程（ZA 轴分别是 0xA5—0xA8）

4.1 程序模式命令的使用

程序模式命令是用于程序模式下控制程序运行的。程序模式命令的发送格式是属于系统命令类型的数据包。

4.1.1 程序模式命令的标准格式

主机命令	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
	F4	UO.LEVEL	UO.OBJECT	UO.CLASS	UO.TARGET	UO.OCR
从机应答	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
接收正确	AA	UO.LEVEL	UO.OBJECT	0X00	F3	UO.OCR
接收出错	AB	UO.LEVEL	UO.OBJECT	0X00	F3	UO.OCR

另外，运动模式下使用查询程序执行状态的命令时还会返回系统数据类型的数据包。

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
命令	等级	对象	种类	项目	参数内容 0	参数内容 1	参数内容 2	参数内容 3	校验
UD.ORDER	UD.LEVEL	UD.OBJECT	UD.CLASS	UD.TARGET	UD.PARAMETER[0]	UD.PARAMETER[1]	UD.PARAMETER[2]	UD.PARAMETER[3]	UD.OCR

具体的程序模式下的命令规则：

手动模式动作	UO.CLASS	UO.TARGET	说明
急停	0x01	0x00	等同急停按钮触发
停止	0x02	0x00	等同停止按键触发
执行程序	0x03	0x02（程序号）	启动程序 2
清零运动行程	0x04	0xA1	和手动模式的清零功能一样，详情参考手动模式
启动延时	0x05	0x01（延时秒数）	启动命令触发后延时一秒启动
程序继续执行	0x06	0x00	程序中途停止后，接收到这个命令会继续运行，等同启动键
结束程序	0x07	0x00	程序中途停止后，接收到这个命令会结束运行，等同停止键
单步运行和调试模式	0x08	0x00	程序运行时，使用调试或者单步运行的方式
不使用单步运行和调试模式	0x09	0x00	程序运行时，不使用调试或者单步运行的方式
查询程序执行状态	0x0A	0x00	查询程序号，执行第几条指令，运行情况

举例：主机控制序号为 3 的从机自动执行程序 2，程序执行前先延时 2 秒，整个通信过程如下：

主机命令 1: 0x5A 0xA5 0x0A 0xAC 0x01 0x03 0xAC 0xAC 0x08 说明：先查询从机是否正常存在

从机 : 0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xAC 0x59

主机命令 2: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x05 0x02 0x9B 说明：先设置好启动延时为 2 秒（0x02 代表 2 秒）

从机 : 0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xF4 0xA1

主机命令 3: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x03 0x02 0x9B 说明：启动程序 2

从机 : 0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xF4 0xA1

以上 3 条命令收到正确应答数据包后，从机在延时 2 秒就执行程序 2 的程序内容，在程序执行期间，主机如果需要获取从机的程序执行情况，可以发送查询程序执行状态命令（0x0A）来读取数据。如下通信情况：

主机命令 3: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x0A 0x00 0x02 说明：查询从机 3 的程序执行情况

从机 : 0x5A 0xA5 0x0B 0xF4 0x00 0x03 0x01 0x02 0x08 0x01 0x00 0x00 0x02

从机发送回来的数据包是属于系统数据类型的数据包，所以首先说明一下这个数据包中每个参数的代表的意义，由下表可以知道，主机命令 3 得到的从机数据是：从机正在执行程序，程序号是 2，UD.PARAMETER[1]--UD.PARAMETER[0]的十六进制值为 0x0108，十进制是 264，就是执行程序 2 的第 264 条指令。

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
命令	等级	对象	种类	项目	参数内容 0	参数内容 1	参数内容 2	参数内容 3	校验
UD.ORDER	UD.LEVEL	UD.OBJECT	UD.CLASS	UD.TARGET	UD.PARAMETER[0]	UD.PARAMETER[1]	UD.PARAMETER[2]	UD.PARAMETER[3]	UD.OCR
命令	从机等级	从机序号	程序状态	程序号	命令序号 0	命令序号 1	预留	预留	校验码

			程序状态不同的值代表程序处于不同的执行情况	从机在执行程序 2, 那么 UD.TARGET=0x02	UD.PARAMETER[1] 和 UD.PARAMETER[0]合并为一个整体, 数值就是程序正在执行的指令序号			
--	--	--	-----------------------	------------------------------	---	--	--	--

UD.CLASS	0x00	0x01	0x02	0x03	0x04	0x05	
程序状态	空闲等待启动	程序正在运行	程序运行过程中, (1) 停止键触发后处于等待, 可以继续执行或者退出程序, (2) 调试模式或者单步运行时, 等待执行下一条指令	急停 (中断), 这个状态不同 0x02 的停止, 这个是突发急停, 停止后可以结束程序或继续运行			

根据上面的数据包, 再举例说明其他几个可能的状况。

举例: 主机控制序号为 3 的从机停止, 整个通信过程如下:

主机命令 1: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x0A 0x00 0x02 说明: 查询从机 3 的正在执行程序 2 第 264 条指令

从机 : 0x5A 0xA5 0x0B 0xF4 0x00 0x03 0x01 0x02 0x08 x01 0x00 0x00 0x02

主机命令 2: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x02 0x00 0xFA 说明: 停止从机 3 的程序执行, 进入等待状态

从机 : 0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xF4 0x08

主机命令 3: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x0A 0x00 0x02 说明: 查询从机 3 的状态, 已经停止等待
 从机 : 0x5A 0xA5 0x0B 0xF4 0x00 0x03 0x02 0x02 0x08 x01 0x00 0x00 0x04

主机发送完命令 3 后, 并且收到正确应答, 表明从机 3 已经停止, 并且正在等待主机下一步的指示, 如果主机想程序继续运行, 可以发送下面的主机命令 4, 如果结束程序的运行, 可以发送主机命令 5。

主机命令 4: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x06 0x00 0x02 说明: 继续执行程序
 从机 : 0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xF4 0x08

主机命令 4: 0x5A 0xA5 0x0A 0xF4 0x01 0x03 0x07 0x00 0x03 说明: 结束程序的执行
 从机 : 0x5A 0xA5 0x0C 0xAA 0x00 0x03 0x00 0xF4 0x08

5.1 扩展端口命令的使用

扩展端口命令是预留的命令, 用于从机需要单独扩展端口的时候使用。

6.1 系统数据类型的数据包

系统数据类型数据包主要用于两个使用场合: (1) 主机查询从机的各项参数时, 从机反馈相应的查询数据, 返回数据的时候使用的就是系统数据类型数据包, (2) 主机主动修改从机各项参数的时候, 把需要修改的参数值打包成系统数据类型数据包的方式, 通过总线发送给从机, 从机接收到后, 就自动修改对应的参数。

系统数据类型数据包的结构如下表。而数据包的标识是 0x0B

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
命令	等级	对象	种类	项目	参数内容 0	参数内容 1	参数内容 2	参数内容 3	校验
UD.ORDER	UD.LEVEL	UD.OBJECT	UD.CLASS	UD.TARGET	UD.PARAMETER[0]	UD.PARAMETER[1]	UD.PARAMETER[2]	UD.PARAMETER[3]	UD.OCR

在使用系统类型数据包的时候，有几个特点可以作为分析是主机发出的修改参数用于还是从机应答是返回的数据包：

- 1) UD.LEVEL：如果是从机发出，这个值为 0，其他值都是主机设备发出的修改参数的数据包；
- 2) UD.OBJECT：在系统数据类型的数据包中，这个值都只会是从机的序号。

系统数据类型数据包可以修改系统以下参数：

系统数据包	作用	应答数据	作用	应答类型	
0XF0	修改系统多个参数	0XAA(0XBB)	返回成功应答或者失败应答	应答	
0XF7	修改电机的速度参数（最低速度，最高速度等）	0XAA(0XBB)	返回成功应答或者失败应答	应答	
0XF6	修改电机的配置参数（细分、步进角，导程）	0XAA(0XBB)	返回成功应答或者失败应答	应答	
0XF8	修改从机的输出端口状态	0XAA(0XBB)	返回成功应答或者失败应答	应答	
0XF2	修改电机对应的行程的坐标值	0XAA(0XBB)	返回成功应答或者失败应答	应答	

从上面的表格可以看到，数据包都是主机发出的，都是修改从机的参数，所以出现一种结果，只要从机是连接在总线上，并且是主机需要操作的从机（目标从机），那么这个从机都只能被动的接收数据包，并且根据接收的情况作出应答，如果接收失败，就返回失败应答数据包，如果接收成功，就返回接收成功的应答数据包，并且执行数据包的修改内容。（收到接收失败的应答数据包，证明从机是收到系统数据类型数据包的全部内容，失败的原因只有是校验失败，就是接收到的数据校验的结果跟校验码不一致，这个问题通常是数据包在总线上传输的时候受到干扰，“失真”了）

无论是命令类型数据包还是数据类型数据包，在总线上传输的过程中，不能避免的是干扰，除了需要注意和提高总线的传输环境，还需要有一定的软件措施，来提高通信质量以及处理通信失败的应对方案。

在通信的时候，主机是处于主动地位，所以主机可以对失败应答作出处理，而从机只要老老实实，数据有问题就返回失败应答。主机对于失败应答，都是忠于一个可能--目标从机是存在，主机可以采取一个纠错机制，对于接收到失败应答，可以再重复发送有限次数的数据包，在设定的次数条件内只要收到成功应答，就认为这次的通信是成功的。当然，在有限次数的条件下，收到的都是失败应答，我们就要思考是总线网络有问题还是目标从机根本不存在（尝试检查一下主机的目标从机和连接在总线上的从机的序号是否一致，不一致就是白忙活了）

6.2 系统数据类型数据包修改系统参数

通过系统数据类型数据包，主机可以修改从机的系统相关参数，首先针对系统数据类型数据包来做一个对比分析，如下表。

数据来源	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
	命令	等级	对象	种类	项目	参数内容 0	参数内容 1	参数内容 2	参数内容 3	校验
	UD.ORDE R	UD.LEVEL	UD.OBJEC T	UD.CLASS	UD.TARG ET	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
主机	0xF1	0x01	0x03	0xE2	0xA1	0xC8	0x00	0x00	0x00	0x40
从机	0xF1	0x00	0x03	0xE2	0xA1	0xC8	0x00	0x00	0x00	0x3F

上面的数据来源于主机和从机，数据的差异就在于 UD.LEVEL，主机的 LEVEL 等于 1，收据的接收对象 UD.OBJECT=0x03，也就是序号为 3 的分机；而来源于从机的数据中，LEVEL=0，UD.OBJECT=0x03，意思是告诉总线上的主机，数据是来自序号为 3 的从机。同样的命令 UD.ORDER=0xF1，由主机发出就是修改从机的相应参数，由分机发出就是告诉主机相应的信息。

6.2.1 修改从机的电机行程值 F1（E1）

每个电机的行程，在控制系统内部都是以脉冲的方式来记录，因为电机位置与脉冲个数直接相关，没有经过任何的数学运行，因而没有数值上的损失，也不会因为四舍五入，加减乘除等运算后造成误差。在修改数值的时候也需要把原来长度的单位先转化为脉冲个数。

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x01	0x03	0xE1	0xA1	0x0C	0x10	0x00	0x00	0x93
从机应答	AA	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAA	0x00	0x03	0x00	0xF1	0x9E				

出错应答	AB	UO.LEVEL	UO.OBJECT	0X00	UO.ORDER	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机位置包含几个关键参数，如电机处于正半轴还是负半轴，也就是说行程值是正还是负，所以通过 UD.TARGET 这个值可以指定修改那个参数值。

UD. TARGET	行程值	行程方向	
X 轴电机	0xA1	0xA2 (=0 正半轴, =1 负半轴)	
Y 轴电机	0xB1	0xB2	
Z 轴电机	0xC1	0xC2	
A 轴电机	0xD1	0xD2	

举例：主机修改序号为 3 的从机设备上 X 轴的位置信息，通信过程如下：

主机命令 1: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE1 0xA1 0x0C 0x10 0x00 0x00 0x93

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 2: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE1 0xA2 0x01 0x00 0x00 0x00 0x79

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的 X 轴电机位置，命令 1 的数据是 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x0000010C，十进制的数值为 268，意思是电机的位置处于行走了 268 个脉冲，具体长度值需要根据整个系统的参数计算出来；命令 2 的数据是 UD.PARAMETER[0]，16 进制数据为 0x01，电机处于负半轴，行程值符号为负。

如果要计算具体的位置值，举个例子说明一下，X 轴电机控制一个丝杆的电动滑台，滑台的丝杆导程是 5 毫米，电机是 1.8°（电机转一圈需要 200 个脉冲， $360^\circ \div 1.8^\circ = 200$ ），驱动器是 16 细分，这些参数可以分析出电机转一圈需要 $200 \times 16 = 3200$ 个脉冲（有的朋友称为 3200 当量），而电机转一圈滑台移动 5 毫米。

现在主机修改了从机的位置数据，X 轴电机处于负半轴，移动了 268 个脉冲，计算得出的数值是 0.41875，显示的时候只保留三位小数，处于负半轴所以显示为负值。（ $0.41875 = 268 \div 3200 \times 5$ ）

6.2.2 修改从机的电机速度值 F1 (E2)

每个电机的速度，在控制系统内部都是以频率赫兹为单位的方式来记录，电机转动的时候最基本的计算单位是频率，而根据整个运动系统的参数，我们可以演变出其他的速度表达方式，如多少转每分钟、多少毫米每秒。

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x01	0x03	0xE2	0xA1	0xC8	0x00	0x00	0x00	0x40
从机应答	AA	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAA	0x00	0x03	0x00	0xF1	0x9E				
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机速度包含几个关键参数，如电机的最低速度，最高速度，加速时间，手动速度等，所以通过 UD.TARGET 这个值可以指定修改那个参数值。

UO. TARGET	最低速度	最高速度	加速时间	手动速度	
X 轴电机	0xA1	0xA2	0xA3	0xA4	
Y 轴电机	0xB1	0xB2	0xB3	0xB4	
Z 轴电机	0xC1	0xC2	0xC3	0xC4	
A 轴电机	0xD1	0xD2	0xD3	0xD4	

举例：主机修改序号为 3 的从机设备上 X 轴的最低速度信息，通信过程如下：

主机命令 1: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE2 0xA1 0x40 0x01 0x00 0x00 0xB9

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 2: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE2 0xA2 0x80 0x0C 0x00 0x00 0x05

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 3: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE2 0xA3 0x05 0x00 0x00 0x00 0x7F

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的 X 轴电机速度, 命令 1 修改的最低速度, 数据 UD.PARAMETER[3]- UD.PARAMETER[0], 组合成 32 位的 16 进制数据为 0x00000140, 十进制的数值为 320, 意思是电机的最低速度是 320 赫兹; 命令 2 修改的最高速度, 数据 UD.PARAMETER[3]- UD.PARAMETER[0], 组合成 32 位的 16 进制数据为 0x00000C80, 十进制的数值为 3200, 意思是电机的最高速度是 3200 赫兹; 命令 3 修改的加速时间, 数据 UD.PARAMETER[3]- UD.PARAMETER[0], 组合成 32 位的 16 进制数据为 0x00000005, 十进制的数值为 5, 意思是电机从最低速度加速到最高速度, 需要 0.5 秒的时间。

关于单位的转换, 可以参考一下例子。X 轴电机连接在一个送料的滚轮上, 滚轮的外圆的圆周长是 300 毫米, 电机是 1.8° , 驱动是 16 细分, 按照上面的最低速度是 320 赫兹, 最高速度是 3200 赫兹, 可以得出其他单位的速度方式, 送料的最低速度 30 毫米/秒, 最高速度为 300 毫米/秒。

计算方式如下:

- 电机是 1.8° , 驱动是 16 细分, 转一圈需要 3200 个脉冲;
- 最低速度是 320 赫兹, 也就是 1 秒产生 320 个脉冲, 电机一秒转 0.1 圈, 所以外圆转动了 0.1×300 毫米=30 毫米, 速度为 30 毫米/秒
- 最低速度是 3200 赫兹, 也就是 1 秒产生 3200 个脉冲, 电机一秒转 1 圈, 所以外圆转动了 300 毫米, 速度为 300 毫米/秒

6.2.3 修改从机的电机参数 F1 (E3)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x01	0x03	0xE3	0xA1	0xC8	0x00	0x00	0x00	0x40
从机应答	AA	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAA	0x00	0x03	0x00	0xF1	0x9E				

出错应答	AB	UO.LEVEL	UO.OBJECT	0X00	UO.ORDER	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机和驱动器参数包含几个关键参数，如电机的步进角，驱动器细分，导程，所以通过 UD.TARGET 这个值可以指定修改对应参数。

UO. TARGET	电机步进角	驱动器细分	导程		
X 轴电机	0xA1	0xA2	0xA3		
Y 轴电机	0xB1	0xB2	0xB3		
Z 轴电机	0xC1	0xC2	0xC3		
A 轴电机	0xD1	0xD2	0xD3		

举例：主机修改序号为 3 的从机设备上 X 轴的最低速度信息，通信过程如下：

主机命令 1: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE3 0xA1 0x2C 0x01 0x00 0x00 0xA6

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 2: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE3 0xA2 0x10 0x00 0x00 0x00 0x8A

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 3: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE3 0xA3 0x88 0x13 0x00 0x00 0x16

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的 X 轴电机参数，命令 1 修改的电机的步进角，数据 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x0000012C，十进制的数值为 300，意思是电机的步进角是 1.2° ($360^\circ \div 1.2^\circ = 300$)；命令 2 修改的驱动器细分，数据 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00000010，十进制的数值为 16，意思是电机的驱动器细分是 16；命令 3 修改的是导程，数据 UD.PARAMETER[3]- UD.PARAMETER[0]，组合成 32 位的 16 进制数据为 0x00001388，十进制的数值为 5000，导程的计算单位是 0.001 毫米，所以修改后的导程是 $5000 \times 0.001 = 5$ 毫米。

综合上面读取回来的数据，电机是 1.2° ，驱动器细分是 16，所以电机转一圈需要 4800 个脉冲；导程是 5 毫米，如果是丝杆带动的滑台，也就是说电机转一圈，滑台移动了 5 毫米。

6.2.4 修改从机的输出端口状态 F1 (E4)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x01	0x03	0xE4	0x00	0x88	0x00	0x00	0x00	0x61
从机应答	AA	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAA	0x00	0x03	0x00	0xF1	0x9E				
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

同时，由于电机和驱动器参数包含几个关键参数，如电机的步进角，驱动器细分，导程，所以通过 UD.TARGET 这个值可以指定修改对应参数。

UD.PARAM ETER[0]	输出端口 P1	输出端口 P2	输出端口 P3	输出端口 P4	输出端口 P5	输出端口 P6	输出端口 P7	输出端口 P8	端口 P1—P8
开 (ON)	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	0x9F
关 (OFF)	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90

举例：主机修改序号为 3 的从机设备上 X 轴的最低速度信息，通信过程如下：

主机命令 1: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE4 0x00 0x22 0x00 0x00 0x00 0xFB 说明：打开输出端口 P2

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 2: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE4 0x00 0x88 0x00 0x00 0x00 0x61 说明：打开输出端口 P8

从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 3: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE4 0x00 0x20 0x00 0x00 0x00 0xF9 说明: 关闭输出端口 P2
 从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 4: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE4 0x00 0x9F 0x00 0x00 0x00 0x78 说明: 打开全部输出端口
 从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机命令 5: 0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE4 0x00 0x90 0x00 0x00 0x00 0x69 说明: 关闭全部输出端口
 从机 : 0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

6.2.5 修改从机的其他系统参数 F1 (E5)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
主机命令	F1	UD.LELVE	UD.OBJEC T	UD.CLASS	UD.TARGE T	UD.PARA METER[0]	UD.PARA METER[1]	UD.PARA METER[2]	UD.PARA METER[3]	UD.OCR
	0xF1	0x01	0x03	0xE5	0xC1	0x01	0x00	0x00	0x00	0x9C
从机应答	AA	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAA	0x00	0x03	0x00	0xF1	0x9E				
出错应答	AB	UO.LEVEL	UO.OBJEC T	0X00	UO.ORDE R	UO.OCR				
	0xAB	0x00	0x03	0x00	0xF1	0x9F				

系统参数还包含其他多个项目, 如电机使能、输入设置、刹车设置等, 所以通过 UD.TARGET 这个值可以指定修改对应参数。

UD. TARGET	0xC1	0xC2	0xC3	0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xCB
参数	电机使能	输入设置	刹车设置	声音设置	读取密码	密码设置	启动设置	系统信息	脉冲方式	恢复出厂	从机序号

举例 1：主机修改序号为 3 的从机设备上的电机使能信息，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xC1 0x01 0x00 0x00 0x00 0x38 说明：修改使能

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的电机使能信息，命令发送的数据 UD.PARAMETER[0]，16 进制数据为 0x01，二进制的数值为 00000001，从低位向高位分别代表 XYZA 轴的使能状态，现在最低位是 1，也就是 X 轴是一直使能，YZA 都是没有使能，电机处于可以自由旋转的状态。

举例 2：主机修改序号为 3 的从机设备的输入设置信息，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xC2 0x06 0x07 0x08 0x00 0xB1 说明：修改输入设置

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的输入设置，命令发送的数据 UD.PARAMETER[0]=0x06，意思是系统的启动键是 K6，UD.PARAMETER[1]=0x07，意思是系统的停止键是 K7，UD.PARAMETER[2]=0x08，意思是系统的暂停键是 K8，如果 UD.PARAMETER[2]=0x00，也就是没有设置外部的暂停键功能，暂停键只能使用 K8，启动和停止键可以选用外部输入端口的 K1—K7。

举例 3：主机修改序号为 3 的从机的电机刹车设置信息，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xC3 0x0C 0x00 0x00 0x00 0xA9 说明：修改刹车设置

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的电机刹车信息，命令发送的数据 UD.PARAMETER[0]，16 进制数据为 0x0C，二进制的数值为 00001100，从低位向高位分别代表 XYZA 轴的刹车状态，现在最低位是 1100，也就是 X 轴和 Y 轴是不带刹车功能，ZA 都是带刹车。（刹车功能是在手动模式或者直线命令的时候，电机在停止前带有减速过程，如果不带刹车，停止的时候，电机是立刻停止，机器会有一定的抖动）

举例 4：主机修改序号为 3 的从机系统声音设置信息，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xC4 0x01 0x00 0x00 0x00 0x9F 说明：声音设置

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的声音设置，命令发送的数据 UD.PARAMETER[0]=0x01，意思是系统的声音功能打开，电机转动的时候有模拟电机转动的声音输出，如果 UD.PARAMETER[0]=0x00，系统的声音功能关闭。

举例 5：主机修改序号为 3 的从机启动设置信息，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xC7 0x01 0x00 0x00 0x00 0xA2 说明：修改启动设置

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的启动设置，命令发送的数据 UD.PARAMETER[0]=0x01，意思是系统在上电的时候自动运行，UD.PARAMETER[0]=0x00，系统上电的时候是不会自动运行。

举例 6：主机修改序号为 3 的从机的电机脉冲方式，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xC9 0x01 0x00 0x00 0x00 0xA4 说明：修改脉冲方式

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的电机脉冲方式，命令发送的数据 UD.PARAMETER[0]=0x01，16 进制数据为 0x01，二进制的数值为 00000001，从低位向高位分别代表 XYZA 轴的脉冲方式，现在最低位是 1，也就是 X 轴的脉冲输出方式是双脉冲，YZA 轴的脉冲输出方式是单脉冲（脉冲+方向）。

注意：单脉冲（脉冲+方向）；双脉冲（正转脉冲+反转脉冲）

举例 7：主机修改序号为 3 的从机，让从机的系统参数恢复到出厂状态，通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xCA 0x00 0x00 0x00 0x00 0xA4 说明：恢复出厂状态

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

只要收到恢复出厂命令，从机就会把系统参数全部恢复到出厂值，参数包括本章节系统数据类型数据包可以修改的参数（电机行程、电机速度、电机参数、输入输出、其他参数），命令启动后，大概过 10 秒后，把从机断电重新上电才能正常工作。

举例 8：主机修改序号为 3 的从机的序号，重新定义从机的序号（分机号）通信过程如下：

主机：0x5A 0xA5 0x0B 0xF1 0x01 0x03 0xE5 0xCB 0x06 0x00 0x00 0x00 0xAB 说明：修改启动设置

从机：0x5A 0xA5 0x0A 0xAA 0x00 0x03 0x00 0xF1 0x9E

主机修改从机 3 的序号，命令发送的数据 UD.PARAMETER[0]=0x06，意思是主机把现在正在连接的从机 3 的序号改为 6，也就是说主机下次要连接这个从机，必须使用序号 6。

7.1 产品目录