

舵机SCS通信协议

目录

- [舵机SCS通信协议](#)
- [1 通信协议概要](#)
- [2 指令帧](#)
- [3 应答帧](#)
- [4 指令类型](#)
 - [4.1 查询状态指令PING](#)
 - [4.2 读指令READ DATA](#)
 - [4.3 写指令WRITE DATA](#)
 - [4.4 异步写指令REG WRITE](#)
 - [4.5 执行异步写指令ACTION](#)
 - [4.6 同步写指令SYNC WRITE](#)
 - [4.7 同步读指令SYNC READ](#)
 - [4.8 状态重置指令RESET](#)
 - [4.9 位置校准指令](#)
 - [4.10 参数恢复指令](#)
 - [4.11 参数备份指令](#)
 - [4.12 重启指令](#)

1 通信协议概要

通讯电平采用兼容高速通信的TTL电平方式与具有较强抗干扰能力的RS485方式，通讯仍是采用异步双工，发送与接收信号是异步处理。

控制器和舵机之间采用问答方式通信，控制器发出指令帧，舵机返回应答帧。

一个总线控制网络中允许有多个舵机，所以每个舵机都分配有一个网络内唯一的ID号。控制器发出的控制指令中包含ID信息，只有匹配上ID号的舵机才能完整接收这条指令，并返回应答信息。

通信方式为串行异步方式，一帧数据分为1位起始位，8位数据位和1位停止位，无奇偶校验位，共10位。

内存表的部分参数采用两个字节的值域时，两个字节顺序根据舵机型号区分，电位器类型舵机为大端格式（高字节在前低字节在后），磁编码类型舵机为小端格式（低字节在前高字节在后）。每款舵机因功能略有不同，因此实际控制时请参照具体型号的内存表为准。

2 指令帧

字头	ID号	数据长度	指令	参数	校验和
0xFF 0xFF	ID	Length	Instruction	Parameter1 ... Parameter N	Check Sum

- 字头：连续收到两个0xFF ,表示有数据包到达。
ID号：每个舵机都有一个ID号。ID号范围0~253,转换为十六进制0x00~0xFD。
- 广播ID：ID号254为广播ID,若控制器发出的ID号为254(0xFE)，所有的舵机均接收指令，除PING指令外其它指令均不返回应答信息(多个舵机连接在总线上不能使用广播PING指令)。
- 数据长度：等于待发送的参数N加上2，即“N+2”。
- 指令：数据包操作功能码，详见1.3指令类型。
- 参数：除指令另外需要补充的控制信息，参数最大支持双字节参数表示一个内存值，字节顺序参考舵机使用手册内存控制表(不同型号舵机字节顺序不一样)。
- 校验和：校验和Check Sum，计算方法如下
 $Check\ Sum = \sim (ID + Length + Instruction + Parameter1 + \cdots Parameter\ N)$ 若括号内的计算和超出255, 则取最低的一个字节，“~”表示取反。

3 应答帧

字头	ID号	数据长度	状态码	参数	校验和
0xFF 0xFF	ID	Length	ERROR	Parameter1 ... Parameter N	Check Sum

返回的应答帧包含舵机的当前状态ERROR，若舵机当前工作状态不正常，会通过这个字节反映出来（各状态所代表意义详见手册内存控制表），若ERROR为0，则舵机无报错信息。

4 指令类型

指令	功能	值	参数长度
PING（查询）	查询工作状态	0x01	0
READ DATA（读）	查询控制表里的数据	0x02	2
WRITE DATA（写）	往控制表里写入数据	0x03	不小于2
REGWRITE DATA（异步写）	类似于WRITE DATA，但是控制字符写入后并不马上动作，直到ACTION指令到达	0x04	不小于4
ACTION（执行异步写）	触发REG WRITE写入的动作	0x05	0
SYCNREAD DATA（同步读）	用于同时查询多个舵机	0x82	不小于3
SYCNWRITE DATA（同步写）	用于同时控制多个舵机	0x83	不小于2
RESET（状态重置）	重置舵机状态(重置舵机圈数)	0x0A	0
位置校准	当前位置重校准为指令值	0x0B	0
参数恢复指令	恢复舵机除ID号外的其它参数	0x06	0
参数备份指令	备份当前参数，用于恢复指令	0x09	0
重启指令	用于重启舵机	0x08	0

4.1 查询状态指令PING

- 功能：读取舵机的工作状态
- 长度：0x02
- 指令：0x01
- 参数：无
- PING指令使用广播地址，舵机同样返回应答信息。

例1：读取ID号为1的舵机的工作状态。

指令帧：FF FF 01 02 01 FB（按十六进制发送）

代码块

1

字头：FF FF

2

ID：01

3

长度：02

4

指令：01

5

校验和：FB

应答帧：FF FF 01 02 00 FC（十六进制显示）

代码块

- 1 字头：FF FF
- 2 ID：01
- 3 长度：02
- 4 状态：00
- 5 校验和：FC

4.2 读指令READ DATA

- 功能：从舵机内存控制表里读出数据
- 长度：0x04
- 指令：0x02
- 参数1：数据读出段的首地址
- 参数2：读取数据的长度

例2：读取ID为1的舵机的当前位置（低位字节在前，高位字节在后）。位置参数内存表地址为0X38，连续二个字节。

指令帧：FF FF 01 04 02 38 02 BE（按十六进制发送）

代码块

- 1 字头：FF FF
- 2 ID：01
- 3 长度：04
- 4 指令：02
- 5 参数：38 02（当前位置地址 读数据长度）
- 6 校验和：BE

应答帧：FF FF 01 04 00 18 05 DD（十六进制显示）

代码块

- 1 字头：FF FF
- 2 ID：01
- 3 长度：04
- 4 状态：00
- 5 参数：18 05
- 6 校验和：DD

读出两个字节数据分别是（小端结构）：低位字节L 0x18 高位字节H 0x05，两字节合成16位数据0X0518，用十进制表示当前的位置为1304。

4.3 写指令WRITE DATA

- 功能：写数据到舵机内存控制表
- 长度：N+2（N为参数长度）
- 指令：0x03
- 参数1：数据写入段的首地址
- 参数2：写入第1个数据
- 参数3：写入第2个数据
- ...
- 参数N：写入第n个数据，N=n+1

例3：使用广播ID (0xFE)把任意编号舵机的ID设置为1，在内存表里保存ID号的地址为5。

指令帧：FF FF FE 04 03 05 01 F4 (按十六进制发送)

代码块

```
1  字头： FF FF
2  ID： 01
3  长度： 04
4  指令： 03
5  参数： 05 01 (ID地址 新ID值)
6  校验和： F4
```

因为采用广播ID发送指令，所以不会有数据返回。另外内存表EPROM 具有保护锁开关，修改ID前需要把它关掉（置0），不然示例ID号掉电将不保存。详细操作请查阅具体舵机型号的内存表或操作手册。

例4：控制ID1舵机以1000步每秒的速度转动到2048的位置。内存表目标位置的首地址为0x2A，所以在地址0x2A处开始写入连续六个字节数据。

- 位置数据0x0800(2048)
- 预留数据0x0000(0)
- 速度数据0x03E8(1000)

指令帧：FF FF 01 09 03 2A 00 08 00 00 E8 03 D5（按十六进制发送）

代码块

```
1  字头： FF FF
2  ID： 01
3  长度： 09
```

```
4  指令：03
5  参数：
6  2A（首地址）
7  00 08（位置）
8  00 00（预留）
9  E8 03（速度）
10 校验和：D5
```

应答帧：FF FF 01 02 00 FC（十六进制显示）

代码块

```
1  字头：FF FF
2  ID：01
3  长度：02
4  状态：00
5  校验和：FC
```

返回工作状态为0，表示舵机正确无错接收到指令并已开始执行。发送指令包的ID使用非广播ID (0xFE)，所以指令接收完成后舵机将返回状态包。

4.4 异步写指令REG WRITE

REG WRITE指令相似于WRITE DATA，只是执行的时间不同。当收到REG WRITE指令帧时，把收到的数据储存在缓冲区备用，并把异步写标志寄存器置1。当收到ACTION指令后，储存的指令最终被执行。

- 长度：N+2（N为参数长度）
- 指令：0x04
- 参数1：数据要写入区的首地址
- 参数2：写入的第1个数据
- 参数3：写入的第2个数据
- 参数N：写入的第n个数据，N=n+1

例5：控制ID1到ID10舵机以1000每秒的速度转动到2048的位置。

代码块

```
1  ID 1：异步写指令帧：FF FF 01 09 04 2A 00 08 00 00 E8 03 D4
2  ID 1：应答帧：FF FF 01 02 00 FC
3  ID 2：异步写指令帧：FF FF 02 09 04 2A 00 08 00 00 E8 03 D3
4  ID 2：应答帧：FF FF 02 02 00 FB
5  ID 3：异步写指令帧：FF FF 03 09 04 2A 00 08 00 00 E8 03 D2
6  ID 3：应答帧：FF FF 03 02 00 FA
7  ID 4：异步写指令帧：FF FF 04 09 04 2A 00 08 00 00 E8 03 D1
```

```
8 ID 4: 应答帧: FF FF 04 02 00 F9
9 ID 5: 异步写指令帧: FF FF 05 09 04 2A 00 08 00 00 E8 03 D0
10 ID 5: 应答帧: FF FF 05 02 00 F8
11 ID 6: 异步写指令帧: FF FF 06 09 04 2A 00 08 00 00 E8 03 CF
12 ID 6: 应答帧: FF FF 06 02 00 F7
13 ID 7: 异步写指令帧: FF FF 07 09 04 2A 00 08 00 00 E8 03 CE
14 ID 7: 应答帧: FF FF 07 02 00 F6
15 ID 8: 异步写指令帧: FF FF 08 09 04 2A 00 08 00 00 E8 03 CD
16 ID 8: 应答帧: FF FF 08 02 00 F5
17 ID 9: 异步写指令帧: FF FF 09 09 04 2A 00 08 00 00 E8 03 CC
18 ID 9: 应答帧: FF FF 09 02 00 F4
19 ID10: 异步写指令帧: FF FF 0A 09 04 2A 00 08 00 00 E8 03 CB
20 ID10: 应答帧: FF FF 0A 02 00 F3
```

4.5 执行异步写指令ACTION

- 功能：触发REG WRITE指令
- 长度：0x02
- 指令：0x05
- 参数：无

1、ACTION 指令在同时控制多个舵机时非常有用。

2、在控制多个舵机时，使用ACTION指令可以使第一个和最后一个舵机同时执行各自的动作，中间无延时。

3、对多个舵机发送ACTION指令时，要用到广播ID（0xFE）因此，发送此指令不会有数据帧返回。

例6：控制ID1到ID10舵机以1000每秒的速度转动到2048的位置的异步写指令发完后，需要执行异步写指令。

代码块

```
1 指令帧: FF FF FE 02 05 FA
2 应答帧: 无
```

4.6 同步写指令SYNC WRITE

- 功能：用于同时控制多个舵机。
- ID：0xFE
- 长度：(L+1)*n+4（L：发给每个舵机的数据长度，n：舵机的个数）
- 指令：0x83
- 参数1：写入数据的首地址

- 参数2：写入的数据的长度(L)
- 参数3：第1个舵机的ID号
- 参数4：写入第1个舵机的第1个数据
- 参数5：写入第1个舵机的第2个数据
- ...
- 参数L+3：写入第1个舵机的第L个数据
- 参数L+4：第2个舵机的ID号
- 参数L+5：写入第2个舵机的第1个数据
- 参数L+6：写入第2个舵机的第2个数据
- ...
- 参数2L+4：写入第2个舵机的第L个数据
- ...

不同于REG WRITE+ACTION指令的是实时性比它更高，一条SYNC WRITE指令可一次修改多个舵机的控制表内容，而REG WRITE+ACTION指令是分步做到的。尽管如此，使用SYNC WRITE指令时，写入的数据长度和保存数据的首地址必须相同。

例7：对ID1-ID4共4个舵机首地址0x2A写入位置0x0800时间0X0000和速度0x03E8 (低位字节在前，高位字节在后)。

指令帧: FF FF FE 20 83 2A 06 01 00 08 00 00 E8 03 02 00 08 00 00 E8 03 03 00 08 00 00 E8 03 04 00 08 00 00 E8 03 58（按十六进制发送）

代码块

- 1 字头：FF FF
- 2 ID：FE
- 3 有效数据长度：20
- 4 指令：83
- 5 参数：
- 6 2A 06（首地址 数据长度）
- 7 01 00 08 00 00 E8 03（ID1舵机指令）
- 8 02 00 08 00 00 E8 03（ID2舵机指令）
- 9 03 00 08 00 00 E8 03（ID3舵机指令）
- 10 04 00 08 00 00 E8 03（ID4舵机指令）
- 11 校验和：58

4.7 同步读指令SYNC READ

- 功能：用于同时查询多个舵机。
- ID：0xFE

- 长度：n+4（n为舵机个数）
- 指令：0x82
- 参数1：读数据的首地址
- 参数2：读数据的长度
- 参数3：第1个舵机的ID号
- 参数4：第2个舵机的ID号
- ...
- 参数N：第n个舵机的ID号，N=n+2

一条SYNC READ指令可一次查询多个舵机的控制表内容，同步读指令中指定了需要查询舵机的ID，舵机返回应答包顺序按指令包中ID顺序返回，使用SYNC READ指令时，所有查询的数据长度和数据的首地址必须相同（此指令针对部分串行总线舵机开放）。

例8：查询ID1-ID2共2个舵机的当前位置、当前速度、当前负载、当前电压、当前温度（首地址0x38，共8字数据，低位字节在前，高位字节在后）。

指令帧：FF FF FE 06 82 38 08 01 02 36

代码块

```
1  字头： FF FF
2  ID： FE
3  长度： 06
4  指令： 82
5  参数：
6  38 08（数据首地址 数据长度）
7  01 02（ID01 ID02）
8  校验和： 36
```

应答帧：

代码块

```
1  ID01舵机： FF FF 01 0A 00 00 08 00 00 00 00 79 1E 55
2  ID02舵机： FF FF 02 0A 00 FF 07 00 00 00 00 77 23 53
```

应答帧可按读指令进行解码

4.8 状态重置指令RESET

- 功能：重置舵机状态（重置舵机圈数）
- 长度：0x02

- 指令：0x0A
- 参数：无

例9：复位舵机，ID号为01。

代码块

- 1 指令帧：FF FF 01 02 0A F2（按十六进制发送）
- 2 应答帧：FF FF 01 02 00 FC（十六进制显示）

4.9 位置校准指令

- 功能：当前位置重校准为设定值
- 长度：0x02或0x04
- 指令：0x0B
- 参数：无或设定值

备注：位置校准指令无参数表示当前位置校准为中位；校准指令只支持部分型号舵机，舵机型号支持情况查阅下表说明。

例10：当前位置重校准为中位。

代码块

- 1 指令帧：FF FF 01 02 0B F1（按十六进制发送）
- 2 应答帧：FF FF 01 02 00 FC（十六进制显示）

例11：当前位置重校准为1024。

指令帧：FF FF 01 04 0B 00 04 EB（按十六进制发送）

代码块

- 1 字头：FF FF
- 2 ID：01
- 3 长度：04
- 4 指令：0B
- 5 设定值：00 04（1024）
- 6 校验和：EB

应答帧：FF FF 01 02 00 FC（十六进制显示）

代码块

- 1 字头：FF FF

- 2 ID：01
- 3 长度：02
- 4 状态：00
- 5 校验和：FC

舵机型号	固件版本	生效条件	备注
SMS舵机	<=2.54	无	支持扭矩开关写128实现中位校准
STS舵机	>=3.10	无	支持无参数与有参数指令，支持扭矩开关写128实现中位校准
HLS舵机	>=3.43	无	支持无参数与有参数指令，不支持扭矩开关写128实现中位校准

4.10 参数恢复指令

- 功能：恢复舵机除ID号外其它参数
- 长度：0x02
- 指令：0x06
- 参数：无

例11：恢复舵机参数。

代码块

1 指令帧：FF FF 01 02 06 F6（按十六进制发送）

2 应答帧：FF FF 01 02 00 FC（十六进制显示）

备注：恢复舵机参数前先解锁eprom参数

4.11 参数备份指令

- 功能：参数备份（用于参数恢复指令）
- 长度：0x02
- 指令：0x09
- 参数：无

例12：备份舵机参数。

代码块

1 指令帧：FF FF 01 02 09 F3（按十六进制发送）

2 应答帧：FF FF 01 02 00 FC（十六进制显示）

备注：备份舵机参数前先解锁eprom参数

4.12 重启指令

- 功能：重启指令（重启舵机）
- 长度：0x02
- 指令：0x08
- 参数：无

例13：重启舵机。

代码块

- 1 指令帧：FF FF 01 02 08 F4（按十六进制发送）
- 2 应答帧：无（重启时间约800ms）

备注：重启舵机前，先关闭扭矩开关