



## 德晟串口总线智能舵机协议文件 (公共标准版)

使用权限：公开

协议版本：v4.01

修订日期：2017.10.19

## 一、通信协议概要

本协议文件专门为德晟智能科技开发的普通 UART-TTL 标准双线串口总线，普通 UART-TTL 单线串口总线，以及 RS485 工业串口总线系列智能舵机控制通信编写。

本协议采用半双工主从问答式通信，即舵机控制器作为主机发送读写指令给舵机，舵机作为从机根据具体情况执行相应动作或应答。一条总线上允许挂载多个舵机，且总线上每个舵机都有一个唯一的 ID (1~250)，控制器通过舵机 ID 对特定的舵机进行访问，也可以通过广播 ID (254) 向总线上的全部舵机发送数据。

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。

## 二、指令格式

### 2.1、控制器下发指令格式：

字头	ID 号	数据长度	指令类型	参数	数据校验
2 字节	1 字节	1 字节	1 字节	N 字节	1 字节

如：FF FF FE 07 03 2A 08 00 03 E8 DA

字头：0xFF 0xFF，连续收到两个 0xFF 表示有数据包到达。

ID 号：指定舵机 ID (1~250)，广播 ID (254)，其他无效。

数据长度：参数值=【ID 号+指令类型+参数】的字节总数，即=N+2。

指令类型：数据包类型，如读指令、写指令等，详见指令类型列表。

参数：指令包中需要向舵机传达的有效数据，字节数可以为 0~N。

①、无参数指令：参数字节数为 0

②、写数据指令：参数的第一个字节为需要发送数据的首地址，之后则是需要发送的数据，数据长度根据不同的地址而定。

③、读数据指令：参数字节数固定为 2 字节，第一个字节为需要读取数据的首地址，第二个字节是需要读取数据的字节数。

数据校验：采用单字节和校验方式，计算方法如下

$$\text{Checksum} = (\sim (\text{ID 号} + \text{数据长度} + \text{指令类型} + \text{参数 1} + \dots + \text{参数 N})) \& 0xFF$$

### 2.2、舵机应答包指令格式：

字头	ID 号	数据长度	舵机状态	参数	数据校验
2 字节	1 字节	1 字节	1 字节	N 字节	1 字节

如：FF F5 01 03 00 20 DB

字头：0xFF 0xF5 开始的数据包表示是由舵机应答主机的数据包。

ID 号：舵机自身的 ID 号(1~250)。

数据长度：参数值=【ID 号+指令类型+参数】的字节总数，即=N+2。

舵机状态：返回舵机当前的一些状态(详见 2.4 章节)。

参数：舵机返回给控制器的数据，参数长度可以为 0。

数据校验：采用单字节和校验方式，计算方法如下

$$\text{Checksum} = (\sim (\text{ID} + \text{数据长度} + \text{舵机状态} + \text{参数 1} + \dots + \text{参数 N})) \& 0xFF$$

## 2.3、指令类型

指令类型	指令值	参数长度	描述
PING（查询）	0x01	0	查询舵机/快速查询舵机状态
READ DATA（读）	0x02	2	查询指定地址的数据
WRITE DATA（写）	0x03	≥1	向指定地址写数据
REG WRITE（异步写）	0x04	≥2	向指定地址预写数据，等收到 ACTION 指令后才执行，主要用于控制多个舵机时能让舵机同时启动
ACTION（执行异步写）	0x05	0	触发执行 REG WRITE 指令
RESET（恢复出厂设置）	0x06	0	把寄存器恢复为出厂设定值
SYNC WRITE（同步写）	0x83	>2	用于同时向多个舵机写入不同参数

### 2.3.1、查询指令-PING

该指令不能使用广播 ID 发送。

例：控制器向 ID=1 的舵机发送查询指令：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0x01	0x02	0x01	（无）	0xFB

舵机向控制器返回应答：

字头	ID 号	数据长度	舵机状态	参数	数据校验
0xFF 0xF5	0x01	0x02	0x00	（无）	0xFC

### 2.3.2、读指令- READ DATA

该指令不能使用广播 ID 发送。

例 1：控制器向 ID=1 的舵机发送查询当前位置（0x38）指令：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0x01	0x04	0x02	0x38 0x02	0xBE

舵机向控制器返回应答：

字头	ID 号	数据长度	舵机状态	参数	数据校验
0xFF 0xF5	0x01	0x04	0x00	0x07 0xFF	0xF4

注意：本协议采用【大端模式】处理多字节数据，即高字节数据存放在前面，低字节数据存放在后面，同时在收发数据时也是先发高字节数据，例如上面的应答返回的两字节参数【0x07 0xFF】合并成的 16 位数据是 0x07FF。

### 2.3.3、写指令- WRITE DATA

例 1：控制器向所有舵机发送把 ID 重置为 1 的指令：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0xFE	0x04	0x03	0x05 0x01	0xF4

说明：指令中数据写入的数据地址是 0x05，写入的数据是 0x01。

### 2.3.4、异步写指令- REG WRITE

该指令和 WRITE DATA 写指令类似，也是向指定地址写入数据，但解析到有效数据后不会立即执行动作而是会把数据先存放在临时数据缓存中，直到收到 ACTION（执行异步写）指令后才执行，特别说明，在收到 ACTION（执行异步写）指令前新到的数据会覆盖旧数据。

该指令仅可以使用舵机 ID 发送。

**注 1：**该指令仅用于位置转动控制，即只支持地址 0x2A 的写指令。

**注 2：**该指令主要用于使多个不同 ID 的舵机可以同时启动。

例：控制器向 ID=1 的舵机发送更新目标位置到 2000，运行时间为 1000ms 的指令：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0x01	0x07	0x04	0x2A 0x07 0xD0 0x03 0xE8	0x07

说明：舵机解析指令后不会执行转动动作。

### 2.3.5、执行异步写指令- ACTION

该指令和 REG WRITE 写指令配套使用的，用于触发上一条 REG WRITE 写指令的动作。

该指令只使用广播 ID 发送。

例：控制器向 ID=1 的舵机发送更新目标位置到 2000，运行时间为 1000ms 的指令，之后向 ID=3 的舵机发送更新目标位置到 1000，运行时间为 2000ms 的指令，然后让这两个舵机同时开始转动：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0x01	0x07	0x04	0x2A 0x07 0xD0 0x03 0xE8	0x07
0xFF 0xFF	0x03	0x07	0x04	0x2A 0x03 0xE8 0x07 0xD0	0x05
0xFF 0xFF	0xFE	0x02	0x05	(无)	0xFA

说明：ID1 和 ID3 的舵机解析完第三条指令后开始按照指令内容转到目标位置，其他 ID 的舵机由于没有收到异步写指令所以不会对第三条指令有反应。

### 2.3.6、同步写指令- SYNC WRITE

该指令用于在一条指令包里同时控制多个舵机转动，使用广播 ID 发送。

该指令在参数位置和其他指令有所区别，第一个字节仍然是写入数据的地址，但第二个字节是向舵机写入的字节数（每个舵机写入的地址和长度相同），之后的参数是：舵机 IDa+舵机 IDa 的参数+舵机 IDb+舵机 IDb 的参数+……+舵机 IDx+舵机 IDx 的参数。

**注 1：**目前该指令仅用于位置转动控制，即只支持地址 0x2A 的写指令。

例：控制器向 ID=1，和 ID=3 的舵机发送更新目标位置到 2000，运行时间为 1000ms 的指令（目标位置和运行时间可以不一样）：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0xFE	0x0E	0x83	0x2A 0x04 0x01 0x07 0xD0 0x03 0xE8 0x03 0x07 0xD0 0x03 0xE8	0x8B

说明：ID1 和 ID3 舵机收到指令后立即转到 2000 位置并且转动的时间是 1000ms。

### 2.3.7、恢复出厂设置指令- RESET

该指令用于恢复舵机出厂默认参数。

例：控制器向 ID=1 的舵机发送恢复出厂默认设置指令：

字头	ID 号	数据长度	指令类型	参数	数据校验
0xFF 0xFF	0x01	0x02	0x06	(无)	0xF6

说明：舵机收到指令后立即把所有数据设置到默认值。

2.4、舵机应答包中的舵机状态

舵机把自己的一些状态在应答指令中体现，这样可以让控制器快速知道舵机是否出现异常并做出相应措施。

Bit	状态名称	状态描述
7	0	0
6	0	0
5	过载（堵转）	舵机触发保护后置 1
4	0	
3	0	
2	温度异常	温度超过指定范围置 1
1	0	
0	电压异常	电压超过指定范围置 1

### 三、命令地址控制表

注 1：本协议采用【大端模式】处理多字节数据，即高字节数据存放在低地址，低字节数据存放在高地址，同时在收发数据时也是先收发高字节数据。

命令地址	命令名称	读写	数据类型	初始值	描述	掉电保存
0x03 0x04	软件版本	读	Bytes	--	记录舵机软件版本信息，格式为 vA.B 如 v1.28 =0x011C	是
0x05	舵机 ID	读写	UInt8	1	舵机自身 ID 号，有效范围：1~250 注：254 为广播 ID	是
0x06	堵转保护时间	读写	UInt8	3	单位：/S 舵机堵转一段时间后保护，默认 3 秒	是
0x09 0x0A	最小角度限制	读写	UInt16	0	有效范围：？ 具体描述？	是
0x0B 0x0C	最大角度限制	读写	UInt16	4095	有效范围：？ 具体描述？	是
0x10 0x11	最大扭矩	读写	UInt16	800	有效范围：？ 具体描述？	是
0x12	速度调整	读写	UInt8	30	有效范围：？ 具体描述？	是
0x13	卸载条件	读写	UInt8	0	有效范围：？ 具体描述？	是
0x14 0x15	中位调整	读写	Int16	0	位置调整偏移量，正数往 4095 方向调整， 负数往 0 方向调整 有效范围：？	是
0x16 0x17	设定目标位置一	读写	UInt16	2048	设定用户自定义位置一，可以读出数据， 但写入的时候是系统自动把当前位置写入， 具体用法为： 1、先把摇臂用外力转到要设定的位置 2、发送写入指令，指令中的参数只需要 首地址即可，不需要位置参数	是
0x18 0x19	设定目标位置二	读写	UInt16	2048	同【目标位置一】	是
0x1A 0x1B	设定目标位置三	读写	UInt16	2048	同【目标位置一】	是
0x1C	舵机电机模式	读写	UInt8	--	？	？
0x1D	电机模式方向	读写	UInt8	--	？	？
0x28	扭矩开关	读写	UInt8	--	0：扭矩关闭 非 0：扭矩打开	

0x2A 0x2B	目标位置	读写	UInt16	--	有效范围：0~4095 超过该范围时则自动转成最小或最大值	
0x2C 0x2D	运行时间	读写	UInt16	0	单位：ms 该值为 0 时则以最快速度转动	
0x2E 0x2F	当前电流	读	UInt16	--	单位：mA	
0x30	锁标志(急停)	读写	UInt8	--	?	
0x38 0x39	当前位置	读	UInt16	--	有效范围：0~4095 对应目标位置	
0x3A 0x3B	运行速度	读	UInt16	--	?	
0x3C	运行目标位置一	写	--	--	运行到设定好的目标位置一 命令中只需要地址参数	
0x3D	运行目标位置二	写	--	--	运行到设定好的目标位置二 命令中只需要地址参数	
0x3E	运行目标位置三	写	--	--	运行到设定好的目标位置三 命令中只需要地址参数	
0x3F	当前温度	读	UInt8	--	单位：℃	
0x40	REG WRITE 标志	读	UInt8	--	?	
0x41 0x42	速度调整	读写	UInt16	--	?	