

UART 总线伺服舵机

RA8-U25-M

产品特点

- UART 总线通讯协议, 支持位置与状态回读
- 12-bit 绝对值位置编码器 (4,096 阶分辨率), 可任意设定原点
- 多圈角度最大控制范围 $\pm 368,640^\circ$ (1,024 圈), 具备断电角度记忆功能
- 内置梯形加减速控制算法, 实现平滑动作曲线
- 提供三种停止模式: 锁力保持 / 失锁释放 / 阻尼控制
- 集成温度、电压、堵转、功率、电流多重保护机制, 具备智能功率限制功能
- 配套可视化上位机调试软件, 支持固件升级



型号定义

R	P	8	—	U	45	H	—	M
外观	马达类型	尺寸		通讯协议	扭力范围	电压		位置编码器

外 观	R: 双轴	H: 单轴	
马达类型	X: 无刷	P: 空心杯	A/L: 铁芯
尺 寸	6: 31.5×21×27.6mm	8: 40×20×40mm	18: 63×34×47mm
通讯协议	U: UART/TTL P/A: PWM	R: RS-485	C: CAN
电 压	[-]: 7.4V	H: 12V	W: 24V
位置传感器	[-]: 电位器	M: 绝对值位置编码器(磁编码)	

订购型号

- RA8-U25-M

目 录

1	产品参数	1
1.1	基本参数	1
1.2	特性参数	1
1.3	T-N 特性曲线	2
1.4	机械过载曲线	2
2	图纸及安装说明	3
2.1	外观尺寸图	3
2.2	接口定义	3
2.3	连线说明	4
2.4	安装说明	4
3	开发与兼容性	5
4	保护功能	7
4.1	温度保护	7
4.2	电压保护	7
4.3	堵转保护	8
4.4	功率保护	8
4.5	电流保护	8
5	指令与协议	9
5.1	控制指令	9
5.2	协议格式	10
6	运动与控制指令	11
6.1	通讯检测	11
6.2	单圈角度控制	11
6.3	多圈角度控制	11
6.4	圈数重置 / 断电记忆	12
6.4.1	圈数重置	12
6.4.2	断电角度记忆	13

6.5	阻尼模式	14
6.6	停止指令	14
6.7	同步指令	14
6.8	异步指令	14
6.9	工作状态数据读取	15
6.9.1	数据读取	15
6.9.2	舵机状态	15
6.9.3	数据监控	15
6.10	设置原点	15

1 产品参数

1.1 基本参数

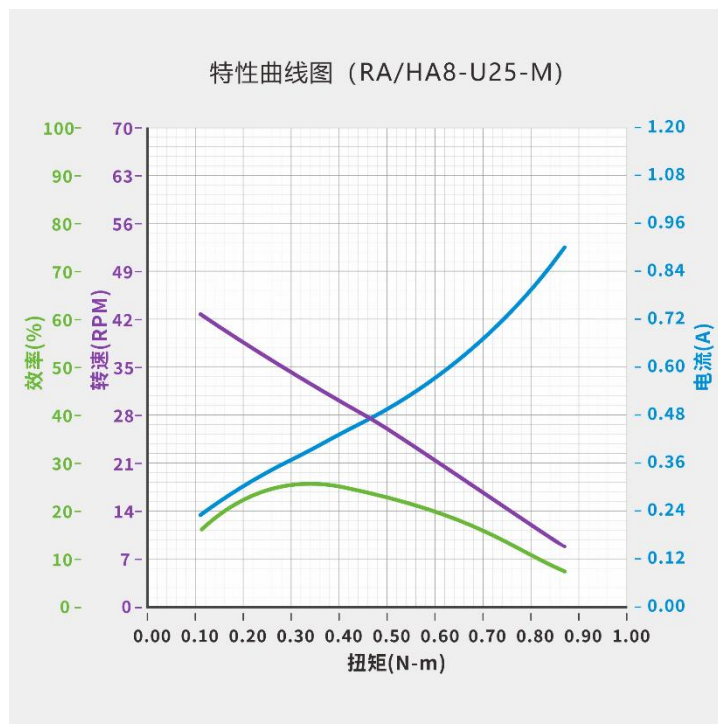
参 数	规 格
工作电压	6.0-8.4v
马达类型	铁芯马达
位置传感器	12bit 非接触式绝对值编码器（磁编码）
有效角度（行程范围）	±180°（单圈角度） ±368,640°（多圈角度）
分 辨 率	2048 阶/360°（0.176°）
处 理 器	32bit MCU
通信类型	UART/TTL 半双工
波 特 率	9,600bps~1Mbps
ID 范 围	0~254
减 速 比	273:1
输出齿规格	铜/ Ø6mm / 25T
齿轮材料	全金属铜铝组合
接口类型	PH2.0-3Pin
外壳材料	铝合金中段/上下壳工程塑胶
尺 寸	40×20×40mm
重 量	65g
工作温度	-10~60°C
工作模式	单圈角度 多圈角度 阻尼模式

1.2 特性参数

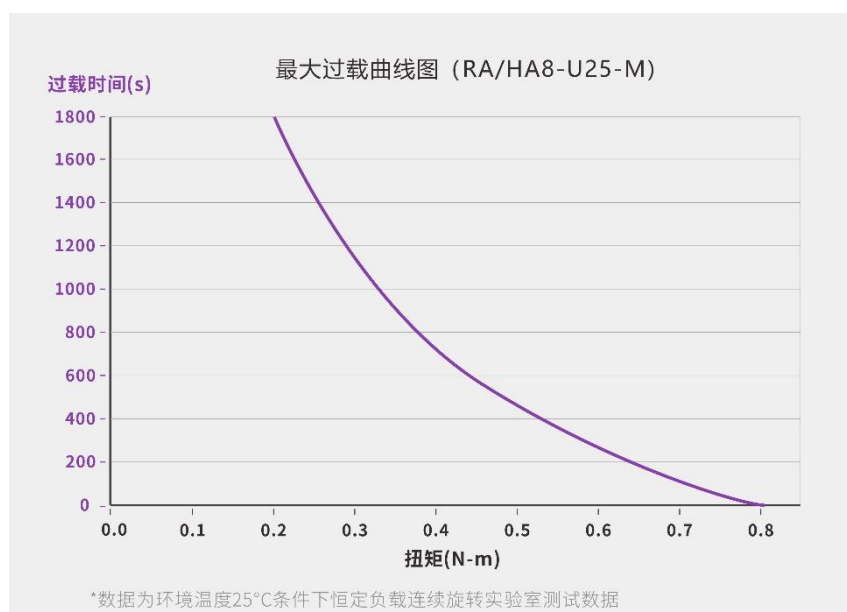
参 数	规 格(7.4v)
最大静态扭矩(堵转)	2.45N·m (25kg-cm)
最大动态扭矩	0.88N·m (9kg-cm)
额定扭矩	0.44N·m (3.4kg-cm)
额定转速	18rpm (0.556sec@60°)
空载转速	51rpm (0.198sec@60°)
空载电流	< 200mA

待机电流	< 20mA
峰值电流	3A
轴向负载	20N
径向负载	40N

1.3 T-N 特性曲线

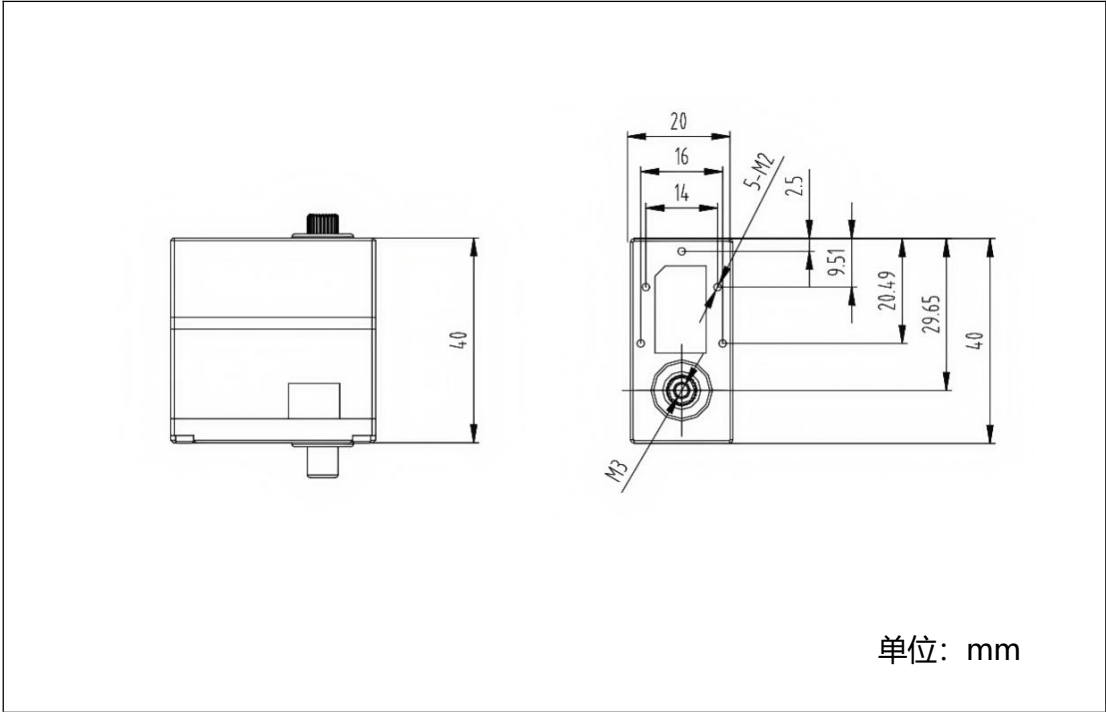


1.4 机械过载曲线

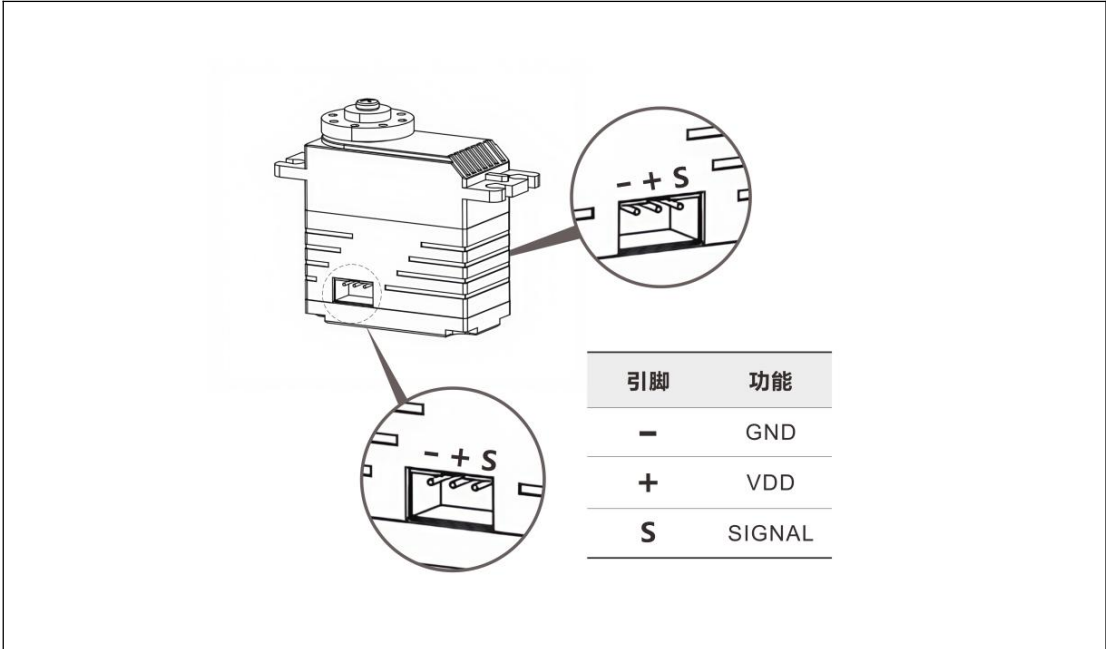


2 图纸及安装说明

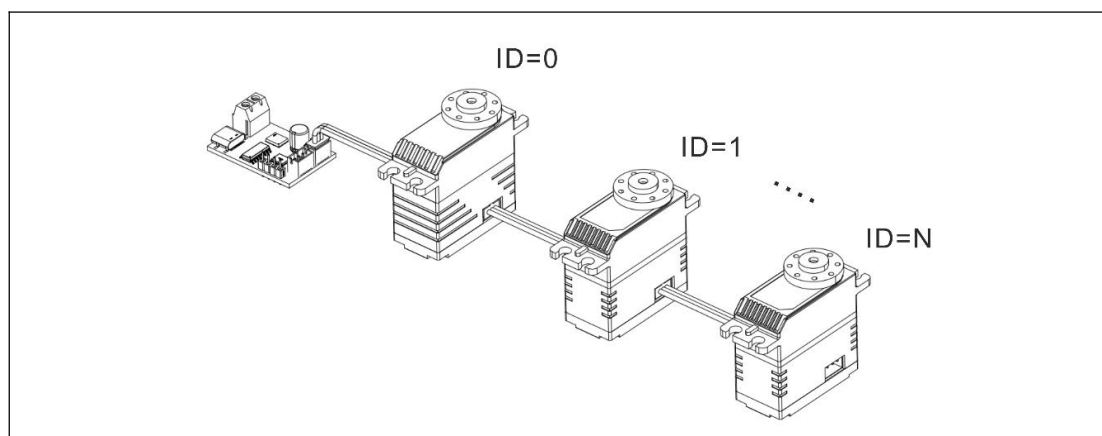
2.1 外观尺寸图



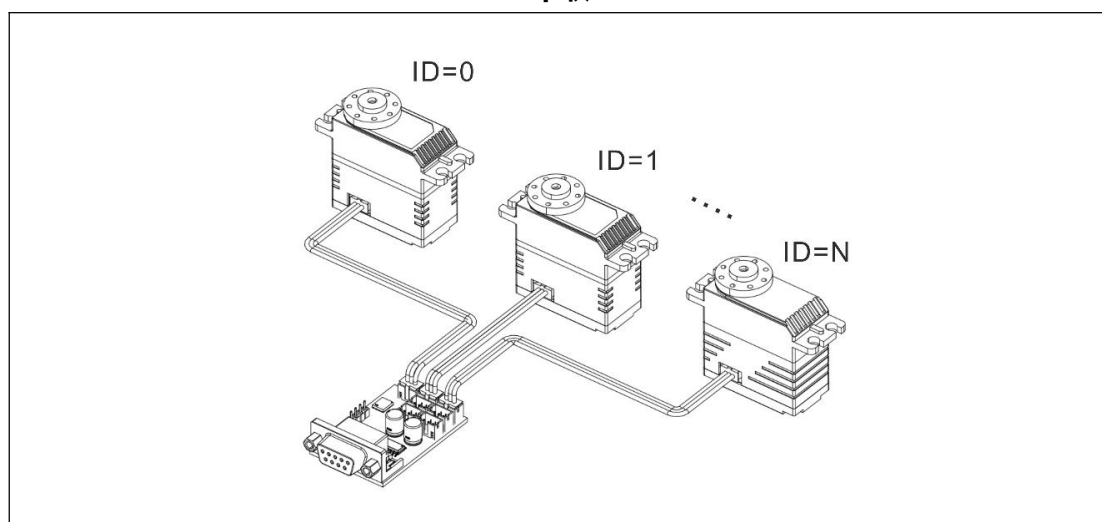
2.2 接口定义



2.3 连线说明

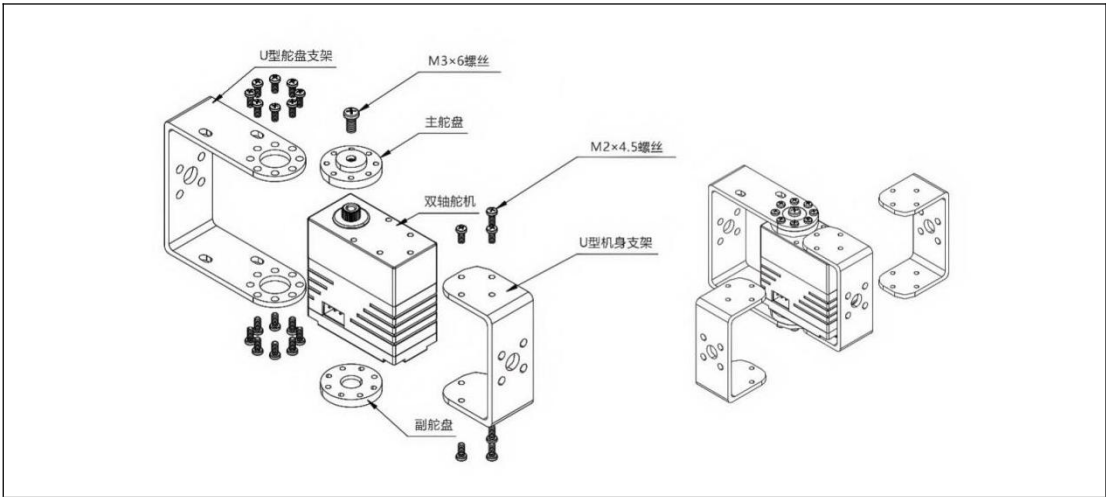


串联



并联

2.4 安装说明



3 开发与兼容性

总线伺服舵机系列采用统一的硬件平台与系统架构,不同型号间可无缝兼容, 统一支持标准协议与控制指令, 显著简化系统集成与开发流程。

为加速项目部署, 我们提供完整的 SDK(软件开发工具包), 内含丰富示例代码、驱动程序与详细技术文档, 支持多种主流开发环境与编程语言。具体支持信息详见下表。

欢迎访问官方网站 wiki.fashionstar.com.hk/zh/ 获取更多技术资料。

平台/环境/语言	兼容型号/内容
Arduino	• Uno • Mega2560 • STM32F103C8T6 Core • Mixly
ESP32	• NodeMCU-32S
STM32	• STM32F103 SDK • STM32F407 SDK
树莓派(Python)	• Pi 4B • Pi 5

编程语言	• Python • MicroPython • C++ • C#
机器人框架	• ROS2

4 保护功能

- 所有保护参数都可以通过上位机设定修改。
- 状态标志位定义如下：1 为进入保护, 0 为工作正常。

bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
指令执行中	指令执行错误	堵转保护	高压保护	低压保护	电流保护	功率保护	温度保护

⚠ 警告

任何参数的修改都可能对产品造成损害或影响其正常运行。请谨慎操作，并在更改前确保已充分评估可能的风险。

4.1 温度保护

- 通过上位机设定「温度保护」参数，超出范围将触发温度保护。
- 出厂默认保护温度为 70℃。
- 温度保护被触发时，伺服舵机自动切换至低功率维持基础运动功能。
- 可以通过查询对应的舵机状态标志位[bit7]，来判断是否进入温度保护。
- 当温度低至所设定保护温度值 10℃的时候，伺服舵机将再次自动恢复正常功率工作，同时，相应标志位重新被置 0。

4.2 电压保护

- 通过上位机设定「高压保护」和「低压保护」参数，超出范围将触发电压保护；
- 出厂默认工作电压如下，
 - 7.4v 版本：6.0-8.4v
 - 12v 版本：9.0-12.6v
 - 24v 版本：20.0-25.2v
- 电压保护被触发时，伺服舵机会自动释放锁力；
- 可以通过查询对应的舵机状态标志位[bit3]/ [bit4]，来判断是否进入电压保护。
- **必须要再次上电，且工作电压在正常范围内，才能使伺服舵机恢复工作。**

4.3 堵转保护

- 设定「堵转失锁保护」为**开**。
- 设定「功率保护值」参数, 超出范围将触发堵转保护。
- 堵转保护被触发时, 伺服舵机会自动释放锁力。
- 可以通过查询对应的舵机状态标志位[bit2], 来判断是否进入堵转保护。
- 无需断电, 通过发送停止指令, 即可恢复正常运行。

4.4 功率保护

- 设定「堵转失锁保护」为**关**。
- 设定「堵转功率上限」参数, 此参数为功率保护触发后的参考运行功率;
- 设定「功率保护值」参数, 超出范围将触发功率保护。
- 功率保护被触发时, 伺服舵机会自动降低功率, 至 **堵转功率上限** 参数值运行。
- 可以通过查询对应的舵机状态标志位[bit6], 来判断是否进入功率保护。

4.5 电流保护

- 设定「电流保护」参数, 超出范围将触发电流保护。
- 电流保护被触发时, 伺服舵机会自动释放锁力。
- 可以通过查询对应的舵机状态标志位[bit5], 来判断是否进入电流保护。
- 当工作电流低于电流保护值, 伺服舵机自动恢复工作;
- 此参数可以结合堵转或功率保护使用, 作为前两者均未触发的时候的保障。

5 指令与协议

总线舵机采用 **UART/RS485** 总线通讯协议，基于半双工异步串行通信机制，采用指令-响应的方式，实现主控与多舵机之间的控制指令下发、状态回读，并通过为每个舵机分配唯一 ID 完成总线寻址与设备识别（默认 ID = 0）。

5.1 控制指令

- 帧格式为 8 位数据位 + 1 位停止位（无奇偶校验）。
- TxD 与 RxD 不可同时工作，任一时刻仅允许一个设备发送数据，其余设备需处于接收待命状态。
- 建议连续指令的发送间隔控制在 5-10 ms。

指令名称	指令编号	响应封包类型
通讯检测	01 (0x01)	固定
简易单圈角度控制	08 (0x08)	可配置
高级单圈角度控制 (基于时间)	11 (0x0B)	可配置
高级单圈角度控制 (基于速度)	12 (0x0C)	可配置
单圈当前角度读取	10 (0x0A)	固定
简易多圈角度控制	13 (0x0D)	可配置
高级多圈角度控制 (基于时间)	14 (0x0E)	可配置
高级多圈角度控制 (基于速度)	15 (0x0F)	可配置
多圈当前角度读取	16 (0x10)	固定
重置圈数	17 (0x11)	可配置
阻尼控制	09 (0x09)	可配置
停止指令	24 (0x18)	可配置
同步指令	25 (0x19)	无
异步写入指令	18 (0x12)	无
异步执行指令	19 (0x13)	无
数据读取	03 (0x03)	固定
数据监控	22 (0x16)	固定
设置原点	23 (0x17)	可配置

5.2 协议格式

指令封包

指令封包是主控向舵机下发控制或查询命令时所使用的标准数据结构。

字节位	0~1	2	3	4~N+3	N+4
内容	请求头标识 0x12 0x4C	指令编号	内容长度N	数据内容	校验和

- header: 固定为 0x12 0x4C, 用于标识指令封包的起始位置。
- cmd_id: 本次封包的控制指令
- length: 表示后续数据内容 (content) 的字节数, 用于解析封包。
- content: 根据命令字不同, 存放控制参数 (如舵机 ID、目标角度、运动时间、功率值等)。
- checksum: 所有字节累加求和后取模 256 的结果, 用于校验数据完整性。

响应封包

响应封包是舵机接收指令封包并校验通过后, 解析参数并向主控返回执行结果与相关数据的标准数据结构。

其总体结构与指令封包一致, 仅起始位置标识位和数据内容定义有所区别。

字节位	0~1	2	3	4~N+3	N+4
内容	响应头标识 0x05 0x1C	指令编号	内容长度N	数据内容	校验和

- header: 固定为 0x05 0x1C, 用于标识响应封包的起始位置。
- cmd_id: 本次封包的控制指令
- length: 表示后续数据内容 (content) 的字节数, 用于解析封包。
- content: 根据命令字不同, 返回执行结果, 或相应数据 (如当前角度、电压、温度、版本、回读参数等)。
- checksum: 所有字节累加求和后取模 256 的结果, 用于校验数据完整性。

6 运动与控制指令

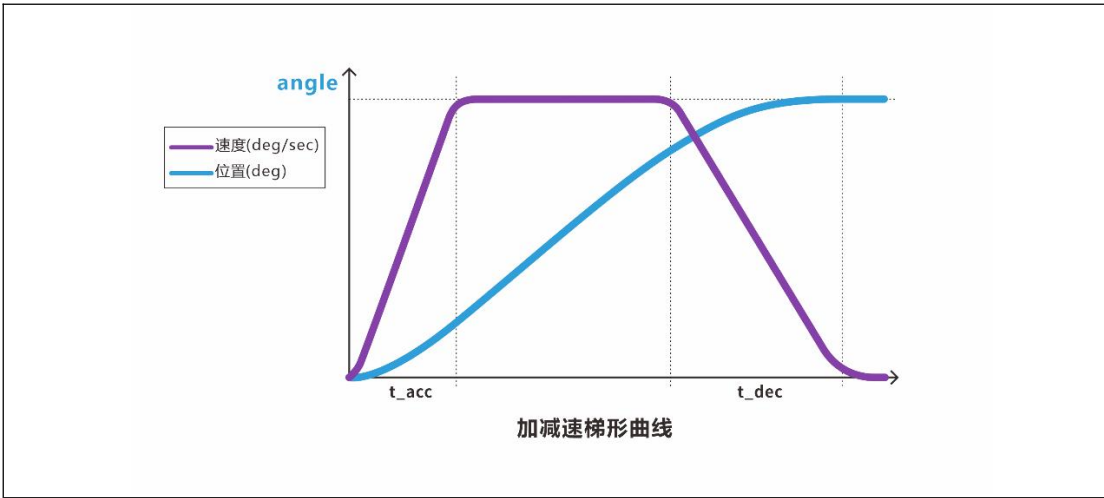
6.1 通讯检测

- 通过发送对应 ID 的 Ping 指令, 根据回包情况判断该舵机是否在线。

6.2 单圈角度控制

- 支持基于时间或速度两类控制方式, 并可通过单圈角度读取指令获取当前位置。
- 控制范围为 $\pm 180^\circ$, 最小控制精度 0.1° 。

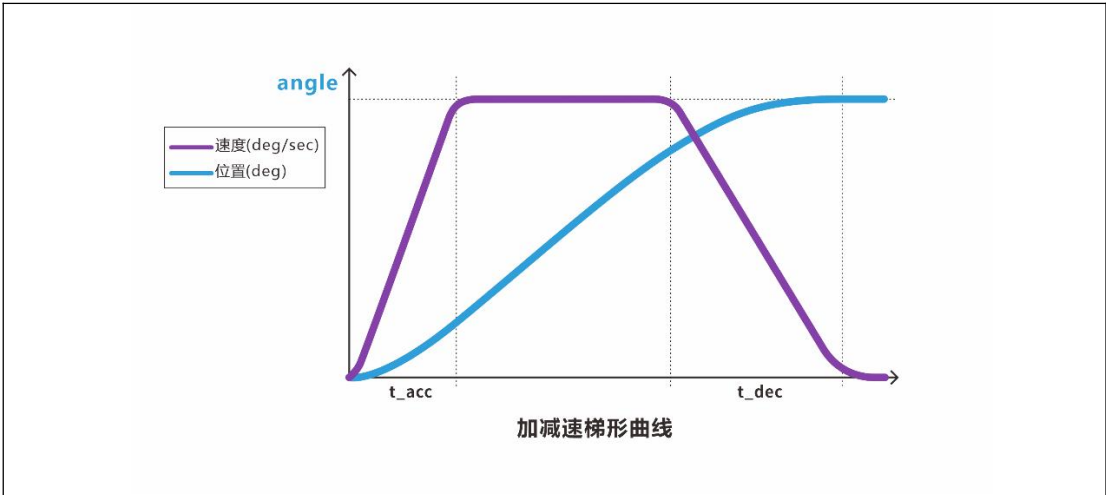
指令类型	参数
简易单圈角度控制	目标角度、运动时间、运行功率
高级单圈角度控制 (基于时间)	目标角度、运动时间、加速时间、减速时间、运行功率
高级单圈角度控制 (基于速度)	目标角度、运动速度、加速时间、减速时间、运行功率
单圈当前角度读取	单圈当前角度读取



6.3 多圈角度控制

- 支持基于时间或速度两类控制方式, 并可通过多圈角度读取指令获取当前位置。
- 控制范围为 $\pm 368,640^\circ$ ($\pm 1,024$ 圈), 最小控制精度 0.1° 。

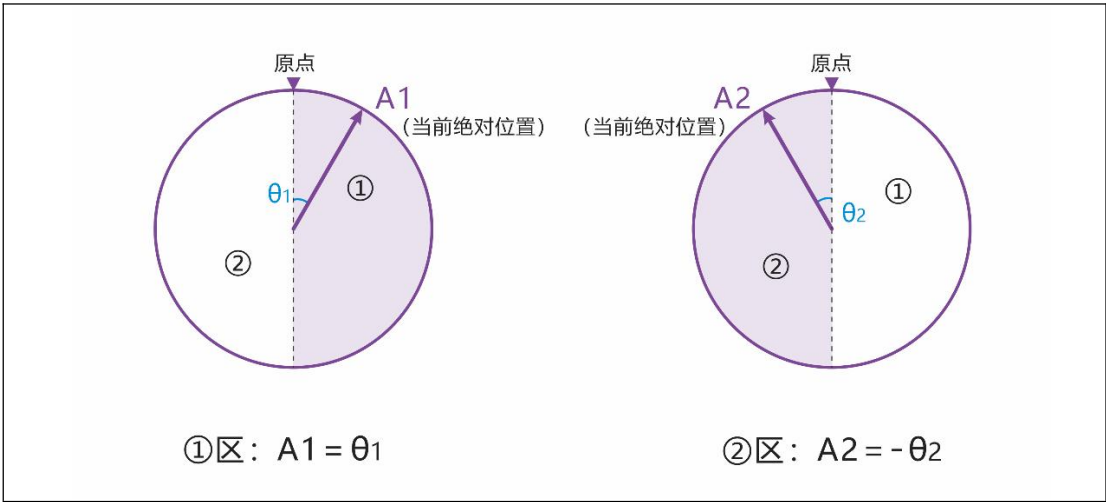
指令类型	参数
简易多圈角度控制	目标角度、运动时间、运行功率
高级多圈角度控制 (基于时间)	目标角度、运动时间、加速时间、减速时间、运行功率
高级多圈角度控制 (基于速度)	目标角度、运动速度、加速时间、减速时间、运行功率
多圈当前角度读取	响应封包 = 舵机当前角度



6.4 圈数重置 / 断电记忆

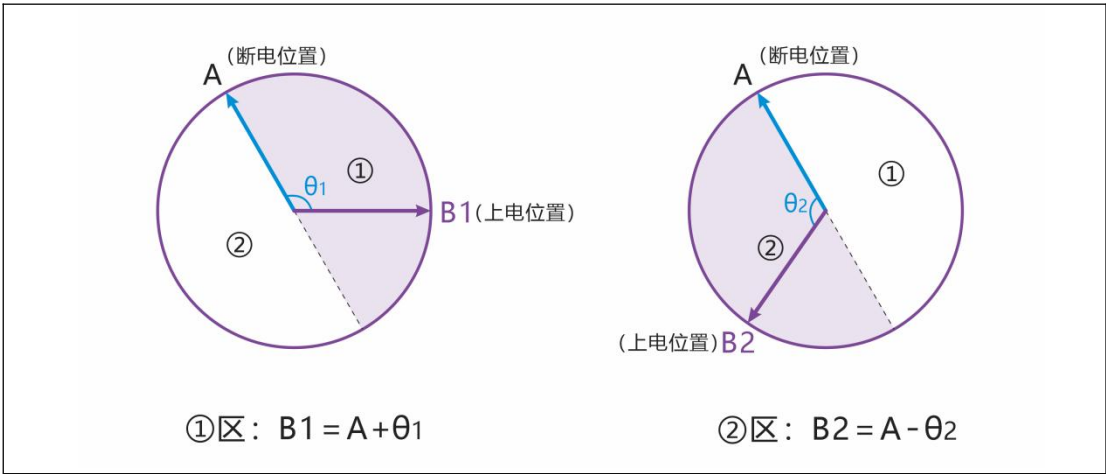
6.4.1 圈数重置

- 在舵机处于释放锁力状态时, 可通过上位机或指定指令来重设当前角度数据。
- 重设后, 舵机会以当前绝对位置来重设角度, 再次设定后的角度初始值将在 -180° 至 $+180^{\circ}$ 范围内。
例如: 如图所示, A1 点当前角度为 $6,880^{\circ}$, 重设后的角度为 θ_1 。A2 点当前角度为 $6,800^{\circ}$, 重设后的角度为 $-\theta_2$ 。



6.4.2 断电角度记忆

- 支持断电角度记忆功能。
- 断电后, 若伺服舵机的角度未发生变化, 则上电后读取的当前角度值保持不变。
例如: A 点为断电前的角度位置 $6,800^\circ$, 断电期间角度未改变, 舵机仍停留在 A 点, 则上电后读取的角度仍为 $6,800^\circ$ 。
- 断电后, 若由于外力作用导致舵机角度发生变化, 则再次上电后读取到的角度值将落在记忆角度 $\pm 180^\circ$ 的范围内。
例如: 如图所示, A 点为断电前的角度 $6,800^\circ$ 。若断电期间舵机被外力转动, 最终停在 B1 点, 则上电后读取角度为 $6,920^\circ$; 若停在 B2 点, 则读取角度为 $6,680^\circ$ 。



6.5 阻尼模式

- 允许伺服舵机在外部力作用下调整到不同的角度位置，同时保持一定的阻尼效果。
- 阻尼系数可自定义。

6.6 停止指令

- 客户可以根据不同运动控制需要，选择合适的停止指令类型，具体类型详见下表。
- 停止指令也可被用于伺服舵机在堵转保护下，恢复正常工作状态使用。
- 当伺服舵机在失锁状态下，发送“保持锁力”指令，可使其从当前位置重建锁力。

停止指令类型	动作模式
失去锁力	舵机立即停止运动，并 释放 锁力。
保持锁力	舵机立即停止运动，并 维持 锁力，或在无锁力状态恢复锁力。
保持阻尼	舵机立即停止运动，并进入阻尼模式，外力可以调整角度。

6.7 同步指令

- 单条指令同时包含多个伺服舵机的控制指令，适用于多个舵机协同动作的场景。
- 每个伺服舵机通过唯一的 ID 与指令内容中的参数进行匹配，仅解析并响应与自身 ID 相关的控制信息。
- 所有伺服舵机接收完指令后，将同时开始执行各自的指令，实现同步动作效果。

6.8 异步指令

- 异步指令由 **异步写入指令** 和 **异步执行指令** 两部分组成。
- 异步写入指令发送后，再发送需要执行的运动指令，使该指令暂存在对应 ID 伺服舵机的寄存器中，该运动指令不会立即执行，需通过异步执行指令统一启动。
- 当异步执行指令下达时，所有已暂存运动指令的伺服舵机将同时执行对应动作，实现多个伺服舵机的同步控制效果。
- 已暂存的运动指令，在未重新写入或未断电的情况下将持续保留，不会因其他指令的运行而被覆盖或清除。
- 异步指令被触发执行后，相关参数将被自动清除，不再保留。

6.9 工作状态数据读取

6.9.1 数据读取

下列表格的工作状态数据, 均可通过 03 (0x03) 指令单独读取

参数编号	参数名称	单位
01	当前工作电压	mV
02	当前工作电流	mA
03	当前工作功率	mW
04	当前工作温度	ADC
05	舵机状态	

6.9.2 舵机状态

bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
指令执行中	指令执行错误	堵转保护	高压保护	低压保护	电流保护	功率保护	温度保护

6.9.3 数据监控

下列表格的工作状态数据, 可以通过数据监控指令, 实现批量回读;

字节编号	参数名称	单位
[04]	伺服舵机 ID	
[5, 6]	当前工作电压	mV
[7, 8]	当前工作电流	mA
[9, 10]	当前工作功率	mW
[11, 12]	当前工作温度	ADC
[13]	舵机状态	
[14, 17]	当前舵机角度	0.1°
[18, 19]	当前圈数	

6.10 设置原点

在舵机处于锁力释放的状态下, 通过上位机或指定指令, 可将当前舵机的角度重置为零位, 便于装配后的零位校准, 同时为算法提供后续动作的起始角度。