



DR EMPOWER

大然机器人无刷伺服驱动器

ASCII 通讯协议

**DrEmpower**

大然机器人  
Daran Robot

## 目录

第一章	无刷伺服驱动器 ASCII 通讯协议概要 .....	2
1.1	硬件连接指南 .....	2
1.2	电机协议解析 .....	2
1.3	命令查询表 .....	3
1.4	命令详解 .....	5
1.4.1	控制指令 .....	5
1.4.2	读写指令 .....	9
第二章	ASCII 通讯协议示例 .....	12
2.1	控制指定编号电机按照指定速度转动到指定角度 .....	12
2.2	控制指定编号电机按照指定速度转动 .....	12
2.3	控制制定编号电机按照指定扭矩运动 .....	13
2.4	获取电机 ID 号，角度，速度，电压，电流，设置电机 ID 号 .....	14

## 第一章 无刷伺服驱动器 ASCII 通讯协议概要

### 1.1 硬件连接指南

电机通过 ASCII 协议进行通讯时，允许使用以下两种方法连接到电机进行通讯：

#### 1、通过 USB 接口

通过 USB 进行连接时，首先使用 mini USB 线缆连接电机和电脑，此时电脑会多出来一个 USB 串行设备，随后便可以连接该串口的串口号，通过串口助手或串口通讯脚本进行 ASCII 通讯。

#### 2、通过板载 UART 接口

将 UART 模块 RX 连接到电机 TX 端口，将 TX 端口连接到电机 RX 端口，将 GND 端口连接到电机 GND 端口，然后将 UART 模块插入电脑中，选择对应的 UART 模块串口号，通过串口助手或串口通讯脚本进行 ASCII 通讯。

#### 3、通讯参数

使用 ASCII 通讯方式进行通讯时，串口控制端只可以同时控制至多一个与控制端直接连接的电机。在通讯的过程中，因使用的是虚拟串口模式，关于串口的波特率可为任意一个常用的波特率（例如：115200、9600 等），奇偶校验为无，数据位为 8，停止位为 1。

### 1.2 电机协议解析

#### 1、标准 ASCII 协议指令结构

指令段	电机号	输入值 1	输入值 2	...	校验位	结尾符
-----	-----	-------	-------	-----	-----	-----

参数含义解释：

指令段	指令段确定该指令的主要功能，一般为单个字母字符构成
电机号	电机号为电机的 ID 号，对应的属性参数为 axis0.config.can_node_id，这个属性参数值必须大于 0。 <b>所有的电机的公共 ID 为 0，如果用公共 ID 作为指令中的电机号，则所有电机都会响应这条指令。如果控制器同时连接有多个电机，则在读取类型指令中不能使用公共 ID。</b>
输入值*	根据参数值的不同，该值的数量可能为 0~3 个，输入的值为字母字符串或由字符串构成的数字组合
校验位	校验位由*号及异或校验和组成，如发送端带有校验位，则接收端每一行也会带有校验位。（ <b>校验位是可选位，不带校验也可以</b> ）
结尾符	所有指令的结尾符都是以换行符 ‘\n’ 结尾

注:

1) 两个参数之间用一个空格字符' '将两个参数隔开,当电机收到换行符(结尾符)时开始解释换行符之前的指令,如电机速度指令'v motor 100.5 2.3 \n'。

2) 异或校验和计算方法具体为,将一条完整指令每一个字符的 ASCII 的值转换为对应 8 位 2 进制(包括空格字符),先将前两位 8 位 2 进制字符按位异或,得到一个值,再将该值与下一个字符对应的 8 位 2 进制值进行异或,直到最后一位空格字符。最后得到一个 8 位 2 进制校验值,将该 8 位 2 进制校验值换算成整数值,加\*号,即为校验位数值,如注 1 中的电机速度指令'v motor 100.5 2.3 \n',计算得到的校验值为'\*24',如果需要发送带有检验位的命令则需要发送的字符串为'v motor 100.5 2.3 \*24\n'。

### 1.3 命令查询表

指令作用	指令段	电机号	输入值 1	输入值 2	输入值 3
轨迹跟踪模式	t	0~63	目标位置	目标速度	目标角度输入滤波带宽
梯形轨迹模式	q	0~63	目标位置	目标速度	目标加速度
前馈控制模式	p	0~63	目标位置	速度前馈	扭矩前馈
电机速度指令	v	0~63	目标转速	模式参数	目标模式
速度控制指令的输入值 3 用来指定速度控制模式, 1 表示速度前馈模式,此时输入值 2 中的模式参数为前馈扭矩 (Nm) 2 表示速度爬升模式,此时输入值 2 中的模式参数为速度爬升速率 ((r/min)/s)					
电机扭矩指令	c	0~63	目标扭矩	模式参数	目标模式
扭矩控制指令的输入值 3 用来指定扭矩控制模式, 1 表示直接控制模式,此时输入值 2 中的模式参数无作用 6 表示扭矩爬升模式,此时输入值 2 中的模式参数为扭矩爬升速率 (Nm/s)					
阻抗控制模式	d	0~63	Pdes	Vdes	Tff Kp Kd
阻抗控制指令输入值参数一共有 5 个,其含义如下: Pdes: 电机目标角度 (degree) Vdes: 电机目标速度 (r/min) Tff: 前馈扭矩 (Nm) Kp: 刚度系数 Kd: 阻尼系数  阻抗控制为 MIT 开源方案中的控制模式,其目标输出扭矩计算公式如下:  $\text{Torque} = Kp * (Pdes - pos_) + Tff + Kd * (Vdes - vel_)$  其中 pos_ 和 vel_ 分别为输出轴当前实际位置 (degree) 和当前实际速度 (r/min), Kp 和 Kd 为刚度系数和阻尼系数,系数比例与 MIT 等效。					

运动参数预设指令	<b>ms</b>	0~63	参数 1	参数 2	参数 3
参数 1~参数 3 含义由起始指令 mt~mc 决定					
起始指令	参数 1	参数 2		参数 3	
mt	目标位置	目标速度（无作用）		目标角度输入滤波带宽	
mq	目标位置	目标时间		目标加速度	
mp	目标位置	前馈速度		前馈扭矩	
mv	目标速度	模式参数		目标模式	
mc	目标扭矩	模式参数		目标模式	
轨迹跟踪模式起始指令	<b>mt</b>	0	0~1	\	\
梯形轨迹模式起始指令	<b>mq</b>	0	0~1	\	\
前馈控制模式起始指令	<b>mp</b>	0	0~1	\	\
角度控制模式起始指令的输入值 1 用来指定是否为绝对角度还是相对角度， 0 表示绝对角度，目标角度相对于输出轴零点 1 表示相对角度，目标角度相对于当前位置					
电机速度控制起始指令	<b>mv</b>	0	\	\	\
电机扭矩控制起始指令	<b>mc</b>	0	\	\	\
m*指令都是用来实现多个电机同时控制，指令包含两步： 1. 第一步是通过 ms 运动参数预设指令将各个电机的运动参数发送过去，电机收到之后并不会直接运动，而是等待起始指令。 2. 第二步发送起始指令 mt~mc，由于控制多个电机，所以起始指令需要用公共 ID（0）进行发送。电机接收到起始指令后，即按照起始指令的类型及预设指令中的三个参数进行运动。					
参数读	<b>r</b>	axis 电机号.属性名称		\	\
参数写	<b>w</b>	axis 电机号.属性名称		设定值	\
保存配置	<b>ss</b>	0~63	\	\	\
擦除配置	<b>se</b>	0~63	\	\	\
重启	<b>sr</b>	0~63	\	\	\
清除错误标志	<b>sc</b>	0~63	\	\	\
设置电机零点	<b>sz</b>	0~63	\	\	\
电机急停	<b>st</b>	0~63	\	\	\
读取帮助信息	<b>h</b>	0~63	\	\	\
读取电机当前位置和速度	<b>f</b>	0~63	\	\	\
读取电机电源电压和 q 轴电流	<b>g</b>	0~63	\	\	\
读取版本信息	<b>i</b>	0~63	\	\	\
读取异常标志	<b>e</b>	0~63	\	\	\
更新看门狗	<b>u</b>	0~63	\	\	\

注：

- 1、输入值中带有 ‘\’ 的表示空缺，对应位置不需要输入值。

2、控制指令中位置相关单位为度，电流相关单位为安培（A），速度相关单位为转每分钟（r/min），且都是以电机输出轴作为控制对象。

3、参数读取和写入具体的单位及控制对象（可能是电机输出轴或者电机转子）由读取或写入的参数在程序中具体的设定决定。

4、f 指令回读信息为[当前输出轴位置，当前输出轴速度]

5. g 指令返回的信息为[当前母线电压，q 轴电流]

## 1.4 命令详解

上表中的指令大致可以分为 5 类，分别用不同的颜色进行了区分。

第一类为单个电机控制指令，包括 ‘t’，‘q’，‘p’，‘v’，‘c’，覆盖了单个电机的位置、速度及扭矩控制。

第二类为多个电机控制指令，包括运动参数预设指令 ms 和运动起始指令 ‘mt’，‘mq’，‘mp’，‘mv’，‘mc’，覆盖了多个电机的位置、速度及扭矩控制。

第三类为读写指令，包括 ‘r’ 和 ‘w’，用来读写电机相关参数。

第四类为系统指令，都是以 ‘s’ 开头，主要是用来完成一些系统操作。

第五类为辅助指令，包括 ‘h’，‘f’，‘i’，‘e’，‘u’，主要用来读取一些相关信息。

下面逐一进行讲解，具体说明如下：

### 1.4.1 控制指令

1、 ‘t’ 指令，指令格式为：t 电机号 目标位置（度）目标速度（r/min） 角度输入滤波带宽（1/s）

示例：“t 1 10 0 25\n” 表示控制 1 号电机（输出轴）以轨迹跟踪模式转动到 10 度位置，角度滤波带宽为 25

该指令为轨迹跟踪模式的位置控制指令，只有一个目标位置参数，无需指定速度等参数。可用于一般的位置控制。当外部控制器以一定频率向电机连续发送该位置指令时，电机会自动进行平滑位置控制，保证整个运动足够平滑，自动对输入轨迹点进行跟踪控制，因此该模式被称为轨迹跟踪模式。

该指令最后一个参数输入滤波带宽(controller.config.input\_filter\_bandwidth)，一般情况下推荐将输入滤波带宽设置为位置指令发送频率的一半。另外该指令中的速度参数没有作用，可直接给 0。电机转动速度是通过调节相邻两条指令间的目标位置疏密来实现的。

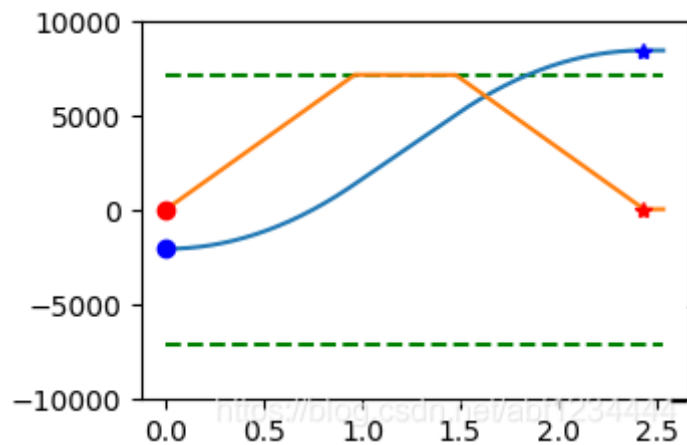
例如控制器没 20ms 给电机发送一次 t 控制指令，则可将该参数设置为配置

滤波带宽: `axis0.controller.config.input_filter_bandwidth = 1/0.02*0.5=25 (1/s)`

2、‘q’指令, 指令格式为: **q 电机号 目标位置(度) 目标速度(r/min) 目标加速度((r/min)/s)**

示例: “q 2 160 30 10 \n” 表示控制 2 号电机(输出轴)以梯形轨迹模式转动到 160 度位置, 最大速度为 30r/min, 加速度为 10(r/min)/s.

该指令为梯形轨迹模式的位置控制指令, 控制过程为电机首先按照指定加速度加速到目标速度, 然后再按目标速度运动一段时间, 最后速度按照相反的加速度减速到 0 并到达目标位置。当目标位置离当前位置比较近或加速度很小的时候, 电机可能出现未到达目标速度就开始减速, 整个速度曲线是一个三角形。一个标准的梯形轨迹控制模式的位置速度曲线如下图所示, 其中蓝色的为位置曲线, 绿色的速度曲线。



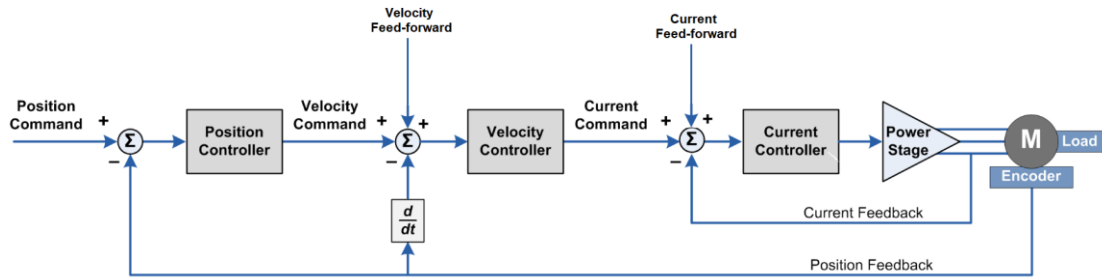
该指令也有一个相关的控制参数: 负载惯量(`axis0.output_shaft.load_inertia`). 此项和电机负载转矩和加速度相关, 默认为 0。设置为 0 即忽略此项影响, 如果将此调整到合适的值可以改善系统的响应速度。负载惯量指的是负载相对于电机轴线的转动惯量, 单位为[Nm/(r/ s^2)]。

3、‘p’指令, 指令格式为 **p 电机号 目标位置(r) 速度前馈(r/s) 扭矩前馈(Nm)**

示例: “p 0 10 5 3 \n” 表示控制 0 号电机(输出轴)以前馈控制模式转动到 10 度位置, 前馈速度为 5r/min, 前馈扭矩为 3Nm

该指令的作用是进行位置前馈控制, 其控制框图如下:





前馈控制将干扰测量出来并直接引入调节装置，对于干扰的克服比反馈控制及时。详细的关于前馈控制的介绍可以参考下面的链接 <http://www.szvector.com/index.php?id=318>。

前馈控制位置模式属于偏底层的控制方式，一般情况下用户无需使用，直接采用另外两种位置控制方式即可。值得一提的是，在‘q’指令的梯形轨迹位置控制模式中，就是采用了前馈控制实现的。

**4、‘v’指令，指令格式为 v 电机号 目标速度(r/min) 前馈扭矩(Nm)或目标加速度((r/min)/s) 目标模式（1 或者 2）**

**示例 1：**”v 0 20 3 1 \n” 表示控制 0 号电机以转速 20r/min 转动，且前馈扭矩为 3Nm，模式为速度前馈模式。

**示例 2：**”v 2 15 3 2 \n” 表示控制 2 号电机以转速 15r/min 转动，且目标加速度为 3(r/min)/s，模式为速度爬升模式。

该指令为速度控制指令，该指令有两种控制模式，一种为速度前馈控制模式（类似于位置控制的‘p’指令），发送指令后，电机转速会立即变化到目标速度值。该模式可以指定扭矩前馈量，为相对底层的控制方式，一般情况下如果不知道扭矩前馈量给多少，可以直接给 0，表示忽略前馈控制。另一种速度爬升控制模式（类似位置控制的‘q’指令），可以指定速度上升过程的加速度。

**5、‘c’指令，指令格式为 c 电机号 目标扭矩(Nm) 扭矩上升速率 目标模式（1 或者 6）**

**示例 1：**”c 0 5 0 1 \n” 表示控制 0 号电机控制输出扭矩为 5Nm，模式为直



接控制模式。

**示例 2：**”c 2 5 3 6 \n”表示控制 2 号电机控制输出转矩为 5Nm，且扭矩上升率 3Nm/s，模式为扭矩爬升模式。

该指令为扭矩控制指令，该指令有两种控制模式，一种为扭矩直接控制模式，发送完指令后，电机输出扭矩会立刻变化到目标扭矩值。另一种扭矩爬升控制模式（类似于速度爬升控制模式），可以指定扭矩上升速率。

最大速度限制使能标志位(axis0.controller.enable\_current\_mode\_vel\_limit).

扭矩控制时，电机的转速会根据负载的大小变化。如果电机转速超过您设置的 vel\_limit，电机输出的力矩将会减小。  
axis0.controller.enable\_current\_mode\_vel\_limit = False 来禁止力矩减小。

**6、 ‘ms’ + ‘m\*’指令，多个电机控制指令（由运动参数预设指令和起始指令组成），与上述 1~5 中的单个电机控制指令对应。指令格式分别为 ms 电机号 参数 1 参数 2 参数 3 和 m\* 0 (x)**

运动参数预设指令 ms 中的参数 1~参数 3 含义由起始指令 m\*决定。

**示例 1：**” ms 1 120 5 10 \n”、” ms 2 -100 5 10 \n”、” mq 0 1 \n”表示以梯形轨迹模式控制 1 号电机相对于当前位置正向转动 120 度，2 号电机相对于当前位置反向转动 100 度，转动时间为 5s，加速度为 10(r/min)/s。梯形轨迹模式下的多个电机控制指令，速度参数用时间表示，从而保证多个电机同时开始转动，同时到达目标位置。

**注意：**m+x 指令都是用来实现多个电机同时控制，指令包含两步：

1. 第一步是通过 ms 运动参数预设指令将各个电机的运动参数发送过去，电机收到之后并不会直接运动，而是等待起始指令。
2. 第二步发送起始指令 mt~mc，由于控制多个电机，所以起始指令需要用公共 ID（0）进行发送。电机接收到起始指令后，即按照起始指令的类型及预设指令中的三个参数进行运动。

**7、 在所有控制指令中还存在两个全局参数，对所有控制模式均有效。分别为**

axis0.motor.config.current\_lim = <Float>

axis0.controller.config.vel\_limit = <Float>

两个参数分别限制电机运行过程中的最大电流和最大速度，不管在何种控制模式下，两个参数均有效，始终保证电机速度和电流不超过设置的最大值。一般情况下，用户不要轻易修改上述两个值，以防损坏电机。

#### 1.4.2 读写指令

属性名称	中文解释	单位	类型
<b>vbus_voltage</b>	<b>总线电压</b>	<b>V</b>	<b>float32</b>
<b>ibus</b>	<b>总线电流</b>	<b>A</b>	<b>float32</b>
<b>config.uart_baudrate</b>	<b>UART 波特率</b>	<b>9600~115200</b>	<b>uint32</b>
<b>can.config.baud_rate</b>	<b>CAN 通信波特率</b>	<b>125K~1M</b>	<b>uint32</b>
axis0.config.can_node_id	can 总线节点 ID(电机 ID)	1~63	uint32
axis0.controller.trjectory_done	角度控制电机是否到达目标位置	0 或 1	bool
axis0.controller.config.pos_gain	位置环增益	(turn/s)/turn	float32
axis0.controller.config.vel_gain	速度环增益	Nm/(turn/s)	float32
axis0.controller.config.vel_integrator_gain	速度环积分增益	Nm/(turn/s^2)	float32
axis0.controller.config.vel_limit	最大转速	turn/s	float32
axis0.controller.config.input_filter_bandwidth	轨迹跟踪模式角度输入滤波带宽	1/s	float32
axis0.motor.config.current_lim	电机最大运行电流	A	float32
axis0.motor.config.torque_lim	电机（转子）输出最大扭矩	Nm	float32
axis0.motor.config.current_control_bandwidth	电流控制环的控制带宽		float32
axis0.motor.current_control.Iq_measured	通过电流采样获取的交轴电流	A	float32
axis0.fet_thermistor.error	电机驱动板过温保护错误标志		uint32
axis0.fet_thermistor.temperature	电机驱动板当前温度	°	float32
axis0.fet_thermistor.config.enabled	电机驱动板过温保护使能		bool
axis0.fet_thermistor.config.temp_limit_lower	电机驱动板过温保护最低温度	°	float32
axis0.fet_thermistor.config.temp_limit_upper	电机驱动板过温保护最高温度	°	float32
axis0.motor_thermistor.error	电机过温保护错误标志		uint32
axis0.motor_thermistor.temperature	电机当前温度	°	float32
axis0.motor_thermistor.config.enabled	电机过温保护使能		bool
axis0.motor_thermistor.config.temp_limit_lower	电机过温保护最低温度	°	float32
axis0.motor_thermistor.config.temp_limit_upper	电机过温保护最高温度	°	float32
axis0.output_shaft.input_pos	电机输出轴输入目标角度	degree	float32
axis0.output_shaft.input_vel	电机输出轴输入目标速度	r/min	float32
axis0.output_shaft.input_torque	电机输出轴输入目标扭矩	Nm	float32
axis0.output_shaft.pos_setpoint	电机输出轴当前目标角度	degree	float32
axis0.output_shaft.vel_setpoint	电机输出轴当前目标速度	r/min	float32
axis0.output_shaft.torque_setpoint	电机输出轴当前目标扭矩	Nm	float32
axis0.output_shaft.pos_estimate	电机输出轴当前实际角度	degree	float32
axis0.output_shaft.vel_estimate	电机输出轴当前实际速度	r/min	float32
axis0.output_shaft.torque_estimate	电机输出轴当前实际扭矩	Nm	float32
axis0.output_shaft.circular_setpoint_min	电机输出轴软件限位最小角度	degree	float32
axis0.output_shaft.circular_setpoint_max	电机输出轴软件限位最大角度	degree	float32
axis0.output_shaft.vel_limit	电机输出轴最大速度限制	r/min	float32
axis0.output_shaft.torque_lim	电机输出轴最大扭矩限制	Nm	float32
axis0.output_shaft.vel_ramp_rate	电机输出轴速度上升速率	r/(min*s)	float32
axis0.output_shaft.torque_ramp_rate	电机输出轴扭矩上升速率	Nm/s	float32
axis0.output_shaft.steps_per_turn	电机输出轴每圈对应的脉冲	count/r	float32

属性名称	中文解释	单位	类型
	数		
axis0.output_shaft.torque_constant	电机输出轴扭矩常数	Nm/A	float32
axis0.output_shaft.load_inertia	电机输出轴端负载转动惯量	Nm/(r/s^2)	float32

## 1、‘r’ 指令，指令格式为 r axis 电机号.属性

示例 1: “r axis1.output\_shaft.pos\_setpoint \n”表示读取 1 号电机(输出轴)的目标角度。

示例 2: “r axis5.motor.current\_control.Iq\_measured \n”表示读取 5 号电机的当前 q 轴电流。

示例 3: “r axis0.config.can\_node\_id \n”表示读取 0 号电机的当前 ID。

由于表格中前四项属性名称中不含 axis，故单独举例：

示例 4: “r config.uart\_baudrate 1 \n”表示读取 1 号电机 UART 波特率。

电机属性名称可以分为两类，

第一类属性名称中带 axis，该种属性的读写指令的电机号包含在了属性参数的 axis 后，例如 “r axis5.motor.current\_control.Iq\_measured \n”。

第二类属性名称中不带 axis，对于名称中不含 axis 的属性，电机的 ID 号写在命令最后，换行符之前，例如 “r config.uart\_baudrate 1 \n”。

## 2、‘w’ 指令，指令格式为 w axis 电机号 属性 值

示例 1: “w axis1.output\_shaft.pos\_setpoint 150 \n”表示将 1 号电机（输出轴）的目标位置修改为 150 度。

示例 2: “w axis0.config.can\_node\_id 5 \n”表示将电机 ID 号改为 5。这里电机号采用的是公共 ID，所以会将连接的所有电机 ID 都改为 5。

由于表格中前四项没有 axis，故单独举例：

示例 3: “w config.uart\_baudrate 9600 1 \n”表示将 1 号电机 UART 波特率改为 9600。

‘w’ 指令与 ‘r’ 指令相比多了一个设定值参数，表示需要将该属性参数修改为给定值。需要注意的是 axis 对象的属性中有一些只能读，不能修改，这些属性不能用 ‘w’ 指令进行修改。对于属性名称中不含 axis 的属性，电机的 ID 号同样是写在整条命令的最后，换行符之前。

另外，‘w’ 指令修改参数后需要调用系统指令 ‘ss’ 指令进行保存，否则电机重新上电后将恢复成设置前的值。

注：

读写命令均只能执行浮点数和整形的读写，bool 类型无法直接读取和写入，

需要用 0 和 1 替代 **False** 和 **True**。

## 第二章 ASCII 通讯协议示例

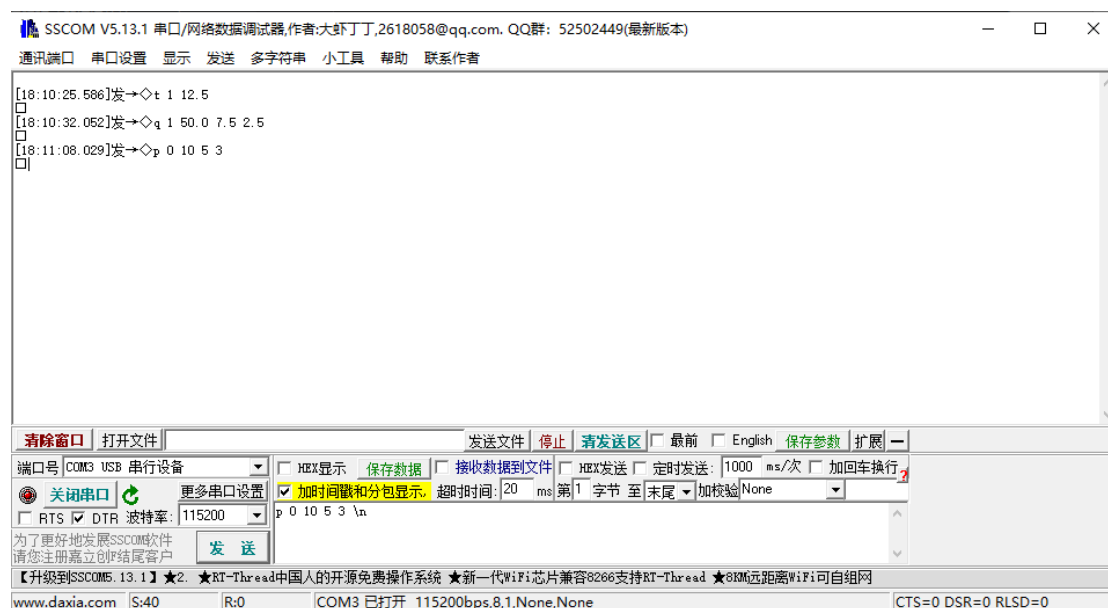
### 2.1 控制指定编号电机按照指定速度转动到指定角度

使用串口助手发送信息：

“t 1 12.5 \n”，表示 1 号电机（输出轴）以轨迹跟踪模式转动到 12.5°，

“q 1 50.0 7.5 2.5 \n”，表示 1 号电机以梯形轨迹模式转动到 50 度位置，最大速度为 7.5r/min，加速度为 2.5(r/min)/s，

“p 0 10 5 3 \n”表示控制 0 号电机以前馈控制模式相对零点正向转动 10 度，前馈速度为 5r/min，前馈扭矩为 3Nm

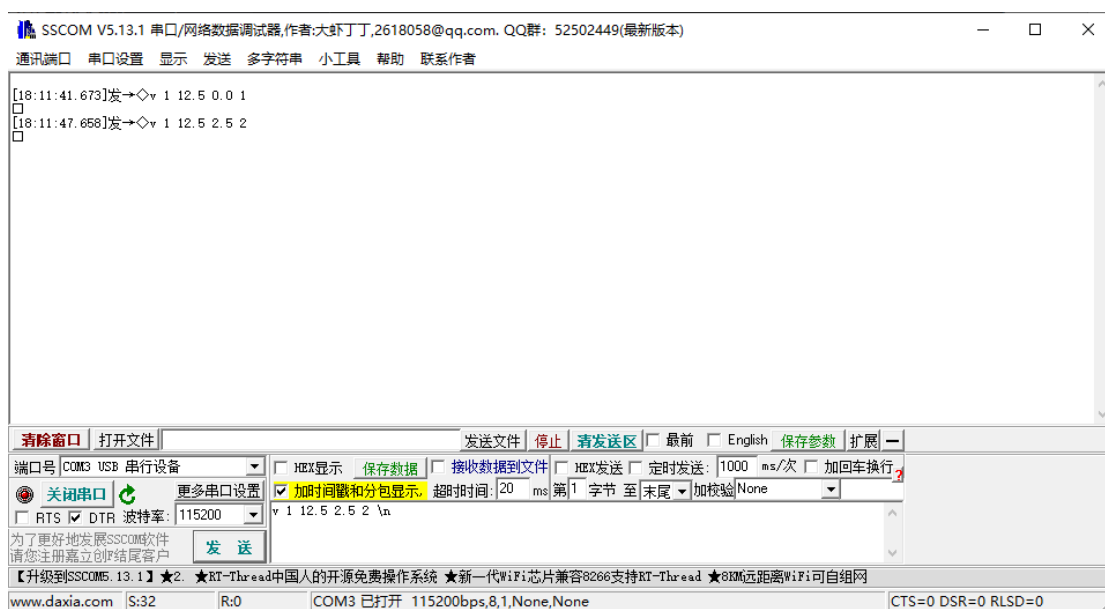


### 2.2 控制指定编号电机按照指定速度转动

使用串口助手发送信息：

“v 1 12.5 0.0 1 \n”表示 1 号电机以前馈控制模式控制电机输出轴以 12.5r/min 运行，前馈扭矩为 0.0Nm

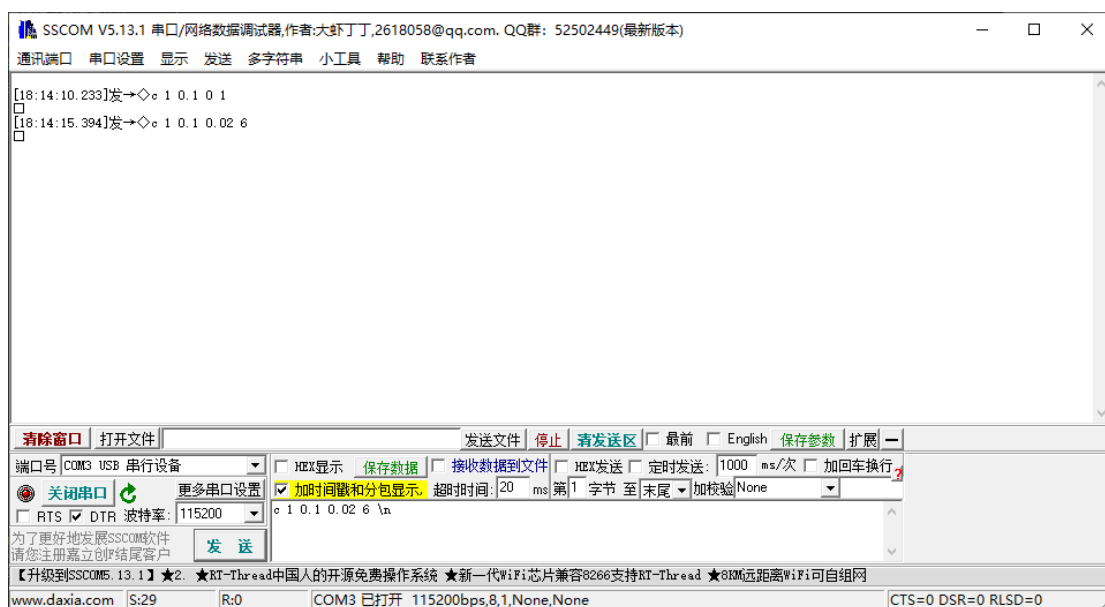
“v 1 12.5 2.5 2 \n”表示 1 号电机以速度爬升模式控制电机输出轴以加速度 2.5(r/min)/s 加速至 12.5r/min 运行



## 2.3 控制制定编号电机按照指定扭矩运动

“c 1 0.1 0 1 \n”表示 1 号电机输出轴输出 0.1Nm 扭矩

“c 1 0.1 0.02 6 \n”表示 1 号电机输出轴输出扭矩以 0.02Nm/s 加速至 0.1Nm



## 2.4 获取电机 ID 号，角度，速度，电压，电流，设置电机 ID 号

“r axis0.config.can\_node\_id \n” 表示读取当前电机的 ID 号，  
返回值为当前电机 ID 号。

“w axis0.config.can\_node\_id 1 \n” 表示修改当前电机的 ID 号为 1，  
之后需发送“ss 1 \n”才能保存数据。

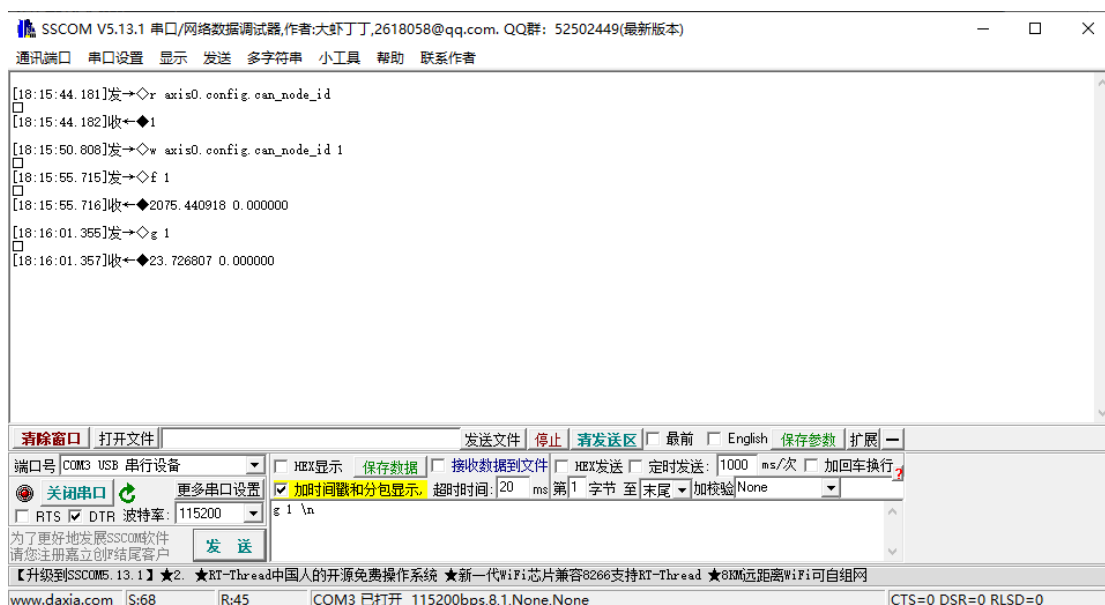
“f 1 \n”表示读取 1 号电机角度，速度

返回值为[a, b]，a 为电机输出轴当前角度，b 为电机输出轴当前速度。



“g 1 \n”表示读取 1 号电机电压值和电流值

返回值为[a, b], a 为电机当前电压值, b 为当前 q 轴电流值（可表示扭矩大小）。



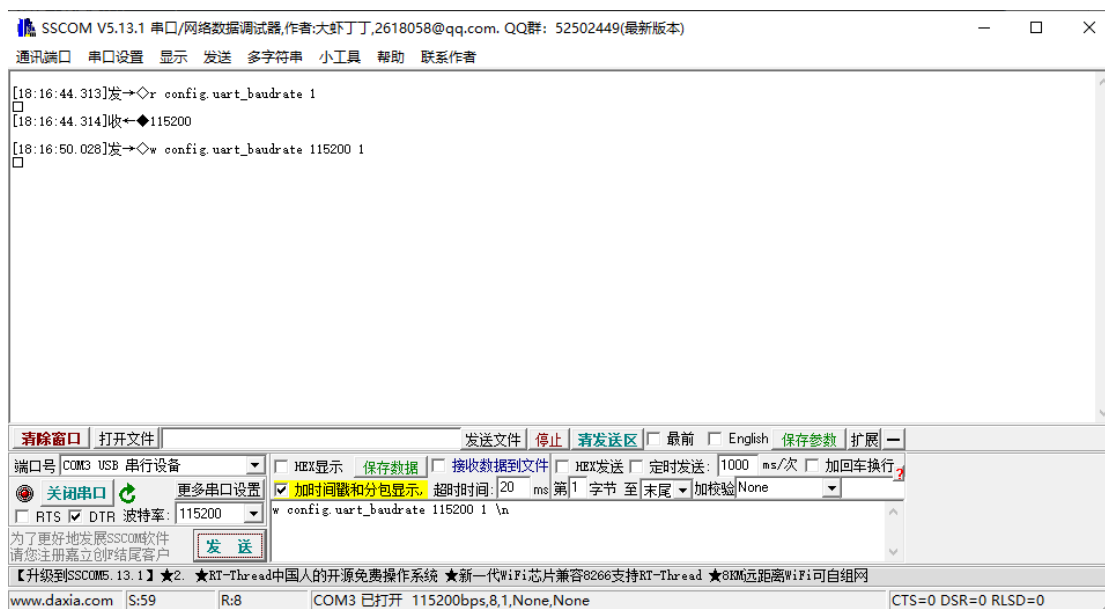
## 2.5 获取及修改电机 UART 波特率

由于部分参数名没有 axis, 在设置时将电机 ID 号写在命令最后, 换行符之前。

“r config. uart\_baudrate 1 \n”表示读取 1 号电机 UART 波特率,

返回值为电机 UART 波特率。

“w config. uart\_baudrate 115200 1 \n”表示将 1 号电机 UART 波特率改为 115200。



Daran Robot

Website: [www.daran.tech](http://www.daran.tech) Tel: 022-8369-9286 Post: 300000