RS232 串口通信协议

本篇文档适用于 RMDS 系列 101、102、103、104、105、105+、106、106+、107、201、301、302、303、303+、401 等全系列的 RoboModule 驱动器。

本段介绍如何使用 RS232 串口的方式操作驱动器来控制电机的各种方式转动。

在 RS232 控制方式下需要连接的线有：
电源线（24V、GND）、
编码器（CHA、CHB、GND、+5V）、
电机（MT1、MT2）、
RS232 线（232T、GND、232R）。
其中，232T 要连接到主控器的 RS232 接收端（R 端），232R 要连接到主控中心的 RS232 发送端（T 端）。

对于本段协议，RS232 的串口波特率默认为 115200，数据位 8，停止位 1，无校验，无流控制。RS232 波特率支持在<RoboModule 电机驱动器调试软件>上修改，支持如下 10 种波特率：921600、460800、230400、115200、57600、38400、19200、14400、9600、4800 等。
在<RoboModule 电机驱动器调试软件>中对应的调试界面如下：

由于 RS232 串口是一对一的传输方式，所以，一个主控器不能通过同一个 RS232 接口来挂接两个驱动器，如果要使用多个驱动器并且使用 RS232 串口挂接在主控器上，则要求该主控器有多个 RS232 接口。比如要控制三个驱动器，则要求使用三个 RS232 串口才能实现完整的控制。
当不具备这种多 RS232 串口资源的时候，CAN 总线和 RS485 则是更好的选择。一条 CAN 总线，可以支持挂接 120 个驱动器，一条 RS485 总线，可以支持挂接 15 个驱动器。

下面正式介绍 RS232 串口通信协议：

因为 RS232 串口通信只能是一个主机（电脑作为主机或者 MCU 主控器作为主机）和一个从机（当前连接的

驱动器作为从机），所以，所发送的所有指令，都只有本串口所连接的驱动器能收到，所以不存在选中哪一个驱动器的问题，所以不需要像 CAN 总线或 RS485 那样用标识符来区分谁是谁。

每段串口命令都是由 10 个字节组成，

主控器发送给驱动器的指令，有如下 12 种：

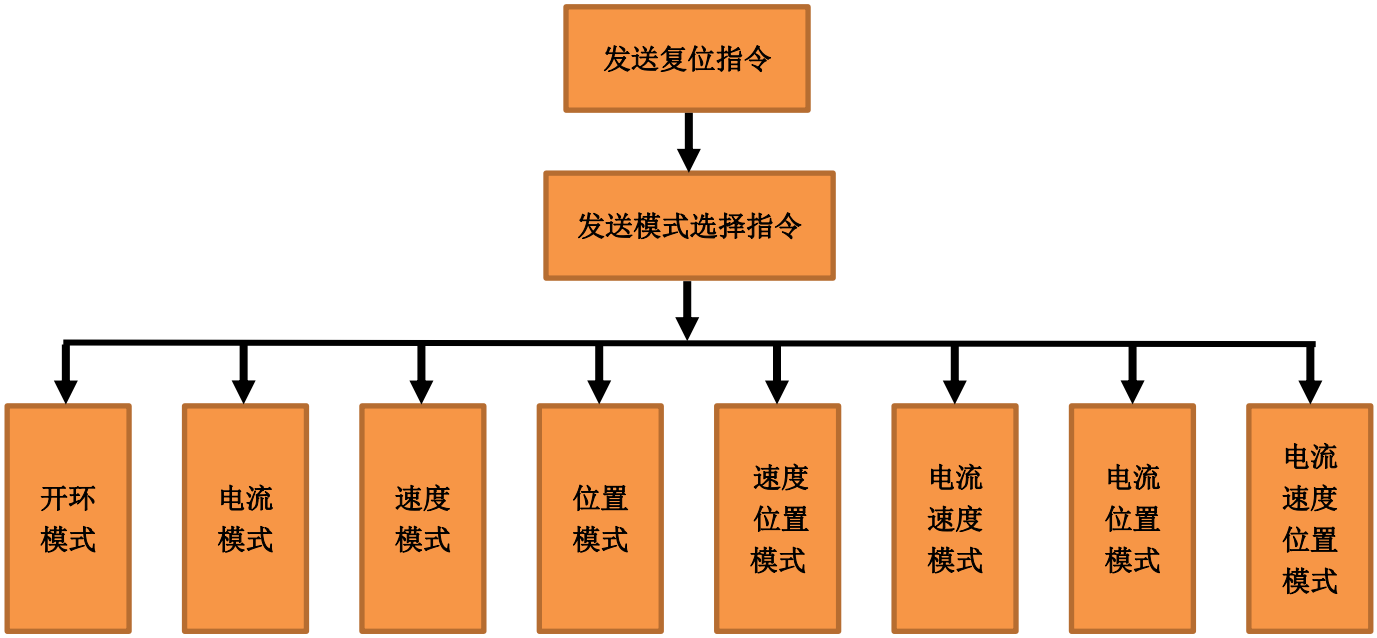
1. 让当前驱动器复位。
2. 让当前驱动器进入某种模式，有 8 种可选模式：
   ① 开环模式
   ② 电流模式
   ③ 速度模式
   ④ 位置模式
   ⑤ 速度位置模式
   ⑥ 电流速度模式
   ⑦ 电流位置模式
   ⑧ 电流速度位置模式
3. 在“开环模式”下给驱动器发送目标 PWM 的指令。
4. 在“电流模式”下给驱动器发送限制 PWM 和目标电流的指令。（电流值的单位是 mA）
5. 在“速度模式”下给驱动器发送限制 PWM 和目标速度的指令。（速度值的单位是 RPM）
6. 在“位置模式”下给驱动器发送限制 PWM 和目标位置的指令。（位置值的单位是 qc）
7. 在“速度位置模式”下给驱动器发送限制 PWM、限制速度、目标位置的指令。
8. 在“电流速度模式”下，给驱动器发送限制电流和目标速度的指令
9. 在“电流位置模式”下，给驱动器发送限制电流和目标位置的指令
10. 在“电流速度位置模式”下，给驱动器发送限制电流和限制速度和目标位置的指令
11. 配置指令，配置驱动器对外发送电流、速度、位置等信息的时间间隔和 CTL 端口的电平对外发送的时间间隔。（2018.01.26 修改）
12. 检测驱动器是否在线。

驱动器发送给主控器的指令，有如下 3 种：

1. 驱动器对外发送电流、速度、位置等信息。
2. 驱动器对外发送 CTL1、CTL2 端口电平状态。
3. 驱动器对外发送在线检测反馈。

下面来分解每个指令的具体内容：

一、控制流程图：



使用 RS232 串口的方式控制驱动器，控制流程如下：

1. 发送复位指令
2. 等待 500ms
3. 发送模式选择指令，使驱动器进入某种模式
4. 等待 500ms
5. 在已经进入的模式下发送数据指令。（周期性发送本条指令，间隔最短为 1ms，推荐间隔 10ms）

零、复位指令：

本指令在任何状态下都会直接生效。
发送本指令后，驱动器会立即复位，即程序从头开始运行。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x00	0x55	0x55	0x55	0x55	0x55	0x55	0x55	0x55

发送完本指令后，驱动器上的蜂鸣器会响一声，持续时间为 300ms。

一、模式选择指令：

本指令只在驱动器未进入任何模式的情况下生效。
如果驱动器已经进入某种模式，再发送此指令则会报错（报错的具体表现为红灯闪烁，驱动器的蜂鸣器不断的鸣叫）。
所以在发送本指令前，建议先发送复位指令，等待驱动器复位完成（大约 500ms），再发送本指令。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x01	待选择	0x55	0x55	0x55	0x55	0x55	0x55	0x55

Data[2]与所选模式的对应值如下：

模式名称	Data[2]的值
开环模式	0x01
电流模式	0x02
速度模式	0x03
位置模式	0x04
速度位置模式	0x05
电流速度模式	0x06
电流位置模式	0x07
电流速度位置模式	0x08

当驱动器进入上述 8 种模式中的任何一种的时候，蜂鸣器都会响一声，持续时间为 70ms，表示进入模式成功。

二、“开环模式”下的数据指令：

本指令只有在驱动器进入“开环模式”之后才生效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x02	待计算	待计算	0x55	0x55	0x55	0x55	0x55	0x55

举例说明：
让当前驱动器连接的电机在“开环模式”下，让它以 temp_pwm 的占空比转动：
则

```
Data[0] = 0x23;
Data[1] = 0x02;
Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)(temp_pwm&0xff);
Data[4] = 0x55;
Data[5] = 0x55;
Data[6] = 0x55;
Data[7] = 0x55;
Data[8] = 0x55;
Data[9] = 0x55;
```

其中：
temp_pwm 的取值范围为：-5000~+5000。

参数意义说明：
当 temp_pwm 给定值为+4000 的时候，表示给定 PWM 占空比为 4000/5000 = 80%，设置为正转。
当 temp_pwm 给定值为-4000 的时候，表示给定 PWM 占空比为 4000/5000 = 80%，设置为反转。

PWM 占空比假设为 1000，也就是 1000/5000 = 20%，假设此时电源电压为 25V，则 25*20%=5V，使用万用表在电机线上测量电压，黑色表笔连到 GND，红色表笔连到 MT1 或 MT2 上，可以测到 MT1 或 MT2 上会有精确的 5V 的电压。
也就是说，PWM 占空比的宏观表现则是电压的比例值，以此来改变电压进行调速。

三、“电流模式”下的数据指令：

本指令只在驱动器进入“电流模式”之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的值、电流的值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x03	待计算	待计算	待计算	待计算	0x55	0x55	0x55	0x55

举例说明：
让当前驱动器连接的电机以 temp_pwm 的限制 PWM，以 temp_current 的最大电流工作：
则

```
Data[0] = 0x23;
Data[1] = 0x03;
Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)(temp_pwm&0xff);
Data[4] = (unsigned char)((temp_current>>8)&0xff);
Data[5] = (unsigned char)(temp_current&0xff);
Data[6] = 0x55;
Data[7] = 0x55;
Data[8] = 0x55;
Data[9] = 0x55;
```

其中：
temp_pwm 的取值范围为：0~+5000。
temp_current 的取值范围为：-32768~+32767。(16 位有符号整型数取值范围，单位 mA)

参数意义说明：
因为电流输出方向由 temp_current 决定，所以 temp_pwm 给定值不带方向。
当 temp_pwm 给定值为 4000 的时候，表示内部 PID 调节后最终输出的 PWM 占空比不能超过 4000/5000 = 80%。

当 temp_current 给定值为+100 的时候，表示最大输出电流不能超过 100mA,输出力矩方向为正转方向。
当 temp_current 给定值为-100 的时候，表示最大输出电流不能超过 100mA,输入力矩方向为反转方向。

四、“速度模式”下的数据指令：

本指令只在驱动器进入“速度模式”之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的限制值和给定的速度值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x04	待计算	待计算	待计算	待计算	0x55	0x55	0x55	0x55

举例说明：
让当前驱动器连接的电机以 temp_pwm 的限制 PWM，以 temp_velocity 的速度转动：
则

```
Data[0] = 0x23;
Data[1] = 0x04;
Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)(temp_pwm&0xff);
Data[4] = (unsigned char)((temp_velocity>>8)&0xff);
Data[5] = (unsigned char)(temp_velocity&0xff);
Data[6] = 0x55;
Data[7] = 0x55;
Data[8] = 0x55;
Data[9] = 0x55;
```

其中：
temp_pwm 的取值范围为：0~+5000。
temp_velocity 的取值范围为：-32768~+32767。(16 位有符号整型数的范围，单位 RPM)

参数意义说明：
由于电机转动方向由 temp_velocity 决定，所以 temp_pwm 给定值不带方向。
当 temp_pwm 给定值为 4000 的时候，表示内部 PID 调节后最终输出的 PWM 占空比不能超过 4000/5000 = 80%。

当 temp_velocity 给定值为+100RPM 的时候，表示让电机以 100 转每分钟的速度转动，转动方向为正方向。
当 temp_velocity 给定值为-100RPM 的时候，表示让电机以 100 转每分钟的速度转动，转动方向为负方向。

五、“位置模式”下的参数指令：

本指令只在驱动器进入“位置模式”之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的限制值和给定的目标位置值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x05	待计算	待计算	0x55	0x55	待计算	待计算	待计算	待计算

举例说明：
让当前驱动器连接的电机以 temp_pwm 的限制电流，转动到 temp_position 的位置：
则

```
Data[0] = 0x23;
Data[1] = 0x05;
Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)((temp_pwm)&0xff);
Data[4] = 0x55;
Data[5] = 0x55;
Data[6] = (unsigned char)((temp_position>>24)&0xff);
Data[7] = (unsigned char)((temp_position>>16)&0xff);
Data[8] = (unsigned char)((temp_position>>8)&0xff);
Data[9] = (unsigned char)(temp_position&0xff);
```

其中
temp_pwm 的取值范围为：0~+5000。
temp_position 的取值范围为：-2147483648~+2147483647。（32 位有符号整型数的范围）

参数意义说明：
电机转动方向由 temp_position 决定，所以 temp_pwm 给定值不带方向。
当 temp_pwm 给定值为 4000 的时候，表示内部 PID 调节后最终输出的 PWM 占空比不能超过 4000/5000 = 80%。

当 temp_position 给定值为+1000 的时候，表示让电机转动到+1000qc 位置，转动方向为正方向。
当 temp_position 给定值为-1000 的时候，表示让电机转动到-1000qc 位置，转动方向为负方向。

此处实际位置具体是多少，要根据编码器的精度来定，假如编码器是 500 线的，则四倍频后，转一圈 2000qc。则转动 1000qc 为半圈 = 180° 。
另外，假如电机带有减速箱，减速箱为 16 倍力矩放大的，则转 1000qc，实际输出轴转动是 180° /16 = 11.25°

六、“速度位置模式”下的参数指令：

本指令只在驱动器进入“速度位置模式”之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的限制值，运行速度值和给定的目标位置值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x06	待计算	待计算	待计算	待计算	待计算	待计算	待计算	待计算

让当前驱动器连接的电机以 temp_pwm 的限制 PWM，temp_velocity 的限制速度，转动到 temp_position 的位置：
则

```
Data[0] = 0x23;
Data[1] = 0x06;
Data[2] = (unsigned char)((temp_pwm>>8)&0xff);
Data[3] = (unsigned char)((temp_pwm)&0xff);
Data[4] = (unsigned char)((temp_velocity>>8)&0xff);
Data[5] = (unsigned char)(temp_velocity&0xff);
Data[6] = (unsigned char)((temp_position>>24)&0xff);
Data[7] = (unsigned char)((temp_position>>16)&0xff);
Data[8] = (unsigned char)((temp_position>>8)&0xff);
Data[9] = (unsigned char)(temp_position&0xff);
```

其中：
temp_pwm 的取值范围为：0~+5000。
temp_velocity 的取值范围为：0~+32767。（16 位有符号整型数的正数范围，单位 RPM）
temp_position 的取值范围为：-2147483648~+2147483647。（32 位有符号整型数的范围，单位 qc）

参数意义说明：
电机转动方向由 temp_position 决定，所以 temp_pwm 和 temp_velocity 给定值不带方向。
当 temp_pwm 给定值为 4000 的时候，表示内部 PID 调节后最终输出的 PWM 占空比不能超过 4000/5000 = 80%。
当 temp_velocity 给定值为 500 的时候，表示运行过程速度不能超过 500 转每分钟。
当 temp_position 给定值为+1000 的时候，表示让电机转动到+1000qc 位置，转动方向为正方向。
当 temp_position 给定值为-1000 的时候，表示让电机转动到-1000qc 位置，转动方向为负方向。

此处实际位置具体是多少，要根据编码器的精度来定，假如编码器是 500 线的，则四倍频后，转一圈为 2000qc。则转动 1000qc 为半圈 = 180° 。
另外，假如电机带有减速箱，减速箱为 16 倍力矩放大的，则转 1000qc，实际输出轴转动是 180° /16 = 11.25°

七、“电流速度模式”下的数据指令：

本指令只在驱动器进入电流速度模式之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改电流的限制值和给定的速度值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x07	待计算	待计算	待计算	待计算	0x55	0x55	0x55	0x55

举例说明：
让当前驱动器连接的电机以 temp_current 的限制电流，以 temp_velocity 的速度转动：
则

```
Data[0] = 0x23;
Data[1] = 0x07;
Data[2] = (unsigned char)((temp_current>>8)&0xff);
Data[3] = (unsigned char)(temp_current&0xff);
Data[4] = (unsigned char)((temp_velocity>>8)&0xff);
Data[5] = (unsigned char)(temp_velocity&0xff);
Data[6] = 0x55;
Data[7] = 0x55;
Data[8] = 0x55;
Data[9] = 0x55;
```

其中：
temp_current 的取值范围为 0~+32767。（16 位有符号整型数的范围，单位 mA）
temp_velocity 的取值范围为：-32768~+32767。（16 位有符号整型数的范围，单位 RPM）

参数意义说明：
因为力矩输出方向由 temp_velocity 决定，所以 temp_current 给定值不带方向。
当 temp_current 给定值为 2000 的时候，表示驱动器输出电流值最大不能超过 2A。

当 temp_velocity 给定值为+1000 的时候，表示让电机以 1000 转每分钟的速度转动，转动方向为正方向。
当 temp_velocity 给定值为-1000 的时候，表示让电机以 1000 转每分钟的速度转动，转动方向为负方向。

八、“电流位置模式”下的参数指令：

本指令只在驱动器进入“电流位置模式”之后才有效，其他任何状态下发送本指令都会让驱动器报错。
支持连续发送本指令来修改 PWM 的限制值，运行速度值和给定的目标位置值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x08	待计算	待计算	0x55	0x55	待计算	待计算	待计算	待计算

举例：
让当前驱动器连接的电机以 temp_current 的限制电流，转动到 temp_position 的位置：
则

```
Data[0] = 0x23;
Data[1] = 0x08;
Data[2] = (unsigned char)((temp_current>>8)&0xff);
Data[3] = (unsigned char)((temp_current)&0xff);
Data[4] = 0x55;
Data[5] = 0x55;
Data[6] = (unsigned char)((temp_position>>24)&0xff);
Data[7] = (unsigned char)((temp_position>>16)&0xff);
Data[8] = (unsigned char)((temp_position>>8)&0xff);
Data[9] = (unsigned char)(temp_position&0xff);
```

其中
temp_current 的取值范围为 0~+32767。（16 位有符号整型数的范围，单位 mA）
temp_position 的取值范围为：-2147483648~+2147483647。（32 位有符号整型数的范围，单位 qc）

参数意义说明：
因为力矩输出方向由 temp_velocity 决定，所以 temp_current 给定值不带方向。
当 temp_current 给定值为 2000 的时候，表示驱动器输出电流值最大不能超过 2A。

当 temp_position 给定值为+1000 的时候，表示让电机转动到+1000qc 位置，转动方向为正方向。
当 temp_position 给定值为-1000 的时候，表示让电机转动到-1000qc 位置，转动方向为负方向。

此处实际位置具体是多少，要根据编码器的精度来定，假如编码器是 500 线的，则四倍频后，转一圈 2000qc。则转动 1000qc 为半圈 = 180° 。
另外，假如电机带有减速箱，减速箱为 16 倍力矩放大的，则转 1000qc，实际输出轴转动是 180° /16 = 11.25°

九、“电流速度位置模式”下的参数指令：

本指令只在驱动器进入“电流速度位置模式”之后才有效，其他任何状态下发送本指令都会让驱动器报错。支持连续发送本指令来修改 PWM 的限制值，运行速度值和给定的目标位置值，但连续发送的时间间隔不能小于 2 毫秒，建议以 10 毫秒为周期。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x09	待计算	待计算	待计算	待计算	待计算	待计算	待计算	待计算

让当前驱动器连接的电机以 temp_current 的限制电流，temp_velocity 的限制速度，转动到 temp_position 的位置：
则

```
Data[0] = 0x23;
Data[1] = 0x09;
Data[2] = (unsigned char)((temp_current>>8)&0xff);
Data[3] = (unsigned char)((temp_current)&0xff);
Data[4] = (unsigned char)((temp_velocity>>8)&0xff);
Data[5] = (unsigned char)(temp_velocity&0xff);
Data[6] = (unsigned char)((temp_position>>24)&0xff);
Data[7] = (unsigned char)((temp_position>>16)&0xff);
Data[8] = (unsigned char)((temp_position>>8)&0xff);
Data[9] = (unsigned char)(temp_position&0xff);
```

其中
temp_current 的取值范围为 0~+32767。（16 位有符号整型数的正数范围，单位 mA）
temp_velocity 的取值范围为：0~+32767。（16 位有符号整型数的正数范围，单位 RPM）
temp_position 的取值范围为：-2147483648~+2147483647。（32 位有符号整型数的范围，单位 qc）

参数意义说明：
因为力矩输出方向由 temp_velocity 决定，所以 temp_current 给定值不带方向。
当 temp_current 给定值为 2000 的时候，表示驱动器输出电流值最大不能超过 2A。

当 temp_velocity 给定值为 500 的时候，表示电机速度不能超过 500 转每分钟。
当 temp_position 给定值为+1000 的时候，表示让电机转动到+1000qc 位置，转动方向为正方向。
当 temp_position 给定值为-1000 的时候，表示让电机转动到-1000qc 位置，转动方向为负方向。

此处实际位置具体是多少，要根据编码器的精度来定，假如编码器是 500 线的，则四倍频后，转一圈 2000qc。则转动 1000qc 为半圈 = 180° 。
另外，假如电机带有减速箱，减速箱为 16 倍力矩放大的，则转 1000qc，实际输出轴转动是 180° /16 = 11.25°

十、配置指令：

配置指令，目前包含两个功能：
①可以决定是否让驱动器以某个固定的时间间隔通过 RS232 串口对外发送电机当前的实时电流、速度、位置值等信息。
②可以决定 CTL1 和 CTL2 端口在作为左右限位功能后，以某个固定时间间隔对外发送 2 个端口的电平状态。

本指令在任何状态下都可以生效。
但驱动器只在进入上文的 8 种运动模式之后，且配置了固定周期后，才会周期性的对外发送以上所述的电流、速度、位置等信息。
同理，只有进入上文的 8 种运动模式之后，且配置了固定周期后，才会周期性的对外发送限位开关的电平状态。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x0A	待给定	待给定	0x55	0x55	0x55	0x55	0x55	0x55

举例：
让当前驱动器以 10 毫秒为周期的对外发送电流、速度、位置等信息的指令为：
让当前驱动器以 5 毫秒为周期的对外 CTL 电平状态的指令为：

```
Data[0] = 0x23;
Data[1] = 0x0A;
Data[2] = 0x0A; //发送电流速度位置的周期，0x00 为不发送，单位为毫秒。(2016 年 11 月 17 日修改)
Data[3] = 0x05; //发送 CTL 端口电平状态的周期，0x00 为不发送，单位为毫秒。(2018 年 01 月 26 日修改)
Data[4] = 0x55;
Data[5] = 0x55;
Data[6] = 0x55;
Data[7] = 0x55;
Data[8] = 0x55;
Data[9] = 0x55;
```

其中：
当 Data[2] = 0x00 的时候，不允许 232 串口对外发送电流速度位置信息
当 Data[3] = 0x00 的时候，不允许 232 串口对外发送 CTL 的电平状态

注意：对于 102 107 201 301 302 303 303+ 401 除外的驱动器，因为其无 CTL 端口，所以请将 Data[3]=0x00 即可。

十一、数据反馈：

以下是驱动器对外发送电流、速度、位置等信息的 RS232 消息的格式。
需要特别注意，这条消息是由驱动器发出，发出的周期可以通过上述的配置指令来确定。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x0B	待接收	待接收	待接收	待接收	待接收	待接收	待接收	待接收

举例：
驱动器当前的电流值是 real_current，当前的速度值是 real_velocity，当前的位置是 real_position，则驱动器则会对外发送如下的 RS232 消息：

```
Data[0] = 0x23;
Data[1] = 0x0B;
Data[2] = (unsigned char)((real_current>>8)&0xff);
Data[3] = (unsigned char)(real_current&0xff);
Data[4] = (unsigned char)((real_velocity>>8)&0xff);
Data[5] = (unsigned char)(real_velocity&0xff);
Data[6] = (unsigned char)((real_position>>24)&0xff);
Data[7] = (unsigned char)((real_position>>16)&0xff);
Data[8] = (unsigned char)((real_position>>8)&0xff);
Data[9] = (unsigned char)(real_position&0xff);
```

对于主控而言，还原电流、速度、位置的反馈值，可以如下：

```
int16 real_current  = (Data[2]<<8)|Data[3];
int16 real_velocity = (Data[4]<<8)|Data[5];
int32 real_position = (Data[6]<<24)|(Data[7]<<16)|(Data[8]<<8)|Data[9];
```

参数意义说明：
real_current 的单位是 mA，即假如电流为 100，就是 100mA
real_velocity 的单位是 RPM
real_position 的单位是 qc

用户可以利用此项功能进行检测驱动器工作状态，例如如下所述：
1. 可以利用电流反馈值来监测母线电流的值，以此可以在主控上设计一个长时堵转保护功能。
2. 可以利用速度反馈，来分析带负载情况下速度的变化曲线。
3. 可以利用位置反馈，来检测位置环的执行程度，监测位置是否到位，以便设计一个时间紧凑的执行流程。

十二、左右限位的定时反馈：

以下是驱动器周期性对外发送 RS232 数据包的格式。
特别注意，这条 RS232 消息是由驱动器发出，需要满足两个条件，驱动器才会对外发出该数据包
1. 通过 RS232 配置指令，对 CTL1、CTL2 的端口电平发送，设置了不为 0 的周期。
2. 驱动器进入了任意一种运动模式。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x0C	待接收	待接收	0x55	0x55	0x55	0x55	0x55	0x55

举例：

当 CTL1 端口的电平为低电平时候，或光耦为熄灭状态时，Data[2] = 0x00，为高电平时候或光耦为点亮状态时，Data[2] = 0x01
当 CTL2 端口的电平为低电平时候，或光耦为熄灭状态时，Data[3] = 0x00，为高电平时候货光耦为点亮状态时，Data[3] = 0x01

注意:本条左右限位反馈的内容，仅对除 102、107、201、301、302、303、303+、401 等版本的驱动器有效，其他版本驱动器因无 CTL 端口，故本条指令并无意义。

十三、预留：

十四、预留：

十五、在线监测：

本指令在任何状态下都会直接生效。
发送本数据包后，驱动器会返回一条一模一样的数据包。

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
0x23	0x0F	0x55	0x55	0x55	0x55	0x55	0x55	0x55	0x55