

《机械臂控制器软件接口手册》

版本号:1.0

作者:吴雪铭

联系方式:微信 K15837821186 , QQ 1362385699

1.前言

本机械臂控制器采用指令方式进行设备的管理控制，以便于后续的扩展和用户的使用。

指令分为 2 类，分别为基础指令和组合指令。基础指令是有控制器底层代码直接进行执行，用法简单高效的;组合指令则是根据不同的功能，使用不同组合的基础指令来进行一些特定任务的完成。

2.指令表一览

指令名称	编号	简介
NEW	0x00	新数据，获取轨迹数据的长度
PENDING	0x01	悬起态，读取轨迹数据
RUNING	0x02	执行态，执行轨迹
STOPPED	0x03	静止态，处于数据执行完毕的状态
UPLOAD_START	0x04	重新开启 Location 数据的上传，默认开启
UPLOAD_STOP	0x05	关闭 Location 数据的上传，默认开启
PWM_START	0x06	打开 PWM
PWM_STOP	0x07	关闭 PWM
USART_START	0x08	开启 USART 通信中断
USART_STOP	0x09	关闭 USART 通信中断
USART_SEND	0x0a	发送数据到串口
USART_RECV	0x0b	接收串口数据并上传至上位机
PINO_ON	0x0c	GPIO_0 高电平
PINO_OFF	0x0d	GPIO_0 低电平
PIN1_ON	0x0e	GPIO_1 高电平
PIN1_OFF	0x0f	GPIO_1 低电平
TOGGLE_ENABLE_PINS	0x10	ENABLE 使能翻转，包括夹具，可以做到失力和使能的切换

RETURN	0x11	返回零点，组合指令
JOINTS_COUNT	0x12	设置关节数量，默认为 6 轴，每次连接非 6 轴机械臂之后 2s 才能调用此命令，比较影响效率，尽量不要使用此接口，如果使用 7 轴、8 轴，可以联系控制器售后，直接在出场时设计成默认 7 轴、8 轴的机械臂控制器
LOCATION_SETTING	0x13	手动设置 location 的数值，给当前机械臂位置进行赋值

3. 指令用法

本章讲解指令的具体用法，和使用效果。每条指令前后都有用于数据校验的序列号和校验位。tag 则为指令码。

Sequence	Tag	Data	CRC
1byte	1byte	?byte	1byte

3.1 NEW 0x00 新路径数据指令

新路径数据指令，在有新路径时使用，其 Data 部分数据的取值范围为 0-0xffff，数据含义定义如下

指令	路径点数量的高 8 位	路径点数量的低 8 位
----	-------------	-------------

*此指令属于状态码

3.2 PENDING 0x01 悬起态指令

悬起态指令，轨迹路径点信息传送时使用，此时机械臂将保持静止状态。其 Data 部分数据则是一个一个轨迹点数据结构，数据定义可在 xxxRobot.h 文件中查到。数据结构定义如下

```
struct Point
{
    volatile int32_t duration;
    volatile int16_t numberOfFullPeriod[8];
    volatile int16_t restPeriod[8];
    volatile int16_t numberOfPeriod[8];
    volatile int32_t period[8];
    volatile int32_t position[8];
};
```

指令结构如下

指令	路径点 1	路径点 2	路径点 X
----	-------	-------	-------	-------

*此指令属于状态码

3.3 RUNNING 0x02 执行态指令

执行态指令，当轨迹点全部传输完毕后进行调用，以便启动机械臂来去执行轨迹。

没有 Data 部分，即 Data 部分数据无效。



*此指令属于状态码

3.4 STOPPED 0x03 停止态指令

停止态指令，此指令在需要紧急制动时使用，如果网络正常，机械臂将会瞬间停止。



*此指令属于状态码

3.5 UPLOAD_START 0x04 数据上传启动指令

开启机械臂当前位置信息反馈。



3.6 UPLOAD_STOP 0x05 数据上传停止指令

停止机械臂当前位置信息反馈。



3.7 PWM_START 0x06 脉宽调制(PWM)开启指令

用于 PWM 引脚的开启，用于舵机夹爪和步进电机夹爪抓取物体使用。使用之后，夹爪将会进行运动。主要通过操作 PWM 结构的实例 `pwm_handle` 进行控制。定义如下

```
struct PWM
{
    //PSC ARR CCR1 有效位数都为 16 位数，即需要控制在 0-0xffff 之间

    int32_t PSC; // PRC，预分频器系数

    int32_t ARR; // ARR，自动重载寄存器

    int32_t CCR1; // CCR1，捕获/比较寄存器 1，只使用通道 1

    int32_t PluseCount; // 执行脉冲的数量，只有此位不为 0 时有效，可为负数
};
```

其中 PSC ARR CCR1 都是 stm32 中 pwm 使用的标准接口，用户可以查询 stm32 pwm 资料来大致了解此部分功能的使用。PluseCount 执行脉冲的数量，在为 0 时，控制器将会将夹爪当作舵机进行控制；在不为 0 时，控制器将会将夹爪当作步进电机进行控制，其数值的正负号确定电机的转向，即顺时针旋转或逆时针旋转，而其绝对值则决定在当前位置下步进电机旋转的角度，这个角度则是由脉冲数进行映射。

例如，在驱动器调整为 6400/rad 时，PluseCount 设置为+3200，就说明电机将会正转 180 度。

发送格式如下

指令	脉宽调制实例
----	--------

3.8 PWM_STOP 0x07 脉宽调制(PWM)关闭指令

脉宽调制(PWM)关闭指令，通常用于舵机的控制，也可以用于步进电机夹爪的紧急制动。格式如下

指令

3.9 USART_START 0x08 串口(USART)开启指令

用于串口通信的开启，设置串口的读写模式、波特率等数据，通过从 stm32 的 hal 库引入的 UART_InitTypeDef 结构体进行参数的设置。介绍如下，此结构体，自 STM32 库函数中引入，除波特率外，通常赋值 0 即可 除 Mode 外，其他可直接参考 STM32 的 hal 库函数

```
struct UART_InitTypeDef
{
    uint32_t BaudRate;
    uint32_t WordLength;
    uint32_t StopBits;
    uint32_t Parity;
    uint32_t Mode;
    uint32_t HwFlowCtl;
    uint32_t OverSampling;
};
```

其中 Mode 数据定义和 stm32 中有些区别，0 为收发，1 为只收不发，2 为只发不收，若为其他数值，则认定为收发模式。

格式如下

指令	串口初始化实例
----	---------

3.10 USART_STOP 0x09 串口(USART)关闭指令

用于串口通信的关闭。使用后，将不会有任何控制器接收到的串口数据进行上传，格式为

指令

3.11 USART_SEND 0x0a 串口发送指令

控制器接收上位机的数据并通过串口向外设发送数据。串口数据长度在 1-255 之间格式为

指令	串口数据
----	------

3.12 USART_RECV 0x0b 串口接收指令

控制器通过串口接收外设的数据，并发送数据到上位机。有效数据和无效数据的总长度为 256。格式如下表

串口数据长度	有效数据	无效数据
--------	------	------

*此指令由控制器调用，上位机调用无效。

3.13 PIN0_ON 0x0c 拉高 GPIO_0 引脚电平指令

用于把控制器上面的 PIN0 接口置为高电平。格式为

指令

3.14 PIN0_OFF 0x0d 拉低 GPIO_0 引脚电平指令

用于把控制器上面的 PIN0 接口置为低电平。格式为

指令

3.15 PIN1_ON 0x0e 拉高 GPIO_1 引脚电平指令

用于把控制器上面的 PIN1 接口置为高电平。格式为

指令

3.16 PIN1_OFF 0x0f 拉低 GPIO_1 引脚电平指令

用于把控制器上面的 PIN1 接口置为高电平。格式为

指令

3.17 TOGGLE_ENABLE_PINS 0x10 使能翻转指令

用于机械臂和夹具，可以做到失力和使能的切换。格式为

指令

3.18 RETURN 0x11 机械臂返回零点

用于机械臂，返回机械零点。由 ROS 接口直接调用。主要通过读取绝对值编码器的数据，而进行返回机械零点的操作。可以直接参考本控制此部分的代码。

*此指令为组合指令

3.19 JOINTS_COUNT 0x12 关节数量设置指令

用于机械臂关节数量设置，默认为 6 轴，6 轴机械臂不需要再调用此指令。其它轴机械臂，只有在上位机和控制器网络连接成功 2s 后，才能调用此指令。此指令比较影响效率和用户体验。因此，如果有在 7 轴 8 轴机械臂上使用的需求，可以联系控制器官方售后，以便于在出场时就设计成默认为 7 轴机械臂的控制器。

指令	关节数量
----	------

3.20 LOACTION_SETTING 0x13 当前位置设置指令

用于机械臂校准,当机械臂当前的物理位置,和控制器系统有差异时,调用此指令,则可以将当前机械臂的物理位置,与此指令设置的机械臂位置信息(以脉冲数进行映射)进行对应。格式为

指令	关节位置信息实例
----	----------

其中关节位置信息的数据结构为

```
struct Location
{
    volatile int32_t position[8];
    volatile int8_t state;
};
```

此指令中 state 没有用。位置设置放在 position[8]中进行。