



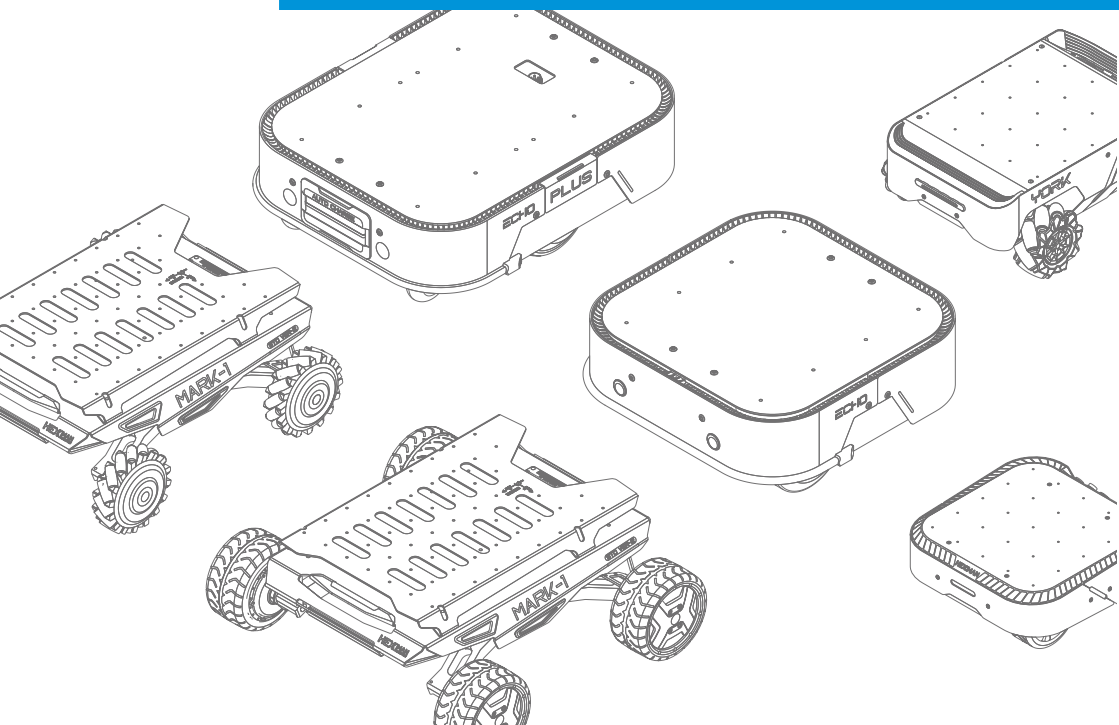
# XSTD

VEHICLE PART

标准化通讯协议  
底盘部分

用户手册

V1.1.1



### 重要安全信息

- 请在使用前对设备的使用电压、功率、安装等参数进行确认,超参数导致的问题不在保修范围内。
- 请评估使用环境的IP等级、温度和设备的适用性等要求,恶劣环境导致的问题不在保修范围内。
- 该设备并不提供防撞、防跌落、生物接近预警等稳定的相关安全功能,请对集成的产品进行安全评估,并符合对应的法规及认证,确保开发的产品没有重大安全隐患。
- 请阅读相关的保养要求,防止应为错误操作导致的不可逆转的损害,例如:电池过放、轮胎低气压、低压导致轮毂损坏、缺少润滑导致轴磨损等。
- 首次使用请将设备放置于安全、空旷环境中,没有载重物,按照操作说明步骤启动,并测试所有相关功能,如有问题请联系购买渠道服务人员。
- 当设备出现异常或发生意外时,请立即使用急停或关闭电源,避免造成二次损伤,并联系相关技术人员,不要自行拆卸。



注意:用户任何自行拆卸、改装、非正常使用以及自然灾害等不可抗力造成的损害将不在保修范围内。

对用户集成后的设备造成的安全事故,本产品不承担相应责任,请用户自行评估风险与可靠性测试,特此声明。

## 版本更新信息

V1.0.0: 起始版本

V1.0.1: 更新底盘状态设置, 增加刹车功能

V1.0.2: 增加零点校准指令

V1.1.0: 增加设置编号协议, 修改错误码, 增加特殊功能控制

V1.1.1: 增加底盘错误信息协议, 增加急停开关报警, 修改底盘状态信息反馈

## 目录

1. 协议介绍.....	5
1.1 构架简介.....	5
1.2 设备状态定义.....	5
1.3 设备的分类与识别.....	5
1.4 指令类型.....	5
2. CAN通讯协议.....	6
2.1 CAN数据结构.....	6
2.2 指令集查询说明.....	7
2.3 CAN指令目录.....	8
2.4 设备通用指令.....	9
2.4.1 设备重启/准备重启.....	9
2.4.2 软件版本查询.....	10
2.4.3 通用设置/通用设置成功.....	11
2.4.4 特殊状态复位 / 特殊复位成功.....	12
2.4.5 运动状态复位 / 运动复位成功.....	13
2.4.6 编号设置 / 编号设置成功.....	14
2.4.7 设备心跳包.....	15
2.5 设备专用指令.....	16
2.5.1.1 底盘状态设置 / 底盘状态信息.....	16
2.5.2 底盘运动控制 / 底盘运动信息.....	18
2.5.3 里程计反馈.....	19
2.5.4.1 遥控器信息发送使能 / 遥控器信息.....	20
2.5.4.2 拨动开关档位表.....	21
2.5.5.1 安全传感器信息发送使能 / 安全传感器信息.....	22
2.5.5.2 触碰开关表.....	22
2.5.6 驱动器运动信息使能 / 驱动器运动信息.....	23
2.5.7.1 驱动器状态信息使能 / 驱动器状态信息.....	24
2.5.7.2 驱动器特殊状态表.....	24
2.5.8 底盘零点较准.....	25
2.5.9 底盘错误信息.....	26
3. 串口通讯协议.....	28
3.1.1 串口基本参数.....	28
3.1.2 串口帧结构.....	28
3.2 串口通讯案例.....	29
3.2.1 案例底盘ID.....	29
3.2.2 串口发送到设备案例表.....	29
3.2.3 串口从设备接受案例表.....	29
4. 软件支持.....	30
4.1 XVIEW测试系统.....	30
4.1 XROS支持.....	31

## 1.1 框架简介

XSTD是一套基于CAN网络,可扩展到串口使用的通讯协议规范。该规范为支持多设备和模组设计,在保持简洁易懂的同时提供给各模组充分的拓展空间,并支持多个同种设备同时在线工作。协议对同类型设备采用统一规范,牺牲少量效率换来极大的便捷性,非常便于早期功能开发与学习。

## 1.2 设备状态定义

### 使能(通讯设备默认状态)

- 设备所有功能正常运行。
- 对CAN和COM指令做出反馈。
- 默认关闭次要反馈,可通过指令开启。

### 失能(普通设备默认状态)

- 设备进入通讯静默状态,内部任然运行。
- 只接受通用指令,不接受专用指令。
- 只发送心跳反馈和通用指令的反馈,不发送专用指令反馈。
- 射频遥控器等设备自身安装的控制设备任然有效,并拥有更高优先级。

## 1.3 设备的分类与识别

### 每个设备有三个主要识别属性:大类、型号、编号

- 大类是按设备类型共分为30类,编码从0x01~0x1E,0x1F作为广播编码。
- 型号是从0x01~0xFE的序号,在同一大类中,型号是确定且唯一的,与设备名称绑定的,0xFF作为广播编码。
- 编号是从0x01~0xFE的序号,可以在XVIEW上设置,用于区分同大类且同型号的设备,0xFF作为广播编码。

在使用指令时,需要对应的大类、型号、编号,对应的大类与型号代码请在对应设备的用户手册中查询。

## 1.4 指令类型

### 指令分为通用指令和专用指令

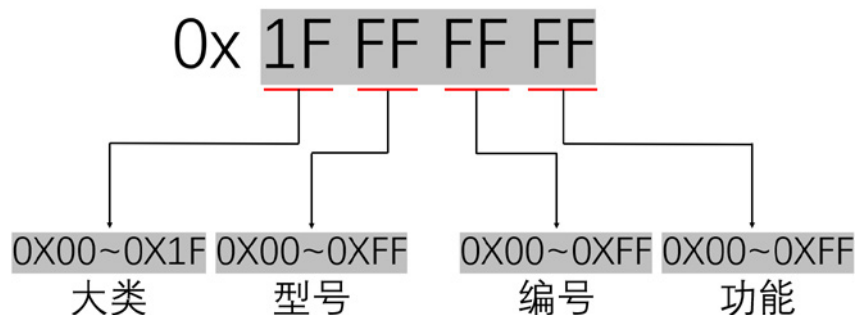
- 通用指令只通过CAN ID区分接受设备,数据内容对所有大类都是相同的,主要用于基础功能设置和反馈
- 专用指令是按大类区分,每个大类的功能是不同的,同个大类中功能按需取用
- 通用指令的功能编码从0x 01开始,对应反馈从 0x A1开始(心跳包反馈为0x B0)
- 专用指令的功能编码从0x 11开始,对应反馈从 0x B1开始



## 2.1 CAN数据结构

产品中CAN通讯采用CAN2.0B(扩展帧),通讯波特率为500K。

CAN ID 编码规范:



CAN 数据位:Byte[0]~Byte[7]:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]
---------	---------	---------	---------	---------	---------	---------	---------

CAN使用的CAN信息为上述提到的CANID与CAN\_DATA,总计12个字节数据。使用时,针对不同CANID进行相应的CAN\_DATA数据传输。

CANID中大类与型号是设备的固定参数,数值不会改变,编号默认为1,当有多个同类设备同时使用时,可以通过通用指令修改。

CAN数据帧					
数据位	ID[3]	ID[2]	ID[1]	ID[0]	DATA[0~7]
数据类型	unsigned int8	unsigned int8	unsigned int8	unsigned int8	8 x unsigned int8
数据说明	大类	型号	编号	功能	8字节CAN数据

底盘系列固定大类 **01**

默认编号为 **01**

底盘系列CANID基础格式为 **01 XX 01 XX**

## 2.2 指令集查询说明

CAN协议分为设备通用指令与设备专用指令。通用指令操作对系统所有模块配件均适用,常用于设备启动,系统升级,版本查询,系统复位等;设备专用指令专属于不同设备模块,不同模块根据需要选择使用部分专用指令,对特殊设备可定制专用指令。

协议目录									
命令协议集					反馈协议集				
对象	命令内容	X 主类 0	Y 子类 0-F	Z 辅助类 1-F	反馈内容	X 主类 0	Y 子类 0-F	Z 辅助类 1-F	是否周期性
所有设备通用	设备重启	0	0	1	设备重启	0	0	2	否
	进入升级模式	0	1	1	/	/	/	/	/
	升级模式专用	0	1	2-F	升级模式专用	0	1	2-F	/
	软件版本查询	0	2	1	设置成功	0	2	2	否
	通用设置	0	3	1	复位成功	0	4	2	否
移动底盘类设备	特殊状态复位	0	4	1	底盘运动控制	1	x	2	是
	/	/	/	/	底盘运动信息	1	x	3	是
	底盘状态设置	1	x	3	底盘状态信息	1	x	4	是
	/	/	/	/	主里程计	1	x	5	是
	使能周期发送	1	x	6	次里程计	1	x	6	可选
通讯设备	使能周期发送	1	x	7	车载安全传感器	1	x	7	可选
	使能周期发送	1	x	8	驱动器运动信息	1	x	8	可选
	使能周期发送	1	x	9	驱动器状态信息	1	x	9	可选
	/	/	/	/	通讯设备心跳	7	x	1	是

通用指令

专用指令

CAN闭环控制系统,对绝大多数指令均有应答机制。即命令帧,反馈帧一一对应。根据反馈帧实时监控数据流准确性。

协议目录									
命令协议集					反馈协议集				
对象	命令内容	X 主类 0	Y 子类 0-F	Z 辅助类 1-F	反馈内容	X 主类 0	Y 子类 0-F	Z 辅助类 1-F	是否周期性
所有设备通用	设备重启	0	0	1	准备重启	0	0	2	否
	进入升级模式	0	1	1	/	/	/	/	/
	升级模式专用	0	1	2-F	升级模式专用	0	1	2-F	/
	软件版本查询	0	2	1	软件版本反馈	0	2	2	否
	通用设置	0	3	1	设置成功	0	3	2	否
移动底盘类设备	特殊状态复位	0	4	1	复位成功	0	4	2	否
	/	/	/	/	反馈内容	X 设备类 1-7	Y 设备编号 1-F	Z 功能类 1-F	是否周期性
	底盘运动控制	1	x	2	底盘设备心跳	1	x	1	是
	底盘状态设置	1	x	3	底盘运动信息	1	x	2	是
	/	/	/	/	底盘状态信息	1	x	3	是
通讯设备	使能周期发送	1	x	6	主里程计	1	x	4	是
	使能周期发送	1	x	7	次里程计	1	x	5	是
	使能周期发送	1	x	8	车载安全传感器	1	x	6	可选
	使能周期发送	1	x	9	驱动器运动信息	1	x	7	可选
	使能周期发送	1	x	9	驱动器状态信息	1	x	8	可选
					通讯设备心跳	7	x	1	是

命令指令

反馈指令

## 2.3 CAN指令目录

命令协议集						反馈协议集					
对象	命令内容	大类 00~1F	型号 00~FF	编号 00~FF	功能 00~FF	命令内容	大类 00~1F	型号 00~FF	编号 00~FF	功能 00~FF	周期性
所有设备通用	设备重启	01	XX	XX(默认01)	01	准备重启	01	XX	XX(默认01)	A1	否
	软件版本查询	01	XX	XX(默认01)	02	软件版本反馈	01	XX	XX(默认01)	A2	否
	通用设置	01	XX	XX(默认01)	03	通用设置成功	01	XX	XX(默认01)	A3	否
	特殊状态复位	01	XX	XX(默认01)	04	特殊复位成功	01	XX	XX(默认01)	A4	否
	运动状态复位	01	XX	XX(默认01)	05	运动复位成功	01	XX	XX(默认01)	A5	否
	编号设置	01	XX	XX(默认01)	06	编号设置成功	01	XX	XX(默认01)	A6	否
	/	/	/	/	/	心跳包	01	XX	XX(默认01)	B0	是
对象	命令内容	大类 00~1F	型号 00~FF	编号 00~FF	功能 00~FF	命令内容	大类 00~1F	型号 00~FF	编号 00~FF	功能 00~FF	周期性
移动	底盘状态设置	01	XX	XX(默认01)	11	底盘状态信息	01	XX	XX(默认01)	B1	是
	底盘运动控制	01	XX	XX(默认01)	12	底盘运动信息	01	XX	XX(默认01)	B2	是
	/	/	/	/	/	主里程计	01	XX	XX(默认01)	B3	是
底盘	/	/	/	/	/	次里程计	01	XX	XX(默认01)	B4	是
	遥控器信息发送使能	01	XX	XX(默认01)	15	车载遥控器信息	01	XX	XX(默认01)	B5	使能后是
	安全传感器发送使能	01	XX	XX(默认01)	16	车载安全传感器信息	01	XX	XX(默认01)	B6	使能后是
	驱动器运动信息发送使能	01	XX	XX(默认01)	17	驱动器运动信息	01	XX	XX(默认01)	B7	使能后是
	驱动器状态信息发送使能	01	XX	XX(默认01)	18	驱动器状态信息	01	XX	XX(默认01)	B8	使能后是
	底盘零点校准	01	XX	XX(默认01)	19	零点校准信息	01	XX	XX(默认01)	B9	否

## 2.4 设备通用指令

### 2.4.1 设备重启/准备重启

设备重启				
发送方	接收方	CAN ID	数据长度	周期(ms)
外部	本设备	0x 01 XX XX 01	8	/
数据位	功能	数据类型	说明	
byte[0]	大类	unsigned int8	恒为0x01	
byte[1]	型号	unsigned int8	查询设备用户手册获得	
byte[2]	编号	unsigned int8	默认为0x01	
byte[3~7]	/	/	/	

注:当系统中有多台机器同时使用时,可更改设置设备编号,设备编号默认为0x01。发送指令数据长度为8,无效位补0(后文同)。

准备重启				
发送方	接收方	CAN ID	数据长度	周期(ms)
本设备	外部	0x 01 XX XX A1	0	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

注:设备重启指令对系统进行重启操作,当设备接收到指令后进行准备重启反馈应答,系统重启,两秒后蜂鸣器响三声,重启完毕。

举例:向型号0x02 编号0x03的底盘发送CANID:0x01 02 03 01 数据:01 02 03 00 00 00 00 00,系统重启。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位(L->H)
设备	0x01 02 03 01	数据帧	扩展帧	8	01 02 03 00 00 00 00 00
上位机	0x01 02 03 A1	数据帧	扩展帧	0	/

## 2.4.2 软件版本查询

软件版本查询				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 02	8	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

软件版本反馈				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX A2	8	/
数据位	功能	数据类型	说明	
byte[0]	硬件版本字符[0]	unsigned int8 ASCII	/	
byte[1]	硬件版本字符[1]	unsigned int8 ASCII	/	
byte[2]	硬件版本字符[2]	unsigned int8 ASCII	/	
byte[3]	硬件版本字符[3]	unsigned int8 ASCII	/	
byte[4]	硬件版本字符[4]	unsigned int8 ASCII	/	
byte[5]	硬件版本字符[5]	unsigned int8 ASCII	/	
byte[6]	硬件版本字符[6]	unsigned int8 ASCII	/	
byte[7]	硬件版本字符[7]	unsigned int8 ASCII	/	

注: 读取系统版本号。

举例: 向型号0x02 编号0x03 的底盘发送CANID:0x01 02 03 02 数据:00 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 02	数据帧	扩展帧	8	00 00 00 00 00 00 00 00
上位机	0x01 02 03 A2	数据帧	扩展帧	8	XX XX XX XX XX XX XX XX

## 2.4.3 通用设置/通用设置成功

通用设置				
发送方	接收方	CAN ID	数据长度	周期(ms)
外部	本设备	0x01 XX XX 03	8	/
数据位	功能	数据类型	说明	
byte[0]	大类	unsigned int8	恒为0x01	
byte[1]	型号	unsigned int8	查询设备用户手册获得	
byte[2]	编号	unsigned int8	默认为0x01	
byte[3]	使能	unsigned int8	0x00:关闭设备 / 0x01:开启设备	
byte[4~7]	/	/	/	

通用设置成功				
发送方	接收方	CAN ID	数据长度	周期(ms)
本设备	外部	0x 01 XX XX A3	0	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

注:通用设置指令用于对设备进行设备使能操作。

底盘默认上电是失能状态。

举例:使能设备型号0x02 编号0x03 的底盘,向系统ID:0x01 02 03 03发送指令01 02 03 01 00 00 00 00;设备使能。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位(L->H)
设备	0x01 02 03 03	数据帧	扩展帧	8	01 02 03 01 00 00 00 00
上位机	0x01 02 03 A3	数据帧	扩展帧	0	/

### 2.4.4 特殊状态复位 / 特殊复位成功

特殊状态复位				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 04	8	/
数据位	功能	数据类型	说明	
byte[0]	状态清除	unsigned int8	0xCC:清除 / 其他值无操作	
byte[1~7]	/	/	/	

特殊复位成功				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX A4	0	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

注:特殊状态复位用于系统异常(发生错误)时清错重置。

举例:向型号0x02 编号0x03 的底盘发送CANID:0x01 02 03 04 数据:CC 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 04	数据帧	扩展帧	8	CC 00 00 00 00 00 00 00
上位机	0x01 02 03 A4	数据帧	扩展帧	0	/

### 2.4.5 运动状态复位 / 运动复位成功

特殊状态复位				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 05	8	/
数据位	功能	数据类型	说明	
byte[0]	运动状态清除	unsigned int8	0xCC:清除 / 其他值无操作	
byte[1~7]	/	/	/	

特殊复位成功				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX A5	0	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

注:运动状态复位用于系统重置运动数据(例如里程计数据)。

举例:向型号0x02 编号0x03 的底盘发送CANID:0x01 02 03 05 数据:CC 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 05	数据帧	扩展帧	8	CC 00 00 00 00 00 00 00
上位机	0x01 02 03 A5	数据帧	扩展帧	0	/

## 2.4.6 编号设置

通用设置				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x01 XX XX 06	8	/
数据位	功能	数据类型	说明	
byte[0]	大类	unsigned int8	恒为0x01	
byte[1]	型号	unsigned int8	查询设备用户手册获得	
byte[2]	编号	unsigned int8	默认为0x01	
byte[3]	新编号	unsigned int8	如果不改变编号则与原编号相同	
byte[4~7]	/	/	/	

通用设置成功				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x01 XX XX A6	0	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

注: 编号设置指令用于对设备设置新编号。当系统只有单一设备时, 编号始终默认为0x01, 当系统中有多台设备时可自定义设置设备编号。

举例: 设备型号0x02 编号0x03 的底盘更改设备编号为0x02, 向系统ID:0x01 02 03 06发送指令01 02 03 02 00 00 00 00; 设备编号由0x03切换为0x02。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 06	数据帧	扩展帧	8	01 02 03 02 00 00 00 00
上位机	0x01 02 03 A6	数据帧	扩展帧	0	/

## 2.4.7 设备心跳包

心跳包				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B0	1	500
数据位	功能	数据类型	说明	
byte[0]	使能状态	unsigned int8	0x00:已关闭 / 0x01:已使能	
byte[1~7]	/	/	/	

注:心跳包主要用于确认设备在线状态,是在任何状态下一定保持500ms的发送周期,也是确认设备是否使能的关键帧。

举例:型号0x02 编号0x03 使能的底盘反馈心跳包。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	/	/	/	/	/
上位机	0x01 02 03 B0	数据帧	扩展帧	1	01

## 2.5 设备专用指令

### 2.5.1 底盘状态设置 / 底盘状态信息

底盘状态设置				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 11	8	/
数据位	功能	数据类型	说明	
byte[0]	设置控制模式	unsigned int8	0x00:待机 / 0x01:遥控 / 0x02:CAN控制 / 0x03:自由	
byte[1]	蜂鸣器使能	unsigned int8	0:关闭 / 1:开启 (默认)	
byte[2]	使能刹车	unsigned int8	0x00: 松开 (无刹车则默认) / 0x01: 锁止 (有刹车则默认)	
byte[3]	使能特殊功能	unsigned int8	0x00: 关闭 (默认) / 0x01: 开启	
byte[4~7]	/	/	/	

底盘状态信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B1	6	100
数据位	功能	数据类型	说明	
byte[0]	设备状态	unsigned int8	0x00:正常 / 0x01:异常 (出现错误报警)	
byte[1]	控制模式	unsigned int8	0x00:待机 / 0x01:遥控 / 0x02:CAN控制 / 0x03:自由	
byte[2]	电池电压低8位	unsigned int16	单位:0.1V	
byte[3]	电池电压高8位			
byte[4]	蜂鸣器使能状态	unsigned int8	0:关闭 / 1:开启 (默认)	
byte[5]	遥控器状态	unsigned int8	0: 在线 / 1: 离线	
byte[6]	刹车状态	unsigned int8	0x00: 松开 (无刹车则默认) / 0x01: 锁止 (有刹车则默认)	
byte[7]	特殊功能状态	unsigned int8	0x00: 关闭 (默认) / 0x01: 开启	

注:当使用有线方式控制底盘时,需先切换控制模式至CAN模式。如果使用了遥控器,请将遥控器SWA拨到最下面的CAN模式,遥控器说明请看设备的用户手册。

刹车使能只适用于有刹车功能的底盘,无刹车功能的默认该位保持0x00,有刹车功能的默认开启刹车。

特殊功能根据底盘而不同,例如阿克曼底盘为差速锁功能,不是所有底盘都有特殊功能。特殊功能也可由遥控器开启,参见对应设备的用户手册。

举例:型号0x02 编号0x03 的底盘更改为CAN模式控制并使能蜂鸣器,发送CANID:0x 01 02 03 11 数据:02 01 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位(L->H)
设备	0x01 02 03 11	数据帧	扩展帧	8	02 01 00 00 00 00 00 00
上位机	0x01 02 03 B1	数据帧	扩展帧	7	00 02 XX XX 01 00 00

## 2.5.2 底盘运动控制 / 底盘运动信息

底盘运动控制				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 12	8	50 (推荐值)
数据位	功能	数据类型	说明	
byte[0]	x方向速度低8位	signed int16	单位:mm/s	
byte[1]	x方向速度高8位			
byte[2]	y方向速度低8位 <sup>®</sup>	signed int16	单位:mm/s	
byte[3]	y方向速度高8位 <sup>®</sup>			
byte[4]	角速度低8位	signed int16	单位:0.001rad/s	
byte[5]	角速度高8位			
byte[6~7]	/	/	/	

底盘运动信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B2	6	20
数据位	功能	数据类型	说明	
byte[0]	x方向速度低8位	signed int16	单位:mm/s	
byte[1]	x方向速度高8位			
byte[2]	y方向速度低8位 <sup>®</sup>	signed int16	单位:mm/s	
byte[3]	y方向速度高8位 <sup>®</sup>			
byte[4]	角速度低8位	signed int16	单位:0.001rad/s	
byte[5]	角速度高8位			
byte[6~7]	/	/	/	

- 注：1. Y方向速度只对麦克纳姆轮系等具有平移功能的车型才有效,其他车型默认为0  
 2. 由于电机执行有精度问题,系统反馈值会与设定值有细微偏差  
 3. 数据使用有符号变量,符合C语言标准

举例:型号0x02 编号0x03 的底盘以0.5m/s的速度前进,同时以-0.1rad/s的角度转动,发送CANID:0x01 02 03 12 数据:F4 01 00 00 9C FF 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 12	数据帧	扩展帧	8	F4 01 00 00 9C FF 00 00
上位机	0x01 02 03 B2	数据帧	扩展帧	6	F4 01 00 00 9C FF

## 2.5.3 里程计反馈

主里程计				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B3	8	20
数据位	功能	数据类型	说明	
byte[0]	左轮里程计[7:0]	signed int32	四轮差速系统中为左侧平均里程 单位:mm	
byte[1]	左轮里程计[15:8]			
byte[2]	左轮里程计[23:16]			
byte[3]	左轮里程计[31:24]			
byte[4]	右轮里程计[7:0]	signed int32	四轮差速系统中为右侧平均里程 单位:mm	
byte[5]	右轮里程计[15:8]			
byte[6]	右轮里程计[23:16]			
byte[7]	右轮里程计[31:24]			

次里程计				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B4	8	20
数据位	功能	数据类型	说明	
byte[0]	左后轮里程计[7:0] <sup>Ⓢ</sup>	signed int32	适用全向轮和麦轮系统 单位:mm	
byte[1]	左后轮里程计[15:8] <sup>Ⓢ</sup>			
byte[2]	左后轮里程计[23:16] <sup>Ⓢ</sup>			
byte[3]	左后轮里程计[31:24] <sup>Ⓢ</sup>			
byte[4]	右后轮里程计[7:0] <sup>Ⓢ</sup>	signed int32	适用全向轮和麦轮系统 单位:mm	
byte[5]	右后轮里程计[15:8] <sup>Ⓢ</sup>			
byte[6]	右后轮里程计[23:16] <sup>Ⓢ</sup>			
byte[7]	右后轮里程计[31:24] <sup>Ⓢ</sup>			

注：两轮差速和四轮差速只使用主里程计，全向轮和麦克纳姆轮系统同时使用两组里程计

举例：型号0x02 编号0x03 的底盘反馈里程计数据。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	/	/	/	/	/
上位机	0x01 02 03 B3	数据帧	扩展帧	8	XX XX XX XX XX XX XX XX

## 2.5.4.1 遥控器信息发送使能 / 遥控器信息

遥控器信息发送使能				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 15	8	/
数据位	功能	数据类型	说明	
byte[0]	使能周期发送	unsigned int8	0:关闭(默认) / 1~20:20ms / 21~255:设定值	
byte[1~7]	/	/	/	

遥控器信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B5	7	20~255(默认关闭)
数据位	功能	数据类型	说明	
byte[0]	拨动开关	unsigned int8	按照 拨动开关档位表	
byte[1]	左摇杆水平轴	signed int8	范围:-100~100	
byte[2]	左摇杆垂直轴	signed int8	范围:-100~100	
byte[3]	右摇杆水平轴	signed int8	范围:-100~100	
byte[4]	右摇杆垂直轴	signed int8	范围:-100~100	
byte[5]	左旋钮VRA	signed int8	范围:-100~100	
byte[6]	右旋钮VRA	signed int8	范围:-100~100	
byte[7]	/	/	/	

注：仅限反馈官方配套车载射频遥控器数据，车载射频遥控器为选配设备，请联系客服咨询

### 2.5.4.2 拨动开关档位表

拨动开关档位表		
位	状态值	状态说明
bit[0]	0: 失效 / 1: 上档	SWA
bit[1]	2: 中档 / 3: 下档	
bit[2]	0: 失效 / 1: 上档	SWB
bit[3]	2: 中档 / 3: 下档	
bit[4]	0: 失效 / 1: 上档	SWC
bit[5]	2: 中档 / 3: 下档	
bit[6]	0: 失效 / 1: 上档	SWD
bit[7]	2: 中档 / 3: 下档	

举例: 型号0x02 编号0x03 的底盘设置100ms遥控反馈, 发送CANID:0x01 02 03 15 数据: 64 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 15	数据帧	扩展帧	8	64 00 00 00 00 00 00 00
上位机	0x01 02 03 B5	数据帧	扩展帧	7	XX XX XX XX XX XX XX

### 2.5.5.1 安全传感器信息发送使能 /安全传感器信息

安全传感器发送使能				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 16	8	/
数据位	功能	数据类型	说明	
byte[0]	使能周期发送	unsigned int8	0:关闭(默认) / 1~20:20ms / 21~255:设定值	
byte[1~7]	/	/	/	

安全传感器信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B6	1	20~255(默认关闭)
数据位	功能	数据类型	说明	
byte[0]	触边开关	unsigned int8	按位对应开关编号	
byte[1~7]	/	/	/	

注:仅限安装了防撞边条或触碰传感器的设备才反馈该数据

### 2.5.5.2 触碰开关表

触碰开关表		
位	状态值	状态说明
bit[0]	0: 未触发 / 1: 触发	触碰开关[0]
bit[1]	0: 未触发 / 1: 触发	触碰开关[1]
bit[2]	0: 未触发 / 1: 触发	触碰开关[2]
bit[3]	0: 未触发 / 1: 触发	触碰开关[3]
bit[4]	0: 未触发 / 1: 触发	触碰开关[4]
bit[5]	0: 未触发 / 1: 触发	触碰开关[5]
bit[6]	0: 未触发 / 1: 触发	触碰开关[6]
bit[7]	0: 未触发 / 1: 触发	触碰开关[7]

举例:型号0x02 编号0x03的底盘设置100ms安全反馈,发送CANID:0x01 02 03 16 数据:64 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位(L->H)
设备	0x01 02 03 16	数据帧	扩展帧	8	64 00 00 00 00 00 00 00
上位机	0x01 02 03 B6	数据帧	扩展帧	1	XX

## 2.5.6 驱动器运动信息使能 / 驱动器运动信息

驱动器运动信息使能				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 17	8	/
数据位	功能	数据类型	说明	
byte[0]	使能周期发送	unsigned int8	0:关闭(默认) / 1~20:20ms / 21~255:设定值	
byte[1~7]	/	/	/	

驱动器运动信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B7	7	20~255(默认关闭)
数据位	功能	数据类型	说明	
byte[0]	驱动编号	unsigned int8	0~7 (轮流发送)	
byte[1]	转速低8位	signed int16	单位:PRM	
byte[2]	转速高8位			
byte[3]	位置值 [7:0]	signed int32	单位:脉冲数	
byte[4]	位置值 [15:8]			
byte[5]	位置值 [23:16]			
byte[6]	位置值 [31:24]			
byte[7]	/	/	/	

注：此数据为系统最底层来自驱动器的直接记数，用户可根据原始数据自行求解里程计和速度

举例：型号0x02 编号0x03 的底盘设置100ms驱动器运动反馈，发送CANID:0x01 02 03 17 数据:64 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 17	数据帧	扩展帧	8	64 00 00 00 00 00 00 00
上位机	0x01 02 03 B7	数据帧	扩展帧	7	XX XX XX XX XX XX XX

## 2.5.7.1 驱动器状态信息使能 / 驱动器状态信息

驱动器状态信息使能				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x01 XX XX 18	8	/
数据位	功能	数据类型	说明	
byte[0]	使能周期发送	unsigned int8	0:关闭(默认) / 1~20:20ms / 21~255:设定值	
byte[1~7]	/	/	/	

驱动器状态信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x01 XX XX B8	8	20~255(默认关闭)
数据位	功能	数据类型	说明	
byte[0]	驱动编号	unsigned int8	0~7(轮流发送)	
byte[1]	电压低8位	unsigned int16	单位:0.1V	
byte[2]	电压高8位			
byte[3]	电流低8位	unsigned int16	单位:0.1A	
byte[4]	电流高8位			
byte[5]	驱动器温度	signed int8	单位:1°C	
byte[6]	电机温度	signed int8	单位:1°C	
byte[7]	驱动器特殊状态	unsigned int8	按照 驱动器特殊状态表	

## 2.5.7.2 驱动器特殊状态表

触碰开关表		
位	状态值	状态说明
bit[0]	0: 正常 / 1: 过低	驱动低压报警
bit[1]	0: 正常 / 1: 过温	驱动过温报警
bit[2]	0: 正常 / 1: 过流	电机过流报警
bit[3]	0: 正常 / 1: 过温	电机过温报警
bit[4~7]	/	/

举例: 型号0x02 编号0x03 的底盘设置100ms驱动器状态反馈, 发送CANID: 0x01 02 03 18 数据: 64 00 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 18	数据帧	扩展帧	8	64 00 00 00 00 00 00 00
上位机	0x01 02 03 B8	数据帧	扩展帧	8	XX XX XX XX XX XX XX XX

## 2.5.8 底盘零点较准

底盘零点较准				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 19	8	/
数据位	功能	数据类型	说明	
byte[0]	零点偏移量1[7:0]	signed int32	单位: mrad	
byte[1]	零点偏移量1[15:8]			
byte[2]	零点偏移量1[23:16]			
byte[3]	零点偏移量1[31:24]			
byte[4]	零点偏移量2[7:0]	signed int32	单位: mrad	
byte[5]	零点偏移量2[15:8]			
byte[6]	零点偏移量2[23:16]			
byte[7]	零点偏移量2[31:24]			

零点较准信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX B9	8	/
数据位	功能	数据类型	说明	
byte[0]	零点偏移量1[7:0]	signed int32	单位: mrad	
byte[1]	零点偏移量1[15:8]			
byte[2]	零点偏移量1[23:16]			
byte[3]	零点偏移量1[31:24]			
byte[4]	零点偏移量2[7:0]	signed int32	单位: mrad	
byte[5]	零点偏移量2[15:8]			
byte[6]	零点偏移量2[23:16]			
byte[7]	零点偏移量2[31:24]			

注: 单阿克曼使用偏移量1; 前后阿克曼前转向用1, 后转向用2; 双舵轮左边用1, 右边用2。

举例: 型号0x02 编号0x03 的单阿克曼底盘设置1000mrad偏移, 发送CANID: 0x01 02 03 19 数据: E8 03 00 00 00 00 00 00。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	0x01 02 03 19	数据帧	扩展帧	8	E8 03 00 00 00 00 00 00
上位机	0x01 02 03 B9	数据帧	扩展帧	8	E8 03 00 00 00 00 00 00

## 2.5.9 底盘错误信息

错误状态反馈				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX BA	4	500
数据位	功能	数据类型	说明	
byte[0]	电机故障	unsigned int8	查看电机故障表	
byte[1]	驱动故障	unsigned int8	查看驱动故障表	
byte[2]	通讯故障	unsigned int8	查看通讯故障表	
byte[3]	其他故障	unsigned int8	查看其他故障表	
byte[4~7]	/	/	/	

电机故障表		
位	状态值	状态说明
bit[0]	0: 正常 / 1: 故障	电机过流报警
bit[1]	0: 正常 / 1: 故障	电机过温报警
bit[2]	0: 正常 / 1: 故障	编码器故障
bit[3]	0: 正常 / 1: 故障	霍尔传感器故障
bit[4~7]	/	/

驱动故障表		
位	状态值	状态说明
bit[0]	0: 正常 / 1: 故障	驱动低压报警
bit[1]	0: 正常 / 1: 故障	驱动过温报警
bit[2~7]	/	/

通讯故障表		
位	状态值	状态说明
bit[0]	0: 在线 / 1: 离线	1号驱动通讯故障
bit[1]	0: 在线 / 1: 离线	2号驱动通讯故障
bit[2]	0: 在线 / 1: 离线	3号驱动通讯故障
bit[3]	0: 在线 / 1: 离线	4号驱动通讯故障
bit[4~7]	/	/

其他故障表		
位	状态值	状态说明
bit[0]	0: 正常 / 1: 警告	电池低电压提醒 (<15%)
bit[1]	0: 正常 / 1: 故障	电池低电压报警 (<5%)
bit[2]	0: 未触发 / 1: 触发	触碰传感器触发
bit[3]	0: 未触发 / 1: 触发	急停开关
bit[4~7]	/	/

举例: 型号0x02 编号0x03 的底盘反馈触碰传感器触发。

发送实例					
传输目标	CAN ID	帧格式	帧类型	数据长度	数据位 (L->H)
设备	/	/	/	/	/
上位机	0x01 02 03 BA	数据帧	扩展帧	4	00 00 00 04

## 3.1 串口通讯协议

串口通讯协议是对CAN协议的打包封装, 添加了帧头、帧长、帧ID以及校验和。指令细节和使用案例请查阅 [CAN通讯协议部分](#)

### 3.1.1 串口基本参数

项目	参数
波特率	460800
校验	无校验
数据位长度	8
停止位	1

### 3.1.2 串口帧结构

为简化系统设计, 提高系统复用率, 将串口byte3-byte14与CAN协议中使用到的十位数据保持一致。起始位, 帧长, 帧ID为默认值, **校验和为前15字节总和后的值取末尾1字节数据。**

数据定义	HEAD	LEN	DATA[0-11]	FRAME_ID	SUM
数据类型	Uin8	Uin8	12* Uin8	Uin8	Uin8
说明	0x55	0x10	12位CAN报文数据 串口报文是对CAN报文的封装	帧ID可固定也可累加, 作为连贯数据的序列戳	前15位内容的累加和, 再取最低位的Uin8数据

串口数据位	DATA[0]	DATA[1]	DATA[2]	DATA[3]	DATA[4~11]
CAN数据位	ID[3]	ID[2]	ID[1]	ID[0]	DATA[0~7]
数据类型	Uin8	Uin8	Uin8	Uin8	8 x Uin8
数据说明	大类	型号	编号	功能	8字节CAN数据

## 3.2 串口通讯案例

以下以ECHO-PLUS底盘为例，列出常用的底盘控制指令

### 3.2.1 案例底盘ID

底盘固定大类

01

案例型号

03

案例编号

05

串口演示案例 CANID 固定格式

01 03 05 XX

### 3.2.2 串口发送到设备案例表

起始位	帧长度	CANID[0~3]	CAN数据[4~11]	帧ID	和校验	说明
55	10	01 03 05 01	01 03 05 00 00 00 00 00	01	79	设备重启
55	10	01 03 05 03	01 03 05 01 00 00 00 00	01	7C	设备使能
55	10	01 03 05 04	CC 00 00 00 00 00 00 00	01	3F	复位特殊状态
55	10	01 03 05 05	CC 00 00 00 00 00 00 00	01	40	复位运动状态
55	10	01 03 05 11	02 01 00 00 00 00 00 00	01	83	设置CAN模式
55	10	01 03 05 12	F4 01 00 00 00 00 00 00	01	76	直线速度 0.5 m/s
55	10	01 03 05 12	00 00 00 00 DC 05 00 00	01	62	旋转速度 1.5 rad/s
55	10	01 03 05 15	64 00 00 00 00 00 00 00	01	E8	遥控器反馈100ms
55	10	01 03 05 17	32 00 00 00 00 00 00 00	01	B8	驱动运动反馈50ms

### 3.2.3 串口从设备接受案例表

起始位	帧长度	CANID[0~3]	CAN数据[4~11]	帧ID	和校验	说明
55	08	01 03 05 A1	/	01	08	准备重启
55	09	01 03 05 B0	01	01	19	心跳包
55	08	01 03 05 A4	/	01	0B	特殊状态复位成功
55	08	01 03 05 A5	/	01	0C	运动状态复位成功
55	0E	01 03 05 B1	00 02 09 01 01 00 00	01	2B	底盘状态信息(24.5V)
55	0E	01 03 05 B2	F4 01 00 00 00 00	01	14	底盘运动信息(0.5m/s)
55	10	01 03 05 B3	DC 05 00 00 D0 07 00 00	01	DA	主里程计(左1.5m右2m)
55	0F	01 03 05 B5	03 00 00 00 00 00 00	01	26	遥控器信息(SWA下)
55	0F	01 03 05 B7	02 64 00 33 44 55 00	01	57	2号驱动信息(100RPM)

## 4.1 XVIEW测试系统

该系统是为调试和开发设计的一款人机交互软件,能在线测试各个设备功能并实时反馈设备数据,还可用于串口和CAN总线监控和测试。**该系统需要配合CAN-COM HUB使用,如没有HUB请联系客服。**使用步骤如下:

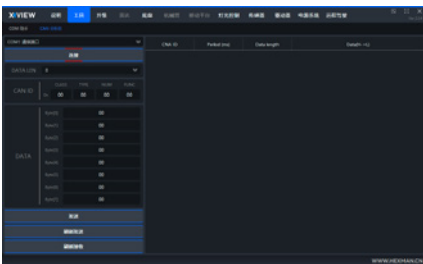
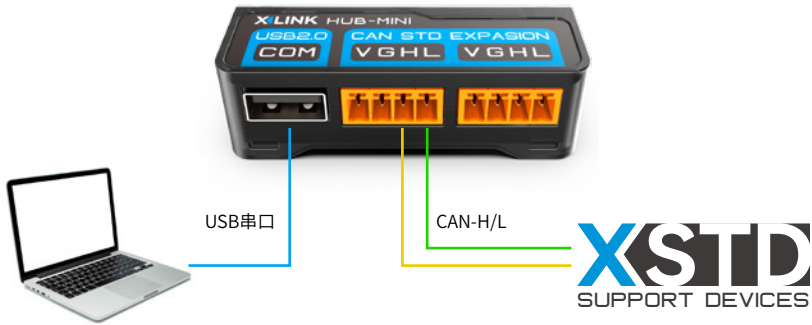
1: 下载驱动包CP210x\_Universal\_Windows\_Driver。

[https://www.silabs.com/documents/public/software/CP210x\\_Universal\\_Windows\\_Driver.zip](https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip)

2: 通过HUB模块进行设备接线(详见图7.1所示),HUB模块蜂鸣器响三声,HUB模块启动。

3: 点击XVIEW上沿的测试模块按钮可进入对应的专用测试功能。如果需要通用测试工具,请点击工具按钮,XVIEW提供串口助手和CAN分析仪可供使用(CAN分析仪仅限采用扩展帧的设备使用)。

XVIEW详细使用说明请在软件中点击上方说明按键查看



CAN分析仪



串口助手

## 4.2 XROS支持

XROS是一套基于ROS的开源开发包,帮助用户在ROS平台快速使用我们的各种设备和产品,减少了大码的开发量。本代码包支持全套XSTD协议标准。请联系客服获取开发包以及使用说明。使用前需自行安装ubuntu操作系统和ROS机器人操作系统,本ros包适配ROS和ROS2版本如下,使用方式请参考XROS使用手册。

注:使用XROS需要配套CAN-COM HUB作为转换设备,请联系客服获得更多技术支持。



Ubuntu 18  
Ubuntu 20

ROS

Melodic  
Noetic

ROS2

Galactic  
Foxy



**HEXMAN**  
ROBOTICS

启原机器人(东莞)有限公司

WWW.HEXMAN.CN

TEL:+86 0769-23078112

MOBILE:+86 18652867127