底盘类CAN协议标准

为了更好的兼容不同设备，我们规范了CAN协议的标准，使得同类型不同型号设备能无缝切换，自动被ROS识别并直接使用，方便更换硬件时候的软件移植工作。以下为底盘使用的专用CAN协议。

请AN调试助手拆开抢以下内容

请注意使用设备CANID替换表中XX部分

底盘类设备的大类ID为 01

底盘指令总表

此表展示了专用指令的所有指令内容，方便快速查询和定位

指令集							反馈集							
设备对象	命令内容	大类 (00~1F)	型号 (00~FF)	编号 (00~FF)	功能 (00~FF)		大类 (00~1F)	型号 (00~FF)	编号 (00~FF)	功能 (00~FF)	反馈内容	周期性	重要性	说明
移动底盘	底盘状态	01	xx	xx	11		01	xx	xx	B1	底盘状态	周期反馈	必备	
	底盘运动	01	xx	xx	12		01	xx	xx	B2	底盘运动	周期反馈	必备	
	/	/	/	/	/		01	xx	xx	B3	主轮里程	周期反馈	必备	
	/	/	/	/	/		01	xx	xx	B4	次轮里程	周期反馈	选备	主要用于
	遥控器信	01	xx	xx	15		01	xx	xx	B5	车载遥控	使能后周	必备	
	安全传感	01	xx	xx	16		01	xx	xx	B6	车载安全	使能后周	选备	
	驱动器运	01	xx	xx	17		01	xx	xx	B7	驱动器运	使能后周	选备	
	驱动器状	01	xx	xx	18		01	xx	xx	B8	驱动器状	使能后周	选备	
	底盘零点	01	xx	xx	19		01	xx	xx	B9	零点较准	对命令反	选备	主要用于
	/	/	/	/	/		01	xx	xx	BA	错误状态信息	周期反馈	必备	
	获取RTC	01	xx	xx	1B		01	xx	xx	BB	RTC时间	对命令反	选备	需要具有
	设置RTC	01	xx	xx	1C		01	xx	xx	BC	设置RTC	对命令反	选备	

设置唤醒	01	xx	xx	1D		01	xx	xx	BD	设置唤醒	对命令反	选备	用于需要睡眠模式的设备
电源输出	01	xx	xx	1E		01	xx	xx	BE	电源输出	对命令反	选备	配合睡眠模式使用
机械参数	01	xx	xx	1F		01	xx	xx	BF	机械参数	对命令反	必备	
odom设置	01	xx	xx	20		01	xx	xx	C0	odom-位移	周期反馈	选备	
/	/	/	/	/		01	xx	xx	C1	odom-速度	周期反馈	选备	

1. 底盘状态设置 / 底盘状态信息

底盘的基础设置，某些功能如刹车、特殊功能等只对部分车型有效，请查阅对应车型的用户手册

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	4	/
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: 随	
byte[1]	使能蜂鸣器	unsigned int8	0x00: 关闭 / 0x01: 开启（默认） 手动控制蜂鸣器，不影响报警的蜂鸣器	
byte[2]	使能刹车	unsigned int8	0x00: 松开 / 0x01: 锁止（如有刹车则默认）	
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	8	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: 随	

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]	蜂鸣器状态	unsigned int8	0x00: 关闭 / 0x01: 开启（默认） 手动控制蜂鸣器，不影响报警的蜂鸣器
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 / 0x01: 锁止（如有刹车则默认）
byte[7]	特殊功能状态	unsigned int8	0x00: 关闭（默认） / 0x01: 开启

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。

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	4	02 01 00 00
上位机	0x01 02 03 B1	数据帧	扩展帧	7	00 02 XX XX 01 00 00

2. 底盘运动控制 / 底盘运动信息

底盘的基础控制指令，对不同车型有所不同，注意区分

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	/
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	signed int16	单位: mm/s	
byte[3]	y方向速度高8位			
byte[4]	角速度低8位	signed int16	阿克曼模式不可作为输入控制，但可以反馈 单位: mrad/s	
byte[5]	角速度高8位			
byte[6]	角度低8位	signed int16	阿克曼模式专用 单位: mrad	
byte[7]	角度高8位			

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	8	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	signed int16	单位: mm/s	
byte[3]	y方向速度高8位			
byte[4]	角速度低8位	signed int16	单位: mrad/s	
byte[5]	角速度高8位			
byte[6]	角度低8位	signed int16	阿克曼模式专用 单位: mrad	
byte[7]	角度高8位			

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

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

[illegible]

> [illegible] [illegible]0x02 [illegible]0x03 [illegible]0.5m/s[illegible]-0.1rad/s[illegible]CANID[illegible]0x01 02 03 12 [illegible]F4 01 00
>
> | [illegible] | | | | | |
> |---|---|---|---|---|---|
> | [illegible] | CAN ID | [illegible] | [illegible] | [illegible] | [illegible]L->H[illegible] |
> | [illegible] | 0x01 02 03 12 | [illegible] | [illegible] | 8 | F4 01 00 00 9C FF 00 00 |
> | [illegible] | 0x01 02 03 B2 | [illegible] | [illegible] | 8 | F4 01 00 00 9C FF 00 00 |

---

3. [illegible]

[illegible]					
[illegible]	[illegible]	CAN ID	[illegible]		[illegible]ms[illegible]
[illegible]	[illegible]	0x 01 XX XX B3	8		20
[illegible]	[illegible]	[illegible]	[illegible]		
byte[0]	[illegible] [7:0]	signed int32	[illegible]mm		
byte[1]	[illegible] [15:8]				
byte[2]	[illegible][23:16]				
byte[3]	[illegible][31:24]				
byte[4]	[illegible] [7:0]	signed int32	[illegible]mm		
byte[5]	[illegible][15:8]				
byte[6]	[illegible][23:16]				
byte[7]	[illegible][31:24]				

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	[illegible]ms[illegible]
[illegible]	[illegible]	0x 01 XX XX B4	8	20
[illegible]	[illegible]	[illegible]	[illegible]	
byte[0]	[illegible][7:0]	signedint32	[illegible]mm	
byte[1]	[illegible][15:8]			
byte[2]	[illegible][23:16]			
byte[3]	[illegible][31:24]			
byte[4]	[illegible][7:0]	signedint32	[illegible]mm	
byte[5]	[illegible][15:8]			
byte[6]	[illegible][23:16]			
byte[7]	[illegible][31:24]			

[illegible]

> [illegible] [illegible]0x02 [illegible]0x03 [illegible]

[illegible]					
[illegible]	CAN ID	[illegible]	[illegible]	[illegible]	[illegible]L->H[illegible]
[illegible]	/	/	/	/	/
[illegible]	0x01 02 03 B3	[illegible]	[illegible]	8	XX XX XX XX XX XX XX XX

4. [illegible] /[illegible]

[illegible]

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	ms[illegible]
[illegible]	[illegible]	0x 01 XX XX 15	1	/
[illegible]	[illegible]	[illegible]	[illegible]	
byte[0]	[illegible]	unsigned int8	0: [illegible] / 1~20:20ms / 20~255:[illegible]	
byte[1~7]	/	/	/	

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	[illegible]ms[illegible]
[illegible]	[illegible]	0x 01 XX XX B5	7	20~255[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	
byte[0]	[illegible]	unsigned int8	[illegible]	
byte[1]	[illegible]	signed int8	[illegible]:-100~100	
byte[2]	[illegible]	signed int8	[illegible]:-100~100	
byte[3]	[illegible]	signed int8	[illegible]:-100~100	
byte[4]	[illegible]	signed int8	[illegible]:-100~100	
byte[5]	[illegible]VRA	signed int8	[illegible]:-100~100	
byte[6]	[illegible]VRA	signed int8	[illegible]:-100~100	
byte[7]	/	/	/	

[illegible]		
[illegible]	[illegible]	[illegible]
bit[0]	0: [illegible] 1: [illegible] 2: [illegible] 3: [illegible]	SWA
bit[1]		

bit[2]	0: 失效 1: 上档 2: 中档 3: 下档	SWB
bit[3]		
bit[4]	0: 失效 1: 上档 2: 中档 3: 下档	SWC
bit[5]		
bit[6]	0: 失效 1: 上档 2: 中档 3: 下档	SWD
bit[7]		

> 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。

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	1	64
上位机	0x01 02 03 B5	数据帧	扩展帧	7	XX XX XX XX XX XX XX

5. 安全传感器信息发送使能 /安全传感器信息

该指令用提供碰撞传感器和超声波等安全传感器信息

安全传感器信息发送使能				
发送方	接收方	CAN ID	数据长度	[illegible]
外部	本设备	0x 01 XX XX 16	1	/
数据位	功能	数据类型	说明	
byte[0]	发送周期设置	unsigned int8	0: 关闭（默认） / 1~20:20ms / 20~255:设定值	

byte[1~7]	/	/	/

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	[illegible]ms[illegible]
[illegible]	[illegible]	0x 01 XX XX B6	7	20~255[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	
byte[0]	[illegible]	unsigned int8	[illegible]	
byte[1]	[illegible]0	unsigned int8	0[illegible]/1~255[illegible]cm	
byte[2]	[illegible]1	unsigned int8	0[illegible]/1~255[illegible]cm	
byte[3]	[illegible]2	unsigned int8	0[illegible]/1~255[illegible]cm	
byte[4]	[illegible]3	unsigned int8	0[illegible]/1~255[illegible]cm	
byte[5]	[illegible]4	unsigned int8	0[illegible]/1~255[illegible]cm	
byte[6]	[illegible]5	unsigned int8	0[illegible]/1~255[illegible]cm	
byte[7]	/	/	/	

[illegible]		
[illegible]	[illegible]	[illegible]
bit[0]	0[illegible] / 1[illegible]	[illegible]0
bit[1]	0[illegible] / 1[illegible]	[illegible]1
bit[2]	0[illegible] / 1[illegible]	[illegible]2
bit[3]	0[illegible] / 1[illegible]	[illegible]3
bit[4]	0[illegible] / 1[illegible]	[illegible]4
bit[5]	0[illegible] / 1[illegible]	[illegible]5
bit[6]	0[illegible] / 1[illegible]	[illegible]6
bit[7]	0[illegible] / 1[illegible]	[illegible]7

仅限安装了防撞边条或触碰传感器的设备才反馈该数据

> 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 。

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	1	64
上位机	0x01 02 03 B6	数据帧	扩展帧	7	XX XX XX XX XX XX XX

6. 驱动器运动信息使能 / 驱动器运动信息

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

驱动器运动信息发送使能				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 17	1	/
数据位	功能	数据类型	说明	
byte[0]	发送周期设置	unsigned int8	0: 关闭（默认） / 1~20:20ms / 20~255: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	可选0, 1, 2, 3, 4, 5, 6, 7 (轮流发送)	
byte[1]	转速低8位	signed int16	单位: Rpm	

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]	/	/	/

> 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 。

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	1	64
上位机	0x01 02 03 B7	数据帧	扩展帧	7	XX XX XX XX XX XX XX

7. 驱动器状态信息使能 / 驱动器状态信息

此数据为驱动器和电机的工作数据，可用于驱动的安全监控

驱动器状态信息发送使能				
发送方	接收方	CAN ID	数据长度	[illegible]
外部	本设备	0x 01 XX XX 18	1	/
数据位	功能	数据类型	说明	
byte[0]	发送周期设置	unsigned int8	0: 关闭（默认） / 1~100: 100ms / 100~255: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	0x 01 XX XX B8	8	100~255 (默认关闭)
数据位	功能	数据类型	说明	
byte[0]	驱动编号	unsigned int8	可选0, 1, 2, 3, 4, 5, 6, 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]	特殊状态	unsignedint8	按照驱动器特殊状态表	

触碰开关表		
位	值	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]	0: 正常 / 1: 卡死	电机卡死报警
bit[5~7]	/	/

> 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 。

[illegible]					
[illegible]	CAN ID	[illegible]	[illegible]	[illegible]	[illegible]L->H[illegible]
[illegible]	0x01 02 03 18	[illegible]	[illegible]	1	64
[illegible]	0x01 02 03 B8	[illegible]	[illegible]	8	XX XX XX XX XX XX XX XX

---

8. [illegible] / [illegible]

[illegible]

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	[illegible]ms[illegible]
[illegible]	[illegible]	0x 01 XX XX 19	8	/
[illegible]	[illegible]	[illegible]	[illegible]	
byte[0]	[illegible]1	int32	[illegible]mrad	
byte[1]				
byte[2]				
byte[3]				
byte[4]	[illegible]2	int32	[illegible]mrad	
byte[5]				
byte[6]				
byte[7]				

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	[illegible]ms[illegible]
[illegible]	[illegible]	0x 01 XX XX B9	8	/

[illegible]	[illegible]	[illegible]	[illegible]
byte[0]	[illegible]1	int32	[illegible]mrad
byte[1]			
byte[2]			
byte[3]			
byte[4]	[illegible]2	int32	[illegible]mrad
byte[5]			
byte[6]			
byte[7]			

[illegible]1[illegible]1[illegible]2[illegible]1[illegible]2

> [illegible] [illegible]0x02 [illegible]0x03 [illegible]1000mrad[illegible]CANID[illegible]0x01 02 03 19 [illegible]E8 03 00 00 00 00 0

[illegible]					
[illegible]	CAN ID	[illegible]	[illegible]	[illegible]	[illegible]L->H[illegible]
[illegible]	0x01 02 03 19	[illegible]	[illegible]	8	E8 03 00 00 00 00 00 00
[illegible]	0x01 02 03 B9	[illegible]	[illegible]	8	E8 03 00 00 00 00 00 00

9. [illegible]

[illegible]

[illegible]				
[illegible]	[illegible]	CAN ID	[illegible]	[illegible]ms[illegible]

本设备	外部	0x 01 XX XX BA	5	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]	电源故障	unsigned int8	查看其电源障表	
byte[5]	/	/	/	

电机故障表		
位	值	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	值	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	值	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	值	说明
bit[0]	0: 正常 / 1:警告	电池低电压提醒 (<15%) (蜂鸣器提醒，设备正常
bit[1]	0: 正常 / 1:故障>	电池低电压报警 (<5%)(设备输出异常状态，驱
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]	/	/

电源故障表		
位	值	说明
bit[0]	0: 正常 / 1:故障>	主继电器故障
bit[1]	0: 正常 / 1:故障>	缓启动故障
bit[2]	0: 正常 / 1:故障>	缓启动升压故障
bit[3]	0: 正常 / 1:故障>	输出过流故障
bit[4]	0: 正常 / 1:故障>	纽扣电池故障
bit[5~7]	/	/

> 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	5	00 00 00 04 00

10. 获取RTC时间

对安装有RTC计数模块的底盘可以获取芯片时间

获取RTC时间				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 1B	0	/
数据位	功能	数据类型	说明	
byte[0~7]	/	/	/	

RTC时间信息				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX BB	4	/
数据位	功能	数据类型	说明	
byte[0]	当前unix 时间[0:7]	unsigned int32	unix时间, 从格林威治时间1970年01月01日00时00分0	
byte[1]	当前unix 时间[8:15]			
byte[2]	当前unix时间[16:23]			
byte[3]	当前unix时间[24:31]			
byte[4~7]	/	/	/	

底盘必须具有RTC时间功能，该功能为特别订制

> 举例：型号0x02 编号0x03 的底盘获取时间为2024-01-01 00:00:30。

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 1B	数据帧	扩展帧	0	/
上位机	0x01 02 03 BB	数据帧	扩展帧	4	1E 90 91 65

11. 设置RTC时间

设置RTC时间				
发送方	接收方	CAN ID	数据长度	周期(ms)
外部	本设备	0x 01 XX XX 1C	4	/
数据位	功能	数据类型	说明	
byte[0]	unix时间[0:7]	unsigned int32	unix时间，从格林威治时间1970年01月01日00时00分0	
byte[1]	unix时间[8:15]			
byte[2]	unix时间[16:23]			
byte[3]	unix时间[24:31]			
byte[4~7]	/	/	/	

设置RTC时间反馈				
发送方	接收方	CAN ID	数据长度	周期(ms)
本设备	外部	0x 01 XX XX BC	1	/
数据位	功能	数据类型	说明	
byte[0]	设置RTC反馈	unsigned int8	0: 设置成功 1: 数据错误，设置失败	
byte[1~7]	/	/	/	

> 举例：型号0x02 编号0x03 的底盘设置RTC时间为2024-01-01 00:00:30。

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 1C	数据帧	扩展帧	4	1E 90 91 65
上位机	0x01 02 03 BC	数据帧	扩展帧	1	00

12. 设置唤醒时间

设置RTC时钟的自动唤醒功能，在使用休眠等节能需求时候使用

设置唤醒时间				
发送方	接收方	CAN ID	数据长度	周期（ms）
外部	本设备	0x 01 XX XX 1D	4	/
数据位	功能	数据类型	说明	
byte[0]	unix时间 [0:7]	unsigned int32	unix时间，从格林威治时间 1970年01月01日00时00分 0	
byte[1]	unix时间[8:15]			
byte[2]	unix时间[16:23]			
byte[3]	unix时间[24:31]			
byte[4~7]	/	/	/	

设置唤醒时间反馈				
发送方	接收方	CAN ID	数据长度	周期（ms）
本设备	外部	0x 01 XX XX BD	5	/
数据位	功能	数据类型	说明	
byte[0]	设置唤醒时间反馈	unsigned int8	0：设置成功 1-闹钟时间小于当前时间，设置失败	
byte[1]	唤醒时间 [0:7]	unsigned int32	unix时间，从格林威治时间 1970年01月01日00时00分 0	
byte[2]	唤醒时间[8:15]			
byte[3]	唤醒时间[16:23]			
byte[4]	唤醒时间[24:31]			
byte[5]	/	/	/	

> 举例：型号0x02 编号0x03 的底盘设置唤醒时间为2024-01-01 00:00:30。

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 1D	数据帧	扩展帧	4	1E 90 91 65
上位机	0x01 02 03 BD	数据帧	扩展帧	5	00 1E 90 91 65

13. 电源输出控制

控制底盘对外部的电源输出，如果是电池直出则不受影响

电源输出控制				
发送方	接收方	CAN ID	数据长度	(ms)
外部	本设备	0x 01 XX XX 1E	1	/
数据位	功能	数据类型	说明	
byte[0]	电源输出控制	unsigned int8	0:关闭电源 1:启动电源 2:查询当前电源状态	
byte[1~7]	/	/	/	

电源输出反馈				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX BE	2	/
数据位	功能	数据类型	说明	
byte[0]	电源输出控制反馈	unsigned int8	0:操作成功 1:唤醒时间未设置，关闭失败 2:开启失败，电源未关闭	
byte[1]	电源输出状态反馈	unsigned int8	0:电源已关闭 1:电源已开启	
byte[2~7]	/	/	/	

如果有工控机等外部设备请谨慎使用该功能，该功能需要具备RTC时间功能的底盘，并需要先设置唤醒时间才能关闭电源

举例：型号0x02 编号0x03 的底盘关闭输出电源。

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 1E	数据帧	扩展帧	1	00
上位机	0x01 02 03 BE	数据帧	扩展帧	2	00 01

14. 机械参数设置

底盘的结构参数，用于补偿里程计和速度差异

机械参数设置				
发送方	接收方	CAN ID	数据长度	周期 (ms)
外部	本设备	0x 01 XX XX 1F	2	/
数据位	功能	数据类型	说明	
byte[0]	设置轮径	uint16	0: 不设置 单位: mm	
byte[1]				
byte[2~7]	/	/	/	

机械参数反馈				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX BF	7	/
数据位	功能	数据类型	说明	

byte[0]	运动模型	uint8	参考运动模型ID表
byte[1]	轴距	uint16	单位: mm
byte[2]			
byte[3]	轮距	uint16	单位: mm
byte[4]			
byte[5]	轮径	uint16	单位: mm
byte[6]			
byte[7]	/	/	/

运动模型ID表		
字节	状态码的取值	状态说明
byte[0]	0x00	未知模型
	0x01	两轮差速
	0x02	四轮差速
	0x03	阿克曼
	0x04	麦轮
	0x05	三正交轮
	0x06	四正交轮

机械参数设置会严重影响里程计与速度数据输出，请谨慎使用

> 举例：型号0x02 编号0x03 轴距500mm 轮距300mm 的底盘设置轮径为200mm。

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 1F	数据帧	扩展帧	1	C8 00
上位机	0x01 02 03 BF	数据帧	扩展帧	7	01 F4 01 2C 01 C8 00

15. odom设置

用于开启底盘计算odom模式，会比上位机计算精确，并可开启IMU较准

odom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 20	2	/
数据位	功能	数据类型	说明	
byte[0]	使能odom反馈	uint8	0: 失能（默认） / 1: 使能	
byte[1]	IMU较准	uint8	0: 不使用 / 1: 使用（默认） 如果底盘没有IMU则不可使用	
byte[2~7]	/	/	/	

odom-位移反馈				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX C0	8	20
数据位	功能	数据类型	说明	
byte[0]	odom_x [7:0]	signed int32	单位: mm	
byte[1]	odom_x[15:8]			
byte[2]	odom_x[23:16]			
byte[3]	odom_x[31:24]			
byte[4]	odom_y[7:0]	signed int32	单位: mm	

byte[5]	odom_y[15:8]
byte[6]	odom_y[23:16]
byte[7]	odom_y[31:24]

odom-旋转反馈				
发送方	接收方	CAN ID	数据长度	周期 (ms)
本设备	外部	0x 01 XX XX C1	4	20
数据位	功能	数据类型	说明	
byte[0]	odom_th [7:0]	signed int32	单位: mrad	
byte[1]	odom_th [15:8]			
byte[2]	odom_th [23:16]			
byte[3]	odom_th [31:24]			
byte[4~7]	/	/	/	

只有具备车载IMU功能的底盘才能开启IMU较准

> 举例：型号0x02 编号0x03 底盘开启IMU较准 odom, odom位置为x=1m y= 1m。

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 20	数据帧	扩展帧	2	01 01
上位机	0x01 02 03 C0	数据帧	扩展帧	8	E8 03 00 00 E8 03 00 00
上位机	0x01 02 03 C1	数据帧	扩展帧	4	00 00 00 00

版本号 #11
由 Yixuan Jiang 创建于 16 七月 2024 08:53:57
由 Yixuan Jiang 更新于 28 十月 2024 12:06:19