



TRACER

AgileX Robotics Team

用户手册 V.2.0.0 2021.06

本章包含重要的安全信息, 在机器人第一次通电前, 任何个人或者机构在使用设备之前必须阅读并理解这些信息。有任何相关使用的疑问都可以联系我们support@agilex.ai必须遵守并执行本手册其他章节中的所有组装说明和指南, 这一点非常重要。应特别注意与警告标志相关的文本。

⚠ 重要安全信息 Safety Information

本手册中的信息不包含设计、安装和操作一个完整的机器人应用，也不包含所有可能对这一完整的系统的安全造成影响的周边设备。该完整的系统的设计和使用需要符合该机器人安装所在国的标准和规范中确立的安全要求。

TRACER的集成商和终端客户有责任确保遵循相关国家的切实可行的法律法规，确保完整的机器人应用实例中不存在任何重大危险。这包括但不限于以下内容：

1. 有效性和责任

- 对完整的机器人系统做一个风险评估。
- 将风险评估定义的其他机械的附加安全设备连接在一起。
- 确认整个机器人系统的外围设备包括软件和硬件系统的设计和安装准确无误。
- 本机器人不具备一个完整的自主移动机器人具备的自动防撞、防跌落、生物接近预警等相关安全功能但不局限于上述描述，相关功能需要集成商和终端客户遵循相关规定和切实可行的法律法规进行安全评估，确保开发完成的机器人在实际应用中不存在任何重大危险和安全隐患。
- 收集技术文件中的所有的文档：包括风险评估和本手册。

4. 操作

- 保证遥控时周围区域相对空旷。
- 在视距内遥控控制。
- TRACER最大的载重为100KG，在使用时，确保有效载荷不超过100KG。
- TRACER安装外部扩展时，确认扩展的质心位置，确保在旋转中心。
- 当设备电压低于22.5V时请及时充电。
- 当设备出现异常时，请立即停止使用，避免造成二次伤害。

5. 保养

- 为保证电池的蓄电能力，电池应带电存放，长时间不使用也要定时充电。

2. 环境

- 首次使用，请先仔细阅读读本手册，了解基本操作内容与操作规范。
- 遥控操作，选择相对空旷区域使用，车上本身是不带任何自动避障传感器。
- 在-10°C~45°C的环境温度中使用。
- 如果车辆非单独定制IP防护等级，车辆防水、防尘能力为IP22。

3. 检查

- 确保各设备的电量充足。
- 确保车辆无明显异常。
- 检查遥控器的电池电量是否充足。
- 使用时确保急停开关已经被释放。

- 当设备出现异常时，请联系相关技术人员，请勿擅自处理。
- 请根据设备的IP防护等级在满足防护等级要求的环境中使用。
- 请勿直接推车。
- 充电时，确保周围环境温度大于0°C。

目录

1 TRACER简介 Introduction	1	3.3.3 CAN 指令控制的实现	12
1.1 产品列表	1	3.4串口通信协议	13
1.2 性能参数	1	3.4.1 串口协议介绍	13
1.3 开发所需	1	3.4.2 串口协议内容	13
		3.4.3 串口的连接	18
2 基本介绍 The Basics	2	3.5固件升级	18
2.1 状态指示	3	3.6 TRACER ROS Package 使用示例	19
2.2 电气接口说明	3		
2.2.1 尾部电气接口说明	3	4 注意事项 Attention	20
2.3 遥控说明	4	4.1电池注意事项	20
2.4 控制指令与运动说明	5	4.2 使用环境注意事项	20
		4.3 电气外部扩展注意事项	20
3 使用与开发 Getting Started	6	4.4 其他注意事项	20
3.1 使用与操作	6		
3.2 充电	6	5 常见问题与解决 Q&A	20
3.3 开发	6		
3.3.1 CAN 接口协议	6	6 产品尺寸 Product Dimensions	21
3.3.2 CAN 线的连接	12	6.1 产品外形尺寸说明图	21

1 TRACER 简介 Introduction

TRACER 是一款全能型行业应用 UGV (UNMANNED GROUND VEHICLE)。它是一款采用模块化、智能化的设计理念的多功能模块化的行业应用移动机器人开发平台，具有强大载荷能力和强劲动力系统的它有广泛的应用领域。两轮差速的和轮毂电机的搭配使得它能够在室内灵活运动。立体相机、激光雷达、GPS、IMU、机械手等设备可选择加装至 TRACER 作为扩展应用。TRACER 可被应用到无人巡检、科研、物流等领域。

1.1 产品列表

名称	数量
TRACER 机器人本体	x1
电池充电器 (AC 220V)	x1
遥控器 (选配)	x1
USB 转 RS232	x1
航空插头公头 (4 Pin)	x1
USB 转 CAN 通讯模块	x1

1.2 性能参数

参数类型	项目	指标
机械参数	长 x 宽 x 高 (mm)	685 X 570 X 155
	轮距 (mm)	360
	车体重量 (Kg)	28~30
	电池类型	锂电池 24V 15Ah
	电机	直流无刷 2 X 150W
	驱动形式	独立驱动
	悬挂形式	摇摆臂独立悬挂
	转向	差速转向
性能参数	安全装备	伺服刹车/防撞管
	空载最高车速 (m/s)	1.8
	最小转弯半径	可原地转弯
	码盘参数	1024 线
	最小离地间隙 (mm)	30
控制参数	最大爬坡能力	8°
	控制模式	遥控控制指令模式
	遥控器	2.4G / 极限距离 200m
	通讯接口	CAN / RS232

1.3 开发所需

TRACER 出厂时可选配遥控器，用户可以通过遥控器控制 TRACER 移动机器人底盘，完成移动和旋转操作；TRACER 配备了 CAN 和 RS232，用户可以通过 CAN 和 RS232 接口进行二次开发。

2 基本介绍 The Basics

本部分内容将会对 TRACER 移动机器人底盘作一个基本的介绍, 便于用户和开发者对于TRACER底盘有一个基本的认识。如下图 2.1 与 2.2 所示, 为整个移动机器人底盘的概览视图。

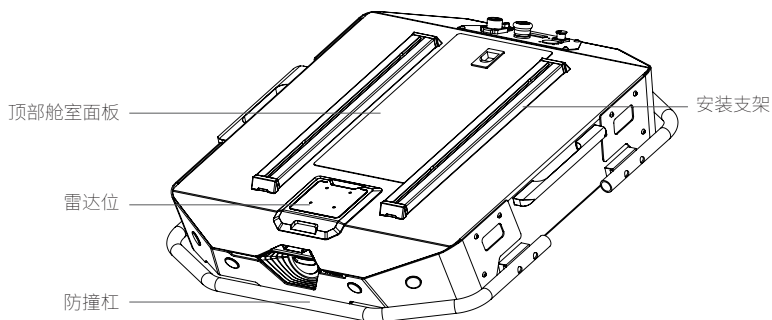


图2.1前部概览视图

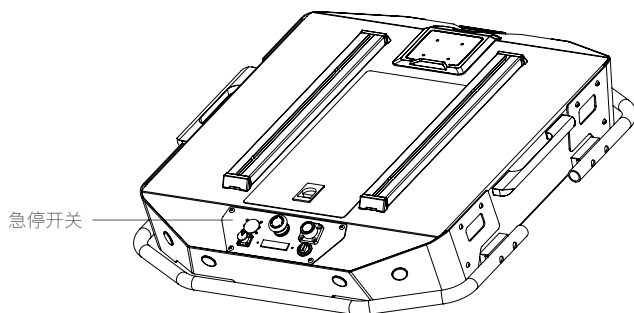


图2.2尾部概览视图

TRACER 整体上采用了模块化和智能化的设计理念, 加上动力强劲的直流轮毂电机, 使得 TRACER 机器人底盘开发平台在室内平坦的地面可以灵活运动。车体四周均安装安全防护管, 可在发生紧急事故时, 减缓对车体的损伤。

车体前安装有灯光, 前侧采用白光设计, 可进行照明。

尾部有一个紧急停车开关, 在发生紧急情况时可快速进行紧急停车操作, 避免发生安全事故, 降低或避免不必要的损失。

在TRACER的尾部配置了开放的电气接口和通讯接口, 方便客户进行二次开发, 电气接口在设计选型上采用了航空防水接插件, 一方面利于用户的扩展和使用, 另外一方面使得机器人平台可以在一些严苛的环境种使用。在车体顶部有一个卡口式的开放舱室预留给用户使用。

2.1 状态指示

用户可以通过安装在 TRACER 上的电压表以及灯光来确定车体的状态。具体可以参考表 2.1。

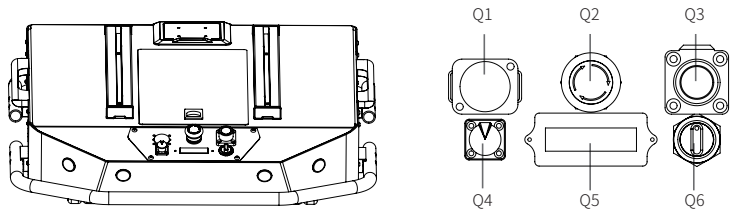
状态	描述
当前电压	当前电池电压可通过尾部电气面板中的电压表查看，精确到1V
低电压报警	当电池电压低于22V，车体会发出“滴-滴-滴”刺耳的声音进行提示。当检测到电池电压低于21.5V时，TRACER为了防止电池损坏，会主动切断外部扩展供电和驱动器供电，此时底盘将无法进行运动控制和接受外部指令控制。
上电显示	尾部电压表亮起

表格 2.1 车体状态说明表

2.2 电气接口说明

2.2.1 尾部电气接口说明

尾部的扩展接口如图 2.3 所示，其中 Q1 为 DB9 串口；Q2 为急停开关；Q3 为电源充电口；Q4 为 CAN 和 24V 电源扩展接口；Q5 为电量显示表；Q6 为旋钮开关，是电气部分总开关。



关于 Q1 的其具体引脚定义如图 2.3 所示。

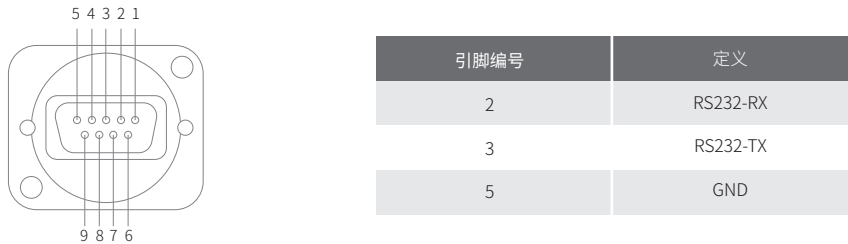


图 2.4 Q4 引脚说明图

尾部配备了 CAN 通信接口和 24V 电源接口。其线序的具体定义如图 2.5 所示。



图 2.5 尾部航空接口引脚说明图

2.3 遥控器说明

富斯遥控器为 TRACER 产品选配配件，客户可根据实际需求选配；使用遥控器可以轻松控制 TRACER 通用机器人底盘，在本产品中我们采用左手油门的设计。其定义及其功能可参考图 2.6。



图 2.6 富斯遥控器按键示意图

按键的功能定义为：SWA、SWD 暂时未被启用，其中 SWB 为控制模式选择按钮，拨至最上方为指令控制模式，拨至中间为遥控控制模式；SWC 为灯光控制按钮；S1 为油门按钮，控制 TRACER 前进和后退；S2 控制旋转，POWER 为电源按钮，同时按住即可开关机。

2.4 控制指令与运动说明

我们将地面移动车辆根据 ISO 8855 标准建立如图 2.7 的坐标参考系。

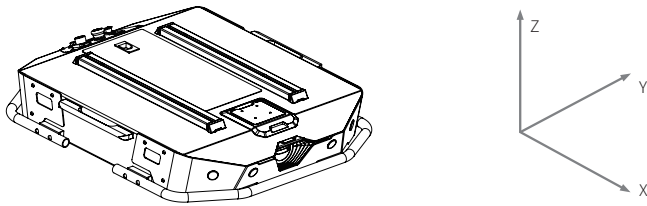


图 2.7 车身参考坐标系示意图

正如 2.7 所展示的, TRACER 车体与建立的参考坐标系 X 轴为平行状态。

在遥控器控制模式下, 遥控器摇杆S1往前推动则为往 X 正方向运动, S1 往后推动则往 X 负方向运动, S1 推动至最大值时, 往X正方向运动速度最大, S1推动至最小值时, 往 X 负方向运动速度最大; 遥控器摇杆S2 左右控制车体的旋转运动, S2往左推动车体则由 X 轴正方向往 Y 正方向旋转, S2往右推动车体则由 X 轴正方向往 Y 负方向旋转, S2往左推动至最大值时, 逆时针方向旋转线速度最大, S2往右推动至最大值时, 顺时针旋转线运动速度最大。

在控制指令模式下, 线速度的正值表示往X轴正方向运动, 线速度的负值表示往X轴负方向运动; 角速度的正值表示车体由X轴正方向往 Y 轴正方向运动, 角速度的负值表示车体由 X 轴正方向往Y轴负方向运动。

3 使用与开发 Getting Started

本部分主要介绍 TRACER 平台的基本操作与使用,介绍如何通过外部 CAN 口,通过 CAN 总线协议来对车体进行二次开发。

3.1 使用与操作

检查	启动
• 检查车体状态。检查车体是否有明显异常;如有,请联系售后支持;	• 急停开关状态。确认急停按钮均处于释放状态; • 旋转旋钮开关(电气面板中Q6),正常情况下,电压表正常显示电池电压,前灯正常亮起;
关闭操作	急停
• 旋转旋钮开关,即可切断电源;	• 按下车体左右尾部的急停开关即可;

遥控控制基本操作流程
• 正常启动 TRACER 移动机器底盘后,启动遥控器,将控制模式选择为遥控控制模式,即可通过遥控器控制 TRACER平台运动。

3.2 充电

TRACER 产品默认随车配备一个 10A 的充电器,可满足客户的充电需求。

充电具体操作流程如下:

• 确保TRACER底盘处于停机断电状态。充电前请确认尾部电气控制台中Q6(旋钮开关)处于关闭状态; • 将充电器的插头插入车尾电气控制面板中 Q3充电界面中;	• 将充电器连接电源,将充电器开关打开,即可进入充电状态。 • 正常充电时,底盘没有指示灯说明。具体说明请看充电器指示灯说明。
--	---

3.3 开发

TRACER 产品针对用户的开发提供了 CAN 和 RS232的接口,用户可选择其中一种接口对车体进行指令控制。

3.3.1 CAN 接口协议

TRACER产品中 CAN 通信标准采用的是 CAN2.0B 标准,通讯波特率为 500K,报文格式采用 MOTOROLA 格式。通过外部 CAN 总线接口可以控制底盘的移动的线速度以及旋转的角速度;TRACER会实时反馈当前的运动状态信息以及 TRACER 底盘的状态信息等。

协议包含系统状态回馈帧、运动控制回馈帧、控制帧,查询配置帧,协议内容具体如下:

系统状态回馈指令包含了当前车体状态回馈、控制模式状态回馈、电池电压回馈以及故障回馈,协议内容如表 3.1 所示。

表格 3.1 TRACER 底盘系统状态回馈帧

指令名称		系统状态回馈指令		
发送节点	接收节点	ID	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	0x211	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	当前车体状态	unsigned int8	0x00 系统正常	
			0x01 紧急停车模式	
			0x02 系统异常	
byte [1]	模式控制	unsigned int8	0x00 遥控模式	
			0x01 CAN指令控制模式	
			0x02 串口控制模式	
byte [2]	电池电压高八位	unsigned int16	实际电压X 10 (精确到0.1V)	
byte [3]	电池电压低八位			
byte [4]	故障信息	unsigned int16	详见备注[故障信息说明]	
byte [5]	保留	-	0x00	
byte [6]	保留	-	0x00	
byte [7]	计数校验 (count)	unsigned int8	0~255循环计数	

表格 3.2 故障信息说明表

故障信息说明		
byte [4]	bit [0]	电池欠压故障 (0:无故障 1:故障) 保护电压为 22V
	bit [1]	电池欠压警告 (0:无警告 1:警告), 报警电压22.5V
	bit [2]	遥控器失联保护 (0:正常 1:遥控器失联)
	bit [3]	保留
	bit [6]	保留
	bit [7]	保留

运动控制回馈帧指令包含了当前车体的运动线速度、运动角速度回馈，协议具体内容如表 3.3 所示。

表格 3.3 运动控制回馈帧

指令名称		运动控制回馈指令		
发送节点	接收节点	ID	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	0x221	20ms	无
数据长度		0x08		
位置	功能	数据类型	说明	
byte [0]	移动速度高八位	signed int16	车体行进速度 单位:mm/s	
byte [1]	移动速度低八位			
byte [2]	旋转速度低八位	signed int16	车体旋转角速度 单位:0.001rad/s	
byte [3]	旋转速度高八位			
byte [4]	保留	-	0x00	
byte [5]	保留	-	0x00	
byte [6]	保留	-	0x00	
byte [7]	保留	-	0x00	

控制帧包含了线速度控制开度、角速度控制开度, 其具体协议内容如表 3.4 所示。

表格 3.4 运动控制指令控制帧

指令名称		控制指令		
发送节点	接收节点	ID	周期 (ms)	接收超时(ms)
决策控制单元	底盘节点	0x111	20ms	500ms
数据长度		0x08		
位置	功能	数据类型	说明	
byte [0]	线速度高八位	signed int16	车体行进速度 单位mm/s 有效值±1800	
byte [1]	线速度低八位			
byte [2]	角速度高八位	signed int16	车体旋转速度 有效值±1000 单位0.001rad/s	
byte [3]	角速度低八位			
byte [4]	保留	—	0x00	
byte [5]	保留	—	0x00	
byte [6]	保留	—	0x00	
byte [7]	保留	—	0x00	

灯光控制回馈帧指令包含了当前前向灯光状态的反馈 具体内容如表 3.5所示。

表格 3.5 灯光控制指令帧

指令名称		灯光控制帧		
发送节点	接收节点	ID	周期 (ms)	接收超时(ms)
决策控制单元	线控底盘	0x121	25ms	无
数据长度		0x08		
位置	功能	数据类型	说明	
byte [0]	灯光控制使能标志	unsigned int8	0x00 控制指令无效 0x01灯光控制使能	
byte [1]	前侧灯光模式	unsigned int8	0x00 常关 0x01 常开 0x02 呼吸灯模式 0x03 客户自定义亮度	
byte [2]	前侧灯光自定义亮度	unsigned int8	[0,100],其中0为不亮, 100最亮注[5]	
byte [3]	保留	--	0x00	
byte [4]	保留	--	0x00	
byte [5]	保留	--	0x00	
byte [6]	保留	-	0x00	
byte [7]	计数校验 (count)	unsigned int8	0~255循环计数, 每发送一条指令计数加一次	

注[5]: 此值自在自定义模式下有效

灯光控制帧指令包含了灯光控制的模式和开度 具体内容如表 3.6所示。

表格 3.6灯光控制反馈帧

指令名称		灯光控制反馈帧		
发送节点	接收节点	ID	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	0x231	20ms	无
数据长度		0x08		
位置	功能	数据类型	说明	
byte [0]	当前灯光控制使能标志	unsigned int8	0x00 控制指令无效 0x01灯光控制使能	
byte [1]	前前侧灯光模式	unsigned int8	0x00 常关 0x01 常开 0x02 呼吸灯模式 0x03 客户自定义亮度	
byte [2]	当前前侧灯光自定义亮度	unsigned int8	[0,100],其中0为不亮, 100最亮	
byte [3]	保留	--	0x00	
byte [4]	保留	--	0x00	
byte [5]	保留	--	0x00	
byte [6]	保留	-	0x00	
byte [7]	计数校验 (count)	unsigned int8	0~255循环计数, 每发送一条指令计数加一次	

控制模式帧包用于设定底盘的控制方式，具体如表3-7

表格 3.7 控制模式设定帧说明表

控制模式设定帧				
指令名称	发送节点	接收节点	ID	周期 (ms) 接收超时(ms)
线控底盘	决策控制单元	0x421	无	无
数据长度	0x01			
位置	功能	数据类型	说明	
byte [0]	控制模式	unsigned int8	0x00 遥控模式 0x01 CAN指令控制模式[1] 0x02 串口控制模式	

注 [1]控制模式说明

默认是待机模式，需要切换到指令模式才能发送运动控制指令。若打开遥控器，遥控器拥有最高权限，可以屏蔽指令的控制。当遥控器切换到指令模式时，还是需要先发送控制模式设定指令才能响应速度指令。

状态置位帧包用于清除错误信息，具体如表3-8

表格 3.8 状态置位帧说明表

状态置位帧				
指令名称	发送节点	接收节点	ID	周期 (ms) 接收超时(ms)
线控底盘	决策控制单元	0x441	无	无
数据长度	0x01			
位置	功能	数据类型	说明	
byte [0]	控制模式	unsigned int8	0x00 清除所有错误 0x01 清除电机1错误 0x02 清除电机2错误	

表格 3.9 里程计反馈帧说明表

里程计反馈指令				
指令名称	发送节点	接收节点	ID	周期 (ms) 接收超时(ms)
线控底盘	决策控制单元	0x311	无	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	左轮里程计最高位	signed int32	左轮里程计数据 单位mm	
byte [1]	左轮里程计次高位			
byte [2]	左轮里程计次低位			
byte [3]	右轮里程计最低位			
byte [4]	右轮里程计最高位	signed int32-	右轮里程计数据 单位mm	
byte [5]	右轮里程计次高位			
byte [6]	右轮里程计次低位			
byte [7]	右轮里程计最低位			

除了底盘的状态信息会进行反馈以外，底盘反馈的信息还包括电机信息。下面的帧反馈是电机信息反馈：在底盘中两个电机编号对应为如下图所示：

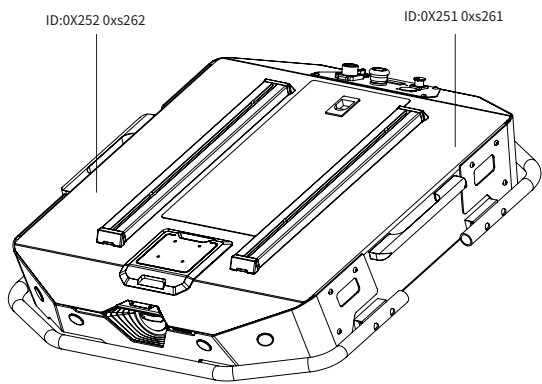


图 3.0 电机反馈 ID 示意图

表格 3.10电机高速信息反馈帧

电机高速信息反馈帧					
指令名称	发送节点	接收节点	ID	周期 (ms)	接收超时(ms)
决策控制单元	线控底盘	0x251~0x252		20ms	无
数据长度	0x08				
位置	功能	数据类型	说明		
byte [0]	电机转速高八位	signed int16	电机当前转速		
byte [1]	电机转速低八位		单位:RPM		
byte [2]	保留	-	0x00		
byte [3]	保留	--	0x00		
byte [4]	保留	--	0x00		
byte [5]	保留	--	0x00		
byte [6]	保留	-	0x00		
byte [7]	保留	-	0		

表格 3.11电机低速信息反馈

指令名称		电机低速信息反馈帧		
发送节点	接收节点	ID	周期(ms)	接收超时(ms)
决策控制单元	线控底盘	0x261~0x262	100ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	保留	-	0x00	
byte [1]	保留	-	0x00	
byte [2]	保留	-	0x00	
byte [3]	保留	-	0x00	
byte [4]	保留	-	0x00	
byte [5]	驱动器状态	-	详见表3-12	
byte [6]	保留	-	0x00	
byte [7]	保留	-	0	

表格 3.12 故障信息说明表

故障信息说明		
byte [5]	bit [0]	保留
	bit [1]	保留
	bit [2]	保留
	bit [3]	保留
	bit [4]	CAN通讯是否掉线(0:正常 1:掉线)
	bit [6]	保留
	bit [7]	保留

3.3.2 CAN 线的连接

线的定义可参考表 3.2。



红色：VCC (电池正极)
黑色：GND (电池负极)
蓝色：CAN_L
黄色：CAN_H

图 3.2 航空插头公头示意图

注：此版本中电源最大可提供5A的电流。

3.3.3 CAN 指令控制的实现

正常启动TRACER移动机器人底盘，打开FS遥控器，然后将控制模式切换至指令控制，即将FS遥控器SWB模式选择拨至最上方，此时TRACER底盘会接受来自CAN接口的指令，同时主机也可以通过CAN总线回馈的实时数据，解析当前底盘的状态，具体协议内容参考CAN通讯协议。

3.4串口通信协议

3.4.1 串口协议介绍

它是在1970年由美国电子工业协会(EIA)联合贝尔系统、调制解调器厂家及计算机终端生产厂家共同制定的用于串行通讯的标准。它的全名是“数据终端设备(DTE)和数据通讯设备(DCE)之间串行二进制数据交换接口技术标准”该标准规定采用一个25个脚的DB-25连接器,对连接器的每个引脚的信号内容加以规定,还对各种信号的电平加以规定。后来IBM的PC机将RS232简化成了DB-9连接器,从而成为事实标准。而工业控制的RS-232口一般只使用RXD、TXD、GND三条线。

协议说明

起始位		帧长度	指令类型	指令ID	数据域			帧ID	校验和
SOF		frame_L	CMD_TYPE	CMD_ID	data [0]	...	data[n]	frame_id	check_sum
byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	...	byte 6+n	byte 7+n	byte 8+n
5A	A5								

协议包含起使位, 帧长度, 帧指令类型, 指令ID, 数据域, 帧ID, 校验和组成。其中帧长度是指除去起始位和校验和以外的长度, 校验和为起使位到帧ID所有数据和求和; 帧ID位0~255循环计数, 每发送一条指令计数自加一次。

协议内容

系统状态回馈指令

指令名称		系统状态回馈指令	
发送节点	接收节点	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	20ms	无
帧长度	0x0a		
指令类型	反馈指令 (0xAA)		
指令ID	0x01		
数据域长度	6		
位置	功能	数据类型	说明
byte [0]	当前车体状态	unsigned int8	0x00 系统正常
			0x01 紧急停车模式
			0x02系统异常
			0x00 遥控模式
byte [1]	模式控制	unsigned int8	0x01 指令控制模式[1]
			0x02 串口控制模式
byte [2]	电池电压高八位	unsigned int16	实际电压X 10 (精确到0.1V)
byte [3]	电池电压低八位		
byte [4]	故障信息	unsigned int8	详见备注[故障信息说明]
byte [5]	保留	--	0x00

3.4.2 串口协议内容

通讯基本参数

项目	参数
波特率	115200
校验	无检验
数据位长度	8位
停止位	1位

```

/**
 * @BRIEF SERIAL MESSAGE CHECKSUM EXAMPLE CODE
 * @PARAM[IN] *DATA : SERIAL MESSAGE DATA STRUCT POINTER
 * @PARAM[IN] LEN :SERIAL MESSAGE DATA LENGTH
 * @RETURN THE CHECKSUM RESULT
 */
STATIC UINT8 AGILEX_SERIALMSGCHECKSUM(UINT8 *DATA, UINT8 LEN)
{
    UINT8 CHECKSUM = 0X00;
    FOR(UINT8 I = 0; I < (LEN-1); I++)
    {
        CHECKSUM += DATA[I];
    }
}

```

图 3.3 串口校验算法代码示例

故障信息说明		
字节	位	含义
byte [4]	bit [0]	电池欠压故障 (0:无故障 1:故障) 故障电压为22V
	bit [1]	电池欠压警告 (0:无警告 1:警告) 报警电压为22.5V
	bit [2]	遥控器失联故障 (0:无故障 1:故障)
	bit [3]	预留, 默认0
	bit [4]	预留, 默认0
	bit [5]	预留, 默认0
	bit [6]	预留, 默认0
	bit [7]	预留, 默认0

运动指控回馈指令

指令名称		运动控制回馈指令	
发送节点	接收节点	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	20ms	无
帧长度	0x0A		
指令类型	反馈指令 (0xAA)		
指令ID	0x02		
数据域长度	6		
位置	功能	数据类型	说明
byte [0]	移动速度高八位	signed int16	实际速度X 1000 (精确到0.001m/s)
byte [1]	移动速度低八位		
byte [2]	旋转速度高八位	signed int16	实际速度X 1000 (精确到0.001rad/s)
byte [3]	旋转速度低八位		
byte [4]	保留	-	0x00
byte [5]	保留	-	0x00

运动控制指令

指令名称		控制指令	
发送节点	接收节点	周期 (ms)	接收超时(ms)
决策控制单元	底盘节点	20ms	500ms
帧长度	0x0A		
指令类型	控制指令 (0x55)		
指令ID	0x01		
数据域长度	6		
位置	功能	数据类型	说明
byte [0]	控制模式	unsigned int8	0x00 遥控模式
			0x01 CAN指令控制模式[1]
			0x02 串口控制模式
byte [1]	线速度高八位	signed int16	单位mm / s,最大速度1.8m/s
byte [2]	线速度低八位		
byte [3]	角速度高八位	signed int16	单位0.001rad / s最大速度1.0235 rad/s
byte [4]	角速度低八位		
byte [5]	保留	-	0x00

1号电机驱动器信息反馈帧

1号电机驱动器信息反馈帧				
指令名称	发送节点	接收节点	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	20ms	无	
帧长度	0x0A			
指令类型	反馈指令 (0xAA)			
指令ID	0x03			
数据域长度	6			
位置	功能	数据类型	说明	
byte [0]	1号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)	
byte [1]	1号驱动器转速低八位			
byte [2]	1号驱动器状态	unsigned int8	驱动器状态: Bit[0-3]:保留 Bit[4]:CAN通讯状态 0-正常 1-异常 Bit[5-6]:保留	
byte [3]	保留	--	0x00	
byte [4]	保留	--	0x00	
byte [5]	保留	--	0x00	

2号电机驱动器信息反馈帧

2号电机驱动器信息反馈帧				
指令名称	发送节点	接收节点	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	20ms	无	
帧长度	0x0A			
指令类型	反馈指令 (0xAA)			
指令ID	0x04			
数据域长度	6			
位置	功能	数据类型	说明	
byte [0]	2号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)	
byte [1]	2号驱动器转速低八位			
byte [2]	2号驱动器状态	unsigned int8	驱动器状态: Bit[0-3]:保留 Bit[4]:CAN通讯状态 0-正常 1-异常 Bit[5-6]:保留	
byte [3]	保留	--	0x00	
byte [4]	保留	--	0x00	
byte [5]	保留	--	0x00	

灯光控制帧

指令名称		灯光控制帧	
发送节点	接收节点	周期 (ms)	接收超时(ms)
决策控制单元	底盘节点	20ms	500ms
帧长度	0x0A		
指令类型	控制指令 (0x55)		
指令ID	0x02		
数据域长度	6		
位置	功能	数据类型	说明
byte [0]	灯光控制使能标志	unsigned int8	0x00 控制指令无效 0x01 灯光控制使能
byte [1]	前侧灯光模式	unsigned int8	0x00 常关 0x01 常开 0x02 呼吸灯模式 0x03 客户自定义亮度
byte [2]	前侧灯光自定义亮度	unsigned int8	[0,100],其中0为不亮, 100最亮
byte [3]	尾部灯光模式	unsigned int8	0x00 常关 0x01 常开 0x02 呼吸灯模式 0x03 客户自定义亮度
byte [4]	尾部灯光自定义亮度	unsigned int8	[0,100],其中0为不亮, 100最亮
byte [5]	保留	--	0x00

灯光控制反馈帧

指令名称		灯光控制反馈帧	
发送节点	接收节点	周期 (ms)	接收超时(ms)
线控底盘	决策控制单元	20ms	无
帧长度	0x0A		
指令类型	反馈指令 (0xAA)		
指令ID	0x07		
数据域长度	6		
位置	功能	数据类型	说明
byte [0]	当前灯光控制使能标志	unsigned int8	0x00 控制指令无效 0x01 灯光控制使能
byte [1]	当前前侧灯光模式	unsigned int8	0x00 常关 0x01 常开 0x02 呼吸灯模式 0x03 客户自定义亮度
byte [2]	当前前侧灯光自定义亮度	unsigned int8	[0,100],其中0为不亮, 100最亮
byte [3]	当前尾部灯光模式	unsigned int8	0x00 常关 0x01 常开 0x02 呼吸灯模式 0x03 客户自定义亮度
byte [4]	当前尾部灯光自定义亮度	unsigned int8	[0,100],其中0为不亮, 100最亮
byte [5]	保留	--	0x00

示例数据

通过控制底盘以0.1m/s的线速度进行前进运动，下面是具体数据内容

起始位		帧长度	指令类型	指令ID	数据域			帧ID	校验和
byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	...	byte 6+n	byte 7+n	byte 8+n
0x5A	0xA5	0x0A	0x55	0x01	...	...	...	0x00	0x6B

下面是数据域内容：

位置	功能	值
byte [0]	控制模式	0x02
byte [1]	线速度高八位	0x00
byte [2]	线速度高八位	0x64
byte [3]	角速度高八位	0x00
byte [4]	角速度低八位	0x00
byte [5]	保留	0x00

整串数据内容为：5A 05 A0 55 01 02 00 64 00 00 00 00 C5

3.5 固件升级

为了方便客户对TRACER所使用的固件版本进行升级，给客户带来更加完善的体验，TRACER 提供了固件升级的硬件接口以及与之对应的客户端软件。其客户端界面如图3.3所示。

升级准备

- 串口线 X 1
- USB 转串口 X 1
- TRACER 底盘 X 1
- 电脑(WINDOWS 操作系统) X 1

固件升级软件

- https://github.com/agilexrobotics/agilex_firmware

3.4.3 串口的连接

使用我们的通讯配套工具里面的USB转RS232串口线，与车尾部的串口相连，使用串口工具，设置好相应波特率，使用上面的提供的示例数据即可测试，如果遥控器是开启的状态，需要将遥控器切换至指令控制模式，如果遥控器未开启，直接发送控制指令即可，需要注意的是，指令必须要是周期性的发送，如果底盘超过500ms未接收到串口指令，进入失联保护状态。

升级过程

- 连接前保证机器人底盘电源处于断开状态；
- 使用串口线连接至 TRACER底盘尾部串口；
- 串口线连接至电脑；
- 打开客户端软件；
- 选择端口号；
- TRACER底盘上电，立即点击开始连接（TRACER底盘会在上电前3S等待，如果时间超过3S则会进入应用程序）；若连接成功，会在文本框提示“连接成功”；
- 加载 BIN 文件；
- 点击升级，等待升级完成的提示即可；
- 断开串口线，底盘断电，再次通电即可。

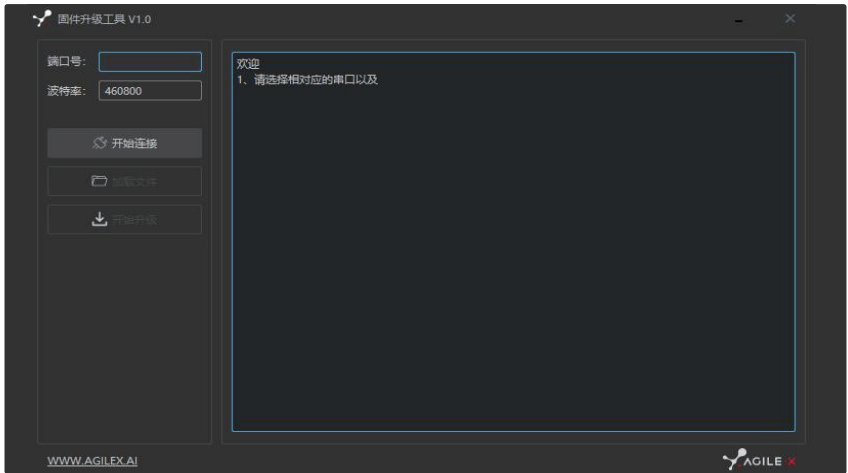


图 3.3 固件升级客户端界面

3.6 TRACER ROS Package 使用示例

ROS提供一些标准操作系统服务,例如硬件抽象,底层设备控制,常用功能实现,进程间消息以及数据包管理。ROS是基于一种图状架构,从而不同节点的进程能接受,发布,聚合各种信息(例如传感,控制,状态,规划等等)。目前ROS主要支持UBUNTU。

开发准备

硬件准备	使用示例环境说明
- CANlight can通讯模块 X1 - Thinkpad E470 笔记本电脑 X1 - AGILEX TRACER 移动机器人底盘 X1 - AGILEX TRACER配套遥控器FS-i6s X1 - AGILEX TRACER顶部航空插座 X1	- Ubuntu 16.04 LTS (此为测试版本,在Ubuntu 18.04 LTS测试过) - ROS Kinetic (后续版本亦测试过) - Git

硬件连接与准备

- 将TRACER顶部航空插头或者尾部插头CAN线引出,将CAN线中的CAN_H和CAN_L分别与CAN_TO_USB适配器相连;
- 打开TRACER移动机器人底盘旋钮开关,检查来两侧的急停开关是否释放;
- 将CAN_TO_USB连接至笔记本的usb口。连接示意如图3.4所示。

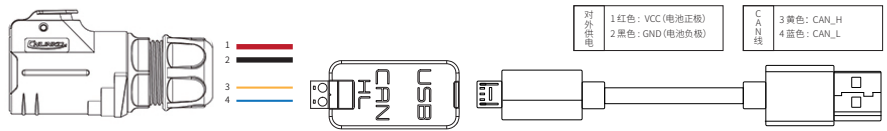


图3.4CAN线连接示意图

ROS 安装和环境设置

安装具体可以参考<http://wiki.ros.org/kinetic/Installation/Ubuntu>

测试CANABLE硬件与CAN 通讯

- 设置CAN-TO-USB适配器
- 使能 gs_usb 内核模块
`$ sudo modprobe gs_usb`
 - 设置500k波特率和使能can-to-usb适配器
`$ sudo ip link set can0 up type can bitrate 500000`
 - 如果在前面的步骤中没有发生错误,您应该可以使用命令立即查看can设备
`$ ifconfig -a`
 - 安装并使用can-utils来测试硬件
`$ sudo apt install can-utils`
 - 若此次can-to-usb已经和TRACER机器人相连,且小车已经开启的情况下,使用下列指令可以监听来自TRACER底盘的数据了
`$ candump can0`
 - 参考来源:
[1]https://github.com/agilexrobotics/agx_sdk
[2]https://wiki.rdu.im/_pages/Notes/Embedded-System/Linux/can-bus-in-linux.html

AGILEX TRACER ROS PACKAGE 下载与编译

- 下载ros 依赖包
`$ sudo apt install ros-$ROS_DISTRO-teleop-twist-keyboard`
`$ sudo apt install libasio-dev`
- 克隆编译tracer_ros源码
`$ cd ~/catkin_ws/src`
`$ git clone --recursive https://github.com/agilexrobotics/ugv_sdk.git`
`$ git clone https://github.com/agilexrobotics/tracer_ros.git`
`$ cd ..`
`$ catkin_make`
参考来源:https://github.com/agilexrobotics/tracer_ros

启动ROS 节点

- 通过can口启动基础节点
`$ roslaunch tracer_bringup tracer_robot_base.launch`
- 启动键盘远程操作节点
`$ roslaunch tracer_bringup tracer_teleop_keyboard.launch`

4 注意事项 Attention

本部分包含一些使用和开发TRACER应该注意的一些事项。

4.1 电池注意事项

- TRACER 产品出厂时电池并不是满电状态的,具体电池电量可以通过TRACER底盘尾部电压显示表显示或者 CAN 总线通信接口读取得到,充电时间以充电器亮绿色指示灯表示充电完毕,但是绿灯亮起后电池依然会以 0.1A 的电流缓慢充电,可以再充 30 分钟左右;
- 请不要在电池使用殆尽以后再进行充电,在TRACER 提示电量低的情况下请及时充电;
- 静态存放条件:存储的最佳温度为-10~45°C,电池在不使用的情况下存放,必须是 2 个月左右充放电一次,然后使电池处于满电压状态进行存放,请勿将电池放入火中,或对电池加热,请勿在高温下存储电池;
- 充电:必须使用配套的锂电池专用充电器进行充电,请勿在0°C以下给电池充电,请勿使用非原厂标配的电池、电源、充电器。

4.2 使用环境注意事项

- TRACER的工作温度为-10~45°C,请勿在工作温度范围外使用;
- TRACER 的使用环境的相对湿度要求是:最大 80%,最小 30%;
- 请勿在存在腐蚀性、易燃性气体的环境或者靠近可燃性物质的环境中使用;
- 不要存在在加热器或者大型卷线电阻等发热体周围;
- 除特别定制版(IP 防护等级定制),TRACER 不具有防水功能,请勿在有雨、雪、积水的环境使用;
- 建议使用环境海拔高度不超过 1000M;
- 建议使用环境昼夜温差不超过 25°C;

4.3 电气外部扩展注意事项

- 尾部扩展电源电流不超过 5A,总功率不超过 120W;
- 当系统检测到电池电压低于安全电压以后,外部电源扩展会被主动切断,所以如果外部扩展识别涉及到重要数据的存储且无掉电保护,建议用户注意。

4.5 其他注意事项

- 搬运以及设置作业时,请勿落下或者倒置;
- 非专业人员,请不要私自拆卸。

4.4 安全注意事项

- 使用过程有疑问,请按照相关说明手册进行操作或者咨询相关技术人员;
- 使用设备前,注意现场情况,避免误操作导致人员安全问题发生;
- 请勿未经技术支持和允许,私自改装内部设备结构。

5 常见问题与解决 Q&A

Q:TRACER启动正常,使用遥控器控制车体不移动?

A:首先确认驱动供电是否正常,急停开关是否被释放;然后确认遥控器的左侧上方模式选择开关选择的控制模式是否正确。

Q:TRACER遥控控制正常,底盘状态、运动信息反馈正常,下发控制帧协议,车体控制模式无法切换,底盘不响应控制帧协议?

A:正常情况下,TRACER若可以通过要遥控器控制正常情况下,说明底盘运动控制正常,可以接收到底盘的反馈帧,说明CAN扩展链路正常。请检查发送的CAN控制帧,看是否切换了指令控制模式,,可以通过底盘反馈的状态帧中错误位中校验错误标志的状态情况。

Q:TRACER在运行中发出“滴-滴-滴...”的声音,该如何处理?

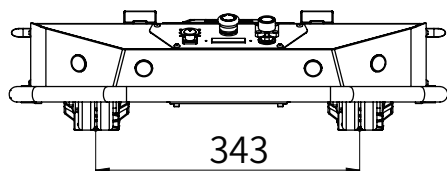
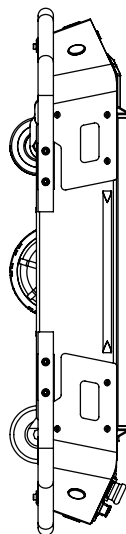
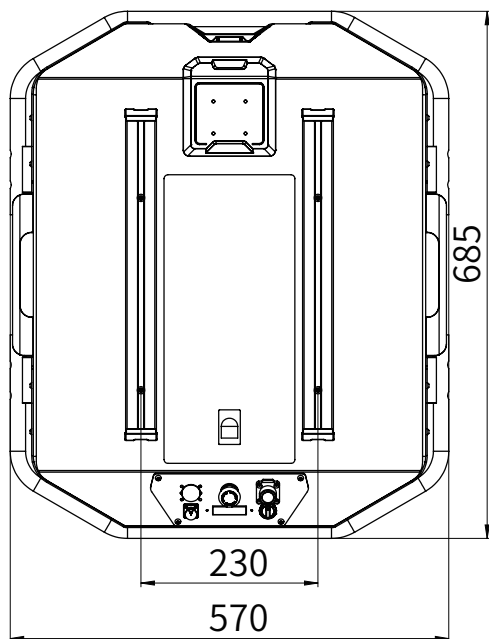
A:若TRACER发出连续的“滴-滴-滴...”表明电池已经处于警报电压状态,请及时充电;出现相关声音以后,亦可能是内部出现相关错误了,可以通过CAN总线检查相关的错误代码,或与相关技术人员沟通。

Q:通过CAN总线进行相关通讯时,底盘反馈指令正常,下发控制小车无响应?

A:TRACER的内部有通讯保护机制,底盘在处理来自外部的CAN控制指令时存在超时保护机制,假设小车收到一帧通讯协议以后,小车超过500MS未收到下一帧控制指令,小车会进入通讯保护,速度为0,所以来自上位机的指令必须时周期性的发布。

6 产品尺寸 Product Dimensions

6.1 产品外形尺寸说明图



AGILE·X

松灵机器人(东莞)有限公司

WWW.AGILEX.AI

TEL:+86-769-22892150

MOBILE:+86-19925374409

