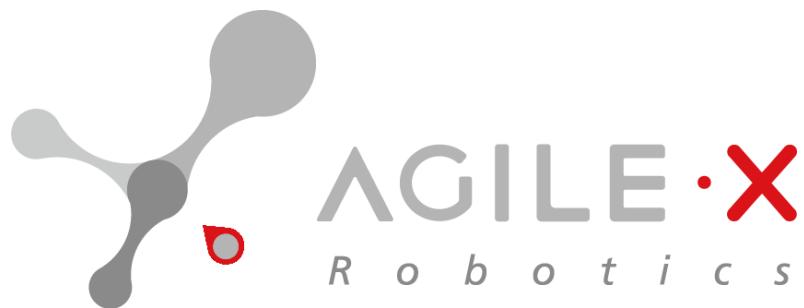


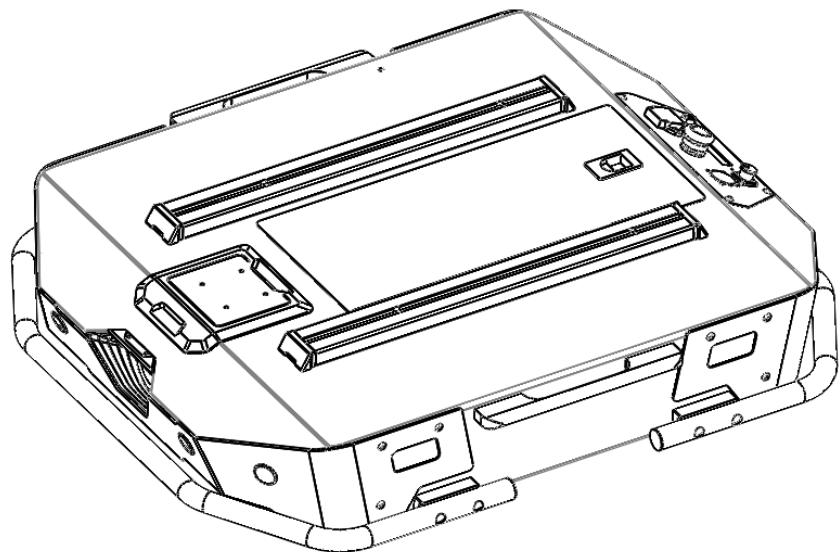
202004



松灵机器人产品 TRACER 用户手册

AGILEX ROBOTICS SUPPORT TEAM

VERSION 1.1 RELEASE



版本信息说明：

版本	变更内容	编写人	审核人	时间
v1.0.0	[新增]建立文档	谢志强	谢志强	2019/11/22
v1.0.1	优化文档内容	何士玉	谢志强	2020/04/16
v1.1.0	更新协议为 2.0	韦克营	谭柱	2020/04/18

0 重要安全信息 Safety Information

1 环境

- 首次使用, 请先**仔细**阅读本手册, 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。

2 检查

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

3 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**。
- 当设备已经有**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**。

3 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**。

目录

1、MINIAGV (TRACER) 简介 Introduction.....	5
1.1 产品列表	5
1.2 性能参数	5
1.3 开发所需	6
2 基本介绍 The Basics	7
2.1 状态指示	8
2.2 电气接口说明	9
2.2.1 尾部电气接口说明	9
2.3 遥控说明	10
2.4 控制指令与运动说明	11
3 使用与开发 Getting Started	12
3.1 使用与操作	12
3.2 充电	13
3.3 开发	13
3.3.1 CAN 接口协议	13
3.3.2 CAN 线的连接	21
3.3.3 CAN 指令控制的实现	21
3.4 串口通信协议	21
3.4.1 串口协议介绍	21
3.4.2 串口协议内容	21
3.4.3 串口的连接	29
4 注意事项 Attention	31
4.2 使用环境注意事项	32
4.3 电气外部扩展注意事项	32
4.4 机械负载注意事项	32
4.5 其他注意事项	32
5 常见问题与解决 Q&A	33
6 产品尺寸 Product Dimensions	34

6.1 产品外形尺寸说明图 34

1、MINIAGV (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他能够在室内灵活运动。立体相机、激光雷达、GPS、IMUS、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	X 1
电池充电器(AC 220V)	X 1
遥控器(选配)	X 1
USB 转串口线	X 1

1.2 性能参数

参数类型	项目	指标
机械参数	长 x 宽 x 高 (mm)	700 X 500X 163
	轴距 (mm)	350
	车体重量 (Kg)	25~30
	电池类型	锂电池 24V 30aH
	电机	直流无刷 2 X 150W
	驱动形式	独立驱动
	悬架	非独立悬架
	转向	差速转向
	安全装备	伺服刹车/防撞管
性能参数指针	空载最高车速 (m/s)	≤2.3
	最小转弯半径	可原地转弯
	最小离地间隙 (mm)	36
	最大爬坡能力	≤10
控制参数	控制模式	遥控控制控制指令模式
	遥控器	2.4G / 极限距离 1Km

	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一个概览视图。

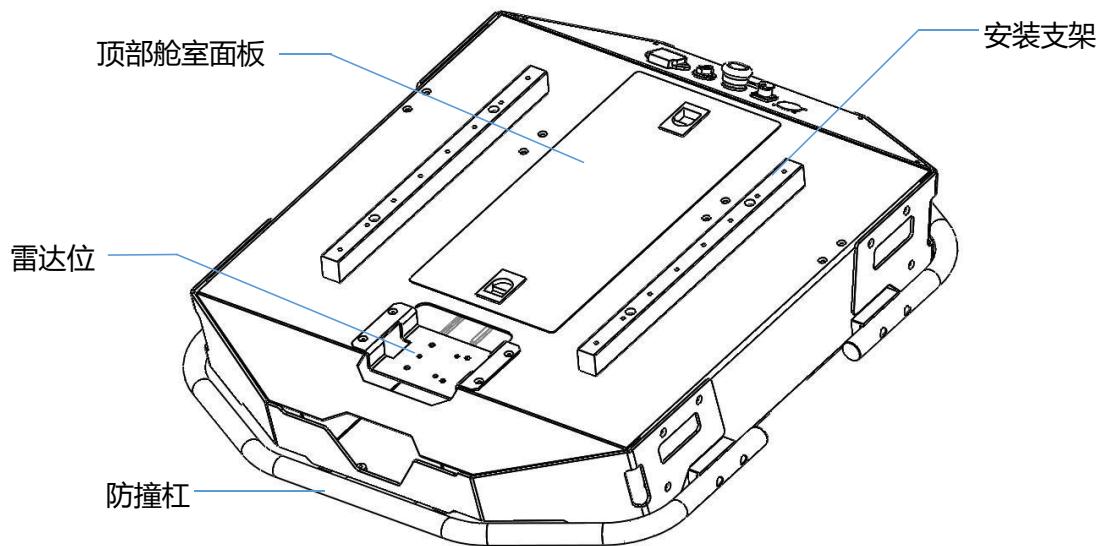


图 2.1 前部概览视图

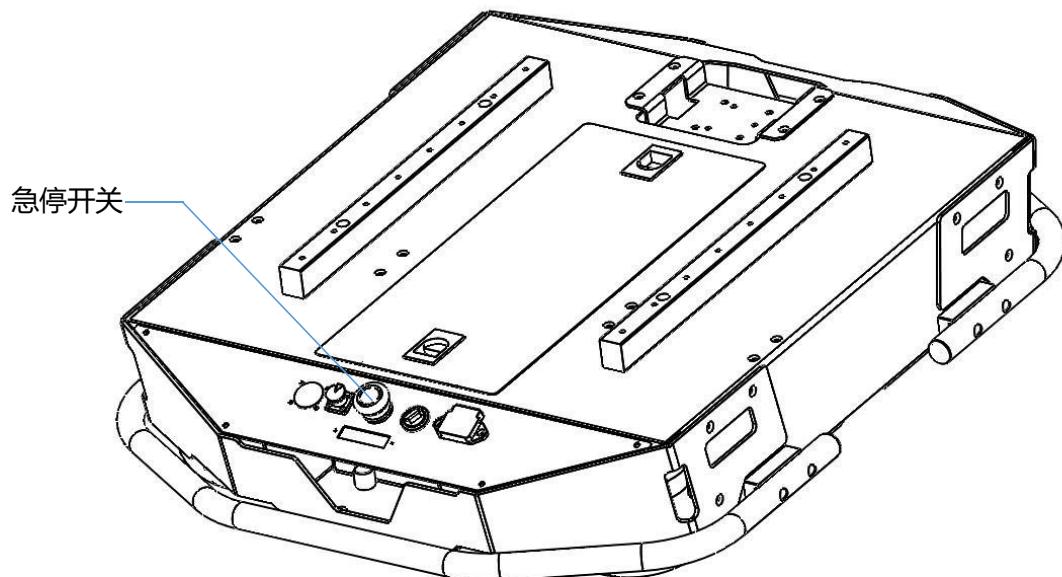


图 2.2 尾部概览视图

TRACER 整体上采用了模块化和智能化的设计思想，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撞管，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，使得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，一方面利用客户的扩展和使用，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。

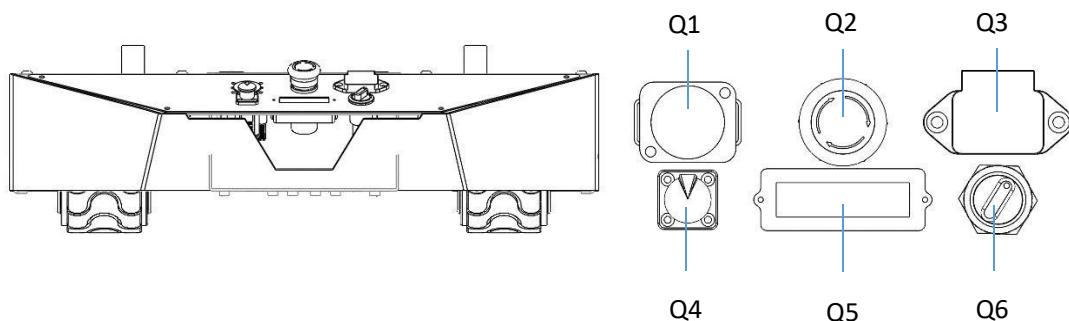
状态	描述
当前电压	当前电池电压可通过尾部电气面板中的电压表查看，精确到 1V
低电压报警	当电池电压低于 22.5V，车体会发出“滴-滴-滴”刺激的声音进行提示。 当检测到电池电压低于 22V 时，TRACER 为了防止电池损坏，会主动切断外部扩展供电和驱动器供电，此时底盘将无法进行运动控制和接受外部指令控制。
上电显示	前灯光亮起

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 2.5 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 2.6 所示。

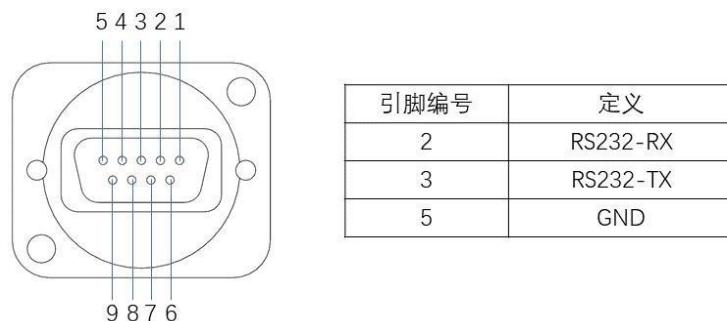


图 2.6 Q4 引脚说明图

尾部配备了和顶部一致的 CAN 通信接口和 24V 电源接口，在内部他们是导通的。其线序的具体定义如图 2.7 所示。



图 2.7 尾部航空接口引脚说明图

2.3 遥控说明

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 2.9。



图 2.9 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 3.0 的坐标参考系。

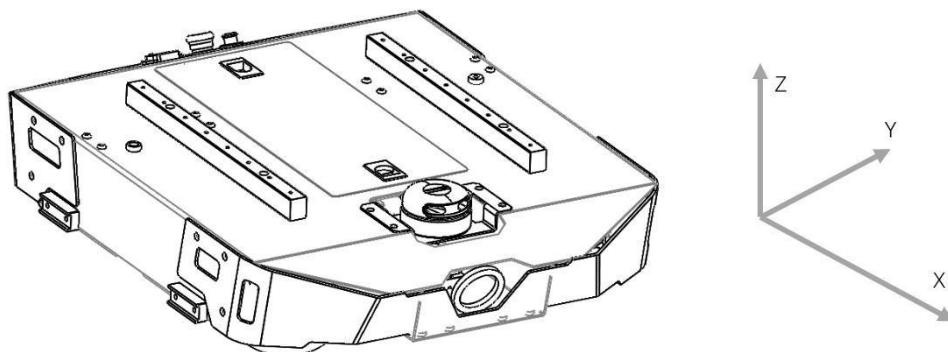


图 3.0 车身参考坐标系示意图

正如 3.0 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 C1 往前推动则为往 X 正方向运动，C1 往后推动则往 X 负方向运动，C1 推动至最大值时，往 X 方向运动速度最大，C1 推动至最小值时，往 X 方向负方向运动速度最大；遥控器摇杆 C2 左右控制车体的旋转运动，C2 往左推动车体则由 X 轴正反向往 Y 正方向旋转，C2 往右推动车体则由 X 轴正方向往 Y 负方向旋转，C2 往左推动至最大值时，逆时针方向旋转线速度最大，C2 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

启动操作基本操作流程如下：

检查

- 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；
- 检查急停开关状态。确认急停按钮均处于释放状态；

启动

- 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；
- 检查电池电压，若电量低，请充电；

关闭操作

- 旋转钥匙开关，即可切断电源；

急停

- 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，将充电器中开关打开，即可进入充电状态。

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

系统状态回馈指令				
指令名称	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点	接收节点	0x211	20ms	无
线控底盘	决策控制单元			
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 【表 3.2】	
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]	旋转速度高八位	signed int16	车体旋转角速度 单位: 0.001rad/s	
byte [3]	旋转速度低八位			
byte [4]	保留	-	0x00	
byte [5]	保留	-	0x00	
byte [6]	保留	-	0x00	
byte [7]	保留	-	0x00	

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	
byte [1]	线速度低八位			
byte [2]	角速度高八位	signed int16	车体旋转速度 单位 0.001rad/s	
byte [3]	角速度低八位			
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 灯光控制帧

灯光控制帧				
指令名称	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点				
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 ^[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

灯光控制反馈帧				
指令名称	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点				
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]	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]	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	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点				
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 串口控制模模式	

注 1, 控制模式说明

TRACER 在遥控器不上电的情况下, 控制模式默认是指令控制模式, 即可以直接通过指令控制底盘, 但是即使底盘处于指令模式下, 如果要成功执行指令中的速度指令, 在指令中的控制模式依然需要设为 0x01。若在打开遥控器, 遥控器具有最高权限, 可以屏蔽指令的控制, 可以切换控制模式。

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	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点	接收节点	0x441	无	无
线控底盘	决策控制单元			
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 状态置位帧说明表

运动控制回馈指令				
指令名称	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点	接收节点	0x311	20ms	无
线控底盘	决策控制单元			
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-		
byte [5]	右轮里程计次高位			
byte [6]	右轮里程计次低位		右轮里程计数据 单位 mm	
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：

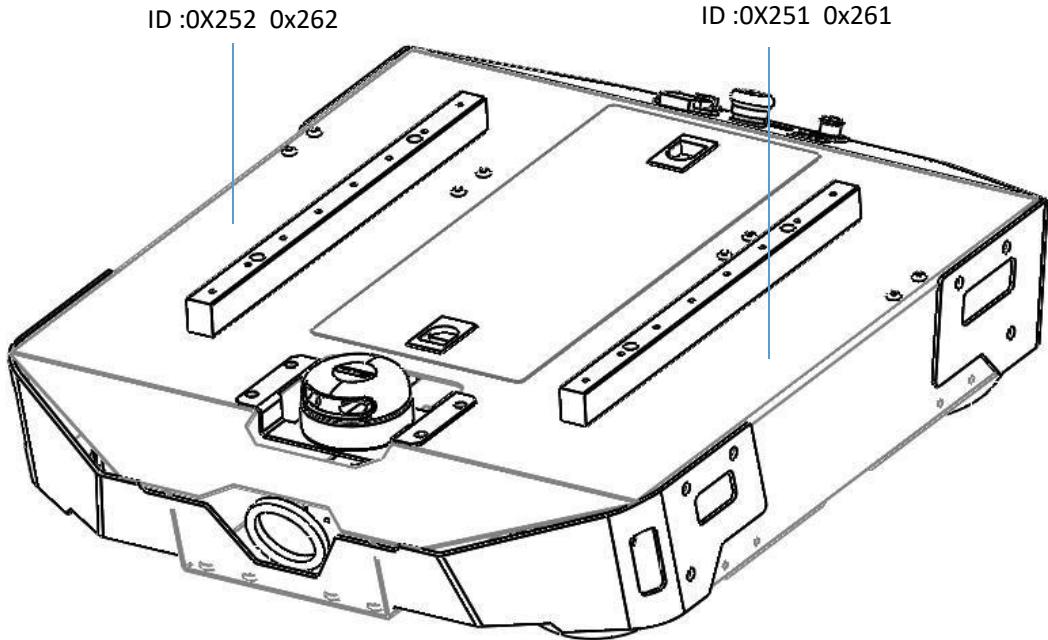


图 3.0 电机反馈 ID 示意图

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	接收节点	ID	周期 (ms)	接收超时(ms)
发送节点	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 单位： RPM	
byte [1]	电机转速低八位			
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 2.2。



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 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。

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

● 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)

```
/*
 * @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

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 (未启用) 0x01 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 ^[1] 0x02 串口控制模模式
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]	故障信息高八位	unsigned int16	详见备注 ^[**]
byte [5]	故障信息低八位*		

故障信息说明		
字节	位	含义
byte [4]	bit [0]	CAN 通信控制指令校验错误 (0:无故障 1: 故障)
	bit [1]	电机驱动过温警告 ^[1] (0:无警告 1: 警告) 温度限制为 55°C
	bit [2]	电机过流故警告 ^[1] (0:无警告 1: 警告) 电流有效值 15A
	bit [3]	电池欠压警告 (0:无警告 1: 警告) 报警电压为 22.5V
	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
byte [5]	bit [0]	电池欠压故障 (0:无故障 1: 故障) 保护电压为 22V
	bit [1]	电池过压故障 (0:无故障 1: 故障)
	bit [2]	电机 1 通讯故障 (0:无故障 1: 故障)
	bit [3]	电机 2 通讯故障 (0:无故障 1: 故障)
	bit [4]	电机 3 通讯故障 (0:无故障 1: 故障)
	bit [5]	电机 4 通讯故障 (0:无故障 1: 故障)

	bit [6]	电机驱动过温保护 ^[2] (0:无保护 1: 保护) 温度限制为 65°C
	bit [7]	电机过流保护 ^[2] (0:无保护 1: 保护) 电流有效值为 20A

[1]: 机器人底盘固件版本 V1.2.8 后续版本支持, 之前版本需要升级固件方可支持

[2]: 电机驱动器过温和电机过流警告内部不处理, 仅作置位处理, 提供给上位机做一些预先处理, 如果驱动器过流, 建议作降低车速的处理, 如果是过温建议降低速度, 等待温度降低, 此标志位会随温度回归正常, 电流回归正常主动清除;

[3]: 电机驱动器过温和电机过流保护内部会做出处理, 当电机驱动器温度高于保护温度时, 会限制驱动器输出, 车辆会缓慢停下, 运动控制指令速度控制值无效, 此标志位不会主动清除, 需要上位机发送清除故障保护指令, 清除指令后才可以正常执行运动控制指令内容。

■ 运动指控行回反馈指令

运动控制回反馈指令			
指令名称	接收节点	周期 (ms)	接收超时(ms)
发送节点			
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

控制指令			
指令名称	接收节点	周期 (ms)	接收超时(ms)
发送节点			
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 串口控制模模式
byte [1]	故障清除指令	unsigned int8	详见备注 2*
byte [2]	线速度百分比	signed int8	最大速度 1.5m/s, 值域为 (-100, 100)
byte [3]	角速度百分比	signed int8	最大速度 0.7853 rad/s, 值域为 (-100, 100)
byte [4]	保留	-	0x00
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	接收节点	周期 (ms)	接收超时(ms)
发送节点	接收节点	20ms	无
线控底盘	决策控制单元		
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号驱动器电流高八位	unsigned int16	实际电流 X 10 (精确到 0.1A)
byte [1]	1号驱动器电流低八位		
byte [2]	1号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)
byte [3]	1号驱动器转速低八位		
byte [4]	1号驱动器温度	signed int8	实际温度 (精确到 1°C)
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	接收节点	周期 (ms)	接收超时(ms)
发送节点	接收节点	20ms	无
线控底盘	决策控制单元		
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号驱动器电流高八位	unsigned int16	实际电流 X 10 (精确到 0.1A)
byte [1]	2号驱动器电流低八位		
byte [2]	2号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)
byte [3]	2号驱动器转速低八位		
byte [4]	2号驱动器温度	signed int8	实际温度 (精确到 1°C)
byte [5]	保留	--	0x00

■ 灯光控制帧

灯光控制帧			
指令名称			
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 [5]
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

灯光控制反馈帧			
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	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 0.15m/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]	故障清除指令	0x00
byte [2]	线速度百分比	0x0A
byte [3]	角速度百分比	0x00
byte [4]	保留	0x00
byte [5]	保留	0x00

整串数据内容为：

5A A5 0A 55 01 02 00 0A 00 00 00 00 6B

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。

3.5 固件升级

为了方便解决客户对 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

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 6S 等待，如果时间超过 6S 则会进行进入应用程序）；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。

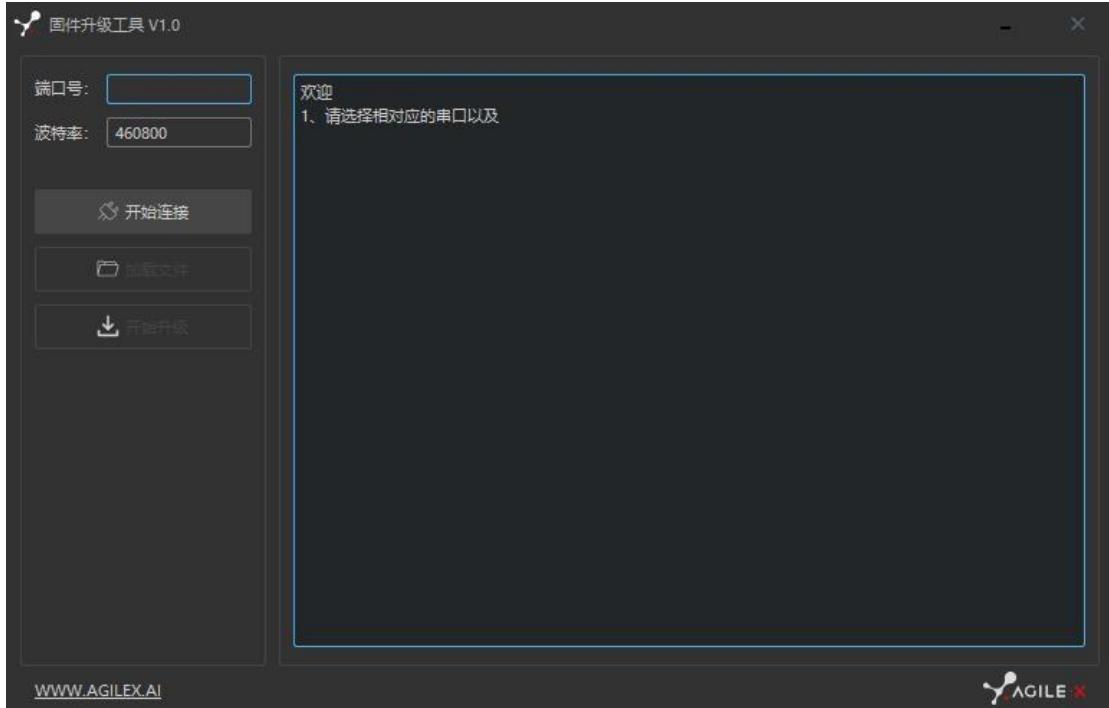


图 3.3 固件升级客户端界面

4 注意事项 **Attention**

本部分包含一些使用和开发 TRACER 的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-20℃~60℃，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℃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 室外工作温度为-10℃~45℃，请勿在室外温度低于-10℃、高于 45℃ 环境中使用；
- TRACER 室内工作温度为 0℃~42℃，请勿在室内温度低于 0℃、高于 42℃ 环境中使用；
- 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℃；

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换，所以如果外部扩展设备涉及到重要数据的存储且无掉电保护，建议用户注意。

4.4 机械负载注意事项

4.5 其他注意事项

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

5 常见问题与解决 Q&A

6 产品尺寸 Product Dimensions

6.1 产品外形尺寸说明图

