SCOUT 2.0

AgileX Robotics Team

用户手册 V.2.0.0 2020.08

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。
SCOUT的集成商和终端客户有责任确保遵循相关规定和切实的法律法规，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包括但不限于自动防撞、防跌落、生物接近预警等相关安全功能，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。
- 在操作和使用设备之前已经知晓可能存在的安全风险。

2.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。

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。

4.操作

- 保证使用时周围区域相对空旷。
- 在视距内遥控控制。
- SCOUT2.0最大的载重为50KG，在使用时，确保有效载荷不超过50KG。
- SCOUT 2.0安装外部扩展时，确认扩展的质心位置，确保在旋转中心。
- 当设备低电量报警时请及时充电。
- 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。
- 如果车辆原地旋转过程中出现摇晃，调节悬挂。

5.保养

- 轮胎气压定期检查，轮胎气压保持在1.8BAR~2.0BAR。
- 轮胎磨损严重或者爆胎，请及时更换。
- 如果长时间不使用电池，需要按照2到3个月对电池进行周期性充电。

目录

1 SCOUT2.0简介 Introduction

SCOUT2.0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具有广泛的应用领域。立体相机、激光雷达、GPS、IMU、机械手等设备可选择加装至SCOUT2.0作为扩展应用。SCOUT2.0 可被应用到无人巡检、安防、科研、勘探、物流等领域。

1.1 产品列表

名称	数量
SCOUT2.0 机器人本体	x1
电池充电器(AC 220V)	x1
航空插头公头（4Pin）	x2
USB转RS232	x1
USB转CAN通讯模块	x1
富斯遥控器(选配)	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 ）	930 X 699 X 348
	轴距 (mm)	498
	前 / 后轮距（mm）	582/ 582
	车体重量（Kg）	62
	电池类型	锂电池24V 30Ah
	电机	直流无刷 4 X 200W
	减速箱	1:30
	驱动形式	四轮独立驱动
	悬架	单摇臂独立悬架
	转向	四轮差速转向
	安全装备	伺服刹车/防撞管
性能参数	空载最高车速（m/s）	1.5
	最小转弯半径	可原地转弯
	最大爬坡能力	30°
	最小离地间隙（mm）	135
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

SCOUT2.0出厂时可选配遥控器，用户可以通过遥控器控制 SCOUT2.0 移动机器人底盘，完成移动和旋转操作；SCOUT2.0配备了 CAN 和 RS232接口，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SCOUT2.0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SCOUT2.0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。

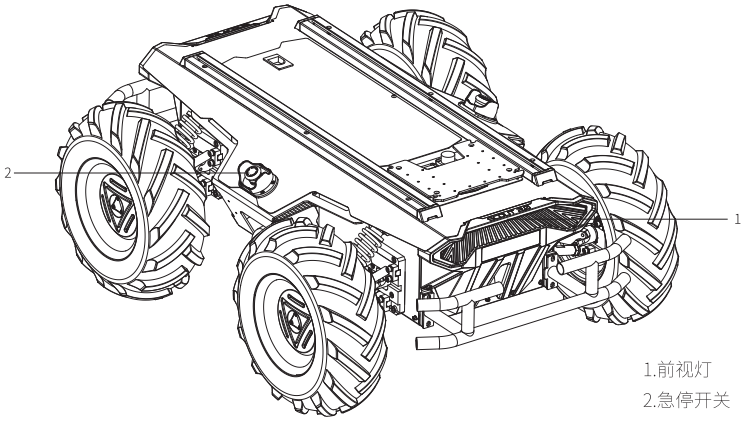


图 2.1 前部概览视图

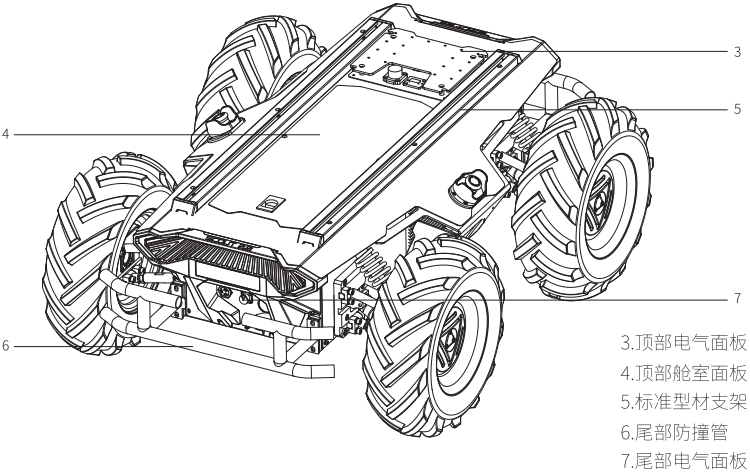


图 2.2 尾部概览视图

SCOUT2.0整体上采用了模块化和智能化的设计理念，在动力模块上采用充气胶轮与独立悬挂的复合设计，再加上动力强劲的直流无刷伺服电机，使得SCOUT2.0机器人底盘开发平台具有很强的通过性和地面适应性，可在不同的地面灵活运动。
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。
车体前后均有安装灯光，前侧采用白光设计，可进行照明；尾部采用显眼的红色，可用作警示灯和指示灯。

左右两侧分别有一个紧急停车开关，采用冗余设计，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。
在车的尾部和顶部均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中使用。
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SCOUT2.0上的电压表、蜂鸣器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”刺耳的声音进行提示。 **当检测到电池电压低于22V时，SCOUT2.0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	前后灯光亮起

表格 2.1 车体状态说明表

2.2 电气接口说明

2.2.1顶部电气接口说明

SCOUT2.0提供了3组4引脚航空外部扩展接口以及一组DB9（RS232）通信控制接口。顶部航空接口位置如图2.3所示。

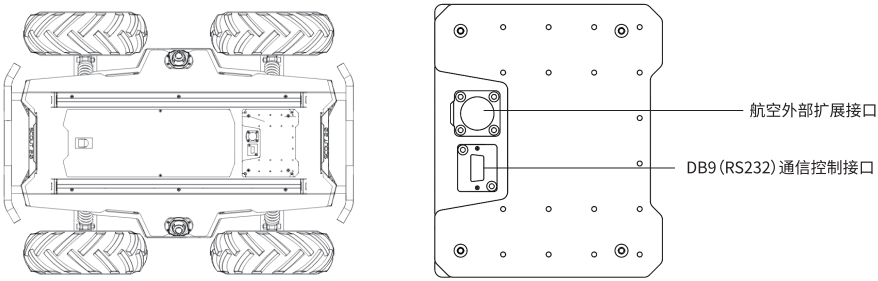
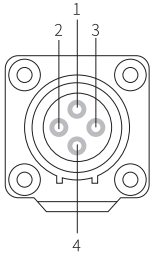


图 2.3 SCOUT 2.0顶部电气界面示意图

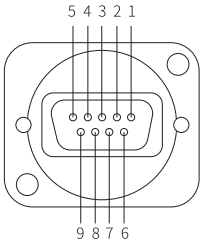
SCOUT2.0顶部、前面以及尾部均设置有一个航空扩展接口，航空扩展接口配置了一组电源接口以及一组CAN通讯接口。便于使用者给扩展设备提供电源，以及通讯使用。其具体引脚定义图2.4。
需要注意的是，这里的扩展电源受内部控制，当电池电压低于安全电压时会主动切断供电，所以用户需要注意，在达到临界电压前SCOUT2.0平台会发出低电压报警通知，用户在使用过程中注意充电。



引脚编号	引脚类型	功能及定义	备注
1	电源	VCC	电源正，电压范围23~29.2v,最大电流10A
2	电源	GND	电源负
3	CAN	CAN_H	CAN总线高
4	CAN	CAN_L	CAN总线低

图 2.4顶部航空拓展接口引脚说明图

关于顶部其具体引脚定义如图2.5所示。



引脚编号	定义
2	RS232-RX
3	RS232-TX
5	GND

图 2.5 顶部串口接口引脚说明图

2. 2. 2 尾部电气接口说明

尾部的扩展接口如图2.6所示，其中Q1为旋钮开关，是电气部分总开关；Q2为充电接口；Q3为驱动系统供电开关；Q4为蜂鸣器；Q5为CAN和24V电源扩展接口；Q6为电源显示交互。

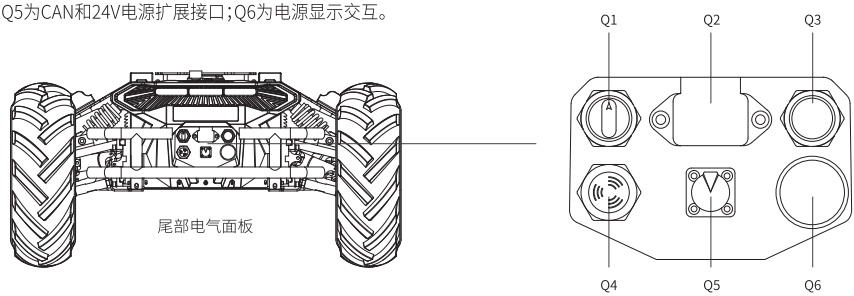
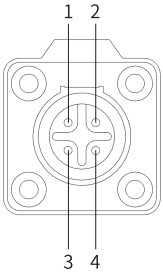


图 2.6尾部视图



引脚编号	引脚类型	功能及定义	备注
1	电源	VCC	电源正，电压范围23~29.2v,最大电流10A
2	电源	GND	电源负
3	CAN	CAN_H	CAN总线高
4	CAN	CAN_L	CAN总线低

图 2.7 前面和尾部航空接口引脚说明图

2.3 遥控说明

富斯遥控器为SCOUT2.0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SCOUT2.0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。

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SCOUT2.0前进和后退；S2控制旋转，POWER为电源按钮，同时按住即可开机 。

图2.8 富斯遥控器按键示意图

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2.9的坐标参考系。

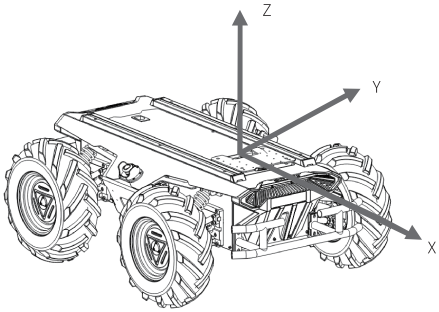


图2.9 车身参考坐标系示意图

正如2.9所展示的，SCOUT2.0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方向运动速度最大，S1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S2左右控制车体的旋转运动，S2往左推动车体则由X轴正反向往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。

2.5 灯光控制说明

SCOUT2.0前后均配置了灯光，为了方便客户，SCOUT2.0对外开放灯光控制接口。同时为了节省能源，在遥控器预留灯光控制接口。
遥控的版本目前只支持FS遥控器，其他遥控器的适配工作还在进行中。目前遥控器的灯光模式有3种，模式切换可以通过SWC拨杆切换：
模式控制说明：SWC拨杆拨至最下方为常闭模式，中间为常开模式，最上方为呼吸灯模式.

- **常闭模式**：在常闭的模式下，如果底盘静止，前灯会关闭，尾灯为了表示当前的工作状态，会进入呼吸灯模式；如果底盘在正常速度行驶状态，尾灯关闭，前灯打开；
- **常开模式**：在常开的模式下，如果底盘静止不动，前灯常开，尾灯为了表示静止状态会进入呼吸灯模式；如果在运动模式下，尾灯关闭，前灯打开；
- **呼吸灯模式**：前灯尾灯在各种状态下均为呼吸灯模式。

3 使用与开发 Getting Started

本部分主要介绍SCOUT2.0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

- 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。确认两个急停按钮均处于释放状态；
- 取下尾部电气面板盖板，可看见尾部电气面板；
- 初次使用时确认尾部电气面板中Q3（驱动电源开关）是否被按下，如按下，请按下后释放，则处于释放状态，此时驱动器处于断电状态。

启动

- 旋转旋钮开关(电气面板中Q1)，正常情况下，电压表正常显示电池电压，前后尾灯均正常亮起；
- 检查电池电压，如未有“滴-滴-滴…”连续蜂鸣器声音，表示电池电压正常，若电量低，请充电；
- 按下Q3（驱动电源开关按钮）。

关闭操作

- 旋转旋钮开关，即可切断电源。

急停

- 按下SCOUT2.0车体左右两侧的急停开关即可。

遥控控制基本操作流程

正常启动SCOUT2.0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SCOUT2.0平台运动。

3.2 充电

SCOUT2.0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SCOUT2.0底盘处于停机断电状态。充电前请确认尾部电气控制台中Q1(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Q2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。

注意：当前电池从22V充满电状态大约需要3~5小时，电池充满电电压约为29.2V（此处的电池电压为三元锂电池类型，如果电池类型为磷酸铁锂最高电压为26.8V）；充电时间计算 30aH÷10A=3H

3.3 开发

SCOUT2.0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

SCOUT2.0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移动的线速度以及旋转的角速度；SCOUT2.0会实时反馈当前的运动状态信息以及SCOUT2.0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，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 SCOUT 2.0 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	0x151	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] byte [3]	电池电压高八位 电池电压低八位	unsigned int16	实际电压X 10 (精确到0.1V)	
byte [4] byte [5]	故障信息高八位 故障信息低八位	unsigned int16	详见备注[故障信息说明]	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

表格 3.2 故障信息说明表

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]	电机过流故障警告[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]	遥控器失联保护（0：正常，1：遥控器失联）
	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][3]（0:无保护 1：保护）温度限制为65°C
	bit [7]	电机过流保护[2][3]（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，此标志位会随着温度回归正常，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	0x131	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	移动速度高八位 移动速度低八位	signed int16	实际速度X 1000 (精确到0.001m/s)	
byte [2] byte [3]	旋转速度高八位 旋转速度低八位	signed int16	实际速度X 1000 (精确到0.001rad/s)	
byte [4]	保留	-	0x00	
byte [5]	保留	-	0x00	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

控制帧包含了模式控制、故障清除指令、线速度控制开度、角速度控制开度以及检验和，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	0x130	20ms	500ms
数据长度	0x08			
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]	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0.5235rad/s,值域为 (-100，100)	
byte [4]	保留	—	0x00	
byte [5]	保留	—	0x00	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

[注1]控制模式说明

SCOUT2.0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。

[注2]故障清除指令信息：

- 0X00 无故障清除指令
- 0X01 清除电池欠压故障
- 0X02 清除电池过压故障
- 0X03 清除电机1通讯故障
- 0X04 清除电机2通讯故障
- 0X05 清除电机3通讯故障
- 0X06 清除电机4通讯故障
- 0X07 清除电机驱动过温故障
- 0X08 清除电机过流故障

[注3]示例数据，以下数据仅供测试使用

1，小车以0.15/S的速度前进

byte [0]	byte [1]	byte [2]	byte [3]	byte [4]	byte [5]	byte [6]	byte [7]
0x01	0x00	0x0a	0x00	0x00	0x00	0x00	0x44

2. 小车以0.05235RAD/S旋转

byte [0]	byte [1]	byte [2]	byte [3]	byte [4]	byte [5]	byte [6]	byte [7]
0x01	0x00	0x00	0x0a	0x00	0x00	0x00	0x44

3. 小车静止，切换控制模式为指令模式（遥控器不开启的情况下测试）

byte [0]	byte [1]	byte [2]	byte [3]	byte [4]	byte [5]	byte [6]	byte [7]
0x01	0x00	0x00	0x00	0x00	0x00	0x00	0x3a

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的电流信息、编码器以及温度信息。下面的帧反馈是电机的电流信息、编码器信息以及电机温度信息：

在底盘中四个电机的电机编号对应为如下图所示：

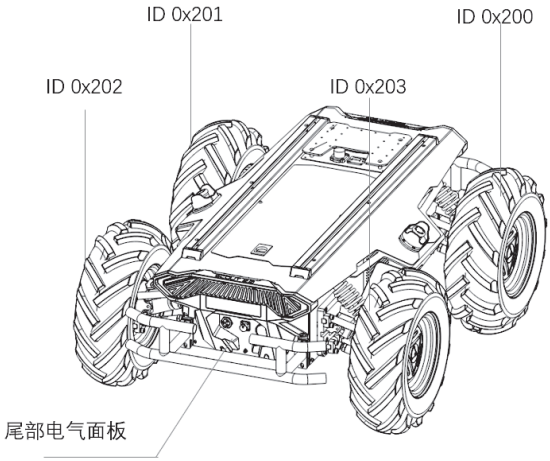


图3.0 电机反馈ID示意图

表格 3.5 1号电机信息反馈

指令名称	1号电机驱动器信息反馈帧			
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	0x200	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	1号驱动器电流高八位 1号驱动器电流低八位	unsigned int16	实际电流X 10 (精确到0.1A)	
byte [2] byte [3]	1号驱动器转速高八位 1号驱动器转速低八位	signed int16	电机输出轴实际速度 (RPM)	
byte [4]	1号驱动器温度	signed int8	实际温度 (精确到1°C)	
byte [5]	1号电机温度	signed int8	实际温度 (精确到1°C)	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

表格 3.6 2号电机信息反馈

指令名称	2号电机驱动器信息反馈帧			
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	0x201	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	2号驱动器电流高八位 2号驱动器电流低八位	unsigned int16	实际电流X 10 (精确到0.1A)	
byte [2] byte [3]	2号驱动器转速高八位 2号驱动器转速低八位	signed int16	电机输出轴实际速度 (RPM)	
byte [4]	2号驱动器温度	signed int8	实际温度 (精确到1°C)	
byte [5]	2号电机温度	signed int8	实际温度 (精确到1°C)	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

表格 3.7 3号电机信息反馈

指令名称	3号电机驱动器信息反馈帧			
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	0x202	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	3号驱动器电流高八位 3号驱动器电流低八位	unsigned int16	实际电流X 10 (精确到0.1A)	
byte [2] byte [3]	3号驱动器转速高八位 3号驱动器转速低八位	signed int16	电机输出轴实际速度 (RPM)	
byte [4]	3号驱动器温度	signed int8	实际温度 (精确到1°C)	
byte [5]	3号电机温度	signed int8	实际温度 (精确到1°C)	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

表格 3.8 4号电机信息反馈

指令名称	4号电机驱动器信息反馈帧			
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	0x203	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	4号驱动器电流高八位 4号驱动器电流低八位	unsigned int16	实际电流X 10 (精确到0.1A)	
byte [2] byte [3]	4号驱动器转速高八位 4号驱动器转速低八位	signed int16	电机输出轴实际速度 (RPM)	
byte [4]	4号驱动器温度	signed int8	实际温度 (精确到1°C)	
byte [5]	4号电机温度	signed int8	实际温度 (精确到1°C)	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

表格 3.9 灯光控制帧

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	0x140	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	
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	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

[5]: 此值自在自定义模式下有效

表格 3.10 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	0x141	25ms	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[5]	
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	
byte [6]	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	
byte [7]	校验位(checksum)	unsigned int8	校验位	

数据校验位为每一帧CAN消息的数据段最后一个有效字节，其校验和的计算方法checksum =(ID_H + ID_L + data_length+ can_msg.data[0] + can_msg.data[1] + can_msg.data[2] + can_msg.-data[3] + can_msg.data[4]+ …+ can_msg.-data[n]) & 0xFF：

- ID_H 与 ID_L为ID分别是帧ID的高八位和低八位。比如ID为0x540，那么对应的ID_H为0x05，ID_L为0x40；
- Data_length为数据长度为一帧CAN消息中数据段有效数据长度，包含校验和这个字节；
- can_msg.data[n]为有效数据段中具体每个字节的具体内容，计数校验位是需要参与校验和计算的，校验和本身不参与计算。

```
/**
 *@brief CAN message checksum example code
 * @param[in]  id : can id
 * @param[in]  *data : can message data struct pointer
 * @param[in]  len : can message data length
 * @return  the checksum result
 */
static uint8 Agilex_CANMsgChecksum(uint16 id, uint8 *data, uint8 len)
{
  uint8 checksum = 0x00;
  checksum = (uint8)(id & 0x00ff) + (uint8)(id >> 8) + len;
  for(uint8 i = 0 ; i < (len-1); i++)
  {
      checksum += data[i];
  }
  return checksum;
}
```

图 3.1 CAN消息校验算法

3.3.2 CAN线的连接

SCOUT2.0随车发货提供了两个航空插头公头如图3.2。线的定义可参考表2.2。

3.3.3 CAN指令控制的实现

正常启动SCOUT2.0移动机器人底盘，打开遥控器，然后将控制模式切换至指令控制，即将遥控器SWB模式选择拨至最上方，此时SCOUT2.0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。

图 3.2 航空插头公头示意图

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。

```
/**
* @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];
    }
    return checksum;
 }
```

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 (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 0x02 串口控制模模式
byte [2] byte [3]	电池电压高八位 电池电压低八位	unsigned int16	实际电压X 10 (精确到0.1V)
byte [4] byte [5]	故障信息高八位 故障信息低八位	unsigned int16	详见备注[故障信息说明]

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][3]（0:无保护 1：保护）温度限制为65°C
	bit [7]	电机过流保护[2][3]（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，此标志位会随着温度回归正常，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] byte [1]	移动速度高八位 移动速度低八位	signed int16	实际速度X 1000 (精确到0.001m/s)
byte [2] byte [3]	旋转速度高八位 旋转速度低八位	signed int16	实际速度X 1000 (精确到0.001rad/s)
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]	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0.5235 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

指令名称	1号电机驱动器信息反馈帧		
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	500ms
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] byte [1]	1号驱动器电流高八位 1号驱动器电流低八位	unsigned int16	实际电流X 10 (精确到0.1A)
byte [2] byte [3]	1号驱动器转速高八位 1号驱动器转速低八位	signed int16	电机输出轴实际速度 (RPM)
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

指令名称	2号电机驱动器信息反馈帧		
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	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

3号电机驱动器信息反馈帧

指令名称	3号电机驱动器信息反馈帧		
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	0x05		
数据域长度	6		
位置	功能	数据类型	说明
byte [0]	3号驱动器电流高八位	unsigned int16	实际电流X 10 (精确到0.1A)
byte [1]	3号驱动器电流低八位		
byte [2]	3号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)
byte [3]	3号驱动器转速低八位		
byte [4]	3号驱动器温度	signed int8	实际温度 (精确到1°C)
byte [5]	保留	-	0x00

4号电机驱动器信息反馈帧

指令名称	4号电机驱动器信息反馈帧		
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	0x06		
数据域长度	6		
位置	功能	数据类型	说明
byte [0] byte [1]	4号驱动器电流高八位 4号驱动器电流低八位	unsigned int16	实际电流X 10 (精确到0.1A)
byte [2] byte [3]	4号驱动器转速高八位 4号驱动器转速低八位	signed int16	电机输出轴实际速度(RPM)
byte [4]	4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

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	无
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

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

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

- 其他说明：**本协议需要固件版本V1.3.3以上**

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

为了方便用户对SCOUT2.0所使用的固件版本进行升级，给客户带来更加完善的体验，SCOUT2.0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
- SCOUT2.0 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

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SCOUT2.0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；
- SCOUT2.0底盘上电，立即点击开始连接（SCOUT2.0底盘会在上电前6S，如果时间超过6S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。

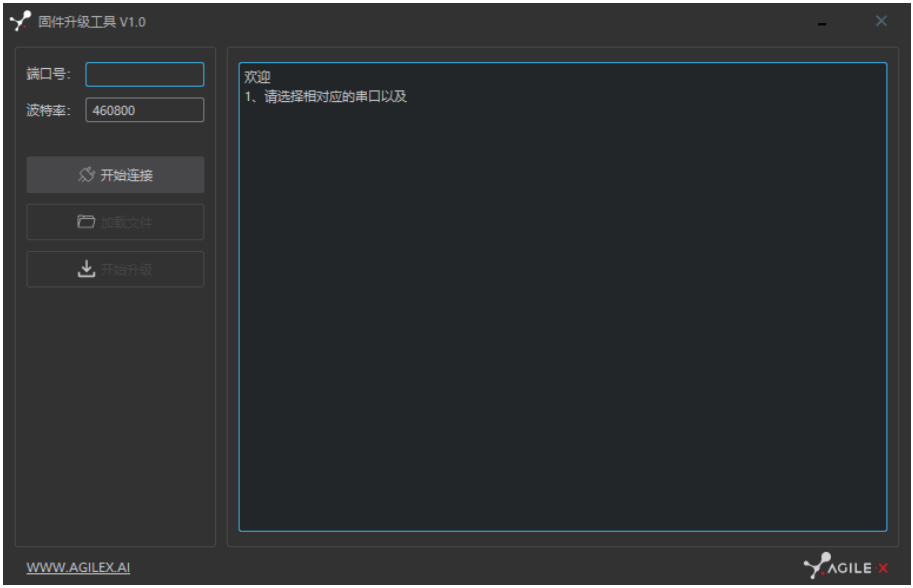


图 3.3 固件升级客户端界面

3.6 SCOUT2.0 SDK 使用示例

为了使用户更加便捷的进行机器人相关方面的开发，我们针对SCOUT2.0移动机器人平台开发了相应的跨平台支持的SDK。SDK软件包中提供一个基于C++接口，用于与SCOUT2.0移动机器人底盘进行通讯，可以获取机器人的最新状态和控制机器人的基本动作。目前通讯适配的有CAN和RS232。我们在NVIDIA JETSON TX2进行了相关测试。

3.7 SCOUT2.0 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

- CANlight can通讯模块 X1
- Thinkpad E470 笔记本电脑 X1
- AGILEX SCOUT 2.0 移动机器人底盘 X1
- AGILEX SCOUT 2.0 配套遥控器FS-i6s X1
- AGILEX SCOUT 2.0 顶部航空插座 X1

使用示例环境说明

- 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

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SCOUT2.0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SCOUT 2.0底盘的数据了
  $ candump can0
- 参考来源：[1]https://github.com/agilexrobotics/scout_sdk
  [2]https://wiki.rdu.im/_pages/Notes/Embedded-System/Linux/can-bus-in-linux.html

AGILEX SCOUT ROS PACKAGE 下载与编译

- 下载ros 依赖包
  $ sudo apt install ros-melodic-controller-manager
  $ sudo apt install ros-melodic-teleop-twist-keyboard
- 克隆编译scout_ros源码
  $ cd ~/catkin_ws/src
  $ git clone https://github.com/agilexrobotics/scout_ros.git
  $ cd ..
  $ catkin_make
- 参考来源：https://github.com/agilexrobotics/scout_ros

硬件连接与准备

- 将SCOUT2.0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SCOUT2.0移动机器人底盘旋钮开关，按下驱动器电源控制开关，正常情况下可以听见蜂鸣器‘滴’的一声，表示驱动器已经正常上电，如果没有此声音，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，可以再次测试一下CAN是否正常

启动ROS 节点

- 启动基础节点
  $ roslaunch SCOUT_bringup SCOUT_minimal.launch
  启动键盘远程操作节点
- $ roslaunch SCOUT_bringup SCOUT_teleop_keyboard.launch

4 注意事项 Attention

本部分包含一些使用和开发SCOUT2.0应该注意的一些事项。

4.1电池注意事项

- SCOUT2.0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SCOUT2.0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SCOUT2.0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°C~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。
- 保护：针对因为过载、短路或者外部东西覆盖导致的过热和过流，在电池中BMS中带有过温、过流、过压、欠压保护。

4.2使用环境注意事项

- SCOUT2.0工作温度为-10°C~45°C，请勿在温度低于-10°C、高于45°C环境中使用；
- SCOUT2.0 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；
- 不要保存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），SCOUT2.0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；
- 轮胎气压定期检查，轮胎气压保持在1.8BAR~2.0BAR。
- 轮胎磨损严重或者爆胎，请及时更换。

4.3电气外部扩展注意事项

- 顶部扩展电源电流不超过10A,总功率不超过150W；
- 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。

4.5其他注意事项

- SCOUT2.0前后为塑料件，请勿直接捶打，否则容易损坏；
- 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；
- 遇到紧急情况，通过拍下停急停按钮，断电设备；
- 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：SCOUT2.0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，驱动器的电源开关是否被按下，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:SCOUT2.0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，SCOUT2.0若可以通过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，看数据校验是否正确，控制模式是否置为指令控制模式，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。

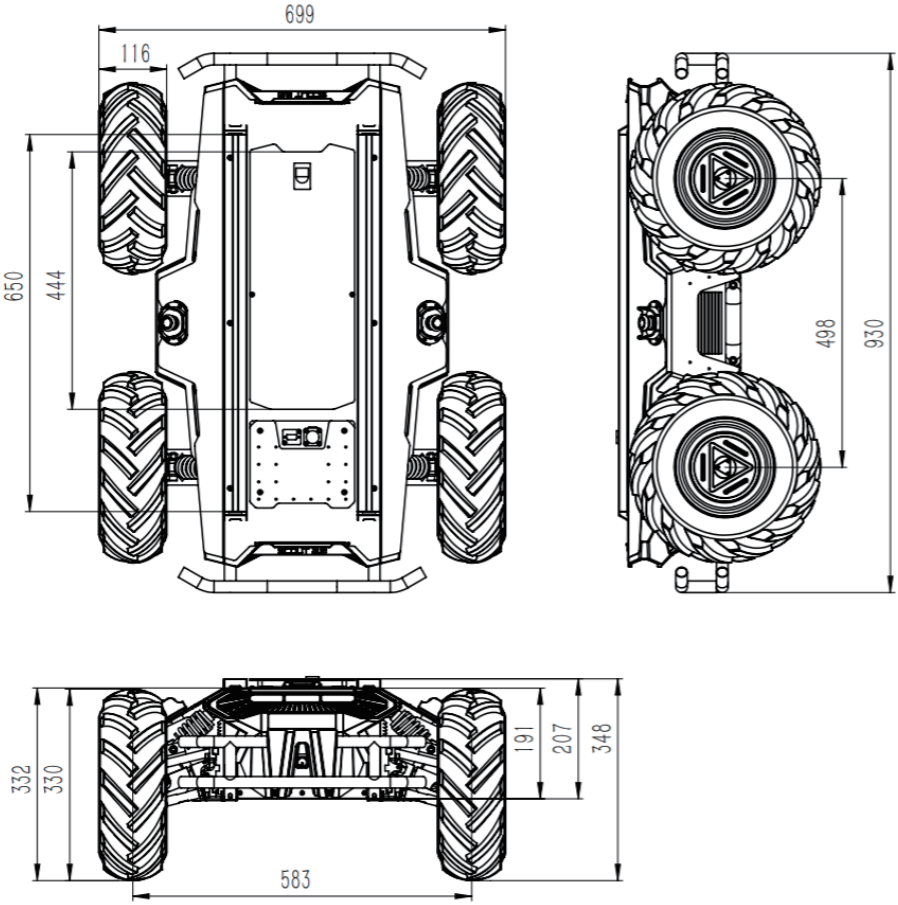
Q:SCOUT2.0在运行中发出“滴-滴-滴…”的声音，该如何处理？
A:若SCOUT2.0发出连续的“滴-滴-滴…”表明电池已经处于警报电压状态，请及时充电;出现相关声音以后，亦可能是内部出现相关错误了，可以通过CAN总线检查相关的错误代码，或与相关技术人员沟通。

Q:SCOUT2.0 在运行过程中出现轮胎磨损情况是属于正常现象嘛？
A:SCOUT2.0在运行过程中出现轮胎磨损属于正常现象。由于SCOUT2.0采用四轮差速转向的设计，在车体旋转的过程中会出现滑动摩擦和滚动摩擦并存的情况，如果地面不光滑，表面粗糙，这个时候对轮胎表面存在磨损情况。为了减少磨损或者减缓磨损的过程，可以采用小角度转弯的形式，尽量减少原地旋转的形式。

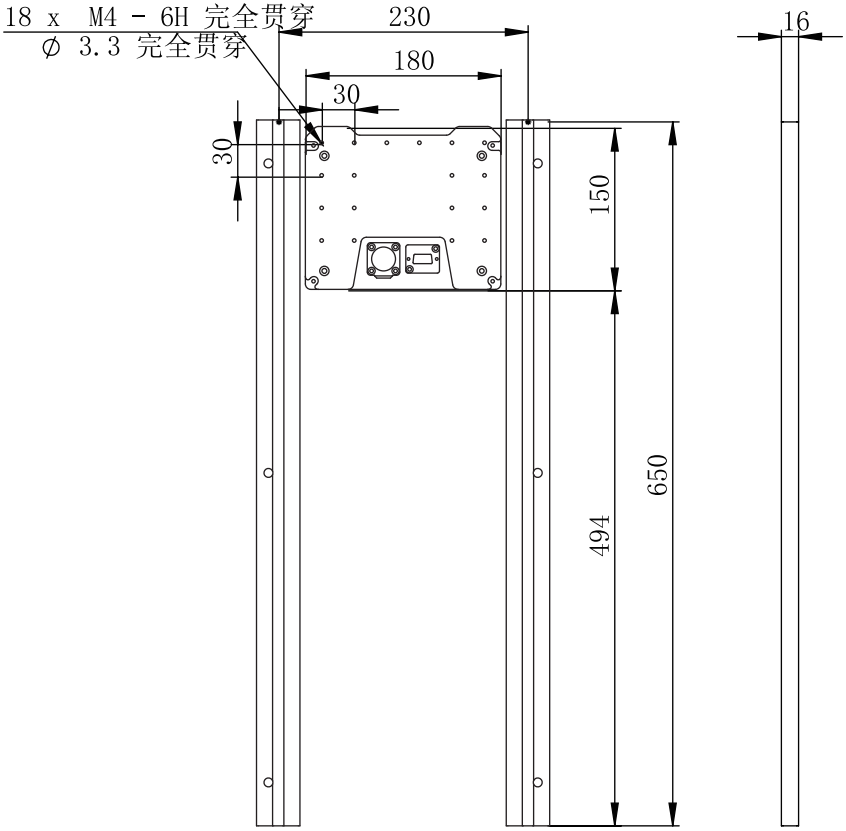
Q:通过CAN总线进行相关通讯时，底盘反馈指令正常，下发控制小车无响应？
A: SCOUT2.0的内部有通讯保护机制，底盘在处理来自外部的CAN控制指令时存在超时保护机制，假设小车收到一帧通讯协议以后，小车超过500MS未收到下一帧控制指令，小车会进入通讯保护，速度为0，所以来自上位机的指令必须时周期性的发布。

6 产品尺寸 Product Dimensions

6.1 产品外形尺寸说明图



6.2 顶部扩张支架尺寸说明图



AGILE·X
松灵机器人（东莞）有限公司
WWW.AGILEX.AI
TEL:+86-0769-22892150
MOBILE：+86-19925374409