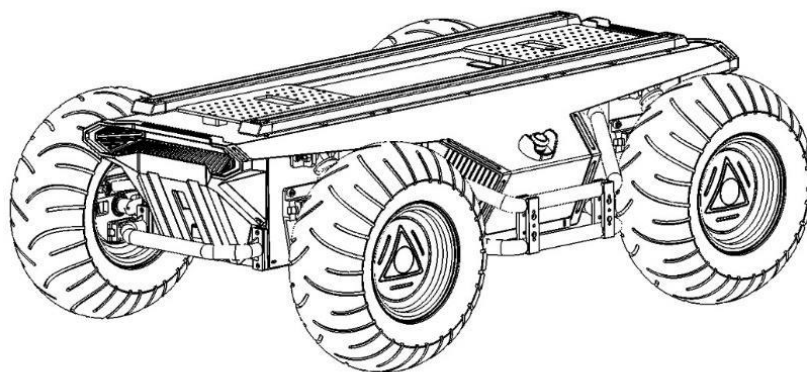




松灵机器人产品 SCOUT 用户手册

AgileX Robotics Team

Version 1.2 Release



版本信息说明

版本	变更内容	责任人	时间
v1.2.3	1、变更指令控制周期和反馈指令周期 2、反馈帧协议增加错误类型，协议向下兼容 3、修改最大速度为 1.5m/s 4、修改已知错误	谢志强	2019/04/2
V1.2.4	1、修改已知错误 2、反馈帧协议增加错误类型，协议向下兼容	谢志强	2019/4/4
V1.2.5	1、修改最大旋转角速度设置为 0.7853rad/s 2、增加遥控器控制模式下过流、过温保护功能	谢志强	2019/4/28
V1.2.6	1、修改示例数据 2、增加支持遥控器支持类型	谢志强	2019/4/29
V1.2.7	1、[新特性]增加电机数据反馈 2、[新特性]增加灯光控制接口和灯光状态反馈	谢志强	2019/5/8
V1.2.8	1、[新特性]新增遥控器灯光控制说明	谢志强	2019/5/10
V1.2.9	1、[修改]修改产品列表，增加内容 2、[新特性]新增串口通讯协议帧	谢志强	2019/5/16
V1.2.10	1、[修改]修改 FS 遥控器附图说明错误 2、[修改]修改灯光控制帧的部分表述错误 3、[新特性]电机信息反馈 ID 示意图	谢志强	2019/6/12
V1.2.11	1、[修改]修改描述错误 2、[新特性]增加在 ROS 环境下的开发教程	谢志强	2019/7/9
V1.2.12	1、[修改]修改最大旋转角速度为 0.5235rad/s 2、[新特性]增加串口控制协议	谢志强	2019/7/26
V1.2.13	1、[新增]使用注意事想 2、[新增]常见问题的处理与解决	谢志强	2019/7/30

目录

1 SCOUT 简介 Introduction	4
1.1 产品列表.....	4
1.2 性能参数.....	4
1.3 开发所需.....	5
2 基本介绍 The Basics.....	6
2.1 状态指示.....	7
2.2 电气接口说明	8
2.2.1 顶部电气接口说明	8
2.2.2 尾部电气接口说明	9
2.3 遥控说明.....	10
2.3.1 DJI 遥控控制说明	10
2.3.2 FS_i6_S 遥控说明.....	11
2.4 控制指令与运动说明.....	11
2.5 灯光控制说明	12
3 使用与开发 Getting Started.....	14
3.1 使用与操作.....	14
3.2 充电.....	15
3.3 开发.....	15
3.3.1 CAN 接口协议	15
3.3.2 CAN 线的连接	24
3.3.3 CAN 指令控制的实现	24
3.4 串口通信协议.....	24
3.4.1 串口协议介绍	24
3.4.2 串口协议内容	25
3.4.3 串口的连接	31
3.5 固件升级.....	31

3.6 SCOUT SDK 使用示例	32
3.7 SCOUT ROS Package 使用示例	33
4 注意事项 Attention	35
4.1 电池注意事项	35
4.2 使用环境注意事项	35
4.3 电气外部扩展注意事项	36
4.4 机械负载注意事项	36
4.5 其他注意事项	36
4.6 安全注意事项	36
5 常见问题与解决 Q&A	37
6 产品尺寸 Product Dimensions	38
6.1 产品外形尺寸说明图	38
6.2 顶部扩张支架尺寸说明图	39

1 SCOUT 简介 Introduction

SCOUT 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具有广泛的应用领域。立体相机、激光雷达、GPS、IMUS、机械手等设备可选择加装至 SCOUT 作为扩展应用。SCOUT 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
SCOUT 机器人本体	X 1
钥匙锁	X 1
电池充电器(AC 220V)	X 1
航空插头公头 (4Pin)	X 2
USB 转 RS232	X 1
遥控器(选配)	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)	1080 X 726 X 336
	轴距 (mm)	648
	前 / 后轮距 (mm)	576 / 576
	车体重量 (Kg)	45~50
	电池类型	锂电池 24V 30aH
	电机	直流无刷 4 X 200W
	减速箱	1: 32
	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
性能参数指针	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

SCOUT 出厂时可选配 DJI 遥控器，用户可以通过遥控器控制 SCOUT 移动机器人底盘，完成移动和旋转操作；SCOUT 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 SCOUT 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 SCOUT 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。

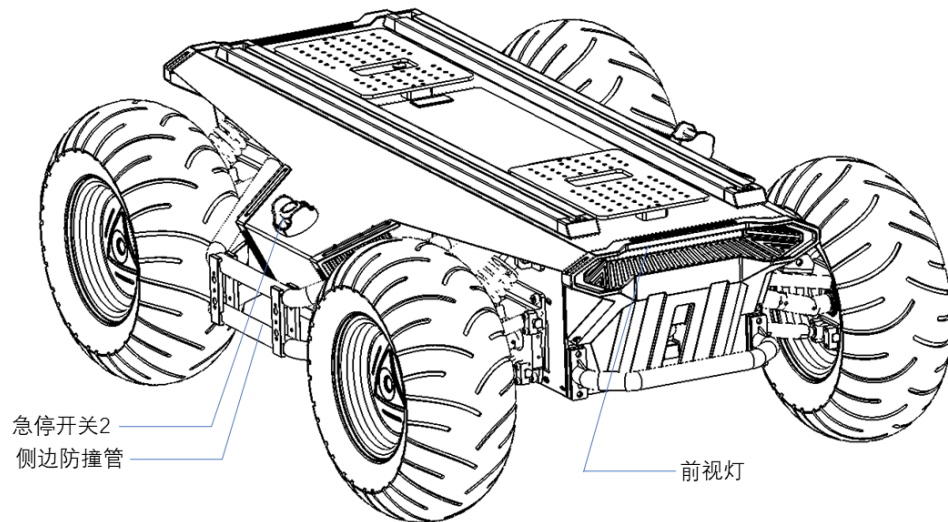


图 2.1 前部概览视图

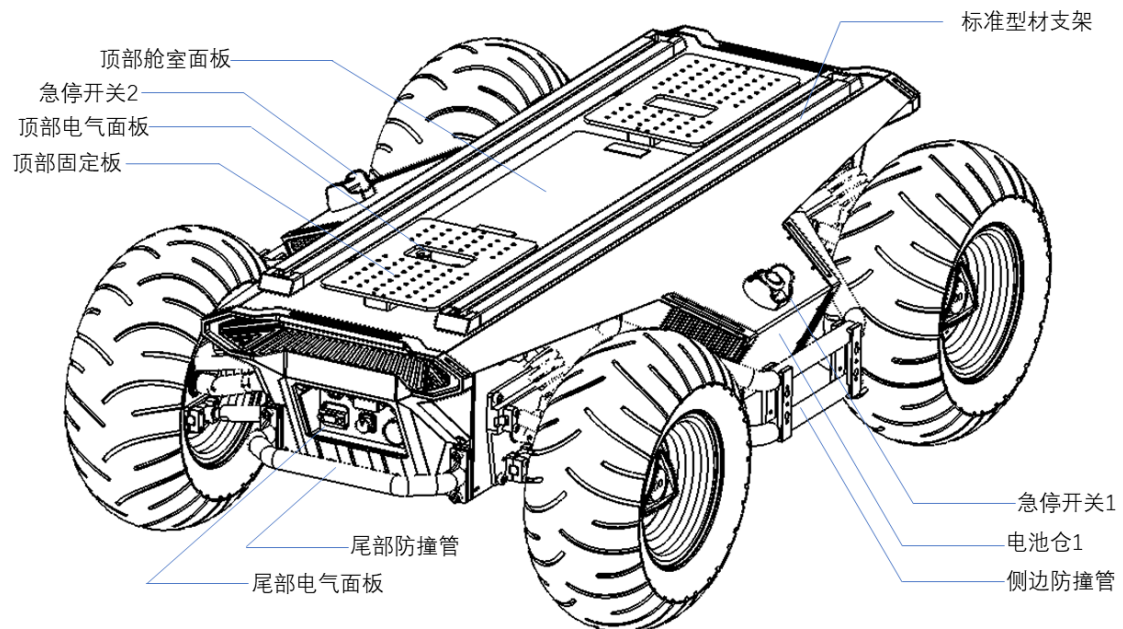


图 2.2 尾部概览视图

SCOUT 整体上采用了模块化和智能化的设计思想，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 SCOUT 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。

车体前后均安装有灯光，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。

在汽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 SCOUT 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”刺激的声音进行提示。当检测到电池电压低于 22V 时，SCOUT 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

SCOUT 提供了两组 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。

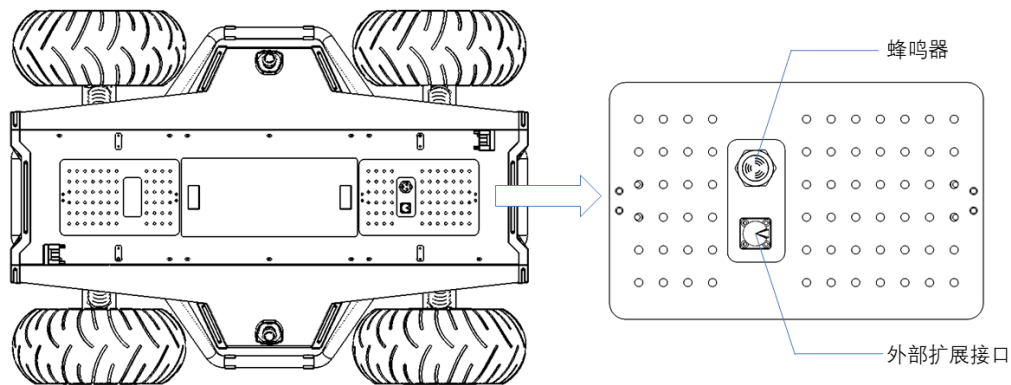
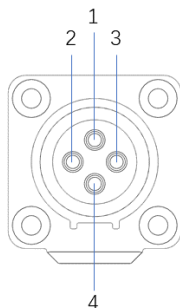


图 2.3 SCOUT 顶部电气界面示意图

SCOUT 顶部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以及一组 CAN 通讯接口。便于使用者可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。

这里需要注意的是，这里的扩展电源受内部控制，当电池电压低于安全电压会主动切断供电，所以客户需要注意，在达到临界电压前 SCOUT 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		GND	电源负
3	CAN	CAN_H	CAN总线高
4		CAN_L	CAN总线低

图 2.4 顶部航空扩展接口引脚定义图

2.2.2 尾部电气接口说明

尾部的扩展接口如图 2.5 所示，其中 Q1 为钥匙开关，是电气部分总开关；Q2 为充电界面；Q3 驱动系统供电开关；Q4 为 DB9 串口；Q5 为 CAN 和 24V 电源扩展接口；Q6 为电源显示交互。

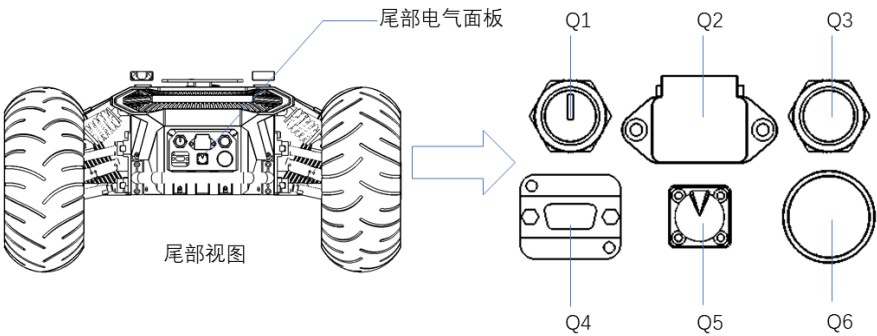


图 2.5 尾部视图

关于 Q4 的其具体引脚定义如图 2.6 所示。

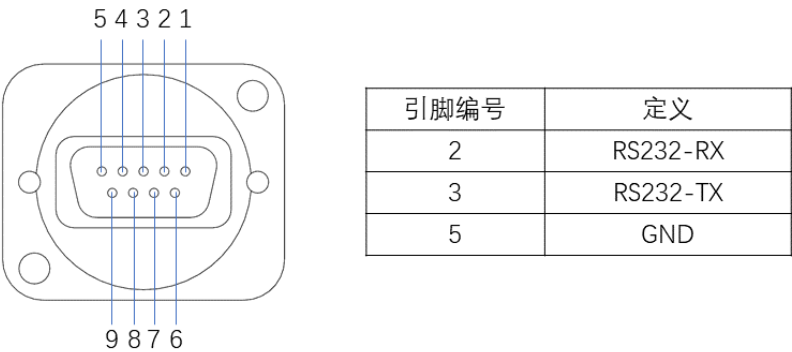


图 2.6 Q4 引脚说明图

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

2.3.1 DJI 遥控控制说明

DJI DT7 遥控器为 SCOUT 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 SCOUT 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 2.8。

S1 定义为功能选择按钮，S2 未使用；C1 为油门按钮，控制 SCOUT 前进和后退；C2 控制旋转。



图 2.8 遥控器示意图

2.3.2 FS_i6_S 遥控说明

富斯遥控器为 SCOUT 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 SCOUT 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 SCOUT 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。

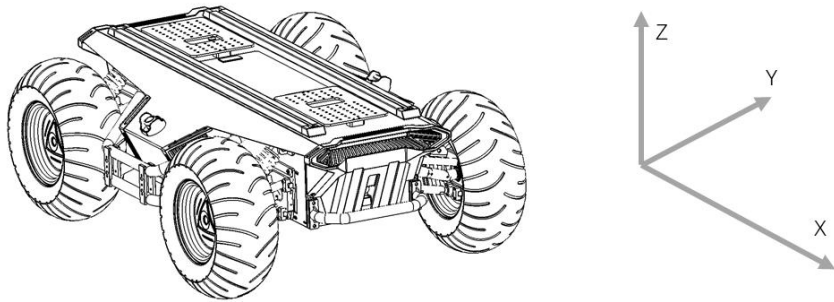


图 3.0 车身参考坐标系示意图

正如 3.0 所展示的，SCOUT 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 轴正方向往 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。

2.5 灯光控制说明

SCOUT 前后均配置了灯光，为了方便客户，SCOUT 的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：**

- 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，尾灯会关闭，前灯会打开；

- 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。

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。

3 使用与开发 Getting Started

本部分主要介绍 SCOUT 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

- 旋转钥匙开关(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

- 旋转钥匙开关，即可切断电源；

● 急 停

- 按下 SCOUT 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 SCOUT 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 SCOUT 平台运动。

3.2 充电

SCOUT 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 SCOUT 底盘处于停机断电状态。充电前请确认尾电气控制台中 Q1(钥匙开关)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，将充电器中开关打开，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；充电时间计算 $30\text{ aH} \div 10\text{A} = 3\text{h}$

3.3 开发

SCOUT 产品针对用户的开发提供了 CAN 和 RS232（**当前版本未开放**）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

SCOUT 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；SCOUT 会实时反馈当前的运动状态信息以及 SCOUT 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 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 (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 ^[1]	
byte [5]	故障信息低八位*			
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℃
	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℃
	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]	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	
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

SCOUT 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。

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 0.07853rad/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：

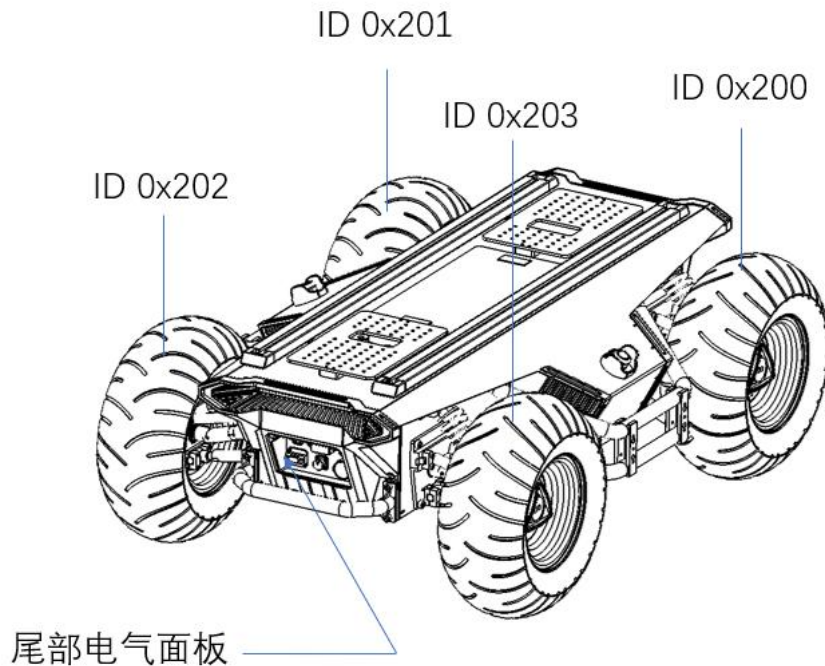


图 3.0 电机反馈 ID 示意图

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]	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℃)	
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	

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]	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℃)	
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	

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]	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℃)	
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	

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]	4 号驱动器电流高八位	unsigned int16	实际电流 X 10 (精确到 0.1A)	
byte [1]	4 号驱动器电流低八位			
byte [2]	4 号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)	
byte [3]	4 号驱动器转速低八位			
byte [4]	4 号驱动器温度	signed int8	实际温度 (精确到 1℃)	
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	

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 ^[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	
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[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	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]	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

$$\text{checksum} = (\text{ID_H} + \text{ID_L} + \text{data_length} + \text{can_msg.data}[0] + \text{can_msg.data}[1] + \text{can_msg.data}[2] + \text{can_msg.data}[3] + \text{can_msg.data}[4] + \dots + \text{can_msg.data}[n]) \& 0xFF;$$

- 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

SCOUT 随车发货提供了两个航空插头公头如图 3.2，客户需要自己焊接将线引出。线的定义可参考表 2.2。



图 3.2 航空插头公头示意图

注：当前 SCOUT 版本对外扩展接口仅顶部接口开放。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 SCOUT 移动机器人底盘，打开 DJI 遥控器，然后将控制模式切换至指令控制，即将 DJI 遥控器 S1 模式选择拨至最上方，此时 SCOUT 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

```
/**
 * @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;
}
```

图 3.3 串口校验算法代码示例

和；帧 ID 位 0~255 循环计数，没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 (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 × 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 ^[1]
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℃
	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]	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℃
	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。

■ 运动指控反馈指令

指令名称	运动控制反馈指令		
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]	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

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	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℃)
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℃)
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℃)
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]	4 号驱动器电流高八位	unsigned int16	实际电流 X 10 (精确到 0.1A)
byte [1]	4 号驱动器电流低八位		
byte [2]	4 号驱动器转速高八位	signed int16	电机输出轴实际速度 (RPM)
byte [3]	4 号驱动器转速低八位		
byte [4]	4 号驱动器温度	signed int8	实际温度 (精确到 1℃)
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 ^[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

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

● 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 SCOUT 所使用的固件版本进行升级，给客户带来更加完善的体验，SCOUT 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
- SCOUT 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 SCOUT 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;
- SCOUT 底盘上电, 立即点击开始连接 (SCOUT 底盘会在上电前 6 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。

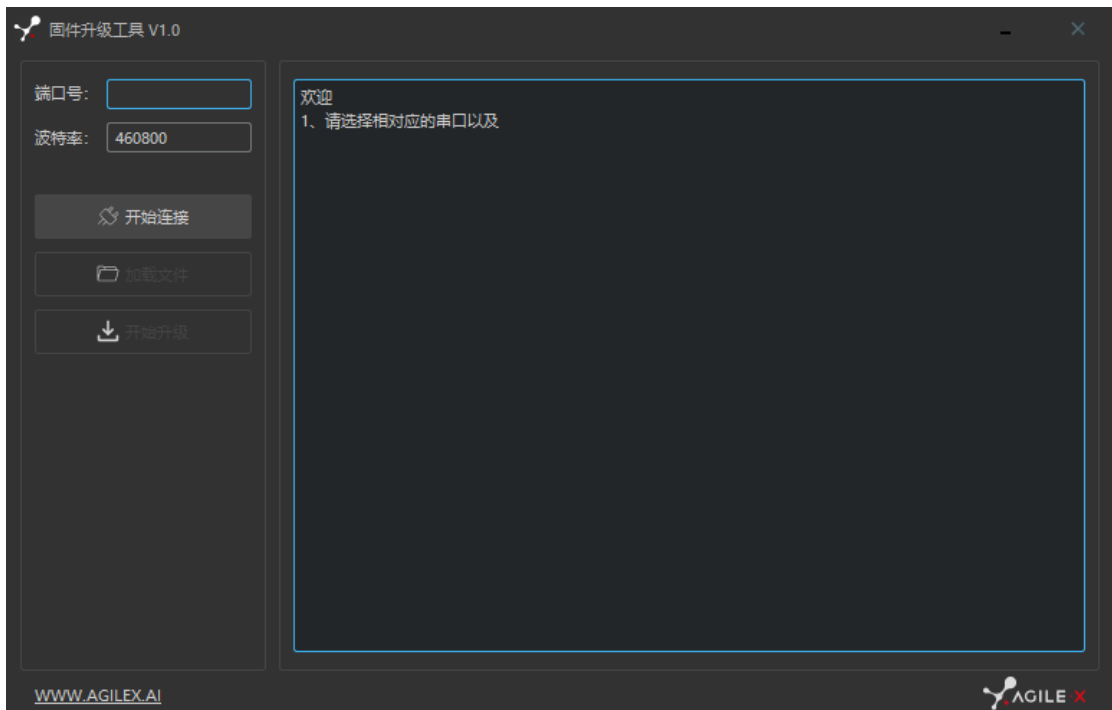


图 3.3 固件升级客户端界面

3.6 SCOUT SDK 使用示例

为了使客户更加便捷的进行机器人相关方面的开发, 我们针对 SCOUT 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 SCOUT 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 CAN, 基于 RS232 的还在适配中。

基于此, 我们在 Nvidia Jetson TX2 进行了相关测试。

3.7 SCOUT 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 移动机器人底盘 X1
 - AGILEX SCOUT 配套遥控器 FS-i6s X1
 - AGILEX SCOUT 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 SCOUT 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 底盘的数据了


```
$ candump can0
```

参考来源: [1] https://github.com/westonrobot/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/westonrobot/scout_ros.git  
$ cd ..  
$ catkin_make
```

- 参考来源: https://github.com/westonrobot/scout_ros

硬件连接与准备

- 将 SCOUT 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 SCOUT 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 SCOUT 的应该注意的一些事项。

4.1 电池注意事项

- SCOUT 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 SCOUT 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 SCOUT 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

- SCOUT 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；
- SCOUT 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；
- SCOUT 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），SCOUT 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℃；
- 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 6.25A,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换,所以如果外部扩展设别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

- SCOUT 前后为塑料件,请勿直接捶打,否则容易损坏;
- 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。

4.6 安全注意事项

- 使用过程有疑问,请按照相关说明手册进行或者咨询相关技术人员;
- 使用设备过程前,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: SCOUT 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: SCOUT 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，SCOUT 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，可以接受到底盘的反馈帧，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，控制模式中是否置为指令控制 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: SCOUT 在运行中发出“滴-滴-滴…”的声音，改如何处理？

A: 若 SCOUT 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: SCOUT 在运行过程中出现轮胎磨损情况是属于正常现象嘛？

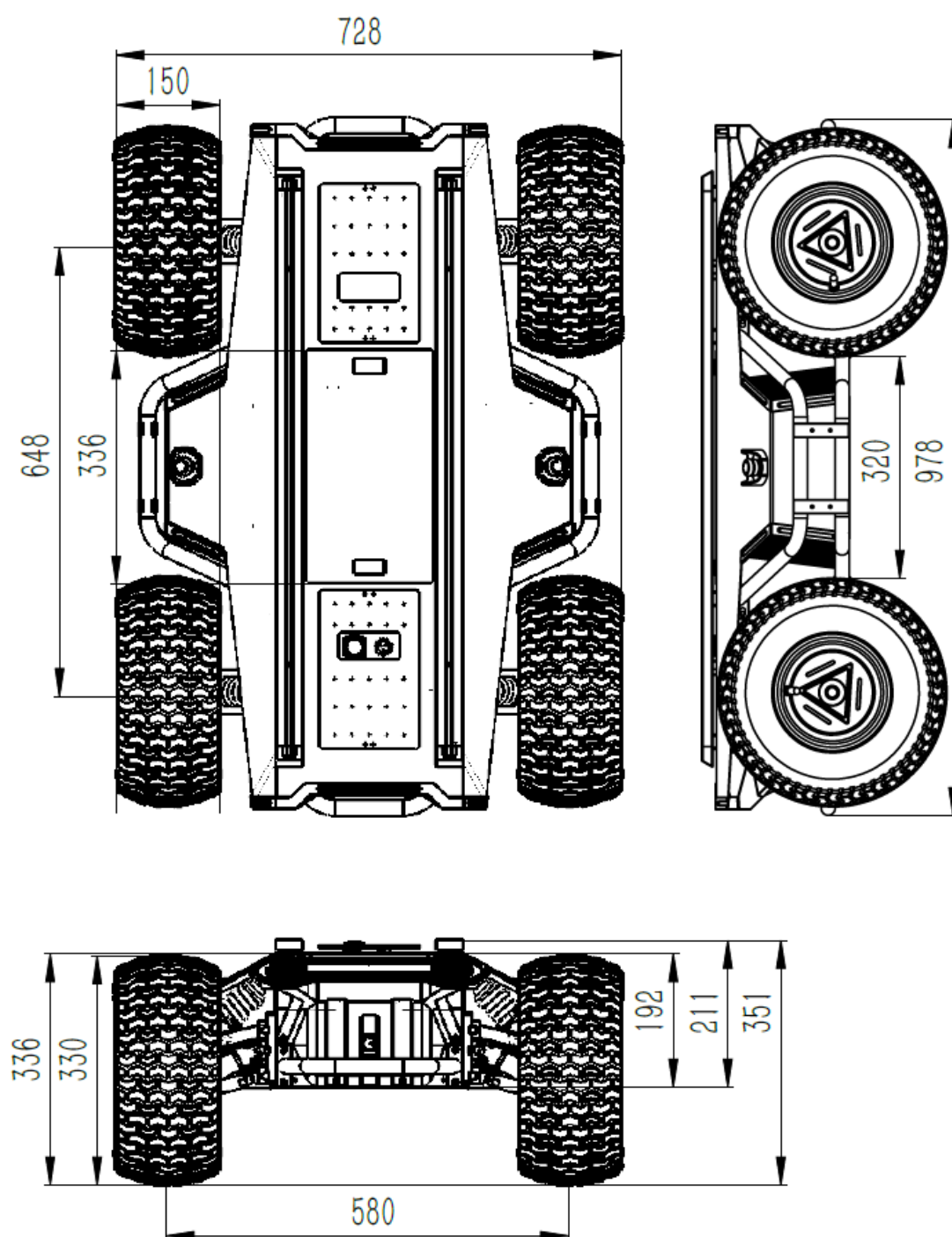
A: SCOUT 在运行过程中出现轮胎磨损属于正常现象。由于 SCOUT 采用的是四轮差速转向的设计，在车体旋转的过程中会出现滑动摩擦和滚动摩擦并存的情况，如果地面不光滑，表面粗糙，这个时候对轮胎表面存在磨损情况。为了减少磨损或者减缓磨损的过程，可以采用小角度转弯的形式，尽量减少原地旋转的形式。

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: SCOUT 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

