

TITAN 用户手册



TITAN

AgileX Robotics Team

用户手册 V.2.0.1 2023.08

编号	版本	日期	修改人	审核人	备注
1	V1.0.0	2022/10/15	何士玉		初稿
2	V2.0.0	2023/3/23	李圣望、何士玉		更新协议内容
3	V2.0.1	2023/09/02	谢瑞亲		

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

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。

TITAN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

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。

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0°C~40°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IP54。

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。

使用注意事项

保证操作时周围区域相对空旷。
在视距内遥控控制。
TITAN最大的载重为 300KG，在使用时，确保有效载荷不超过 300KG。
TITAN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。
尾部扩展电源电流不超过 15A,总功率不超过 720W

电池注意事项

TITAN 产品出厂时电池并不是满电状态的，具体电池电压和电量可以通过 TITAN 底盘尾部电压显示表显示或者通过遥控器上 vol 和 batt 显示

请不要在电池使用殆尽以后再进行充电，在 TITAN 遥控低电量低于15%或者尾部电压显示低于45V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~40°C，电池在不使用的情况下存放，必须是1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

使用环境注意事项

TITAN 工作温度为-10°C~40°C，请勿在温度低于 -10°C、高于40°C环境中使用

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

请不要在加热器或者大型卷线电阻等发热体周围使用

TITAN 防水防尘等级为IP54，请注意使用环境，定期检查清锈

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

定期检查和维护履带张紧轮

安全注意事项

使用过程中有疑问，请按照相关说明手册进行或者咨询相关技术人员

使用设备操作前，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

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。

概述

1. TITAN简介

TITAN是一款可编程前后双转向型UGV (UNMANNED GROUND VEHICLE) ，它是一款采用模块化设计的底盘。相较于前AKM底盘，TITAN不仅大大减小了转弯半径，而且更加稳定和安全。具有更广泛的应用。TITAN底盘，适用于多种复杂地形。同时可以搭载立体相机、激光雷达、GPS、IMU、机械手等设备，被运用到无人巡检、安防、科研、勘探物流等领域。

1.1性能参数

参数类型	项目	指标
机械参数	长×宽×高 (mm)	1640×980×1440
	轮距 (mm)	860
	轴距 (mm)	845
	额定功率 (w)	1000
	额定扭矩 (N.M)	556

性能参数	最大速度 (km/h)	10.8
	运动模式	前后阿克曼
	最大越障 (mm)	100 (垂直障碍满载)
	最大爬坡 (°)	10
	自重 (kg)	280(两个电池)
	载重 (kg)	300
	续航时间	> 5h(1.4km/h测试)
	充电效率	1C
	电池类型	磷酸铁锂
	单电池容量 (ah)	24 (最多可支持2块电池)
	额定电压 (v)	48
功能	运用场景	工程勘测
		能源巡检
		矿区运输
		智慧安防
		物流配送
		农业运输采集

1.2开发所需

TITAN 出厂时可选配FS遥控器，用户可以通过遥控器控制底盘，完成模式切换、移动和转向控制。TITAN 配备了标准CAN通信接口，用户可以通过CAN接口进行二次开发。

2 基本介绍

本部分内容将会对TITAN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TITAN 底盘有一个基本的认识。如下图2.1所示，为整个移动机器人底盘的概览视图。

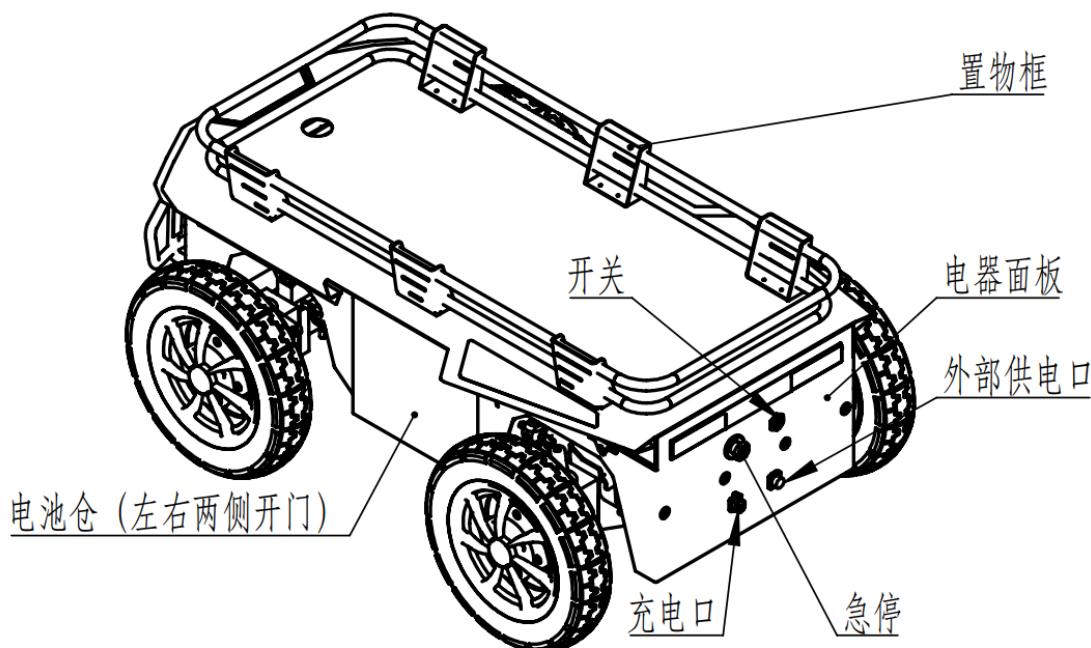


图2.1尾部概览视图

TITAN整体上采用了模块化和智能化的设计理念，在动力模块上采用了充气轮胎设计，再加上动力强劲的永磁同步电机，使得TITAN机器人底盘具有很强的续航以及载重能力，搭载额定功率1000W的电机使得TITAN可在不同的地面上灵活运动。在TITAN 的尾部配置了开放的电气接口和通讯接口，方便用户进行二次开发，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中使用。在TITAN 顶部安装有标准铝型材扩展支架，方便用户搭载外部设备扩展使用。

2.1状态指示

用户可以通过TITAN底盘反馈的CAN报文来确定车体的状态。具体参考表2.1。

状态	描述
当前电压	当前电池电压可通过遥控器显示界面查看
上电显示	灯光亮起

低电压报警	当电池BMS通讯反馈SOC低于15%，车体后灯会闪烁进行提示。当检测到电池电量低于10%时，四轮四转底盘为了防止电池损坏，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
详细状态信息	通过CAN报文查看

表格 2.1 车体状态说明表

2.2.电气接口说明

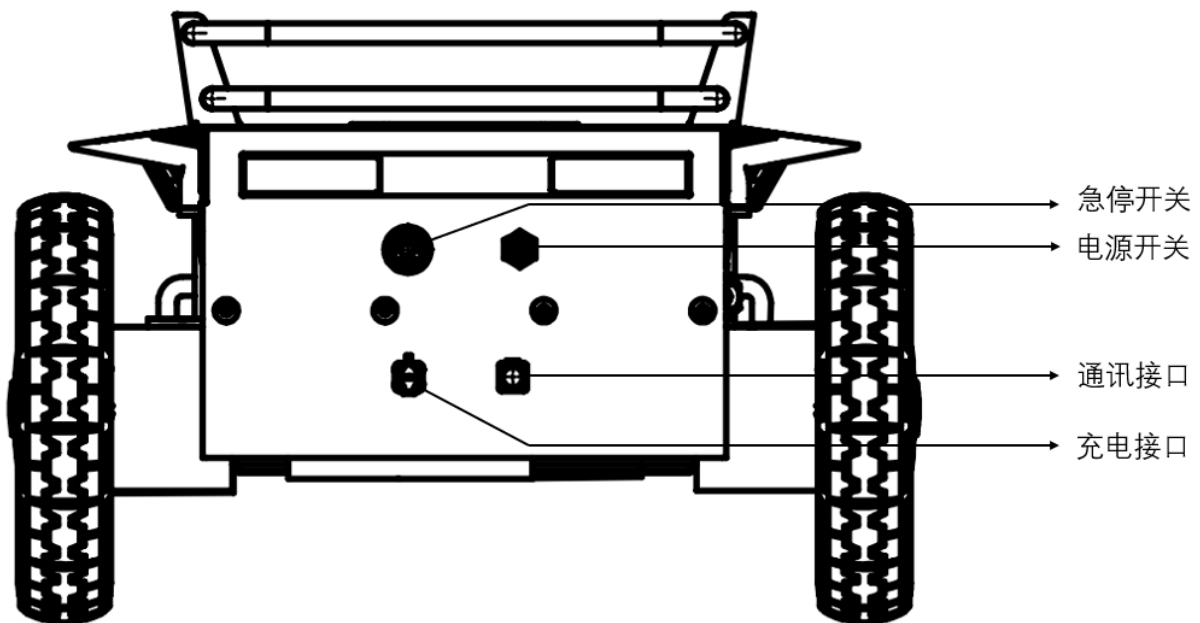
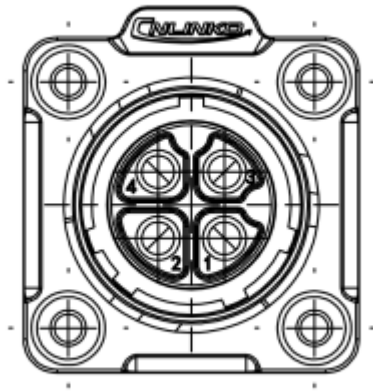


图2.2尾部视图

TITAN底盘尾部配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。便于使用者给扩展设备提供电源（负载电流不能超过15A，电压范围46~50V），以及通讯使用。其具体引脚定义如下图所示。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，所以用户需要注意，在达到临界电压前TITAN 底盘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46~50V， 负载电流不能超过15A。
2	电源	GND	电源负
3	CAN	CAN_H	CAN总线高
4	CAN	CAN_L	CAN总线低

图2.3尾部航空接口引脚说明图

2.3.遥控说明



图2.4遥控器示意图

如上图，按键的功能定义为：SWB为控制模式选择拨杆，拨至最上方为指令控制模式，拨至中间为遥控控制模式；SWA为驻车开关拨杆，拨至最下方解除驻车模式，拨至上方是驻车模式（当驻车模式解除之后才能正常进行遥控）；SWC为灯光控制开关，前灯：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；后灯：拨到最下方为关闭灯光（需SWB先进入遥控模式）；

正常启动TITAN移动机器人底盘后，启动遥控器，将SWB切换为遥控控制模式，即可通过遥控器控制TITAN平台运动。

SWD为越野模式切换拨杆，拨至上方为常规模式，拨至下方为越野模式；越野模式下底盘驱动电机采用前后独立速度闭环模式，避免在有轮子悬空打滑时由于电子差速而导致动力丧失。越野模式的切换需要在底盘完全静止之后方可切换成功。为保证底盘运动控制的平顺性，常规路况下请关闭越野模式。

SWA拨到上 + SWB拨到中 + SWC拨到下 + SWD拨到下 + 左滚轮滑到最上 = 进入前转向校准模式，此时可通过右摇杆调节前轮角度，按下KEY1设定当前位置为零点，重启生效；

SWA拨到上 + SWB拨到中 + SWC拨到下 + SWD拨到下 + 左滚轮滑到最下 = 进入后转向校准模式，此时可通过右摇杆调节后轮角度，按下KEY1设定当前位置为零点，重启生效；

注意：前后转向需要分开设置，即设置完前轮后重启一次方可再设置后轮

任何情况下按下KEY1 = 强制清除底盘所有错误 注意！！仅在确保安全的特殊情况下可使用

3 使用和开发

本部分主要介绍TITAN 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TITAN进行二次开发。

3.1 使用与操作

检查

• 检查TITAN 状态

检查TITAN 是否有明显异常；如有，请联系售后支持；

初次使用时确认尾部电气面板中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TITAN 处于断电状态。

• 启动和关机

按下电源开关，即可启动底盘

尾部标有“STOP”标志的开关为急停开关，顺时针旋转可解除急停，按下为急停模式。

充电

检查电池电压，正常电压范围为46~50V,如有“滴-滴滴...”连续蜂鸣器声音，表示电池电压过低，请及时充电。

本产品默认随车配备一个 20A的充电器，将充电器的插头插入底盘充电插口，将充电器连接电源，将充电器上开关打开，即可进入充电状态。

CAN线的连接

四轮四转底盘随车发货提供了1个航空插头公头，线的定义可参考下图：



- 1红色 : VCC (电池正极)
- 2黑色 : GND (电池负极)
- 3黄色 : CAN_H
- 4蓝色 : CAN_L

图3.1 航空插头示意图

CAN指令控制的实现

正常启动TITAN 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TITAN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。（默认在启动底盘不启动遥控器的时候底盘处于待机模式，此时可以发送can数据切换底盘的控制模式。当遥控器打开的时候，遥控器拥有第一控制权，遥控器处于哪种模式，底盘就处于哪种模式。）

3.2 CAN接口协议

本产品中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总线接口可以进行控制模型切换和控制底盘移动的线速度以及转向角；底盘会实时反馈当前的运动状态信息（包括经过整合处理的整机运动信息和各个轮子的详细运动信息）以及系统状态信息（包含自诊断错误码）。

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
			0x02 系统异常
byte [1]	模式控制	unsigned int8	0x00 待机模式
			0x01 CAN指令控制模式
			0x03 遥控模式
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]	故障信息最高位	unsigned int32	详见故障信息说明表
byte [5]	故障信息次高位		
byte [6]	故障信息次低位		
byte [7]	故障信息最低位		

故障信息说明表

故障信息说明		
字节	位	含义
byte [4]		保留，默认0

byte [5]	bit [0]	前转向编码器状态 (0:无故障 1 : 故障)
	bit [1]	后转向编码器状态 (0:无故障 1 : 故障)
	bit [2]	保留, 默认0
	bit [3]	保留, 默认0
	bit [4]	保留, 默认0
	bit [5]	触边防撞, (0:正常 1 : 触发停机)
	bit [6]	保留, 默认0
	bit [7]	保留, 默认0
byte [6]	bit [0]	驱动器状态错误 (0 : 无故障1 : 故障)
	bit [1]	上层通讯连接状态 (0 : 无故障 1 : 故障)
	bit [2]	保留, 默认0
	bit [3]	保留, 默认0
	bit [4]	保留, 默认0

	bit [5]	保留，默认0
	bit [6]	过温保护（0:无故障 1：故障）
	bit [7]	过流保护（0:无故障 1：故障）
byte [7]	bit [0]	电池欠压故障（0:无故障 1：故障）
	bit [1]	保留，默认0
	bit [2]	遥控器失联保护（0：无故障 1：故障）
	bit [3]	1号电机驱动器通讯故障（0:无故障 1：故障）
	bit [4]	2号电机驱动器通讯故障（0:无故障 1：故障）
	bit [5]	3号电机驱动器通讯故障（0:无故障 1：故障）
	bit [6]	4号电机驱动器通讯故障（0:无故障 1：故障）
	bit [7]	保留，默认0

运动控制反馈帧指令包含了当前车体的运动线速度、转向角度反馈

协议具体内容如下

指令名称		运动控制反馈指令		
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] byte [1]	移动速度高八位 移动速度低八位	signed int16	实际速度X 1000 (单位0.001m/s)
byte [2]	保留		0X00
byte [3]	保留	-	0X00
byte [4] byte [5]	保留	-	0X00
byte [6] byte [7]	转角高八位 转角低八位	signed int16	实际内转角X 1000 (单位0.001rad)

运动控制帧包含了线速度控制指令、转角控制指令，其具体协议内容如下：

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] byte [1]	线速度高八位 线速度低八位	signed int16	车体行进速度，单位mm/s(有效值 + -3000，当转向角度 0.087rad时有效值+ -1500)	
byte [2]	保留	-	0X00	
byte [3]	保留	-	0X00	
byte [4] byte [5]	保留	-	0X00	

byte [6]	转角高八位	signed int16	转向内转角，单位：0.001rad (有效值：+ -367，左转方向为正值)
byte [7]	转角低八位		

PS：在CAN指令模式下，需要保证0X111指令帧以小于500MS的周期（建议周期20MS）发送，否则TITAN会判定为控制信号心跳丢失而进入报错（0X211反馈上层通讯失联），系统报错后会进入待机模式，若此时0X111控制帧恢复正常发送周期，上层通讯失联错误可自动清除，同时控制模式恢复为CAN控制模式。

如图3.2.1，当TITAN在此状态时，反馈的转角为 $(\alpha_3 + \alpha_4) / 2$ ，负值为左转方向，正值为右转方向；反馈的速度为四轮速度平均值（即底盘运动线速度），负值为倒车，正值为前进。若需要查看各个轮子的详细转角和速度信息，参看0X271和0X281反馈帧。

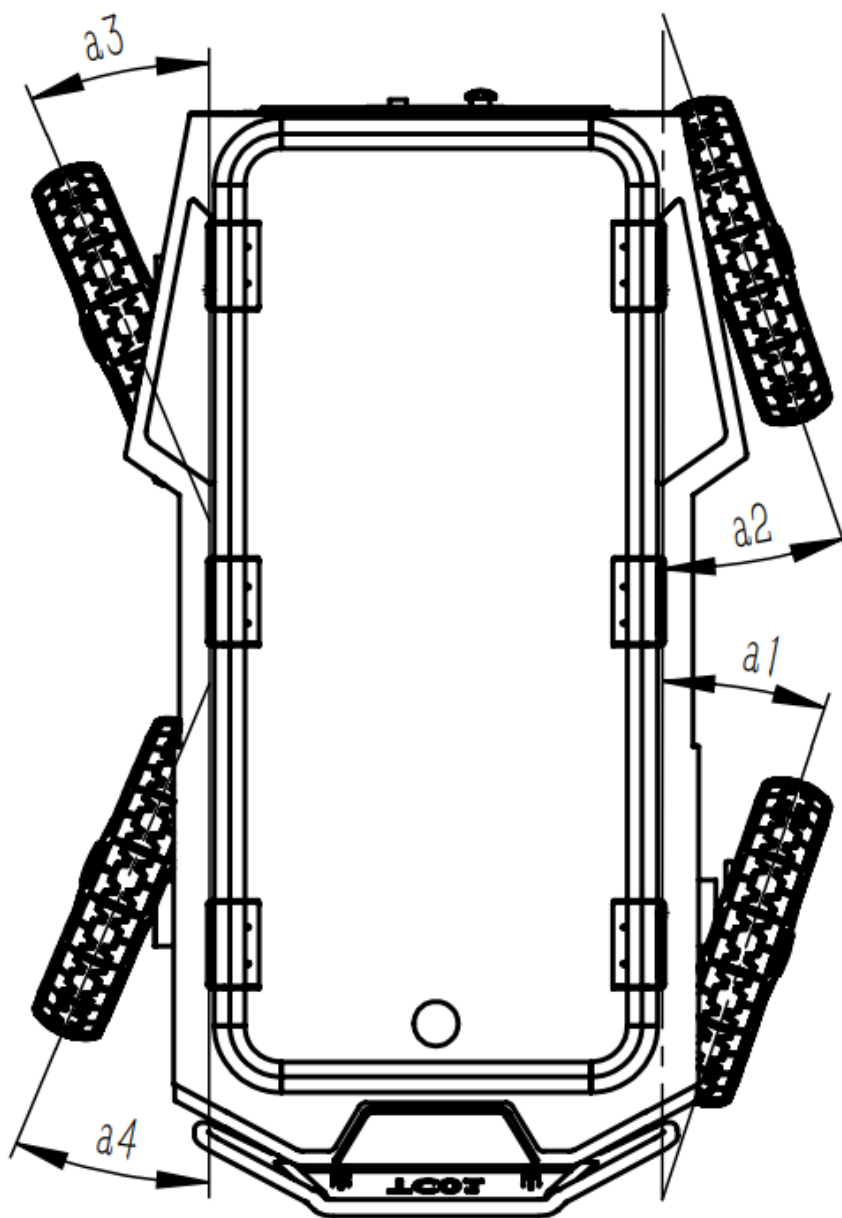


图3.2.1TITAN结构

模式设定帧用于设定终端的控制接口，其具体协议内容如下。

指令名称		控制模式设定指令		
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	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 待机模式 0x01 CAN指令模式 上电默认进入待机模式

控制模式说明：底盘在开机上电，遥控器未连接的情况下，控制模式默认是待机模式，此时底盘只接收控制模式指令，其他指令不做响应，要使用CAN进行控制需要先切换到CAN指令模式。若打开遥控器，遥控器具有最高权限，可以屏蔽指令的控制，切换控制模式。状态置位帧用于清除系统错误，其具体协议内容如下。

指令名称		状态设定指令		
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	0x441	无	无
数据长度	0x01			
位置	功能	数据类型	说明	
byte [0]	错误清除指令	unsigned int8	0x00 清除全部非严重故障 0x01~0x04分别对应清除1~4号电机驱动器通讯故障 0x09 清除电池欠压故障，并尝试恢复动力电源 0x0a 清除遥控信号丢失故障 0x0b~0x0c 分别对应清除前、后转向编码器故障 0x0f 清除过流故障 0x10 清除过温故障	

驻车控制指令用于控制驱动轮的电机抱闸，协议具体内容如下。

指令名称	驻车控制指令

发送节点	接收节点	ID	周期 (ms)	接受超时 (ms)
决策控制单元	底盘节点	0x131	无	无
数据长度	0x01			
位置	功能	数据类型	说明	
byte [0]	驻车指令	unsigned int8	0x00打开驻车(锁抱闸) 0x01关闭驻车(解锁抱闸) 抱闸需要解锁才能进行底盘的速度控制	

指令名称				
CAN升级总线静默模式设定指令				
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	0x422	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 退出CAN升级总线静默模式 0x01 进入CAN升级总线静默模式 CAN升级总线静默模式下底盘VCU 暂停往总线发送CAN数据以便于升级 固件，退出后恢复数据反馈	

如需要越障可进入越野模式，越野模式下底盘驱动电机采用前后独立速度闭环模式，避免在有轮子悬空打滑时由于电子差速而导致动力丧失。越野模式的切换需要在底盘完全静止之后方可切换成功。

指令名称				
越野模式设定指令				
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	0x423	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 底盘退出越野模式 0x01 底盘进入越野模式

示例数据，以下数据仅供测试使用：

byte [0]	byte [1]	byte [2]	byte [3]	byte [4]	byte [5]	byte [6]	byte [7]
0x00	0x96	0x00	0x00	0x00	0x00	0x00	0x00

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四轮的转角和转速，电机的电流信息、编码器以及温度信息。

具体协议内容如下：

电机转速电流位置信息反馈

指令名称		电机驱动器高速信息反馈帧		
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	0x251~0x254	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	电机转速高八位 电机转速低八位	signed int16	电机当前转速 单位RPM	
byte [2] byte [3]	电机电流高八位 电机电流低八位	signed int16	电机当前电流 单位0.1A	

byte [4]	位置最高位	signed int32	电机当前位置 单位：脉冲数
byte [5]	位置次高位		
byte [6]	位置次低位		
byte [7]	位置最低位		

电机温度电压及状态反馈

指令名称		电机驱动器低速信息反馈帧		
发送节点	接收节点	ID	周期 (ms)	接收超时 (ms)
线控底盘	策控制单元	0x261~0x264	100ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	驱动器电压高八位	unsigned int16	当前驱动器电压 单位0.1V	
byte [1]	驱动器电压低八位			
byte [2]	驱动器温度高八位	signed int16	单位1°C	
byte [3]	驱动器温度低八位			
byte [4]	电机温度	signed int8	单位 1°C	
byte [5]	驱动器状态	unsigned int8	详见表2	
byte [6]	保留	-	0X00	
byte [7]	保留	-	0X00	

表格 2 驱动器状态

字节	位	含义
----	---	----

byte[5]	bit[0]	电源电压是否过低 (0 : 正常 1 : 过低)
	bit[1]	电机是否过温 (0 : 正常 1 : 过温)
	bit[2]	驱动器是否过流 (0 : 正常 1 : 过流)
	bit[3]	驱动器是否过温 (0 : 正常 1 : 过温)
	bit[4]	传感器状态 (0 : 正常 1 : 异常)
	bit[5]	驱动器错误状态 (0 : 正常 1 : 错误)
	bit[6]	驱动器使能状态 (0 : 失能 1 : 使能)
	bit[7]	保留

转向零点设定和反馈指令用于校准零位，协议具体内容如下表：

转向零点设定指令

指令名称	转向零点设定指令			
发送节点	接收节点	ID	周期	接收超时 (ms)
决策控制单元	底盘节点	0x431	无	无
数据长度	0x01			
位置	功能	数据类型	说明	
byte 【0】	设定当前位置为零点	unsigned int8	设置当前位置为零点 固定值：0xAA	

转向零点设定反馈指令

指令名称	转向零点设定反馈指令			
发送节点	接收节点	ID	周期	接收超时 (ms)
底盘节点	决策控制单元	0x43A	无	无

数据长度	0x01		
位置	功能	数据类型	说明
byte [0]	设定当前位置为零点	unsigned int8	0xEE 设置当前位置为零点成功

转角反馈

指令名称		转角信息反馈帧		
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	0x271	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0] byte [1]	前转向内转角高八位 前转向内转角低八位	signed int16	当前转角 单位0.001rad	
byte [2] byte [3]	后转向内转角高八位 后转向内转角低八位	signed int16	当前转角 单位0.001rad	
byte [4] byte [5]	保留	-	0X00	
byte [6] byte [7]	保留	-	0X00	

转速反馈

由于底盘采用的是驱动桥，前轮转速实际为驱动轴转速，可以理解为前左和前右轮转速平均值；后轮同理

指令名称	转速信息反馈帧
------	---------

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		20ms	无
数据长度	0x08	0x281		
位置	功能	数据类型	说明	
byte [0] byte [1]	前轮转速高八位 前轮转速低八位	signed int16	当前转速 单位mm/s	
byte [2] byte [3]	后轮转速高八位 后轮转速低八位	signed int16	当前转速 单位mm/s	
byte [4] byte [5]	保留	-	0X00	
byte [6] byte [7]	保留	-	0X00	

里程计信息反馈帧如下

指令名称	里程计信息反馈			
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	20ms	无
数据长度	0x08			
字节	描述	数据类型	说明	
byte [0] byte [1] byte [2] byte [3]	前轮里程计最高位 前轮里程计次高位 前轮里程计次低位 前轮里程计最低位	signed int32	底盘前轮里程计反馈, 单位: mm	

byte [4]	后轮里程计最高位	signed int32	底盘后轮里程计反馈, 单位: mm
byte [5]	后轮里程计次高位		
byte [6]	后轮里程计次低位		
byte [7]	后轮里程计最低位		

遥控器信息反馈帧如下表

指令名称		遥控器信息反馈		
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	0x241	20ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	遥控SW反馈	unsigned int8	bit[0-1]: SWA : 2-上档 3-下档 bit[2-3]: SWB : 2-上档 1-中档 3-下档 bit[4-5]: SWC: 2-上档 1-中档 3-下档 bit[6-7]: SWD: 2-上档 3-下档	
byte [1]	右边拨杆左右	unsigned int8	值域: [-100,100]	
byte [2]	右边拨杆上下	unsigned int8	值域: [-100,100]	
byte [3]	左边拨杆上下	unsigned int8	值域: [-100,100]	
byte [4]	左边拨杆左右	unsigned int8	值域: [-100,100]	
byte [5]	左边旋钮VRA	unsigned int8	值域: [-100,100]	
byte [6]	保留	--	0x00	
byte [7]	计数校验	unsigned int8	0-255循环计数	

电池BMS数据反馈，其具体协议内容如下

指令名称		BMS数据反馈		
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	0x361	500ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	电池SOC	unsigned int8	范围 0~100	
byte [1]	电池SOH	unsigned int8	范围 0~100	
byte [2] byte [3]	电池电压值高八位 电池电压值低八位	unsigned int16	单位 : 0.01V	
byte [4] byte [5]	电池电流值高八位 电池电流值低八位	signed int16	单位 : 0.1A	
byte [6] byte [7]	电池温度高八位 电池温度低八位	signed int16	单位 : 0.1°C	

指令名称		BMS数据反馈		
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	0x362	500ms	无

数据长度	0x04			
位置	功能	数据类型	说明	
byte [0]	Alarm Status 1	unsigned int8	BIT1 : 过压 BIT2 : 欠压 BIT3 : 高温 BIT4 : 低温 BIT7 : 放电过流	
byte [1]	Alarm Status 2	unsigned int8	BIT0 : 充电过流	
byte [2]	Warning Status 1	unsigned int8	BIT1 : 过压 BIT2 : 欠压 BIT3 : 高温 BIT4 : 低温 BIT7 : 放电过流	
byte [3]	Warning Status 2	unsigned int8	BIT0 : 充电过流	

第一块电池BMS数据反馈

指令名称		BMS数据反馈		
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	0x363	500ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	电池SOC	unsigned int8	范围 0~100	
byte [1]	电池SOH	unsigned int8	范围 0~100	
byte [2] byte [3]	电池电压值高八位 电池电压值低八位	unsigned int16	单位 : 0.01V	

byte [4] byte [5]	电池电流值高八位 电池电流值低八位	signed int16	单位 : 0.1A
byte [6] byte [7]	电池温度高八位 电池温度低八位	signed int16	单位 : 0.1°C

指令名称		BMS数据反馈		
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	0x364	500ms	无
数据长度	0x04			
位置	功能	数据类型	说明	
byte [0]	Alarm Status 1	unsigned int8	BIT1 : 过压 BIT2 : 欠压 BIT3 : 高温 BIT4 : 低温 BIT7 : 放电过流	
byte [1]	Alarm Status 2	unsigned int8	BIT0 : 充电过流	
byte [2]	Warning Status 1	unsigned int8	BIT1 : 过压 BIT2 : 欠压 BIT3 : 高温 BIT4 : 低温 BIT7 : 放电过流	
byte [3]	Warning Status 2	unsigned int8	BIT0 : 充电过流	

第二块电池BMS数据反馈

指令名称		BMS数据反馈		
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	0x365	500ms	无
数据长度	0x08			
位置	功能	数据类型	说明	
byte [0]	电池SOC	unsigned int8	范围 0~100	
byte [1]	电池SOH	unsigned int8	范围 0~100	
byte [2] byte [3]	电池电压值高八位 电池电压值低八位	unsigned int16	单位 : 0.01V	
byte [4] byte [5]	电池电流值高八位 电池电流值低八位	signed int16	单位 : 0.1A	
byte [6] byte [7]	电池温度高八位 电池温度低八位	signed int16	单位 : 0.1°C	

指令名称		BMS数据反馈		
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	0x366	500ms	无
数据长度	0x04			
位置	功能	数据类型	说明	
byte [0]	Alarm Status 1	unsigned int8	BIT1 : 过压 BIT2 : 欠压 BIT3 : 高温 BIT4 : 低温 BIT7 : 放电过流	

byte [1]	Alarm Status 2	unsigned int8	BIT0 : 充电过流
byte [2]	Warning Status 1	unsigned int8	BIT1 : 过压 BIT2 : 欠压 BIT3 : 高温 BIT4 : 低温 BIT7 : 放电过流
byte [3]	Warning Status 2	unsigned int8	BIT0 : 充电过流

3.3 TITAN ROS使用说明

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、发布和聚合各种信息（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
- 笔记本电脑 X1
- AGILEX 移动机器人底盘 X1
- AGILEX 配套遥控器FS-i6s X1
- AGILEX 航空插座 X1

使用示例环境说明

- Ubuntu 18.04 LTS
- ROS
- Git

硬件连接与准备

- 将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：

使能 gs_usb 内核模块

∨
复制代码

```
sudo modprobe gs_usb
```

设置500k波特率和使能can-to-usb适配器

∨
复制代码

```
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

∨
复制代码

```
ifconfig -a
```

安装并使用can-utils来测试硬件

∨
复制代码

```
sudo apt install can-utils
```

若此次can-to-usb已经和机器人相连，且小车已经开启的情况下，使用下列指令可以监听来自底盘的数据了

∨
复制代码

```
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

3.4 Github ROS开发包目录与使用说明

titan_base: : TITAN收发层次can消息的核心节点，基于ros的通信机制，可通过topic控制TITAN运动和读取TITAN的状态。

titan_msgs : 定义TITAN状态反馈topic的具体消息格式

titan_bringup : TITAN节点和键盘控制节点的启动文件，以及使能usb_to_can模块的脚本

通过键盘发送指令控制TITAN：

1. 启动底盘节点

```
roslaunch titan_bringup titan_minimal.launch
```

注意启动之前需要先使能松灵配备的usb_to_can模块，使能指令如下：roslaunch titan_bringup bringup_can2usb.bash。该指令只需要每次给上电的usb_to_can模块执行一次。

2. 启动键盘节点

```
roslaunch titan_bringup titan_teleop_keyboard.launch
```

注意查看终端打印，用指定的按键控制TITAN运动

3.5 固件升级

为了方便用户对底盘所使用的固件版本进行升级，TITAN底盘提供了固件升级的硬件接口以及与之对应的客户端软件。其客户端界面如下图所示。

升级准备

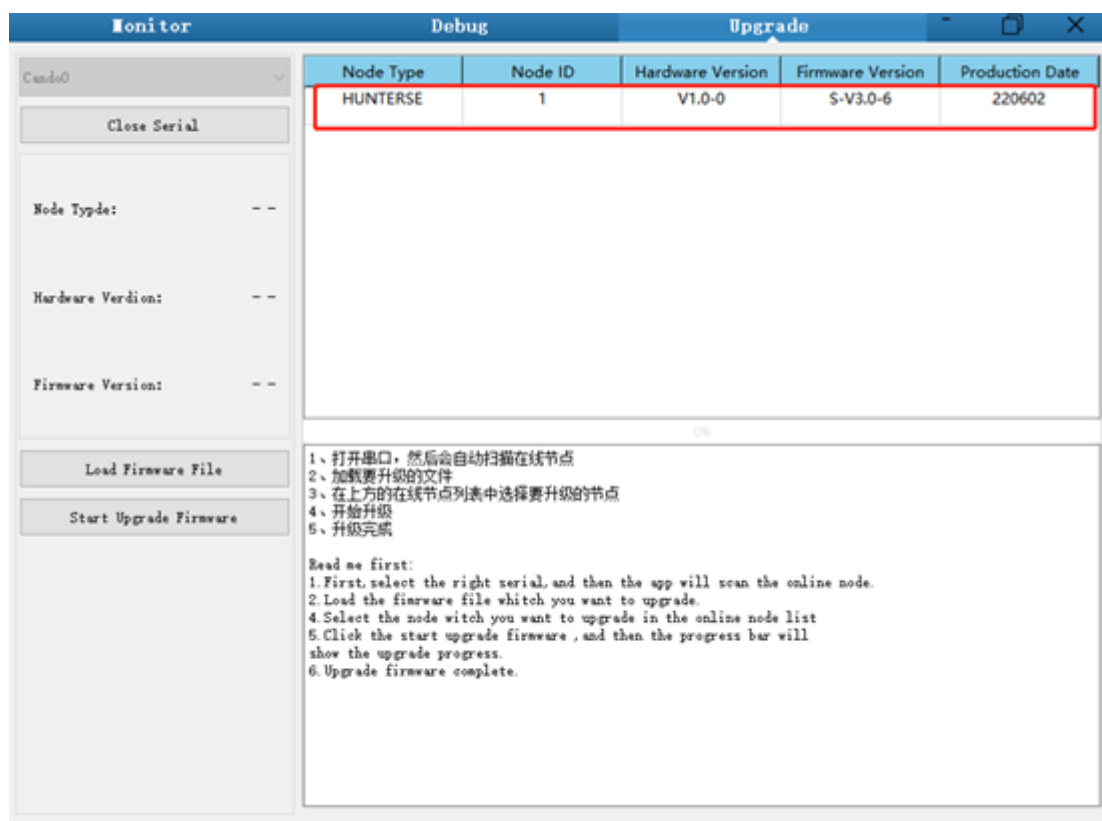
松灵USB TO CAN模块 X 1

micro USB线X 1

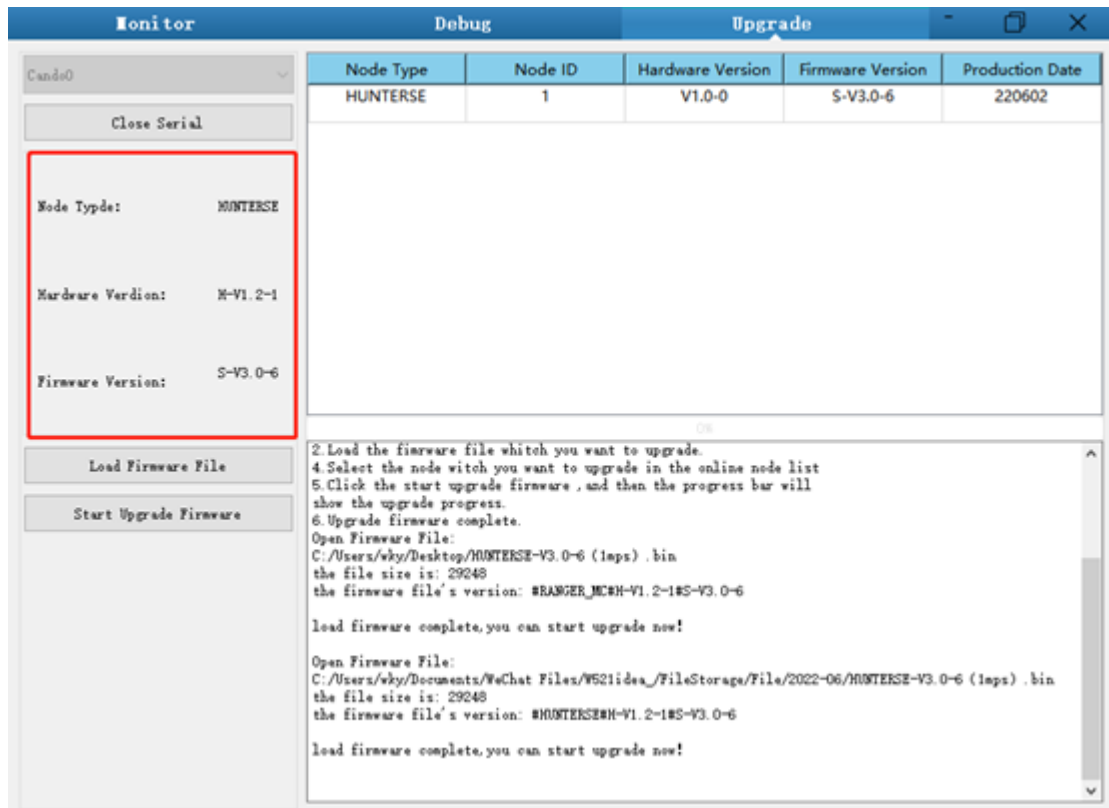
TITAN底盘 X 1

升级过程

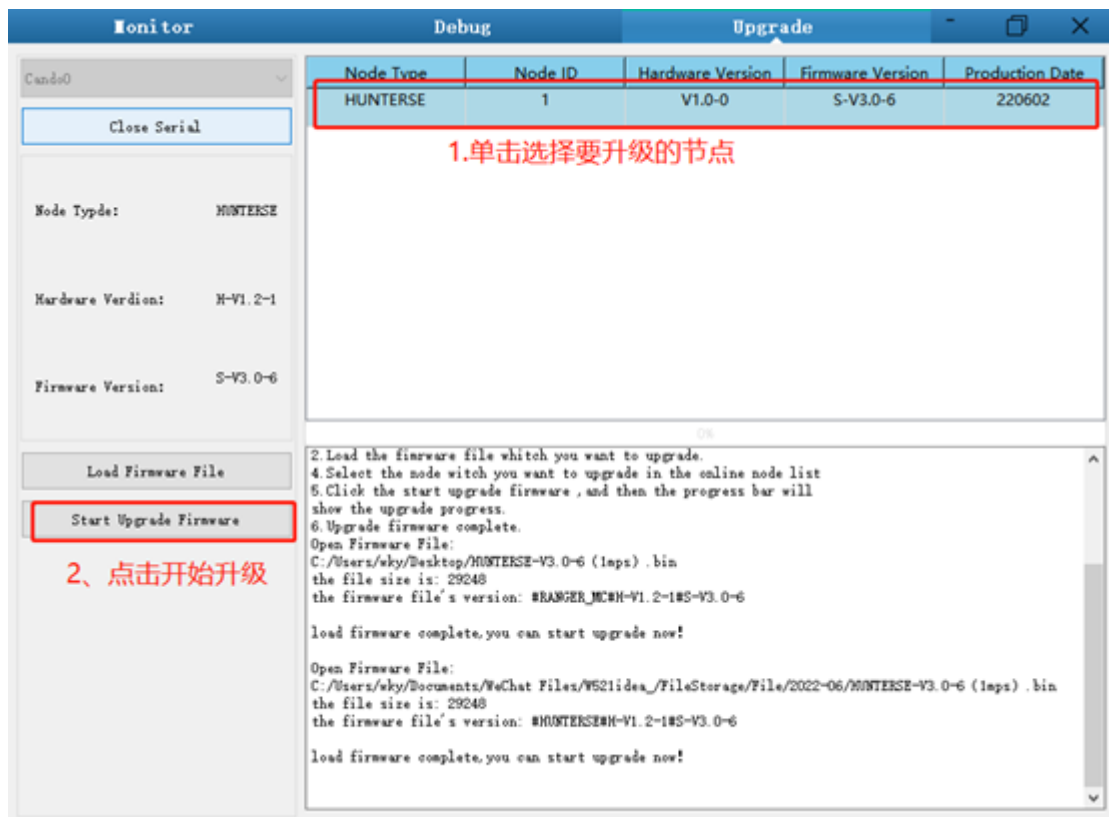
- 1、电脑插上USBTOCAN模块，再打开AgxCandoUpgradeToolV1.3_boxed.exe软件(顺序不能错，先打开软件再插上模块会识别不到设备)
- 2、点击Open Serial按键，然后按下车体电源按键，连接成功的话会识别到主控的版本信息，如图所示

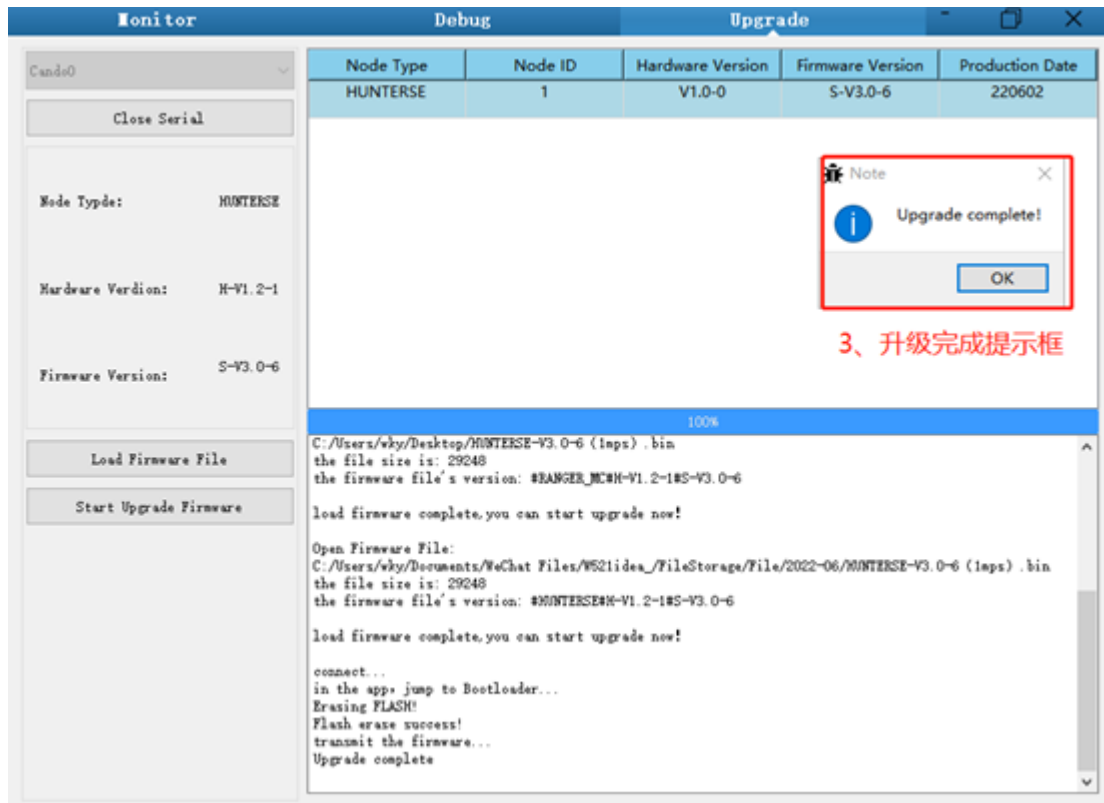


- 3、点击Load Firmware File按钮加载要升级的固件，加载成功会获取到固件信息，如图所示



4、在节点列表框中点击要升级的节点，然后点击Start Upgrade Firmware就可以开始升级固件，升级成功后会弹框提示。





3.6 车体坐标系说明

以车体中心为坐标原点，朝前为x轴正方向，朝左为y轴正方向，朝上为z轴正方向

4 维护说明

4.1 保养方法

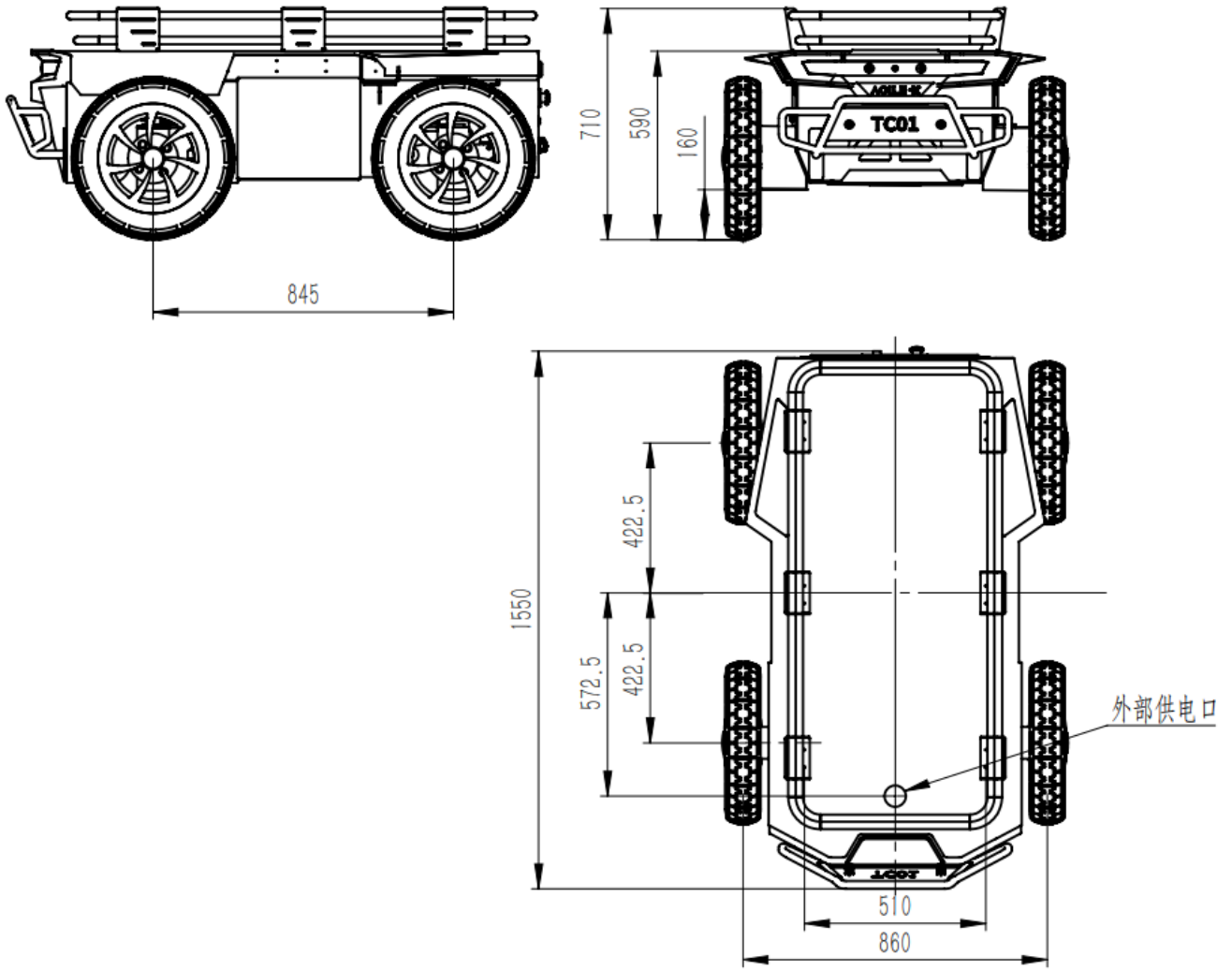
车辆保养

轮胎磨损严重，请及时更换。

电池保养

如果长时间不使用，需要按照2到3个月对电池进行周期性充电。

5、产品尺寸



AGILE·X

松灵机器人(东莞)有限公司

WWW.AGILEX.AI

TEL:+86-0769-22892150

MOBILE:+86-19925374409

