MTools 软件手册

2017

蓝鲸智能机器人（深圳）有限公司

目錄

引言 1.1

一. 硬件接口 1.2

二. 软件安装 1.3

三. 软件概述 1.4

四. 配置驱动器参数 1.5

五. 测试电机相线和霍尔线接线顺序 1.6

六. 驱动器固件更新 1.7

七. 校准IMU 1.8

八. 在ROS中使用 1.9

- 引言

引言

本文档介绍DR03轮毂电机驱动的使用方法，包括驱动器的连接、驱动器的控制、驱动器的参数设置、驱动器的测试等。

轮毂驱动器软件包下载

- 驱动器硬件说明
  - 概述
    - BW-DR03 驱动器参数列表
    - 特点
  - 接口定义
    - 电源输入接口P2
    - 电机相线接口Ma1、Mb1
    - 电机霍尔线接口H0、H1
    - 红外避障模块接口C1、C4、C2、C3
    - 超声波测距模块接口S1、S2
    - 串口通信接口P1
  - 通信协议
    - 电脑下发指令
    - 驱动器上传数据包

驱动器硬件说明

概述

BW-DR03 是本公司研发一款高性能，多功能，低成本的带霍尔传感器直流无刷驱动器。驱动器自带速度闭环PID控制，速度控制精度在1%以内，低速输出扭距大，高速制动迅速。支持同时控制两路电机，自带IMU，可外接4路红外避障模块和2路超声波测距模块。驱动器电机速度控制和参数配置都通过rs232串口实现，配合上位机配置软件和ROS驱动包可以快速搭建高性能机器人底盘。

BW-DR03 驱动器参数列表

参数	说明
电机驱动管输入电压	DC12V-46V
电机霍尔类型	120度
工作电流	最大30A（单路15A）
工作模式	霍尔速度闭环
调速方式	RS232串口指令
IMU	MPU9250带3轴加速度计、3轴陀螺仪、3轴磁力计
红外避障模块	最多支持4路
超声测距模块	最大支持2路
工作环境	场合：无易燃、易爆气体，无粉尘 温度：-10-50摄氏度 振动：小于0.5G，10HZ-60HZ 防护等级：不防水
散热方式	自然风冷
尺寸	128*156*32里面
重量	500克

特点

速度 PID 闭环控制，低速转矩大。

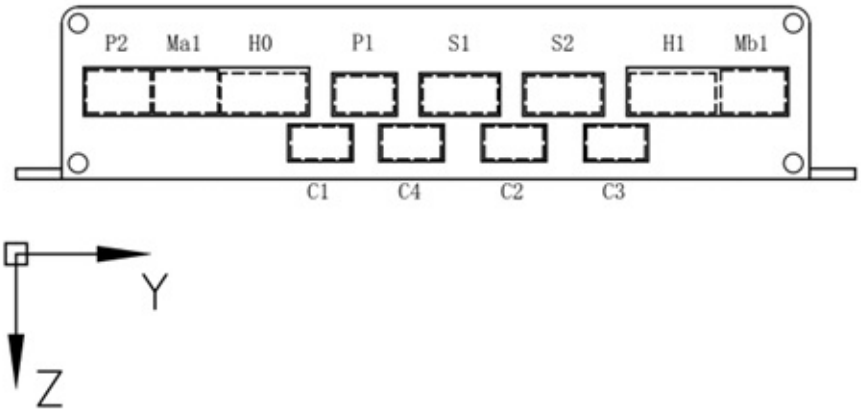
调速范围宽，0-2000RPM。

支持电机电压范围宽，12V-36V。

支持同时控制两路电机，自带IMU，可外接4路红外避障模块和2路超声波测距模块。

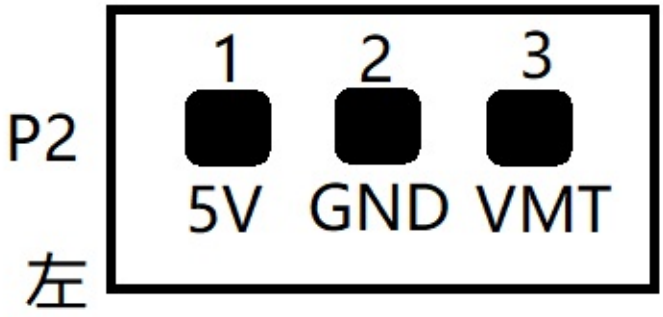
带ROS驱动包，可以直接输出里程计、IMU这些话题数据。

接口定义



驱动器正面接口分布图

电源输入接口P2



接口	说明
1. 5V	RS232版本，此脚被废弃，请悬空不接东西
2. GND	电源地
3. VMT	电机驱动管电源正输入

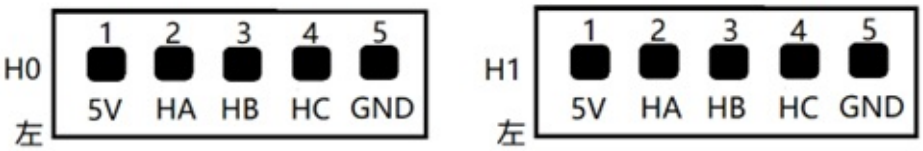
VMT电压值由电机额定电压确定，比如36v电机就接入36v电池的正极即可，注意必须用电池直接供电，通过dc稳压电源供电会烧毁驱动器。

电机相线接口Ma1、Mb1

接口	说明
1. mA	电机相线U(黄)
2. mb	电机相线V(绿)
3. mC	电机相线W(蓝)

电机相线颜色的定义与电机厂家有关，实际接线时，请先将电机相线任意接入mA mB mC、然后调整下面的霍尔相线（理论上任意的电机相线接线，都有一组对应的霍尔接线，因此可以先任意指定电机相线接线顺序），当驱动器在开环模式下可以正常控制电机正反转时就说明霍尔相线和电机相线接线顺序是正确的。具体请参考驱动器配置软件手册相关章节。

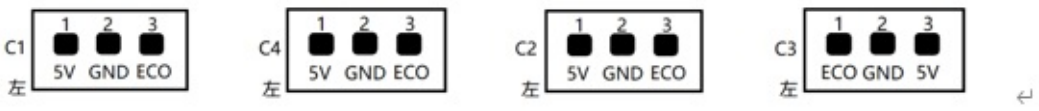
电机霍尔线接口H0、H1



接口	说明
1. 5V	霍尔供电正极
2. HA	霍尔U线(黄)
3. HB	霍尔V线(绿)
4. HC	霍尔W线(蓝)
5. GND	霍尔供电地

H0对应Ma1通道电机， H1对应Mb1通道电机 驱动器正面朝下反面安装时，Ma1对应右轮电机，Mb1对应左轮电机；驱动器正面朝上安装时，Ma1对应左轮电机，Mb1对右轮电机。

红外避障模块接口C1、C4、C2、C3

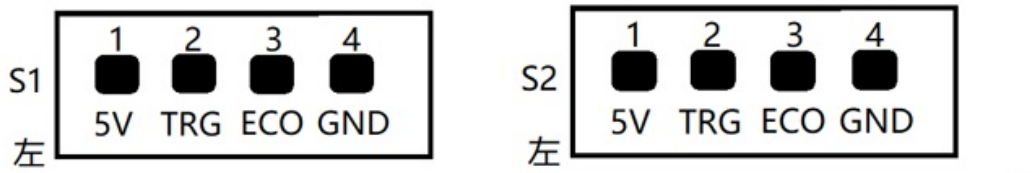


接口	说明
5V	模块电源正极
GND	模块电源地

ECHO	模块输出脚

请注意C3和C1、C4、C2三个的引脚排列方向是不同的，通常C1、C4、C2用于检测底盘前方的障碍物，C3用于检查底盘后方的障碍物，因此排列方向设计成不同。 模块被障碍物触发时，模块ECO脚输出低电平（驱动器反馈0），其它情况下模块ECO输出高电平（驱动器反馈1）。

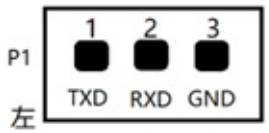
超声波测距模块接口S1、S2



接口	说明
1. 5V	模块电源正极
2. TRG	模块测量触发脚
3. ECO	模块测量数据输出脚
4. GND	模块电源地

本驱动支持的是串口一体化超声波测距模块，驱动器测量频率是5hz，上传数据单位是米。

串口通信接口P1



接口	说明
1. TXD	驱动器串口数据上传端口
2. RXD	驱动器串口命令接收端口
3. GND	串口地

驱动器通信连接方式是RS232串口，串口波特率为 115200，8 个数据位，1 个停止位，无奇偶校验。

通信协议

串口波特率为115200，8个数据位，1个停止位，无奇偶校验。

电脑下发指令

方向运动指令都是6个字节的无符号byte数组，驱动器正面朝下反面安装时，Ma1对应右轮电机，Mb1对应左轮电机；驱动器正面朝上安装时，Ma1对应左轮电机，Mb1对右轮电机。

前进

0xcd	0xeb	0xd7	0x02	0x66	0xXX
包头	包头	包头	命令长度	前进指令	速度大小，数值范围为0到100

后退

0xcd	0xeb	0xd7	0x02	0x62	0xXX
包头	包头	包头	命令长度	后退指令	速度大小，数值范围为0到100

左转

0xcd	0xeb	0xd7	0x02	0x63	0xXX
包头	包头	包头	命令长度	左转指令	速度大小，数值范围为0到100

右转

0xcd	0xeb	0xd7	0x02	0x64	0xXX
包头	包头	包头	命令长度	右转指令	速度大小，数值范围为0到100

停止

0xcd	0xeb	0xd7	0x02	0x73	0xXX
包头	包头	包头	命令长度	停止指令	制动量大小，数值范围为0到100

单个电机独立控制指令是13字节的

0xcd	0xeb	0xd7	0x09	0x74	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0x
包头	包头	包头	命令长度	类型	右轮电机指令	左轮电机指令	保留	保留	控制量大小	控制量大小	保

电机指令有三种状态，‘F’：向前，‘B’：向后，‘S’：刹车 控制量范围是0到100，

例如： tSSSS0000 翻译成hex为 cd eb d7 09 74 53 53 53 53 00 00 00 00 表示让四个电机全部以0%的制动量刹车 目前只有前两个电机有效，第一个电机对应右轮，第二个电机对应左轮

驱动器上传数据包

驱动器以50HZ的频率上传数据，上传的数据包格式：包头+长度+内容

包头：为3个u8字符：205 235 215

长度：1个u8字符，长度不包括包头和长度本身字符，当前数据包长度为125,这 个长度包括字符串结束符0x00

内容：由25个4字节小端模式二进制表示的数字组成，数字之间用空格0x20分开。

完整数据包内容构成一个c语言结构体，结构体具体构成如下所示：

```
typedef struct {
    int status;//驱动器状态，0表示未初始化，1表示正常，-1表示error
    float power;//电源电压【12 46】v
    float theta;//方位角，【0 360】°
    unsigned int encoder_ppr;//车轮1转对应的编码器个数
```

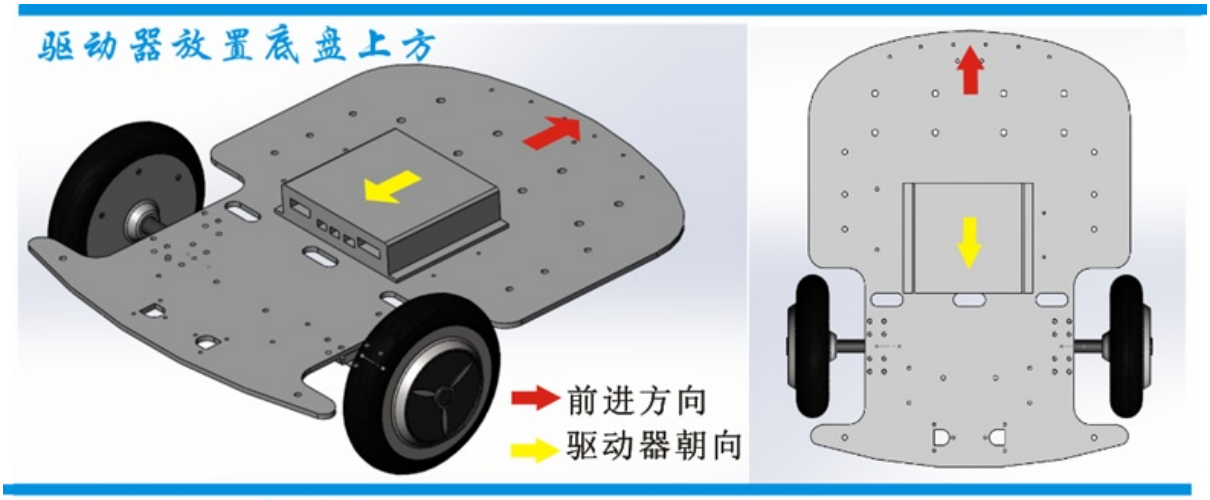
```
    int encoder_delta_r;//右轮编码器增量，  个为单位
    int encoder_delta_l;//左轮编码器增量，  个为单位
    int encoder_delta_car;//两车轮中心位移，个为单位
    unsigned int upwoard;//0表示正面朝下安装，1表示正面朝上安装
    float max speed;//最大转速，圈每秒
    int hbz1;//第一个红外避障模块状态，1表示触发，0表示没有触发
    int hbz2;//第一个红外避障模块状态，1表示触发，0表示没有触发
    int hbz3;//第一个红外避障模块状态，1表示触发，0表示没有触发
    int hbz4;//第一个红外避障模块状态，1表示触发，0表示没有触发
    float distance1;//第一个超声模块距离值 单位cm
    float distance2;//第二个超声模块距离值 单位cm
    float IMU[9];//mpu9250 9轴数据
    unsigned int time_stamp;//时间戳
}UPLOAD_STATUS;
```

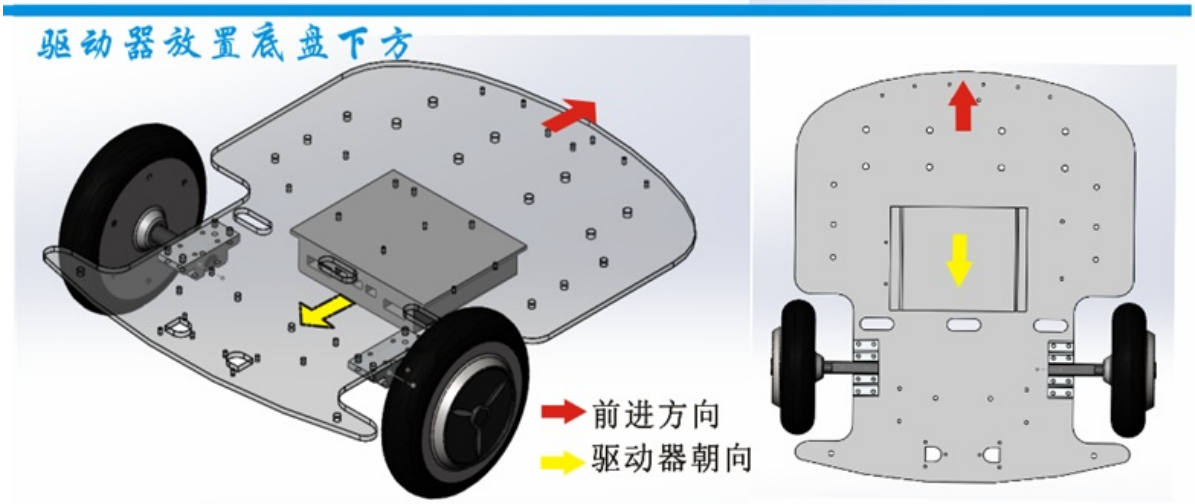
IMU[9]数组里面的数据依次为下表

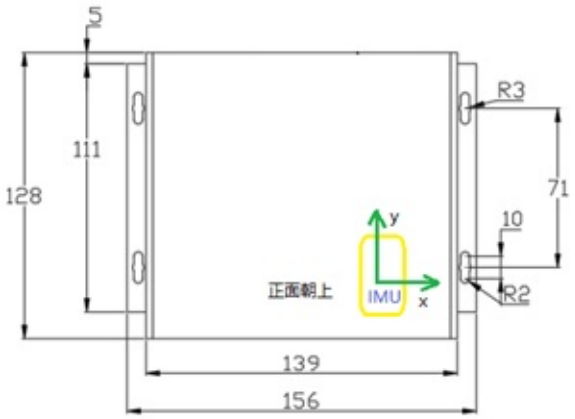
```
    0: 加速度x
    1: 加速度y
    2: 加速度z
    3: 角速度x
    4: 角速度y
    5: 角速度z
    6: 磁力计x
    7: 磁力计y
    8: 磁力计z
```

数据包的使用请参考 ROS驱动包lungu分支对应部分代码。

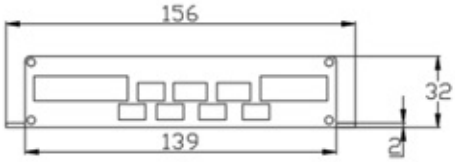
安装尺寸


驱动器放置底盘上方
前进方向
驱动器朝向


驱动器放置底盘下方
前进方向
驱动器朝向



俯视尺寸图

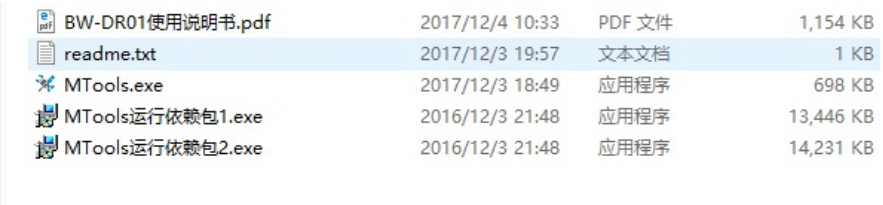


前视尺寸图

- 1.1 软件安装
  - 1.2 串口驱动参数配置

1.1 软件安装

打开驱动器资料包，如下图所示



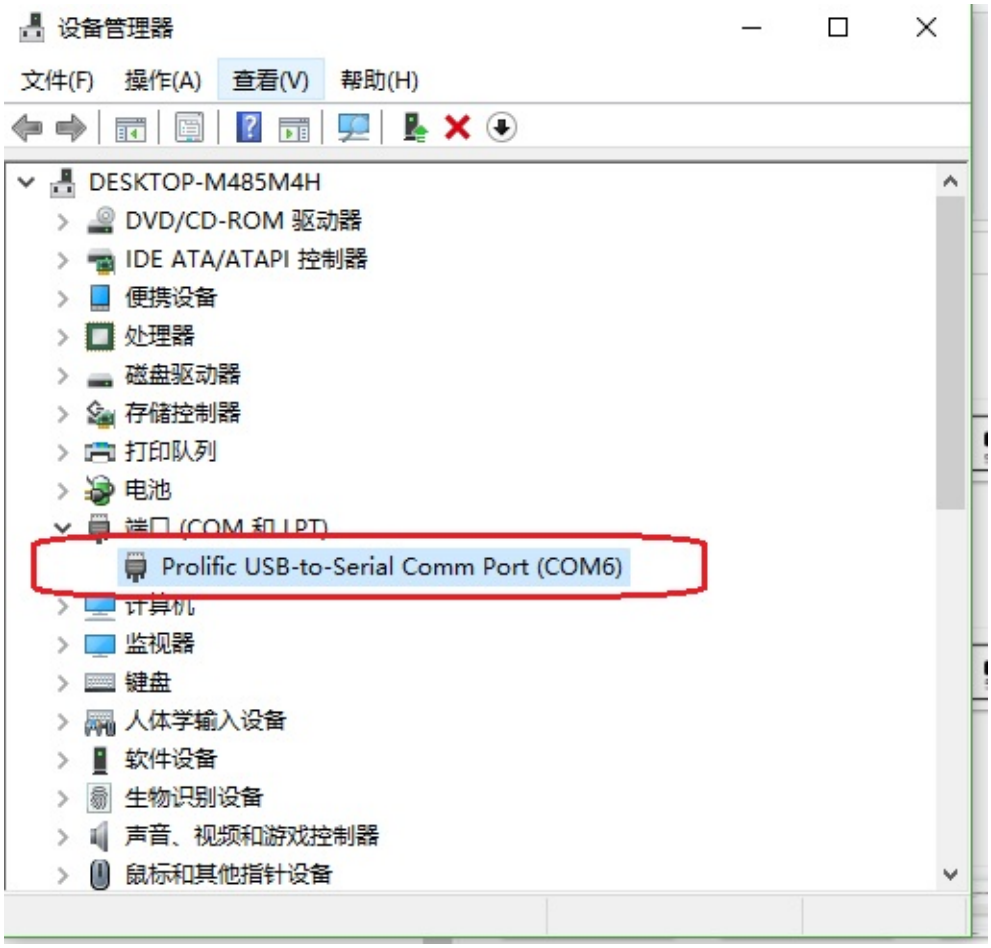
先双击安装MTools运行依赖包1.exe 、MTools运行依赖包2.exe，配置好软件运行依赖环境。

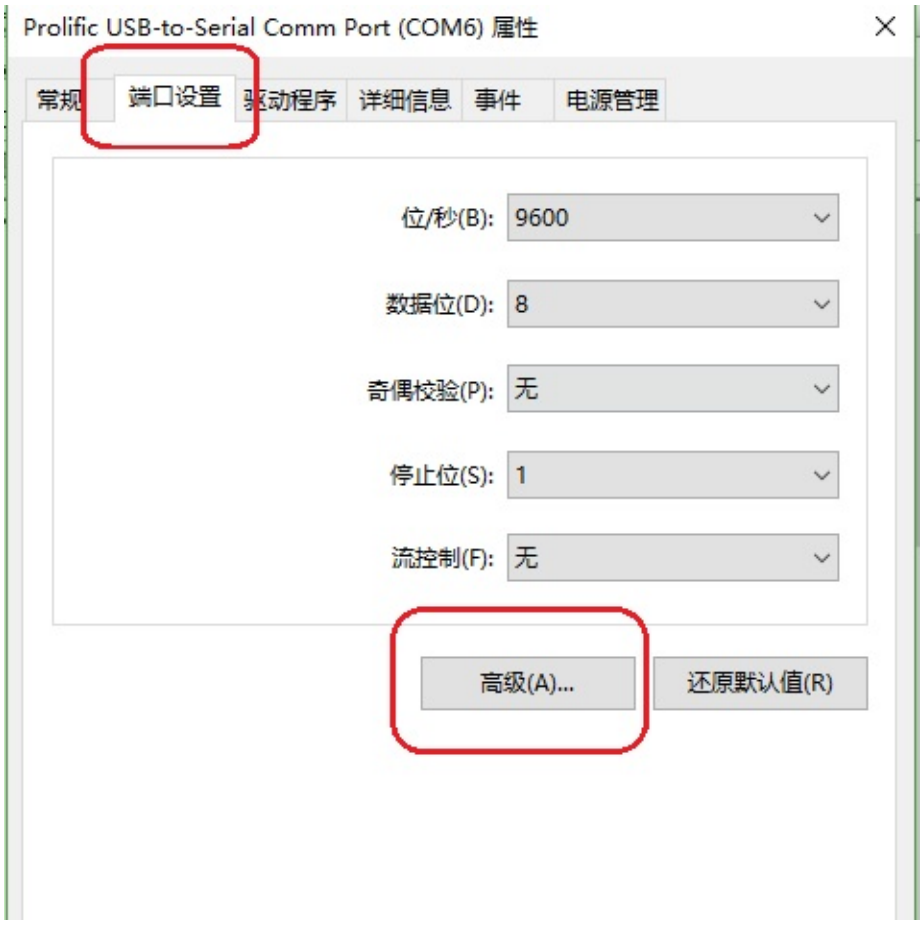
MTools.exe是驱动器调试软件，不用安装，双击直接运行。推荐在桌面建立软件快捷方式，方便后续使用。快捷方式建立步骤：MTools.exe右键 —> “发送到”—>“桌面快捷方式”

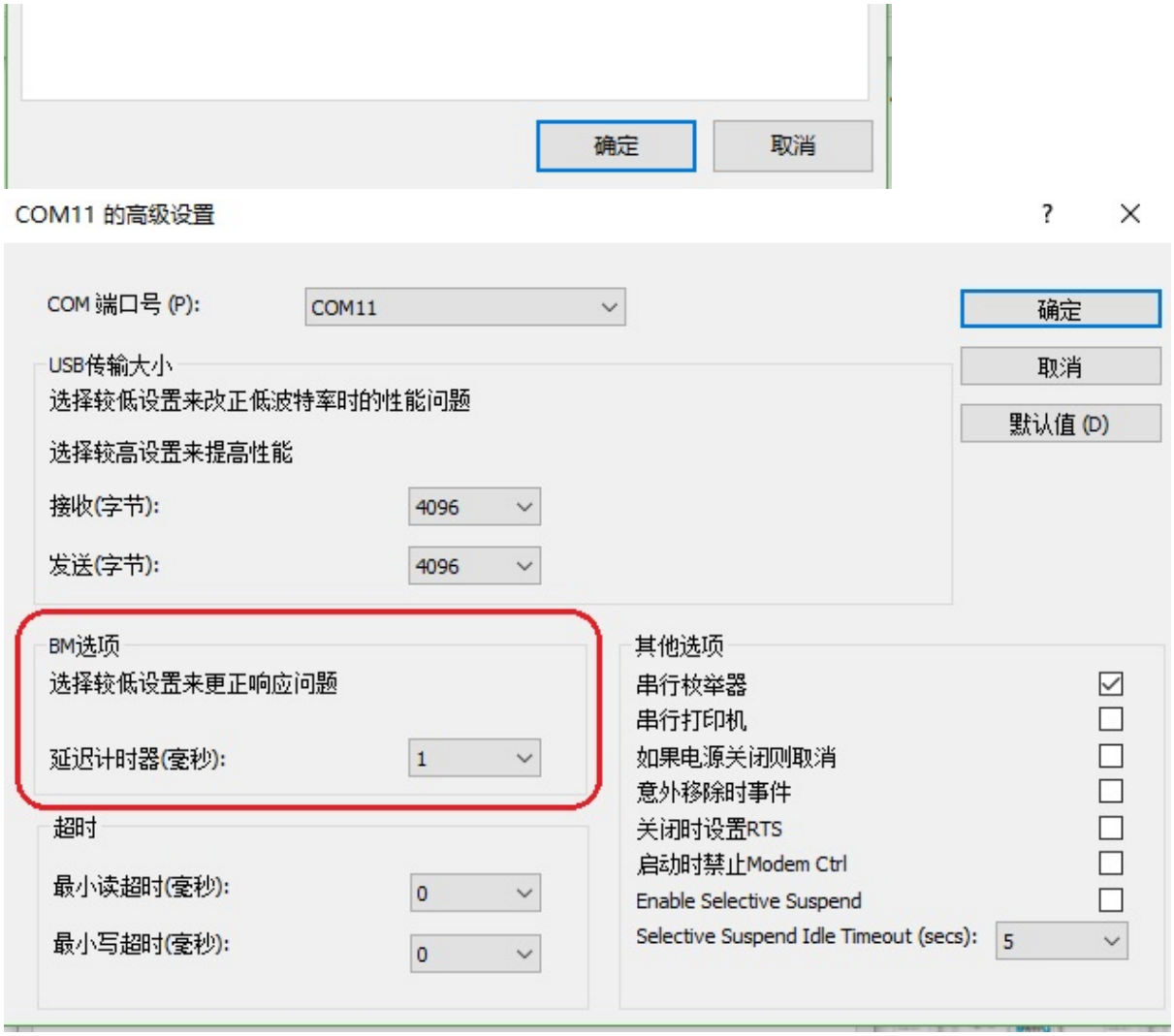
1.2 串口驱动参数配置

MTools.exe软件和驱动器通信连接方式是rs232串口，串口波特率为 115200，8 个数据位，1 个停止位，无奇偶校验。

对于windows系统，usb转串口模块的驱动有一个延时参数，默认值不合理时会导致驱动器上传数据不能及时处理，请根据下图设置成最小值（如果不存在这个选项则不用设置）。


设备管理器
文件(F) 操作(A) 查看(V) 帮助(H)
DESKTOP-M485M4H
DVD/CD-ROM 驱动器
IDE ATA/ATAPI 控制器
便携设备
处理器
磁盘驱动器
存储控制器
打印队列
电池
端口 (COM 和 LPT)
Prolific USB-to-Serial Comm Port (COM6)
计算机
监视器
键盘
人体学输入设备
软件设备
生物识别设备
声音、视频和游戏控制器
鼠标和其他指针设备


Prolific USB-to-Serial Comm Port (COM6) 属性
常规
端口设置
驱动程序
详细信息
事件
电源管理
位/秒(B): 9600
数据位(D): 8
奇偶校验(P): 无
停止位(S): 1
流控制(F): 无
高级(A)...
还原默认值(R)


确定
取消
COM11 的高级设置
COM 端口号 (P):
COM11
确定
取消
默认值 (D)
USB传输大小
选择较低设置来改正低波特率时的性能问题
选择较高设置来提高性能
接收(字节):
4096
发送(字节):
4096
BM选项
选择较低设置来更正响应问题
延迟计时器(毫秒):
1
超时
最小读超时(毫秒):
0
最小写超时(毫秒):
0
其他选项
串行枚举器
串行打印机
如果电源关闭则取消
意外移除时事件
关闭时设置RTS
启动时禁止Modem Ctrl
Enable Selective Suspend
Selective Suspend Idle Timeout (secs):
5

- 二、概述
  - 2.1 软件界面
  - 2.2 注意事项

二、概述

通过MTools.exe软件，可以配置驱动器的基本运行参数，也可以方便地对电机进行在线调试测试。本软件可以实时预览IMU和传感器外设上传的数据，能协助诊断驱动器和外设模块的健康状态。

2.1 软件界面

软件界面主要分为三个区：串口配置面板、数据预览面板、控制面板。界面大小可以任意调整，界面底部蓝色状态栏会根据操作提供实时反馈信息，请留意实际提示内容。

打开串口，连接上驱动器后，数据预览面板会实时更新数据。

驱动器上电后会有一段初始化时间，状态为0。初始化完成后，状态会变成1，同时背景颜色变成绿色，在这种情况下才可以操作“控制面板”。

2.2 注意事项

A．关闭软件或者关闭串口时，如果出现软件死机现象，请用“任务管理器”强制关闭软件，然后重新插拔USB串口设备，重启软件后会恢复正常。

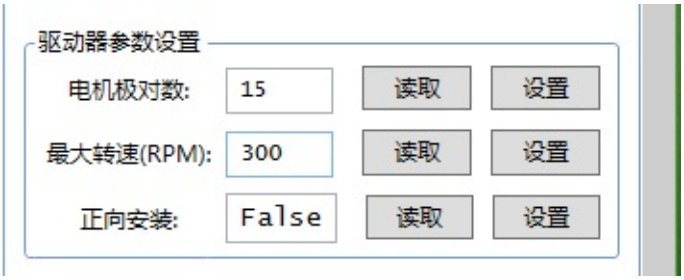
B．驱动器第一次连接电机进行调试时，如果霍尔线序没有确定好，请先操作章节三、四。

否则误操作容易损坏驱动器开关管。

- 配置驱动器参数
  - 3.1电机极对数
  - 3.2最大转速控制
  - 3.3安装方向和左右电机通道对应关系
  - 3.4通用操作

配置驱动器参数

驱动器上电后会有一段初始化时间，状态为0。初始化完成后，状态会变成1，同时背景颜色变成绿色，在这种情况下才可以操作“控制面板”。



3.1电机极对数

电机极对数即指磁极对数，通常的轮毂无刷电机值为15或者12。点击”读取”，可以得到驱动器当前设置值，点击“设置”则将当前值下发写入驱动器ROM。 如果不知道电机的磁极对数实际值，可以通过统计电机转一圈时软件记录的编码器计数值计算得到。

磁极对数 = 1圈计数 ÷ 6

3.2最大转速控制

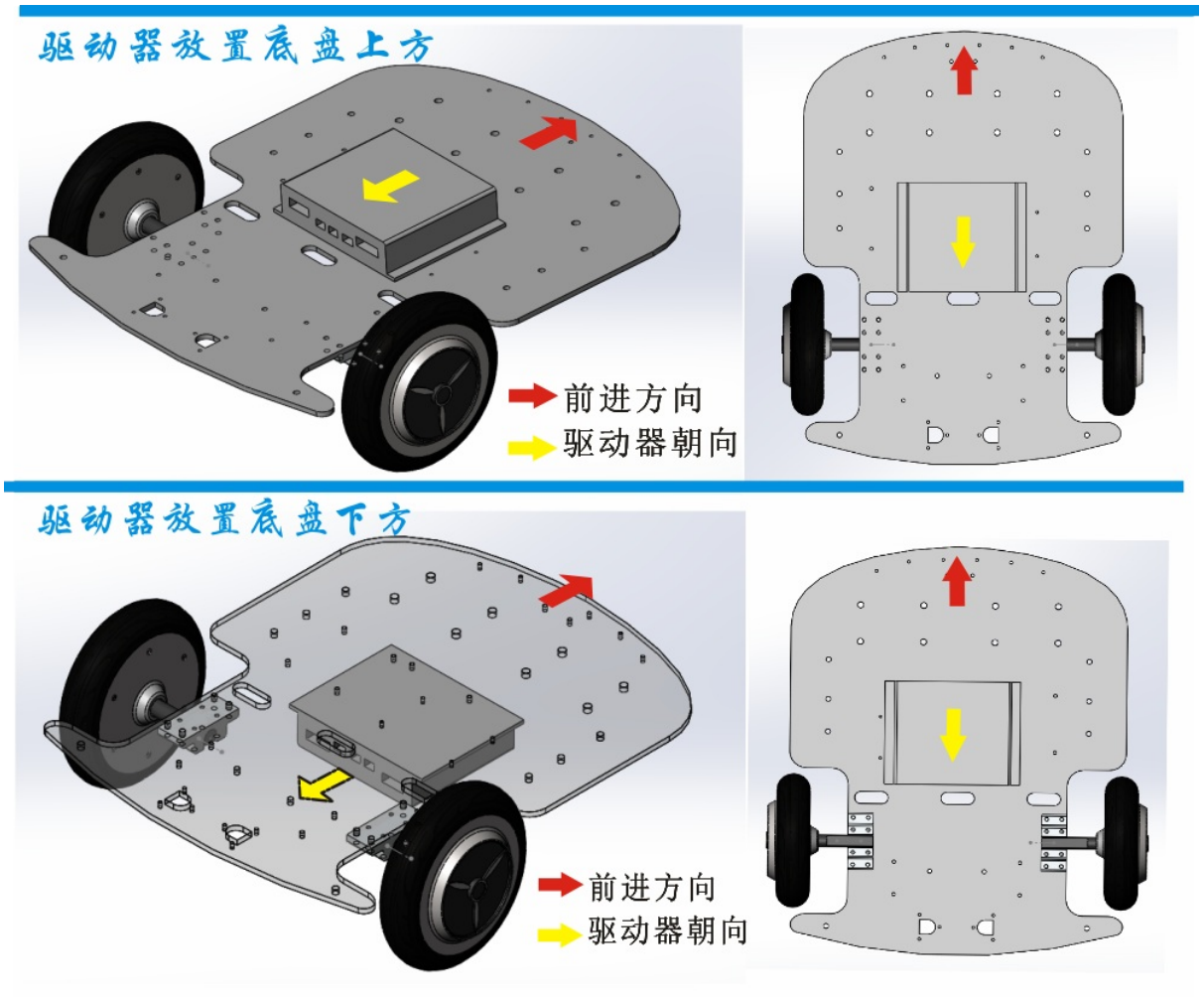
点击”读取”，可以得到驱动器当前设置值，点击“设置”则将当前值下发写入驱动器ROM。 电机的最大转速推荐设成实际需要的值，而不是电机可以实现的最大值，这样可以提高低速控制精度。 对于电机可以实现的最大转速，请在“速度开环”模式下，下发100的转速控制信号，转速反馈值即为最大转速，最大可实现转速和电池电量有关。

3.3安装方向和左右电机通道对应关系

点击”读取”，可以得到驱动器当前设置值，点击“设置”则将当前值下发写入驱动器ROM，值只能为false、true这两种情况。 正向安装是指驱动器正面朝上水平安装，反向安装是指驱动器正面朝下水平安装。两种安装方向下，为了接线方便，驱动器左右电机对应的接线口是相反的。我们假设驱动器是接后驱电机。

电机	正向安装 (True)	反向安装 (False)

左轮电机	Ma1 H0	Mb1 H1
右轮电机	Mb1 H1	Ma1 H0



3.4通用操作

“重启”使驱动器控制芯片重启，“急停”可以同时制动两个电机，“烧录”将驱动器配置成固件更新状态，“开环”将驱动器临时配置成速度开环控制模式，“校准”启动驱动器IMU自动校准程序。

- 四、测试电机相线和霍尔线接线顺序
  - 4.1 速度开环模式
  - 4.2 确定霍尔线顺序
  - 4.3 速度闭环调速测试
  - 4.4 调整电机正转方向

四、测试电机相线和霍尔线接线顺序

对于本驱动器，理论上任意的电机相线接线，都有一组对应的霍尔接线使电机正常运作，因此可以先任意指定电机相线接线顺序，这里我们指定mA接电机黄线、mB接电机绿线、mC接电机蓝线；下文会介绍如何确定霍尔线序。

霍尔线序不正确情况下，驱动器需要设置成“速度开环模式”，同时转速下发指令值不要大于20，这样可以防止异常大电流顺坏电机线圈和驱动管。两个电机的线序是一样的，即左轮和右轮的电机电机线、霍尔线的顺序是一模一样的，因此下文的测试只针对左轮电机。

4.1 速度开环模式

点击“开环”，驱动器会临时进入“速度开环”控制模式，此时电机PWM开关量直接正比于转速指令。重启驱动器会使驱动器退出“速度开环”控制模式，进入默认的速度闭环模式。

4.2 确定霍尔线顺序

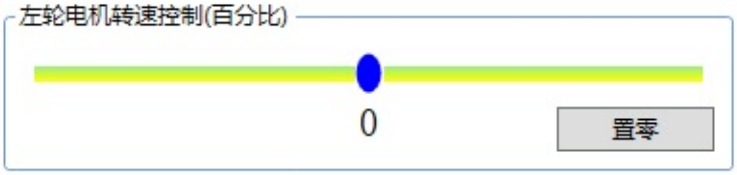
保证驱动器处于“速度开环”模式，以左轮电机为例，假设电机霍尔线三根的颜色分别为黄1、绿2、蓝3。

霍尔线接口	排列1	排列2	排列3	排列4	排列5	排列6
HA	黄1	黄1	绿2	绿2	蓝3	蓝3
HB	绿2	蓝3	黄1	蓝3	黄1	绿2
HC	蓝3	绿2	蓝3	黄1	绿2	黄1

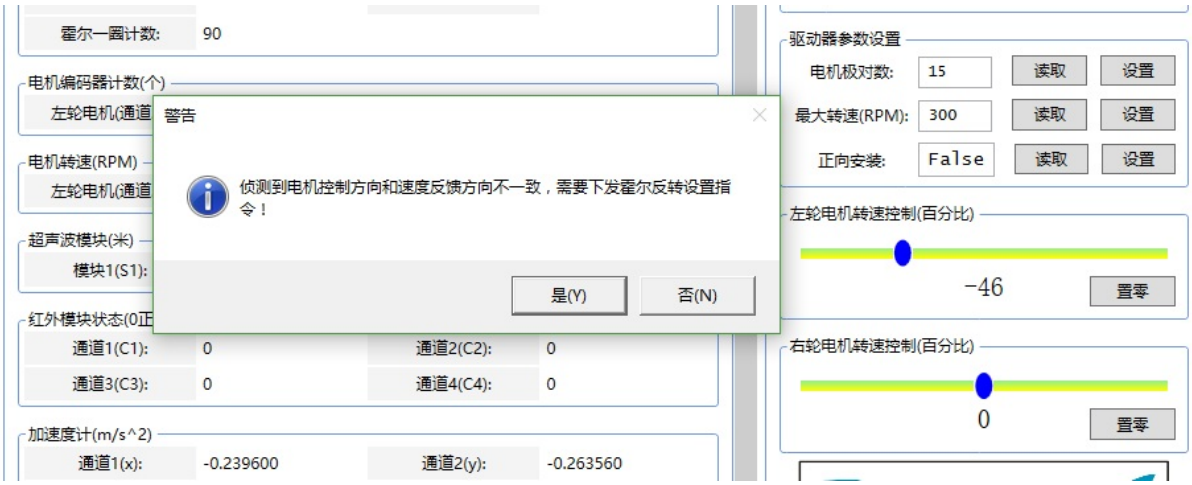
依次尝试上表6种接线顺序，对于每一种接线顺序：

a.下发10控制量看电机是否流畅运转，如果不正常则说明这组顺序是错误，请“置零”然后测试下一组接线；如果正常则执行操作b。

b.下发-10控制量看电机是否流畅运转，如果不正常则说明这组顺序是错误，请“置零”然后回到a测试下一组接线；如果正常则说明霍尔线序是正确的，继续步骤c。



c．现在电机基本可以正常运转了，最后需要确定控制量方向和转速反馈方向是一致的，比如下发大于零的转速控制量，正常应该返回大于零的转速，这样才能速度闭环。先正向加大速度控制量，使转速大于200RPM或者小于-200RPM；然后反向加大速度控制量，同样使转速大于200RPM或者小于-200RPM。操作完成后，软件会自动判断控制方向是否与转速方向相反，如果出现下图提示，则按“是”，如果没有出现，说明调试已经完成可以继续速度闭环测试。



4.3 速度闭环调速测试

根据上节确定的霍尔线序，把右轮电机也接入驱动器，然后重启驱动器，等驱动器初始化完毕。现在任意调整下图滑块，即可设置电机转速。

控制的同时，还可以观察速度反馈值。

4.4 调整电机正转方向

速度闭环控制成功的前提下，如果发现电机正转方向和自己需要的方向是相反的，首先根据3.3节判断安装方向是否设置正确，如果安装方向设置正确，则将电机相线的ma 和Mb对调,同时把霍尔线的HA和HC对调，这样可以把电机正转方向调换一下。

- 五、驱动器固件更新
- 5.1 将驱动器设置成烧录模式
- 5.2 使用Flash Loader Demo工具更新驱动器固件

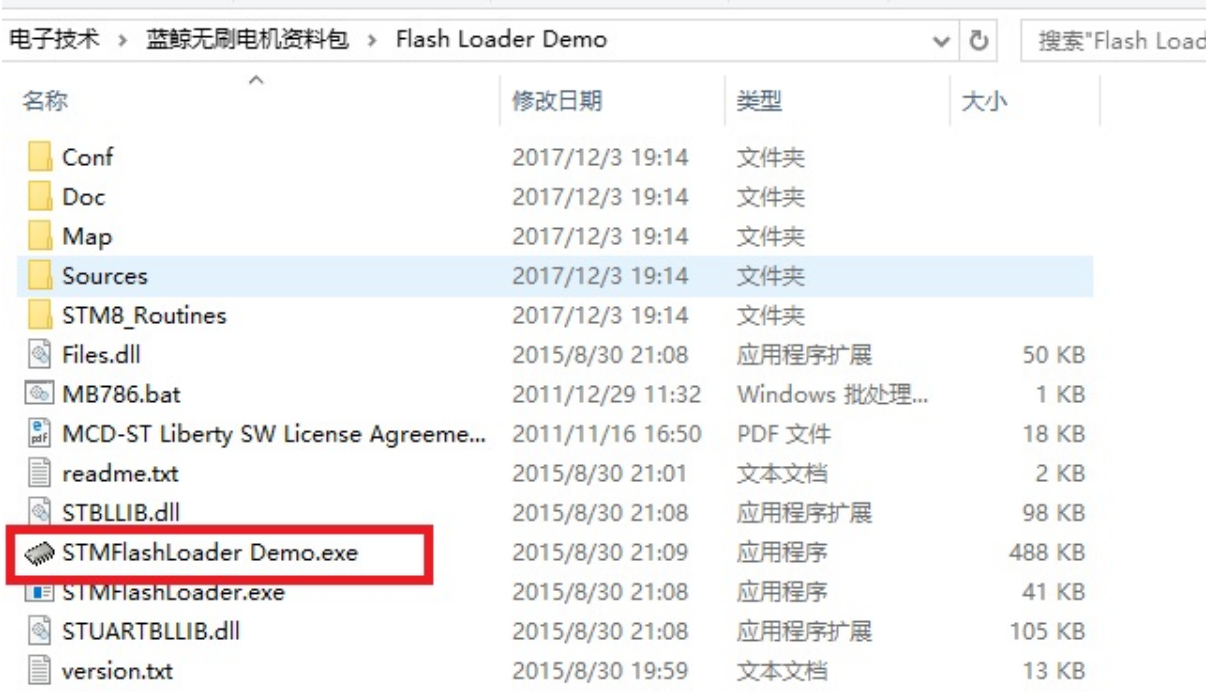
五、驱动器固件更新

5.1 将驱动器设置成烧录模式

打开串口后，点击“烧录”可以将驱动器设置成烧录模式。进入烧录模式后，请关闭驱动器软件，继续下一步骤。

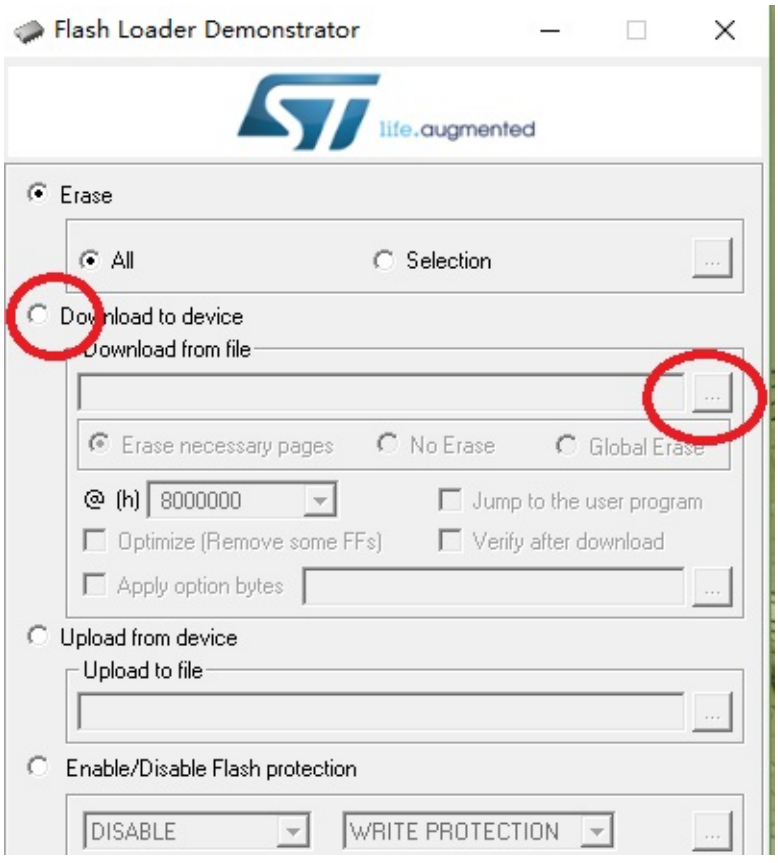
5.2 使用Flash Loader Demo工具更新驱动器固件

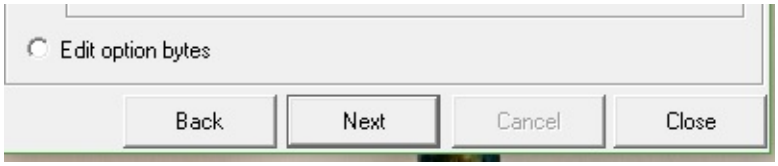
打开STMFlashLoader Demo.exe软件



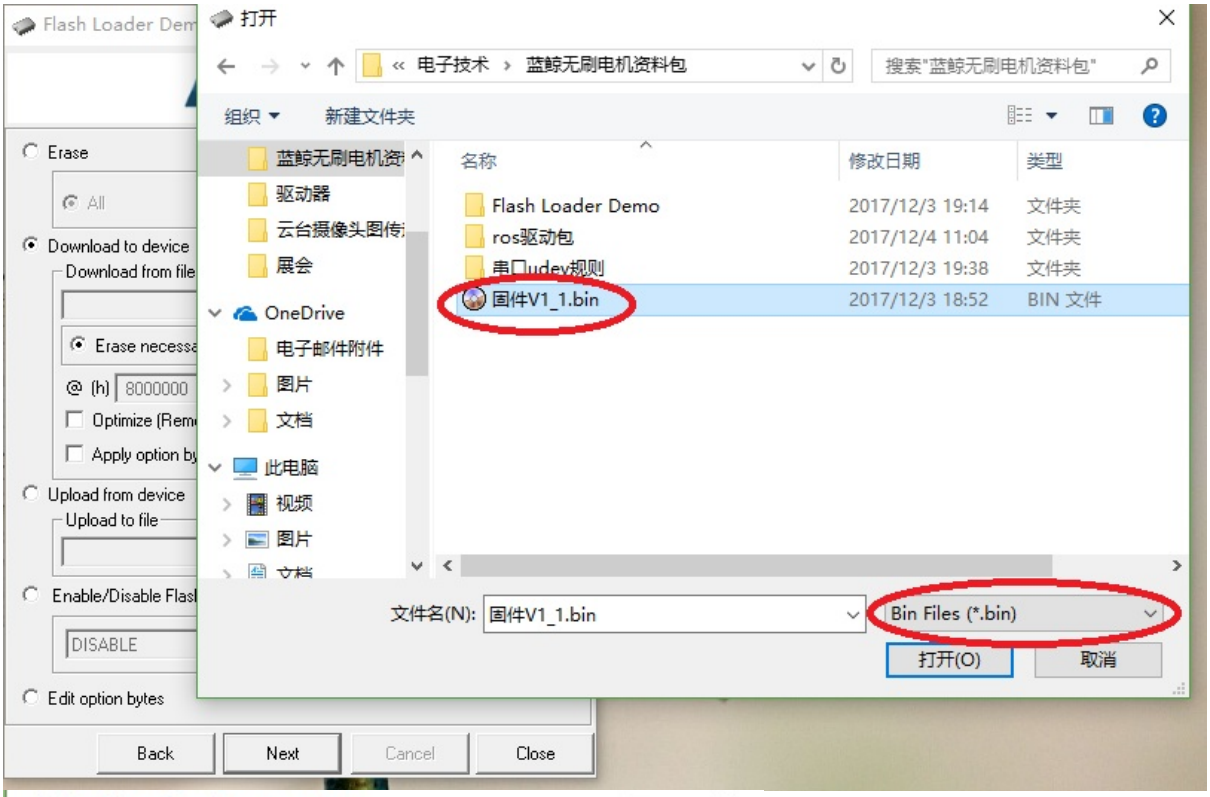
选择串口号，一直点击“next”,直到出现烧录选项界面。

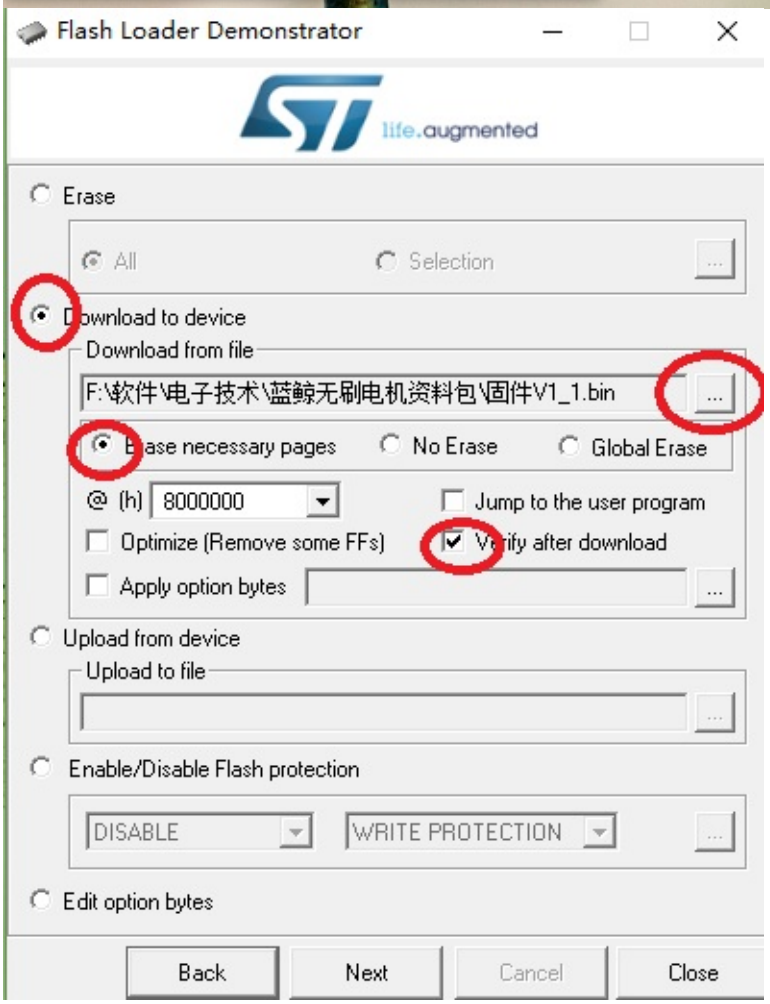
Flash Loader Demonstrator
life.augmented
Select the communication port and set settings, then click next to open connection.
Common for all families
UART
Port Name
COM5
Parity
Even
Baud Rate
115200
Echo
Disabled
Data Bits
8
Timeout(s)
10
Back
Next
Cancel
Close


Flash Loader Demonstrator
life.augmented
Erase
All
Selection
Download to device
Download from file
Erase necessary pages
No Erase
Global Erase
@ (h)
8000000
Jump to the user program
Optimize (Remove some FFs)
Verify after download
Apply option bytes
Upload from device
Upload to file
Enable/Disable Flash protection
DISABLE
WRITE PROTECTION

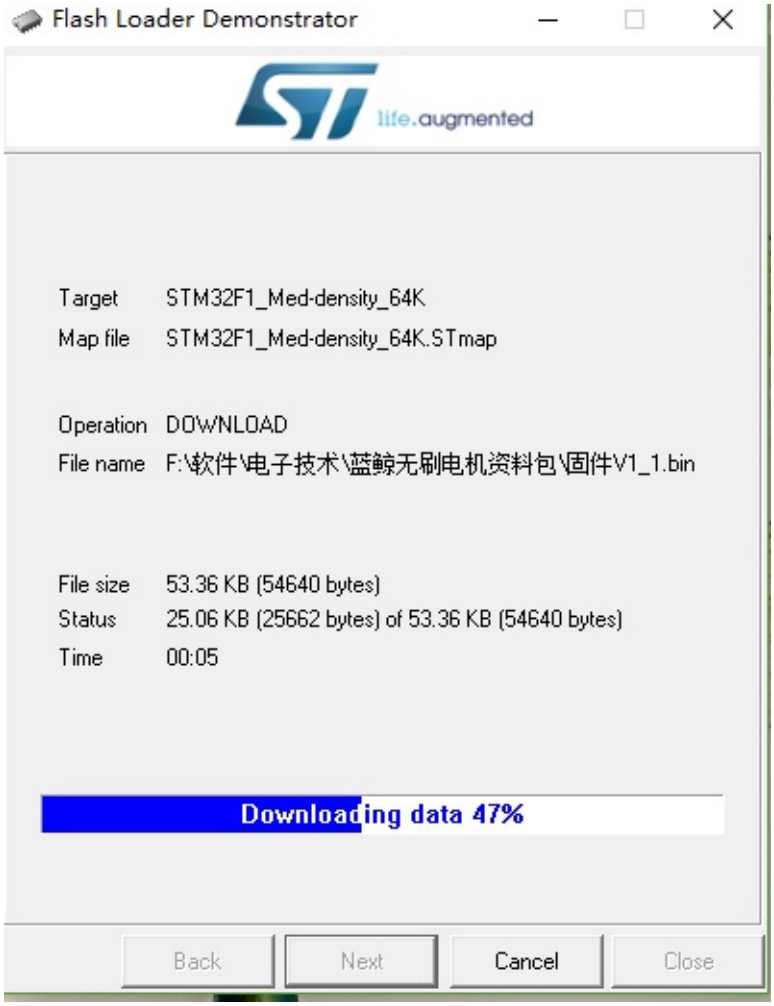


选择第二个“Download to device”选项，加载固件文件。





点击“next”，开始固件更新。



固件更新成功后，关闭软件，给驱动器重新上电，即可正常使用。

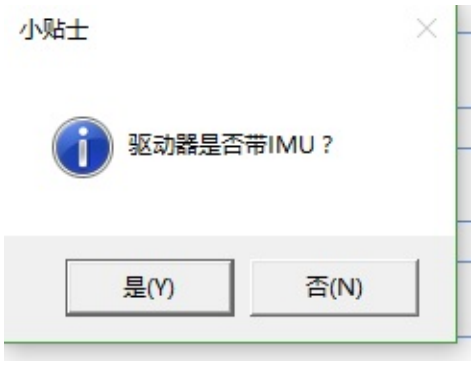
Flash Loader Demonstrator
life.augmented
Target STM32F1_Med-density_64K
Map file STM32F1_Med-density_64K.STmap
Operation DOWNLOAD
File name F:\软件\电子技术\蓝鲸无刷电机资料包\固件V1_1.bin
File size 53.36 KB (54640 bytes)
Status 53.36 KB (54640 bytes) of 53.36 KB (54640 bytes)
Time 00:15
Download operation finished successfully
Back
Next
Cancel
Close

- 六、校准IMU

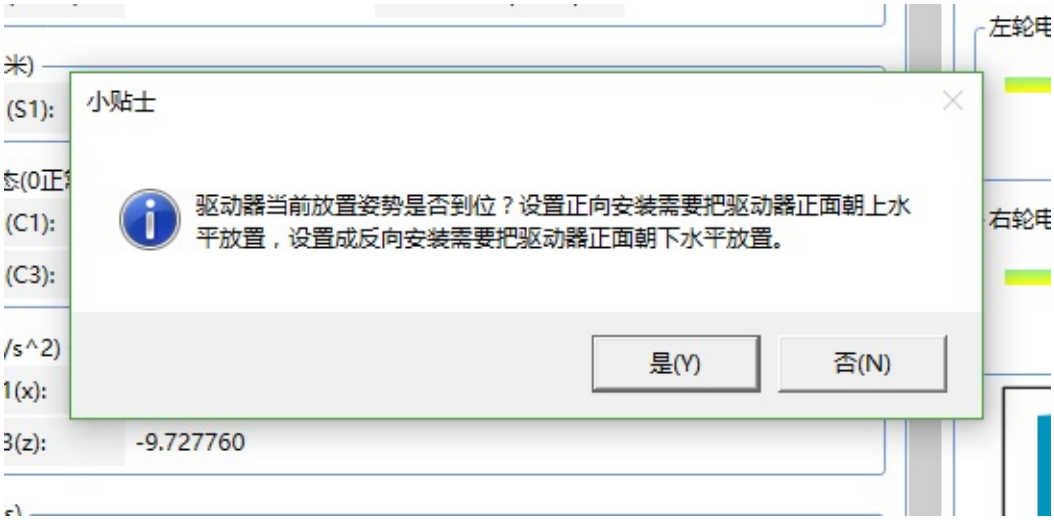
六、校准IMU

驱动器带IMU情况下才能执行校准过程。 首先根据“正向安装”设置，把驱动器水平放置。然后，点击“校准”使驱动器进入IMU校准程序。

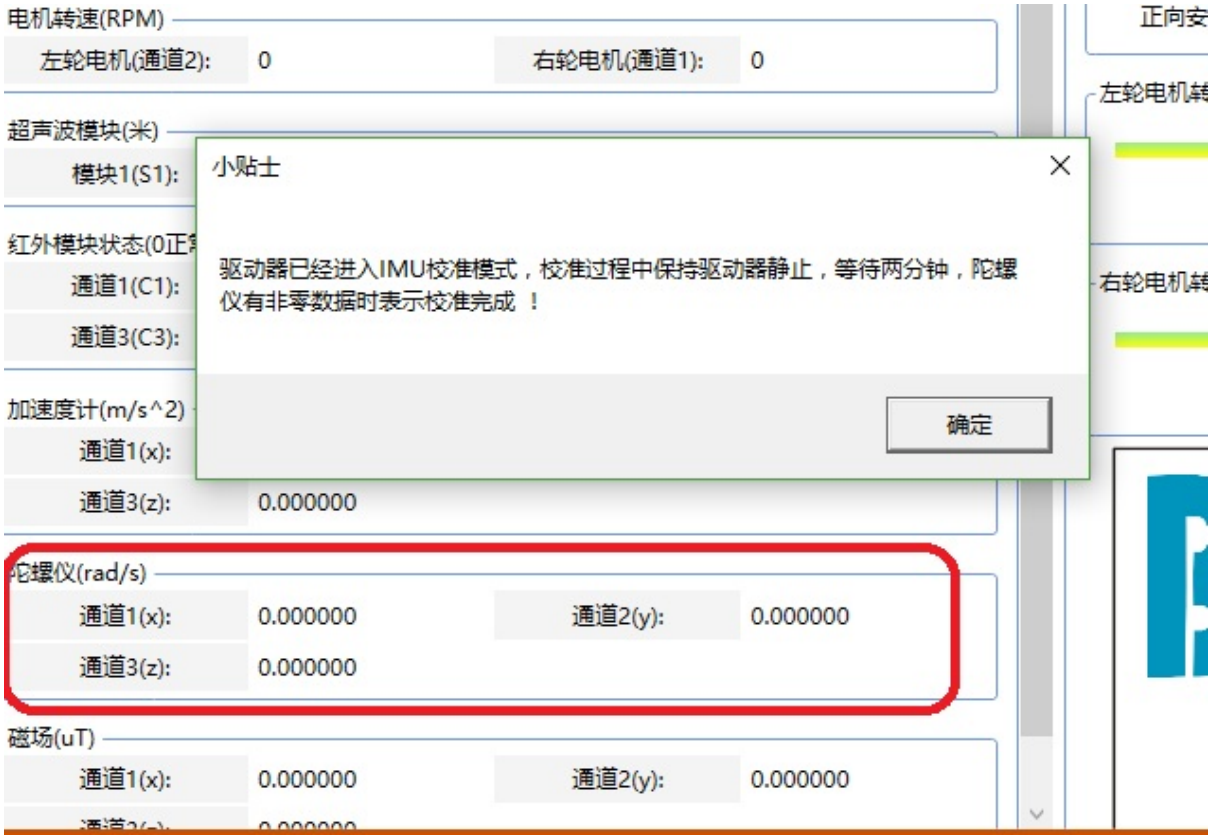
选择“是”



选择“是”，开始校准



校准开始后，陀螺仪数据会变成“0”



等待两分钟，校准结束后，陀螺仪数据会出现非零。校准成功后通道3数据值一般是小于0.004。

- ROS 驱动使用说明
  - 概述
  - 硬件连接
  - Ubuntu 主机串口权限设置和设备号映射
  - 软件包编译
  - 可执行节点 nodes
    - xqserial_server 节点
    - 订阅的话题数据
    - 发布的话题数据，50hz 频率
    - 节点参数
    - 发布的 tf 变换关系
    - 校准 IMU
    - 校准机器人参数

ROS 驱动使用说明

概述

ROS 底盘驱动 xqserial_server 包负责将订阅的/cmd_vel 话题进行差速解算后，转换成串口命令下方给电机驱动器。同时处理驱动器上传的串口包信息，然后整理发布成里程计和传感器数据话题。

硬件连接

电机驱动器和主机通过 USB 转 rs232 模块连接（对应“电机串口”），串口要通过 udev 规则进行 USB 端口绑定，映射成 ttyUSB001。串口参数是波特率为115200，8 个数据位，1 个停止位，无奇偶校验。

`驱动器的接线请根据驱动器手册来操作，在调试 xqserial_server 包之前需要保证用 mtools 软件能正常闭环控制电机的正反转，否则可能会烧毁驱动器。`

Ubuntu 主机串口权限设置和设备号映射

我们使用的是 usb 转 rs232 模块，ubuntu 默认会识别成 ttyUSB0 设备。这个可以通过 ls /dev 指令查询。识别出的设备号是随机的，为了将串口号映射成ttyUSB001，同时设置设备用户读写权限。我们可以通过 udev 规则来实现。

具体参考后面这篇帖子的步骤三， 自动充电功能包的使用和实现原理

软件包编译

涉及到 ros 工作空间、ros 包、catkin_make、ros 工作空间 source 指令、git 这些基础知识，如果不清楚这些请百度学习一下。

先 git clone 下载 xqserial_server 包放入 ros 工作空间 src 文件夹内，切换到lungu 分支，最后 catkin_make 编译。

假设你的 ros 工作空间在~/catkin_ws 内,下面是下载编译全部指令。

```
cd ~/catkin_ws/src/
git clone https://github.com/BluewhaleRobot/xqserial_server.git
cd xqserial_server
git checkout lungu
cd ~/catkin_ws/
catkin_make
```

可执行节点 nodes

catkin_make 编译后会生成可执行节点 xqserial_server，launch 文件夹中已经提供了一份可直接使用的launch 文件 xqserial.launch。script文件夹中的visualization.py 是一个 python 脚本文件，可以用来可视化 xqserial_server 发布的IMU 数据。

xqserial_server 节点

xqserial_server 节点是包主节点，涉及差速解算、里程计、传感器、tf 等数据的处理和发布。硬件上，这个节点会访问驱动器的串口设备。下文将详细介绍这个节点的 ROS 配置信息。

订阅的话题数据

话题	消息类型	描述
/cmd_vel	geometry_msgs/Twist	速度话题，订阅后解算成电机控制指令。
/imu_cal	std_msgs/Bool	IMU 自动标定指令，设为 true 会触发自动标定，此时机器人需要静止等待 2分钟，2 分钟内不能重复触发。
/globalMoveFlag	std_msgs/Bool	设为 true，/cmd_vel 指令生效；设为 false，则禁用/cmd_vel。
/barDetectFlag	std_msgs/Bool	设为 true 则使能超声波避障，设为 false 则关闭超声波和红外避障。

发布的话题数据，50hz 频率

话题	消息类型	描述
/xqserial_server/Odom	nav_msgs/Odometry	底盘里程计，角度部分已经融合了 IMU。
/xqserial_server/StatusFlag	std_msgs/Int32	值为 0 表示正在初始化，值为 1 表示正常。值为 2 则表示红外被触发。
/xqserial_server/Twist	geometry_msgs/Twist	底盘反馈的速度，线速度是编码器反馈值，角速度是 IMU 返回值。
/xqserial_server/Power	std_msgs/Float64	电池电压，单位为 V。
/xqserial_server/Pose2D	geometry_msgs/Pose2D	二维的 pose 数据，可以方便查看里程计坐标和角度。
/xqserial_server/IMU	sensor_msgs/Imu	IMU 话题数据，IMU 的 tf 关系需要自己根据下文在 launch 中设置调整。
/xqserial_server/Sonar1	sensor_msgs/Range	超声波模块 S1 话题数据，tf 关系需要自己根据下文在 launch 中设置调整。
/xqserial_server/Sonar2	sensor_msgs/Range	超声波模块 S2 话题数据，tf 关系需要自己根据下文在 launch 中设置调整。

节点参数

参数	类型	默认值	描述

~port	String	/dev/ttyUSB001	电机驱动器串口名字
~wheel_separation	double	0.36	两侧动力轮或履带中心间距，单位米
~wheel_radius	double	0.0825	动力轮或履带的半径，单位米
~max_speed	double	5.0	车轮最大转速或者履带动力轮最大转速，单位转每秒

发布的 tf 变换关系

odom→base_footprint 50hz 发布频率，base_footprint 在车正中心。

校准 IMU

把小车水平静止放置好，发布 ros 话题启动 IMU 自标定，2 分钟左右的标定过程种不能移动、碰撞小车。

```
rostopic pub /imu_cal std_msgs/Bool '{data: true}' -1
```

校准过程中，/xqserial_server/IMU 话题数据一直是零值，校准完成后恢复正常值，通过这个话题数据可以判断校准进度和校准结果。

校准机器人参数

执行下面操作之前需要先完成步骤 6 的 IMU 校准工作，因为 IMU 会影响里程计反馈精度。

校准轮半径参数 wheel_radius

校准原理：

卷尺可以测量小车实际前进距离(设为 L)，/xqserial_server/Pose2D 这个话题可以查看小车里程计反馈距离(设为 l)，这样我们可以得到 L/l 这个比值（校准之后的值应该为 1），这个比值就是 launch 文件中轮半径参数 wheel_radius 的缩放因子。用这个比值乘以现在的值可以得到校准后的值。

```
校准后 wheel_radius = 校准前 wheel_radius * L /l
```

校准步骤：

将车遥控到起始线，车角度要摆正对齐，然后重启 xqserial_server 包launch 文件，使里程计归零。遥控车直线前进 2 米左右，记录用卷尺测量的小车前进距离 L，和话题反馈值 l，计算 L/l 比值。根据校准原理得到校准后的 wheel_radius 值，将 launch 文件中的轮半径参数修改成这个新值，然后重启 launch 文件，半径就校准完成了。

校准轮间距参数 wheel_separation

注意：校准前需要完成轮半径的校准工作。

校准原理：

/xqserial_server/Twist 话题里面的角速度反馈值是通过 IMU 计算出来的，因此它可以作为一个基准值 W，/cmd_vel 话题里面的角速度值是设定值 w。计算得到的 W/w 比值，就是轮间距参数的缩放因子。这个值可以不用很精准，因为它只会影响角速度控制的精确性，不会影响里程计角度部分的精度。如果要获取精确值，推荐自己写个 python 节点订阅这两个话题，通过计算平均值来得到更精确的比值。

```
校准后 wheel_separation = 校准前 wheel_separation * W /w
```

校准步骤：

遥控车原地旋转，根据校准原理得到校准后的 wheel_separation 值， 将 launch 文件中的轮间距参数修改成这个新值，然后重启 launch 文件，轮间距 参数就校准完成了。