

## 目录

一、综述 .....	2
二、硬件介绍 .....	3
2.1 主控制器 .....	4
2.2 电机驱动介绍 .....	5
2.3 电源板介绍 .....	6
三、硬件安装 .....	7
3.1 主控和电机驱动安装 .....	7
3.2 电机安装 .....	10
3.3 电源板安装 .....	11
四、结构安装 .....	13
4.1 使用树莓派 .....	13
4.2 使用笔记本 .....	18
4.3 使用 TK1 .....	20
4.4 安装深度摄像头 .....	22
4.5 安装急停开关 .....	23
4.6 安装雷达 .....	23
五、环境搭建 .....	24
5.1 快递开箱 .....	24
5.2 硬件检查 .....	25
5.3 工作空间 .....	26
5.4 使用命令 .....	27
5.5 软件接口（进阶） .....	29
5.6 一些问题（重要） .....	33
六、进阶使用 .....	35
6.1 开源主从机工具（待续） .....	35
6.2 开源界面（待续） .....	37

# 一、综述

基于 ROS 开源系统构建，可实现室内建立地图和导航，本小车属于自主研发产品，可作为 ROS 的学习平台，不能达到消费级产品的稳定性导航效果，**需要有一定的 Linux 基础，ROS 基础，没有基础慎拍**，买家可在开源算法上继续优化以达到更好的效果。

底层采用自制的 Arduino2560 控制（2560+328 双控制芯片），遵循 iRobot 开源协议，暂不公开底层源代码，但完全公开通讯协议和底层驱动原理图。

小车基本兼容 Turtlebot 的基本移动命令、构建地图以及导航上位机程序，可直接用<ros by example>（hydro 和 indigo 两个版本）书中 rbx1 的例程，（在 Kinetic 下测试仍然可以使用，但仍建议使用 indigo，我们的测试环境为 Ubuntu14.04+indigo）

另外整个底层控制拥有独立于 ROS 之外的接口：超声波接口\*4，舵机控制接口\*2，一个 UART 接口，一个 IIC 接口，需要自行编写程序（可提供简单的 demo）。

上位机可以选择 TK1、RK3288、Odroid、树莓派 3 或者自配笔记本电脑（推荐 14 寸）。

提供小车配套的资料和学习历程，但不提供二次开发和技术指导，学习交流可以加入 ROS 机器人俱乐部:184903125，欢迎一起交流学习！

## 二、硬件介绍

净重：3KG/5KG/16KG/24KG

负载能力：5kg/10KG/50KG/100KG

车轮数量：2-6（主动轮 2/4 个，从动轮 2/4 个）

驱动方式：差分驱动

精度：厘米级别精度误差

速度：0~1.0m/s

电源输入：DC-12V/2A

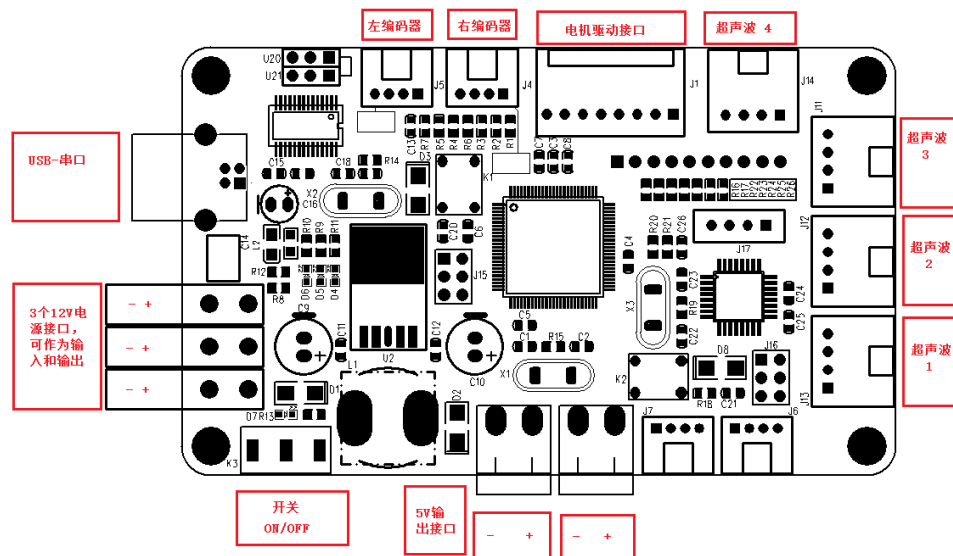
电池容量：9AH-20AH 锂电池/蓄电池

续航时间：续航 3-8h

通信频率：57600/115200

可用 ROS M-Robot 软件包，亦兼容支持 turtlebot rbx1 软件包，支持建地图导航功能。

## 2.1 主控制器



主控资源：

主控芯片：Atmea 2560

超声波控制芯片：Atmega 328P

USB-TTL：FT232

电源：LM2596-5V

板载接口：

电源输入接口：3 个 5557-1P 端口座，输入电压 7~12V

5V 输出接口：2 个绿色端子座，输出 5V/2.5A

开关：12V 电源输入开关

USB 串口：FT232 芯片，USB-Type B 座

编码器输入接口：两个编码器输入接口，适用于光电和霍尔编码器。PH-2.0-

4P 端子座

电机驱动控制接口：XH-2.54-8P 端子座或排座

超声波接口：4 个接口，XH-2.54-4P 端子座

舵机接口：适用于 5V 舵机，排针

遥控接口：点动式遥控器接口，排座

预留接口：

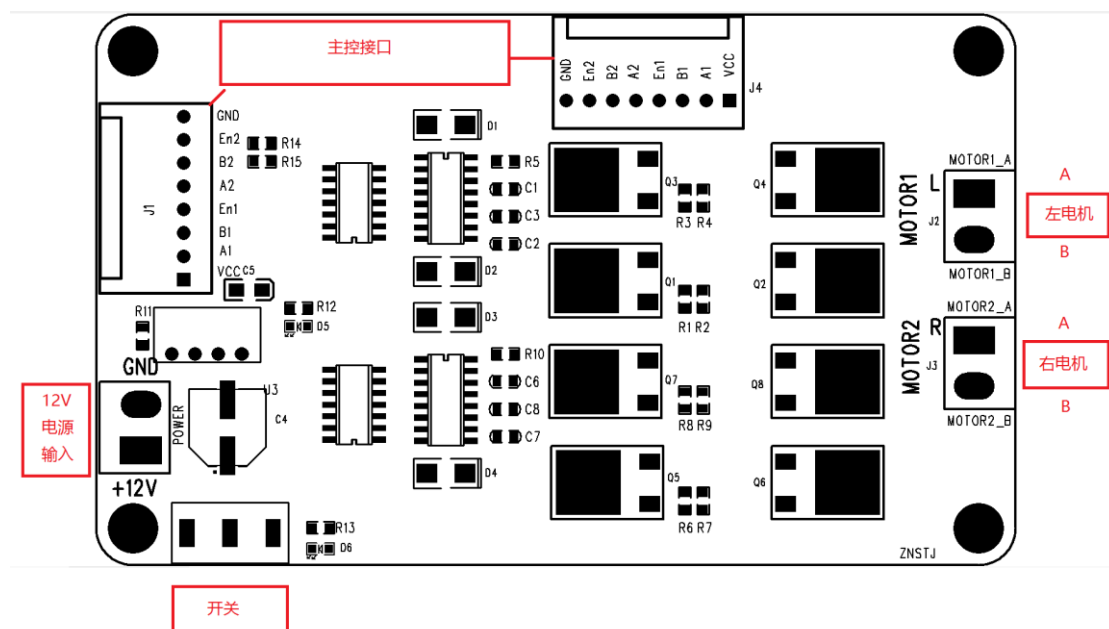
UART：PH-2.0-4P 端子座

IIC：PH-2.0-4P 端子座

下载固件接口：ATMEL-ISP\_6P 标准接口，分别为 ATMEGA 2560 和 ATMEGA 328P 接口，共两个

## 2.2 电机驱动介绍

M-Robot\_V2.0 电机驱动板适用于直流电机驱动，理论电流 100A，控制大功率电机，需加散热器。实物图如下：



接口：

电源输入接口：电机电源，7-12V

电机控制端口：

电机接口：蓝色端子座

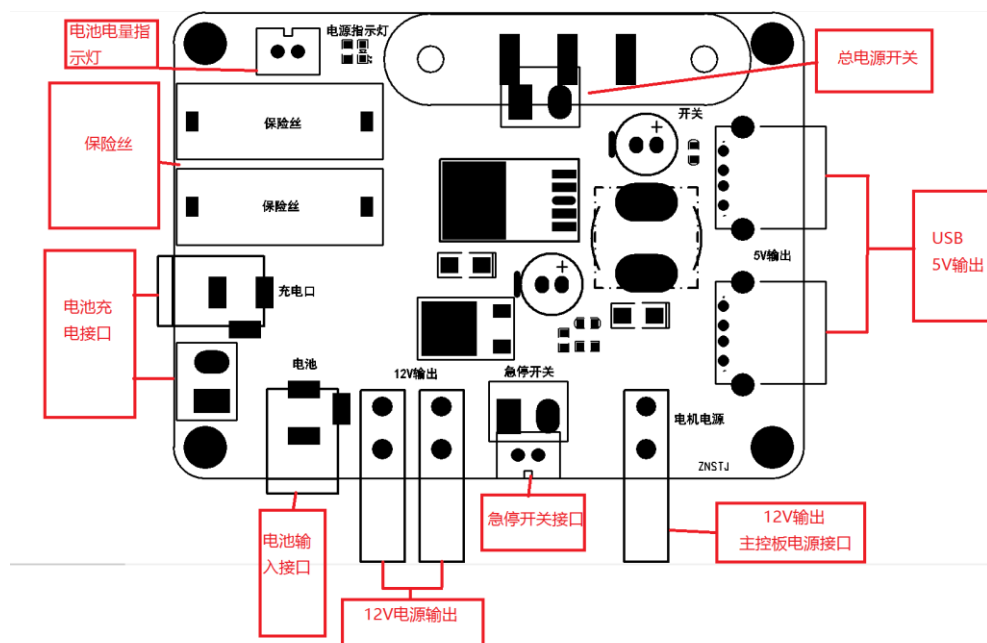
开关：总电源开关

使用说明：VCC：5V

引脚（x=1,2）	实验 1	实验 2	实验 3
ENx	1	1	0
Ax	PWM_A	0	0/PWM_A
Bx	0	PWM_B	PWM_B/0
现象	正转	反转	电机静止

## 2.3 电源板介绍

M-Robot\_Power-V1.1 电源板是配合 M-Robot 的电源控制板 ,适用于锂电和蓄电。



资源：

电源：LM2596-5V

接口：

电池输入接口：DC-005

电池充电口：DC-005

保险丝：10A，两个

电量指示模块接口：XH-2.54-2P

总电源开关：绿色端子座 5.0-2P

急停开关接口：XH-2.54-2P 端子座

12V 主控接口：5557 端子座（急停开关控制）

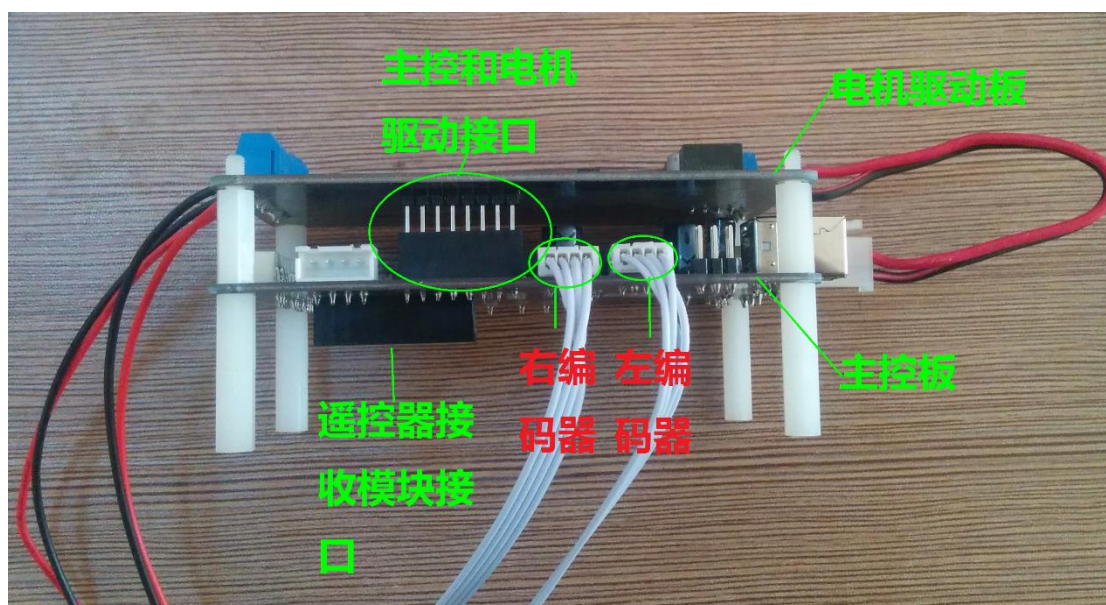
12V 输出：两个 5557 端子座（不受急停开关控制）

5V 输出：USB-A 座

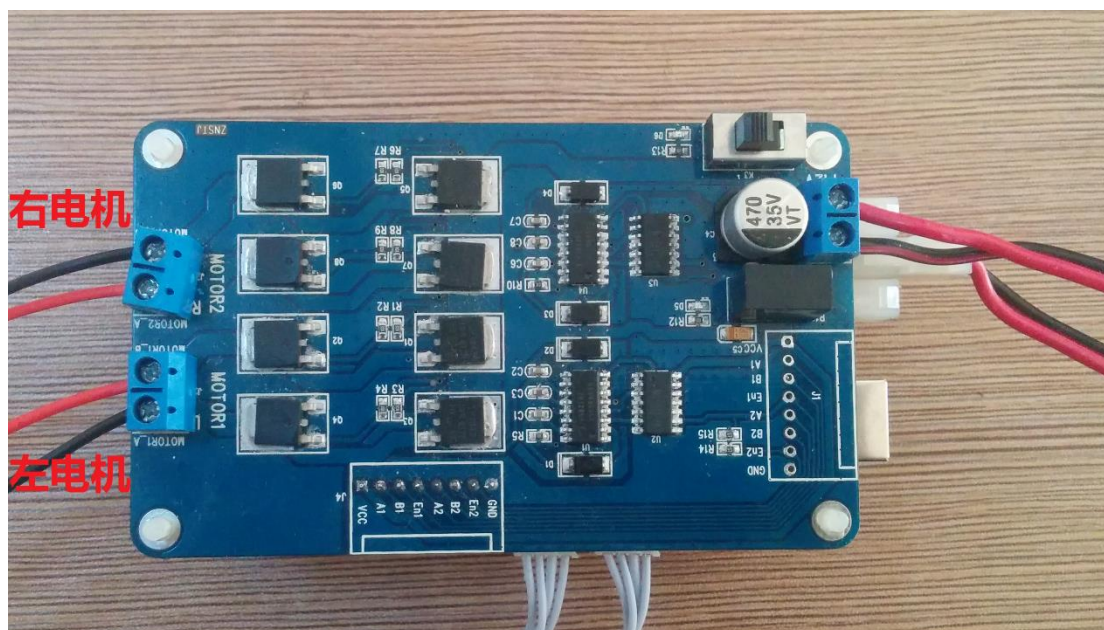
## 三、硬件安装

### 3.1 主控和电机驱动安装

如图，电机驱动板置于主控板之上，主控板和电机驱动板的四个角的安装孔位，用 M3\*15（长）mm 的塑料柱固定，上端用 M3 的塑料螺帽，主控下端用 4 个 M3\*25mm 的塑料螺柱固定到车的底盘。







### 3.1.1 主控供电

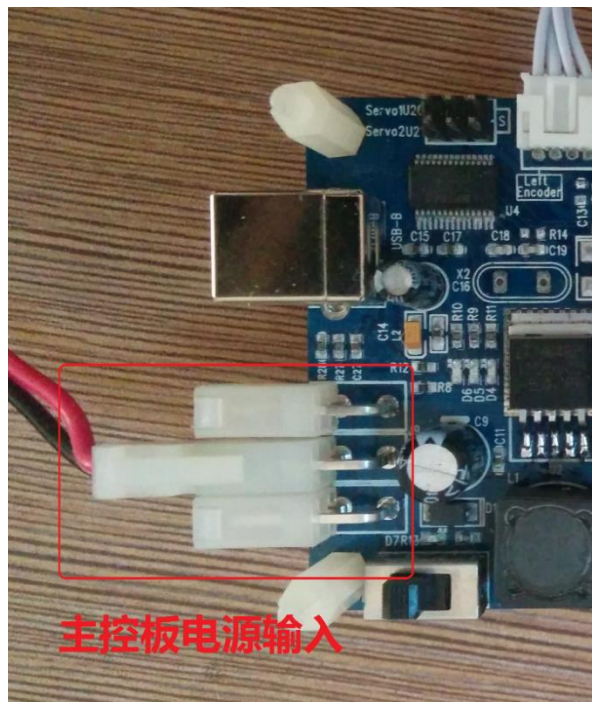
参考 [M-Robot\\_V1.2 主控板说明](#)，主控板有 3 个并行的 5557-1P 端口座，主控电源输入是双头均为 5557-1P 的线，从电源板的主控电源接口连接到主控电源输入口。

电源板：



主控板：





### 3.1.2 电机驱动板供电

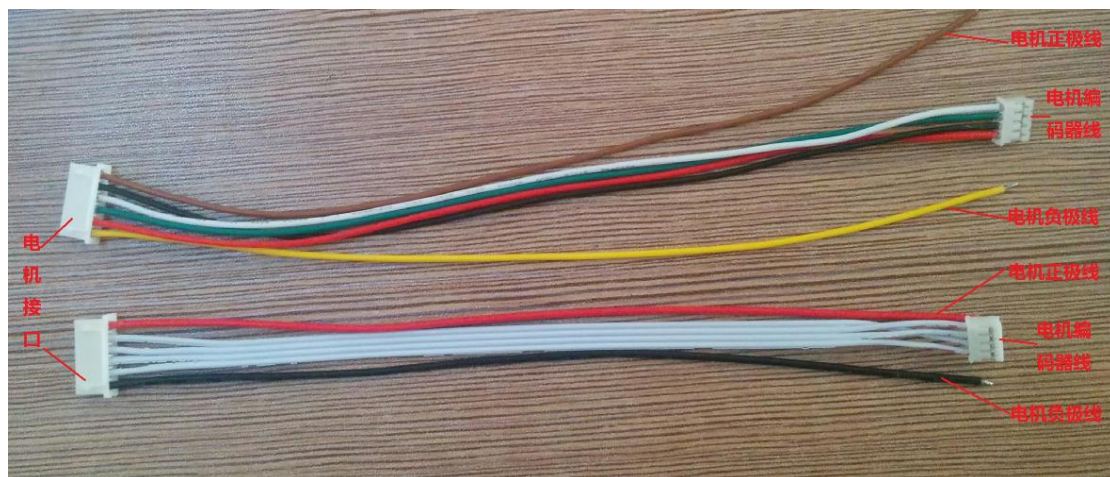
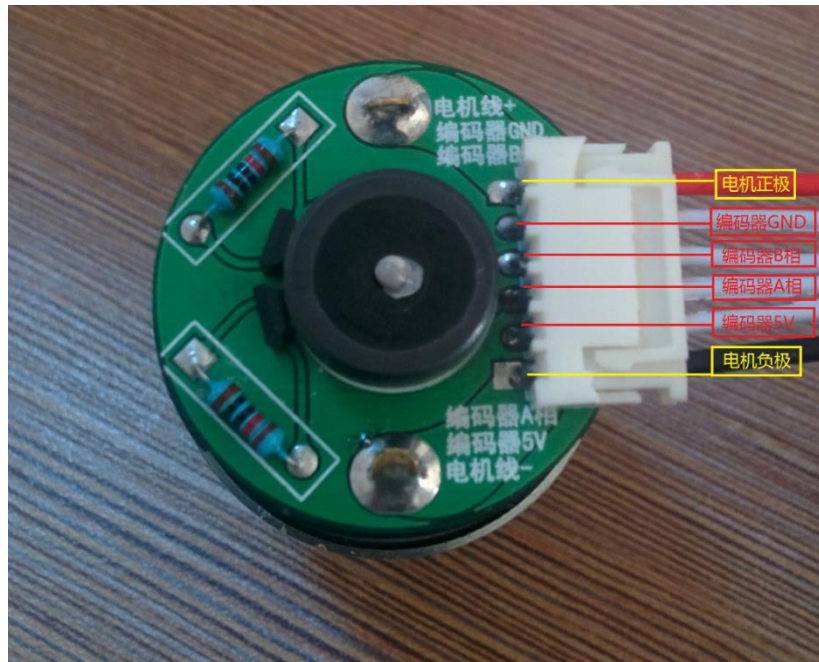
红正黑负



接入主控时直接入 5557 端子座，即可。

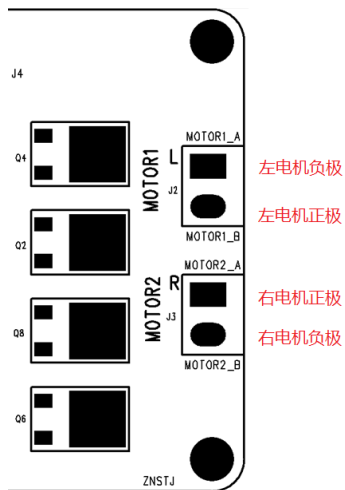
## 3.2 电机安装

### 3.2.1 电机接线



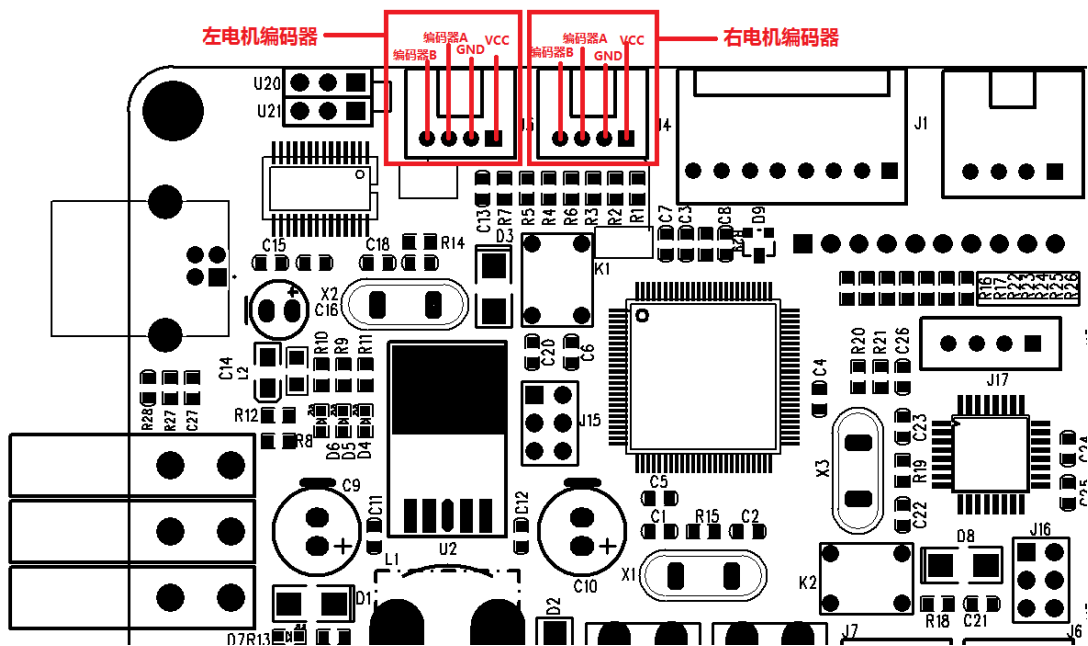
电机供电方式：

电机驱动板的两个蓝色端子座，接线方式如下



### 3.2.2 电机编码器接线

主控板上 PH-2.0 白色端子座为电机编码器端子，接线方式如下



### 3.3 电源板安装

电源板用 M3\*20mm 的塑料螺柱和 M3 的螺帽螺丝固定到小车底盘上。

电源板参考 [M-Robot\\_Power-V1.1 电源板说明](#)。

在电源板电池输入口位置插上电池。

### 3.3.1 充电接入充电口



其中保险丝安装两个 10A 保险丝。

其中 12V 电源输出：给 TK-1 等供电。

其中 USB 电源输出：给树莓派等供电。

### 3.3.2 急停开关

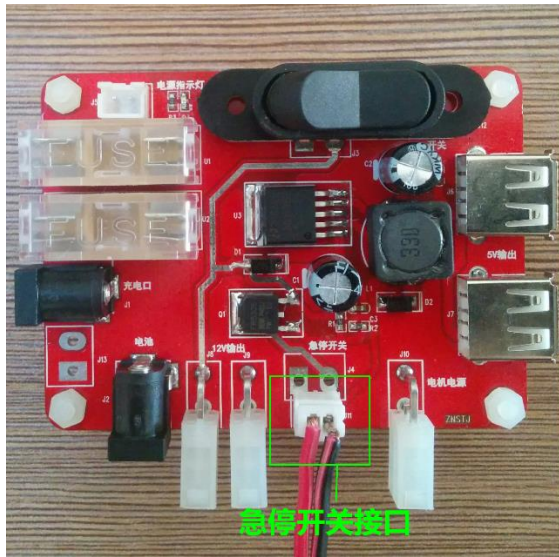
急停开关线 20cm 左右，一端问 XH-2.54-2PIN 接口，另一端为插簧冷压端子座。

急停开关端接线如下：





电源板端接线如下：

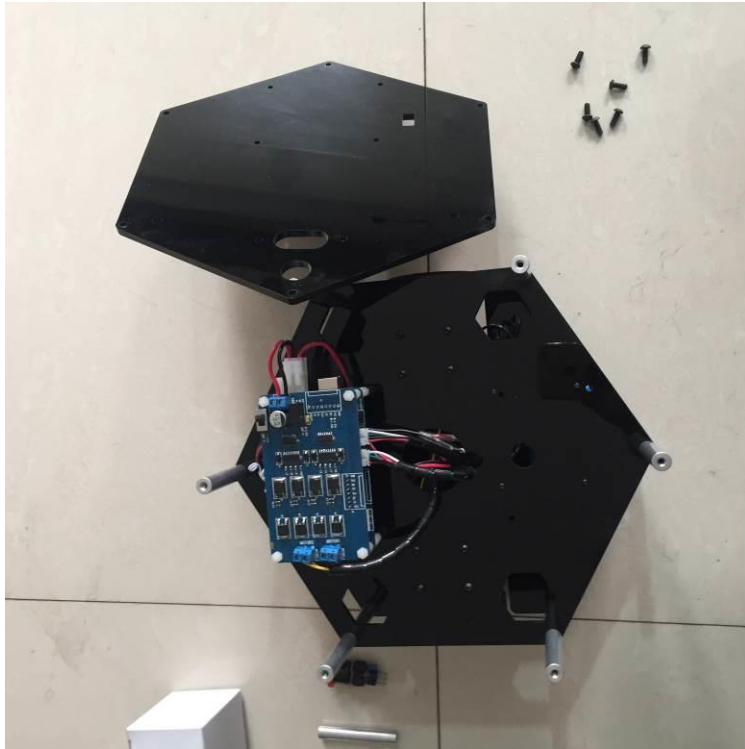


## 四、结构安装

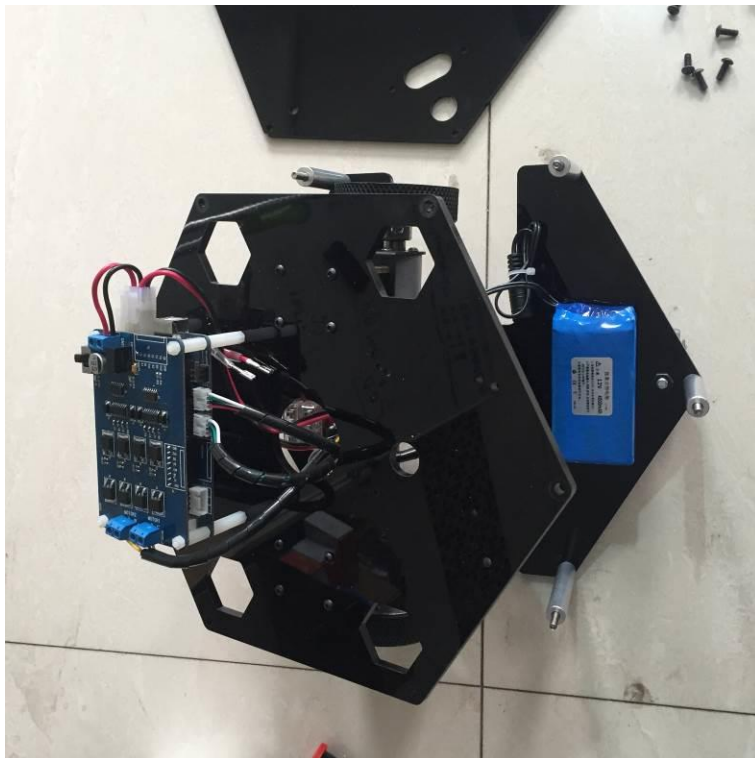
因为发出之前已经安装好了，所以可以不用管。

### 4.1 使用树莓派

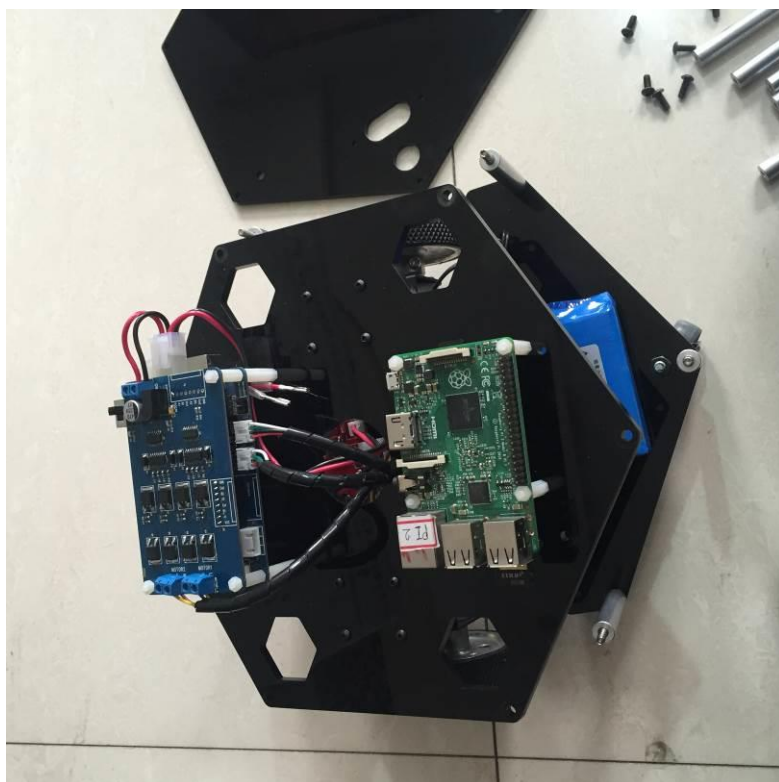
首先，我们拆开下图中的最上层，如下：



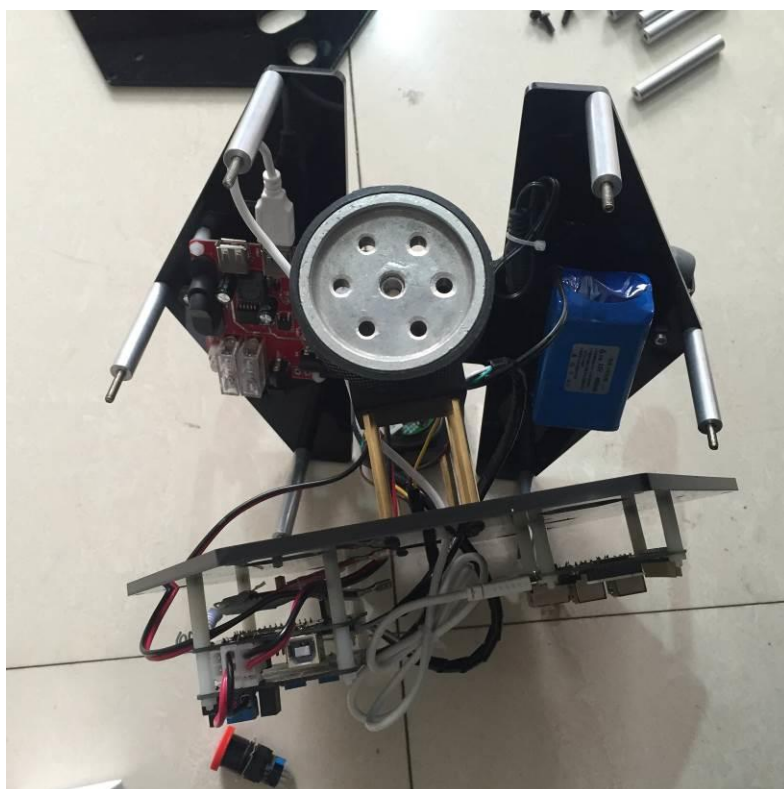
接下来是拆开第二层，如下



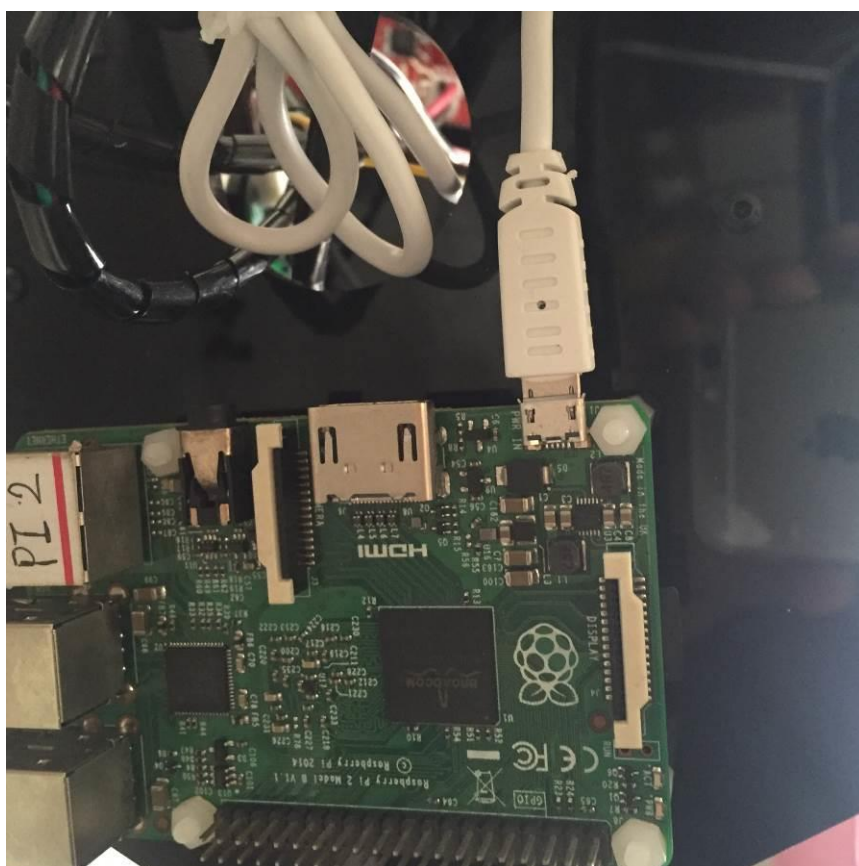
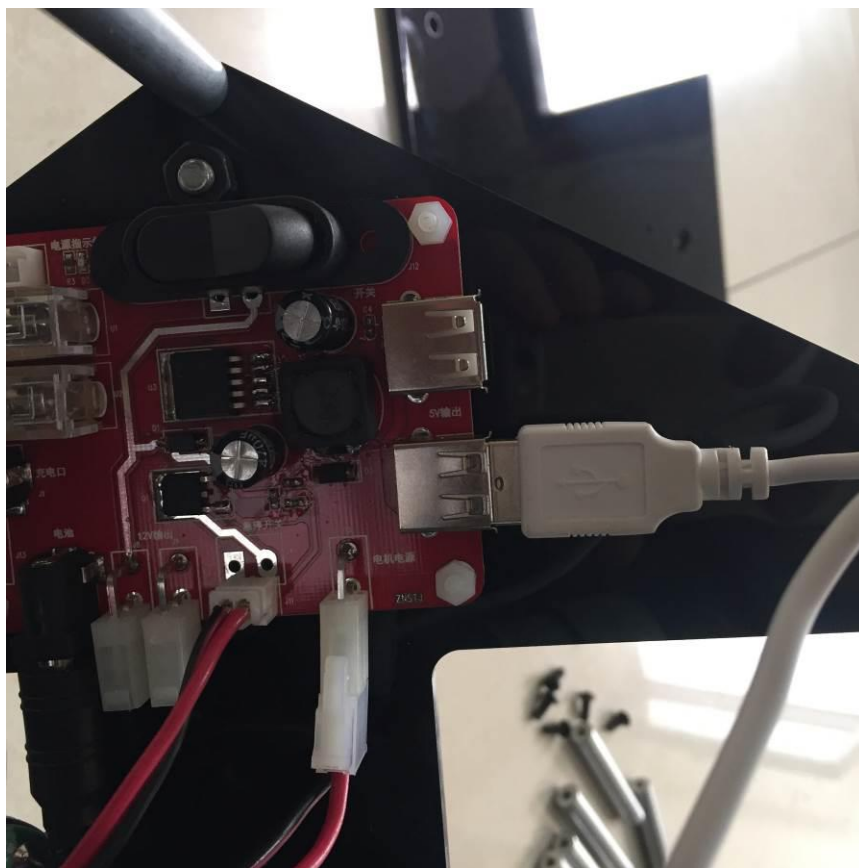
然后把树莓派装到刚才拆开的第二层上，如下：



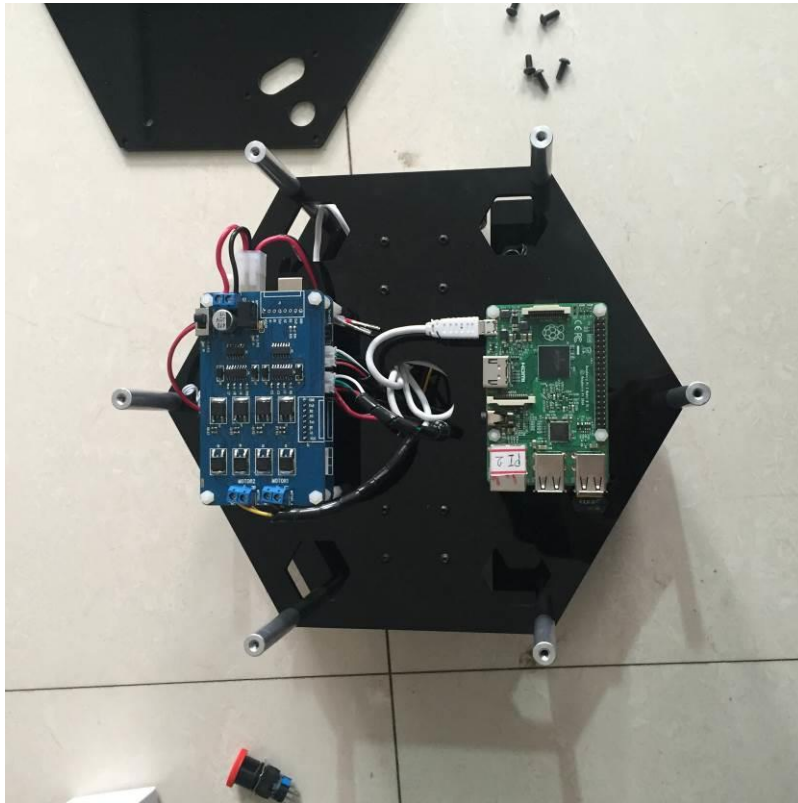
接着，插好树莓派的供电线，如下：



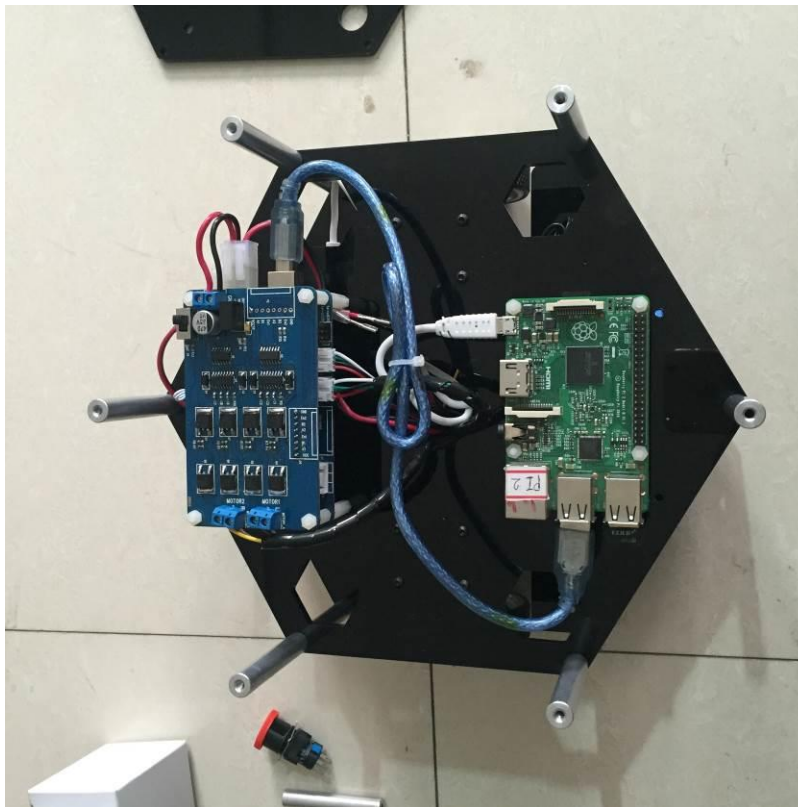




然后原样再把二层装好，如下：



接着用 USB 线把主控和树莓派连好，如下：



## 4.2 使用笔记本

首先需要把摄像头装在支架上（注意支架的方向），然后直接固定在板子上，如下：



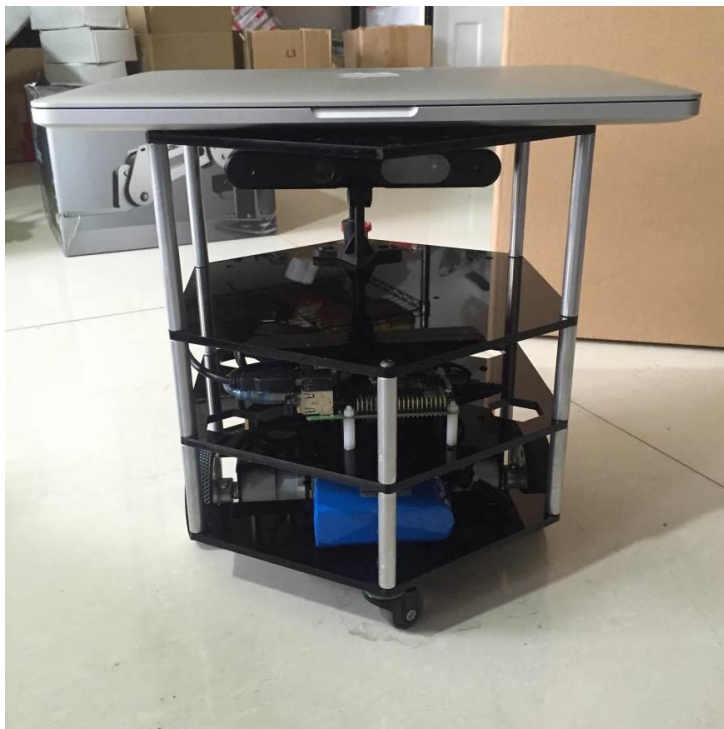
接着，把四根 10cm 长的铝柱固定在这一层上，如下：



最后将放电脑的板子固定在这四根铝柱上，如下图：



把笔记本电脑放到最上面这层板子上，就可以使用了。

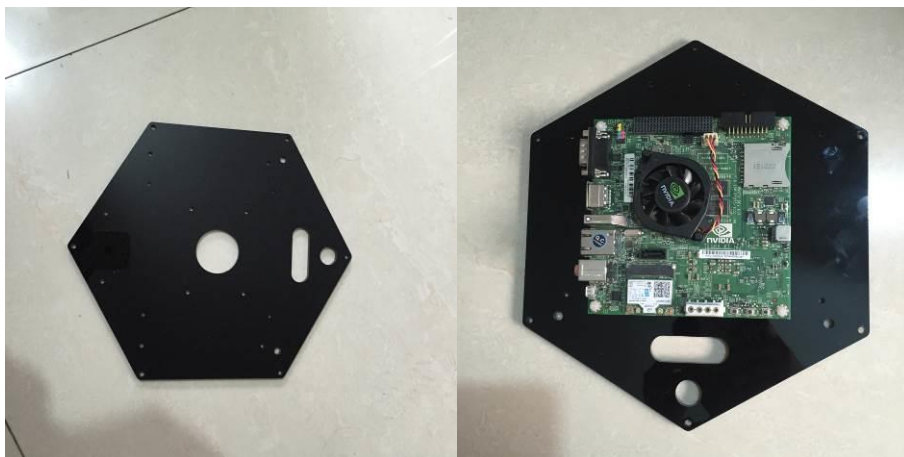






#### 4.3 使用 TK1

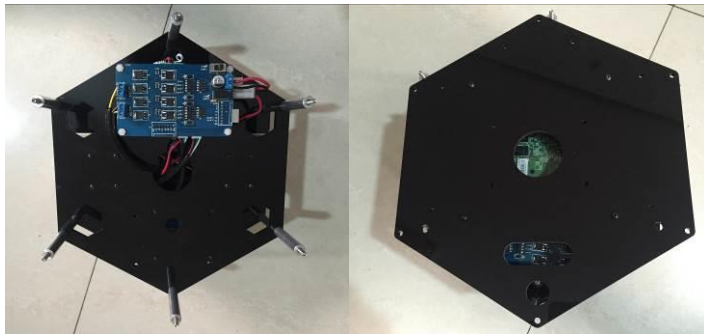
把 TK1 固定到下面这块板子上：



然后把它当做小车的第三层固定好：



把摄像头（或者雷达）、急停开关、TK1 天线装在第四层板上，由于 TK1 上的 USB 口不多，所以需要配合 Hub（带电源的 USB-HUB）使用，Hub 可以放在第二层，线可以从第三层中间的孔穿过：



## 4.4 安装深度摄像头

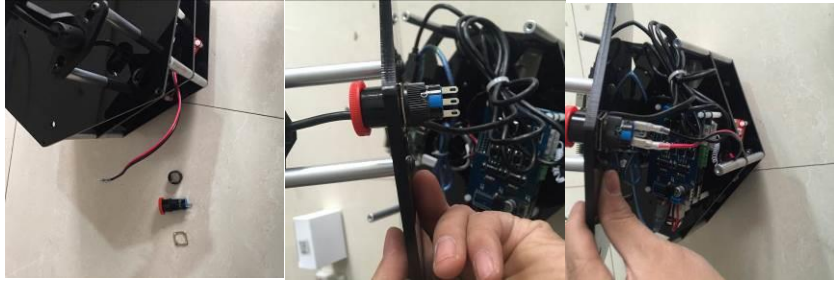
如果用的是深度摄像头，先把摄像头跟支架组装起来（**一定要注意支架的方向**），在最上层装好，然后把 USB 线插好，如下：





## 4.5 安装急停开关

急停开关插左右两个 1 号、3 号位置，空开中间 2 号位置，如下：



装上最上面一层板：



## 4.6 安装雷达

如果你用的是雷达，同样需要提前安装好，USB 线也插好，然后再装好最上



小车到这里就全部装完了。

## 五、环境搭建

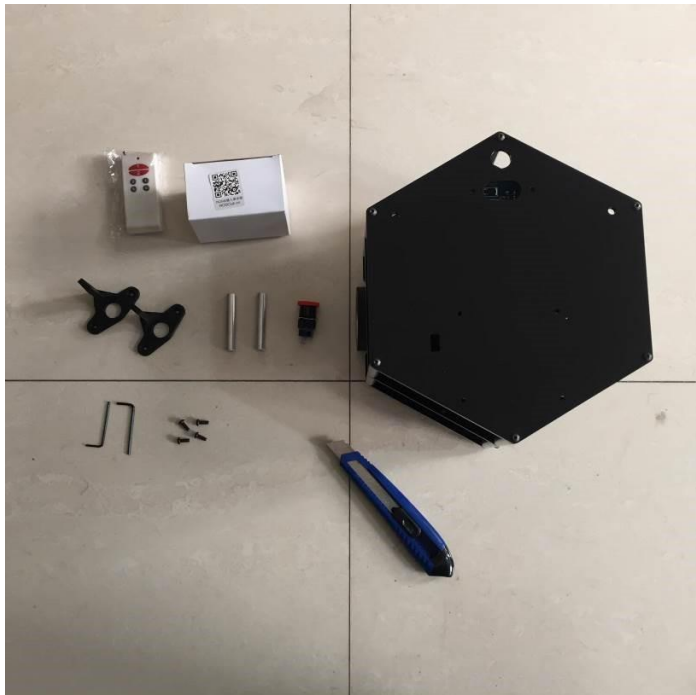
### 5.1 快递开箱

收到快递后，需要快递单面朝上打开，打开后，你会看到下面图片中的样子：



然后挨个取出摆好箱子里的东西，并拆开个别小包装，所有东西如下

根据不同套餐，箱内有所改变，以收到货为准，有问题咨询 [dajianli](#)。



PS：刀子不是。

## 5.2 硬件检查

( 1 ) **确保电池有电** , 打开主控电源开关 ( 底部黑色船型开关 ) , 电源板指示灯亮 , 表示正常。

( 2 ) **确认急停开关弹起** , 主控板指示灯亮。

若急停开关按下 , 主控板和电机驱动断电。

( 3 ) **遥控器测试小车**

1 : 前进

2 : 后退

3 : 左转

4 : 右转

5 : 减速

6 : 加速

测试小车没问题后 , 方能进行下一步。

( 4 ) **连接上位机** ( 笔记本、TK1、RK3288 等 )

USB-B 线将主控板和电脑主机 ( TK1 , 树莓派或其他主板 ) 连接 , 打开电脑主机。

电脑主机环境 : **Ubuntu 14.04 + ROS indigo**

现阶段建议用这个配套版本 , 有经验者可以尝试 ROS Kinetic。

( 5 ) **安装 mrobot 所需的包。**

安装 Turtlebot 包

直接安装依赖包 :

```
sudo apt-get install ros-indigo-turtlebot ros-indigo-turtlebot-apps
```

ros-indigo-turtlebot-interactions ros-indigo-turtlebot-simulator ros-indigo-kobuki-ftdi ros-indigo-rocon-remocon ros-indigo-rocon-qt-library ros-indigo-ar-track-alvar-msgs

也可以源码安装，参考 ROS wiki：

<http://wiki.ros.org/turtlebot/Tutorials/indigo/Turtlebot%20Installation>

## ( 6 ) 硬件配置

在用户目录的.bashrc 文件下添加配置：

```
export TURTLEBOT_BASE=create
```

```
export TURTLEBOT_STACKS=circles
```

```
export TURTLEBOT_3D_SENSOR=asus_xtion_pro    (alternatively  
kinect)
```

```
export TURTLEBOT_SERIAL_PORT=/dev/ttyUSB0    (may appear under  
another ttyUSBn)、
```

## 5.3 工作空间

确保 ROS 安装完成并且配置正确，并在.bashrc 中添加

```
source /opt/ros/indigo/setup.bash
```

建立 catkin\_ws 工作空间

```
mkdir -p ~/catkin_ws/src
```

```
cd ~/catkin_ws/src
```

```
catkin_init_workspace
```

编译

```
cd ~/catkin_ws/
```

```
catkin_make
```

source devel/setup.bash 或者在.barcsh 中添加

```
source /home/youruser/catkin_ws/devel/setup.bash
```

安装 mrobot 包

进入工作空间目录 cd ~/catkin\_ws/src

Clone mrobot 包：

```
git clone https://github.com/ROSClub/mrobot-indigo
```

使用方法见包内 README 文件

编译：

```
cd ~/catkin_ws
```

```
catkin_make
```

注：亦可以下载 rbx1 包，同 mrobot 包兼容：

```
git clone -b indigo-devel https://github.com/pirobot/rbx1
```

测试小车

## 5.4 使用命令

确保以上代码编译成功

### 5.4.1 移动测试

启动底盘：roslaunch mrobot\_bringup mrobot.launch

启动键盘遥控：roslaunch mrobot\_nav keyboard\_teleop.launch

按照屏幕提示按键，小车移动。

### 5.4.2 建图测试

( 1 ) 启动底盘：

```
roslaunch mrobot_bringup mrobot.launch
```

( 2 ) 启动激光雷达或深度摄像头：

深度摄像头 XtionPro：

```
roslaunch mrobot_bringup turtlebot_fake_laser_openni2.launch
```

或 RPLIDAR 激光雷达：

```
roslaunch rplidar_ros rplidar.launch
```

注：请将 rplidar.launch 中的 ttyUSB0 改为 ttyUSB1，并先插入底盘并启动底盘再启动 RPLIDAR。

( 3 ) 启动 gmapping：

```
roslaunch mrobot_nav gmapping_demo.launch
```

( 4 ) 启动 rviz：

```
roslaunch rviz rviz -d `rospack find mrobot_nav`/gmapping.rviz
```

( 5 ) 启动键盘遥控：

```
roslaunch mrobot_nav keyboard_teleop.launch
```

建好地图可以关闭键盘直接点 2D Nav Goal 导航测试

( 6 ) 保存地图：

```
roscd mrobot_nav/maps
```

```
roslaunch map_server map_saver -f your_map
```

## 5.5.2 导航测试

( 1 ) 启动底盘 :

```
roslaunch mrobot_bringup mrobot.launch
```

( 2 ) 启动激光雷达或深度摄像头 :

深度摄像头 XtionPro : `roslaunch mrobot_bringup`

`turtlebot_fake_laser_openni2.launch`

或 RPLIDAR 激光雷达 : `roslaunch rplidar_ros rplidar.launch`

注 : 请将 `rplidar.launch` 中的 `ttyUSB0` 改为 `ttyUSB1` , 并先插入底盘并启动底盘再启动 RPLIDAR。

( 3 ) 加载地图 :

```
roslaunch mrobot_nav tb_demo_amcl.launch map:=your_map.yaml
```

( 4 ) 运行 rviz :

```
roslaunch mrobot_nav nav_test.launch
```

## 5.5 软件接口（进阶使用）

通讯频率 : 57600

数据位 : 8

校验位 : None

停止位 : 1

**协议发送格式 : [操作数] ( n 位数据 )**

**[操作数][功能码]**

**协议返回格式 : [报头] ( n 位数据 )**



或直接（n 位数据）

注：其中-[操作数]，[报头]，[功能码]均为单字节数据。

### 5.5.1 发送控制速度命令

[操作数][RightSpeedHighByte][RightSpeedLowByte][LeftSpeedHighByte]

[LeftSpeedLowByte]

**不返回数据**

[操作数]：**0x91**

[RightSpeedHighByte]：右轮速度高字节数据

[RightSpeedLowByte]：右轮速度低字节数据

[LeftSpeedHighByte]：左轮速度高字节数据

[LeftSpeedLowByte]：左轮速度低字节数据

右轮速度和左轮速度单位均为 mm/s,正数代表正转，负数代表反转。

例如：

控制右轮速度为-100mm/s([0xFF9C])，左轮速度为 200mm/s([0x00C8])，发

送数据如下：

[0x91][0xFF][0x9C][0x00][0xC8]

**注：右轮和左轮速度限制范围为（-500mm/s-500mm/s）**

**注：直行请使两轮速度一致。原地右转例如：左轮速度 100，右轮-100，将顺时针旋转。**

### 5.5.2 读取最近一次底盘行走距离和角度数据

**[操作数][数据段]**

**[操作数] : 0x8E**

**[数据段] : 0x02**

**数据返回 :**

**[数据 1][数据 2][距离高位数据][距离低位数据][角度高位数据][角度低位数据]**

注：其中数据 1，和数据 2 为预留功能，可以忽略。行驶距离为 int 型数据，正向为正数，反向为负数，单位为 mm。行驶角度为 int 型数据，正向为逆时针正数，反向为顺时针负数，单位为度。请务必在数据溢出之前读取，否则将覆盖数据。并且发送读取命令后，行驶距离和角度都将清零。

例如：读取行驶距离和角度数据发送如下

[0x8E][0x91]

返回数据如下：

[0xFF][0x00][0xFF][0xAD][0x00][0x5C]

行驶距离为：[0xFF][0xFD]=-173mm

行驶角度为：[0x00][0x5C]=92 度（角度制）

注：附加

（1）简易行驶距离读取命令（用于仅直行情况下，读取后距离和角度数据都会清零）

**发送：[0xA0]**

**返回：[0xA0][距离高位数据][距离低位数据]**

（2）简易行驶角度读取命令（用于仅原地旋转情况下，读取后距离和角度数据

都会清零 )

**发送 : [0xA1]**

**返回 : [0xA1][角度高位数据][角度低位数据]**

### 5.5.3 读取超声波距离传感器数据

接口 : 4 个 , SR04-(1~4)

更新频率 : 1HZ

有效范围 : 300cm 以上 , 超出距离返回值为 0

返回距离 : int 型数据 , 由两个字节组成 , 高位数据在前 , 低位数据在后。单位为 cm

**发送指令 : [操作数][n]**

**[操作数] :      0x9F**

**[n] :            为超声波通道选择。**

**n=0x01 : 读取第一通道超声波数据**

**n=0x02 : 读取第二通道超声波数据**

**n=0x03 : 读取第三通道超声波数据**

**n=0x04 : 读取第四通道超声波数据**

**n=0xFF : 读取所有通道超声波数据**

数据返回 : n=1~4 时: ( 读取 1~4 通道数据 )

**[报头][n 通道高位数据][n 通道低位数据]**

**n=0xFF 时: ( 读取全部数据 )**

**[报头][1 通道高位数据][1 通道低位数据][2 通道高位数据][2 通道低位数据][3**

**通道高位数据][3 通道低位数据][4 通道高位数据][4 通道低位数据]**

**[报头]：0x9F**

数据：为 int 型数据，由高位字节和低位字节组成，数据单位为 cm

例如：

读取第二通道数据:

发送：[0x9F][0x02]

返回：[0x9F][0x00][0x34]

返回距离为：0x0034=52cm

读取全部通道数据：

发送：[0x9F][0xFF]

返回：[0x9F][0x00][0x02][0x00][0x52][0x00][0x00][0x01][0x05]

返回距离为：第一通道 0x0002=2cm，第二通道 0x0052=82cm，第三通道 0x0000 距离超出范围或超声波被遮挡住或此位置没有超声波传感器，第四通道 0x0105=261cm。

**注：本机器默认为前后两个超声波传感器，前面为第一通道，后面为第二通道，用户可以自行添加最多至 4 个超声波传感器。**

## **5.6 一些问题（重要）**

（1）为防止冲突，串口通信发速度控制 M-Robot 时，当两轮速度不为 0 时，遥控禁止，若需要使能遥控，串口发送两轮速度为零或者重新开机。

（2）超声波传感器数据是独立的，程序没有设置避障，上位机可以读取数据自行控制。

( 3 ) 充电时请勿开机

( 4 ) 遥控器使用说明 :

1 前进、2 后退、3 左转、4 右转、5 减速、6 加速

默认 100mm/s ,

减速每次减 10 , 最低至 60mm/s

加速每次加 10 , 最大至 150mm/s

重启恢复默认值

( 5 ) 请保持负重状态下 , 车的重心于底盘正中心

( 6 ) 使用多个串口解决办法

如果有插其他的 USB 串口 , 请确保主控硬件是 ttyUSB0 , 如果不是 , 把  
TURTLEBOT\_SERIAL\_PORT 改为主控相应的端口。

例如添加 RPLIDAR 雷达。

( 7 ) 串口权限

如果即 PC 端没有串口访问权限或者 USB 线没有插上 , 出现[ERROR]  
[WallTime: 1383454018.987015] Failed to open port /dev/ttyUSB0.  
Please make sure the Create cable is plugged into the computer.

解决方法 :

例如当前用户为 mrobot :

查看 USB 设备 : ls -l /dev/ttyUSB\*

输出 : crw-rw---- 1 root dialout 188, 0 Nov 2 10:31 /dev/ttyUSB0

查看用户组 : groups

输出 : mrobot adm cdrom sudo dip plugdev lpadmin sambashare 表示

用户不在 dialout 用户组内。

添加到用户组：sudo gpasswd -a mrobot dialout

(mrobot 替换为自己的用户名)

重启后生效。

(8) 如果编译出现问题，权限不够

请到 mrobot-indigo\mrobot\_nav\cfg 文件夹下，执行：

chmod +777 CalibrateAngular.cfg 和 chmod +777 CalibrateLinear.cfg

## 六、进阶使用

### 6.1 开源主从机工具（待续）

#### 6.1.1 运行

在 github 上下载 ROSControlPanel 代码库

(项目地址：<https://github.com/ROSClub/ROSControlPanel.git>)

在终端中进入解压后目录中的 tools 文件夹，输入 python tool.py 运行一键配置工具，或在文件管理器中进入 tools 文件夹中 bin 目录，双击 tool 允许一键配置工具，运行后界面如下：



Figure 1 一键配置 ROS 主从机工具界面

### 6.1.2 使用

界面运行时，默认勾选**主机为本机**选项，若想将主机设置在远程设备上，则取消勾选该选项。

在空白的主（从）机 IP 地址栏填写远程设备 IP 地址，并在右侧填写好远程设备的 SSH 登陆的用户名和密码，点击保存按钮。

当界面下部出现了修改好的配置文件内容，同时弹出提示框显示“主从机配置已完成”，整个主从机配置过程就完全完成了；

如果弹出提示框，显示“远程设备无法连接!”，请检查远程设备是否正常运行或是否能在网络内访问。

点击左上角关闭按键可以退出该一键配置工具。



### 6.1.3 测试

在配置好的主机上面（若主机是远程设备，需要先 SSH 登录）开启新终端，运行 `roscore` 命令。

在远程设备上通过 SSH 连接后，输入

`roslaunch rbx1_bringup turtlebot_minimal_create.launch` 在主机上重新打开一个新终端，输入

`roslaunch mrobot_nav keyboard_teleop.launch`

用键盘 `i` , `j` , `k` , `l` 和逗号分别控制小车前进，左转，停止，右转，后退进行测试，如果小车能够正常运行，则主从机配置成功。

## 6.2 开源界面（待续）

在 github 上下载 ROSControlPanel 代码库

(项目地址：<https://github.com/ROSclub/ROSControlPanel.git>)

### 6.2.1 启动软件

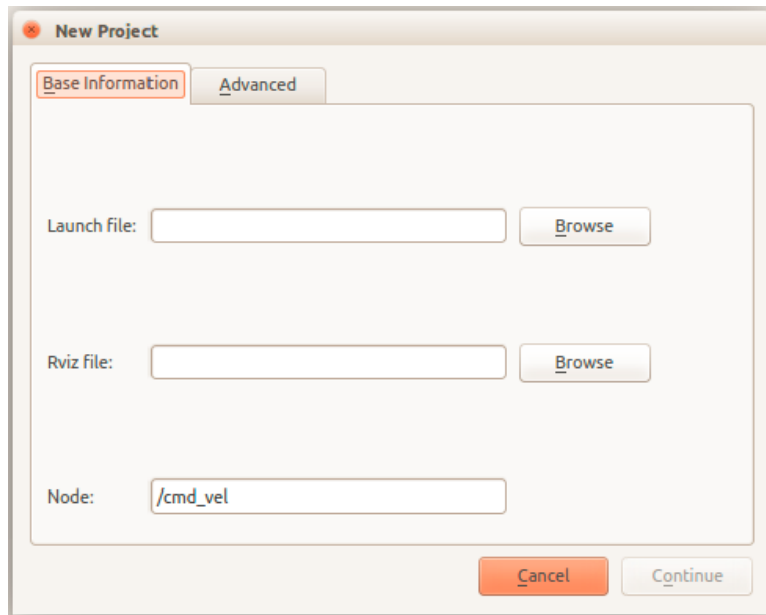
在命令行中切换到 `ControlPanel.py` 所在目录，输入 `./ControlPanel.py` 或 `python ControlPanel.py` 即可启动软件

### 6.2.2 新建项目

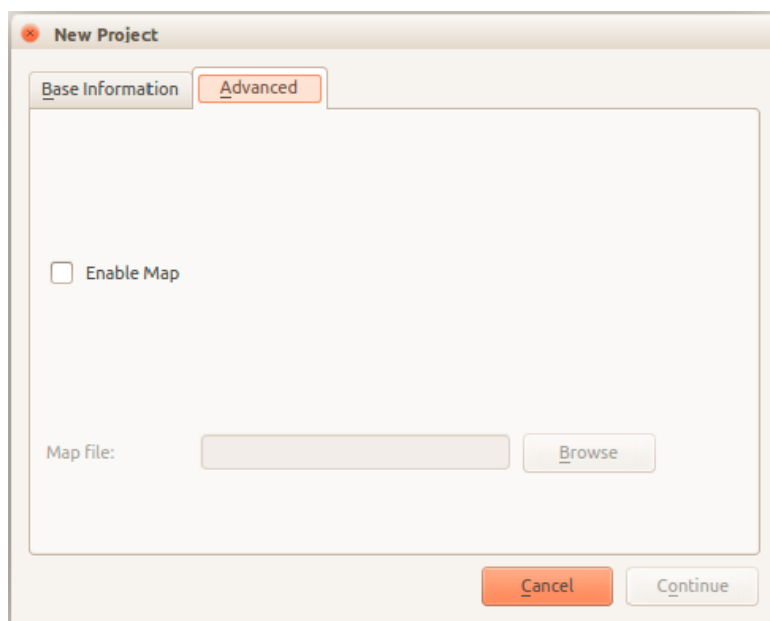
在软件界面最上面的工具栏中点击新建按钮（图 1）



在弹出的新对话框中填写相应的基础信息，如 launch 文件路径，Rviz 文件路径，节点字符串等



在高级（Advanced）选项卡中可以选择开启地图功能并填写地图文件路径，这样可以在新建的项目中使用已有地图

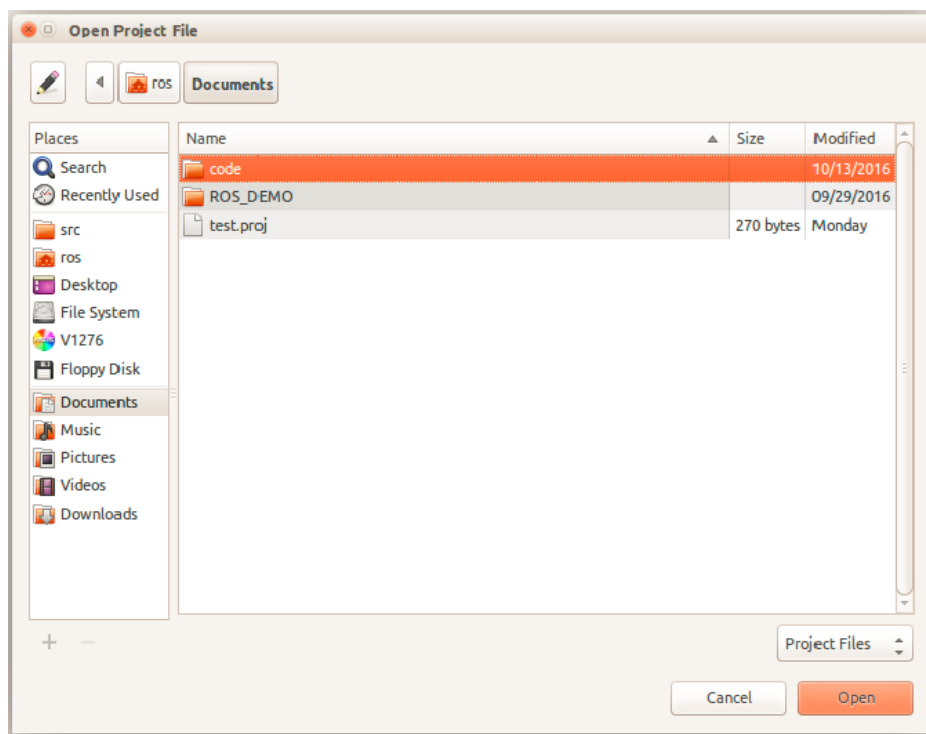


最后点击继续（Continue）按钮完成创建。

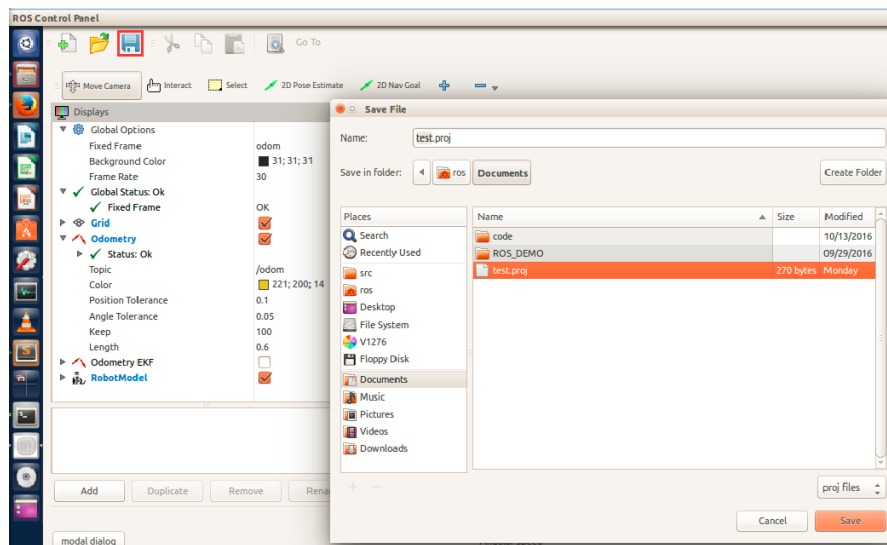
### 6.2.3 打开项目



在工具栏中点击打开按钮，弹出打开文件对话框，选择需要打开的项目文件即可

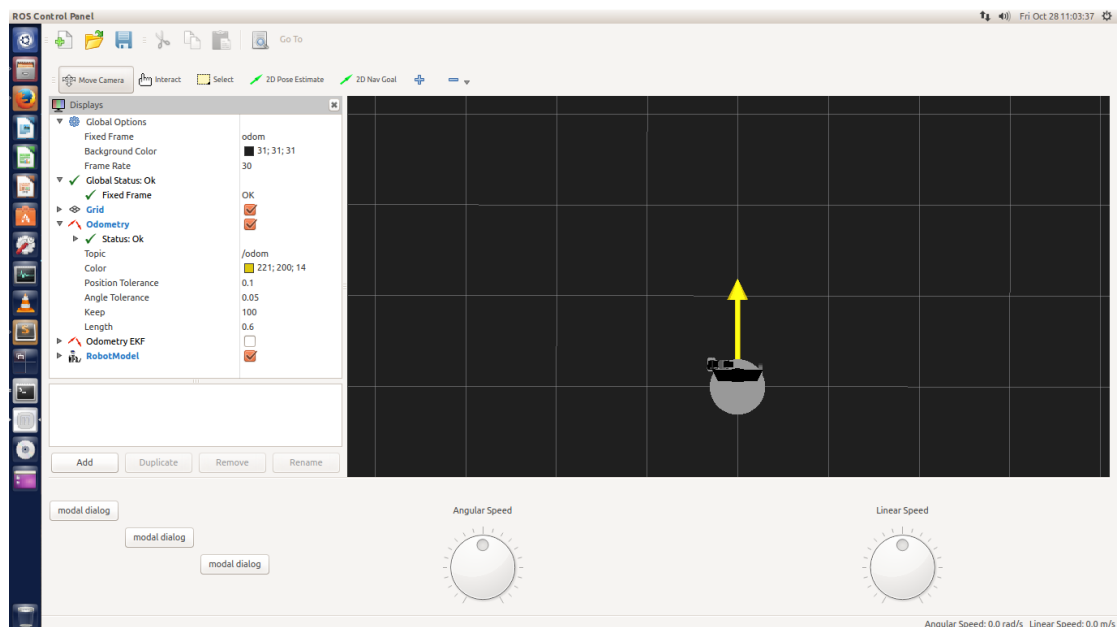


## 6.2.4 保存项目



在工具栏中点击保存按钮，在弹出的对话框中选择保存的路径并输入保存文件的文件名点击保存按钮即可。

## 6.2.5 界面介绍



### 6.2.6 软件界面

界面中的两个旋钮分别用来调整机器人的角速度和线速度，角速度为负的时候是向左转，为正的时候是向右转。线速度为正是前进，线速度为负是后退，左侧下方的三个按钮暂时无法使用。