

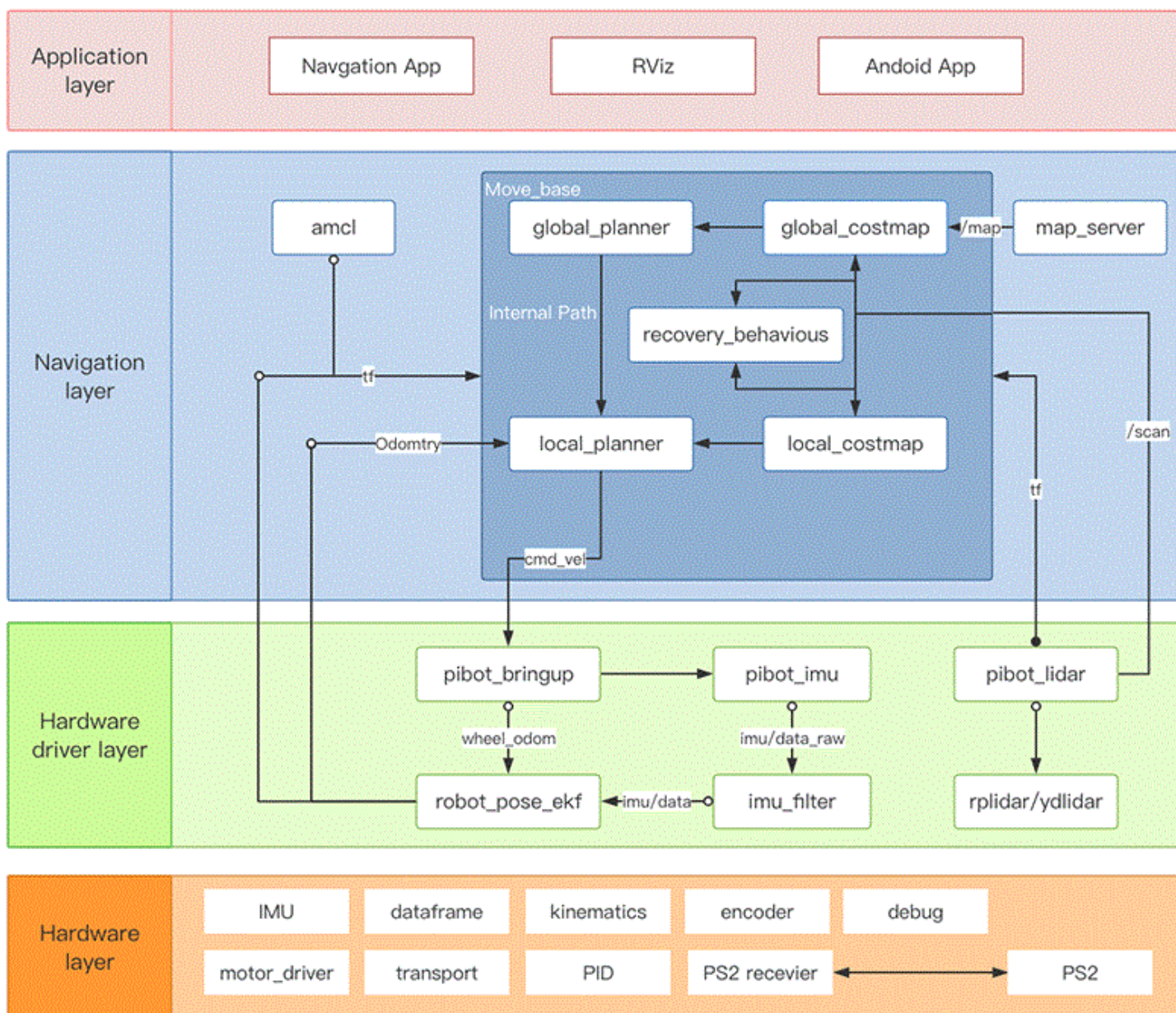
- 1. 概述
- 2. 软件框架
- 3. Ubuntu的刷入、ROS的安装
- 4. ssh远程连接
  - 4.1 安装xshell
    - [通过 Xshell 连接 树莓派/RK3288/RK3399](#)
    - Windows [通过 xshell 传输文件](#)
- 5. 用户主机
- 6. 多机通讯
- 7. ROS驱动开发
  - 7.1 概述
  - 7.2 驱动开发及PID参数动态调整
  - 7.3 目录结构简介
  - 7.4 编译与测试
    - 配置
      - [树莓派/RK3288/RK3399](#)
      - [控制PC](#)
    - 编译
    - 测试
      - [初始化配置](#)
      - [开始测试](#)
- 8. 校准
  - [无IMU校准](#)
  - [IMU校准](#)
- 9. ROS建图与导航
  - 9.1 概述
  - 9.2 建图
    - [两种建图方法](#)
    - [保存地图](#)
- 10 导航
  - [单点导航测试](#)
  - [多点导航](#)
- 11. 模拟器
- 12. Android App
  - [12.1相关功能](#)
  - [12.2 显示视频](#)
- 13. IMU的相关包使用

# 1. 概述

采用 树莓派/RK3288/RK3399/X86工控机/TX1/TX2 作为上位 ROS 主控,基于 ROS 开发适配导航建图算法

# 2. 软件框架

## 系统框架



# 3. Ubuntu的刷入、ROS的安装

具体请参考[树莓派\(raspberry pi 3b\)安装ROS Kinetic Kame](#)与[Firefly RK3288/3399固件刷新与ROS安装](#)

tf卡启动的 nanopi(RK3399) 则同 树莓派 一样操作

## 4. ssh远程连接

windows 中推荐安装 xshell 远程连接

树莓派/ nanopi rk3399 默认开启了热点 ssid 和 password 均为 pibot\_ap , 可以通过连接该热点连接树莓派/ nanopi rk3399 , 树莓派/ nanopi rk3399 的IP为 192.168.12.1

如需要关闭ap模式, 改为连接wifi, 只需要执行 `rm ~/.pibot_ap` 后重启

如需要打开ap模式, 开启则执行 `touch ~/.pibot_ap` 后重启

### 4.1 安装xshell

网盘中下载 xshell 并安装

< > ↕ ↻ | 我的网盘 > Pibot > pibot docs > software >

文件名

SourceInsight-3576

MDK520

Xshell\_5.0.1333.exe

### 通过 Xshell 连接 树莓派/RK3288/RK3399

输入 IP 和用户名密码

- 树莓派3B/3B+ 为 pibot
- firefly RK3288、RK3399 为 firefly
- nanopi 3399 为 pi

类别(C):

- [-] 连接
  - [-] 用户身份验证
    - 登录提示符
    - 登录脚本
  - [-] SSH
    - 安全性
    - 隧道
    - SFTP
  - TELNET
  - RLOGIN
  - SERIAL
  - 代理
  - 保持活动状态
- [-] 终端
  - 键盘
  - VT 模式
  - 高级
- [-] 外观
  - 窗口
- [-] 高级
  - 跟踪
  - 日志记录
- [-] 文件传输
  - X/YMODEM
  - ZMODEM

连接

常规

名称(N): PIBOT

协议(P): SSH

主机(H): 192.168.1.2

端口号(O): 22

说明(D):



重新连接

连接异常关闭时自动重新连接(A)

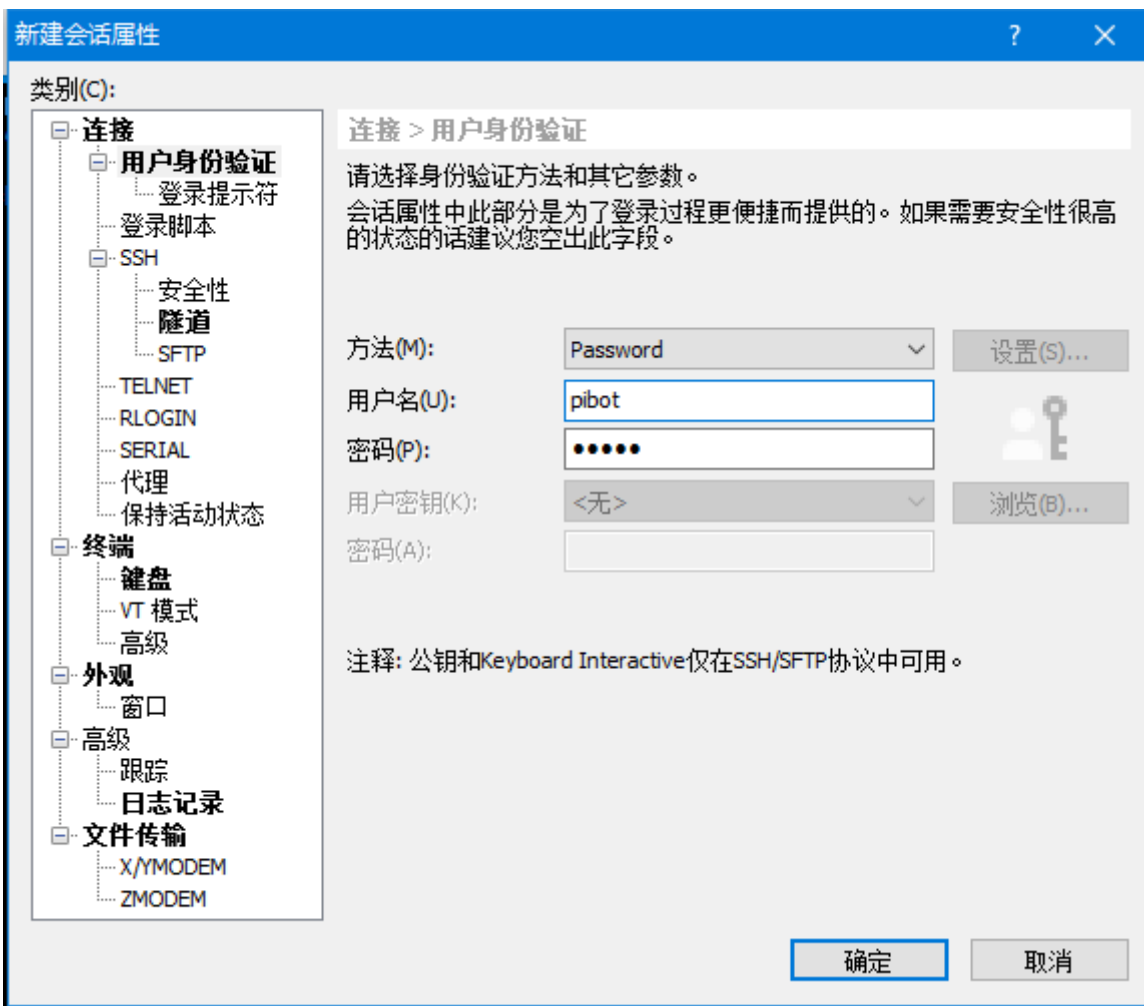
间隔(V): 0 秒 限制(L): 0 分钟

TCP选项

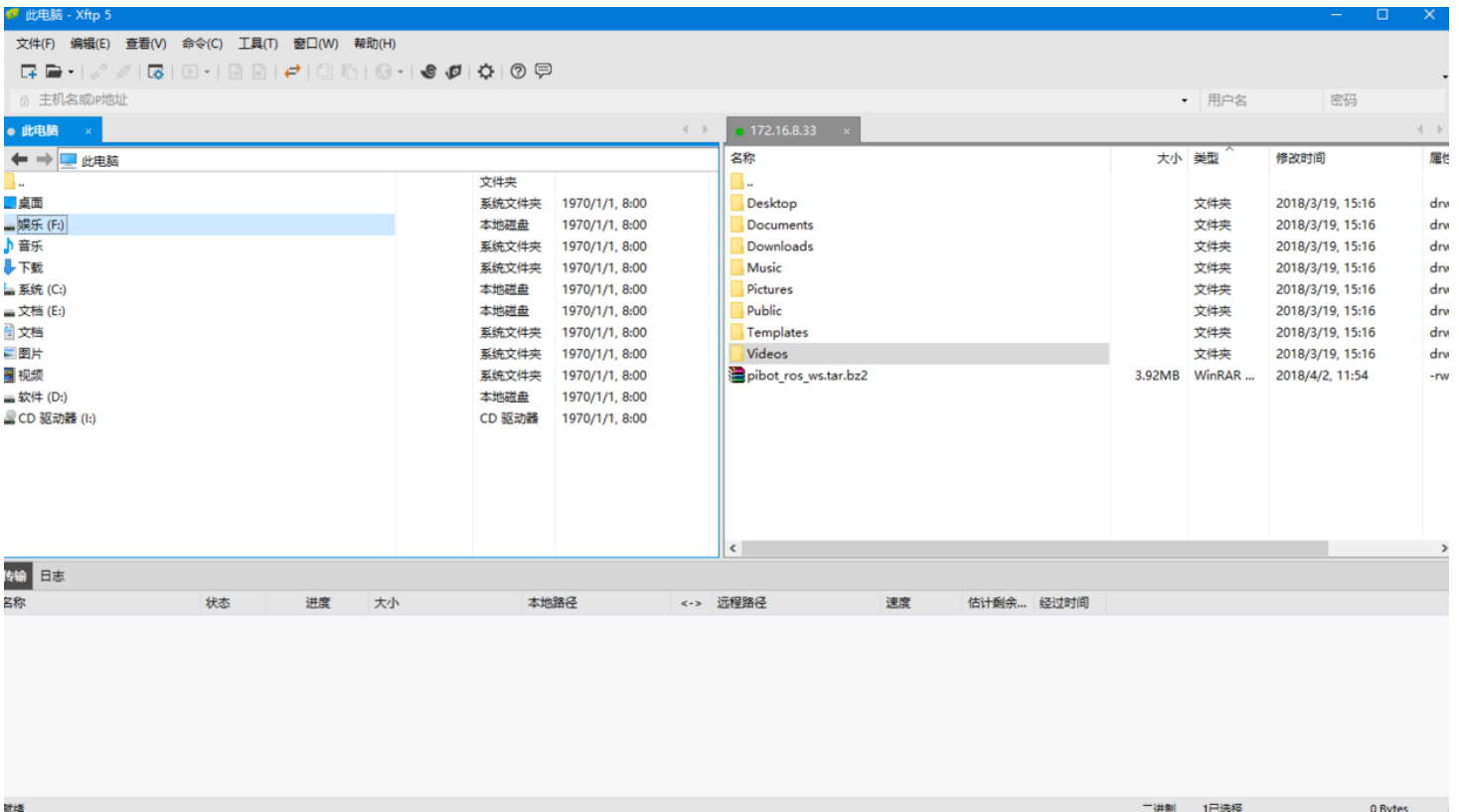
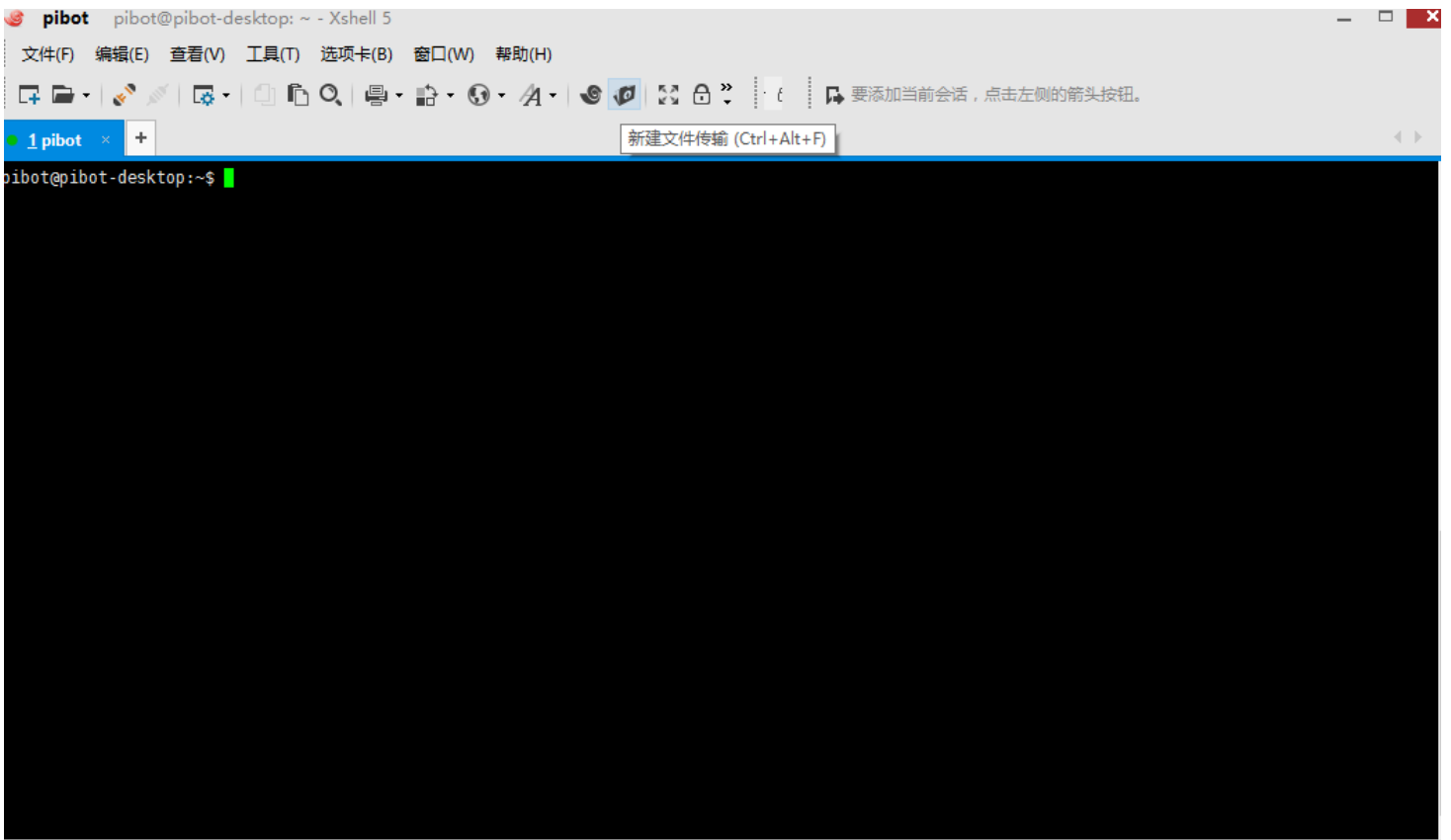
使用Nagle算法(U)

确定

取消



## Windows 通过 xshell 传输文件



## 5. 用户主机

需要一个 PC 安装 ROS 环境，用来显示查看地图或者玩转模拟器等，PIBOT 提供了一个一键安装 ROS 的脚本 `pibot_install_ros`，可以直接在 Ubuntu 下安装 ROS

用户主机环境 Ubuntu 16.04 或者 Windows7/10+Vmware+Ubuntu16.04 虚拟机 Ros kinetic 环境, 安装 ROS 参见[Windows下安装Ubuntu虚拟机及ROS](#)相关章节

## 6. 多机通讯

**用户主机**与 树莓派/RK3288/RK3399/TK1/TX1/TX2/X86主机 怎么建立 ROS 通讯的，PIBOT 提供了一键配置脚本 `pibot_init_env`

如需了解细节

请参考[ROS多机的通讯配置](#)，[pibot\\_init\\_env](#)介绍即可

## 7. ROS驱动开发

### 7.1 概述

下位机及通过串口与 树莓派/RK3288/RK3399/TK1/TX1/TX2/X86主机 通讯，PIBOT 提供了一个简单的协议，通讯协议具体请参见[ROS机器人底盘\(3\)-通讯协议](#)，同时 PIBOT 提供两个一个 ROS 无关的串口控制接口具体可以参见[PIBOT通讯协议的python解析实现](#)

### 7.2 驱动开发及PID参数动态调整

PID 参数已在出厂时候配置，如需了解细节请参考[ROS机器人底盘\(10\)-PIBOT的driver的实现及动态PID调节](#)

### 7.3 目录结构简介

**建议拷贝提供的压缩文件至目标设备( 树莓派/RK3288/RK3399/X86工控机/TK1/TX1/TX2 )上解压或者直接 `git clone`，不然会遇到一些问题，具体问题见Q&A**

PIBOT 的 ROS workspace 目录如下图

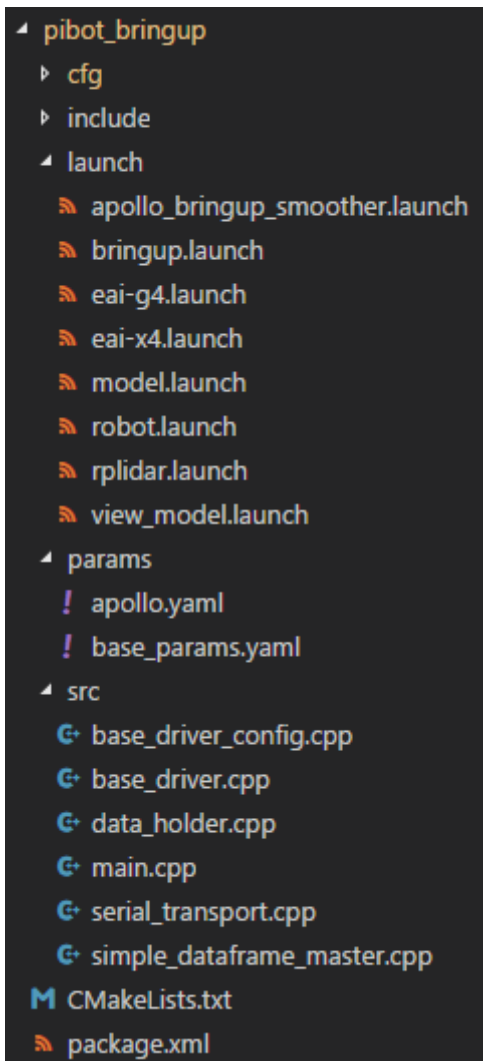
```
src
├── arbotix_ros
├── pibot
├── pibot_bringup
├── pibot_description
├── pibot_navigation
├── pibot_simulator
├── rplidar_ros
└── ydlidar-1.2.1
```

- arbotix\_ros 模拟器
- pibot 工具集

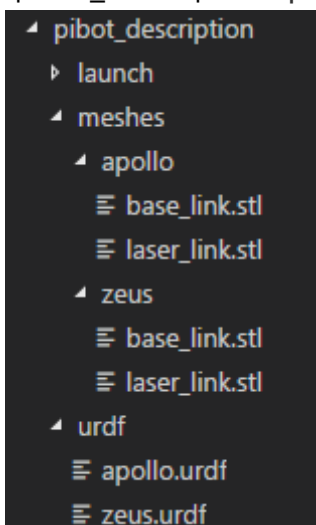
```
pibot
├── cfg
│   ├── CalibrateAngular.cfg
│   └── CalibrateLinear.cfg
├── config
│   ├── joystick-holonomic.config.yaml
│   └── joystick.config.yaml
├── launch
│   ├── joystick-holonomic.launch
│   ├── joystick.launch
│   ├── keyboard_teleop.launch
│   └── usb_camera.launch
├── rules
│   ├── pibot.rules
│   ├── rplidar.rules
│   └── ydlidar.rules
├── scripts
│   ├── calibrate_angular.py
│   ├── calibrate_linear.py
│   ├── launch_demo.py
│   ├── navigation_demo.py
│   ├── setup.sh
│   ├── teleop_twist_keyboard.py
│   └── transform_utils.py
```

- pibot\_bringup pibot驱动包





- pibot\_description pibot urdf文件



- pibot\_navigation 建图导航相关配置项

```
└─ pibot_navigation
  └─ launch
    ├── include
    ├── gmapping.launch
    ├── nav.launch
    └── view_nav.launch
  └─ maps
  └─ params
    ├── ! costmap_common_params_apollo.yaml
    ├── ! costmap_common_params_zeus.yaml
    ├── ! dwa_local_planner_params_apollo.yaml
    ├── ! dwa_local_planner_params_zeus.yaml
    ├── ! global_costmap_params.yaml
    ├── ! global_planner_params.yaml
    ├── ! local_costmap_params.yaml
    └── ! move_base_params.yaml
```

- pibot\_simulator pibot导航模拟器

```
└─ pibot_simulator
  └─ launch
    ├── bringup.launch
    └── nav.launch
  └─ maps
    ├── ≡ blank_map_with_obstacle.pgm
    ├── ! blank_map_with_obstacle.yaml
    ├── ≡ blank_map.pgm
    ├── ! blank_map.yaml
    ├── ≡ test_map.pgm
    └── ! test_map.yaml
  └─ params
    ├── ! arbotix.yaml
    ├── ! costmap_common_params.yaml
    ├── ! dwa_local_planner_params.yaml
    ├── ! global_costmap_params.yaml
    ├── ! global_planner_params.yaml
    ├── ! local_costmap_params.yaml
    └── ! move_base_params.yaml
  └─ CMakeLists.txt
```

- rplidar\_ros rplidar激光雷达驱动包
- ydlidar-1.2.1 eai激光雷达驱动包

## 7.4 编译与测试

### 配置

以 apollo 车型安装 rplidar 为例

## 树莓派/RK3288/RK3399

```
pibot@pibot-desktop:~/pibot_ros$ ./pibot_init_env.sh
please specify pibot model(0:apollo,1:apolloX,2:zeus,3:hera,4:hades,other for user defined):
0
please specify pibot driver board type(0:arduino(mega2560),1:stm32f103,2:stm32f407,other for user defined):
2
please specify your pibot lidar(0:rplidar(a1,a2),1:rplidar(a3),2:eai(x4),3:eai(g4),4:xtion,5:astrax,6:kinectV1,other for user defined):
0
please specify the current machine(ip:192.168.2.231) type(0:onboard,other:remote):
0
*****
model:      apollo
lidar:      rplidar
local_ip:   192.168.2.231
onboard_ip: 192.168.2.231
*****
please execute source ~/.bashrc to make the configure effective
*****
```

小车类型  
驱动板类型  
雷达类型 (包括使用摄像头模拟)  
主机类型 (车载端or控制端)  
生效配置

## 控制PC

```
pibot@pibot-desktop:~/pibot_ros$ ./pibot_init_env.sh
please specify pibot model(0:apollo,1:apolloX,2:zeus,3:hera,4:hades,other for user defined):
0
please specify pibot driver board type(0:arduino(mega2560),1:stm32f103,2:stm32f407,other for user defined):
2
please specify your pibot lidar(0:rplidar(a1,a2),1:rplidar(a3),2:eai(x4),3:eai(g4),4:xtion,5:astrax,6:kinectV1,other for user defined):
0
please specify the current machine(ip:192.168.2.177) type(0:onboard,other:remote):
2
please specify the onboard machine ip for communication:
192.168.2.231
*****
model:      apollo
lidar:      rplidar
local_ip:   192.168.2.177
onboard_ip: 192.168.2.231
*****
please execute source ~/.bashrc to make the configure effective
*****
```

小车类型  
主板类型  
雷达类型 (包括使用摄像头模拟)  
主机类型 (小车or控制端)  
指定控制端, 需要指定小车IP  
生效配置

## 编译

```
cd ~/pibot_ros/ros_ws
catkin_make
```

## 测试

### 初始化配置

重新拔插USB口或者重启 树莓派/RK3288/RK3399

```
ls /dev/pibot -l
```

```
pibot@pibot-desktop:/$ ls /dev/pibot -l
lrwxrwxrwx 1 root root 7 12月 20 10:47 /dev/pibot -> ttyACM0
```

### 开始测试

- 在 树莓派/RK3288/RK3399 运行 `pibot_bringup` 或 `roslaunch pibot_bringup bringup.launch`
- 在**用户主机**运行 `pibot_configure` 或 `roslaunch pibot_configure rqt_reconfigure` 可以查看和修改内置的配置信息，运行 `pibot_control` 或 `roslaunch pibot keyboard_teleop.launch` 即可通过键盘控制小车运动

同时支持小米等手柄的接入，运行 `roslaunch pibot joystick.launch` 即可

## 8. 校准

### 无IMU校准

参见[ROS机器人底盘\(11\)-PIBOT的控制及校准](#)

### IMU校准

参见[ROS机器人底盘\(25\)-PIBOT的IMU校准](#)

## 9. ROS建图与导航

### 9.1 概述

ROS 驱动中提供了 `cmd_vel` 的订阅及 `odom` 的发布，至此再需要一个激光雷达就可以完成建图了

### 9.2 建图

在 树莓派/RK3288/RK3399 运行 `pibot_gmapping` 或 `roslaunch pibot_navigation gmapping.launch`

在**用户主机**运行 `roslaunch pibot_navigation view_nav.launch` 或者 `pibot_view`

#### 两种建图方法

- 运行 `roslaunch pibot keyboard_teleop.launch` 或者 `roslaunch pibot joystick.launch` 即可通过键盘或者遥控手柄开始建图
- 直接选择导航的点( 2D Nav Goal )开始建图

## 保存地图

- 运行下列命令即可(xxx为自定义名称)

```
roslaunch pibot_navigation save_map.launch map_name:=xxx
```

或者

```
roscd pibot_navigation/maps/  
roslaunch map_server map_saver -f xxx`
```

```
pibot@pibot-desktop:~$ roscd pibot_navigation/maps/  
pibot@pibot-desktop:~/pibot_ros/ros_ws/src/pibot_navigation/maps$ roslaunch map_server map_saver -f my_home  
[ INFO] [1528035442.429381816]: Waiting for the map  
[ INFO] [1528035442.695666089]: Received a 608 X 576 map @ 0.050 m/pix  
[ INFO] [1528035442.695809892]: Writing map occupancy data to my_home.pgm  
[ INFO] [1528035442.733498745]: Writing map occupancy data to my_home.yaml  
[ INFO] [1528035442.734178644]: Done  
  
pibot@pibot-desktop:~/pibot_ros/ros_ws/src/pibot_navigation/maps$ ls my_home* -l  
-rw-rw-r-- 1 pibot pibot 350264 6月  3 22:17 my_home.pgm  
-rw-rw-r-- 1 pibot pibot   135 6月  3 22:17 my_home.yaml  
pibot@pibot-desktop:~/pibot_ros/ros_ws/src/pibot_navigation/maps$
```

可以看到生成2个文件

## 10 导航

### 单点导航测试

在 树莓派/RK3288/RK3399 运行 `roslaunch pibot_navigation nav.launch map_name:=xxx.yaml`

在**用户主机**运行 `pibot_view` 或 `roslaunch pibot_navigation view_nav.launch` , 通过 RViz 提供的功能既可以完成导航测试(这里需要先指定初始位置)

如果直接运行 `roslaunch pibot_navigation nav.launch` 而不指定 `map_name` 参数则使用默认参数, `nav.launch` 文件中可以设置默认使用的地图文件

```
launch>
<include file="$(find pibot_bringup)/launch/robot.launch"/>
<param name="use_sim_time" value="false" />
<arg name="map_name" default="my_lab.yaml" />
<node name="map_server" pkg="map_server" type="map_server" args="$(find pibot_navigation)/maps/$(arg map_name)"/>
<include file="$(find pibot_navigation)/launch/include/move_base.launch.xml" />
<include file="$(find pibot_navigation)/launch/include/amcl.launch.xml" />
</launch>
```

## 多点导航

可以通过修改 `pibot/scripts` 中的 `navigation_demo.py` 的 python 脚本完成单点和多点的导航  
具体可以参考[ROS机器人底盘\(18\)-如何在实际项目中应用ROS导航相关 \(1\)](#)

# 11. 模拟器

PIBOT包内置了模拟器，可以直接运行模拟导航

- 运行 `roslaunch pibot_simulator nav.launch` 或者 `pibot_simulator`
- 运行 `roslaunch pibot_navigation view_nav.launch`  
这样无需小车也可以模拟导航了

# 12. Android App

## 12.1 相关功能

保证手机跟跟 PC 或者 树莓派/RK3288/RK3399 连接同一个网络，保证手机能够访问到 `roscore` ( `export ROS_IP=XXX.XXX.XXX.XXX` )具体参见[ROS多机的通讯配置](#)

13:57

0.36K/s \*   HD 

# Map Nav

Master URI: http://172.16.4.177:11311/

READ QRCODE

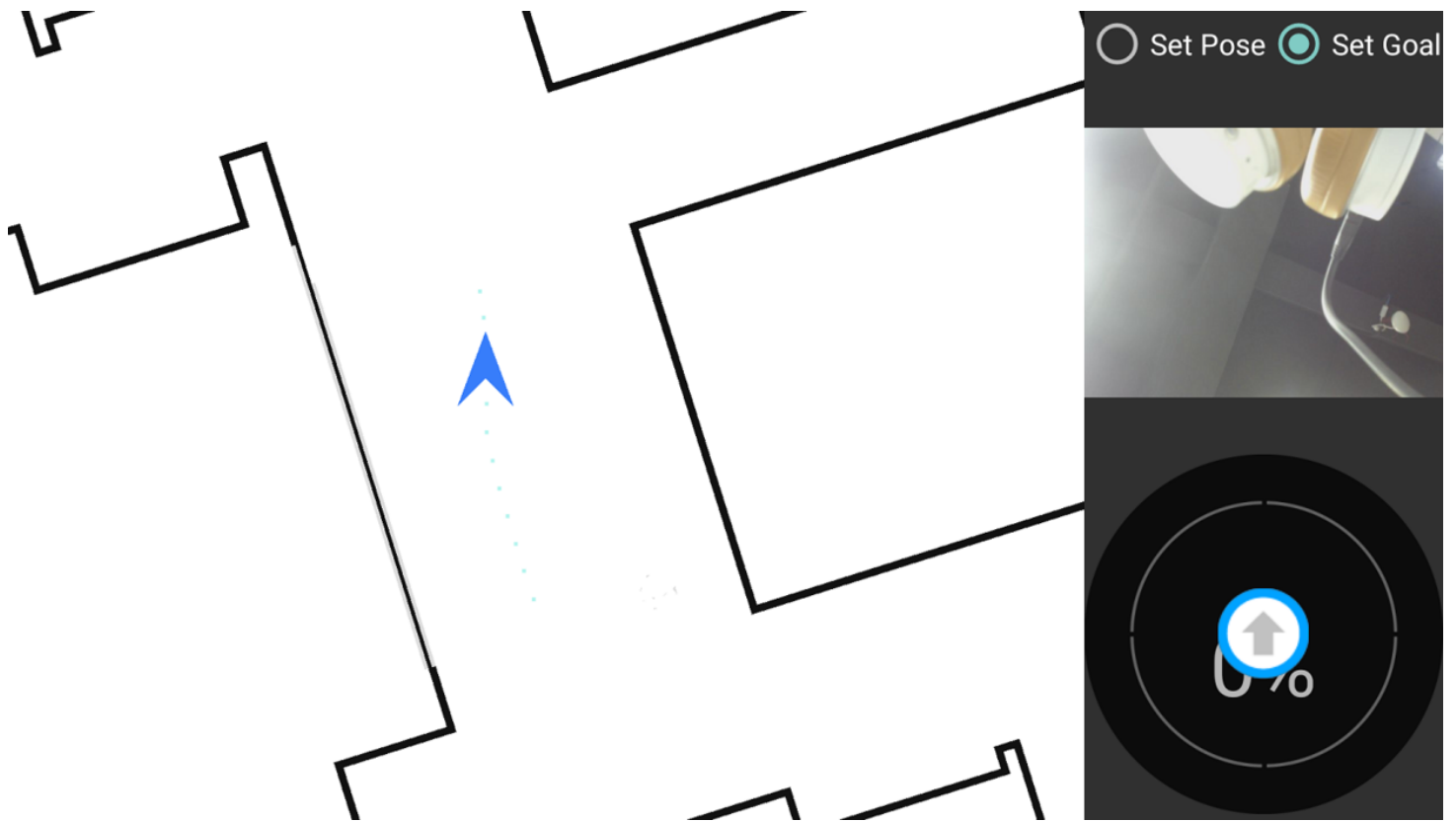
CONNECT

Show advanced options

CANCEL

修改 Master URI 选择 roscore 的 URI 点击 CONNECT

切换 Camera View 与 Map View



- Set Pose 在地图长按 相当于 Rviz 中的 2D Pose Estimate
- Set Goal 在地图长按 相当于 Rviz 中的 2D Nav Goal
- 左下角 Joystick 可以发出 `cmd_vel` topic 控制小车移动

## 12.2 显示视频

显示视频需要硬件摄像头支持同时在 PC 或者 树莓派/RK3288/RK3399 启动 `roslaunch pibot usb_camera.launch`

## 13. IMU的相关包使用

装有 IMU 的 PIBOT 系列小车，底层提供 IMU 的数据采集，上层提供了 IMU 的互补滤波以及融合里程计和 IMU 的扩展的卡尔曼滤波包 [robot\\_pose\\_ekf](#)

启动时只需相应的 `with_imu` 的 launch 文件,例如

```
roslaunch pibot_bringup bringup_with_imu.launch 或者 pibot_bringup_with_imu
roslaunch pibot_navigation gmapping_with_imu.launch 或者 pibot_gmapping_with_imu
roslaunch pibot_navigation nav_with_imu.launch 或者 pibot_navigation_with_imu
```

具体可以参考[ROS机器人底盘\(22\)-IMU和里程计融合](#)

[ROS机器人底盘\(23\)-IMU和里程计融合与单独编码器里程计的对比测试](#)