

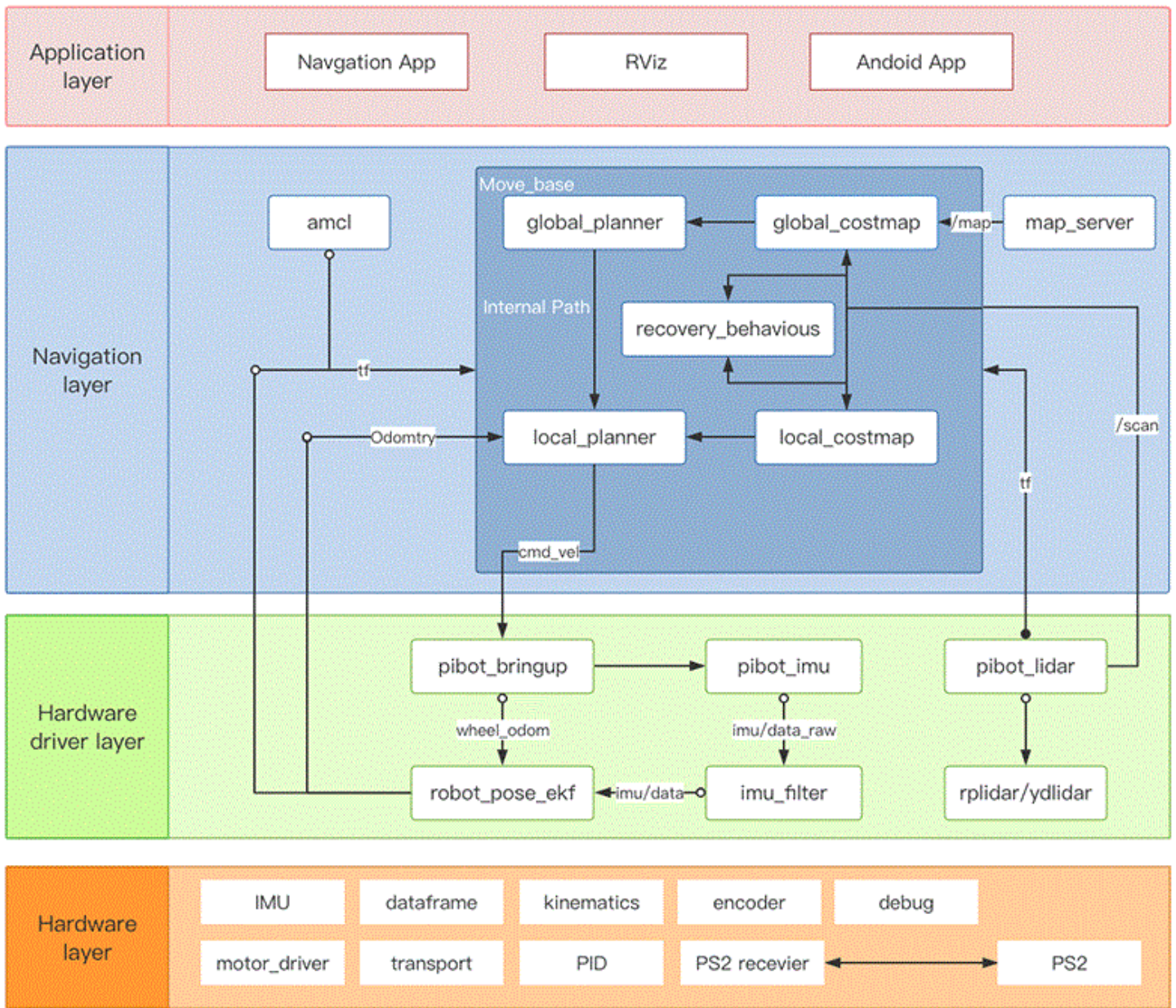
- 1. 概述
- 2. 软件框架
- 3. Ubuntu的刷入、ROS的安装
- 4. ssh远程连接
 - 4.1 安装xshell
 - 通过 Xshell 连接 树莓派/RK3288/RK3399
- 5. 用户主机
- 6. 多机通讯
- 7. ROS驱动开发
 - 7.1 概述
 - 7.2 驱动开发及PID参数动态调整
 - 7.3 目录结构简介
 - 7.4 编译与测试
 - 编译
 - 测试
 - 初始化配置
 - 开始测试
- 8. 校准
 - 无IMU校准
 - IMU校准
- 9. ROS建图与导航
 - 9.1 概述
 - 9.2 建图
 - 两种建图方法
 - 保存地图
- 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/TK1/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\(raspberrypi 3b/4b\)安装ROS](#)

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 `cd ~/pibot_ros && ./pibot_ap.sh stop` 后重启
如需要打开ap模式，开启则执行 `cd ~/pibot_ros && ./pibot_ap.sh start` 后重启

4.1 安装xshell

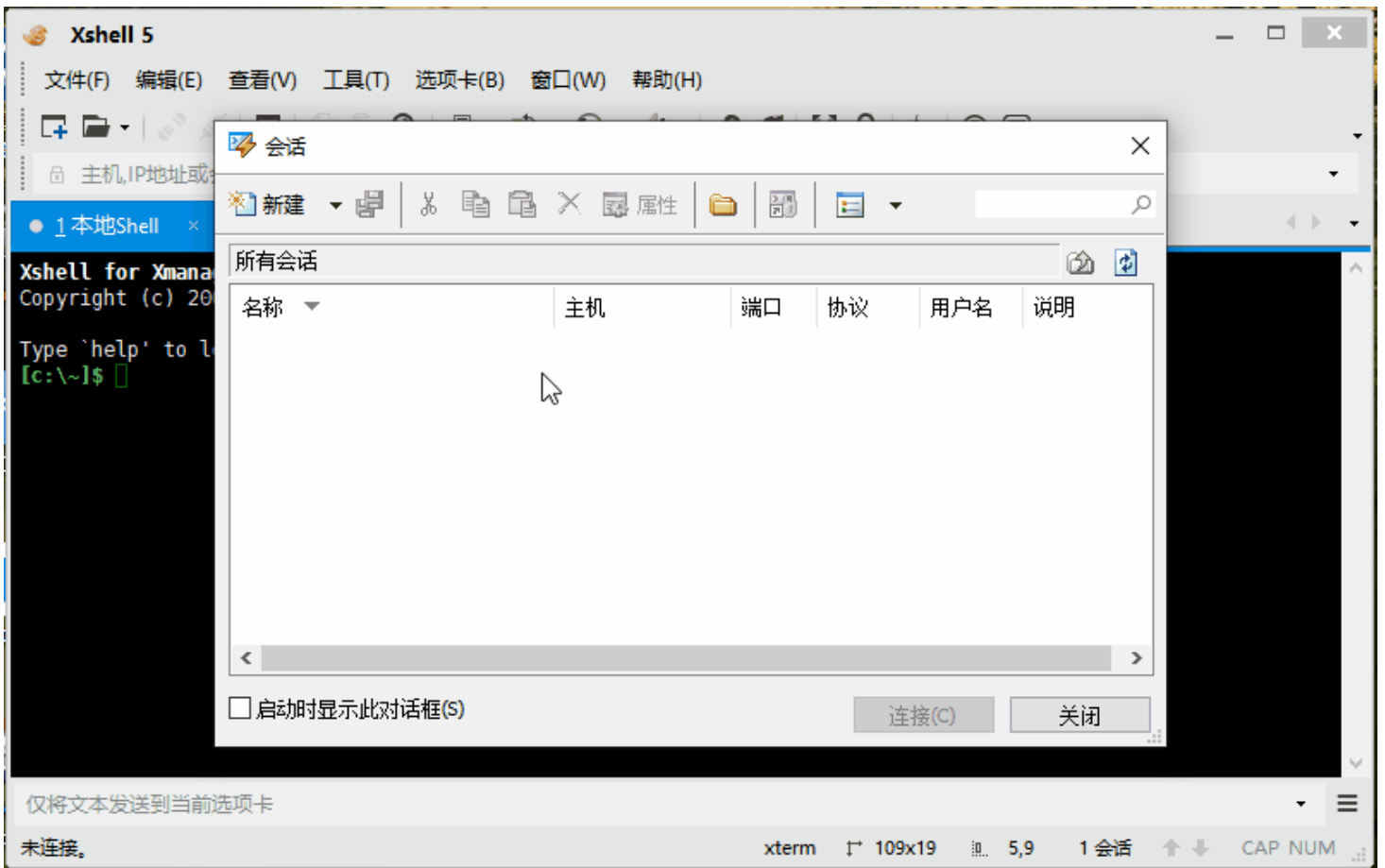
网盘中下载 xshell 并安装



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



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 或者 Windows7/10+Vmware+Ubuntu 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`

- 设备端

```
pi@pi:~$
```

1

- 主机端

```
pi@pi:~$
```

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[协议文档](#)

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[PIBOT的driver的实现及动态PID调节](#)

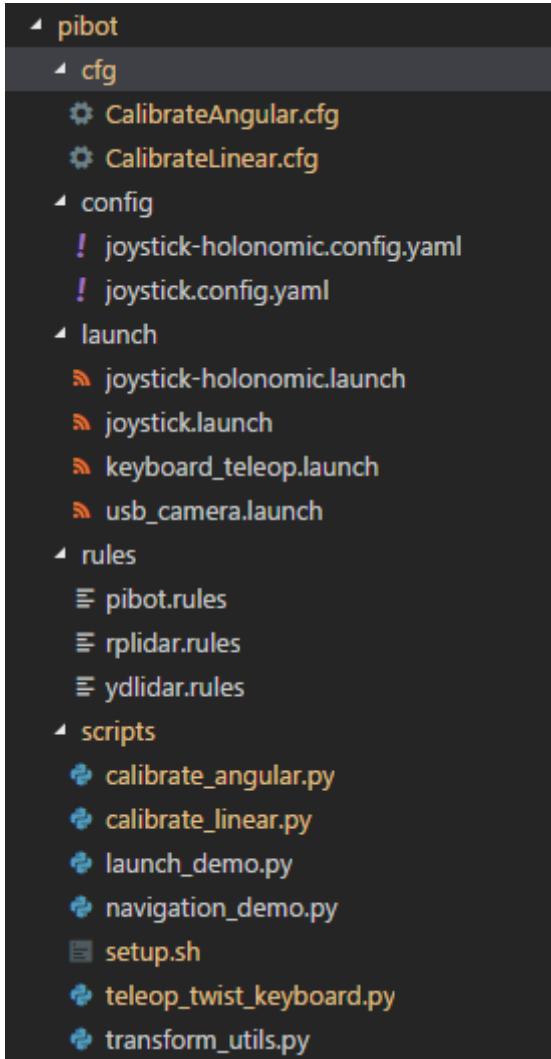
7.3 目录结构简介

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_bringup pibot驱动包

```
└─ pibot_bringup
  ├── cfg
  ├── include
  └─ launch
    ├── apollo_bringup_smoother.launch
    ├── bringup.launch
    ├── eai-g4.launch
    ├── eai-x4.launch
    ├── model.launch
    ├── robot.launch
    ├── rplidar.launch
    └── view_model.launch
  ├── params
  │ ├── apollo.yaml
  │ └── base_params.yaml
  ├── src
  │ ├── base_driver_config.cpp
  │ ├── base_driver.cpp
  │ ├── data_holder.cpp
  │ ├── main.cpp
  │ ├── serial_transport.cpp
  │ └── simple_dataframe_master.cpp
  ├── CMakeLists.txt
  └── package.xml
```

- pibot_description pibot urdf文件

```
└─ pibot_description
  ├── launch
  ├── meshes
  │ ├── apollo
  │ │ ├── base_link.stl
  │ │ └── laser_link.stl
  │ └── zeus
  │   ├── base_link.stl
  │   └── laser_link.stl
  └── urdf
    ├── apollo.urdf
    └── zeus.urdf
```

- 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

编译

```
cd ~/piobot_ros
git pull # 拉取最新的代码
cd ~/piobot_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/piobot -l
```

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

开始测试

打开终端输入 piobot_ ,再输入2次 tab 提示支持的命令

```
 david@david-desktop:~$
```

```
 I
```

这些命令具体定义在 ~/.piobotrc 文件,该文件是 piobot_init_env 脚本自动生成的

8. 校准

无IMU校准

参见[PIBOT的控制及校准](#)

IMU校准

参见[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

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/

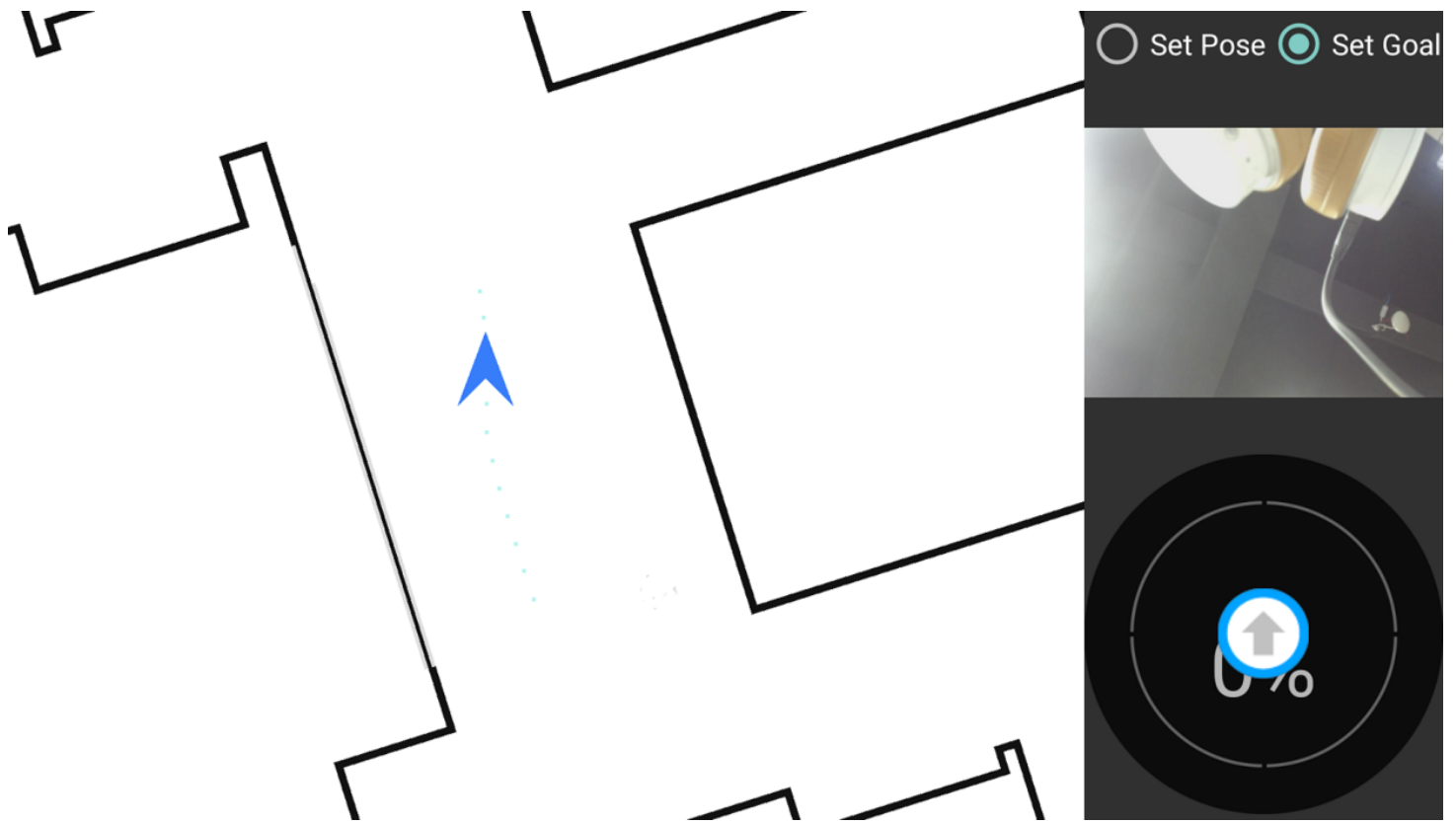
READ QRCODE

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 后
动 `roslaunch pibot usb_camera.launch`
或者直接浏览器访问192.168.12.1:8080选择相应的Topic即可

Available ROS Image Topics:

- /camera/rgb/
 - [image_rect_color \(Snapshot\)](#)
 - [image_raw \(Snapshot\)](#)
- /camera/projector/
- /camera/ir/
 - [image \(Snapshot\)](#)
- /camera/depth_registered/
 - [hw_registered/image_rect \(Snapshot\)](#)
 - [image \(Snapshot\)](#)
 - [hw_registered/image_rect_raw \(Snapshot\)](#)
 - [image_raw \(Snapshot\)](#)
- /camera/depth/
 - [image_rect_raw \(Snapshot\)](#)
 - [image_rect \(Snapshot\)](#)
 - [image_raw \(Snapshot\)](#)
 - [image \(Snapshot\)](#)

192.168.12.1为设备地址, 以具体设备为准

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，具体参见 `pibot_` 命令