中型组仿真赛指导之环境搭建

(一) 清单:

1、Ubuntu 16.04 或者 Ubuntu 14.04（建议选择 Ubuntu 16.04 版本）;

2、ROS（ROS Kinetic 对应 Ubuntu 16.04，ROS Indigo 或者 ROS Jade 对应 Ubuntu14.04，接下来的 demo 以 Kinetic 为例;

3、工程编译环境 QT，建议版本 QT5.3.2，接下来的 demo 为离线安装方式

(二)ROS 安装

参考网页 http://wiki.ros.org/ROS/Installation （见图 1）

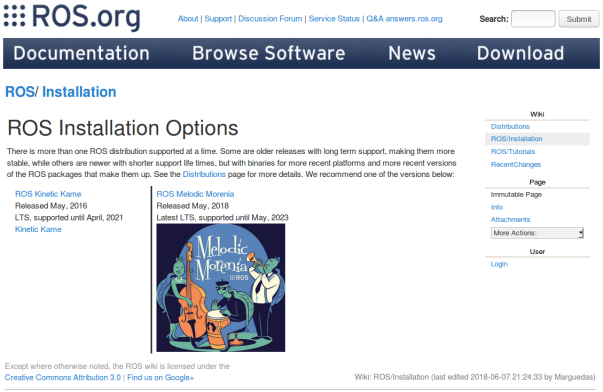

图 1: ROS 安装界面

选择对应版本后，选择对应平台 (目前仅仅限于 Ubuntu)，即可进入安装引导界面（见图 2)。

按照指导依次输入指令即可完成 ROS 安装，注意，如果遇到下载速度特别慢的情况，可以更换镜像，即在 1.2 Setup your source.list 过程中选择 Mirrors（推荐选择 USTC 镜像，见图 3)。

ROS 安装完之后，可以进行简单测试，命令行运行 roscore，如未出现报错，即视为安装成功（见图 4)。

(三)QT 安装

首先，如果电脑上安装了其他版本 qt 由于版本过高或者过低的原因引起无法正常打开文件，可以卸载，进入，卸载方式为进入 qt 安装目录，运行 MaintenanceTool 脚本即可进入卸载的图形化界面（见图 5)。

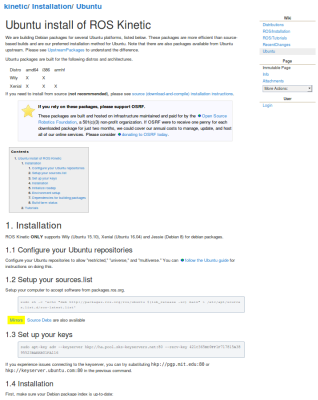

图 2: 安装引导界面

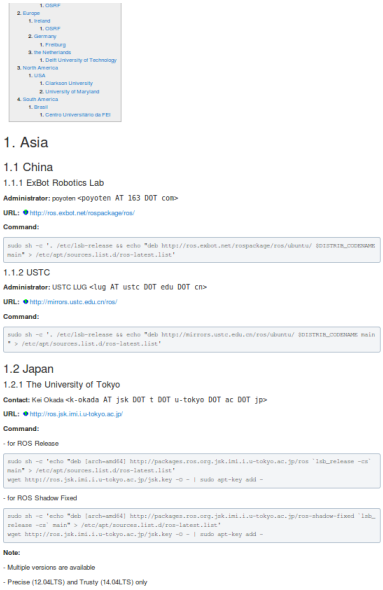

图 3: 镜像选择界面

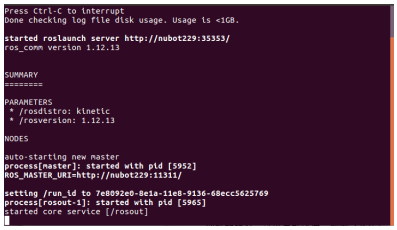

图 4: 安装成功显示

以 qt5.3.2 离线安装为例，首先下载 qt-opensource-linux-x64-5.3.2.run (https://pan.baidu.com/s/1EITYJLg-wTDW-p0XZaPa0Q)，自行百度下载也可。

而后，赋予安装文件权限，命令为 chmod +x qt-opensource-linux-64-5.3.2.run，最后，运行 run 文件可以 (linux 下运行可执行文件的方式为./可执行文件位置)，此时即可进入图形化安装界面，均选择默认选项即可。

(四)QT 与 ROS 环境配置

QT 安装完成后，需要对其进行一些简单配置，使其能够正确索引到 ROS

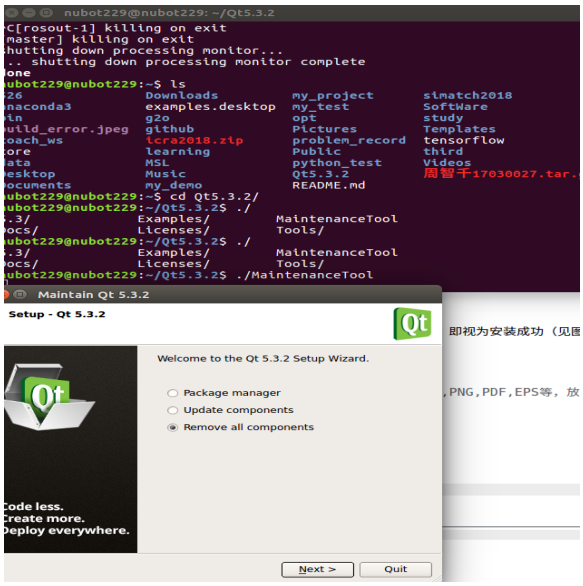

图 5: qt 卸载

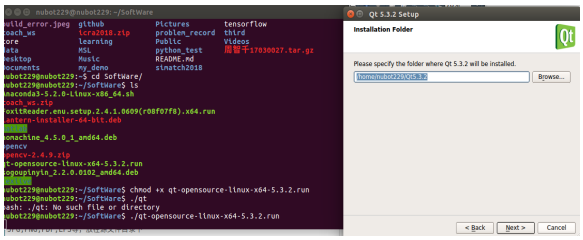

图 6: 运行 run 文件

相关文件 (可参见 simatch 工程中 doc 下的 ROS Qt.pdf)，在此，仅仅以 qt5.3.2 为例，进行环境配置，不对原理进行过多阐述。

首先，如果你的 QT 没有与 ROS 正确链接，在利用 QT 打开 simatch 工程会报错，出现的错误见图 7。

报错的大致意思为，QT 未找到 ROS 中的 catkin_make ，而 simatch 工程是利用 catkin make 搭建的，因此无法构建工程（关于 catkin_make 和 cmake 的区别，catkin_make 工程结构，可以自行学习一些基础知识）。

然后是进行 qt 与 ros 环境的配置,对于默认的安装方式,在 /.local/share/applications 存在 DigiaQtOpenSourceqtcreator.desktop 文件,这想当于 qtcreator 启动时

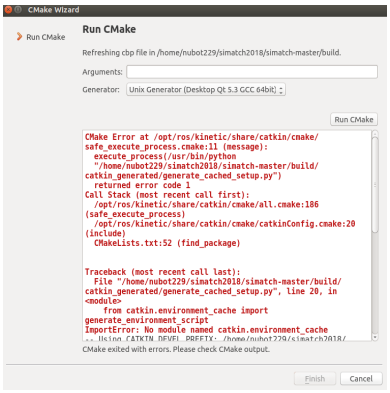

图 7: QT 与 ROS 未链接报错

的脚本, 只需在此添加对 ROS 环境的索引即可, 在 Exec=/home/nubot229/Qt5.3.2/Tools/QtCreator/bin/qtcreator 一行添加 bash -i -c ,改为 Exec=bash -i -c /home/nubot229/Qt5.3.2/Tools/QtCreator/bin/qtcreator。其中 bash -i -c 含义为添加对 /.bashrc 的索引, 而.bashrc 的内容即为命令行的环境配置, 在 ros 安装完成后 /.bashrc 低端会添加一行,source /opt/ros/kinetic/setup.bash, 将 ros 的文件添加到默认环境中。

注意，DigiaQtOpenSourceqtcreator.desktop 文件需要 root 权限，因此必

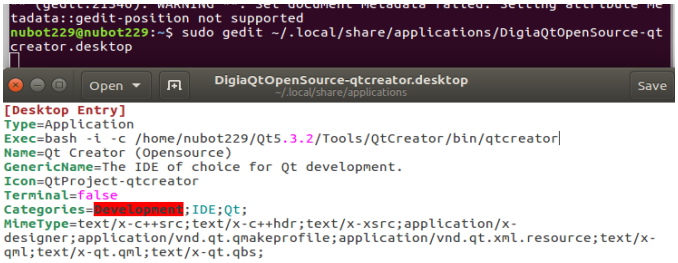

图 8: 更改 DigiaQtOpenSourceqtcreator.desktop

须要 sudo，否则无法进行更改，gedit 是文件编辑器，也可使用 vim，在更改之后，需要重启 QT，即可生效, 配置成功后编译 simatch 工程输出如图 9。

图 9 中的输出结果标红并不是报错，而是突出显示，说明相关环境配置成

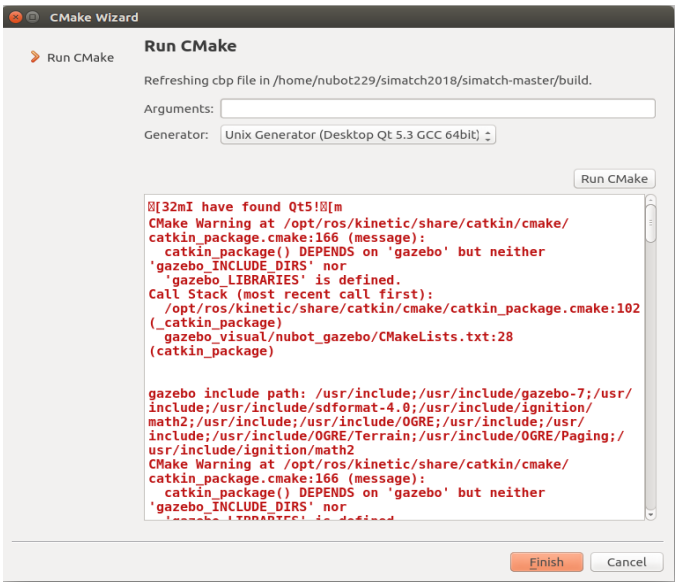

图 9: 配置成功显示

功，只要 finish 变红即可视为工程构建成功。

最终，工程打开结果如图 10。

(五)simatch 工程运行

首先，进入 simatch 工程目录，运行 catkin_make ，进行工程编译（如果 ROS 未安装，catkin_make 会报错，Ubuntu 系统和 catkin_make 相关基础知识需要大家自行学习）。而后运行 source devel/setup.bash，与之前 ROS 安装后需要在.bashrc 中添加 source /opt/ros/kinetic/setup.bash 一样，devel/setup.bash 是对 simatch 工程下所有可执行文件、函数库的索引文件，source 之后终端即可直接调用相关可执行文件（roslaunch、rosrun 本质上都是执行可执行文件），此外，需要注意的是，.bashrc 文件时每次开终端之前都会调用，因此，如果在.bashrc 中对 simatch 中的 setup.bash 进行了 source，那么重启终端后便不在需要这一操作，而如果没有在.bashrc 中添加，则每次重启终端都需要在 simatch 工程目录下运行 source devel/setup.bash。

（六）一些建议 1、关于调试：ROS 本身便具备大量的显示工具，如 rostopic，rqt_graph ，用于显示 topic 的内容，因此，推荐大家多加学习 ROS 相关操作，有利于调试。此外，coach 具备一定信息的实时显示功能，可以考虑使用。

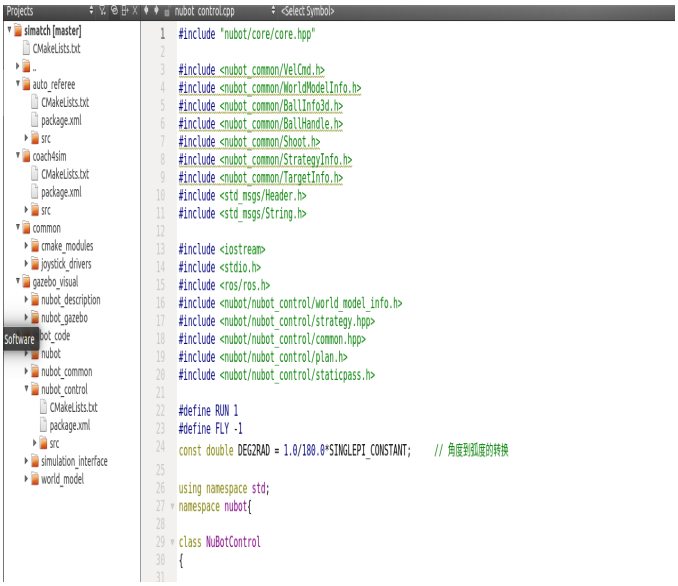

图 10: 工程结构显示

2、关于代码，主要更改部分为 robot_code 部分，gazebo、referee 部分不建议更改，因为在最终比赛时，使用同一版本。当然，对其进行修改以完成调试任务是可以进行的，如确有更改必要，请与组织方联系，告知所有参赛队并经过大多数参赛队同意后方可进行更改。最终比赛版本以最终发布为准。
3、在环境使用过程，如出现问题，可积极交流，以促进共同进步。

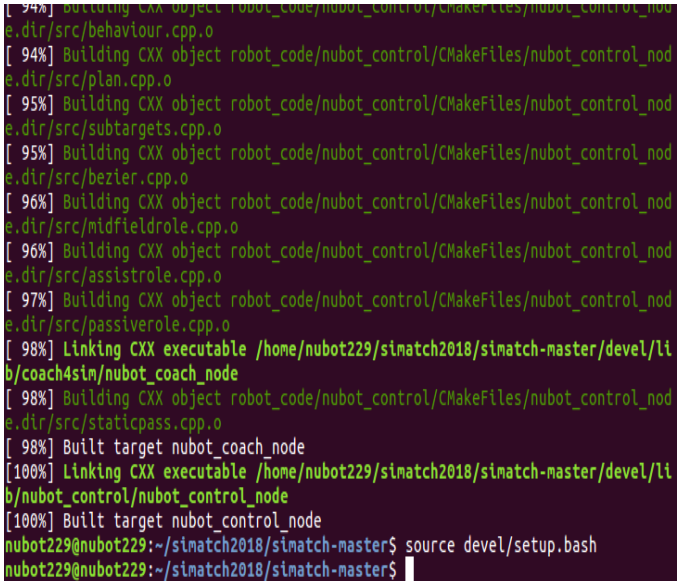

图 11: catkin_make 和 source

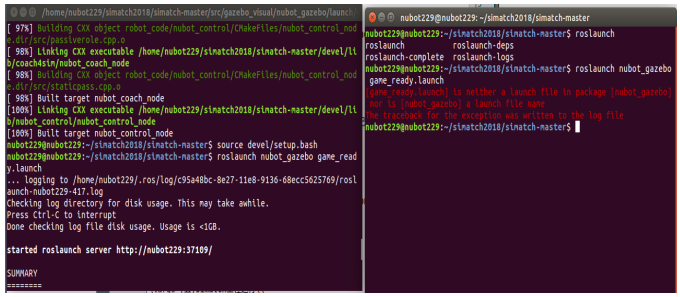

图 12: roslaunch 结果显示