

ROS集成开发环境RoboWare及其使用

高 佼 (Jeff)

济南汤尼机器人科技有限公司
Jinan Tonyrobotics Co., Ltd.

July 24, 2017

内容提要

- 1 RoboWare安装
 - 安装准备工作
 - 下载与安装
- 2 RoboWare本地开发
 - 代码编辑
 - 编译
 - 调试
 - ROS视图及包管理器
- 3 RoboWare远程开发
 - 远程选项配置
 - 远程部署及编译
 - 远程调试
- 4 开发示例
 - baymini工程本地开发及调试
 - baymini工程远程开发及调试
- 5 其它

安装准备工作

安装前,请查看系统环境并确认:

- 操作系统为 Ubuntu。
- 已完成 ROS 的安装配置。ROS 安装步骤可参照官方网站:
<http://wiki.ros.org/indigo/Installation/Ubuntu>
- 可使用catkin_make构建 ROS 包。(若无法构建,需要运行sudo apt-get install build-essential来安装构建工具。)
- 为支持 Python 相关功能,需要安装python-pip。
sudo apt-get install python-pip

下载

- 方式1: 到roboware官网<https://www.roboware.me>下载, 主页上有最新的64位和32位版本。
- 方式2: 到github项目页面下载<https://github.com/tonyrobotics/RoboWare>, 包含了最新版及所有历史版本的安装包及使用手册。

安装包为deb文件, 包含64位版和32位版。

`roboware-studio_VERSION_ARCHITECTURE.deb`

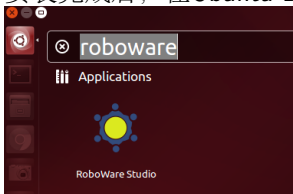
其命名规则为: VERSION表示版本号, ARCHITECTURE表示系统架构。如: `roboware-studio_0.7.0-1500621106_amd64.deb`表示0.7.0版本针对64位系统的安装包。

安装

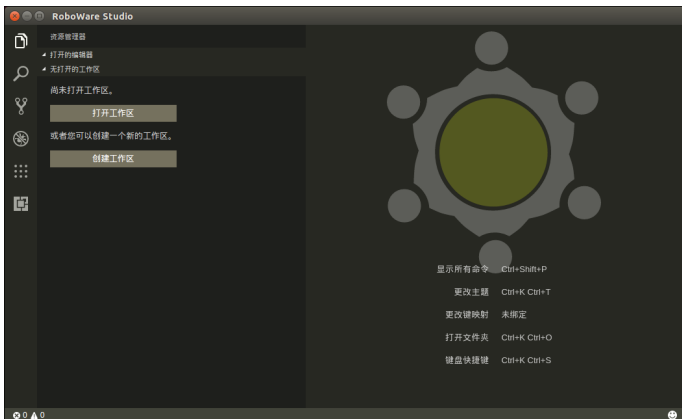
- 方式1：双击deb文件安装。
- 方式2：命令行方式安装：

```
$ cd /path/to/deb/file/  
$ sudo dpkg -i roboware-studio_0.7.0-1500621106_amd64.deb
```

安装完成后，在Ubuntu Dash中搜索RoboWare，并点击启动。



启动界面



升级和卸载

- 升级：下载最新的 RoboWare Studio 安装包 deb 文件,参照安装步骤直接安装即可,旧版本会被自动覆盖。
- 卸载：打开任一终端,执行以下指令卸载 RoboWare Studio:
\$ sudo apt-get remove roboware-studio

功能点

- 代码编辑
- 编译
- 调试
- ROS视图及包管理器

工作区、包、节点等的创建

C++代码

```
#include "ros/ros.h"
int main(int argc, char *argv[]){
    int a[5] = {2,3,5,7,11,13,17,19,23,29};
    ROS_INFO("hello_□ros!");
    for(int i = 1; i < argc; ++i){
        printf("arguments_□%d:□%s\n", i, argv[i]);
    }
    ROS_INFO("exit.");
    return 0;
}
```

工作区、包、节点等的创建

Python代码

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String
def talker():
    pub = rospy.Publisher('chatter', String, queue_size=10)
    rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        hello_str = "hello_world_%s" % rospy.get_time()
        rospy.loginfo(hello_str)
        pub.publish(hello_str)
        rate.sleep()
if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException:
        pass
```



代码跳转、自动补全

DEMO



代码片段 (snippets)

DEMO

- 基本语句：if、for、while、switch语句等
- ROS语句：ROS订阅者、发布者、初始化、循环等
- ROS节点

编译

DEMO

- 编译选项介绍
- 任意多个包的编译
- 清理

C++代码调试

DEMO

- 调试参数设置
- 断点设置，单步调试
- 调用堆栈

Python代码调试

DEMO



ROS视图及包管理器

DEMO

- .bashrc编辑
- ROS工具启动
- ROS活动节点、话题等查看
- ROS包管理

主要内容

- 远程选项配置
- 远程部署
- 远程编译
- 远程调试

生成公钥并拷贝至远程机器

首先，在本地计算机生成公钥和私钥。打开终端，执行命令：

```
$ ssh-keygen
```

一直按回车键选择默认选项，会在`~/.ssh`目录下生成 `id_rsa` 和 `id_rsa.pub` 两个文件。然后将 `id_rsa.pub` 文件复制到远程计算机：

```
$ scp ~/.ssh/id_rsa.pub username@ip_address:/home/user
```

远程机器添加公钥

将公钥文件 id_rsa.pub 拷贝到远程计算机后，SSH 登录到远程计算机：

```
$ ssh user@ip_address
```

登录后，将 id_rsa.pub 的文件内容追加写入到远程计算机的 ~/.ssh/authorized_keys 文件中，并修改 authorized_keys 文件的权限：

```
$ cat id_rsa.pub >> ~/.ssh/authorized_keys
```

```
$ chmod 600 ~/.ssh/authorized_keys
```

远程机器/etc/profile中导出ROS环境变量

首先，登录远程计算机：

```
$ ssh user@ip_address
```

其中 user 为远程计算机用户名， ip_address 为远程计算机的 IP 地址。

登录后，切换到 root 用户权限，将 ROS 环境变量信息写入到/etc/profile 文件中：

```
$ sudo su
```

```
$ echo "source /opt/ros/indigo/setup.bash" >> /etc/profile
```

远程部署及编译

DEMO

远程调试

DEMO

baymini工程本地开发及调试

DEMO

- RGB-D相机启动
- 激光启动
- rviz显示

baymini工程远程开发及调试

DEMO

- RGB-D相机远程启动
- 激光远程启动
- rviz显示

其它

DEMO

- 软件首选项
- ROS bag记录与播放
- Vim模式
- ...

The End