



外部设备的使用

朱笑笑

2018年4月



上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

外部设备的使用



- 机器人有控制器、传感器、执行器等关键部件组成。
- ROS必须要能够与硬件进行交互，来实现实际机器人的各项功能。
- 本节课将涉及到手柄、Kinect、2D laser、arduino



1

Joystick手柄的使用

2

Kinect的使用

3

2D Laser的使用

4

Arduino的使用



1

Joystick手柄的使用

2

Kinect的使用

3

2D Laser的使用

4

Arduino的使用



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

Joystick手柄的使用



- 游戏杆可以看成是一系列的按钮和电位器。通过这些装置，你能够实现或控制很多种运动模式。
- 在ROS 系统中，游戏杆能够通过改变速度和方向来远程控制机器人。
- ROS中joy包负责从手柄中读取信号并发布出来



/dev/input/js0

驱动

joy

/joy

- header:
 - seq: 157
 - stamp:
 - secs: 1357571648
 - nsecs: 430257462
 - frame id: ''
- axes : [0.0, 0.0, 0.0]
- buttons : [0,0,0,0,0,0,0,0,0,0,0,0]

Joystick手柄的使用



Joystick手柄的使用



▪ Step1. 手柄驱动ROS包、手柄例程包安装。

```
$ sudo apt-get install ros-kinectic-joy*
```

- 将手柄USB插入电脑，并检查设备文件，看是系统是否检测到合适的硬件。

```
$ ls /dev/input/
```

- 一般会输出以下内容，其中js0就是我们的手柄（joystick）

```
by-id  event0 event2 event4 event6 event8 js0  mouse0
```

```
by-path event1 event3 event5 event7 event9 mice
```

- 我们可以用手柄驱动自带的命令来测试手柄是否可以正常工作。

```
$ sudo jstest /dev/input/js0
```

- 有类似以下输出

```
Axes: 0: 0 1: 0 2: 0 Buttons: 0:off 1:off 2:off 3:off 4:off
```

```
5:off 6:off 7:off 8:off 9:off 10:off
```

Joystick手柄的使用



- Step2. 利用joy_node发出手柄的运动topic。

```
$ rosrun joy joy_node
```

- joy_node包可以将手柄的参数以topic的形式发出来，利用以下指令可以查看发出的topic信息。

```
$ rostopic echo /joy
```

```
---  
header:  
seq: 157  
stamp:  
  secs: 1357571648  
  nsecs: 430257462  
  frame id: "  
axes: [-0.0, -0.0, 0.0, -0.0, -0.0, 0.0, -0.0, -0.0]  
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
---
```


Joystick手柄的使用



- Step3. 利用joystick数据来在turtlesim中控制一个turtle
 - 写一个监听/joy的回调函数，然后向turtle1/command_velocity发送指令
 - 检查topic名字和topic中msg的类型（rostopic rosmmsg rosnode）
 - 包含对应的头文件
 - 消息类型要对应

1

Joystick手柄的使用

2

Kinect的使用

3

2D Laser的使用

4

Arduino的使用



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

Kinect的使用



- Kinect是微软开发的一款RGBD相机



Fast: 30FPS
Dense: 640*480 Points
accurate: 4%error in 4m
cheap: 150\$



RGB (color)
D (Depth)
Point Cloud

Kinect的使用



- Kinect有三个镜头，中间的镜头是 RGB 彩色摄影机，左右两边镜头则分别为红外线发射器和红外线 CMOS 摄影机所构成的 3D 深度感应器



Kinect的使用



- Kinect通过IR头投射一些“随机”点阵，然后只用了一个普通的CMOS传感器来捕捉这个点阵。简单来说，当场景的深度发生变化时，摄像头看到的点阵也会发生变化，通过这个变化就可以推断出深度信息。



- Kinect 的深度摄像头成像类似于普通的双目立体视觉，只要获取了两个摄像头之间的基线（baseline）和焦距（focal length）、以及视差数据，通过构造矩阵 Q ，利用 OpenCV 的 `reprojectImageTo3D` 函数，也可以计算出三维坐标。

Kinect的使用



- Step1.安装包和驱动。

```
$ sudo apt-get install ros-kinetic-openni-camera ros-kinetic-openni-launch
```

- 安装完成后插入Kinect传感器，运行node可以开始使用。首先运行roscore，然后在新的终端里运行以下命令。

```
$ roslaunch oppenni_launch oppenni.launch
```

Kinect的使用



- Step2.观察Kinect发出的数据
- 首先利用rostopic list命令显示所有topic，结果有很多，注意观察以下几个

```
...  
/camera/rgb/image_color  
/camera/rgb/image_mono  
/camera/rgb/image_raw  
/camera/rgb/image_rect  
/camera/rgb/image_rect_color  
...
```

- 利用image_view包查看图像数据

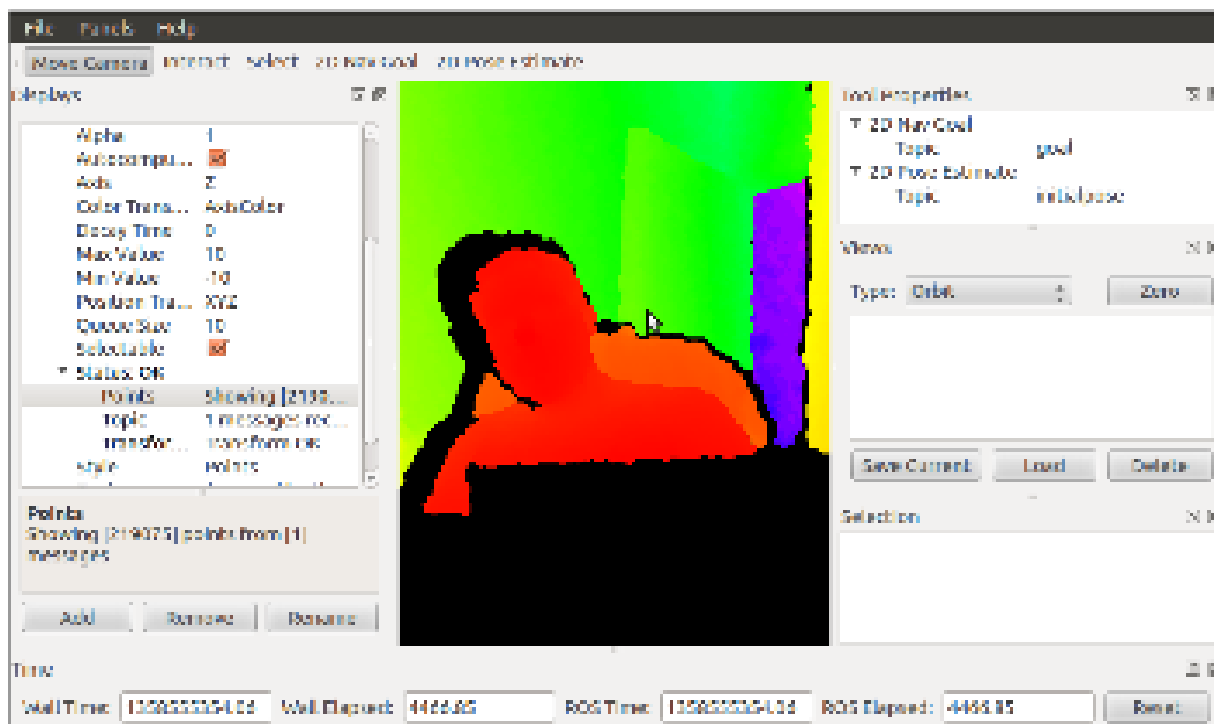
```
$ rosrun image_view image_view image:=/rgb/image_raw
```

Kinect的使用



- Step2.观察Kinect发出的数据

`$ rostopic type /camera/depth/points | rosmmsg show`



Kinect的使用



- Step3.利用Kinect发出的数据
- 利用PCL (Point Cloud Library) 库，对Kinect的点云进行下采样，减少点云密度。将处理后的点云以topic发出来，并用rviz观察。

```
#include <pcl/ros/conversions.h>+  
#include <pcl/point_cloud.h>+  
#include <pcl/point_types.h>+  
#include <pcl/filters/voxel_grid.h>+
```

```
sensor_msgs::PointCloud2 output;+  
pcl::VoxelGrid<sensor_msgs::PointCloud2> sor;+  
sor.setInputCloud(input);+  
sor.setLeafSize(0.02f,0.02f,0.02f);+  
sor.filter(output);+
```

```
CMakeLists.txt  
pcl_conversions  
pcl_ros
```

1

Joystick手柄的使用

2

Kinect的使用

3

2D Laser的使用

4

Arduino的使用



上海交通大学

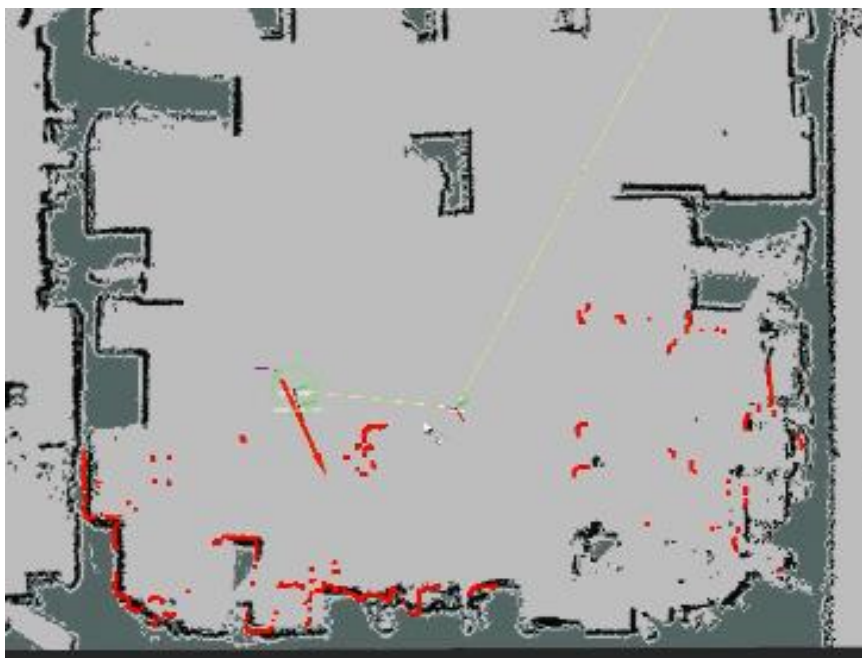
SHANGHAI JIAO TONG UNIVERSITY

2D laser



■ 2D激光传感器

- 能够感知周围环境中一个平面中的障碍物信息。广泛用于机器人的避障，定位，导航等。



2D laser



- 2D激光传感器

- 与Kinect的基于散斑图像3D信息技术不同，2D激光传感器使用的TOF的测距技术

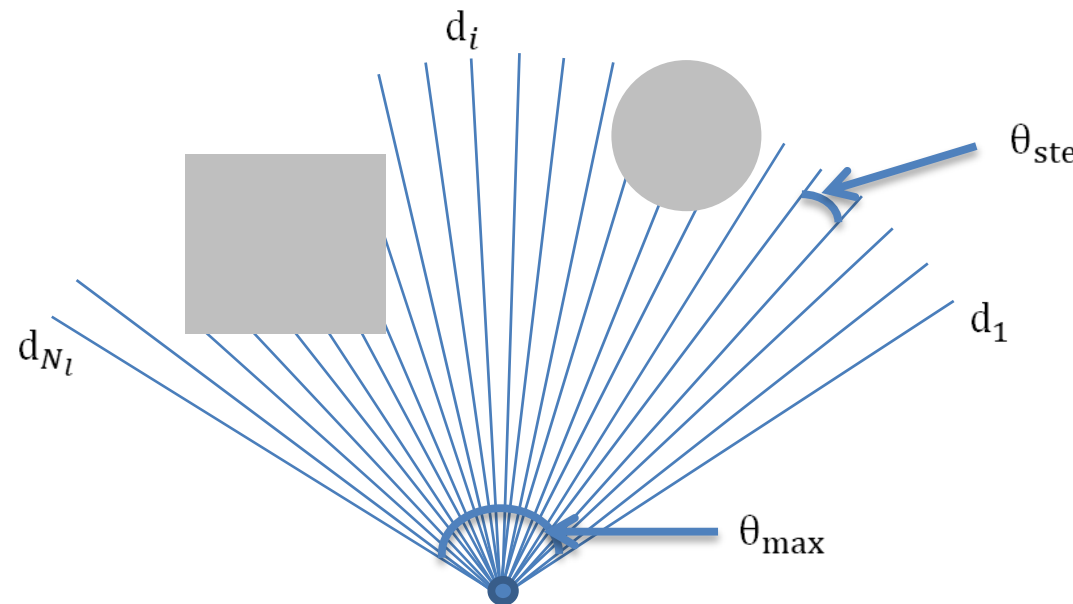


2D laser



■ 数据格式

- Intensity mode off:
 - cluster: 1
 - skip: 1
 - intensity: false
 - min_ang: -2.2689
 - max_ang: 2.2689
- Intensity mode on:
 - cluster: 1
 - skip: 1
 - intensity: true
 - min_ang: -1.047
 - max_ang: 1.047

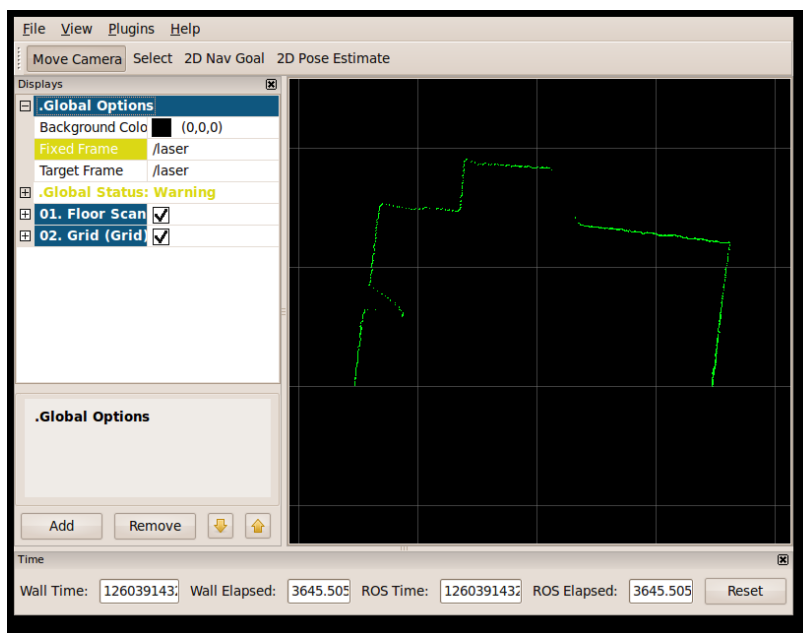


2D laser



■ 2D激光传感器

- 本课程中使用hokuyo传感器，ROS中使用laser-drivers, hokuyo_node包来处理激光传感器信息。
- 使用rviz可以现实Laser数据信息。 (录像)



1

Joystick手柄的使用

2

Kinect的使用

3

2D Laser的使用

4

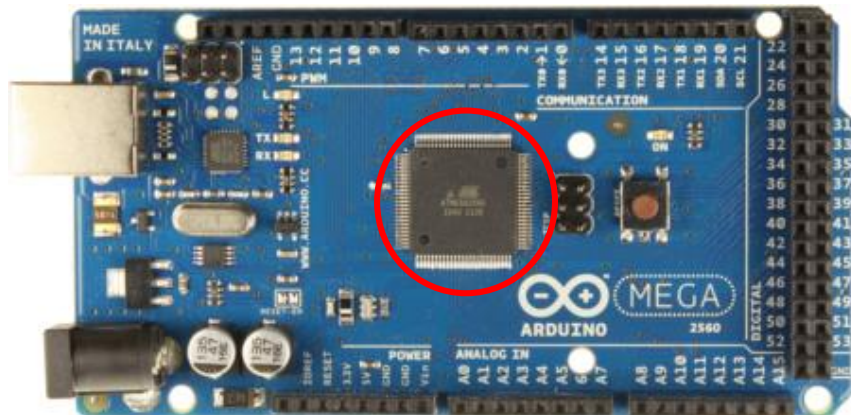
Arduino的使用



Arduino



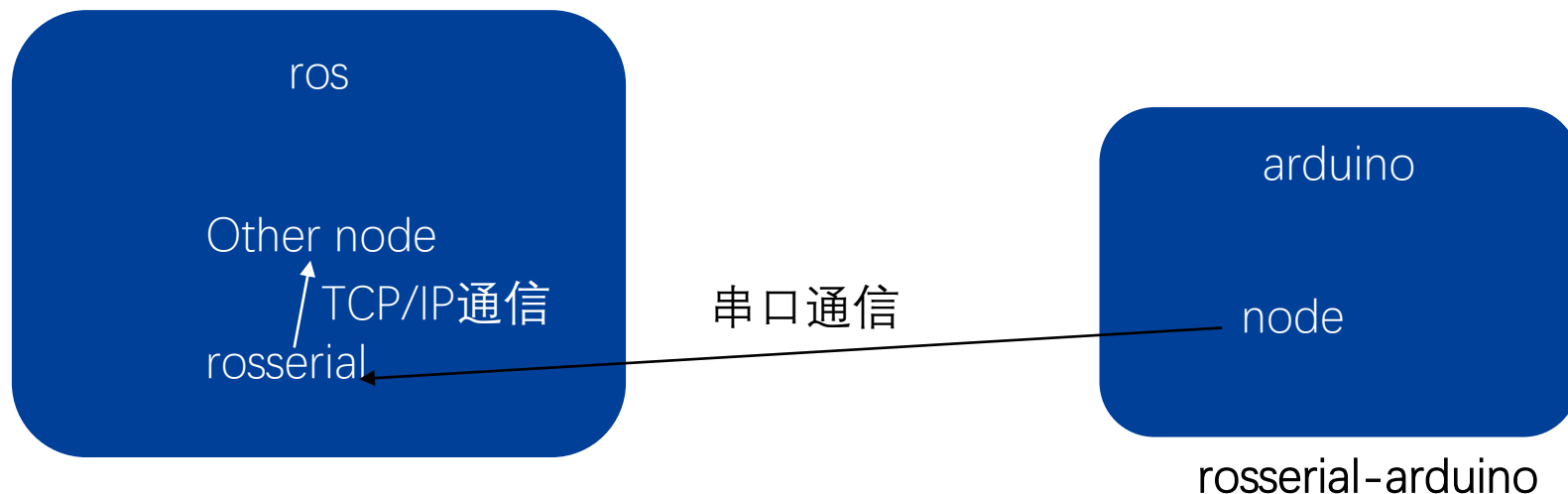
- Arduino, 是一个开放源代码的单芯片微控制器, 它使用了Atmel AVR单片机, 采用了开放源代码的软硬件平台, 建构于简易输出/输入 (simple I/O) 界面, 并且具有使用类似Java、C语言的Processing/Wiring开发环境。



Arduino



- Arduino可以连接各种廉价的硬件，接入ROS后可以大大扩展ROS的硬件支持。
- ROS中roserial能够处理Arduino的信息。
- ROS为Arduino提供了roserial-arduino



Arduino使用



- Step1. 安装roserial软件包及arduino IDE

```
$ sudo apt-get install ros-kinetic-roserial
```

```
$ sudo apt-get install ros-kinetic-roserial-arduino
```

- 下载和安装（解压后sh install.sh）Arduino IDE。完成安装后需要将roserial_arduino包中的ros_lib类库复制到Arduino的sketchbook文件目录下的library文件夹中。该sketchbook文件目录(~/.Arduino)将在首次运行Arduino IDE(./arduino)后自动产生。

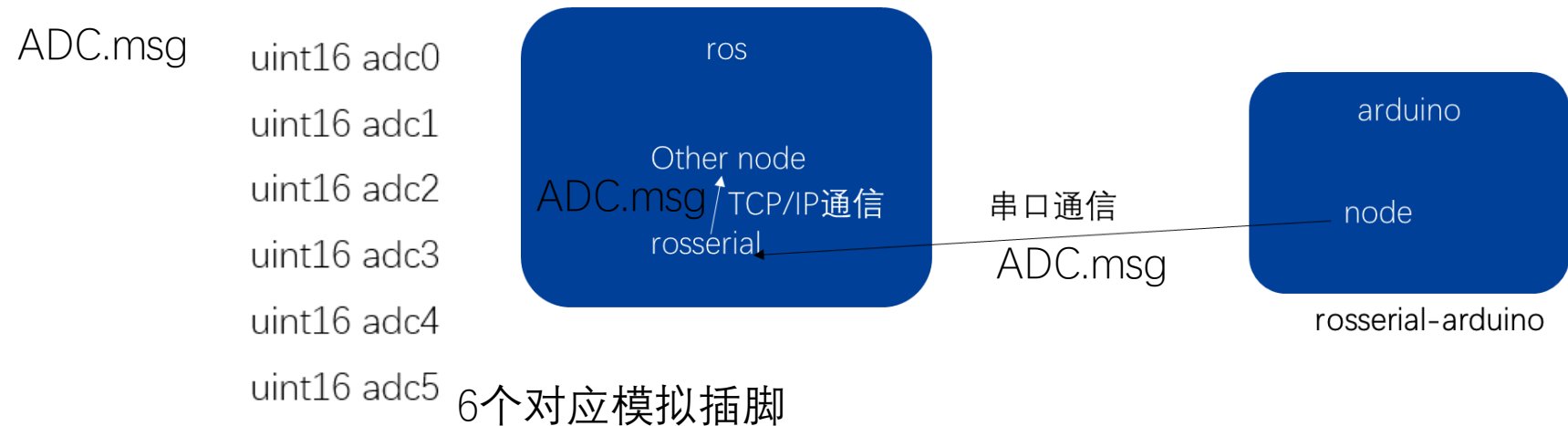
```
$ rosrn roserial_arduino make_libraries.py
```

```
~/.Arduino/libraries
```

Arduino示波器



- 说明：
- 这个教程展示通过Arduino的ADC(模拟到数字转换器)和rqt_plot制作简单的示波器
- 建立一个发布Arduino的6个ADC插脚的值通过topic传递到ROS



Arduino示波器



- 代码可以在Arduino IDE 的
File>Examples>ros_lib>ADC 找到
- 代码

```
#include <WProgram.h>
#include <ros.h>
#include <rosserial_arduino/Adc.h>
ros::NodeHandle nh;
rosserial_arduino::Adc adc_msg;
ros::Publisher p("adc", &adc_msg);
void setup()
{
  pinMode(13, OUTPUT);
  nh.initNode();
  nh.advertise(p);
}
//We average the analog reading to eliminate some of the noise
int averageAnalog(int pin){
  int v=0;
  for(int i=0; i<4; i++) v+= analogRead(pin);
  return v/4;
}
```

```
long adc_timer;

void loop()
{

  adc_msg.adc0 = averageAnalog(0);
  adc_msg.adc1 = averageAnalog(1);
  adc_msg.adc2 = averageAnalog(2);
  adc_msg.adc3 = averageAnalog(3);
  adc_msg.adc4 = averageAnalog(4);
  adc_msg.adc5 = averageAnalog(5);

  p.publish(&adc_msg);

  nh.spinOnce();
}
```


Arduino示波器



- Tools->Board:"Arduino/Genuino Uno"
- Tools->Port:"/dev/ttyACM0(Arduino/Genuino Uno)"

\$Sudo chmod 777 /dev/ttyACM0 //修改权限

- 编译下载 ➡

Arduino示波器



- 测试

- 新窗口打开

```
$ roscore
```

- 新窗口打开, /dev/ttyACM0为Arduino设备端口号

```
$ rosrun rosserial_python serial_node.py
```

```
_port:=/dev/ttyACM0 //转换arduino的信号发出adc的  
topic
```

- 新窗口打开

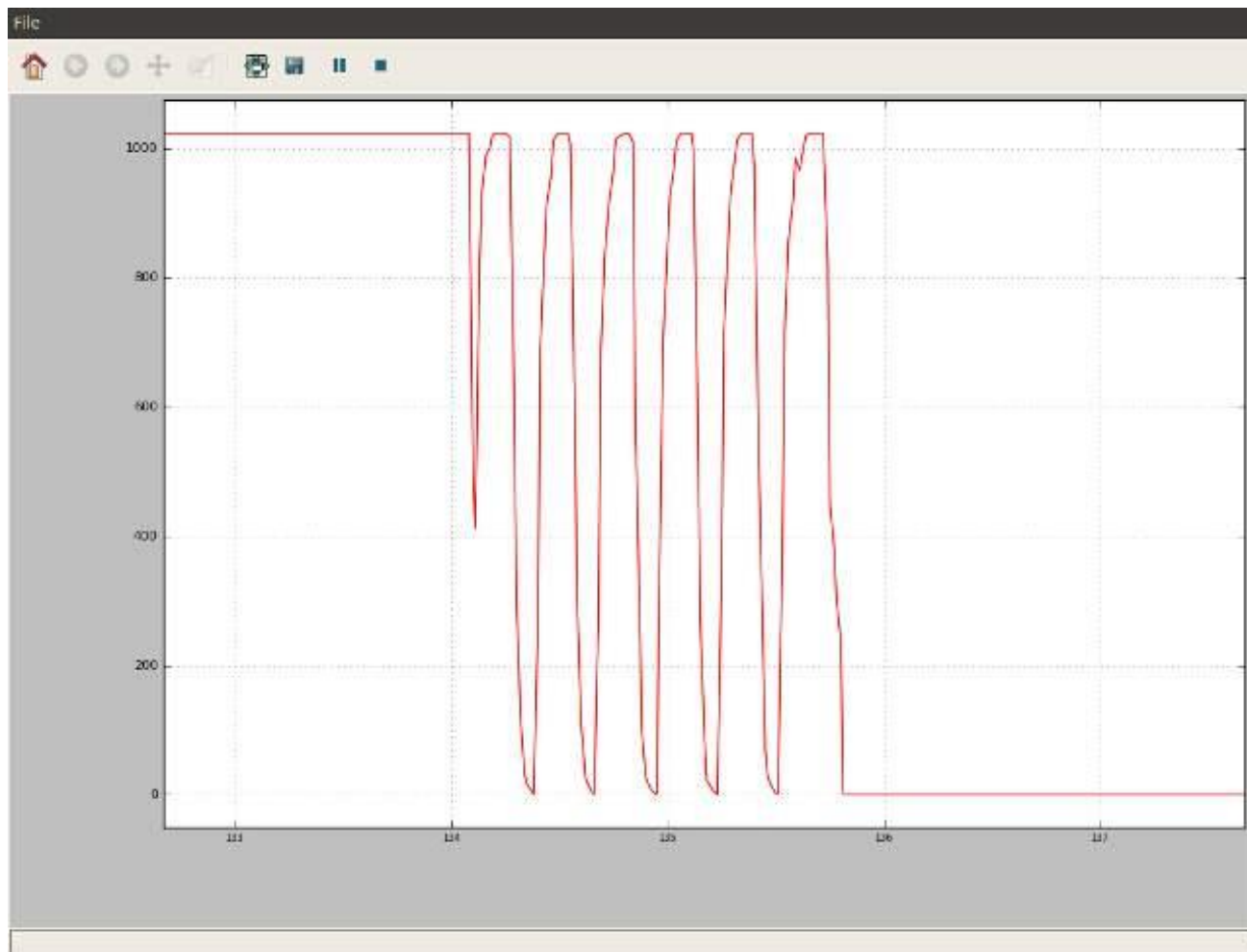
```
$ rqt_plot adc/adc0
```

Arduino示波器



- 在rqt_plot上应该看到adc0插脚模拟值
- 读数从0-1024， 因为Arduino有10-bit的ADC.
- 获取即时电压： $\text{adc_val} / 1024 * \text{Arduino voltage}$
- 尝试连接ADC0到Arduino的GND, 5V, and 3.3V并观察值的变化
- 但没有连接ADC， 将是随机浮动的数值

Arduino示波器



Arduino灯



- 利用ros_lib/Blink,写一个node控制灯的开关。

谢谢！

