

💡 Please ask about problems and questions regarding this tutorial on [answers.ros.org](http://answers.ros.org) (<http://answers.ros.org>). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

# Arduino Oscilloscope

**Description:** In this tutorial, we will be making a poor man's oscilloscope using the Arduino's ADC (analog to digital converter) and rqt\_plot.

**Tutorial Level:**

electric	fuerte	groovy	hydro	indigo	jade	kinetic
----------	--------	--------	-------	--------	------	---------

**Contents**

- 1. Custom Message
- 2. The Code
- 3. Running the Oscilloscope

In this tutorial, we will be making a poor man's oscilloscope using the Arduino's ADC (analog to digital converter) and rqt\_plot. We will set up a publisher which will forward the analog values from each of the Arduino's 6 ADC pins to ROS using roserial. This will be a relatively bad oscilloscope (very low sampling frequency and resolution), but it will be perfect for many quick and dirty setups.

The code for this tutorial can be found in your Arduino IDE at File>Examples>ros\_lib>ADC

If you have not done so already, make sure to clone the roserial stack and add it to your ROS\_PACKAGE\_PATH.

## 1. Custom Message

This code uses a custom message type, the ADC.msg. This message has uint16 field for each of the analog input pins on the standard Arduino.

```
uint16 adc0
uint16 adc1
uint16 adc2
uint16 adc3
uint16 adc4
uint16 adc5
```

This message is a part of the roserial\_arduino package and has been added to your ros\_lib library.

## 2. The Code

The code for this tutorial is fairly straight forward. The code registers a Adc msg publisher, initializes the node handle, and then tries to publish as many Adc messages as possible.

Toggle line numbers

```

1  /*
2   * roserial ADC Example
3   *
4   * This is a poor man's Oscilloscope.  It does not have the sampling
5   * rate or accuracy of a commerical scope, but it is great to get
6   * an analog value into ROS in a pinch.
7   */
8
9  #include <WProgram.h>
10 #include <ros.h>
11 #include <roserial_arduino/Adc.h>
12
13 ros::NodeHandle nh;
14
15 roserial_arduino::Adc adc_msg;
16 ros::Publisher p("adc", &adc_msg);
17
18
19 void setup()
20 {
21     pinMode(13, OUTPUT);
22     nh.initNode();
23
24     nh.advertise(p);
25 }
26
27 //We average the analog reading to eliminate some of the noise
28 int averageAnalog(int pin){
29     int v=0;
30     for(int i=0; i<4; i++) v+= analogRead(pin);
31     return v/4;
32 }
33
34 long adc_timer;
35
36 void loop()
37 {
38
39     adc_msg.adc0 = averageAnalog(0);
40     adc_msg.adc1 = averageAnalog(1);
41     adc_msg.adc2 = averageAnalog(2);
42     adc_msg.adc3 = averageAnalog(3);
43     adc_msg.adc4 = averageAnalog(4);
44     adc_msg.adc5 = averageAnalog(5);
45
46     p.publish(&adc_msg);
47
48     nh.spinOnce();
49 }

```

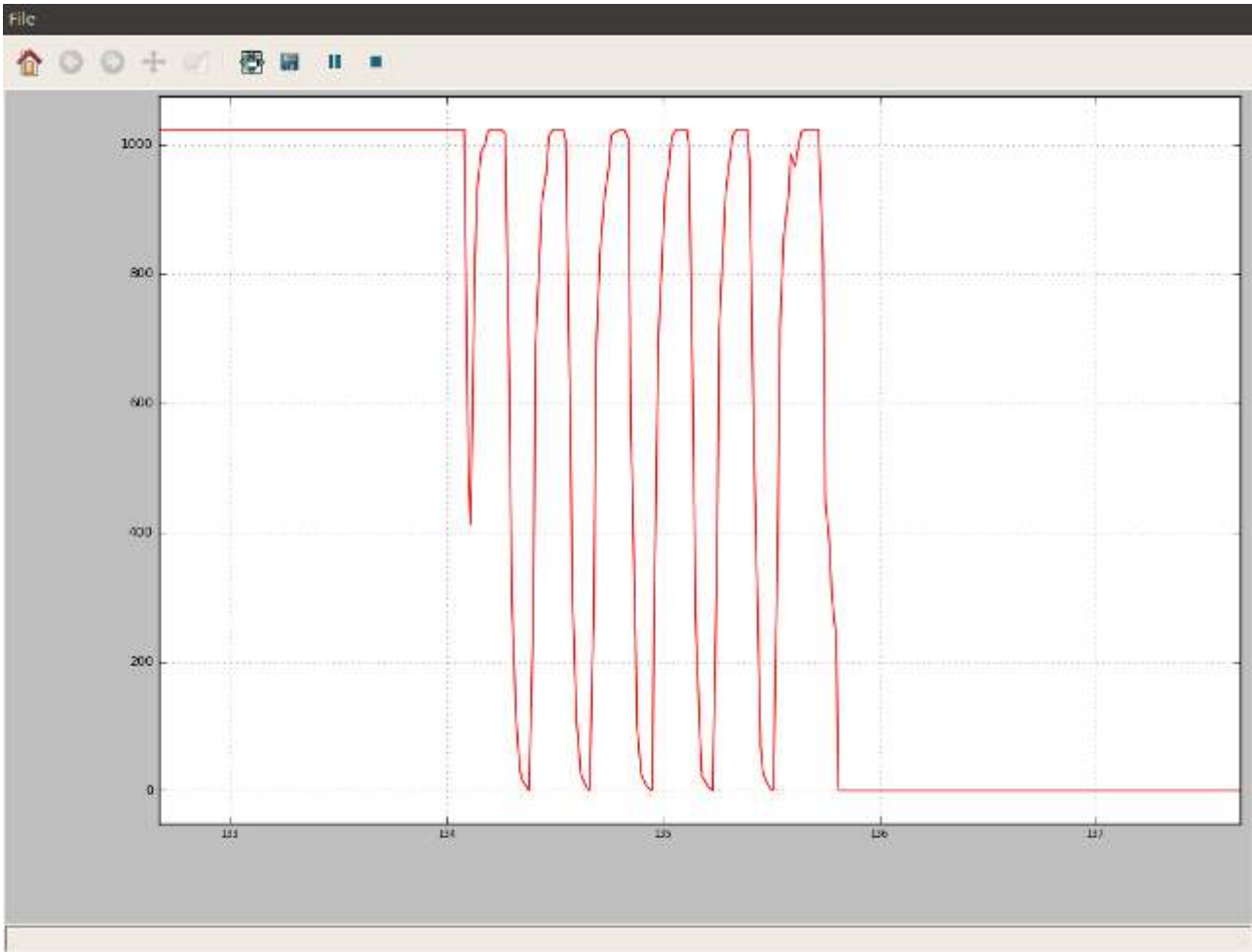
### 3. Running the Oscilloscope

```
roscore
```

```
rqt_plot adc/adc0
```

Now, you should see the raw analog value on adc0 displayed in rqt\_plot (/rqt\_plot). The reported readings are going to be from 0-1024 because the Arduino has a 10-bit ADC. In order to get the actual voltage,  $\text{adc\_val} / 1024 * \text{Arduino supply voltage}$ . Try connecting ADC0 to ground, 5V, and 3.3V on your arduino and observing the value. Also notice that when you don't have anything connected to the ADC, it is a floating analog input and its value will oscillate randomly.

For example, look at the below output where the ADC0 input is switched between 5V and gnd. There are wild oscillations during the transition.



Except where otherwise noted, the ROS wiki is licensed under the Wiki: rosserial\_arduino/Tutorials/Arduino Oscilloscope (last edited 2014-09-22 16:30:32 by AustinHendrix (/AustinHendrix))  
Creative Commons Attribution 3.0  
(<http://creativecommons.org/licenses/by/3.0/>) | Find us on Google+ (<https://plus.google.com/113789706402978299308>)

Brought to you by:  Open Source Robotics Foundation  
(<http://www.osrfoundation.org>)