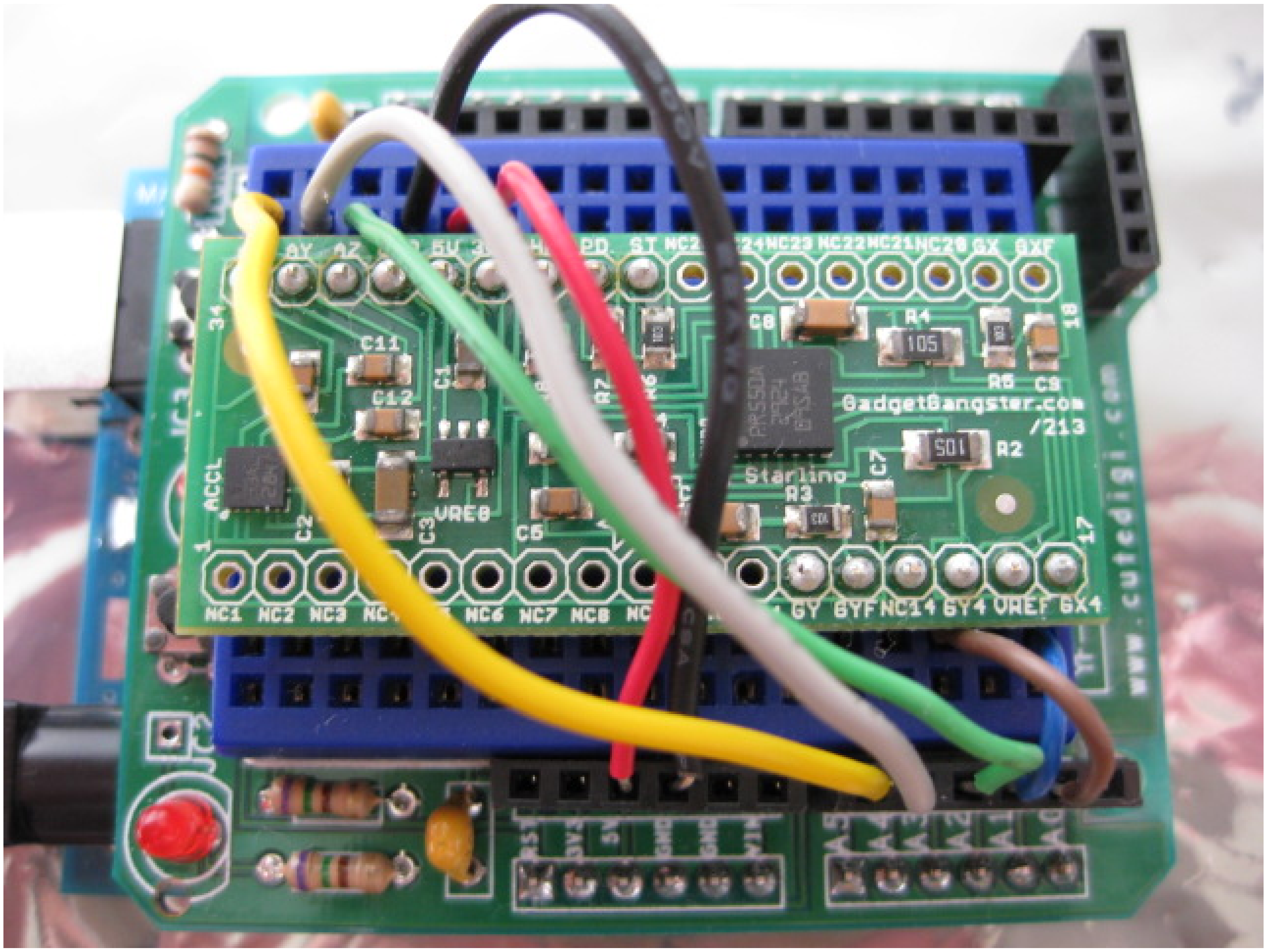
- STARLINO
- Articles
- Store
  - Acc_Gyro 6DOF Analog IMU: 3 Axis Accelerometer + 3 Axis Gyro
  - Single Axis Analog Gyroscope with Noise and Drift Filters
  - Usb Thumb Sized Pic Development Platform (PIC18F14K50)
- About
- Contact
- Books & Tools

Arduino code for IMU Guide algorithm. Using a 5DOF IMU (accelerometer and gyroscope combo)

Posted on: January 22, 2010 by *starlino*
Category: IMU Theory and Experiments
Tags: accelerometer, arduino, filter, gyroscope, imu

This article introduces an implementation of a simplified filtering algorithm that was inspired by Kalman filter. The Arduino code is tested using a 5DOF IMU unit from GadgetGangster – Acc_Gyro . The theory behind this algorithm was first introduced in my Imu Guide article.

The Acc_Gyro is mounted on a regular proto-shield on top of an Arduino Duemilanove board.



Parts needed to complete the project:

- Arduino Duemilanove (or similar Arduino platform)
- Acc_Gyro IMU board
- Protoshield (optional)
- Breadboard
- Hook-up wire 22AWG

The hook-up diagram is as follows:

Acc_Gyro	<—>	**Arduino**
5V	<—>	5V
GND	<—>	GND
AX	<—>	AN0
AY	<—>	AN1
AZ	<—>	AN2
GX4	<—>	AN3
GY4	<—>	AN4

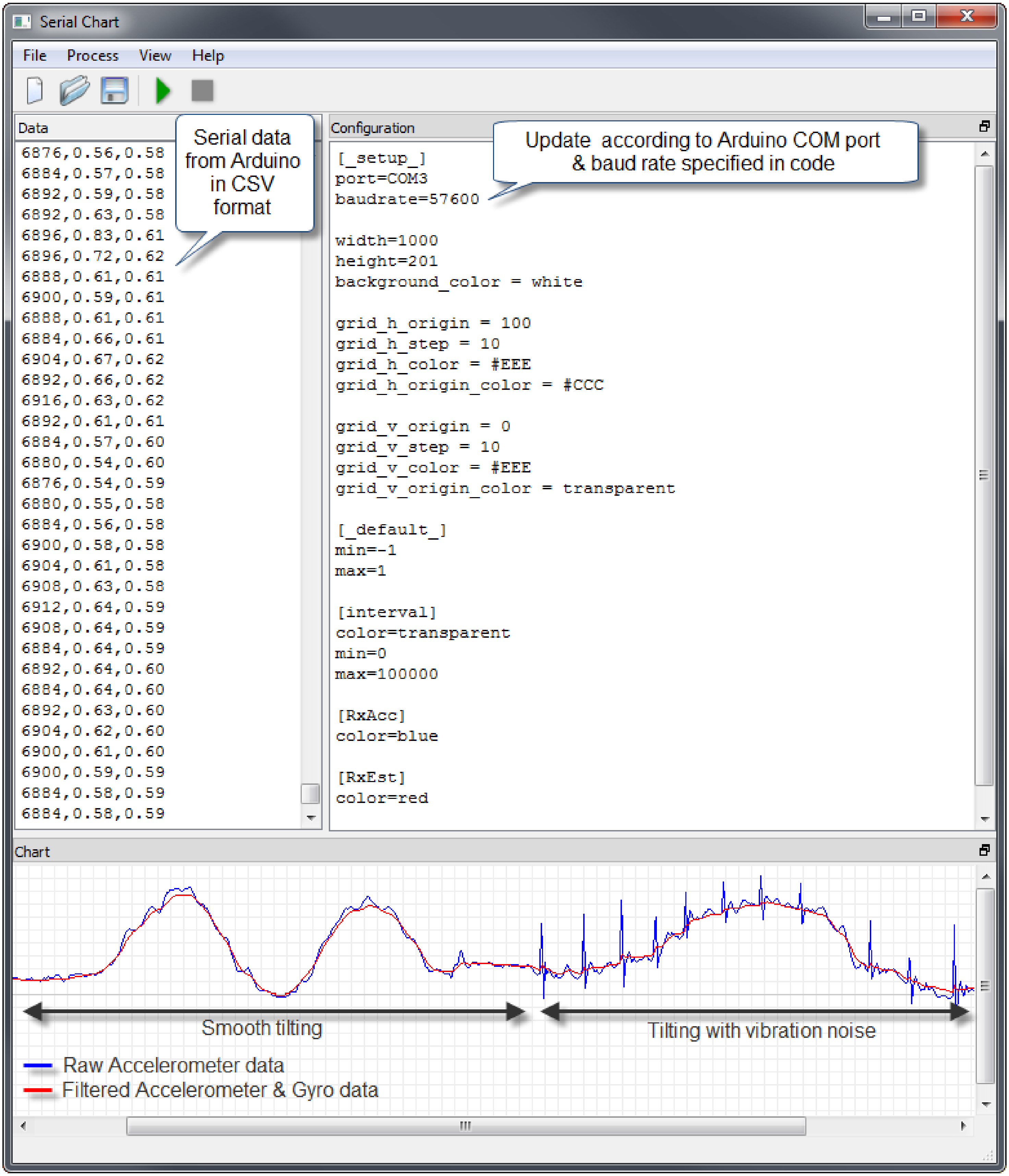
Once you have completed the hardware part, load the following sketch to your Arduino.

imu_arduino.zip

Run the project and make sure you are receiving an output on your serial terminal (you can start the terminal from your Arduino IDE).

To analyze the data I have developed a small utility called SerialChart. It is open-source so feel free to customize it for your own needs.

Here is the output from SerialChart software:



Serial data from Arduino in CSV format

```
6876,0.56,0.58
6884,0.57,0.58
6892,0.59,0.58
6892,0.63,0.58
6896,0.83,0.61
6896,0.72,0.62
6888,0.61,0.61
6900,0.59,0.61
6888,0.61,0.61
6884,0.66,0.61
6904,0.67,0.62
6892,0.66,0.62
6916,0.63,0.62
6892,0.61,0.61
6884,0.57,0.60
6880,0.54,0.60
6876,0.54,0.59
6880,0.55,0.58
6884,0.56,0.58
6900,0.58,0.58
6904,0.61,0.58
6908,0.63,0.58
6912,0.64,0.59
6908,0.64,0.59
6884,0.64,0.59
6892,0.64,0.60
6884,0.64,0.60
6892,0.63,0.60
6904,0.62,0.60
6900,0.61,0.60
6900,0.59,0.59
6884,0.58,0.59
6884,0.58,0.59
```

Update according to Arduino COM port & baud rate specified in code

```
[_setup_]
port=COM3
baudrate=57600

width=1000
height=201
background_color = white

grid_h_origin = 100
grid_h_step = 10
grid_h_color = #EEE
grid_h_origin_color = #CCC

grid_v_origin = 0
grid_v_step = 10
grid_v_color = #EEE
grid_v_origin_color = transparent

[_default_]
min=-1
max=1

[interval]
color=transparent
min=0
max=100000

[RxAcc]
color=blue

[RxEst]
color=red
```

Smooth tilting
Tilting with vibration noise
Raw Accelerometer data
Filtered Accelerometer & Gyro data

The test was performed as follows:

- first I was tilting the board slowly (marked "smooth tilting" on the screenshot)

- next I continued tilting the board, but I also started applying some vibration – by tapping the board quickly with my finger (marked "Titlting with vibration noise")

As you can see from the chart the filtered signal (red line) is indeed more immune to noise than the accelerometer readings alone (blue line). The filtered signal was obtained by combining the Accelerometer and Gyroscope data. Gyroscope data is important, because if you would simply average the Accelerometer data you would get a delayed signal. Given the simplicity of the code and of the algorithm I am satisfied with the results. One feature that I would like to add is compensation for the drift effect that you might encounter with some gyroscopes. However the Acc_Gyro board proved to be very stable in this respect, since it has built-in high pass filters.

If you'd like to experiment on your own, I recommended first reproducing this testing setup , then shift slowly towards your application needs. For example you may take the C code and port it to PIC's C18/C30 or AVR-GCC (it shouldn't be too dificult).

Below are some useful resources and their descriptions.

SerialChart executables can be downloaded from here:

SerialChart_01.zip

Once you start SerialChart application you will need to load the **imu_arduino.scc** configuration file for this project(included in the imu_arduino.zip) archive.

In this configuration file make sure to update the 'port' settings to Arduino's COM port. On my computer Arduino was detected on COM3, on yours it might be different.

For more information on configuration file syntax see:

http://code.google.com/p/serialchart/wiki/ConfigurationFileSyntax

You can also download and compile SerialChart from Google Code:

http://code.google.com/p/serialchart/source/browse/#svn/trunk

You will need a SVN client to checkout the code (I use RapidSVN for Windows).

SerialChart was developed using Qt SDK from Nokia: http://qt.nokia.com/downloads

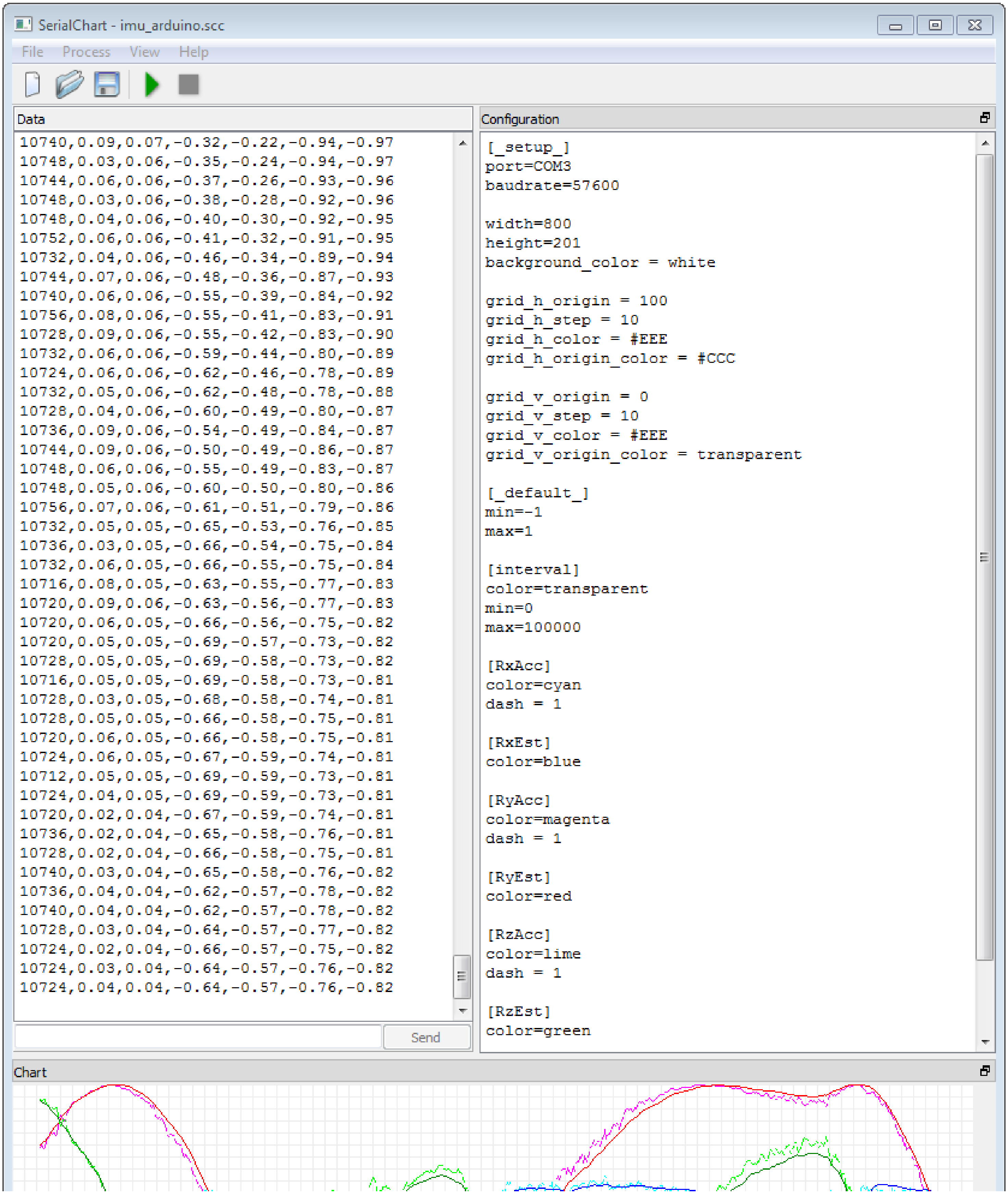
UPDATE 2010-03-18

Many people ask me what about the other 2 axis, here is the code that outputs 3 axis, including the SerialChart configuration script.

Imu_Arduino_3axis_output_2010-03-18.zip

I also removed some overhead code that Alex pointed out in the comments, this reduced the interval between samples.

In the example below I rotate the board around the X axis(blue) which is parallel to the ground.I do it by hand so X is not exactly 0, but close. The axes that change are Y(red) and Z(green). Please note the relationship X^2+Y^2+Z^2 = 1. The dashed cyan, magenta and lime lines are unfiltered signals coming from accelerometer alone (RwAcc).



```
10740,0.09,0.07,-0.32,-0.22,-0.94,-0.97
10748,0.03,0.06,-0.35,-0.24,-0.94,-0.97
10744,0.06,0.06,-0.37,-0.26,-0.93,-0.96
10748,0.03,0.06,-0.38,-0.28,-0.92,-0.96
10748,0.04,0.06,-0.40,-0.30,-0.92,-0.95
10752,0.06,0.06,-0.41,-0.32,-0.91,-0.95
10732,0.04,0.06,-0.46,-0.34,-0.89,-0.94
10744,0.07,0.06,-0.48,-0.36,-0.87,-0.93
10740,0.06,0.06,-0.55,-0.39,-0.84,-0.92
10756,0.08,0.06,-0.55,-0.41,-0.83,-0.91
10728,0.09,0.06,-0.55,-0.42,-0.83,-0.90
10732,0.06,0.06,-0.59,-0.44,-0.80,-0.89
10724,0.06,0.06,-0.62,-0.46,-0.78,-0.89
10732,0.05,0.06,-0.62,-0.48,-0.78,-0.88
10728,0.04,0.06,-0.60,-0.49,-0.80,-0.87
10736,0.09,0.06,-0.54,-0.49,-0.84,-0.87
10744,0.09,0.06,-0.50,-0.49,-0.86,-0.87
10748,0.06,0.06,-0.55,-0.49,-0.83,-0.87
10748,0.05,0.06,-0.60,-0.50,-0.80,-0.86
10756,0.07,0.06,-0.61,-0.51,-0.79,-0.86
10732,0.05,0.05,-0.65,-0.53,-0.76,-0.85
10736,0.03,0.05,-0.66,-0.54,-0.75,-0.84
10732,0.06,0.05,-0.66,-0.55,-0.75,-0.84
10716,0.08,0.05,-0.63,-0.55,-0.77,-0.83
10720,0.09,0.06,-0.63,-0.56,-0.77,-0.83
10720,0.06,0.05,-0.66,-0.56,-0.75,-0.82
10720,0.05,0.05,-0.69,-0.57,-0.73,-0.82
10728,0.05,0.05,-0.69,-0.58,-0.73,-0.82
10716,0.05,0.05,-0.69,-0.58,-0.73,-0.81
10728,0.03,0.05,-0.68,-0.58,-0.74,-0.81
10728,0.05,0.05,-0.66,-0.58,-0.75,-0.81
10720,0.06,0.05,-0.66,-0.58,-0.75,-0.81
10724,0.06,0.05,-0.67,-0.59,-0.74,-0.81
10712,0.05,0.05,-0.69,-0.59,-0.73,-0.81
10724,0.04,0.05,-0.69,-0.59,-0.73,-0.81
10720,0.02,0.04,-0.67,-0.59,-0.74,-0.81
10736,0.02,0.04,-0.65,-0.58,-0.76,-0.81
10728,0.02,0.04,-0.66,-0.58,-0.75,-0.81
10740,0.03,0.04,-0.65,-0.58,-0.76,-0.82
10736,0.04,0.04,-0.62,-0.57,-0.78,-0.82
10740,0.04,0.04,-0.62,-0.57,-0.78,-0.82
10728,0.03,0.04,-0.64,-0.57,-0.77,-0.82
10724,0.02,0.04,-0.66,-0.57,-0.75,-0.82
10724,0.03,0.04,-0.64,-0.57,-0.76,-0.82
10724,0.04,0.04,-0.64,-0.57,-0.76,-0.82
```

```
[_setup_]
port=COM3
baudrate=57600

width=800
height=201
background_color = white

grid_h_origin = 100
grid_h_step = 10
grid_h_color = #EEE
grid_h_origin_color = #CCC

grid_v_origin = 0
grid_v_step = 10
grid_v_color = #EEE
grid_v_origin_color = transparent

[_default_]
min=-1
max=1

[interval]
color=transparent
min=0
max=100000

[RxAcc]
color=cyan
dash = 1

[RxEst]
color=blue

[RyAcc]
color=magenta
dash = 1

[RyEst]
color=red

[RzAcc]
color=lime
dash = 1

[RzEst]
color=green
```