

# CPE 301      Microprocessor System Design Lab

LAB #EC01    (Extra Credit Lab)

Fall 2013

**This extra credit Lab must be completed and demonstrated to Andy before 3pm on November 15 in order to receive the extra credit toward your course grade.**

## ***Objective:***

To learn how to program the ATmega Analog to Digital converters by initializing and using the appropriate special function registers. To accomplish this you will modify an Arduino C language program which reads the Analog to Digital converters so that it will run on the Arduino Mega SBC without using any of the Arduino library functions.

## ***Procedure:***

1. Go to the web site: <http://www.x-io.co.uk/serial-oscilloscope/> and read the text and watch the video. A screen shot of this site is included below.
2. At the bottom of the web site page (not shown in the screen shot) are two links to download the oscilloscope display program which runs on Windows and the Arduino source code for PrintADC which runs on the Arduino UNO. Download both of these to your computer. A screenshot of the software running (as downloaded with no changes) on my office PC and an Arduino UNO SBC is included below.
3. Modify the program to remove ALL Arduino Library functions and substitute your own functions which control the ATmega special function registers.
4. If necessary, make any changes required to run on the Arduino Mega SBC.

## ***Notes:***

1. **ALL programs written for this course need to be documented in a complete manner which will make sense to you if you need to go back and review the program a year or two from now. You need to include your NAME and a revision number on ALL programs as well.**



**Home**

good place to start

**News**

keep up to date

**Products**

purchase online

**Open Source**

projects & source code

**About x-io**

who we are

**Contact**

any questions?

[Home](#) > [Open Source](#) > Serial oscilloscope

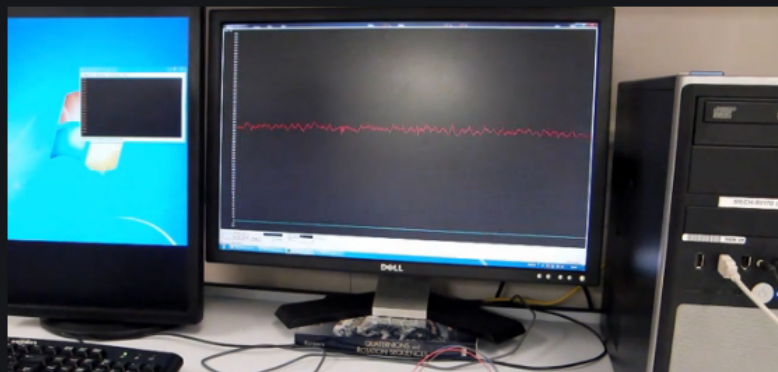
## Serial oscilloscope

Posted on April 6th, 2013 by Seb Madgwick

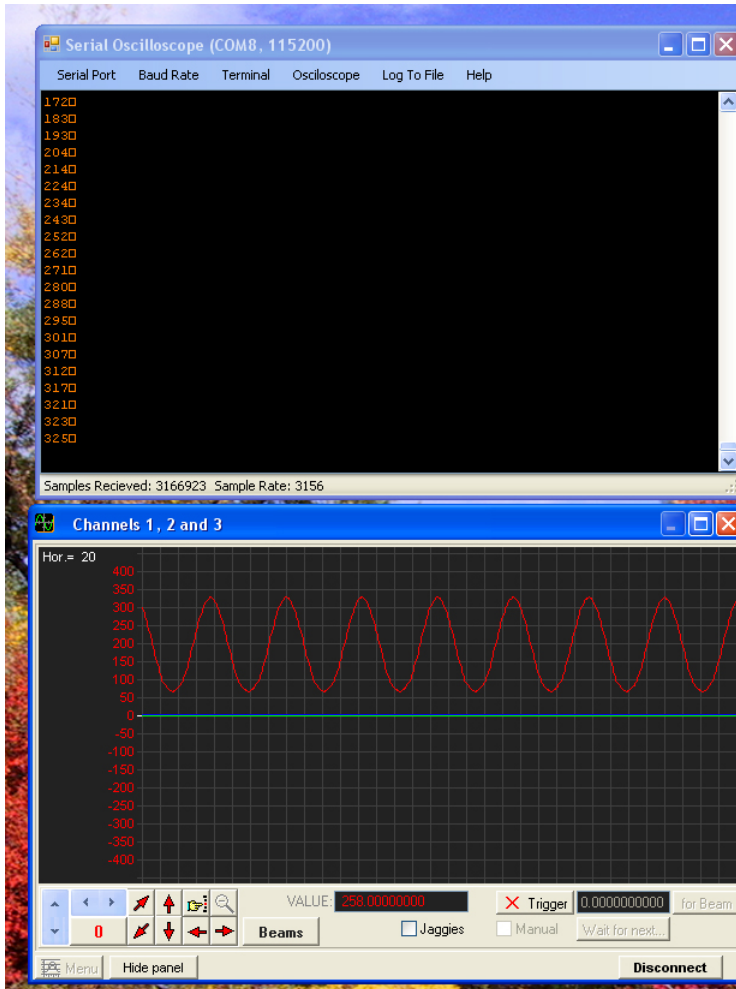
Printing data to a serial terminal is a useful debugging tool when developing embedded systems but often the data of interest is a continuous stream of numbers that is of little use displayed as text. For example, this might be data from a sensor or real-time statistics such as bandwidth performance. Serial Oscilloscope is a Windows application that plots comma-separated variables within any incoming serial stream as channels on a real-time oscilloscope. The application also functions as a basic serial terminal, received bytes are printed to the terminal and typed characters are transmitted. The project uses Michael Bernstein's [oscilloscope library](#) to plot up to 9 channels on 3 different oscilloscope with view and trigger menus.

Serial Oscilloscope is compatible with any serial stream containing comma-separated values terminated by a new-line character ("`\n`"). For example, "`11,22,33\n`" will be interpreted as values 11, 22 and 33 for channels 1, 2 and 3 respectively. The serial stream can also include non numerical characters which will be ignored. For example, "`a=0.5,blue,x=3.14,t1t2t3,8\n`" will be interpreted as values 0.5, 3.14, 123 and 8 for channels 1, 2, 3 and 4 respectively.

The [open-source resources](#) also include an Arduino sketch to send analogue input values over serial. In the video I use the Arduino and Serial Oscilloscope to plot data from an [IR distance sensor](#), a [triple-axis accelerometer](#) and a [microphone](#).



Downloaded software running on:  
UNR Windows XP PC with  
Arduino UNO attached.



The image shows an Arduino IDE window titled "PrintADC | Arduino 1.0.5". The window contains the source code for a sketch named "PrintADC.ino". The code includes comments and function definitions. The "setup" function is defined, and it includes calls to "digitalWrite(A0, HIGH);" and "digitalWrite(A1, HIGH);". Below the code, a status bar indicates "Done uploading." and "Binary sketch size: 2,718 bytes (of a 32,256 byte maximum)". The bottom status bar shows "1" and "Arduino Uno on COM8".

```
/*  
ArduinoPrintADC.ino  
  
Author: Seb Madgwick  
  
Sends up to all 6 analogue inputs values in ASCII as comma separate  
over serial. Each line is terminated with a carriage return charact  
The number of channels is sent by sending a character value of '1'  
the Arduino.  
Tested with "arduino-1.0.3" and "Arduino Uno".  
  
*/  
  
#include <stdlib.h> // div, div_t  
  
void setup() {  
  
    // Enable pull-ups to avoid floating inputs  
    digitalWrite(A0, HIGH);  
    digitalWrite(A1, HIGH);  
}
```