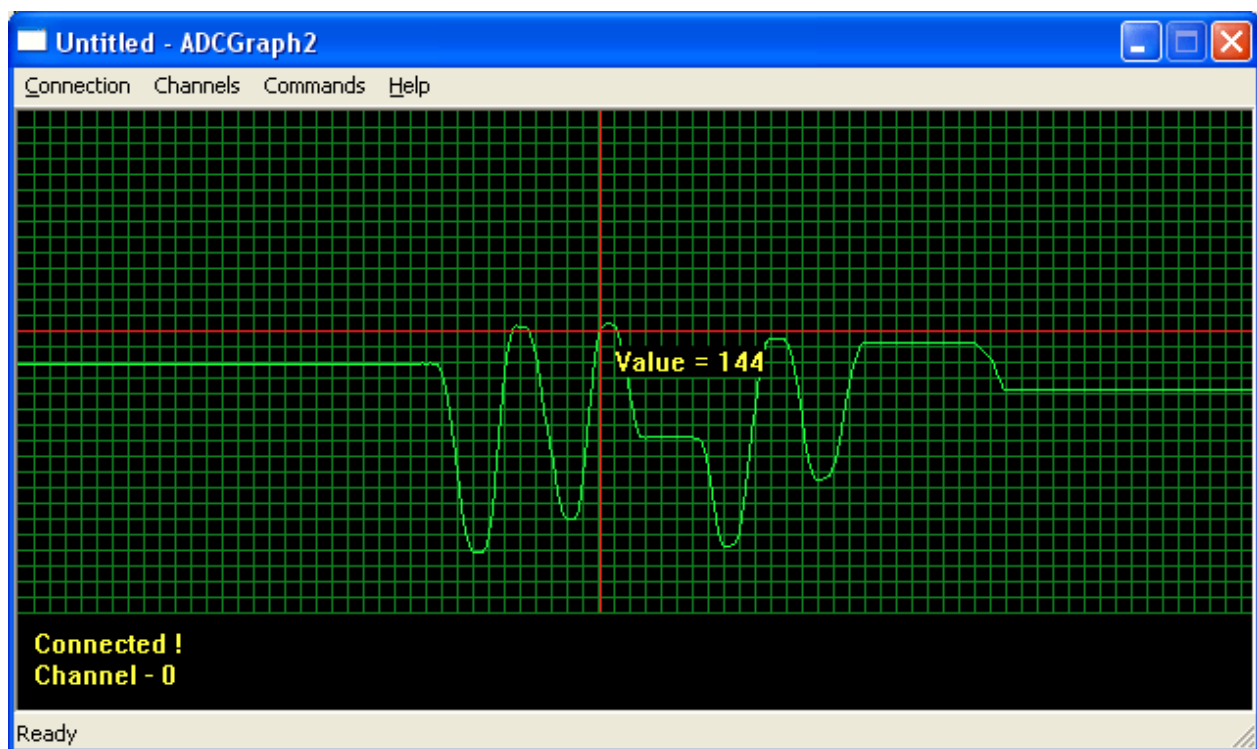


ADCGraph2 - Visualize ADC Data on PC Screen !

ADC ([Analog to digital converter](#)) is a commonly used peripheral. We use it everyday to interface with several analog sensors. Many times a nice visualization of ADC data is required during learning about new sensors. For example you just bought a analog sound sensor, and try to do some fast experiments that how it performs. In this situation this valuable tool will come to your help. The tool I called is **ADCGraph2**, it is being with me from a long time but today I thought of sharing it publicly. I wrote this tool somewhere 4 years back to experiment with sensors like LDRs, Temperature Sensor, Audio input etc. The public version is improved form of my in-house tool.

The ADCGraph2 is a simple Windows Based Tool that connect with an ATmega32 microcontroller (running a special firmware) to receive ADC data from any selected channels (0-7) and show a nice graph on PC screen. You can connect the output of any analog sensor to any ADC input pin and view the sensors output on PC screen in form of a nice graph. The connection is over USART Port so you need a [USB to UART Bridge](#).

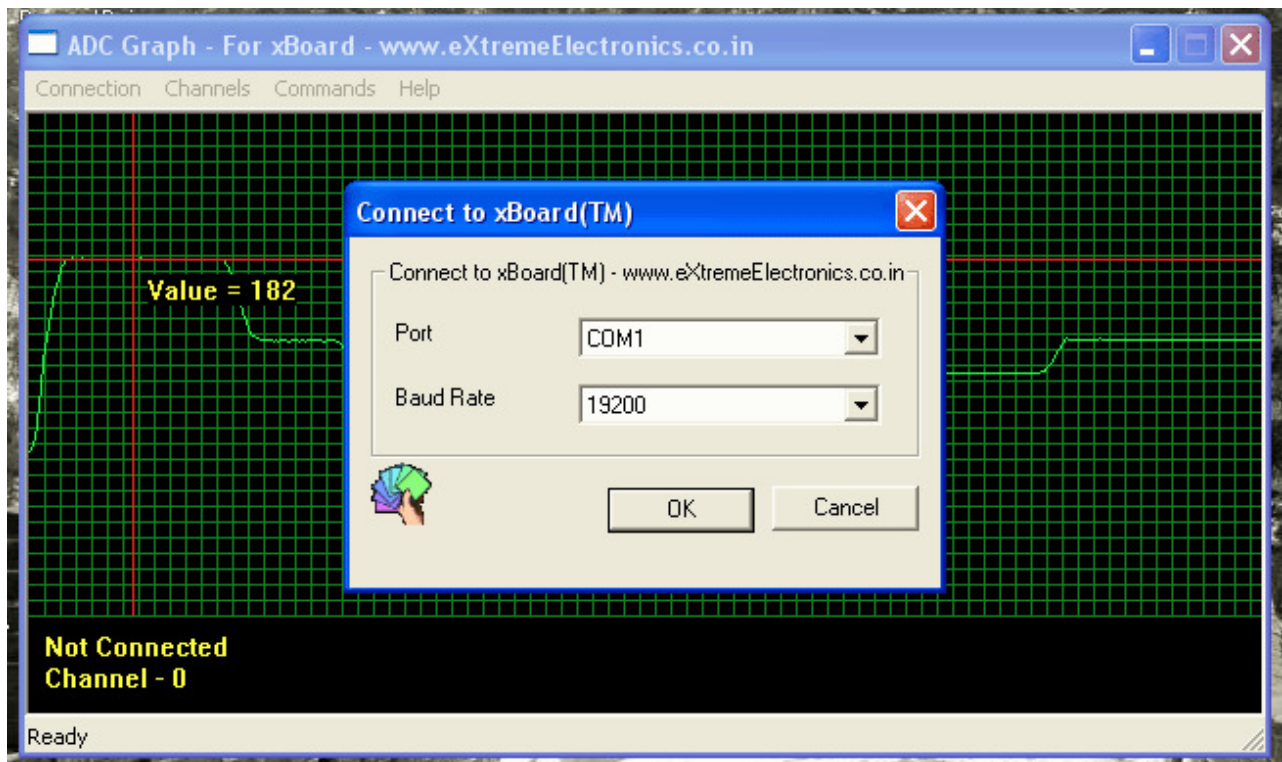


ADCGraph2 for Windows

Using the software is simple. First make sure that xBoard v2.0 is flashed with the ADCGraph firmware (details on building the hardware is given below) and connected to [USB to UART Bridge](#) which is connected to your PC (Make sure drivers are installed, link to driver are given on [this page](#)).

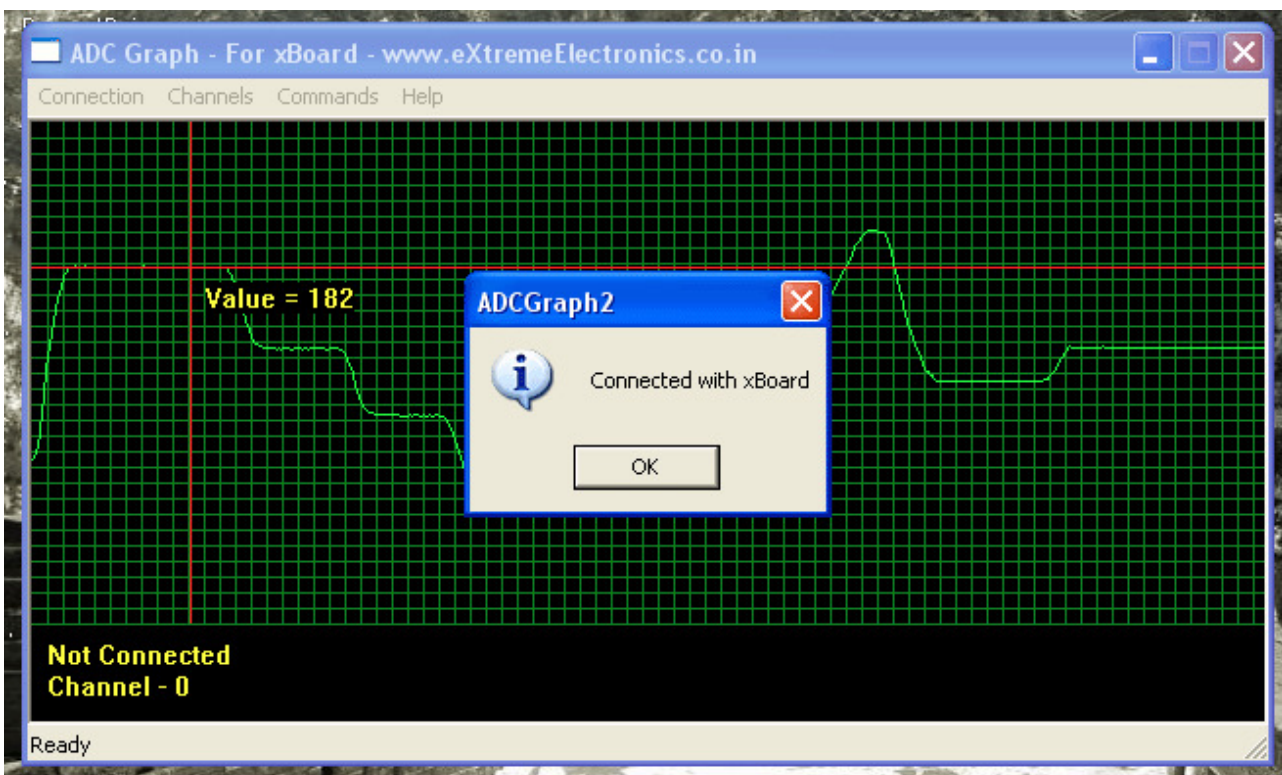
Click "**Connect**" from the "**Connection Menu**" and select the COM Port where [USB to UART Bridge](#) is present. To find out which COM Port number is assigned to the USB to UART Bridge see [this article](#) (scroll to the end).

Select Baud Rate as **19200**. And click OK.



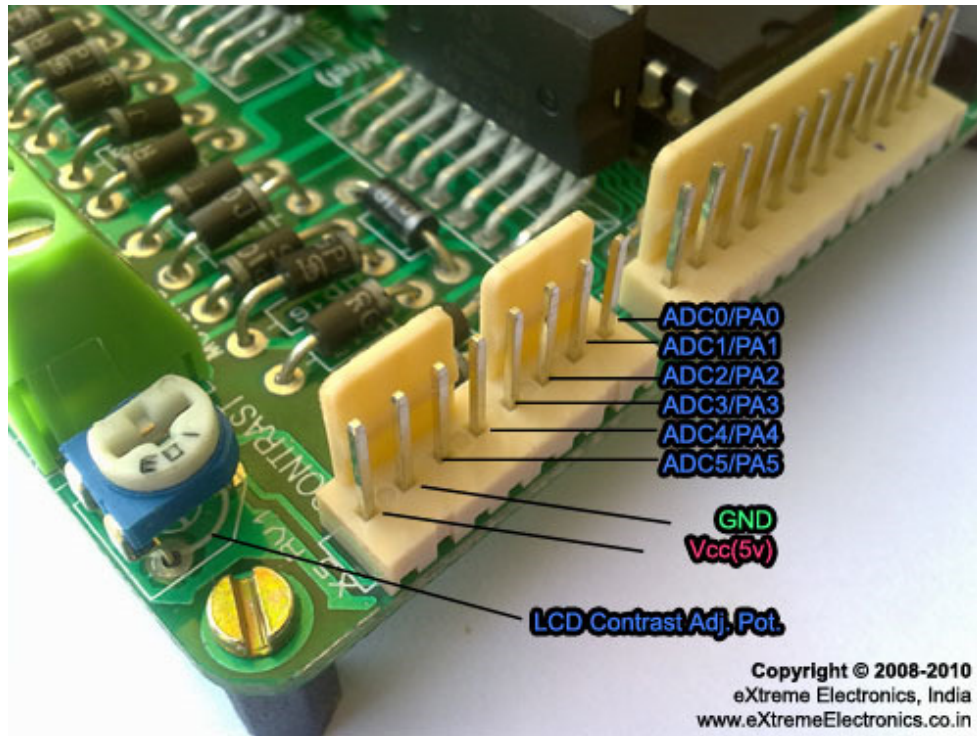
ADCGraph2 Connect Dialog

If everything was OK then a success message will be shown.



Connection Success!

Now you can start sampling the ADC by Selecting **"Start Sampling"** from the **"Command"** menu. On start-up channel 0 is selected, so it will show the analog value that is present on ADC0 pin. The pin out of ADC port on xBoard v2.0 is given below.



PORTA/ADC Port

You can change the input channel anytime by selecting it from "*Channels*" menu.

For testing purpose you may apply signals to the ADCx pins using a variable resistor (POT) as a potential divider as described below.

Connect the LEFT PIN to **GND**, RIGHT PIN to **5v** and the CENTER PIN to **INPUT of ADC** with a 10K resistor in series. Now you can rotate the POT to vary the Input to ADC from 0v to 5v.

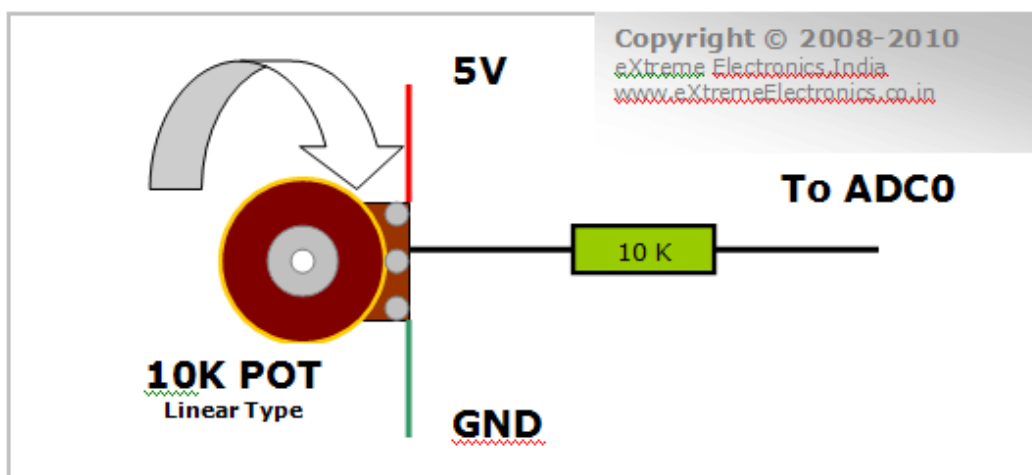
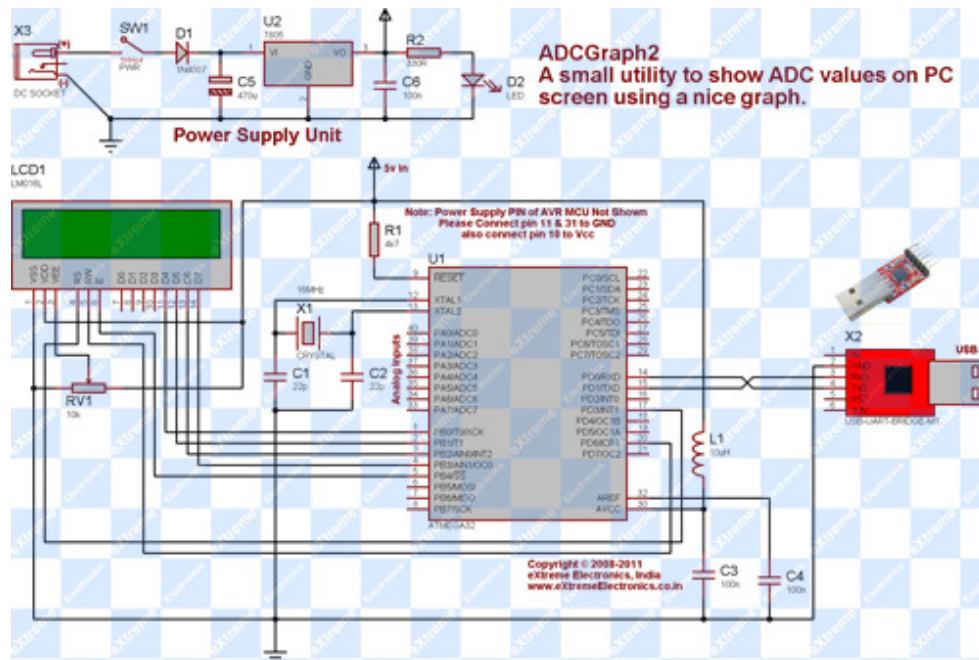


Fig: POT Connected to ADC of AVR

Hardware for ADCGraph2

The complete schematic for the ADCGraph2 hardware is given below.

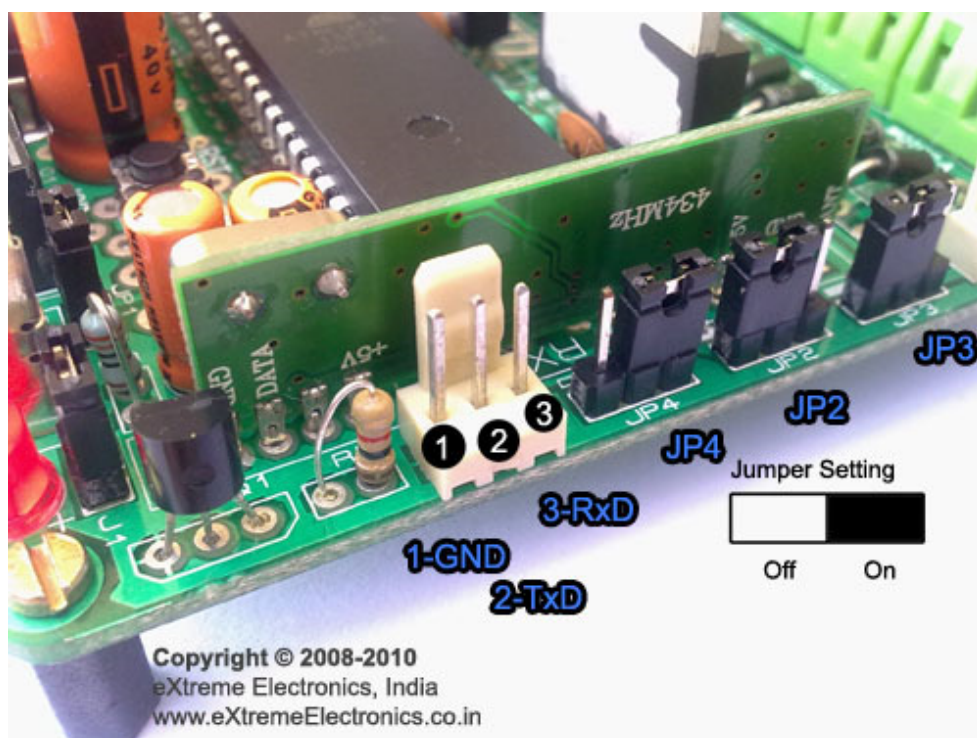


AVR ADCGraph2 Schematic

After turning the system on, adjust RV1 until the display is clearly visible.

If you are using xBoard v2.0 then you do not need to solder anything. Just hook up the LCD module at its connector. And connect the USB to TTL Module using 3 burg wires. Please refer to the following article on how to connect LCD module to xBoard and locate the UART pins.

- [Using LCD Module with xBoard v2.0](#)



USART, JP4, JP2 and JP3

Make sure jumper **JP4** is in **Off** position. (The image above shows **JP4** in **On** position so make sure its reverse of that!)

Help Us!

We try to publish beginner friendly tutorials for latest subjects in embedded system as fast as we can. If you like these tutorials and they have helped you solve problems, please help us in return. **You can donate any amount as you like securely using a Credit or Debit Card or Paypal.**



We would be very thankful for your kind help.

Downloads

- ADCGraph2 Package (including PC Software, Firmware and this documentation)

You can then burn the firmware hex file to your target ATmega32 MCU using any [AVR Programmer](#). **You also need to set the two fuse bytes in the MCU to the following value.**

- High Fuse Byte = 0xC9
- Low Fuse Byte = 0xFF

Important Notice: The project will NOT work at all if the Fuse bytes are NOT programmed.

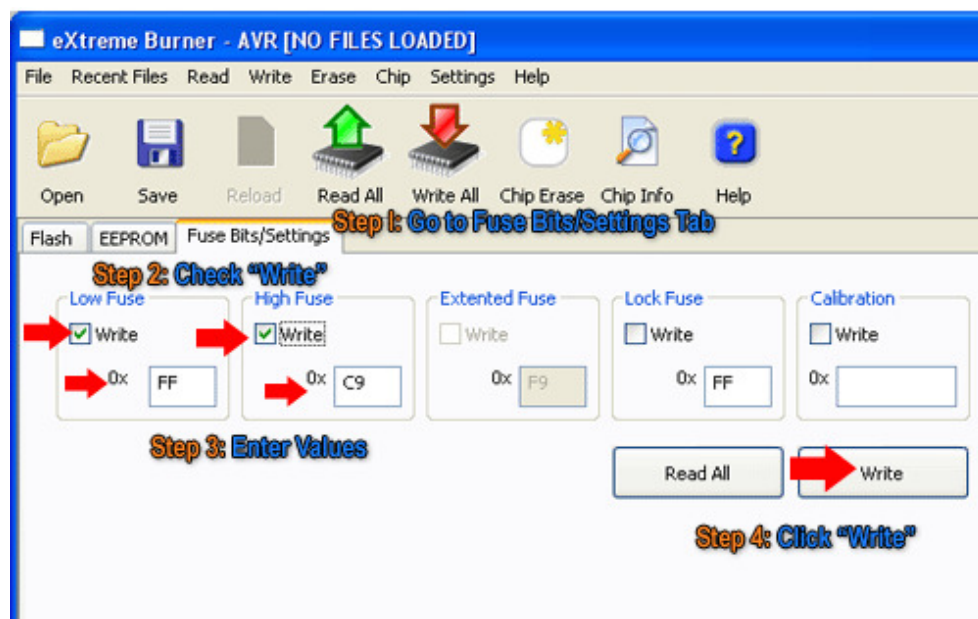


Fig.: Setting the Fuse bytes.

By
Avinash Gupta
www.AvinashGupta.com
me@avinashgupta.com