Project Report

Digital Oscilloscope

Chaitali Joshi (09026009)
Deepshikha Verma (09026017)
Surabhi Sachdev (09026023)

Aim

Our aim was to build a digital oscilloscope using Arduino board and displaying the waveform in our computer.

Hardware and Software Used

Arduino microcontroller board, Computer, Processing software, Firmata library, breadboard, Opamps, Potentiometer, signal generator.

How we proceeded

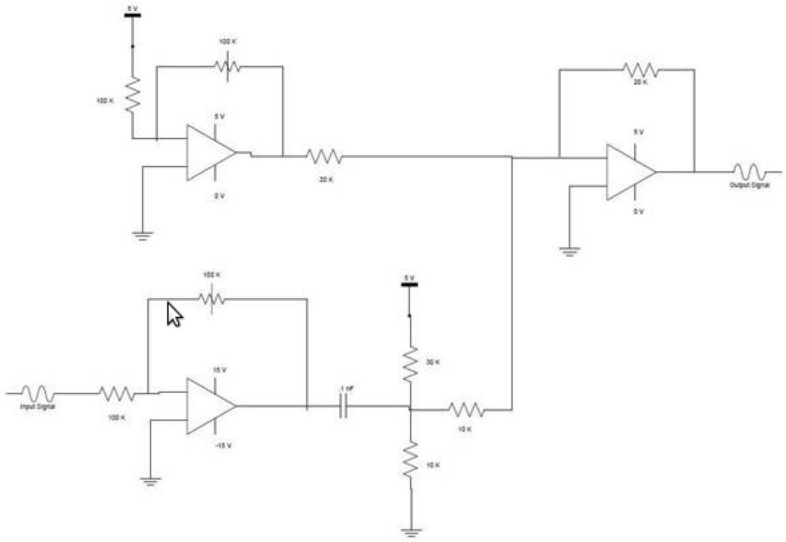
- At first we needed to sample the signal, for which we used the Arduino ADC . Our bandwidth would be restricted by the maximum rate at which the ADC samples the signal. The ADC of the AtMega 168 takes 13 clock cycles (which is about 100μs)for one conversion on an average.

- Secondly , we needed to Serial Communicate the data to the computer. For this, we could have either used the Arduino RAM as a buffer, and sent the data to the PC in chunks at certain intervals. However, since the RAM of the AtMega is restricted to 1KBytes, we realized that the serial communication speed would constrain our sampling in that case. To circumvent this, we used the software Processing, using which we could directly read the data to the computer memory. This interface makes use of the firmata library.

- Apart from providing an easy interface for reading data, the processing software also came in handy to build the actual plotting GUI. The software is java based and we used a plotting library that we modified to suit our requirements of scaling, grids, measurements etc.

- Since the arduino ADC accepts AREF and GND values only between 5 and 0V, and our signals may lie beyond this range. So we built a level and shift circuit that uses two potentiometers and three Op-Amps that can be used to appropriately scale the signal, and add a DC level to prevent negative values.

Circuit Diagram



Results and Possible Improvements

We were sucessful in getting decent waveforms for low frequencies, that pretty much responded to change in amplitudes and frequencies. Since ou sampling rate is restricted by high level functions used by the arduino and processing, we also tried to implement a C level code. However, the USART code interfered with some default settings in the arduino ad we did not have enough time to sort that problem out. The sampling rate can be greatly improved by using an external ADC to sample faster.

References

[1] Useful links:
http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2010/egm23_lad97_rsw35/egm23_lad97_rsw35_website/egm23_rsw35_lad97_final_project_v2.htm#_Toc260906780.
http://www.arduino.cc/en/Reference/HomePage http://www.arduino.cc/playground/Interfacing/Processing http://processing.org/reference/

Acknowledgements

We thank our senior Varun Perumal for exceptionally useful inputs throughout the project.
Updated 12 November 2011.