

Fast Fourier Transform using Perl Interface to Tektronix THS Series Oscilloscopes

Advanced Measurements

EE6904 – Fall 2003

Columbia University

Prof. Robert Melville

Max Baker

Description

For this project I created an interface between a Tektronix THS720A handheld oscilloscope and a laptop computer using a standard RS232 serial port. The THS family of oscilloscopes have an ASCII-based command set that is specified in Tektronix part number 070-9751-01, [TekScope Programmer Manual](#). I created an interface library to the scope as an object-oriented Perl module – *Tektronix::THS*. Using this interface I then created an application that downloaded a waveform from the scope as binary data points and performed a Fast Fourier Transform (FFT) on it. The data is outputted in comma-separated value (CSV) format and future plans for the application include a built-in graphing mechanism. The outputted data is easily graphed in Excel, Matlab, or xplot. The Fast Fourier function chosen performs a Power Spectral Density calculation (V^2 / N^2) and uses a Hamming window and data overlapping for statistically better results.

Findings

The Perl module was created in about 2000 lines of custom code and the use of existing libraries for the FFT functions and the serial port interface. Although not needed for the FFT, the module implements the full command-set for the THS-series devices. The module can be used to easily implement a data-logging system, a remote-control interface to the oscilloscope, or to access the advanced features of the scopes easily. Another possibility would be to use the interface to create measurement macros. I imagine a measurement macro to be a series of measurements requiring different settings on the scope. Combining the module with a computer controlled signal generator or power supply could provide

automation to a test bench. Another possibility for the module would be simultaneous data logging of both waveforms and resistance or voltage data taken from the DMM.

The FFT.pl program has an easy to use command-line interface and outputs data in text format. For testing I used the calibration 1.2kHz square wave as well as a 10kHz and 20kHz sine wave. The use of the overlapping and windowing features in the Math::FFT library improved the accuracy of the resulting spectral lines. Figure 1 shows the spectral makeup of the 1.2kHz calibration square wave. The fundamental, and three harmonics are visible in the graph, and the characteristic envelope of a square wave is discernable.

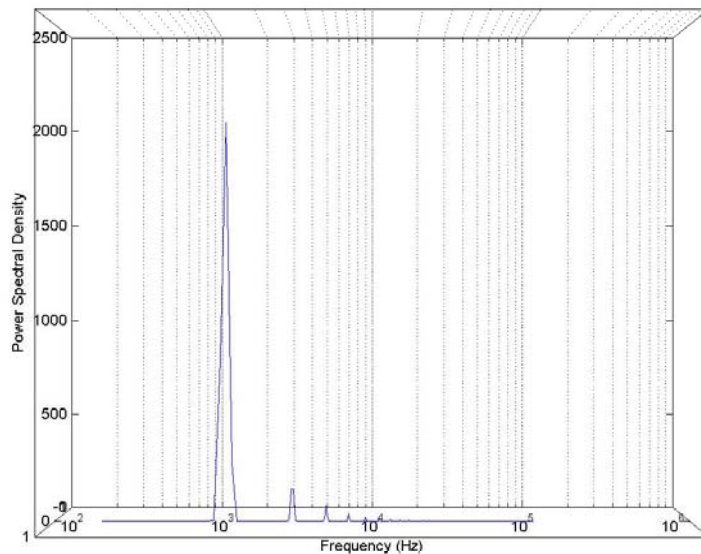


Figure 1 – PSD of 1.2kHz Calibration Square Wave

Figure 2 and Figure 3 show the response of FFT.pl to two input sign waves, one at 10kHz and one at 20kHz. Due to the eight-bit granularity of the waveform data little noise is seen in the resulting spectral data when plotted on a logarithmic scale. However noise is clearly visible on the waveform in the time-domain.

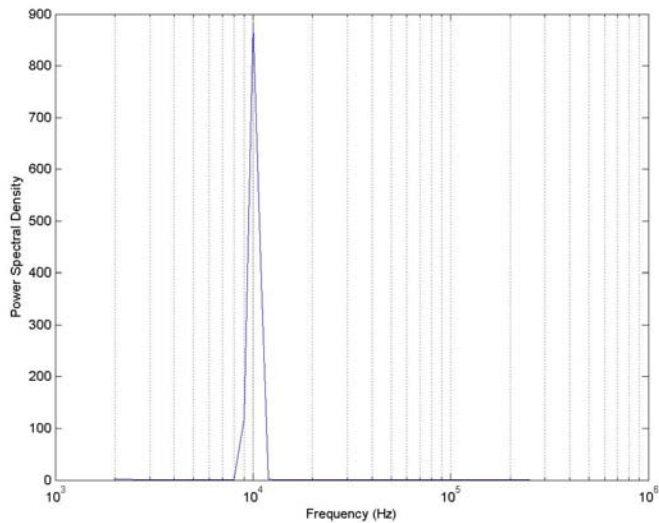


Figure 2 – 10kHz Sine Wave

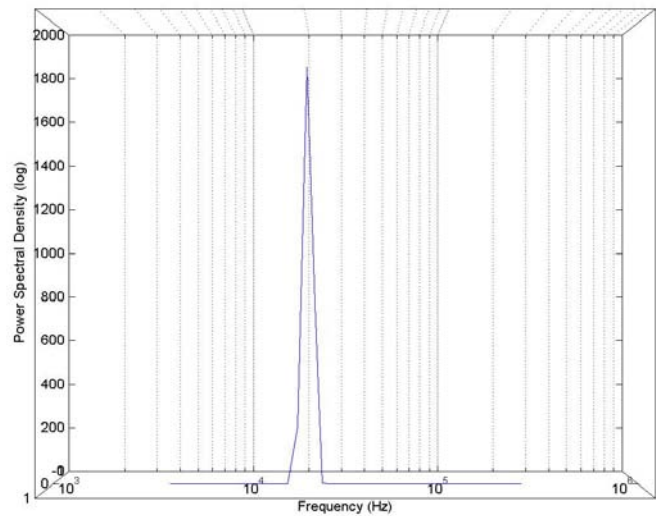


Figure 3 – 20kHz Sine Wave

Conclusion

The THS.pm Perl module and the FFT.pl application have only been tested in one environment for limited applications. Hopefully people will find new and creative ways to use them. One current limitation in the FFT function is the limited range of frequencies available for each time-base setting. For instance on a 1ms/division time-base the FFT is only capable of providing data from 0 to 103kHz. The application will gain a lot of granularity by allowing it to sample the data using multiple time-bases. This is directly analogous to the Resolution Bandwidth of a normal Spectrum Analyzer. Using the THS module the FFT application could change the time-base setting on the oscilloscope and resample the data, and then interpolate the results. Although similar features have already been added, the Perl module would also benefit from more helper methods to further shelter the end-user from the esoteric command language.

These particular Oscilloscopes have a following in the hobby market due to their portability, cost, and features. I hope both the FFT utility and the Perl API to this piece of test equipment will be a contribution to the community of users. This project is released Open Source and can be found at <http://warped.org/tek>.