Case Study: Fedora in an University Engineering Lab

Benjamin Kreuter
Robert Greene

May 14, 2008

Table of Contents

Purpose

This case study provides details on how Fedora 8 Linux was used and relied on in an academic laboratory project. The motivation for choosing Fedora is outlined, and the advantages afforded by this choice are discussed. In addition, the ability to use necessary proprietary software in Fedora 8 is discussed. Block diagrams of specific parts of the design are presented, to aid in illustrating how a Linux system was used to accomplish the goals of the project. Due to concerns over possible plagiarism by future students, the specific results of the project have been omitted, but are available from the authors upon request.

Introduction

The Binghamton University Electrical and Computer Engineering undergraduate program requires all students to take an introductory design course during the spring semester of their Junior year. This course is intended to build upon previous courses in electrical engineering, provide an introduction to real world engineering, and meet the ABET requirements for accreditation. The students are presented with a project that is specifically chosen to emphasize several key design concepts in analog and digital circuit design, and includes a software component.

The project is divided into three phases, which each involve creating digital samples of small signal analog inputs. The first two phases of the project have minimal software components, and use a microcontroller only for performing the analog to digital conversion. The third phase of the project relies heavily on software, and uses a microcontroller to perform analog to digital conversion as well as to communicate with a PC via a USB connection. The third phase also requires a program to be written on the computer to perform various levels of DSP and analysis of a sinusoidal input.

A standard set of components and programs are chosen to allow for an easier debugging and grading process, but with the teacher's permission students may use alternative components, with the understanding that they will receive no assistance. The standard setup uses a Motorola 68HC908 microcontroller, the Windows operating system on the PC, and Matlab. An alternative setup was chosen, using a PIC18F4520 microcontroller, Fedora 8 Linux on the PC, and GNU Octave. This alternative setup allowed for an easier to implement software design, which outperformed other designs on a variety of levels.

Project Overview

The third phase of the project required the students to create a system that could sample a 100 Hz sinusoid, with an amplitude of 100mVpp and an offset of 2.4V. The sine wave was to be processed in hardware to have an amplitude as close to 5Vpp as possible and an offset as close to 2.5Vpp as possible. The digitally sampled signal would then be transferred to a PC and processed using Matlab, showing the sine wave, a frequency domain magnitude, and the effect of zero padding.

This phase of the project was primarily a software design phase, as the necessary hardware had already been set up in previous phases. This software had to be able to collect up to 100 samples of the signal, at a rate of at least 1600 samples per second, and present the samples in Matlab. Matlab was also to be used to compute the discrete Fourier transform of the sample, and to produce a magnitude plot of that transform.

Solution Using Free Software

For the PC, Fedora Linux 8 was chosen as the operating system, using GNU Octave for processing the data, and a PIC18F4520 running the GPL licensed operating system PICos18. To manage the communications, a simple server was written, that continuously read from the USB port (using a /dev/ttyUSB device) and wrote the data to a named pipe. The PIC, running PICos18, continuously sent this data, whether or not the PC actually read from the port. A graphical user interface was written using Qt, which allowed the user to select the number of samples to take, the sampling rate, and the amount of zero padding.

This approach allowed a far more flexible and robust design than the approach suggested by the assignment. By choosing to use a Linux operating system, the task of reading from the USB port was trivial, reduced to reading bytes from a file. Furthermore, because of the support for named pipes in the same directory as programs and data, it was trivial to communicate the results with Octave. By creating such a modular system, it was possible to use Matlab as a drop in replacement, if that was required (although there would have been an unrelated difficulty, as described below), or to send the output from the server directly to GNUPlot, for debugging.

PICos18 was the chosen operating system for the microcontroller. As a GPL licensed system, the source code was available in its entirety, which played a pivotal role in the project. During the development of the project, a major bug was found in the RS-232 driver, which caused the system to crash if too many serial requests occurred in a short amount of time. This problem was quickly solved by examining the source code of the driver, and removing an unnecessary section of code that was causing the problem.

GNU Octave is available for all operating systems, and is compatible in most areas with Matlab. The advantages of using a Linux system, and writing a special server to read the data, were significant, but the school does not provide Matlab licenses to students (because of the cost). Matlab is installed on school computers, but those computers were running a Windows system, which would not have provided the necessary features for the modular design described above. Matlab code was a required deliverable of the project, but the code from Octave ran in Matlab without difficulty.

In designing this project, cost was a concern. The funds available to a typical engineering student are limited, and had to be prioritized for the hardware required for the project. Although the university provides some free licenses for software, the software needed to complete the project as designed could not be obtained entirely free of cost. The Fedora Linux system can be downloaded for free, as can all of the software in the Fedora repositories, including GNU Octave and GNUPlot. PICos18 is also available free of charge, which compounded the benefits of GPL licensing.

Conclusion

By choosing a free software system, a superior software design was achieved, at no cost, in a university engineering laboratory setting. This design was more flexible than designs based on proprietary software, with a shorter development schedule. This project may serve as an example of how engineering students can use Fedora Linux in their education, especially in a demanding course in engineering design.