Session Number 2632

A New Course for Electrical and Computer Engineering Majors: Engineering of Real Time Systems

Massood Z. Atashbar, Hakeem Ogunleye

Department of Electrical and Computer Engineering
Electrical and Computer Engineering Department
Western Michigan University
Kalamazoo, MI 49004

Introduction

University Computer Engineering programs continue to be a popular draw for students. Still, since they are relatively new, their defining curricula continue to evolve. Traditional courses such as digital logic, and digital design, microcontrollers, computer interfacing and computer architecture are mainstays, but there continues to be many holes to fill. Part of the problem is that Computer Engineering (CE) is still considered to be an interface between Electrical Engineering (EE) and Computer Science (CS). Electrical Engineering, where it is usually housed, embraces the notion that computer hardware is fundamental to the discipline while computer science views computer software as the defining entity. The truth is that both are correct and computer engineering students need to understand both disciplines equally well. At the same time, this understanding needs to go beyond simply knowing about EE and CS. Students must be able to apply the principles of high level system analysis and design techniques to electrical engineering applications.

It authors' belief that one of the most common areas between EE and CS is that of digital data acquisition, signal processing, communication and control. Coincidentally, this turns out to one of industry's major needs as well. At the same time, students to be exposed to a reasonable amount of high-level software engineering that is engineering based. However, there is no way that an undergraduate CE program can require each of these courses in an already crowded curriculum. The solution to this problem that has been implemented at Western Michigan University (WMU) is to create a junior level course that teaches high-level software engineering using Visual Basic that is applied to data acquisition, signal processing and network communication. This experiment has, in the opinion of the authors, been highly successful in that students not only learn a great deal of information but also gain experience in applications that are will be useful in further course work and senior projects as well as their future careers.

The Problem

Computer Engineering students traditionally take a subset of courses required for the EE and CS degrees, along with specialized CE courses in architecture, microcontrollers and digital design and integrated circuit design. From an EE view, they have too little hardware experience to useful interfacing and according to CS, programming embedded microcontrollers in assembly or C is insufficient. This would probably be satisfactory if most CE graduates went to work in high technology companies like microelectronic industry, but this simply not the case. A number of CE students wind up being employed as more traditional engineering doing software design. Many will have careers in digital communications and data acquisition, where signal conditioning, digital signal processing and control are very real applications.

While C might be a more than satisfactory language for low-level programming of embedded systems and device drivers, it is used much more sparingly in higher level systems programming. One of the most popular languages is Visual Basic, which along with the rest of the Visual Studio™ produced by Microsoft provides an excellent platform from which to integrate engineering systems. Being an object-oriented language, Visual Basic provides a full-fledged environment that is interrupt based. Since it is supported by Microsoft, there are a number of powerful activeX components available from third parties.

Students need to understand how to conduct high-level system analysis and design. However, being engineers, they need to understand this from the perspective of electrical engineers in a digital application. With the number of computers (likely PCs and PLCs) present on a factory floor and the embedded controllers assigned to appliances, toys, automobiles and every other kind of product, CE graduates are ideally suited for traditional electrical engineering applications. Often such projects quickly reduce to software engineering and interfacing. The processor acquires and processes the data derived from sensors and passes commands along to amplifiers and actuators for control. Or, simple data communications using any number of fieldbus and autobus sources, form a local network. In most cases, this requires data acquisition followed by signal processing and feature extraction, and finally data deposition. These are, first and foremost, computer engineering skills.

The current state of affairs is that most CE curricula ignore data communications and treat signal processing only theoretically and usually at the end of the academic career. This is ironic since a quick review of electrical and computer engineering jobs show significant opportunities in high-level computer systems design, data communications, signal acquisition and control. This is not simply a recent phenomenon; it has been true for years.

The Solution

At WMU a new course entitled *Engineering of Real Time Systems* has been introduced in which we conduct high-level software engineering projects as applied to data acquisition, communications and control systems. This course is a standard three-credit offering with no official laboratory component. However, as is often the case in engineering courses, there is a considerable amount of outside work writing programs and working with instrumentation by the students in an open laboratory environment. In general we found that not only did the students do very well, but they genuinely liked the course, the material and the way in which it was conducted. This is validated by a number of measures.

Prerequisites for the *Engineering of Real Time Systems* course include one year of Object Oriented C++ and one semester of Assembly language. Therefore, students were basically familiar with computer organization and the fundamental of algorithmic programming. Building

this foundation, the first five weeks of the course was devoted primarily to learning the Visual Basic programming environment. From this foundation, the engineering principles of building virtual instrumentation and interfaces were developed from a software engineering point of view. Once this tool was sufficiently understood, we introduced standard engineering components and had the students create virtual oscilloscopes, spectrum analyzers and asynchronous communication modules. These were all done using Visual Basic, data acquisition cards, serial interface ports, and elementary circuitry.

Table 1: Course outline of ECE 351, Engineering of Real Time Systems. * Representing programming assignments.

Visual Basic:	basic components and procedures* (3)
	visual programming and error trapping* (3)
	discrete computation and descriptive statistics* (3)
	virtual oscilloscope and real time data acquisition* (3)
	spectral analyzer and FFT* (3)
Data acquisition	signal sampling and Nyquist critera (3)
	signal conditioning* (3)
	sensors (3)
	third party activeX components* (2)
Communications:	Asynchronous communications protocols and the serial port*(4)
	GPIB communications and the parallel port (3)
	Data Communication protocols
Other topics:	Feature extraction (3)
	Software engineering (2)
Tests:	Tests (2)

Table 1 shows the topics of this course, along with the number of classroom hours devoted to each. During the first five weeks, we not only taught the Visual Basic, but also conducted 15 minute consultation sessions with each student prior to the due date for each assignment. This enabled us to not only motivate a stronger work ethic, but also tutor on the fundamentals of good programming practices. The remainder of the course was spent on data acquisition (sampling, Nyquist criteria, Fourier Transforms and feature extraction), signal conditioning, sensors, serial and parallel asynchronous communications and control. Each topic was demonstrated at the hardware level and formed the basis of a software assignment. Assignments are listed along with a short description in Table 2.

Table 2: Programming assignments used in ECE 351.

1.	Demonstration Program: A Visual Basic version of the "Hello World" program in which a variety of VB components are demonstrated.
2.	Ohm's Law: A program emphasizing error trapping and conditional results.
3.	Descriptive Statistics: Reading/writing files, creating statistical tables, histograms and traditional statistics.
4.	Oscilloscope: Sampling a signal using software interrupts and displaying the results graphically as a virtual oscilloscope in real time.
5.	Spectral Analyzer: Incorporating an FFT module with the Oscilloscope program to create Bode plots in real time.
6.	Data Acquisition: Use of a third part (National Instruments) activeX oscilloscope component along with a DAQ card for data acquisition.
7.	Serial Communication: Creation of a asynchronous serial communication protocol to acquire date from the serial port.
8.	Feature Extraction: Finding and graphing the period as a function of time from the signal acquired from Serial Communication.

Results

This course was taught on a trial basis by the author to a class of twelve junior Computer Engineering students. On the whole, the professor felt that students were most receptive and that the course was successful. Student feedback, in the form of standard student review forms

confirmed this. Further, after the first offering seven of the twelve students have requested related senior design projects with the professor as advisors. This is in contrast to the majority of previous computer engineering senior projects where students simply implemented another previous computer engineering senior projects where students implemented another microcontroller in an industrial setting. Because of this apparent success, the Engineering of Real Time Systems course is now a required in computer engineering curriculum and will be taught in the same format. This will meant that the enrollment will grow to approximately thirty-five students taking this course each fall semester. This will require approximately ten software licenses for Visual Basic, ten workstations along with data acquisition cards, motion controllers and miscellaneous interfacing hardware. While WMU has much of this equipment already available.

An Example Implementation

The project presented below utilizes data acquisition, signal processing and network communication to demonstrate the concept being proposed in the report. The application consists of two parts: the sender and the receiver. On the sender side, the program uses National Instrument's data acquisition card to acquire signals from a signal generator and it displays the signal on the senders' side. If the send button is clicked, it parses and packages the received signals (processing the signal) and then opens a communication port (COMM PORT 1) of the computer and sends the signal to the receiver continuously. It only stops when the send button is deactivated. The graphical view of the sender program is shown Figure 1.

On the receiver end, once activated, the program continuously reads its communication port (COMM PORT 1) every 10 milliseconds for the senders' package. Upon receipt of a message, it unpacks the message and then displays the data on a graph. The graphical view of the receivers program is shown in Figure 2.

The communication link between the sender and the receiver is a RS 232 cable. The RS 232 protocol is implemented by Visual Basic thereby hiding the complexity from the programmer and that is one of the reasons why Visual Basic is suitable software to use in the class.

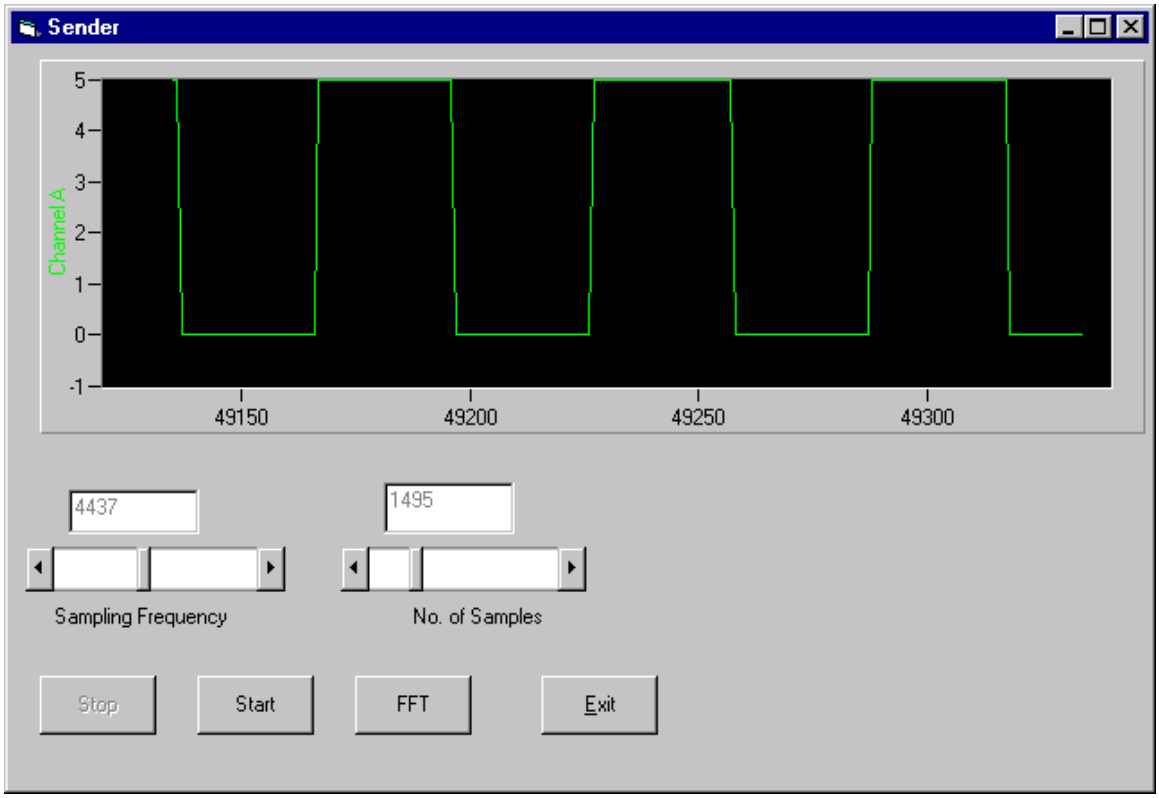

Figure 1: The sampled signal on sender to be sent to the receiver.

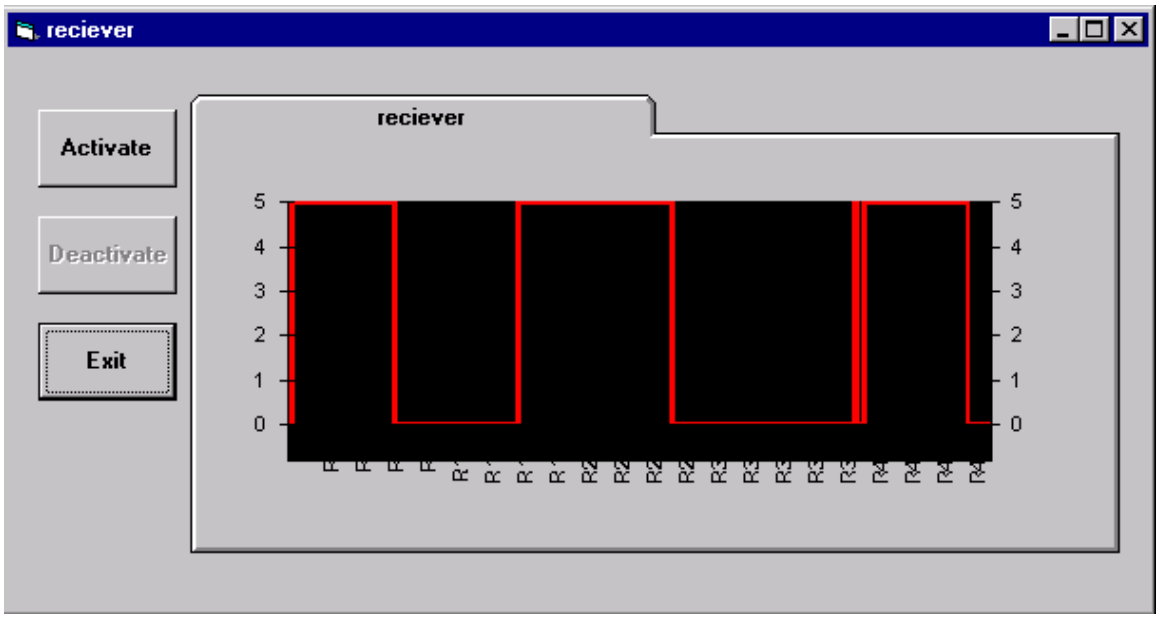

Figure 2: The received signal from sender on receiver

Biographies

MASSOOD ATASHBAR is an Assistant Professor in the Electrical and Computer Engineering Department. His research interests include mobile robots, real time systems, sensors, and VLSI. He may be contacted at: massood.atashbar@wmich.edu.