Engineering EDUCATOR

The latest news about teaching tools, measurement technologies and innovations in engineering education

VOLUME 3, NUMBER 3
Fall 1999

INSIGHTS

“Go High in Turns”

Marsh Faber, Editor

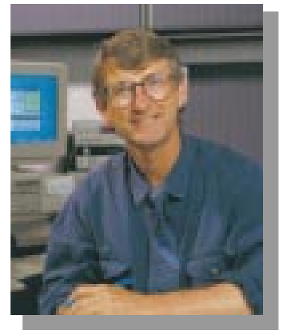
A gaggle of us were migrating to an ASEE event when I noticed that the sign above the bus driver’s head said, “Go High in Turns.” I assumed it meant to swing the bus wide when turning, to avoid creasing any oncoming vehicle; however, as an engineer I was taught not to assume, so I asked the driver.

His reply, “Sure, it means swing wide, but we’re also taught that it means to look above the horizon during the turn. When we turn a corner, our natural tendency is to look down at the road, but with this big bus we also need to know what’s way ahead, so we ‘Go High.’”

I got the same lesson from a birdwatcher friend of mine. Whenever I rode with him, he was always first to spot a hawk. It didn’t take me long to realize that he was also “going high.” I tried it myself—concentrating on the road, but also forcing my eyes to jump to the tops of the fence posts and trees.

Now I spot more birds and reduce my stress level in traffic at the same time. I notice things like circuit breakers on the power lines, and special fixtures that prevent birds from nesting there. I have also saved minutes of commuting time simply by taking in a larger picture, and making a “prediction game” out of anticipating what the traffic will do.

This is a pivotal time in the history of Hewlett-Packard: we are splitting into two separate companies. The new company stems from HP’s Test & Measurement roots. This newsletter and the HP Engineering Education websites, **www.EducatorsCorner.com** and **www.FutureEngineers.com**, are part of the new company.



After 33 years with HP, I am brimming with enthusiasm about the change. I strongly believe it will make us more focused and quicker to respond to customer needs. It’s a terrific opportunity to take a step back and see where we want to go—to “Go High in Turns.”

If any of you have strong opinions about how our new company should interface with engineering education, please drop me a line at marsh_faber@hp.com (this e-mail address will still work for a little while).

Marsh Faber
Education Program Manager
Electronic Measurements Division

IN THIS ISSUE

HEWLETT® PACKARD

Share Your Thoughts

Marsh Faber, Editor

This newsletter, as well as the HP Educator's Corner Website (**www.EducatorsCorner.com**), is intended to help you, the educators responsible for shaping students into competent engineers. But to do a good job of meeting your needs, we ask for your feedback. Please e-mail your comments, questions and concerns to marsh_faber@hp.com.

Make it Interactive!

Congratulations for a cool website on Resources for Future Engineers.

Possibly, the website could be enhanced with a feedback channel and an open interactive forum linkage labeled plainly e.g. "Suggestion," "Idea," "Question and Answer," "Real Time Discussion" from every webpage in addition to the focused webpage "On My Mind."

The benefit is for immediate interaction to satisfy requests from future engineers who visit Resources for Future Engineers. Data could be solicited and collected voluntarily for success figures, and for areas of website improvement. These figures could be published in the website in real-time, and analysis posted in real-time with the decisions and feedback.

Best Regards,

I.T. Cheah
Malaysia

I.T.,
*You're right. Our intent is to add more real-time feedback questions and discussion threads for the student site at **www.FutureEngineers.com**. Keep watching as we grow the site.*

Formula Error

I only would like to make a comment about an error in formula 20 of the coupled oscillator experiment by S. A. Dodds of Rice University. As dimensional analysis readily learns, there should be an extra ^2 in the term $(w_a{}^2 - w_b{}^2)$, so it should be $(w_a{}^2 - w_b{}^2)^2$ (the dimension of this term should be the 4th power of **w**).

I hope you can quickly change formula 20.

Aart van der Pol
Groningen University
The Netherlands

Aart,

It's fixed. It was our omission and not the fault of Professor Dodds. [By the way, we sent Dr. van der Pol a nice gift for finding this bug on our site.]

Preparing Students for RF Careers

I'd like to comment on Jerry Murphy's Editorial about RF engineering. I see the problem from outside, being an Environmental Engineering teacher... but with 20+ years' experience as an amateur radio operator.

Most of the youngsters do not see "magic" in radio anymore. TV and broadcast radios are dependable and you are only a remote away from a wealth of information. Usually, there are no knobs to turn, adjustments to make; it's just switch-and-go. When they come to school they have no clear idea about the future, but they see more opportunities on micro-processor/automation even (arghhh!) software development than in radio. I guess I'm biased, but I see the education they are getting is no good. And not only in my school.

There is, though, someone who is taking advantage of radio to teach students many good things: theory, parts identification, soldering skills, circuit analysis, tuning, etc. He is Professor Rutledge, from CalTech. All his students in a 200-level course have to build a low-power radio transceiver, the Norcal 40A. *(Editor's note: you can get details about the kit they use and a newly published book by Professor Rutledge via **www.EducatorsCorner.com/links**.)*

I guess I've rambled too long. I'll go back to normal work. Be well.

Jon Iza
University of the Basque Country
Bilbao, Spain

Jon,

I looked at the websites…very impressive, and a great hands-on lab for students.

Editor's Note

Note: For those (many) of you who noted with disdain that our "educatorlinks" site didn't contain the promised information from the last newsletter, we apologize. This time, the "links" site will be perfect…well, as perfect as we can get.

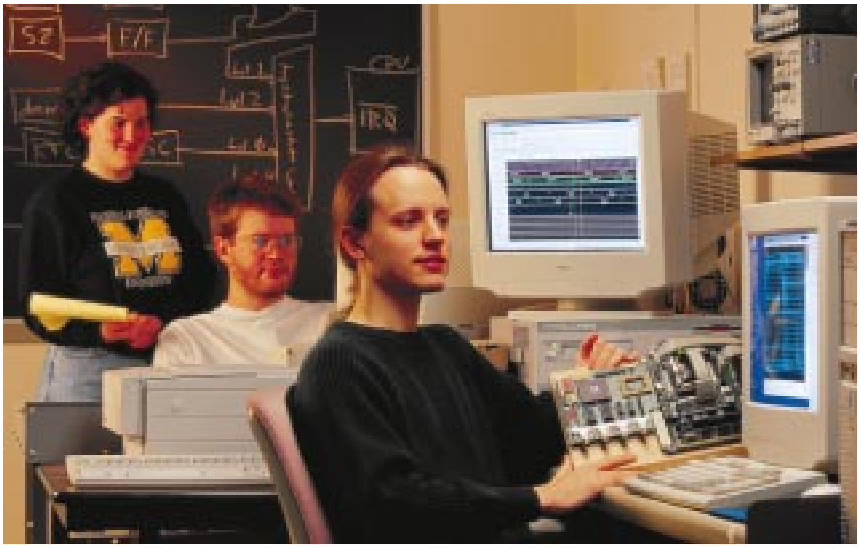
Students Discover Magic of Engineering via 32-bit Microprocessors

Undergraduate students in Professor Steven Reinhardt's "Design of Microprocessor-Based Systems" class at the University of Michigan work long hours in the lab debugging and tweaking their logic circuits. But they're not just learning the nuts and bolts of engineering. They're discovering something much more valuable—the thrill associated with solving complex problems just like the ones they'll face when they graduate and move on to real-world jobs.

"The hardest aspect of the class was getting the concept of how all the pieces fit together," said computer engineering student Bill Glennan. "But as soon as you get down to the lab, that all just seems to fade away. You start trying new things and you start understanding. When you finally get it to work, wow! The magic just happens."

What is it about Professor Reinhardt's lab that ignites that spark in undergraduate students? It's the challenge of doing integrated hardware/software development on modern 32-bit PowerPC microprocessors and FPGAs. It's the excitement of getting to use state-of-the-art tools and equipment. And it's the knowledge that this lab is an important step toward a future career.

"You walk into the lab and suddenly you have to apply what you've been taught," explained Glennan. "The concepts are not just theory anymore. So many times I'll sit in a class wondering, 'When am I ever going to use this stuff?'



University of Michigan students in the microprocessor lab. For insight into their learning process, see the lab exercise on page 5.

"Suddenly it doesn't feel like a class anymore—it feels like the real world—which is exactly what a class like this is designed to prepare you for."

But then you get into a lab like Prof. Reinhardt's, and you realize that you really are going to use this stuff, and you are going to use it NOW! Suddenly it doesn't feel like a class anymore—it feels like the real world—which is exactly what a class like this is designed to prepare you for."

Using 32-bit Motorola microprocessors with PowerPC architecture lets Reinhardt's students gain experience that's applicable to both embedded systems and general-purpose computing applications.

"This is the same type of architecture that's used in a Macintosh computer, for example, or a Kodak digital camera," explained Reinhardt. "They can make the connection that it's not just a microcontroller course, it's really about microprocessor-based design, both for general purpose and embedded systems. Although embedded systems get much less publicity, they vastly outnumber general-purpose systems in the market."

Students Find Jobs in Local Industries

Reinhardt's students quickly figure out that the experience they gain in this class can help them find jobs in a wide variety of industries. Conversely, it didn't take long for companies in the university's neighborhood to realize

Continued on page 4

Students Discover Magic of Engineering via 32-bit Microprocessors

Continued from page 3

that students coming out of Reinhardt's class were skilled at using the same tools and equipment they use in producing their products.

"I think my experience in this class helped me land my upcoming internship at Ford Motor Company," said electrical engineering major Dave Wentzloff. "I'll be working with a Motorola processor at Ford. I'll be building something that's going to simulate a vehicle. This processor will hook up to my simulator instead of a vehicle, so they can test a spark plug failure, for example, just by sending the simulator some code. I'll be working with a bunch of mechanical engineers who need an electrical engineer to design this simulator—and that's me!" exclaimed Wentzloff, clearly excited that he was chosen for the task.

"I believe that students taking this class are really going to benefit," said Wentzloff. "As a teaching assistant in this class last semester, I'd have to remind students who were complaining about the workload that this is a very useful class. We're using the sort of debugging tool that engineers actually use in the field."

Right Tools Make a Difference

"Most schools use older 8- or 16-bit processors and that makes things easier in a number of respects," said Reinhardt. "One is the amount of wiring you have to do. The FPGAs really help though, because we don't have to do any physical wiring—it's all sort of programmed into these chips. The other big issue is the more complex your signals are, the more complex task it is to observe them. With an 8- or 16-bit chip, one of the simpler logic analyzers is sufficient. But with a 32-bit chip, you need a more powerful logic analyzer solution."

"I'll be working with a bunch of mechanical engineers who need an electrical engineer to design this simulator—and that's me!"

Reinhardt and his colleagues chose HP 16600A Logic Analysis Systems for the microprocessor lab. "The 16000-series analyzers have a nice windows interface so it's fairly easy for students to navigate around. The really complex part of using a logic analyzer is configuring it for the particular system and setting up the trigger conditions. We've preconfigured the analyzers with basic trigger conditions so the students can just use the trigger conditions we've designed. The nice thing about it is that it gives them the complete picture of everything that's going on in the system. They see the address bus and the data bus and it shows them the entire contents in hexadecimal, for example, if that's the format they want. They see everything in conjunction with all the control signals and then they can see their own internal logic signals, as well."

Future Directions

Much like his students at work on their logic circuits, Professor Reinhardt has spent the last year debugging and tweaking his microprocessor-based systems program. "We're still in the process of figuring out all the details and enhancing how we're using the equipment. In addition, we're looking at doing something more significant in terms of a project. Right now, our students perform a series of lab exercises, basically. We're planning to move to a model where there are a few less exercises and at the end of the term, the students do a small design project."

You'll find one of Professor Reinhardt's lab exercises on the following pages. The rest are available on the web. You can reach them via the newsletter links page at **www.EducatorsCorner.com/links.**

Professor Reinhardt can be reached at stever@eecs.umich.edu.

Analog-to-Digital Conversion

Professor Steven Reinhardt
University of Michigan

Introduction:

Analog-to-digital converters (ADCs) allow microprocessor-based systems to sense and act on analog signals from sources such as microphones, temperature sensors, etc. In this lab, you will interface an ADC to the MPC823 and study its operation.

Goals:

1. To learn the operation of an analog-to-digital converter.

2. To observe the analog-to-digital conversion process and understand its parameters and limitations.

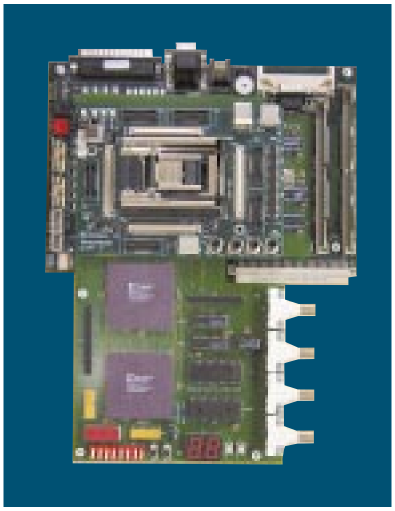
3. To increase your experience with interfacing digital I/O devices to a microprocessor bus.

Specification:

Your system will digitize an analog signal at a rate of 500 Hz and display the digital codes on the seven-segment display. In addition, pressing push-button S1 will buffer 500 consecutive samples (one second worth) in a memory array.

The hardware simply interfaces the National Semiconductor ADC0808 converter found on the *lab expansion board* with the MPC823 via a single 8-bit read/write register. A write to this register initiates a conversion. The data value written selects one of the eight possible analog inputs. A read from the register returns the ADC output, i.e. the digital code corresponding to the input voltage level. The conversion requires a variable amount of time, so use the ADC's end-of-conversion signal to generate an interrupt.

All italicized and underlined items are available on the web. You can reach them via links on ***www.EducatorsCorner.com/links.***



If you are interested in using this lab in your school, you are welcome to reproduce the lab expansion board created by the University of Michigan. Schematic drawings for the board are available on the web, as is information about the company that manufactures them for the University of Michigan. Go to ***www.EducatorsCorner.com/links*** *for fast access to these sites.*

The software you write will initiate conversions at the 500 Hz rate, collect the digitized samples as they become available, and update the seven-segment display and the memory array. To maintain a precise sampling rate, you must initiate conversions at evenly spaced intervals, regardless of when the conversions complete.

Details:

Download and print the *ADC0808 data sheet*. Study it carefully, especially Fig. 5 (p. 7), the conversion timing diagram and Fig. 8 (p. 8). From the timing parameters (p. 4), note that the "max" value for a "minimum" parameter is a worst-case minimum, e.g., the minimum start pulse width is typically 100 ns but may be as much as 200 ns for some devices. In other words, your circuit should provide a start pulse of at least 200 ns.

The "typical application" diagram on p. 12 helps show how the interface register should work. Note this circuit only meets the minimum pulse width and setup and hold specifications if the bus is very slow (a few MHz at most). Your interface will be more complicated since you will need to extend the control signals to meet the timing specifications.

The ADC also requires a clock (to sequence the internal successive-approximation controller). Use a counter to divide the MPC823's bus clock down to a frequency within the ADC's specified range; in this lab, slow the MPC823's bus clock down to 10 MHz.

The Xilinx connections to the ADC are accessible through *the LOCAL_DEVICES macro in the EECS373 library*. The ADC data pins are wired on a common tristate 8-bit bus with the DACs and the SRAM; see the "*Basic I/O*" *schematic* for details. This bus is represented by the D_OUT and D_IN connections and the tristate buffers are controlled by D_OUT_EN. If there is any confusion about the macro, use the hierarchy tool to look inside and see how it's connected to the Xilinx I/O pins. You

Continued on page 6

Analog-to-Digital Conversion

Continued from page 5

should be able to match the pin numbers (the LOC= attributes) in the macro with the pin labels on the schematic.

How you handle S1 is up to you. The important thing is that the 500 entries of the array represent 500 consecutive samples from the digitized signal, regardless of the duration of the S1 button press.

Procedure (in the lab):

Prelab questions are available on the web at ***www.EducatorsCorner.com/links.***

1. Add lines from the ADC's START, ALE, EOC, and CLOCK signals to the test points so they can be viewed on the logic analyzer. Download your ADC interface circuit to the *Xilinx* and write to the ADC register from the SingleStep command line. Verify that the START and ALE signals are asserted and measure their duration. Do they meet the minimum pulse width specification? Does EOC go low, then high? If not, debug your interface logic.

2. Once you observe the ADC control signals working properly in response to a write, verify that you can read a value from the ADC (again using the command line).

3. Once your hardware appears to work, debug your software. Again, verify one feature at a time in isolation before proceeding: the timer ISR, the EOC ISR, etc.

4. Build an adjustable voltage divider using resistors and a potentiometer. Connect a middle divider node to the ADC input while measuring the voltage using the DMM. Adjust the potentiometer until the digital value changes on the seven-segment display. Record this voltage. Vary the voltage back and forth across this boundary: how stable is the voltage at which the converter switches? Continue to vary the potentiometer until you find another digital code transition, and record the voltage.

5. Change the divider so the potentiometer is adjacent to ground, and use the other potentiometer terminal as the input voltage. Can you get the ADC to register a 0 code? At what voltage does it switch to 1?

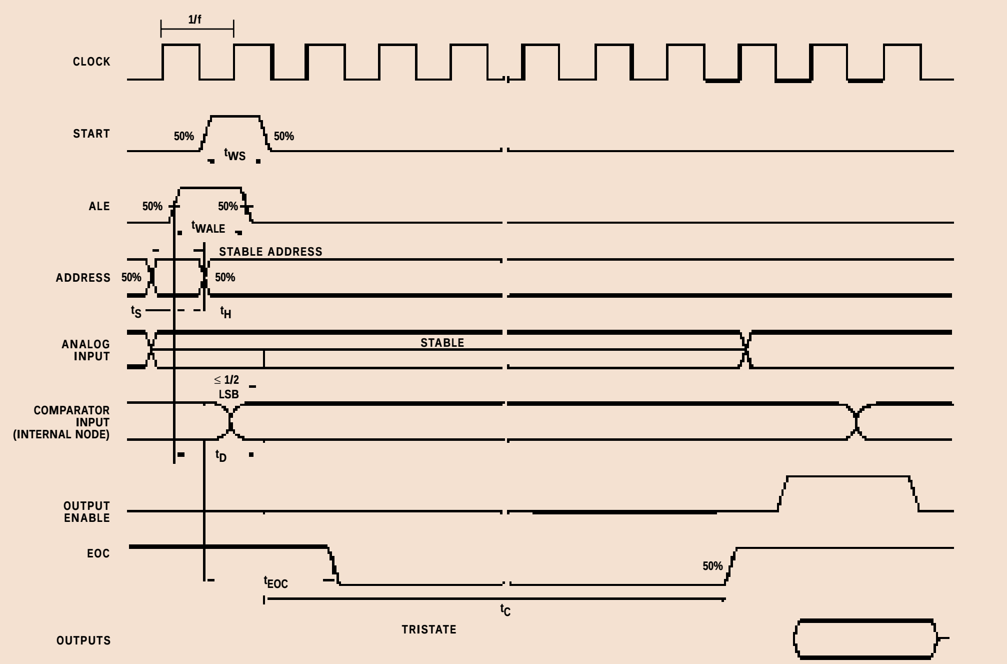
6. Repeat step 5 with the potentiometer at the 5 V side of the divider. Can you get the ADC to register a 255 code? At what voltage does it switch to 254?

7. Set up the waveform generator to output a 100 Hz sine wave, 4 V peak-to-peak amplitude with a DC offset of 2.5 volts. That is, the sine wave should go down to 0.5 V and up to 4.5 V. ***Use the oscilloscope to verify that the signal voltage is in this range before connecting it to the ADC. If the input voltage goes below 0 V or above 5 V, you could fry the ADC chip.***

8. Press S1 to capture 500 samples in your memory array. Stop your program and read the samples using the SingleStep command line. Do they look like they make sense? How many samples occur before the values cycle back to where they started? If everything looks OK, save the data in a file by using '>' to redirect the output of a 'read'.

9. Repeat steps 7 and 8 for sine waves of 200 Hz, 400 Hz, and 800 Hz.

10. Demonstrate your system.



Conversion timing diagram for the ADC0808. Used with permission from National Semiconductor Corporation.

Lab Report:

1. Summarize the operation of your circuit and describe how you arrived at your design. Include a printout of your schematic and a well-commented listing of your program.

2. Given the transition voltages you recorded in step 4, and the digital output, what is the device non-linearity at each transition in LSBs? What is the differential non-linearity for the step between the transitions? Does this agree with the data sheet specification?

3. What were the 0>1 and 254>255 transition voltages? What is the device non-linearity at each of these transitions? Does this agree with the data sheet specification?

4. Plot your data (using a spreadsheet or graphing program) for the 100 Hz, 200 Hz, 400 Hz, and 800 Hz sine waves. Are they all recognizable as sine waves? Estimate the signal frequency directly from each plot. What values do you arrive at? What does this tell you about the relationship between sampling rate and signal frequency?

Tools for Logic Analysis

Do your students know when they should use a scope and when a logic analyzer is a better choice? Because capabilities overlap somewhat, either one may be used in some cases. There's a valuable tutorial on the subject (reach it via **www.EducatorsCorner.com/links)** that should help your students understand which is the better tool for various applications.

HP 54645D Mixed-Signal Oscilloscope

HP 16702A Logic Analyzer

Mixed-Signal Scope

Here's a unique tool. The HP 54645D Mixed-Signal Oscilloscope can be used as a deep-memory scope in the analog lab or as a timing analyzer in the digital lab. Displaying up to 2 analog channels and 16 digital channels, its forte is debugging mixed analog-digital MCU-based designs. That also makes it a great fit for senior design labs and mechatronics labs. Students can immediately see the cause-and-effect relationships between digital and analog signals.

Logic Analyzer

If you need state analysis and measurement versatility, the HP 16600/700-Series logic analysis systems have both. Get the capability you need for your current classes, with the headroom to expand as your curriculum changes. As a bonus, optional emulation modules give you emulator functionality at a fraction of the normal cost.

If your budget is limited, the HP 1660/1670-Series logic analyzers offer pre-configured solutions with up to 136 logic analyzer channels, with optional integrated scope, integrated pattern generator or deep memory.

HP logic analyzers can be used effectively in many undergraduate labs, including digital design and microprocessor design—and they're great tools for senior projects and graduate and post-graduate research.

For product information on the HP 54645D or HP 16702A, see ***www.EducatorsCorner.com/links.***

The Pros and Cons of Remote Labs

"I hear...I forget; I see...I remember; I do...I understand." – Confucius, c. 500 BC

From Confucius to ABET 2000, from Beijing to Berlin and beyond, you'll find little dispute that laboratory exercises and experiments are an important adjunct to lectures when you're teaching engineering. Hands-on experience helps students understand the abstract concepts they hear and see in the classroom.

But just exactly how "hands-on" does a lab experience have to be? Can a student learn effectively using a remote lab, where actual test and measurement instruments can be controlled remotely via a local network or the Internet? What are the benefits and disadvantages of a remote lab? Will remote labs someday replace physical labs stocked with oscilloscopes, digital multimeters and teaching assistants?

We asked these questions of some of the visionary educators who have implemented remote labs at their universities. Everyone talked of the promise the technology holds. But the reviews were mixed when it came to evaluating the results of their remote lab experiences.

By interfacing instrumentation with the computer, data can be automated. In this picture, taken at Carnegie Mellon, the computer system had to be more flexible since the circuit under analysis was unknown (CMU lab: The Black Box, accessible via ***www.EducatorsCorner.com/links****).*

The Promise of Remote Labs

Why have universities devoted scarce resources to developing remote labs? None of the remote lab implementers surveyed expected their remote labs to replace their traditional physical laboratories. They cited specific applications where remote labs add measurable benefit or they viewed it as a supplement or enhancement to their traditional lab program. Remote lab benefits they mentioned include:

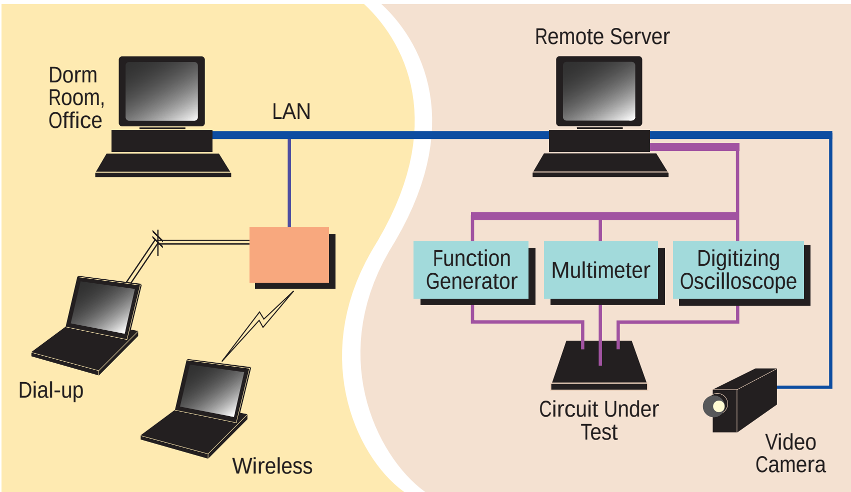
- Students can configure circuits and get results very quickly, which encourages them to do more "what if" exploring than they would in a physical lab.
- Students have the flexibility to log in and complete their assignments from any place in the world, and at any time they choose.
- Remote labs provide broader access to expensive and/or specialized equipment.
- A remote lab prepares students to work in "remote mode," a mode that some expect to become increasingly common in the workplace.
- Remote labs enable hands-on experiences for distance education.

Professor David Naylor of the University of Illinois at Chicago is a remote lab enthusiast. His remote lab has been in use for about a year. Students—or anyone else who wants to—can access his remote lab via a web browser or a downloadable windows application. Once they're online, they can build their own circuits, select circuits from a menu or import pre-built circuits.

"When students are first starting out, it's easier to just use a pre-built circuit," said Naylor. "It enables them to get results with the click of a button. There's relatively little they can do wrong at that point. Then the next exercise is where they've actually got to build the circuit themselves. It's a case of confidence building."

"We use the remote lab here at the university as a supplement to our regular hands-on lab," said Naylor. "We have no intention of taking away the traditional labs. In terms of the kinds of hands-on skills that a practicing engineer needs, I think a hands-on lab is very important. I use this facility to bridge the gap between what we are doing in the classroom and the kinds of things that we do in our traditional lab."

Professor Yi Zheng from St. Cloud State University advocates remote labs for a very specific application: developing software for a digital signal processor board. "The development tool for digital signal processing is quite expensive," CMU ECE Remote Laboratory



he said. "Most universities cannot afford to buy workstations for all their students. Microprocessors for digital signal processing change dramatically and frequently. If you buy ten systems today, a year later they will be outdated and your money will be wasted. But if students need to learn these skills, we have to find a way to do it. This is one way to make it possible."

The Reality

Carnegie Mellon University was one of the first engineering schools to use a remote lab for undergraduate students, beginning in the fall of 1995. Looking back at the school's remote lab history, Professor Daniel Stancil said, "Frankly, our experience at CMU has been mixed, although I continue to believe that remote experimentation is a powerful idea. I taught the remote course twice, each time drawing about ten students or less. This was suitable for experimenting, but was not a large enough level of interest to justify continued offerings. Rather than offering a specific dedicated course, we are now exploring ways that remote labs can be used to enhance existing courses in our curriculum."

Professor Zheng commented on the investment required to build and maintain a remote lab. "It is a big commitment in terms of both staff time and financial resources to develop and maintain a remote lab," he said. "If you don't have enough resources, it is very difficult." His school, St. Cloud University, spent close to $300,000 over a two-year period developing and equipping four remote labs: a beginning circuits lab, an electronics lab, a communications lab and a DSP lab.

According to Zheng's colleague, Professor Michael Heneghan, the weak point of most remote labs is that students can't set up circuits themselves. "But for distance education, it's better than using simulation or watching someone else perform a lab on interactive TV," he said.

Was it worth the $300,000 investment? "That remains to be seen," said Professor Heneghan, "but we are optimistic."

For more information on the remote labs at Carnegie Mellon University, St. Cloud State University and University of Illinois at Chicago, visit ***www.EducatorsCorner.com/links.***

Tips for Developing a Remote Lab

by Professor Daniel Stancil, Carnegie Mellon University

- *The software must be reliable. Frequent software crashes frustrate students.*
- *You must have the ability to power cycle the instruments being controlled to recover from crashes.*
- *The remote lab must be a required part of the course, or offer clear added value to students. If they don't perceive a significant benefit, they won't use the remote lab.*
- *Experiments must be sufficiently flexible to allow students to explore a wide range of questions or conditions. "Cookbook" labs with limited flexibility offer little attraction to bright students.*

A Way to Combine Software Simulation with Hands-On Labs

How do you effectively demonstrate complex concepts like fourier transforms or amplitude modulation to a room full of undergraduate students?

Even when you explain the theory and step them through the mathematical models and equations that describe the signals, the light bulb doesn't always go on. If you think back to your own student days, you'll probably agree that theories can be difficult to grasp when you're dealing purely with abstract concepts.

If, however, you could model the signals and produce them physically, your students could associate the abstract concepts with something more concrete. You could design experiments that give students insight into the mathematical representations of signals, as well as the behavior of those signals on physical devices and circuits.

You can use simulation software to model the signals, but then how do you transfer the calculated signals to a piece of test equipment where they can be produced? It's possible, but cumbersome. First you would generate the sample points of the calculated signal and save the data in a file or transfer it to the Windows clipboard. Then you'd have to write a program that would be capable of reading the data from the file and formatting it into a form that the instrument driver could then send to the instrument. Sounds like a lot of work.

There is a simpler way.

Using ActiveX Instrument Drivers

PC applications and test instruments handle data in different ways. Like a printer driver, an instrument driver is a piece of software that acts as the go-between for a computer program and a physical instrument. A good instrument driver deals with the different ways the application and the instrument represent data and makes it possible to transfer that data between the two.

Transferring simulation software data to hardware lab equipment can be easy if you use instrument drivers implemented as ActiveX Controls. ActiveX Controls are software components, based on PC standards, that allow developers to package software that can be reused by many different types of applications, like Microsoft® Office, Mathcad, or Matlab.

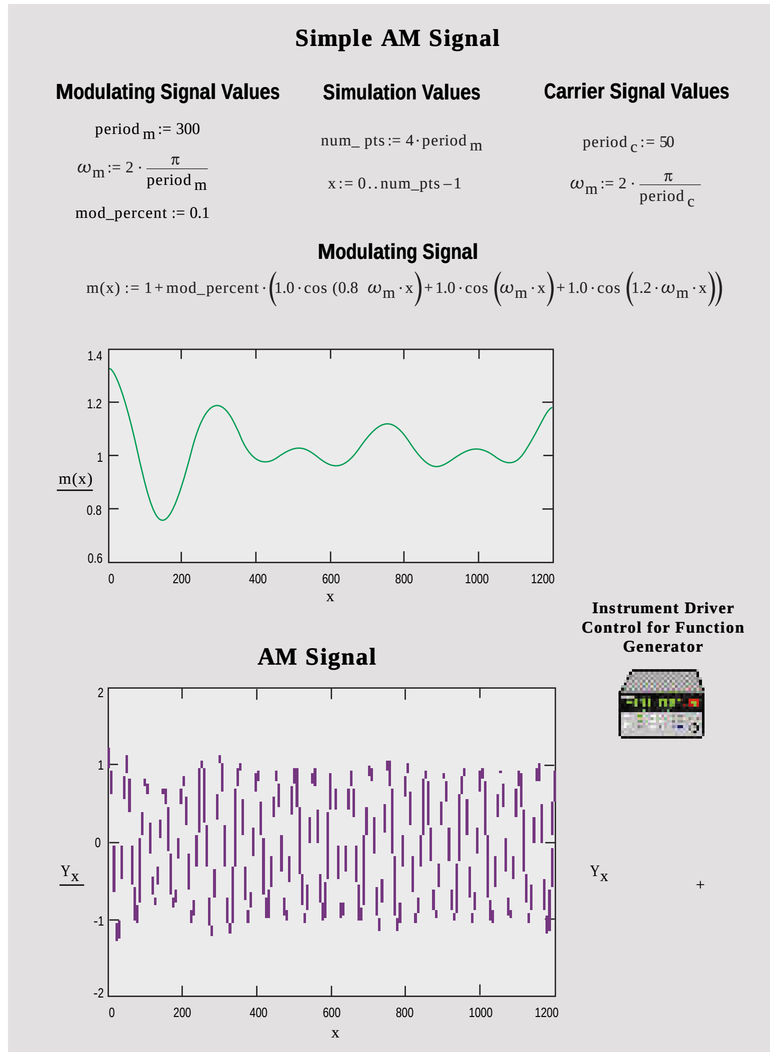


Figure 1: Mathcad sheet with AM equation and instrument driver control.

With an instrument driver implemented as an ActiveX Control, the same driver can be used both in traditional programming languages and in applications. In an application like Excel or Mathcad, you simply pop up the ActiveX toolbar that controls the instrument. There is no separate program to load. To the user, it's almost as if the instrument driver is simply one more command on the Excel toolbar.

As the popularity of using instrument drivers implemented as ActiveX Controls grows, look for more and more instrument vendors to offer these instrument drivers with their

Rick Hester
Connectivity Software Manager
HP Electronics Measurements Division

instruments. Companies like Hewlett-Packard, LeCroy and Keithley already have some instrument driver controls and will most likely offer more. When an instrument driver control isn't available for an instrument, development tools like Visual Basic or Visual C++ have some simple wizards that you can use along with a traditional instrument driver (one implemented as a dynamic link library) to make your own.

A Simple Example

To demonstrate using an instrument driver control with a software application, we'll use Mathcad software running on a personal computer which is connected with a HP 33120A Function/ Arbitrary Waveform Generator. First, we entered the equations for the signal into the Mathcad worksheet. To make it more interesting, we made the modulating signal a simple sum of three sinusoids. In Figure 1, you'll see the Mathcad sheet with equations and plots of their values. The variable x is an array of integer values that serve as distinct time sample points for the signal. The Yx array variable holds the sample values of the AM modulated signal and its reflection.

We added the instrument driver control to the Mathcad worksheet and transferred the Yx array variable through the instrument driver directly to the physical instrument. (See Fig. 1).

Transferring Data

Building upon the simple AM signal example, we can take the output of the arbitrary waveform generator and connect it to the input of an oscilloscope. With an instrument control for the oscilloscope, you can capture the measured waveform data and input it directly into a spreadsheet. Once it's in the spreadsheet, you can graph the data or process it in any number of ways. To aid students in understanding the data, you can capture a screen image of the oscilloscope and include it in the spreadsheet right along with the waveform data. (See Figure 2).

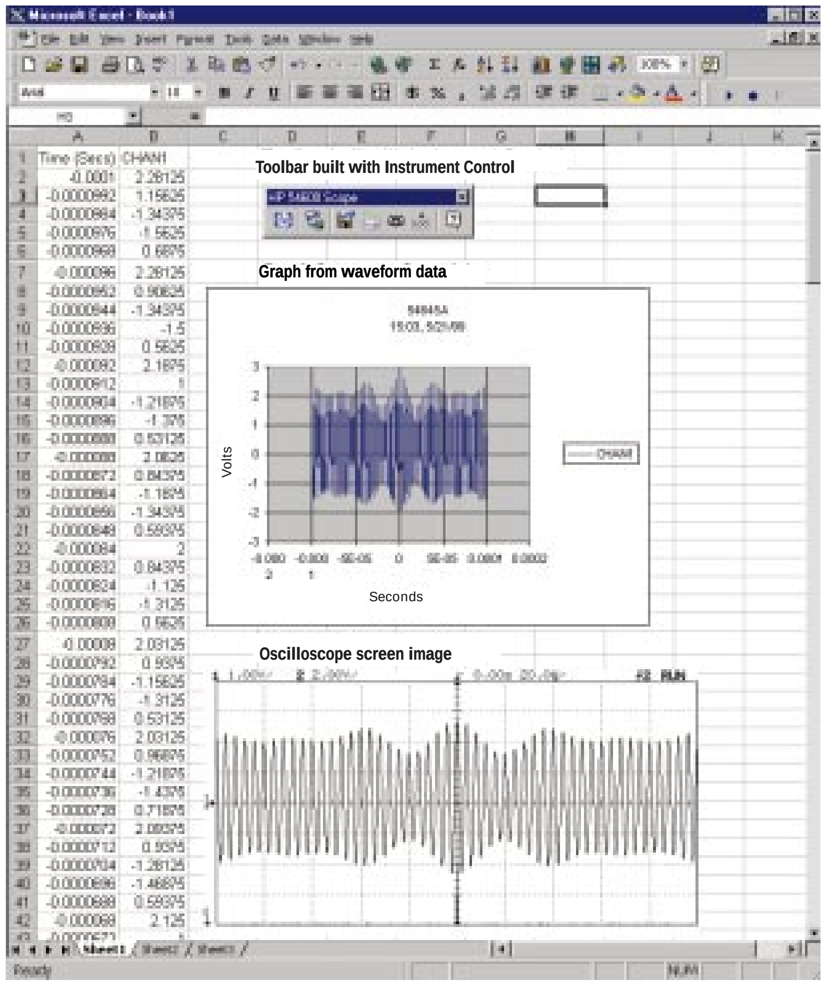
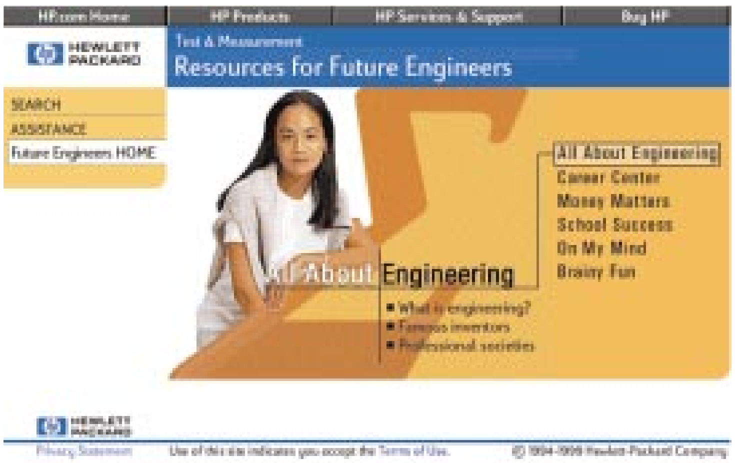


Figure 2: Spreadsheet worksheet with waveform data, graph, oscilloscope screen image, and toolbar from instrument control.

The instrument control also makes it easy for your students to document their work. Both data and screen images can be captured and used from within word processors.

For more information about ActiveX Controls, go to **www.EducatorsCorner.com/links.**

Website Update



Send Your Students to www.FutureEngineers.com

If your students haven't discovered **FutureEngineers.com** yet, do them a favor and point the way. They'll find a wealth of valuable resources for engineering students—from career tips to tutorials—that will help them be successful in school and beyond. And while they're visiting, they can take a humor break on the Brainy Fun page—it's filled with brain teasers, games and puzzles, cartoons and other goodies, all designed especially for the bright and curious students who typically are drawn to engineering.

Much of the material for the FutureEngineers site was contributed by engineering students from around the world. There are several opportunities on the site for your students to contribute their thoughts and suggestions, too. We plan to eventually build the site into an interactive global community for engineering students—a place for them to meet and talk, learn and laugh. Share the URL with your students today!
www.FutureEngineers.com

Here's a brief outline of the resources on the FutureEngineers site:

All About Engineering
• Through Engineers' Eyes • Through the Eyes of our Student Advisors • Professional Societies • Famous Inventors

School Success
• Tips from Professors • Tips from Students • Tips from Practicing Engineers • HP Educator's Corner • Basic Instrument Tutorials • Links to Tutorials

Career Center
• Interviewing: the Good, the Bad & the Ugly • Getting that First Job • Surviving that First Job • What Engineers Really Do

Money Matters
• Money-Saving Tips from Recent Graduates • The 5-1/2 Minute MBA • Show Me the Money

On My Mind
• Here's What I think • This Is How It All Began

Brainy Fun
• Games and Puzzles • Cartoons • Brain Teasers • Links to More Fun • Weird Discussions

Educator's Corner:
A More Logical Approach

Our first focus on **www.EducatorsCorner.com** was on simple DC circuits (see our interactive experiments). Then we moved to analog labs, audio-frequency AC circuits and fourier analysis. Next came RF circuits and tutorials. Today, we have a pretty good assortment of analog labs and tutorials.

But the world is not all analog, and we've been ignoring those of you who teach digital circuits. Your wait is about to come to an end. In the next few months, we hope to add a considerable amount of material on logic analysis and mixed-signal design. Check out the latest additions at **www.EducatorsCorner.com.**

HEWLETT® PACKARD

Hewlett-Packard Company
Test & Measurement Organization
P.O. Box 3828
Englewood, CO 80155-3828

BULK RATE
U.S. POSTAGE
PAID
LOVELAND, CO
PERMIT NO. 107
ZIP CODE 80538

Printed in the U.S.A. 9/99 *5968-6060E*