A Remote Wireless Sensing and Control Laboratory

Erik Cheever, Lynne A. Molter, Bruce A. Maxwell

**Department of Engineering, Swarthmore College,
Swarthmore, Pennsylvania 19063**

Abstract

A laboratory has been designed to provide students with hands-on experience integrating disparate technologies in laboratory exercises and independent projects. Specifically, three areas have been selected for the development of applications: 1) intelligent sensing, 2) web-based data acquisition and control and 3) wireless communication systems. Students can choose modules from any of these three areas to enhance their own work. The intelligent sensing modules give students easy access to video camera and microphone data streams so they can rapidly develop interfaces responsive to those modalities. Students can connect analog and digital electronics they have designed to the web-based modules that consist of small, inexpensive, web-servers. This lets them use a web browser for sophisticated user interfaces, for remote control over the internet, or for both. Only minimal electronics resources (i.e., a serial port) and student time are required. Bluetooth™ technology was chosen to implement the wireless modules. Software drivers have been written to allow an inexpensive Bluetooth receiver/transmitter set to be used to network small microcontrollers so that they can perform wireless data acquisition and control. This nicely complements the web-based system. The students also have access to a Bluetooth protocol analyzer so they can see individual packets as they move around the wireless network. This is both a useful instructional aid as well as an important debugging tool for students working with the receiver/transmitter sets. Projects can incorporate any or all of these modules to allow students across disciplines to develop sophisticated remote sensing and control applications. Typical applications that integrate all three components include an intelligent monitoring and security system, and an energy conservation system based on monitoring room usage.

Introduction

Students are especially motivated when they have an opportunity for hands-on experience. Another strong factor motivating their interests is an increased understanding of the interconnections among the different courses they take and the concepts they learn as they become more sophisticated in their knowledge and understanding of a field. The Remote Wireless Sensing and Control Laboratory enables students to explore more easily the connections between and among a wide range of courses. One way to accomplish this is to use a system in fundamentally different ways in different courses. For example, students in one course can do an in depth analysis of the modulation scheme of a Bluetooth radio transceiver course,

while those in another course may simply use the transceiver as the means for wirelessly transmitting data collected in an experiment.

To allow for a broad range of synergies and to enable a wide variety of different experiments, hardware and/or software resources were developed in three disparate, but related, areas. The first of these is a system for intelligent sensing, specifically a system that has the ability to capture and analyze video and audio streams to, for example, identify when people are in the room. The second system consists of using miniature embedded web-servers to provide a user interface to student-built electronics. This facilitates the design of complex user-interfaces, while simultaneously greatly reducing costs. The third system involves both characterization and applications of a wireless personal communication system using Bluetooth transceivers. Commercial Bluetooth hardware together with student-developed software drivers enable microcontrollers to communicate with each other through a Bluetooth network with minimal allocations of resources, hardware or software, on the part of the microcontroller. Each of these areas is developed in some detail below, along with some possibilities for combining the work into integrated student projects.

Intelligent Sensing System

One aspect of this project is to develop intelligent sensors, sensors that do more than just report raw data. For example, raw video data is not extremely useful except perhaps as a motion detector. The combination of a computer and a camera, however, can create a sensor that is capable of detecting whether a person is in the lab and possibly where that person is currently located. This information, in turn, can be used to manage heating and lighting systems, security systems, or even to move desktops between computers so that the computer closest to the user always has their current desktop available. The ultimate vision is to think of the lab as a single intelligent system that senses and reacts to its environment and users in an appropriate manner.

In the development of this capability we have initially focused on detecting and tracking people using color CCD cameras connected to Linux workstations. The cameras are also equipped with microphones, which enables a second, independent method of sensing the lab environment. We have so far undertaken two projects with this equipment.

The first project is the development of the low level interfaces to the camera and microphone. These interfaces are designed to enable students to build more sophisticated projects using the sensory data without having to worry about the device specific details.

The video interface is a collection of routines that uses the Video4Linux interface to initialize the framegrabber, begin image capture, synchronize the image capture process at particular points in the program, and close down the framegrabber cleanly. To give students a specific example of how to use these routines we also developed a program that displays live video in an X-Window and responds to mouse clicks and keyboard presses inside the window's focus. This example program makes it easy for students to get access to the current video images and display results in the X-Window in real time. The live image display is crucial to debugging even simple real-time image algorithms.

The microphone interface is a set of routines, similar to the video routines, that enable the user to begin sound capture, get access to the data stream, and end the sound capture. As an example of using the microphone data, we developed a program that combines both the live video and live sound sensing. When the user makes a loud enough noise, the program grabs the current image and saves it to disk.

We have used the video interface in the Fundamentals of Digital Systems course to implement particle filters for tracking faces and hands (skin color) (the skin is shown as dark patches in Figure 1)[1]. The example and interface routines enabled the students to complete the project within two weeks and focus on the particle filter concept rather than the video interface. We will also be using the interface in a Computer Vision course to be taught in the fall of '03.



Figure 1

Once we have a reasonable basic capability for detecting and locating people, we can integrate this intelligent sensor with the wireless communication and control systems to make the lab more responsive to the users, such as adjusting climate controls or lighting as appropriate.

Web-Based Data Acquisition and Control System

The world-wide-web provides a very powerful, and nearly universal, method for transmission of information. The web can be an extremely useful tool for data acquisition and control because it enables the collection of sensory information, as well as the control of systems, from anywhere in the world. However, the web has not traditionally been a viable alternative for a wide variety of purposes because it has needed a relatively powerful, and expensive, processor to act as a web server. The recent development of embedded web servers has begun to change this.

The advent of these inexpensive web-servers opens a new set of possibilities for designing interfaces to sensors. One of the most difficult aspects of designing a practical sensor or actuator is to develop the user interface. Typically, for student projects, the interface consists of several push buttons, some LED's and maybe a one- or two-line LCD screen. Interfaces that are more elaborate are available, but their use incurs a high cost – both financially and in terms of student

time. This is especially true when the thrust of the project is the sensor or actuator instead of the interface. One method around this difficulty is to use an embedded web-server to generate web pages using data retrieved from a microcontroller. In this way, interfaces can be built that are, for practical purposes, arbitrarily complex by just writing some HTML (HyperText Markup Language) code. The user interaction is also very intuitive since every student in the lab will be familiar with a web browser's familiar interface. Web-servers are available for fewer than thirty dollars, and the only physical resource needed to communicate is a serial port, available on any microcontroller. An added benefit is that the device is accessible from *any* computer with a web browser connected to the world-wide-web, though that is not the major goal of this work.

Figure 2 shows a block diagram of the web-based system. Note that the diagram is generic; the specific content of each block is not important and therefore not shown in detail.

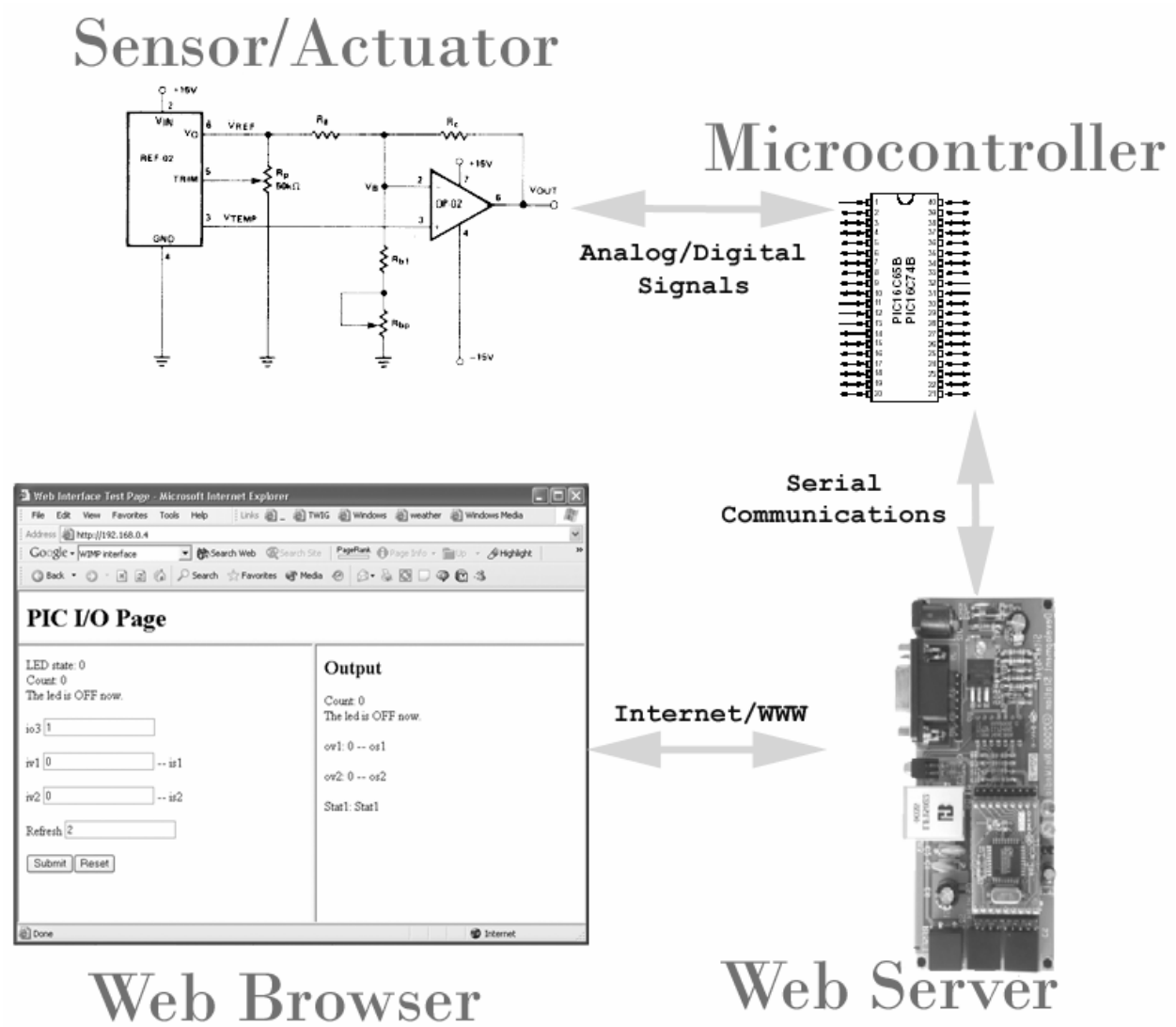


Figure 2

Originally, the web-server was implemented with a standard web-server (a small embedded-server) and custom CGI (Common Gateway Interface) scripts. About midway through the

development process, NetMedia introduced a new device called SitePlayer™. This device incorporates all the functionality necessary at about one-fifth the cost of the embedded server, and obviated the need to write any CGI programs. For these reasons, the student experiments will use SitePlayer.

The SitePlayer web server is integrated onto a small circuit board (less than 8 cm^2) that uses a variant of HTML that allows the user to insert variables into the HTML. When the web page is served, these variables are retrieved from memory and displayed on the web page. There is also a simple mechanism for the browser to be used to change the values of variables on the web server. To complete the block diagram from above, a simple protocol is defined so the microcontroller can read from, and write to, the web server's memory.

An example of a page served by a SitePlayer is shown in Figure 3. In this example, the program on the microcontroller is counting up with a variable that is displayed as "Count". The variables labeled "ov1" and "ov2" are variables that are one less than "Count", and the sum of "iv1" and "iv2", respectively. The additional input variable ("LED" and "Refresh") determine whether or not an LED is on, and the refresh interval (in seconds) for the rightmost frame of the web page. The leftmost frame is refreshed when the "Submit" button is pushed.

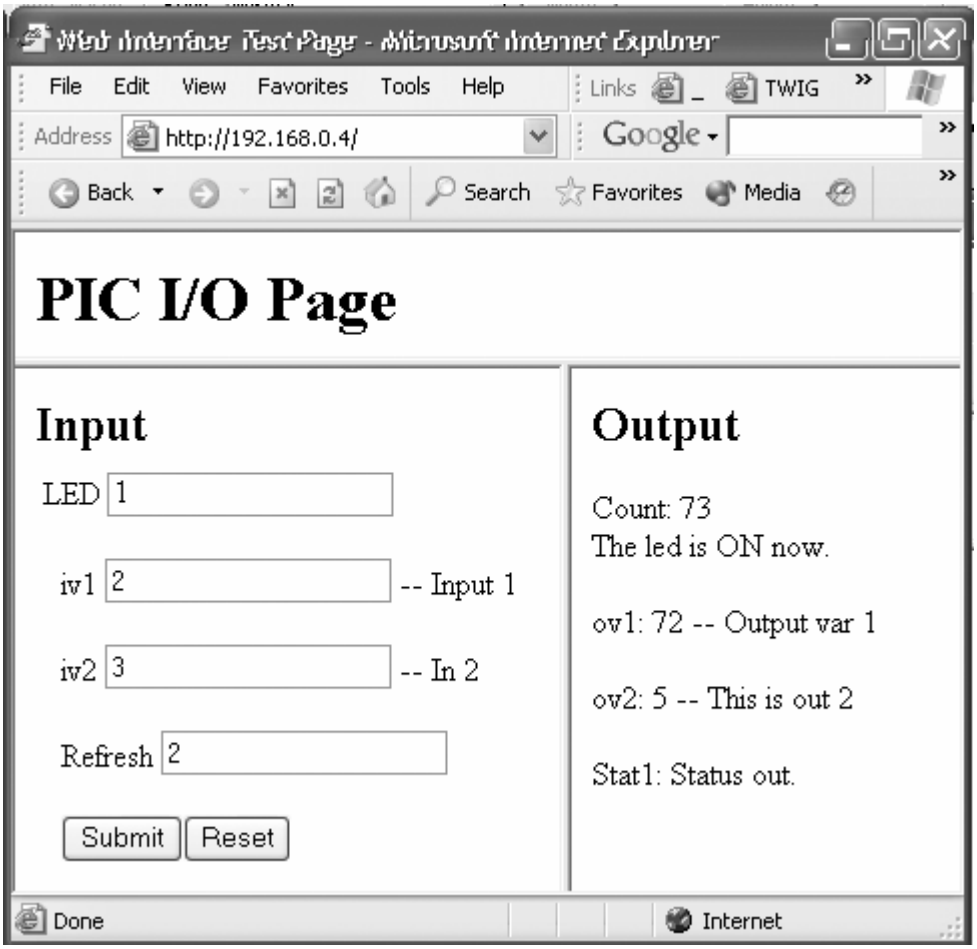


Figure 3

The role of the web server is most easily shown by example. Figure 4 shows the rightmost frame of the page displayed by the browser to the left along with the body of the HTML code used to generate it to the right. Each variable that is stored on the server is denoted in the HTML file by a name preceded by a caret (^), and is highlighted in gray on the browser output. For instance, where the variable name '^*counter*' appears in HTML, the value of that variable stored on the server is displayed. Likewise, the variables '^*ov1*', '^*ov2*', '^*os1*', '^*os2*' and '^*Stat1*' are displayed by the server. Lines 15 to 22 of the HTML comprise a short JavaScript function which reads the variable '^*io3'* (which is connected to an LED (light emitting diode)) and displays a short message signifying whether or not the LED is on.

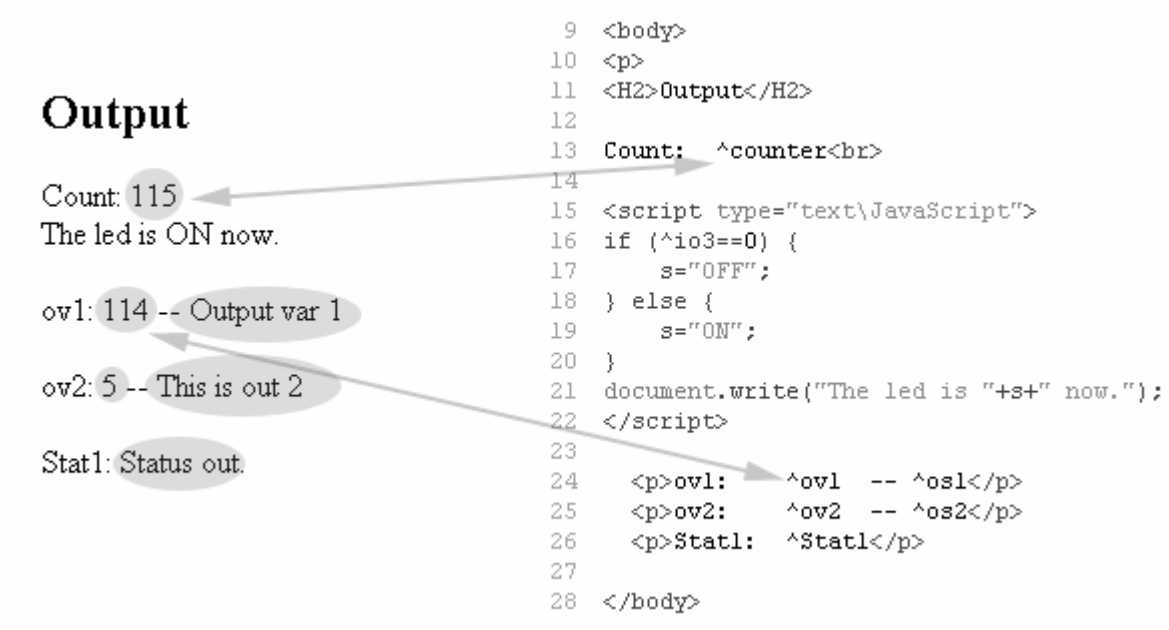


Figure 4

There is a similar simple mechanism to use forms (or JavaScript) to change the value of variables stored on the server. For more information, refer to the documentation at the SitePlayer web site[2].

These web servers enable the development of sophisticated GUI (Graphical User Interface) interfaces for a microcontroller while using minimal resources. Of course, using an embedded web server as a GUI has some shortcomings. The web server has limited memory to store variables (768 bytes) and web pages (48 kbytes). There is another limitation because there is no way for the web server to change the information displayed on it, except through a periodic refresh, or if the user requests a refresh. For the proposed application, none of these is a serious limitation.

Wireless Communication System

One of the major goals of the project is to enable students to explore the use of wireless communication between devices. This exploration entails not only the use of wireless communications, but also the analysis and characterization of the individual components of the

system (e.g., spread spectrum techniques, voltage controlled oscillators, phase locked loops, RF devices such as modulators, filters, and antennas, etc.), and the investigation of properties of signals and their transmission, the effects of noise, and methods of coding and encryption. Several modalities were explored including various cellular phone standards and IEEE 802.1, but Bluetooth technology was chosen for its low cost, low power and its ability to be incorporated into small networks.

The Bluetooth standard specifies a communications protocol for dynamic networking whereby devices may establish communications with each other merely by coming into proximity. A typical application of Bluetooth has a Master interacting with up to seven Slaves joining to form a *piconet*. A device can be active on more than one piconet, effectively joining the piconets in a *scatternet*. The Bluetooth protocol stack specifies how data to and from a host processor is packetized and sent to the baseband radio link. The radio link operates near 2.4 GHz in the Industrial, Scientific and Medical frequency band and uses a frequency hopping technique. The Bluetooth hardware system used in this work consists of BlueCore™ hardware modules and BlueLab™ software development kit along with the Casira™ development system from Cambridge Silicon Radio[3].

Though the Bluetooth hardware and software allow a rich assortment of options for experiments, the thrust of the work described here has been to allow two microcontrollers to communicate over a Bluetooth link while using a minimal amount of the microcontroller resources. In this scheme, a microcontroller is connected to a BlueCore module on an Inter-Integrated Circuit (I^2C) bus that only takes two pins on the microcontroller. The Bluetooth modules then enable the two microcontrollers to communicate together as if connected via the I^2C bus (Figure 5)

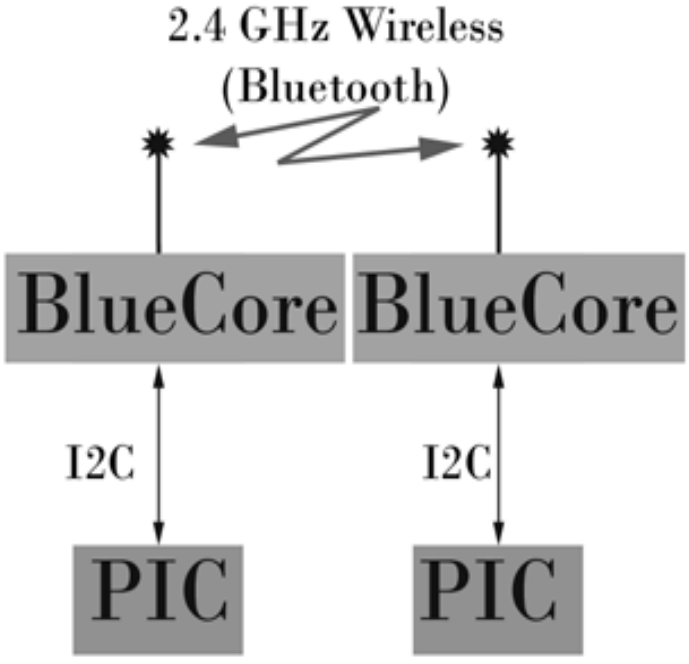


Figure 5

A set of software routines makes the Bluetooth connection unobtrusive to the programmer of the microcontroller. On the BlueCore end, the software initiates contact between the BlueCore modules and with the microcontroller, an manages information flow between the I²C bus and the wireless connection. On the microcontroller side, the software is a set of routines to initiate contact with the BlueCore module and manage information flow on the I²C bus. As different applications are written for the microcontroller, the number of lines of code that must be written that are dedicated to communications is minimal.

Any number of applications is possible with this scheme. One completed application involves communication between a wireless thermostat and a zone-valve for a home-heating system. In this scheme there is a microcontroller running a thermostat application. A user can choose a set temperature, and the thermostat measures the room temperature. The zone-valve has another microcontroller that receives the set and room temperature, compares them, and either opens or closes the valve, depending on which is greater. The zone-valve and thermostat are part of a larger system to allow off-grid heating of a home. In a completed system, the zone-valve might communicate its state to a boiler that would turn on, or off, as appropriate.

As implemented, the wireless system does not use the full capabilities of the Bluetooth protocol. Currently communication is only possible between two devices, and they must know the address of the other device before communications can begin. The software is being revised to allow for dynamic connections in which new devices can enter the piconet with their address determined automatically, i.e., connections need not be determined a priori. Other improvements include attempts to increase data rates by reducing the amount of handshaking performed (which is now much more than the standard requires), and enabling one-to-many communications in which one device can communicate with several others simultaneously instead of one at a time.

Future Work: Integrating the Systems

There are myriad possible applications for the modules that comprise the Remote Wireless Sensing and Control Laboratory. An interesting application, easily understood by students, which uses all of the modules, would be an environmental control system for a laboratory. A microcontroller could implement a programmable thermostat for the room, sensing its temperature and opening and closing valves on radiators, perhaps lowering the set temperature at night when the lab is likely to be unused. The user interface for the thermostat could be a web page, obviating the need for expensive, single-purpose, user interfaces, and the concomitant complex programming. In a large lab, each radiator could have its own temperature sensor and radiator valve control, wirelessly coupled to a central microcontroller. Since the use-pattern of a lab may be unpredictable, there would also have to be provision to override the controller. For example, students going in on the weekend may inform the controller (perhaps over the web) of the time they expect to come in to the lab, and the temperature in the room could be set accordingly. Lights could be set to turn on and off as people enter and leave the room, activated by the vision system. In a large lab, only part of the room needs to be lit (and heated), as determined by vision system, which could use wireless actuators to control individual lights and radiators. Any aspect of this scenario is a likely candidate for development as part of a course.

Conclusions

The individual components of the Remote Wireless Sensing and Control Laboratory are all operational. The vision module was used to develop face detection algorithms in a Digital Systems course in the fall of 2002, the Bluetooth module will be used as part of a Senior Design class in the spring of 2003, and the web-server modules will be used in several classes in the 2003-2004 academic year. As students, and faculty, become more conversant with the capabilities of the lab, the applications developed will increase in sophistication, and the projects developed will become more integrated.

References

[1] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework", Lecture Notes in Computer Science, v1406, pp, 893-908, 1998.

[2] www.siteplayer.com

[3] www.csr.com

Biographical Sketches

ERIK CHEEVER is a Professor of Engineering at Swarthmore College. He obtained his BS in Engineering from Swarthmore, and an M.S. and Ph.D. in Bioengineering from the University of Pennsylvania. He teaches mostly in the area of linear systems and electronics. His research currently involves using genetic algorithms to design analog circuits.

LYNNE A. MOLTER is a Professor of Engineering at Swarthmore College, where she has been teaching for 16 years. She earned the B.S. in Engineering and B.A. in Mathematics from Swarthmore in 1979, and the S.M. and Sc.D. in Electrical Engineering and Computer Science from MIT in 1983 and 1987, respectively. Her research is in the area of optics and photonics.

BRUCE A. MAXWELL is an Associate Professor of Engineering at Swarthmore College. He earned a BS in Engineering and BA in Political Science from Swarthmore College, a M.Phil. from Cambridge University and a Ph.D. Carnegie Mellon. His research interests include graphics, vision, robotics and data mining.