USING ROBOTS TO TEACH COMPLEX REAL TIME EMBEDDED SYSTEM CONCEPTS

Steven F. Barrett[1], Daniel J. Pack[2], Pamela Beavis[1], Mahbub Sardar[1],
Austin Griffith[1], Michael Stephens[1], Julie Sandberg[1], Yi Shi[1],
Lewis Sircin[1], George Janack[1]

[1]Department of Electrical and Computer Engineering
University of Wyoming

[2]Department of Electrical and Computer Engineering
United State Air Force Academy, Colorado

Abstract

Real Time Embedded Systems, also known as Real Time Operating Systems (RTOS), operate in environments where multiple events compete for precious processor operating time. To ensure that all events complete their tasks within a reasonable time frame, the processor must prioritize tasks depending on system requirements. Such real time operating system concepts and the related advanced embedded system topics are challenging to learn for many students mainly due to their technical complexities. One of the most challenging tasks for students is to visualize the intricacies and inter-relationships between different processes of an operating system. In this paper, we present a low-cost, motivational (fun) robotics platform that can significantly enhance the laboratory instruction of advanced real time embedded systems concepts. The robot was developed by a team consisting of faculty members, graduate students, undergraduate students, and laboratory technicians for a senior/graduate level electrical and computer engineering course. In this course each two-student laboratory team is issued a mobile robot for use throughout the course. At the beginning of the semester, the students must program basic tasks such as robot movement and maze wall-detection. The complexity of required programming tasks escalates as the semester progresses. Students are required to program the operating system for the robot that must simultaneously handle wall-detection, and simulated land mine detection, while navigating a maze. These complex scenarios force the students to learn and employ desired real time embedded systems concepts in a motivational atmosphere. This paper discusses robot hardware and software and their use in a real time embedded systems course. Furthermore, we also discuss how one can extend the use of robots to teach other advanced embedded system concepts such as multiple interrupts, fundamentals of fuzzy logic, and structured design techniques.

Overview

When teaching advanced microcontroller theory courses, it is challenging to teach complex concepts such as real time operating systems, multiple interrupts, fuzzy logic, and structured design techniques, which are not readily visible to the students. If paper exercises are used to illustrate these concepts, students often view them as esoteric, obtuse, and dry.

In an attempt to remedy the situation, we developed a low-cost, educational robot platform called PROFBOT (Professor Robot) using off-the-shelf components. To support the use of this robot in a laboratory environment, we also developed a series of laboratory assignments to illustrate the advanced concepts mentioned above. Also, by the nature of the laboratory sequence, students discover on their own the importance of employing structured design techniques during system development.

Using robots as a motivational, educational tool in a laboratory environment is not a new idea. This idea has been used at many institutions[1,2] with great success including Trinity College in Hartford, CT[3,4]; the Massachusetts Institute of Technology[5], and the US Air Force Academy.[6-8] We have extended this idea to teach advanced real time embedded system concepts.

In this paper we provide background information to illustrate some of the difficulties in teaching these upper level concepts, our proposed solution to this educational challenge, software and hardware solution details of the robot, the educational approach used to incorporate these concepts into the laboratory, and the results of our efforts.

Background

In this section we define the advanced concepts taught using the PROFBOT educational robots:

Real Time Operating System (RTOS): "A RTOS is a computer operating system hosted on a single processor. Since only a single, sequential processor is employed in such applications, the operating system must respond to multiple events or tasks and ensure that all tasks are given sufficient, precious processing time to complete their required actions.[10] To accommodate multiple tasks, the tasks may be categorized into priorities (high, medium, and low). In general, higher priority tasks should be executed by the processor first; however, the processor must ensure that eventually all tasks are allowed to complete their required actions.[12-14]

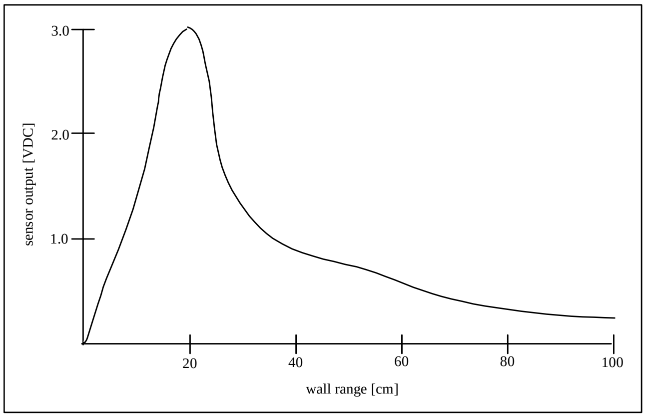
An effective RTOS system responds to all tasks competing for the same precious processor time in such a manner that multiple tasks appear to be handled simultaneously. How can a single processor do this? The processor typically uses various scheduling algorithms to allow the highest priority task to execute for some amount of time. The processor will then temporarily suspend the executing task, store details about its current executing environment (its context)[12-15], and then allow a different task to operate for some length of time. An effective scheduling algorithm will appear as if all tasks are operating simultaneously while seamlessly handing off processor execution time from task-to-task.

By nature the material is quite complex and dynamic. For example, tasks are constantly changing states and the multiple linked lists used to track system status must be constantly updated during system operation. For a student to fully understand RTOS, we believe a visual picture of the system is required.[16]"

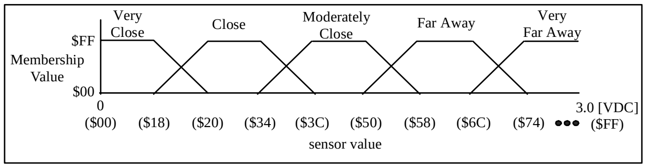
Multiple Interrupts: An interrupt is a planned, but unscheduled event that occurs during the execution of a program. Usually interrupt related tasks are of higher importance than the processor program currently being executed. Therefore, the processor will temporarily suspend its normal processing and execute the pre-planned interrupt related tasks. These interrupt related tasks are usually written in the form of an interrupt service routine (ISR). Once the ISR related tasks are complete, the processor resumes execution of the program that was in place before the occurrence of the interrupt.

In a system containing multiple interrupts, the system designer (the student) must carefully consider the interaction of these interrupts since there is a possibility they may occur simultaneously. Students must consider the priority of each of the individual interrupts and determine required actions for the interrupts occurring individually or simultaneously.

Fuzzy Logic: Fuzzy logic is in the domain of soft computing. The underlying concepts of fuzzy logic were originally developed by Lofti Zadeh at the University of California at Berkley in 1965.[17] Fuzzy logic concepts allow for the control of complex systems without the need of a mathematical model of the system. In place of the mathematical system model, a set of inference based rules are used to link system inputs to desired system outputs. Furthermore, instead of basing system output response on absolute system input thresholds, system inputs are mapped into input membership functions as shown in Figure 1. Typically, the input membership functions consist of a set of overlapping linguistic labeled trapezoids. The input membership functions map actual system inputs into ***fuzzy*** linguistic variables with associated confidence factors, sometimes referred as membership values. For example, in Figure 1 a) the output from an infrared sensor varies from 0 to 3 VDC. The sensor output voltage is mapped to a series of linguistic variables to indicate how far the sensor is from a wall as shown in Figure 1 b). System inputs are combined using a set of inferred rules which determine the desired system response. The system output is also a fuzzy linguistic variable which must be converted (***defuzzified***) to a crisp system output.



a) typical range response for a Sharp Electronics GP2D12 infrared (IR) sensor



b) Sensor Input Membership Functions
(right and left sensors will have similar functions)

Figure 1 a) infrared sensor output and b) sensor output mapped to a fuzzy logic based sensor input membership function.

Structured Design Techniques: The overall goal of structured design is to provide tools to transform system requirements into a plan to implement the system. A thorough discussion of structured design would include topics of "divide-and-conquer" techniques, structure charts, universal machine language (UML) activity diagrams, implementation and testing techniques, and thorough system documentation procedures.[10] Novice system designers often want to brush these topics aside and plunge into hardware and software designs. This latter approach rarely leads to working, easily maintainable and modifiable systems that meet established system requirements.

Methods

In an attempt to enliven the instruction of these advanced topics we have developed a low cost, robot-based educational platform using off-the-shelf components. We have also developed a series of laboratory experiments to support the use of the robot. As we have done in other educational projects[7,8,18], student based research was employed to design and fabricate the robots, develop the laboratory exercises, and develop solutions for each exercise.

The Hardware. The PROFBOT robot prototype was developed with the goal of designing a low cost robot, using off-the-shelf components, that would be easy to fabricate initially and easy to maintain. Early attempts at developing a prototype included a completely in house design.[19,20] However, the fabrication costs rendered this design cost prohibitive. We elected to use an off-the-shelf approach when a student-developed prototype proved to be a sound, low cost alternative to early in house attempts.[21] The prototype PROFBOT robot is illustrated in Figures 2 and 3.

The design is a two-platform robot equipped with servo motor driven wheels. This basic platform configuration equipped with motors and wheels is available as a low cost (US $50) kit from Budget Robotics. The leading edge of the robot is equipped with four infrared emitter-detector pairs (Sharp GP2D12, US $12.95 each). Power is provided to the robot from one of two sources: 1) a rechargeable NiCad battery pack (Tower Hobbies, US $25 per battery pair) or a 2) 7.5 VDC, 1.1 A power supply (Jameco, US $10). The sensor and power subsystem are interfaced to the host microcontroller via an in house designed and produced printed circuit board (PCB). The cost to fabricate and populate the PCB is approximately US $30 per board. The host processor is the Minidragon+ evaluation board (EVB) (US $119) available from Wytec Corporation. The Minidragon+ includes the host Motorola MC9S512DP256 processor with support electronics, edge connectors for easy interface, and a small protoboard prototype area. The EVB cost also includes a power supply, serial interface cable to a host PC, and support documentation. It should be noted that the EVB is equipped with 12K bytes of user RAM, 4k bytes of byte-addressable EEPROM, and 256K bytes of Flash EEPROM. The total parts' cost of the complete robot is US $295 per robot. A detailed parts list is provided in Table 1.

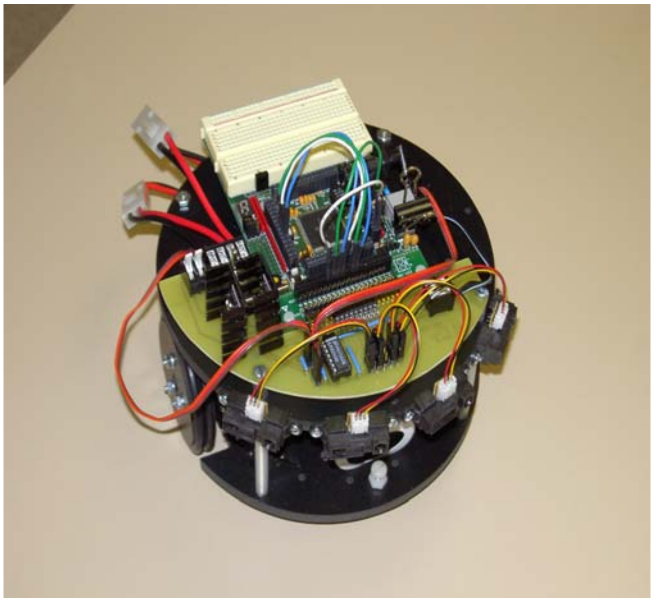


Figure 2. PROFBOT Educational Robot.

The Maze. The robot prototype was designed to operate autonomously in an unknown maze. The robot is placed at a designated starting point in the maze and then proceeds to an exit point. When in transit from the starting location to the end destination, the robot must sense and avoid collisions with the walls. The robot carries its own portable power source, senses walls with its onboard sensors, and navigates through the maze by sending appropriate pulse width modulated drive signals to its motor powered wheels.

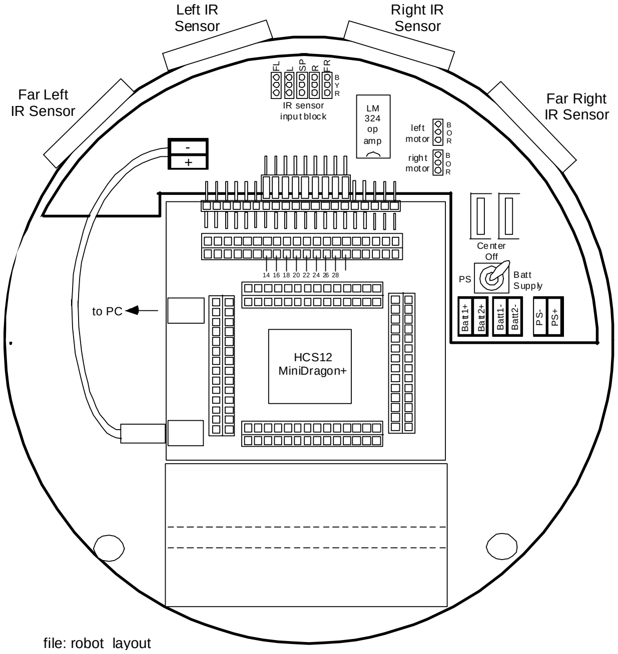


Figure 3. PROFBOT robot layout.

As with other project components, a reconfigurable maze was designed and fabricated by two engineering students. The two students completed this project as a portion of a research experience program for freshman engineering students during the summer. The maze is equipped with walls that are easily moved to allow creation of a new maze on every test run. The maze is illustrated in Figure 4.

Table 1. Robot component part list.

Part Description	Quantity	Manufacturer	Manufacturer Part #	Price Each	Total Price
Robot Platform w/wheels and motors	1	Budget Robotics (www.budgetrobotics.com)	KIT250-II	$54.95	$54.95
Infrared Emitter Detector Pairs	4	Acroname (www.acroname.com)	Sharp GP2D12	$12.95	$51.80
Rechargeable Battery Pack, 7.2 VDC	2	Tower Hobbies (www.TowerHobbies.com)	Available under various part numbers	$25.00/ pair	$25.00
Minidragon+ Evaluation Board (includes MC9S512DP256) with female header connectors	1	Wytec www.wytec.com and www.evbplus.com	Minigragon +/FH	$119.00	$119.00
Power Supply (7.5 VDC, 1.1A)	1	Jameco www.Jameco.com	280250	$6.29	$6.29
Battery cable connectors	2	Radio Shack	23-444	$2.99	$5.98
Aluminum Heat Sinks	2	Radio Shack	276-1363	$1.29	$2.58
Printed Circuit Board w/components	1	Local Manufacture	--	$30.00	$30.00
	1	- DPDT switch (on-off-on)			
	2	- 7805 5 VDC regulator			
	1	- LM324 operational amplifier			
	4	- power supply header blocks			
	1	- edge connector			
	6	- ½” circuit board spacers			
				Total	**$295.60**

The Software. The support software for PROFBOT was developed in tandem with the laboratory exercises. The primary author (sb) and a graduate student (pb) developed a list of laboratory assignments that allow students to start with basic robot operation and expand their capabilities over a semester period to develop a full-fledged real time embedded system. The laboratory sequence is detailed in Table 2.

Students work in teams practice teamwork skills and implement structured designed principles taught in the course. On several of the laboratory assignments, students are provided sample code as a starting point for their solution. The sample code emphasizes the importance of good documentation since the students must rely on the documentation provided with the sample code. The sample code provided for selected assignments is listed in Table 2.

Once the laboratory sequence and associated laboratory handouts were developed, an undergraduate student(ag) worked the laboratory exercises and developed detailed laboratory solutions. Most importantly the student was able to provide timely feedback on the completeness, accuracy, and difficulty of each laboratory assignment.

Table 2. Laboratory Exercises.

Lab Title	**Description**	**Subsystems/Concepts Employed**	**2 hr lab periods**
Introduction to the HCS12 Minidragon	Students use the laboratory compiler to write, debug, download, and test a simple program to the target robot system.	EVB and compiler operation, memory maps	1
Logic Analyzers	Students complete a laboratory exercise on serial communications and use of the logic analyzer as a troubleshooting tool.	logic analyzers, serial communications	1
Robot Vision	Students equip the robot platform with vision using four IR sensors. Each sensor is categorized (voltage vs. range to target) using the analog-to-digital conversion system on the processor.	IR sensors, analog to digital conversion	1
Robot Motor Speed Control	Students equip the robot platform with movement. Using pulse width modulation motor speed control techniques, students must write and test functions to provide robot movement (forward, hard left, bare left, hard right, and bare right). **Sample code:** Students are provided sample code to initialize the PWM system for the processor and to move the robot forward.	pulse width modulation, servo motor speed control, robot movement	1
Robot Operating System I - polling	Using structured design techniques students develop a simple operating system based on polling techniques. IR sensor data is collected and then used to render robot turns such that the robot autonomously steers through an unknown maze.	algorithm development, multi-subsystem operation, structured design techniques	2
Robot Operating System II – polling with interrupts	Students are provided a fifth IR sensor that senses “land mines” (2 cm white spots) on the maze floor. When a land mine is detected the robot will execute evasive movements to avoid the landmine. Also, their previous polling based operating system must be converted to interrupt only operation.	interrupts, multiple interrupts, algorithm design, real time concepts, structured design techniques	2
Robot Communication	Networked processor concepts are examined using the msCAN features of the processor. Students provide communications between the robots and display received messages on a liquid crystal display. **Sample code:** Students are provided sample code for msCAN transmit and receive and basic LCD display functions.	networked processors (msCAN) and liquid crystal display (LCD) interface	2
Robot Operating System III – fuzzy logic	Students must “surgically” remove their interrupt driven control system and replace it with a control system based on fuzzy logic concepts. **Sample code:** Students are provided sample code for a three sensor fuzzy logic robot control system. They must adapt this code for four sensors.	fuzzy logic concepts, algorithm development, structured design techniques	2
Robot Maze Competition	Student robots compete to find out which robot can navigate as quickly as possible through a maze while avoiding wall and landmine contact. Students are allowed to use the control system of their choice. Students are not provided details of the maze layout prior to the competition.	Competition, design optimization	1

Figure 4. Student-developed reconfigurable maze.

Educational Approach

As previously mentioned, all laboratory hardware, software, and educational materials were developed by student engineers. We have found this to be an effective method to develop new material while providing our students hands on engineering experience.

Concerning each of the advanced topics previously identified, the laboratory exercises provide a visual, hands on opportunity to work with each of these concepts.

Real Time Operating Systems: In the second operating systems development laboratory exercise, students are given the opportunity to actually develop, program, and test a real time operating system control algorithm. Students are required to write a foreground/background type operating system that uses polling mechanisms in background activities and interrupts for higher priority foreground activities. The scenario of having the robot navigate about a maze while avoiding landmines emphasizes the importance of dealing with higher priority activities in a timely manner.

Multiple Interrupts: As a follow on activity in the second operating system laboratory, students are required to develop a complete interrupt driven operating system. Here they must contend with issues relate to a multi-interrupt system including interrupt priority and interrupt latency.

Fuzzy Logic Concepts: In the fuzzy logic laboratory exercise, the students are required to "surgically remove" their previously completed interrupt driven control system and substitute a fuzzy logic based control system. If they have properly applied the concepts of structured design techniques, it will be an easy task for them to disengage their operating system from the support functions of the system and then install a completely different control system. The laboratory exercise also allows them to examine the trade-offs between the number of required sensor inputs, how to partition the inputs into linguistic variables, and the development of inference based rules to control robot motion. They then get to experience first hand the impact of the design decisions they have made as they observe their robot operation.

Structured Design Concepts: Structured design concepts are emphasized throughout the course. For each laboratory assignment students are required to develop a structure chart, a UML activity diagram, and fully documented code. Early in the semester students may see these activities as a tedious, unnecessary exercise. However, as the semester progresses the students are required to tie together some of the early developed subsystems into a complete operating system for the robot. They soon discover the importance of these techniques as they begin to modify and existing system. Furthermore, in many cases partially coded laboratory solutions with accompanying structure charts and UML activity diagrams are provided to the students as a starting point for the laboratory exercises. In these cases, the students experience first hand the importance of good documentation.

Results

Over the past year a team of students led by the primary author has developed the material for an embedded system design course. This course is intended as a second course in microcontrollers. With seed money ($3600) obtained from the Electrical and Computer Engineering Department, twelve of the low cost robot platforms have been fabricated in house. Additionally, all laboratory assignments, supporting solutions and documentation have been developed.

Summary and Conclusions

With considerable student engineer efforts we have developed a real time embedded systems course to teach advanced microcontroller concepts including: real time operating systems, multiple interrupts, fuzzy logic concepts, and structured design techniques. We have spent considerable time in attempt to enliven the instruction of these topics using motivational laboratory assignments based on the use of a low cost, educational robot.

All materials associated with this course are readily available for adoption at other institutions. Please visit the teaching materials website at:

http://wwweng.uwyo.edu/electrical/faculty/barrett/68hc12/

Also feel free to contact the primary author at steveb@uwyo.edu.

References

1. I. Verner, S. Waks, and E. Kolberg, “Upgrading Technology Towards the Status of a High School Matriculation Subject: A Case Study,” *Journal of Technology Education*, Volume 9, Number 1, Fall 1997.

2. E. Mar, “Mobile Autonomous Robot”, MSME Thesis, The Cooper Union for the Advancement of Science and Art, 1998.

3. “Trinity College Fire Fighting Home Robot Contest,” Trinity College, Hartford, CT, http://www.trincoll.edu/events/robot, 2001.

4. J. Mendelsohn, “Come On Baby, Unlight My Fire,” *IEEE Intelligent Systems Magazine*, pp. 5-6, 2001.

5. MIT Artificial Intelligence Laboratory, www.ai.mit.edu, 2001.

6. D. Pack, G. York, P. Neal, and S. Stefanov, “Constructing a Wall-Follower Robot for a Senior Design Project”, Proceedings of the 1996 American Society for Engineering Education Annual Symposium, Washington D.C., June 1996.

7. S.F. Barrett, D. Pack, C.H.G. Wright, S. Stefanov, P. Neal, A. Klayton, “Innovative, Student-Centered Educational Tools for the Computer Engineering Curriculum”, *American Society for Engineering Educators 1998 Annual Conference,* Seattle, WA, June 1998.

8. S.F. Barrett, D.J. Pack, G.W.P. York, P.J. Neal, R.D. Fogg, E. Doskocz, S.A. Stefanov, P.C. Neal, C.H.G. Wright, A.R. Klayton, “Student-centered Educational Tools for the Digital Systems Curriculum,” *Computers in Educational Journal*, American Society for Engineering Educators, Vol. IX, No. 1, Jan-Mar 1999.

9. G.H. Miller, Microcomputer Engineering, 3rd Edition, Englewood Cliffs, NJ: Prentice Hall Inc., 2001.

10. S.F. Barrett and D.J. Pack, Embedded Systems Design and Applications with the 68HC12 and HCS12, Englewood Cliffs, NJ: Prentice Hall Inc., 2005.

11. M. Podanoffsky, “Building a Real-Time Multitasking Executive,” Circuit Cellar Ink, The Computer Applications Journal, No. 27, June/July 1992, pp. 14-21.

12. J. Ganssle, “Writing a Real-Time Operating System - Part I A Multitasking Event Scheduler for the HD64180,” Jan/Feb 1989, Circuit Cellar Ink, pp. 45-51.

13. J. Ganssle, “Writing a Real-Time Operating System - Part II Memory Management and Applications for the HD64180,” Mar/Apr 1989, Circuit Cellar Ink, pp. 30-33.

14. P. Laplante, “Real-Time Systems Design and Analysis An Engineer's Handbook,” IEEE Computer Society Press, 1993.

15. G.H. Miller, “Microcomputer Engineering,” second edition, Pearson Education, 1998.

16. S. F. Barrett, D.J. Pack, C. Straley, L. Sircin, G. Janack, “Real-Time Operating Systems: A Visual Simulator,” *Computers in Education Journal*, April – June 2005 issue.

17. B. Kosko, Neural Networks and Fuzzy Systems – A Dynamical Systems Approach to Machine Intelligence, Englewood Cliffs, NJ: Prentice Hall Inc., 1992.

18. S.F. Barrett, A. Wells, C. Hernandez, T. Dibble, Y. Shi, T. Schei, J. Werbelow, J. Cupal, L. Sircin, G. Janack, “Undergraduate Engineers for Curriculum and Laboratory Equipment Development,” *Computers in Education Journal*, Vol. XIII, No. 4, 2003, 46-58.

19. Y. Shi, Master of Science in Electrical Engineering Thesis, “Autonomous Wall-Following Robot with Obstacle Avoidance,” University of Wyoming, August 2003.

20. T. Schei and J. Werbelow, Senior Design Project Report, “Wall Finding Robot,” University of Wyoming, Fall 2002.

21. M. Sardar, Master of Science in Electrical Engineering Project Report, “Microprocessor Based Robot Motion Control,” University of Wyoming, May 2004.

Biographical Information

Steven F. Barrett received the BS Electronic Engineering Technology from the University of Nebraska at Omaha in 1979, the M.E.E.E. from the University of Idaho at Moscow in 1986, and the Ph.D. from The University of Texas at Austin in 1993. He was formally an active duty faculty member with the United States Air Force Academy, Colorado and is now an Associate Professor of Electrical and Computer Engineering, University of Wyoming. He is a member of IEEE (senior) and Tau Beta Pi (chief faculty advisor). His research interests include digital and analog image processing, computer-assisted laser surgery, and embedded controller systems. He is a registered Professional Engineer in Wyoming and Colorado. He co-wrote with Dr. Daniel Pack “68HC12 Microprocessor: Theory and Application,” Prentice-Hall, 2002; “Embedded Systems Design and Applications with the 68HC12 and HS12,” Prentice-Hall, 2005; and “Microcontroller Fundamentals for Engineers and Scientists,” Morgan-Claypool Publishers, 2006. In 2004, Barrett was named “Wyoming Professor of the Year” by the Carnegie Foundation for the Advancement of Teaching. Email: steveb@uwyo.edu

Daniel J. Pack is a Professor in the Department of Electrical Engineering at the United States Air Force Academy, CO. He received the Bachelor of Science degree in Electrical Engineering in 1988, the Master of Science degree in Engineering Sciences in 1990, and the Ph.D. degree in Electrical Engineering in 1995 from Arizona State University, Harvard University, and Purdue University, respectively. He was a visiting scholar at Massachusetts Institute of Technology-Lincoln Laboratory. He co-authored two textbooks on microcontrollers and embedded systems and authored over 70 journal and conference papers. He is a member of Eta Kappa Nu, Tau Beta Pi (faculty advisor), IEEE (senior), and ASEE. He is a registered Professional Engineer in Colorado. In 2005, Pack was named "Colorado Professor of the Year" by the Carnegie Foundation for the Advancement of Teaching. His research interests include cooperative UAVs, intelligent control, automatic target recognition, and robotics. Email: daniel.pack@usafa.edu

Pamela Beavis received the Masters of Science in Electrical Engineering degree from the University of Wyoming in 2005. She obtained the BSEE in 2004. Pam is now employed as a systems engineer for Northrop Grumman in Denver, Colorado.

Mahbub Sardar completed his Masters of Science in Electrical Engineering degree at the University of Wyoming in May 2004. He is now employed as a Systems Engineer at NAVAIR in China Lake, California.

Austin Griffith is a Masters of Science in Electrical Engineering candidate at the University of Wyoming. He completed the BSEE in December 2004.

Michael Stephens is a BSEE candidate at the University of Wyoming.

Julie Sandberg is a BSEE candidate at the University of Wyoming.

William Lewis (Lew) Sircin was a Master Technician in the Electrical and Computer Engineering Department, University of Wyoming from 1976 to 2006. Lew tragically passed away in March 2006 after a lengthy illness. This article is dedicated to Lew and his work with hundreds of students at the University of Wyoming over the past 30 years.

George Janack is a Master Technician in the Electrical and Computer Engineering Department, University of Wyoming.

Yi Shi completed her Masters of Science in Electrical Engineering degree at the University of Wyoming in December 2003.