

Engaging Undergraduate Students with Robotic Design Projects

James O. Hamblen, *Senior Member, IEEE*, and Tyson S. Hall, *Student Member, IEEE*
School of ECE, Georgia Tech, Atlanta, GA 30332-0250
hamblen@ece.gatech.edu

Abstract

This paper describes our experiences developing robotics design projects for undergraduate students in our electrical and computer engineering curriculum at Georgia Tech. Several low-cost alternatives for developing robot-based design projects and designing the associated electronics and sensors to control them are included.

1. Introduction

The nature and background of undergraduate students in Electrical and Computer Engineering (ECE) programs has changed significantly in recent years. Maintaining high levels of student interest in undergraduate design projects can be difficult. Projects involving a robot still excite students and motivate them to work harder. Rather than purchasing a completed robot system, we have developed projects that provide students with a combination of interesting hardware and software design experiences.

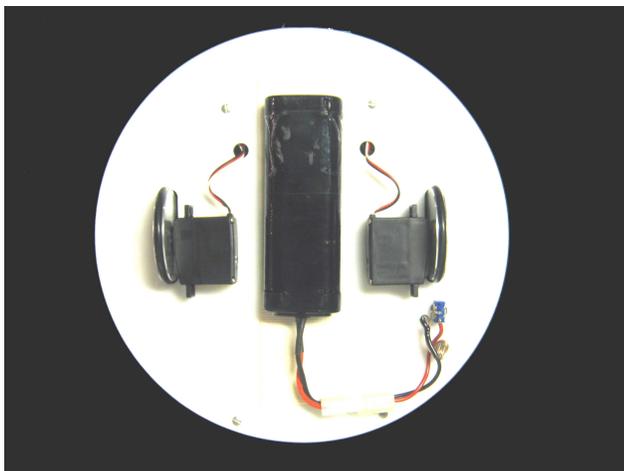


Fig. 1. The bottom view of a simple robot base built using two modified R/C servos with R/C car wheels, a round plastic disk, and a 7.2V R/C battery pack. An FPGA board mounted on the top of the plastic disk is used as a controller [1-3].

For the past two years in ECE 4006, our team-based senior design project course, we have had a wide variety of mobile robot projects. Students are allowed to select their own projects and teams subject to instructor approval. Many students with an interest in computers have selected robotics projects. The total cost for these various robot projects runs from two hundred to around two thousand dollars. This paper describes the different approaches that student design teams have taken.

2. Selecting a Robot Base

A robot base must be constructed or selected for the project. Electronics and sensors are then added to the base to produce an autonomous robot. In the past four years, we have developed a variety of low-cost robots that fall into several broad categories.

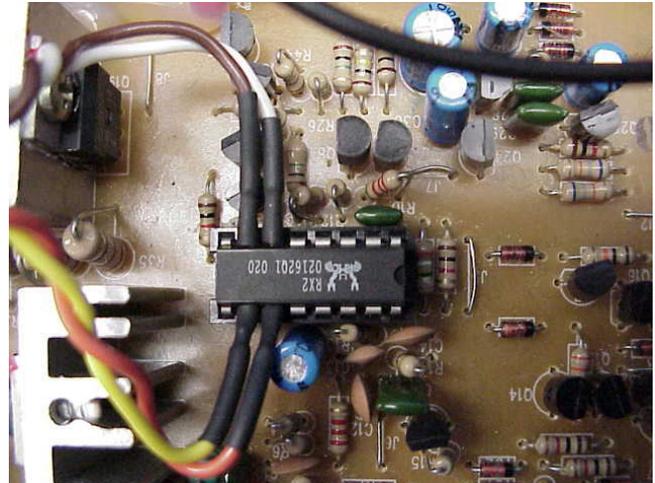


Fig. 2. Motor control signals in this R/C toy truck were tapped at the RF decoder chip in order to control the DC motors using the toy's existing motor drive circuits [3].

The first option is to mechanically design and construct your own robot base. Two modified radio controlled (R/C) servos with a plastic disk and a battery pack can be used to build a simple robot base that moves slowly on flat surfaces [1-3].

While more complex bases can be designed and

constructed by students, our experience has been that this approach takes too much time for a semester-long project. Designing and constructing a robot base does not leave enough time for electrical and computer engineering students to focus their primary efforts on the computer, electronics, sensors, hardware interfacing, and software control aspects of the robot. Other possible robot base options to consider include R/C toys, hobbyist R/C models, and commercial robot platforms.

2.1. Low-cost R/C Toys

R/C toys are perhaps the most economical alternative. These toys are designed to be controlled remotely with a low-cost radio transmitter that is included with the toy. R/C toys usually have only a simple “on” or “off” type control signal for each DC motor.

When R/C toy vehicles are disassembled and reverse engineered, one typically finds a circuit board with a 16 or 18-pin DIP RF decoder chip. Located somewhere on this chip are four output pins that control the two motors. These pins drive the inputs to an H-bridge circuit with four power transistors that drive and reverse the DC motors.

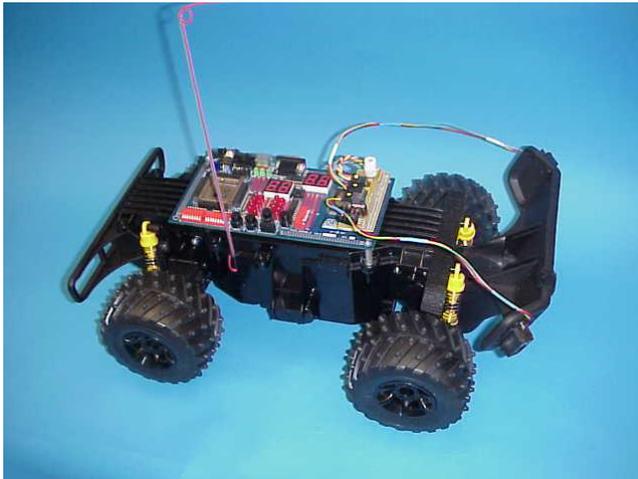


Fig. 3. This R/C toy truck was modified to create an autonomous robot by adding an FPGA board, IR sensors, and a compass. It has four-wheel drive and a suspension that is capable of moving on rough surfaces outdoors [3].

As seen in Fig. 2, tapping off the signals at this point allows reuse of the H-bridge drive circuits for motor control [3]. This type of modification was used to control the robots seen in Figs. 3, 4, and 5. Two bits are used to control each H-bridge circuit. The valid motor control states are off, forward, and reverse. There is a fourth-input state in the H-bridge circuit that must be avoided since it will short the supply voltage through the four power

transistors.

Even though the original toy only supports simple “on” or “off” motor control, it is possible to implement speed control by varying the duty cycle of the motor’s control signals using timing pulses in the 1K to 20K Hz range.



Fig. 4. This R/C toy boat was modified to create a robot boat. PIC microcontrollers, a GPS unit, an electronic compass, a gameboy camera, and a wireless modem are housed inside the cabin. Batteries serve as ballast inside the hull.

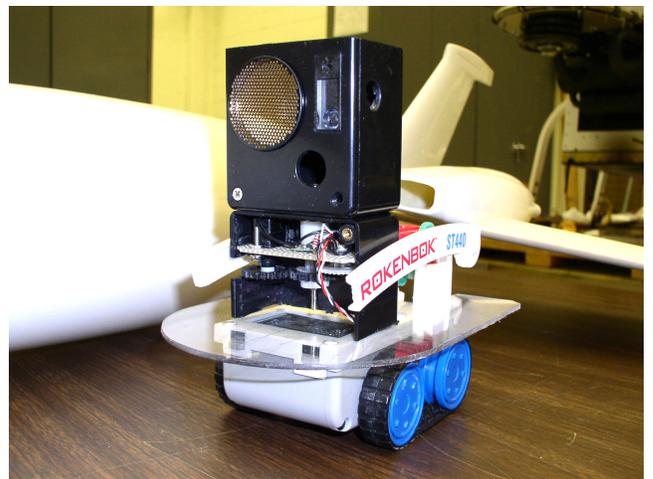


Fig. 5. This small R/C toy front loader was modified to create an autonomous robot by adding a custom FPGA-based SOPC board, and a rotating turret with IR and SONAR sensors.

Timing pulses for speed control can be implemented using computer-based timing routines or custom FPGA hardware in the robot’s control hardware. When reversing a running DC motor, a small time delay with the motor in the off state can also be inserted to minimize back EMF

and the associated power supply surges. Extra decoupling is typically required between motor and computer power supplies that use the same battery.

Speed control is a desirable feature since many of these toys can move faster than a student. On more than one occasion, we have had errant robots outrun pursuing students and crash into walls at full speed destroying hours of work. Larger toys require a larger open space to turn and maneuver. So far, we have not had a potentially dangerous flying robot aircraft project, but an R/C Blimp or the Skybot captive helicopter would be a safer alternative to consider [4,5].

2.2. Hobbyist R/C Models

Another alternative is to start with a hobbyist R/C model. R/C models are somewhat more expensive than toys and are typically sold without the radio, servos, or a speed control circuit. Instead of simple “on” and “off” control, hobbyist R/C models are designed to use radios with proportional control signals.

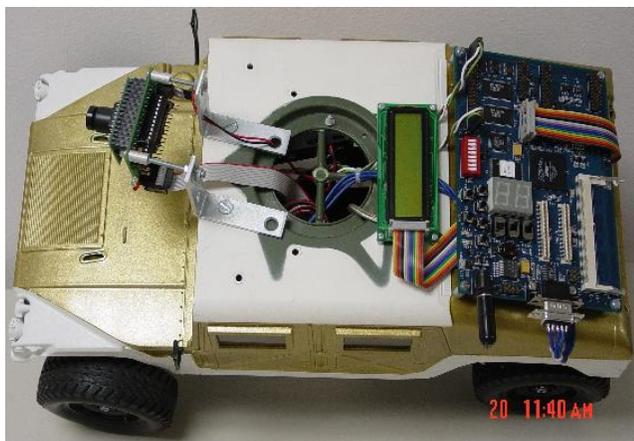


Fig. 6. This R/C hobbyist Hummer was converted to an autonomous robot with vision tracking capabilities using a SOPC board and a CMUCAM. Standard R/C PWM signals control the motors and steering.

Servos and the speed control units all operate using a standard R/C pulse-width-modulated (PWM) signal with 1 to 2ms. wide pulse sent every 20ms. The PWM signals can be generated using computer timer routines, a PWM chip, or with a counter in FPGA hardware. Since a standard 1-bit PWM signal is used for all motor controls and steering, interfacing requires fewer pins and a standard PWM timer setup can be used.

2.3. Small Low-Cost Commercial Robot Bases

Several low-cost commercial robots are also an option to consider. This is the most expensive option, but the

robots are also a bit more durable than R/C toys or models. They can be reused for several rounds of student projects.

The Amigobot seen in Fig. 7 contains a rechargeable battery, two accurate drive motors with encoder feedback, and eight SONAR units [6]. An internal microcontroller controls the motors and reads the sensors. The microcontroller accepts commands and transfers sensor data using a serial port connection on the top of the robot. The Amigobot was originally designed to be remotely controlled using a PC with a serial cable. A computer board can be mounted on top of the robot and the existing serial communication protocol can be used to create an autonomous robot.



Fig. 7. The small Amigobot commercial robot was originally designed to be remotely controlled using a PC with a serial cable. An FPGA-based SOPC board was added to convert it to an autonomous robot.

A new robot base option that has recently appeared is the ER1 seen in Fig. 8 [7]. Acknowledging the fact that a mass-produced notebook PC may have the best price/performance tradeoff for a robot controller and that many people already own a notebook PC, this robot base is designed to carry any notebook PC.

Optional IR sensors and cameras are available that connect to the notebook’s USB port. A USB motor control unit also has extra digital and analog I/O pins available. More advanced robotics software packages are also available for the ER1.

3. Adding Electronics to the Robot

Once a robot base has been selected, the next step is to add electronics and sensors to control the robot. Student projects have used controllers that fall into three main categories: microcontroller chips, commercial embedded

single board computers or notebook PCs, and FPGA boards with memory.



Fig. 8. The ER1 is a commercial robot base designed to carry a notebook PC.

3.1. Using a Microcontroller Chip

PIC microcontroller chips are covered in an elective class, ECE 4075, at Georgia Tech [8]. Students that have already taken this elective course are somewhat more inclined to use PIC microcontroller chips to control their robot.

PICs contain an 8-bit CPU, RAM, ROM, timers, serial ports, A/Ds, D/As, and I/O pins on a single chip. DIP-style packages also make it easier for students to design and construct their own printed circuit boards (PCBs). The PCB shown in Fig. 9 was designed by students and fabricated using a free PCB layout CAD tool and a commercial web-based PCB manufacturing service.

3.2. Using a commercial computer board or notebook PC

For robots that require more intelligence and processing power, commercial embedded computer boards or a notebook PCs offer a higher performance alternative. Embedded computer boards such as Arcom's Olympus board shown in Fig. 10 have roughly the same capabilities as a current generation PC [9]. This board contains an X86 processor with memory and I/O and it can run the standard PC operating systems. It is available with a CompactFlash card that boots Windows CE or XP.

Higher performance, however, comes at a price, and

this board requires quite a bit of battery power when compared to a small microcontroller. Other embedded boards are available between the two extremes with different power/performance tradeoffs.

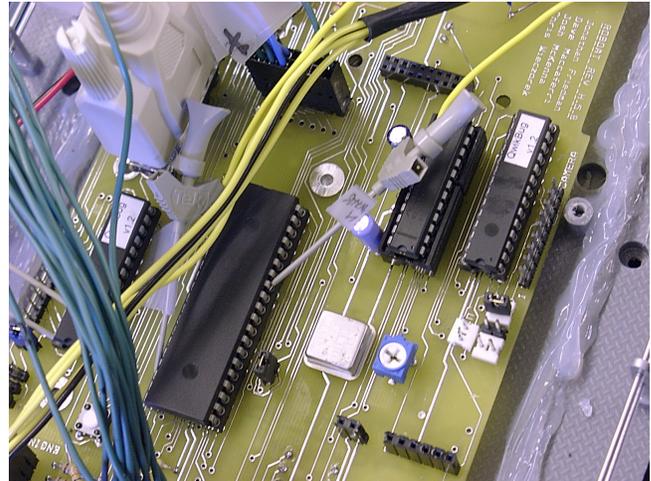


Fig. 9. A Student designed PCB containing three PIC microcontrollers, a GPS unit, an electronic compass, and a wireless modem from the boat in Fig 4.

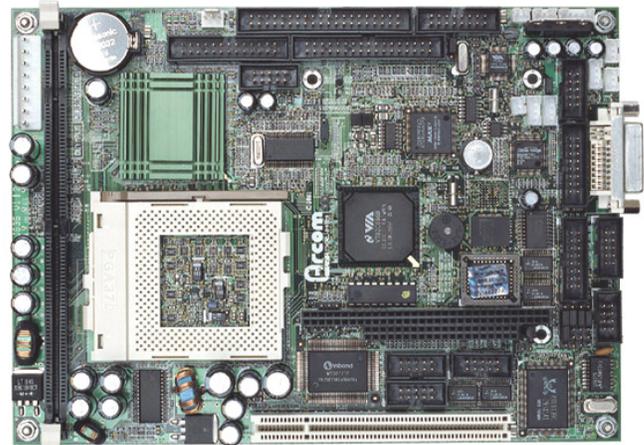


Fig. 10. This Arcom Olympus embedded computer board uses X86 PC chip technology and runs Windows CE or XP from Flash memory. Students are currently working on projects using it to control the Amigobot seen in Fig. 7.

Autonomous robots that require multitasking, image processing, wireless networking, and higher level planning need a more powerful processor than is available in a single chip microcontroller or a slower low-power RISC processor. At Georgia Tech, an elective course, ECE 4180, covers high-end embedded system design and provides students with an introduction to PC hardware

technology and Windows programming tools. Students that have taken this course have the skills necessary for Windows hardware and software development.

Another option to provide a higher performance computer for a robot is to use a commercial notebook PC to control the robot. Surprisingly, the mass production of notebook PCs makes the pricing quite attractive when compared to the lower volume embedded computer boards even though many of the features of the notebook PC are not needed. One minor drawback is that most notebook PCs have a larger form factor than the embedded computer boards.

Since I/O is somewhat limited on notebooks, another board may be required to interface to the sensors. One approach is to use a general purpose low-cost USB I/O board to interface to the sensors using the notebook's USB port [10,11].

3.3. Using an FPGA board

The third controller alternative is based on FPGAs. A small FPGA board can provide control for a simple robot. Serial ports, PWM, and other needed interfaces can be implemented using FPGA logic. Students with HDL and FPGA experience will find this the easiest way to generate new sensor and motor interface logic. A very basic robot can be controlled by a state machine implemented in the FPGA, but more complex robots need a computer.

The current generation of larger FPGAs makes it possible to synthesize an entire processor using the FPGA's logic elements. New commercial SOPC (System on a Programmable Chip) technology is available to support this approach [12-15]. Students using this approach gain the broadest exposure to a wide range of EDA CAD tools and gain additional insight in hardware/software tradeoffs. They select processor options, write control programs in C, and design and simulate interface hardware in VHDL or Verilog.

Previous course experience with FPGA CAD tools and using VHDL or Verilog for logic synthesis is needed. A SOPC processor core configuration tool allows users to customize their processor and I/O features. This tool outputs a VHDL or Verilog model of the processor core and the standard FPGA CAD tool flow is then used to synthesize and/or simulate the processor. A special C compiler is provided to write software applications for the processor core.

The processor core uses only a fraction of the FPGA's logic, and other hardware can be added to interface to the robot. As an example, students building the robot in Fig. 6 implemented PWM controllers in hardware using VHDL.

They also added additional parallel I/O ports to interface to the PWM controllers and LCD panel as well as serial ports for the CMUCAM interface.

In larger FPGAs, several processor cores can be placed in a single chip. SOPC boards such as the Nios board seen in Fig. 11 include a large FPGA with RAM and Flash memory to support the processor. The Flash memory and an on-board CPLD can program the FPGA and load the external program memory at boot time.



Fig. 11. Altera's Nios SOPC board contains a 200,000 gate FPGA with Flash and SRAM memory. It runs a RISC processor IP core. A C compiler is provided for the processor core. It controls the robots seen in Figs. 6 and 7.

With the SOPC approach, very small controller boards are possible, although surface mount devices and multilayer PC boards are required with the high pin counts and package types used in FPGAs.

Given the extreme flexibility of using the FPGA logic for new sensor and control interfaces it is easy to use a standard commercial SOPC board for a wide variety of projects [13-15].

4. Experience with Robot Sensors

A number of sensors are available for use in a small robot. IR distance sensors and SONAR units are commonly used in small low-cost robots [1-3]. However, older electronic fluorescent light ballasts operate in the same 38 to 42KHz frequency range as many IR sensors and will cause interference. In addition, IR and SONAR sensors will sometimes miss objects when directed at sharp corners. Combining information from several types of sensors or an array of sensors is a good way of improving reliability.

Some other interesting sensors we have used recently

include electronic compass sensors, GPS, and the CMUCAM. New generation electronic compass sensors are more tolerant of being tilted, have faster settling times, and can correct for nearby constant magnetic fields [16]. The magnet in a DC motor can cause interference, so an electronic compass should be mounted as far away from the motors as possible. An electronic compass is useful for dead reckoning navigation.

Low-cost GPS receivers and the more accurate DGPS units can be used to provide location information for a robot to within a few feet. Typically, GPS satellite signals cannot be received inside a large building, so they are only appropriate to consider for an outdoor robot.

The CMUCAM is an inexpensive easy way to add vision capabilities to a small robot. It contains a color CCD camera with a PIC microcontroller [17]. Images can be downloaded via a serial port and it can track color blobs and report their location. A low-cost USB video camera can also be used if the robot's control computer supports USB.

5. Conclusion

Robots demonstrate the capabilities of electronics technology and provide students with an interesting and challenging design project. It is possible to keep the cost of a robot project at levels that schools and students can afford by modifying R/C toys and models and by reusing available computer technology for the controller.

Using a high-end commercial embedded computer board can provide students with an introduction to PC hardware technology, OS API calls, and a valuable real-time programming experience. Whereas, using the SOPC approach to implement a robot controller provides students with a significant FPGA, HDL, CAD tool, and C programming experience.

Full details of the student robot projects described in this paper are available online at the ECE 4006 and 4180 class web sites [18, 19].

Acknowledgements

Microsoft's Embedded Systems Grant Program, the Steve W. Chaddick Endowment, and donations from Intel, Altera, and Xilinx have helped students obtain the hardware and software needed for these projects.

References

- [1] K. Newman, J. O. Hamblen, and T. Hall, "An Introductory Digital Design Course Using a Low-cost Autonomous Robot," *IEEE Trans. Educ.*, vol. 37, no. 2, pp. 194-202, May 2002.
- [2] J. O. Hamblen. "Rapid prototyping using field-programmable logic devices," *IEEE Micro*, vol. 20, no. 3, pp. 29-37, May/June 2000.
- [3] J. O. Hamblen and M. D. Furman. *Rapid Prototyping of Digital Systems*, Kluwer Academic Publishers, 254 pages, August 1999.
- [4] R. Klenke, J. Tucker, and J. Blevins, "A New Hardware/Software Codesign Environment and Senior Capstone Design project for Computer Engineering", *Microelectronic Systems Education Conference*, pp. 66-67, June 2003.
- [5] Draganfly Inc., *R/C Toy Catalog* [Online]. Available: <http://www.rctoys.com>.
- [6] Active Media Robotics, *Amigobot Catalog* [Online]. Available: <http://www.amigobot.com>.
- [7] Evolution Robotics, *ERI Personal Robot Datasheet* [Online]. Available: <http://www.evolution.com/erl>.
- [8] J. Peatman, *Design with PIC Microcontrollers*, Prentice Hall, 1998.
- [9] Arcom, *Embedded Single Board Computer Catalog* [Online]. Available: http://www.arcom.com/#Emb_boards.
- [10] Devasys Inc., *USB I2C/IO Datasheet* [Online]. Available: <http://www.devasys.com/usb2icio.htm>.
- [11] Eagle Technology, *USB DAQ Product Catalog* [Online]. Available: <http://www.eagledaq.com>.
- [12] J. Hamblen, "Using an FPGA-Based SOC Approach for Senior Design Projects", *Microelectronic Systems Education Conference*, pp. 18-19, June 2003.
- [13] T. S. Hall and J. O. Hamblen, "System-on-a-Programmable-Chip Development Platforms in the Classroom," to appear in *IEEE Transactions on Education*, 2004.
- [14] *Nios Embedded Processor User's Guide*, PDF File, Altera Corporation, [Online]. Available: <http://www.altera.com/products/devices/nios/>, Jan. 2002.
- [15] *MicroBlaze Hardware Reference Guide*, PDF File, Xilinx Corporation, [Online]. Available: http://www.xilinx.com/ipcenter/processor_central/microblaze/, April 2002.
- [16] PNI Corporation, *Compass Sensor Catalog*, [Online]. Available: <http://www.precisionnav.com/legacy/compass-sensors-radar-sensors-breath-sensors-and-oem-modules/sensors-modules.html#sensormodules>
- [17] Seattle Robotics, *CMUCAM Datasheet*, [Online]. Available: <http://www.seattlerobotics.com/cmucam.htm>.
- [18] J. Hamblen, *ECE 4006 Sr. Design Project Web site*, [Online]. Available: <http://www.ece.gatech.edu/~hamblen/4006/projects>.
- [19] J. Hamblen, *ECE 4180 Embedded Systems Design Web site*, [Online]. Available: <http://www.ece.gatech.edu/~hamblen/4180>.