An Autonomous Arduino-based Racecar for First-Year Engineering Technology Students

Dr. Warren Rosen, Drexel University (Eng.)

Dr. Warren Rosen received his Ph.D. in physics from Temple University. Following graduation he served as assistant professor of physics at Colby and Vassar Colleges where he carried out research in optical physics, solar physics, and medical physics. During this time he was a visiting scientist at Kitt Peak National Observatory and a research scientist at Atmospheric and Environmental Research in Cambridge, MA. He then moved to the Naval Air Warfare Center, Aircraft Division in Warminster, PA where he established an optical communications laboratory for development and characterization of optical components, systems, and protocols for high-performance avionics data networks. Dr. Rosen is currently an assistant clinical professor at Drexel University, where he is responsible for developing and teaching courses in microprocessors, microcontrollers, FPGAs, and optics. Dr. Rosen has carried out research sponsored by the National Security Agency, National Science Foundation, the National Oceanic and Atmospheric Administration, DARPA, the Office of Naval Research, Air Force Office of Scientific Research, and the Missile Defense Agency. Dr. Rosen is the author or coauthor of over 80 publications and conference proceedings and the holder of six U.S. patents in computer networking and signal processing.

Dr. Yalcin Ertekin, Drexel University (Engineering Tech)

Yalcin Ertekin received his Ph.D. degree in mechanical Engineering from Missouri University of Science and Technology (formerly The University of Missouri-Rolla). He is a Certified Quality Engineer (CQE) and Certified Manufacturing Engineer (CMfgE). His teaching responsibilities include Computer Numerical Control, manufacturing processes, applied quality control, mechanical design, and applied mechanics, manufacturing information management systems, introduction to technology and graphical communication as well as senior design courses. He developed two online graduate courses: rapid prototyping and product design and lean manufacturing principles for MSET program. Dr. Ertekin has over six years of industrial experience related to quality and design engineering mostly in automotive industry. He worked for Toyota Motor Corporation as a quality assurance engineer for two years and lived in Toyota City, Japan. His area of expertise is in CAD/CAM, manufacturing processes, machine design with CAE methods, rapid prototyping, CNC machining and quality control. His research interest includes sensor based condition monitoring of machining processes, machine tool accuracy characterization and enhancement, non-invasive surgical tool design and bio-materials applications. During his career, Dr. Ertekin published papers in referred journals and in conference proceedings in his area of research interest. He has also been PI for various NSF research projects including NSF-TUES and MRI programs. Dr. Ertekin is an active member in the Society of Manufacturing Engineers (SME), and currently serves as a chair of Philadelphia SME Chapter-15.

Mr. M. Eric Carr, Drexel University

Mr. Eric Carr is a full-time Laboratory Manager and part-time adjunct instructor with Drexel University’s Engineering Technology program. Eric assists faculty members with the development and implementation of various Engineering Technology courses. A graduate of Old Dominion University’s Computer Engineering Technology program and Drexel’s College of Engineering, Eric enjoys finding innovative ways to use microcontrollers and other technologies to enhance Drexel’s Engineering Technology course offerings. Eric is currently pursuing a Ph.D in Computer Engineering at Drexel, and is an author of several technical papers in the field of Engineering Technology Education.
An Autonomous Arduino-based Racecar
for First-Year Engineering Technology Students

Abstract

This paper describes an autonomous Arduino-based racecar project for use in a first-year engineering technology course. The project is intended to stimulate students’ interest in developing key skills such as computer programming and industrial design and fabrication as well as providing a hands-on introduction to several key topics in electrical, mechanical, and industrial engineering technology. The racecar comprises an Arduino Uno, an Adafruit Motor/Stepper/Servo Shield, three Ultrasonic Ranging Modules, assorted motors and gears, and a chassis that is custom designed by the students and fabricated on a 3-D printer. The ultrasonic ranging module is used by the microcontroller to navigate a simple obstacle course and the groups compete for best time. Expected outcomes include the ability of the students to explain basic computer organization, to write simple code in C, and to design, analyze, and rapidly fabricate useful mechanical components.

Introduction

Robotics offers a useful tool for communicating the excitement of engineering to first-year engineering technology students. This paper describes an autonomous Arduino-based racecar project for use in a first-year engineering technology course. The project is intended to stimulate students’ interest in developing key skills such as computer programming and industrial design and fabrication as well as providing a hands-on introduction to several key topics in electrical, mechanical, and industrial engineering technology.

The racecar comprises an Arduino Uno microcontroller board, an Adafruit motor control shield, ultrasonic ranging modules, assorted motors and gears, and a chassis that is custom designed by the students and fabricated on a 3-D printer. The ultrasonic ranging modules are used by the microcontroller to navigate a simple rectangular racecourse and the groups compete for best time. The total cost for purchased parts is about $60 per racecar.

Project outcomes include the ability of the students to explain basic computer organization, to write simple code in C, and to design, analyze, and rapidly fabricate useful mechanical components. Multiple forms of assessment are used to demonstrate success, including student surveys, lab reports, presentations, and homework.

In the following sections we describe the development of the racecar, including the design and fabrication of the chassis housing the mechanical and electrical components and the development of the microcontroller subsystem. We then describe the racetrack used to race the students’ cars. Finally, we discuss the results obtained with a group of first-year engineering technology students.
Project Description

At Drexel University the Introduction to Engineering Technology course is taken by all first-year ET students, so the class is made up of a mix of students interested in electrical, mechanical, industrial, or biomedical engineering, as well as students interested in dual concentrations. This makes the racecar project an ideal platform since it comprises elements of electrical, mechanical and industrial engineering.

The project comprises a four-week series of lectures and lab exercises culminating in the development and racing of the completed racecar. All the students receive introductory lectures in basic concepts in mechanical design and fabrication and computer organization and programming. Then the class is divided into teams of three or four students. One or more students in each group are tasked with the design and fabrication of the racecar chassis and the other students are assigned the development of the software and integration with the hardware.

The four-week sequence begins with a one-week introduction to basic concepts in mechanical design and fabrication. The students learn to use SolidWorks mechanical design software to generate digital models of the frame the racecar chassis that will house the electro-mechanical components. They are then given practice exercises in fabricating simple shapes using a Fused Deposition Modeler that is available in the Engineering Technology Mechanical Laboratory. In designing the chassis the students must evaluate the tradeoff between light weight and the ability to withstand static/dynamic loading due to moving parts.

In the second week, the students are introduced to basic concepts in computer organization and programming. They then learn the basics of programming in C and write short programs in C that can be downloaded to the Arduino microcontroller. As a first laboratory exercise they are also shown how to breadboard simple circuits containing Light Emitting Diodes (LEDs). They then write short programs to flash the LEDs in sequence.

For the remaining two weeks the students are divided into groups as described above. The student(s) responsible for chassis development design the basic components needed to mount the drive motors, wheels, sensors, and microcontroller. They then integrate them into a prototype chassis design, which is then fabricated. If necessary, the process is iterated.

The student(s) responsible for software development perform two additional laboratory exercises. In the first, they are given prewritten code to poll the rangefinder sensors. They use this to calibrate the sensor results for various distances. In the second exercise, they are given prewritten code to control the DC drive motors using the motor shield. They are then assigned to write their own code to make the racecar drive in a straight line to within 300 mm (about one foot) of an obstruction and then turn 90° to the left. They must then write the complete code needed for the racecar to navigate the racecourse autonomously. Since the chassis are still under development for most of this time, the software groups are given Magician Chassis available from Sparkfun to practice with. Once the student-designed chassis was available, the drive motors and wheels from the Magician Chassis were transferred to it.
Once the software and chassis developments were completed they were integrated and the race held in the final week. The students then documented their technical issues, approaches to solutions and results in a final report and formal presentation.

Racecar Chassis Development and Prototyping

One of the main goals of the project is to simulate a real-life engineering development experience for our freshmen students who will work in teams to develop an autonomous racecar and use Acrylonitrile butadiene styrene (ABS) plastic prototypes using a rapid prototyping machine. This project takes a simple idea of designing a racecar and improves its technology so that it can be run autonomously. Starting with the basic components of manufacturing materials, prototyping, programming, and building basic systems, this project takes engineering to the next level. It incorporates these components, but uses more complex systems in the final product. Instead of “hand-making” materials using manual machine tools, students used CAD software (SolidWorks®, Google SketchUp, etc…) to design the racecar chassis.

The main CAD design software used by the students was SolidWorks®. The students can access the software using Windows Remote Desktop since the software is installed in a virtual lab server. This is very convenient for the students making the tool available anytime and anywhere via an Internet connection.

Student teams initially develop a sketch of basic components (Figure 1a). After initial sketches, CAD software was used to design and improve the chassis, which is the backbone of the racecar. Students had to pay attention how the parts are fit together, obtaining a lightweight design and proper placement of the other components and sensors on the chassis. Due to time limitations, part assembly with CAD modeling was not part of the introductory lectures.

Students save their CAD models in STL format; these are converted into the 3D printer’s native format (cmb file format), using CatalystEX® 3D printing software (Figure 2a). CatalystEX® slices the STL file into layers (layer thickness .010”). Part models can be easily manipulated for best orientation and scaled if they are oversized. The 3D printer – a Stratasys uPrint3D housed in the Engineering Technology’s Mechanical Laboratory – has a work space 6” x 6” x 8” in size. (Figure 2b).

The 3D Printer is able to print out these materials based on what is designed in the CAD software. This approach works well, showing students how software can be used to create the different components of a product. The Laboratory Manager is responsible for the monitoring of prototyping process for successful completion. This very well simulates rapid prototyping assets owned by a technological company in several locations worldwide. By directing prototyping efforts to locations with available machine time, under-utilized prototyping assets are used more efficiently, prototyping productivity can be tremendously improved. In an actual industrial setting, rapid prototyping machines are supervised by operator technicians. After a finishing and cleaning process to remove support materials, the prototype models are obtained (Figure 2c). In the initial class, students discovered several design flaws after initial prototyped models as follows:
Design in general, and mechanical design in particular, is an iterative activity. Iteration, looping back to explore alternatives, is an essential part of the design process. Iterating is, of course, time consuming and costly, but a systematic search for creative solutions early on in the process is not, either extremely costly or terribly time consuming and is fundamental to the design of innovative products. Therefore, in this project, students learn what engineers have to deal with and how to look at a problem and take the steps to fix it.

Figure 1. (a) Student design sketches for various components (b) CAD Design using Google SketchUp (c) CAD Design using SolidWorks®, (d) Improved CAD Design using SolidWorks®.
Microcontroller Control System

The racecar control system is based on an Arduino Uno microcontroller board, shown in Figure 3 (left). To provide motor control, a mating Adafruit Motor/Stepper/Servo Shield\(^1\) is used. This shield provides four H-bridge DC motor controllers, two of which are used to control the speed of the two drive motors bidirectionally. It also offers servo and stepper motor controls. The former would be useful if it was decided to replace the DC motors with more accurate servos. The Arduino board is programmed via USB, which also provides limited power. Both boards can also be powered by a battery connected to the motor shield when the car is running.

Due to the limited amount of time available for the project, three prewritten subroutines are provided to the students for use in their race cars:

\[\text{setLeftSpeed()}, \text{setRightSpeed()}, \text{and getDistance()}.\]
setLeftSpeed() and setRightSpeed() set the speed of the left and right motors, respectively. Since
the race cars have an omnidirectional caster in the front, steering is accomplished by varying the
relative speed (and/or direction) of the left and right motors. Turning the left motor forwards
more quickly than the right, for example, will result in a slight right turn. Both setLeftSpeed()
and setRightSpeed() accept type double as input, which is then clipped to between -100 and 100.
Negative numbers mean the motor turns backwards; positive numbers mean forwards. Zero is
used to stop the motor. The appropriate I2C commands are then created by the subroutines to
send to the motor control shield, which handles the PWM generation. setLeftSpeed() and
setRightSpeed() return C type void. (That is, they provide no return value.)

getDistance() takes two integer arguments: the trigger and echo pins to be used. These are
defined as constants at the top of the provided stationery, so the subroutine call ends up being
fairly straightforward. For example:

    getDistance(leftTriggerPin, leftEchoPin).

getDistance() then returns a value of type double, representing roughly the distance in cm to the
detected object.

As an example, the code below uses the three subroutines. It commands 80% forward speed on
both motors until the front distance is less than 500 cm, at which time it commands a left turn for
300ms and then continues.

Because of the nature of the Arduino loop() function, no outer loop structure is needed: the code
will continue to execute until power is removed.

    void loop(){
        //Go forward until object less than 500cm away
        while(getDistance(frontTriggerPin, frontEchoPin) >= 500){
            setLeftSpeed(80);
            setRightSpeed(80);
            delay(50);     //a short delay, for stability
        }

        //Turn left for 300ms, then continue
        setLeftSpeed(-80);
        setRightSpeed(80);
        delay(300);
    }

During the introductory lecture and second laboratory exercise the students are shown how to use
simple if-else statement and several useful relational operators. These, together with the
prewritten routines above, were sufficient for the students to program the racecar to navigate the
course.
Racecourse

The racecourse used in the project is shown in Figure 4. The walls are constructed from 8’ x 4’ x ¼” plywood sheets cut into 1-foot-wide strips screwed together at the corners. The exterior of the course measures 8’ on a side and the interior course is 2’ wide. The 8’ sides are cut in half for easy storage.

The race consists of one lap around the course, and is run one car at a time; the team with the lowest time wins. Each car starts its race using a sequence employed in a conventional drag race (see Figure 5a). The driver begins by switching the car electronics “on” and is then told to “stage” the car by breaking a laser beam that crosses the course at the checkerboard starting line shown in Figure 5a. This turns on the top row of yellow lights on the drag race “Christmas Tree,” which may be seen on the right in Figure 5a. The next step in the starting sequence is that the next three rows of yellow lights turn on (with a warning beep) from top to bottom with a ½ second delay between each. Finally ½ second later a row of green light turn on to indicate the start, at which time the driver releases the car, which begins to run autonomously. If the driver releases the car early, a row of red lights at the bottom of the tree turns on and the car is disqualified for that race. The race ends when the car traverses the course and again breaks the laser beam. Elapsed time is shown on a monitor in real time as the car runs the course (see Figure 5b). Each car has a chance to run several trials and the car with the lowest individual time is the winner.

![Racecourse](image-url)

Figure 4. Racecourse used for the project.
Figure 5. a) Start of a race. The Christmas tree, which provides the starting sequence, can be seen on the right. b) Race timer showing elapsed time.

Results

The project was used in the fall of 2013 in an introductory engineering technology course. Seven students participated; these were a mix of students who planned either a concentration in electrical engineering technology, mechanical engineering technology, or a dual concentration. Prior to starting the project one of the students had some minimal experience with programming and the others had no experience. None of the students had any experience with computer-aided design application such as SolidWorks or Google SketchUp.

Figure 6 shows one of the completed racecars. The forward and side rangefinders can be seen to the left of the large wheel, and the Arduino Uno/Adafruit motor shield can be seen at the far right of the car. The microcontroller board sits on top of a Vex 7.2 V battery. The drive motors are located below the chassis. The car is 16 cm long, 12.5 cm wide, and 10 cm high at the top of the motor shield. The car can complete one lap of the track in less than 30 seconds.

Figure 6. Racecar completed by one of the student groups. The car measures 16 cm long, 12.5 cm wide, and 10 cm high at the top of the motor shield.
Each of the two groups was able to program their car to complete the course at least once, but neither car could do so reliably due to several problems that increased development time. First, the distance obtained from the rangefinder was not always repeatable. Second, it was discovered that the two motors that drive each wheel did not respond in the same way to identical control signals. Also, both groups had to modify and refabricate their chassis design at least once.

The students were often quite creative in attempting to solve these problems. For example, one group discovered and implemented proportional control. Over time the students made continuous progress toward solutions and we believe that more uniform results would have been achieved if more time were available.

The students’ performance on homework, lab reports, and their final presentations indicated that they had achieved a substantial understanding of the rapid prototyping process and programming in C, particularly in view of the short time available.

At the end of the project students were surveyed using a Likert-type scale to assess how much the course contributed to their understanding of these topics. The rubrics used were 1 = “None” and 5 = “A great deal”. The questions are shown in Table 1, together with average scores and standard deviations for each. Five of the seven students responded, with an average score of 3.7 ($\sigma = 0.8$) for questions 1 through 9. The score for Question 10 (How much did you enjoy this part of the course?) was substantially lower than scores for other questions. This is due to a low score from a single student who strongly disliked programming. If that student’s score is removed the average increases to 3.5 ($\sigma = 0.87$), which is consistent with scores for other questions. In contrast, one student reported to his advisor that, although he thought he would hate programming he was now considering a career in a related field.

<table>
<thead>
<tr>
<th>Question:</th>
<th>Average</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How much did this project contribute to your understanding of how microprocessors/microcontrollers work?</td>
<td>4.00</td>
<td>0.63</td>
</tr>
<tr>
<td>2. How much did this project contribute to your understanding of computer programming?</td>
<td>3.40</td>
<td>0.49</td>
</tr>
<tr>
<td>3. How much did this project contribute to your understanding of autonomous systems?</td>
<td>3.40</td>
<td>0.49</td>
</tr>
<tr>
<td>4. How much did this project contribute to your understanding of microcontroller control systems?</td>
<td>3.20</td>
<td>0.75</td>
</tr>
<tr>
<td>5. How much did this project contribute to your general understanding of how various rapid prototyping processes work?</td>
<td>3.20</td>
<td>0.75</td>
</tr>
<tr>
<td>6. How much did this project contribute to your understanding of how FDM (Fused Deposition Modeling) rapid prototyping process works?</td>
<td>3.00</td>
<td>0.89</td>
</tr>
<tr>
<td>7. How much did this project contribute to your understanding of how SolidWorks CAD modeling is done?</td>
<td>4.40</td>
<td>0.80</td>
</tr>
<tr>
<td>8. How much did this project contribute to your understanding of how design, prototyping, re-design/improve cycle works?</td>
<td>4.40</td>
<td>0.49</td>
</tr>
<tr>
<td>9. How much did this project contribute to your ability to work in teams?</td>
<td>4.00</td>
<td>0.00</td>
</tr>
<tr>
<td>10. How much did you enjoy this part of the course?</td>
<td>3.00</td>
<td>1.26</td>
</tr>
</tbody>
</table>

Table 1. Student survey questions and results.
Conclusions

In this paper we described an autonomous racecar project for use in a first-year engineering technology course. We use this project to stimulate students' interest in developing key skills such as computer programming and industrial design and fabrication. It also provides a hands-on introduction to several key topics in electrical, mechanical, and industrial engineering technology.

The significance of the methodology used in this project is to combine theory and practice to prepare the students to become better problem solvers and obtain practical solutions to real life/simulated problems using a project-based approach. The project also introduces students to the advantages of rapid prototyping, in particular the fact that objects can be formed with any geometric complexity or intricacy, reducing the construction of complex objects to a manageable, straightforward, and relatively fast process. Students are able to build parts that need to be assembled into more complex shapes, and therefore they experience concepts such as design for manufacturing and assembly.

While the project was designed for a class with mixed majors (electrical, mechanical, etc.) it can easily be adapted to a purely electrical engineering technology program by providing the students with a pre-built car and allowing them to program the microcontroller to complete the racecourse in minimum time.

Bibliography

4. Private communication, Mr. Albert Frattarola, Director of Global Engineering & Technology, Southco Inc.