Using Arduino Microcontroller Based Robot Projects to Teach Mechatronics in a Hands-On Mechanical Engineering Curriculum

Andrew Siefert, Jonathan Hoy, Keith Christman,
Dr. Kevin R. Anderson, P.E.
California State Polytechnic University at Pomona
Mechanical Engineering Department
Mechatronics and Robotics Laboratory
3801 West Temple Ave, Bldg. 17-2353
Pomona, CA 91768 USA

Abstract

The use of ARDUINO microprocessors allows for a very top level approach to teaching Mechatronics. The focus on this paper is to motivate the use of ARDUINO microcontrollers to teach Mechatronics and Control Systems in Engineering Education. This is not a research paper per say, rather it is a detailed explanation of an example of "hands-on" pedagogy. The goal of this paper is to merely share the outcomes of an experiential learning environment with the academic community. This paper will present the results of using ARDUINO microcontroller based projects to teach a senior level Mechanical Engineering Mechatronics/Robotics course. The use of "hands-on" learning is documented in this paper via the teaching of Mechatronics through the capacity of having students design/fabricate/engineer/program robot to navigate a maze autonomously. Use of Mechatronics, i.e. a synergy of control systems, data acquisition and sensors, kinematics of machinery, and programming is detailed in this paper using student projects as the learning/instructional vehicle. Rubrics for assessment of such a "hands-on" course will also be shared in this paper.

Introduction

Mechatronics is defined by Bolton[1] as "Electronic Control Systems in Mechanical and Electrical Engineering". We have been teaching a senior level technical elective as ME 499/L "Mechatronics/Lab" at California State Polytechnic University at Pomona (Cal Poly Pomona) in the Department of Mechanical Engineering for over the last 15 years. Over this timeframe as technology has advanced, so has the demand to teach the course at a cost effective level for the students. Faced with economic downturns, the university has forced its faculty to come up with creative alternatives to teaching such state-of-the-art technical electives as robotics in order to fill the need of making our graduates competitive as they enter the industrial workforce and/or graduate school. To this end, over the last couple of years, the faculty teaching robotics and control systems curriculum in the Mechanical Engineering Department at Cal Poly Pomona have devised a series of projects based around using the ARDUINO series of microcontrollers. This paper will describe the use of Arduino Microcontrollers to teach Mechatronics address the following ASEE-PSW 2013 objectives: multidisciplinary - interdisciplinary projects/classes, experiential learning, project-based learning and innovative pedagogies and uses of current and emerging technologies in the classroom.

The Arduino based Mechatronics course taught as ME 499/L at Cal Poly Pomona utilizes a course project and competition. The equipment used by each student varies, but a baseline kit provided by the Robot Shop[1] was used for commonality. This project based course is similar to others used for educational purposes including, Anderson and Jones[2]. Other noteworthy studies which use a similar pedagogy can be found in the literature including the works of Consi[3] and Wang *et al*.[4] The current paper discusses similar pedagogy and how it is implemented via the "hands-on" learning philosophy which Cal Poly Pomona is well known for.

Robot Equipment Used

For the projects, a basic kit was selected. This was done to keep the cost to a minimum. The equipment used is listed below. Many of the components listed are referenced in the compendium reference books given by Waren *et al*.[5], and Mccomb[6]. Additional on-line resource which prove valuable when teach Mechatronics include the Robot Shop[7] at http://www.robotshop.com/ and Trossen Robotics[8] at http://www.trossenrobotics.com/c/arduino-robotics.aspx. The equipment used to build the robot was as follows:

- DFRobotShop Rover V2 Basic Kit - $94.99 (www.robotshop.com)
- DFRobotShop Rover Arduino Board (Arduino Uno R3 + Motor Controller)
- Tamiya Track and Wheel Kit
- Tamiya Dual Motor Drivetrain Kit
- Mounting Brackets
- Sharp Short-Range Infrared Proximity Sensor (x3) - $13.95 ea. (www.sparkfun.com)
- ADJD-S311-CR999 Color Sensor + Breakout Board - $14.95 (www.sparkfun.com)
- 2-Terminal Microswitch (x2) - $1.50 ea. (www.sparkfun.com)
- PerfBoard Sheet - $3.50 (Radio Shack)
- Miscellaneous Mounting Hardware/Electronic Components
- Overall cost approximately $200

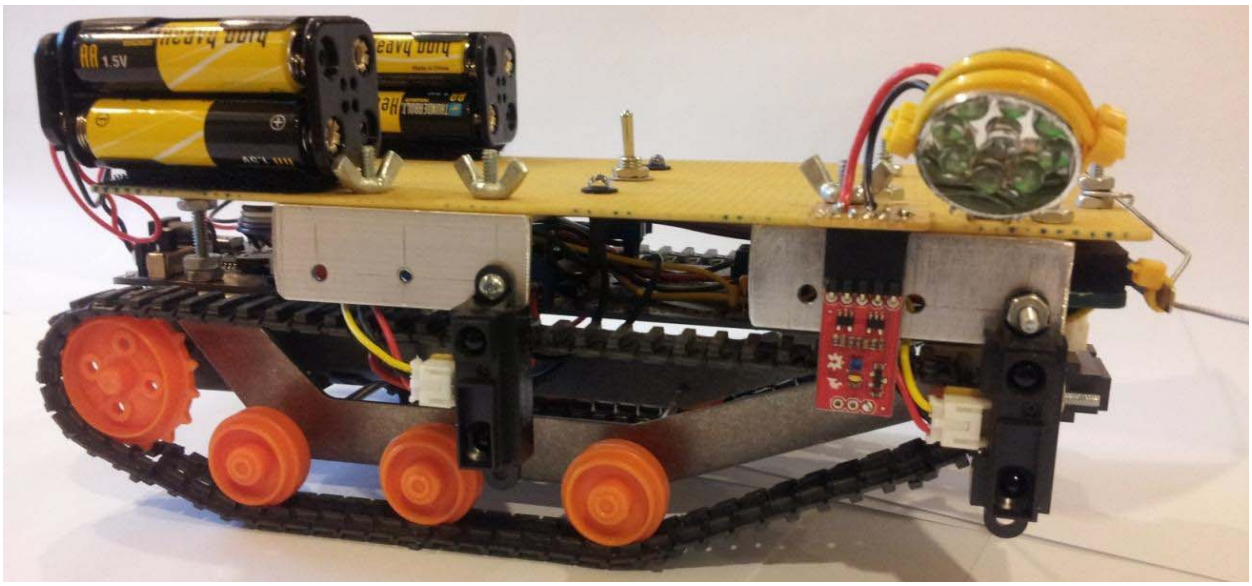

Figure 1- Completed Maze Robot used in Mechatronics ME 499

A novice knowledge of the C programming language is required to be up and running when working with the ARDUINO. A valuable reference to this end is given by Margolis[9].

Course Learning Objectives

The learning objectives of the ME 499 course were

1. Learn how to design, build, program and test an autonomous maze-solving robot
2. Learn how to condition, process, and calibrate sensor data for use in autonomous control logic
3. Learn how to troubleshoot and tune control logic for optimum operation

This robot was designed to find the end of a conventional maze by following a single wall until it finds a red flag. To accomplish this, the robot uses three analog infrared distance sensors and one digital color sensor. The robot is based on the DFRobotShop Rover tracked robot kit. The kit consists of an expanded Arduino board with built-in motor control, an aluminum frame, a set of wheels and tracks, two motors and a dual-drive gearbox as shown in Figure 2. This kit was employed with minimal hardware modifications besides sensor mounting hardware and circuitry added to the built-in prototyping area.

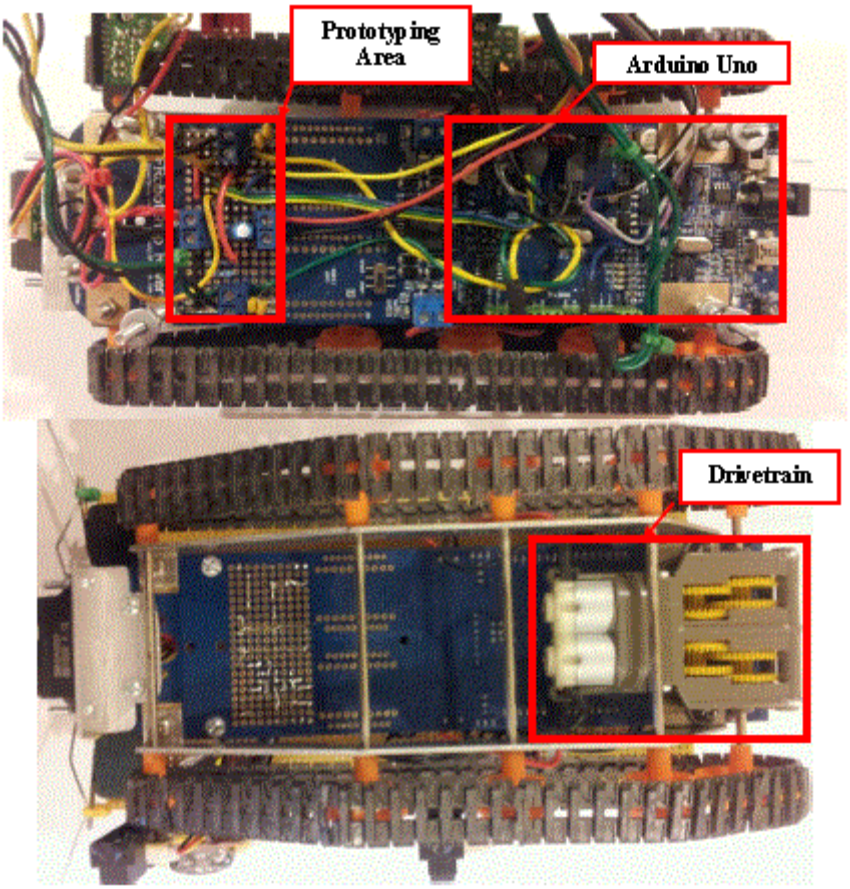

Figure 2- The Main Board on the Finished Robot (Top & Bottom views, respectively)

Anatomy of the Arduino

The Arduino Uno is a microcontroller board based on the ATmega328 (High Performance, Low Power AVR® 8-Bit Microcontroller, Advanced RISC Architecture). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. The Arduino contains everything needed to support the microcontroller; one simply connects it to a computer with a USB cable or powers it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 programmed as a USB-to-serial converter. The ATmega328 has 32 KB. It also has 2 KB of SRAM and 1 KB of EEPROM Revision 3 of the board has the following new features; 1.0 pin out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. Other relevant technical specification of the Arduino include the following:

- Microcontroller=ATmega328
- Operating Voltage=5V
- Input Voltage =7-12V
- Input Voltage (limits)= 6-20V
- Digital I/O Pins=14 (of which 6 provide PWM output)
- Analog Input Pins=6
- DC Current per I/O Pin=40 mA
- DC Current for 3.3V Pin=50 mA
- Flash Memory=32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM=2 KB (ATmega328)
- EEPROM= 1 KB (ATmega328)
- Clock Speed=16 MHz

Sensors and Control

The infrared sensors are the robot's primary method of sensing the maze geometry. One sensor is mounted on the front and two on the side of the robot which faces the maze wall (Figure 3). The one on the front serves to detect walls and other obstacles directly in the robot's path so it can negotiate around them. The sensors on the side detect the wall being followed. The side sensors are aligned vertically so they may better detect sudden changes in wall range caused by a corner.
The sensor at the front corner of the robot is used to judge the error between the robot's actual distance from the wall and the desired distance.

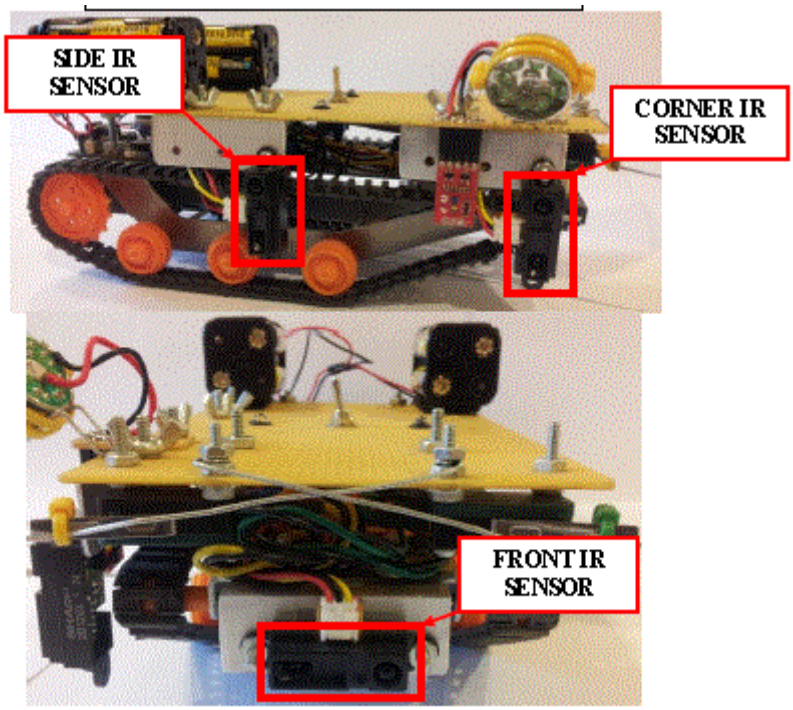

Figure 3- IR Sensors

This error value is passed to a PID algorithm controlling individual motor speed on each side of the robot which attempts to keep the robot at a set distance from the wall. The PID influences both motors simultaneously, always decreasing the speed on one and increasing the speed on the other. The motors have a nominal PWM duty cycle of 200, and the PID value adds to or subtracts from it. Note that the PID is limited to a maximum value of 55 to avoid it influencing one motor more than another. A diagram of the PID control algorithm can be seen in Figure 4.

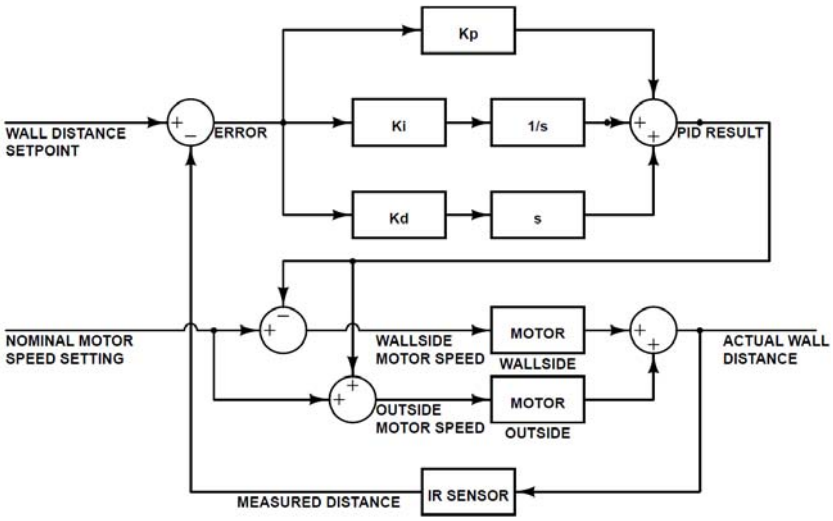

Figure 4 – Control System Block Diagram

The other IR sensor on the side of the robot simply aids in turning by using range measurements to determine inner track speed. To remove noise from the output signal of the sensors, a 100 μF capacitor was placed across the supply voltage to filter out Arduino supply noise. The sensor outputs ere also passed through 100 Hz passive low pass filters wired into the prototyping area of the main board. Since the voltage output of the IR sensors varies non-linearly with object range [1], values were taken from each sensor at ranges from 1.5" to 8.0" in 0.5" increments. These values were used to generate calibration curves for each sensor. The two sensors on the side of the robot yielded nearly identical curves and could therefore use the same calibration equation in the program. The front sensor, however, required its own calibration equation in order to obtain accurate results. The calibration curves can be seen in Figure 5.

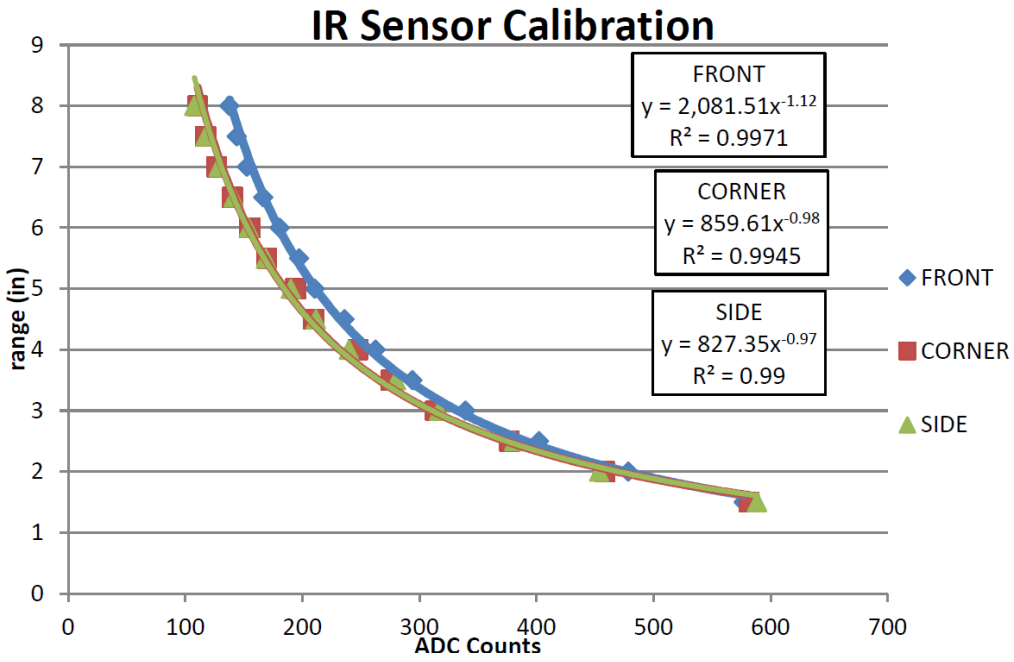

Figure 5– IR Sensor Response Calibration Curve

The robot also has two bump switches on the front of it for collision detection. The switches are connected to bumpers made from bent paper clips to augment their sensing area as shown in Figure 6. These switches are each wired to pins on the Arduino such that when a switch is tripped, the robot will reverse, turn away from the side that detected contact, and resume normal operation. In the early stages of development, switch bounce caused problems with detection. This problem was fixed by two 0.1 µF capacitors in parallel across the terminals of each switch, which served to filter out oscillations from switch bounce.

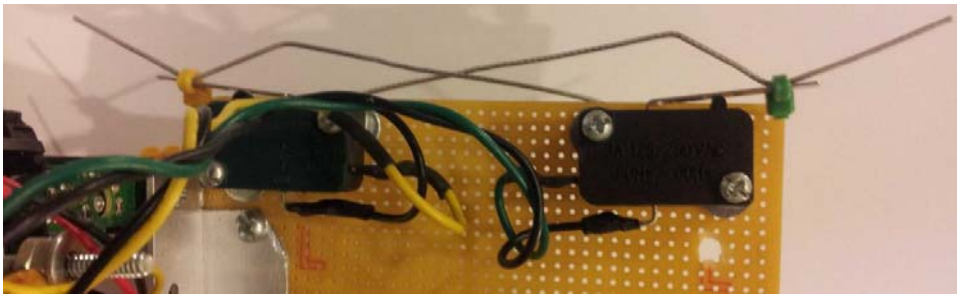

Figure 6– Bump Switches

The robot also has a color sensor to detect the maze start and end flags as shown in Figure 7. This sensor is digital; it has a series of data registers which can be read and written to over an I2C serial bus. Much of the code handling the I2C communication was taken from the open-source example code from SparkFun.com[3]. To obtain a color, the data registers for Red, Green, Blue, and Clear color values must be read. When the Arduino program starts, it initializes the sensor and calibrates it to a white surface. The program then stores values for the start (green) and stop (red) flags. During operation, the program continually polls the color registers and compares them to the stored values

for red and green. If it sees green, it instructs the robot to move. If it sees red, it instructs the robot to stop and idle. Note that the breakout board for this sensor has a built-in surface mount white LED illustrated in Figure 8. This LED is more than bright enough to allow for color sensing, but is mounted so close to the sensor itself that it drives all color registers to their maximum values if the surface being evaluated is more than 2 mm away from the sensor. Since this robot need to evaluate color at a distance of 2 inches, it was necessary to instead use a separate array of 9 white LEDs in a reflective housing to provide enough light to the surface without washing out the measurement. The LED array was taken from a broken flashlight and simply wired to a digital pin on the Arduino through a 30 Ω resistor to limit the pin current to 30 mA.

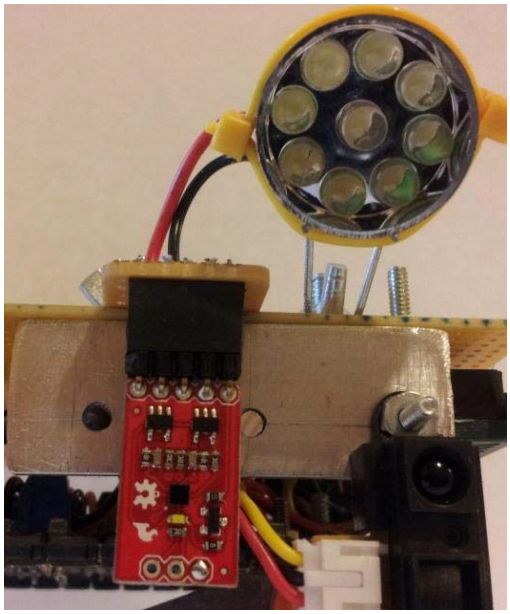

Figure 7– Color Sensing Unit

While building the project, it became immediately apparent the final robot should be able to follow either the left or right wall, as one may be preferable to the other depending on how the maze is configured. Normally, this would mean doubling the number of sensors on the robot, which would be prohibitively expensive and require a second microprocessor. Instead, the robot was designed and built such that all the sensors which detect the maze wall could be mounted quickly and easily on either side of the robot. A switch was also added to the top of the robot to allow the user to select which wall was to be followed, as well as two LEDs to inform the user as to which wall was currently selected as shown below in Figure 8. The typical budget for a project of this nature is on the order of approximately $200 per robot.

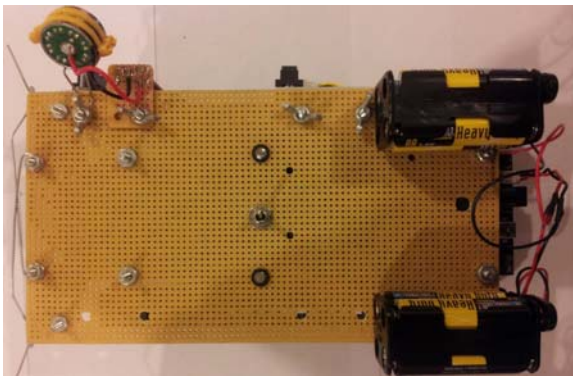

Figure 9– Finished Robot - Top View

When navigating the competition maze, in order to tell the start from the finish lines, a green sign (3 inches by 3 inches)was posted near the start of the course, and a red sign (3 inches by 3 inches) was posted on a wall of the maze marking the finish. Sensors were employed to perform bit map color scheme recognition, so that the robot would know green versus red, and therefore would be able to make a decision to start or finish. This fidelity is shown in Figure 10 and Figure 11, which are electrical schematic and flowcharts for the robot.

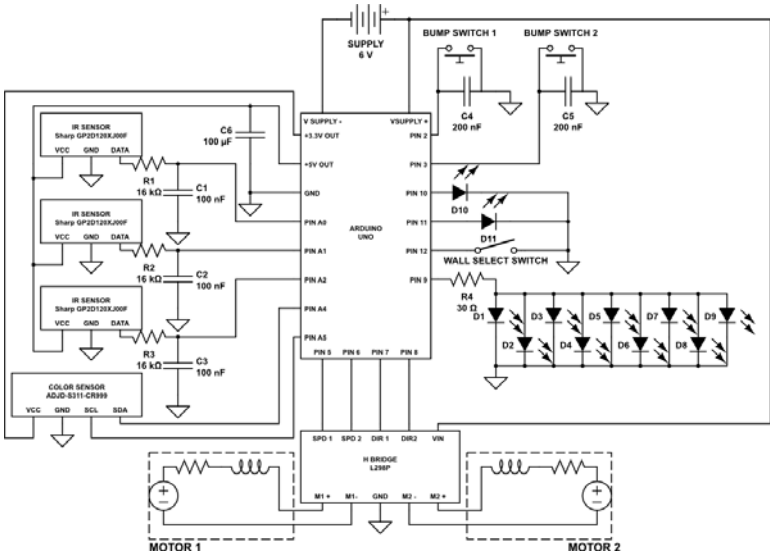

Figure 10– Robot Electrical Schematic

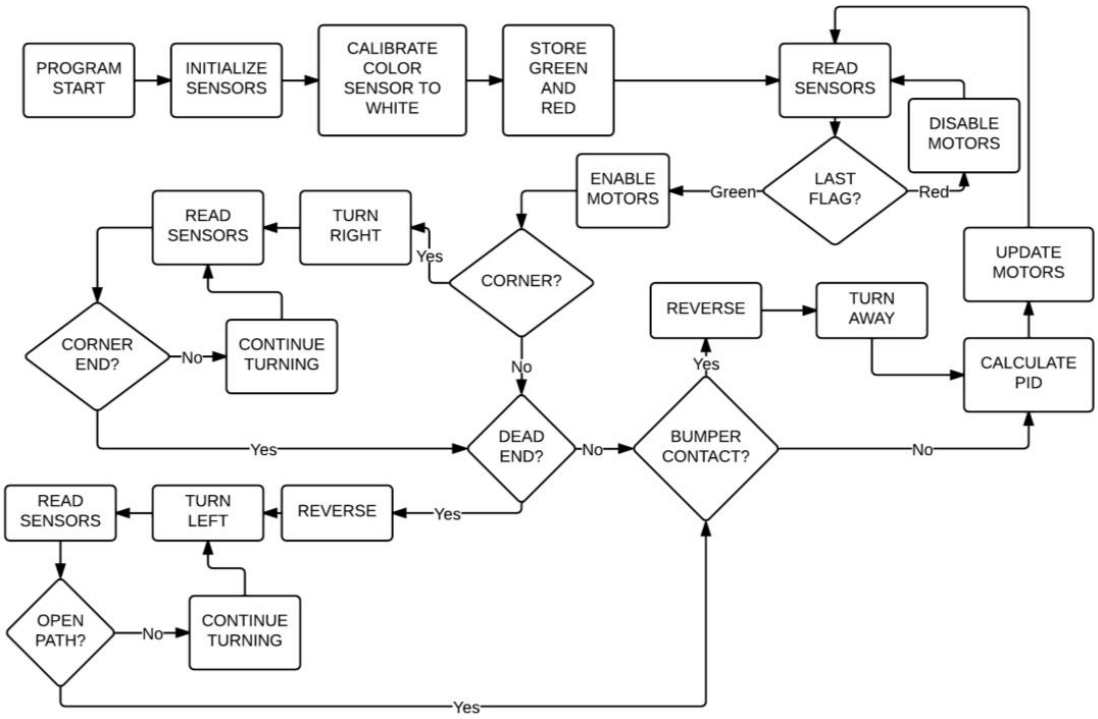

Figure 11– Robot Programming Flow Chart

Course Assessment

In order to assess the performance of each student enrolled in the ME 499/L "Mechatronics" course, a competition was held at the end of the quarter. The competition featured a maze which the robots had to navigate autonomously. Figure 13 shows the maze and two students preparing their robots to navigate the walls of the maze.

Figure 13– ME 499 Mechatronics Competition Maze

The maze is fabricated out of plywood and 2 inch by 4 inch stud walls, all painted white. The maze has a footprint o 8 ft by 8 ft, with a hinge built down the middle seam in order to fold it up

for storage and transport to robot competitions held abroad. The 2 inch by 4 inch stud walls can be removed and placed in various positions, thus the layout of the maze can be changed on-the-fly in order to exercise the true autonomous capability of the ARDUINO robots. White paint was selected, since the Mechatronics lab at Cal Poly Pomona has a large supply of fluorescent lighting, which may adversely affect the performance of photo-sensors when used with the robots. Figure 14 shows a robot in the maze as it makes a maneuver around a corner.

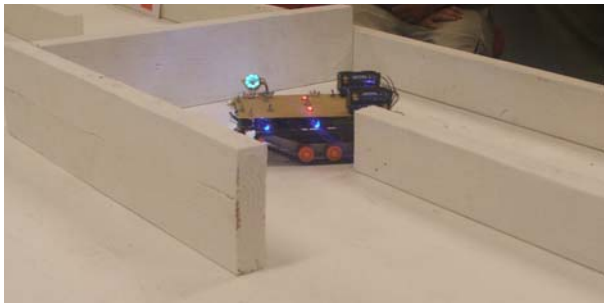

Figure 14– Robot Navigating the Competition Maze

Figure 15 shows a student preparing a robot to start the maze navigation.

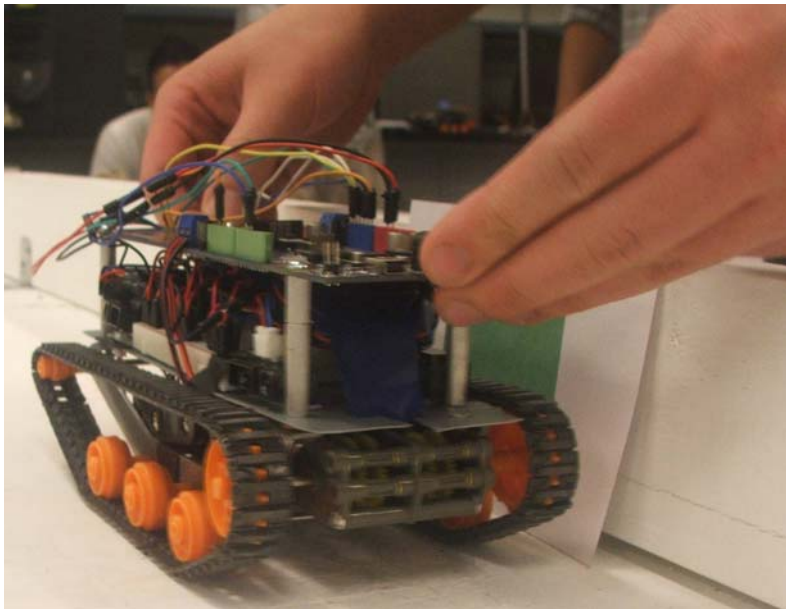

Figure 15– Robot Navigating the Competition Maze

The winner of the competition navigated the maze 35 seconds. Other speeds place second at 47 seconds, third at just over 2 minutes, and some robots did not complete the maze as required. The majority of student's robots finished the maze, however, 25% of the students did not have functioning robots at the end of the 10 week quarter. This course was taught with an industry based model, with schedule and cost the main drivers. The primary deliverables were a functioning robot and a technical write-up. To be successful, students needed to balance their academic workload, part-time jobs, family life and other things in order to make the robot come to fruition. Students engaged

in such projects should be aware that several hours of assembly, programming, de-bugging, etc. is paramount the success of such a project based course. In order to have a successful outcome, Engineering Educators need to be aware of these time commitments and cost restrictions when offering courses dealing with Mechatronics and/or Robotics. The rubric shown below in Figure 16 was used to assess the student's performance on the final paper write-up of the project.

ME 499 PROJECT REPORT & MAZE COMPETITION GRADING RUBRIC FALL 2012
STUDENT NAME:_____

DIFFICULTY/COMPLEXITY/MATURITY OF THE ROBOT DESIGN	/20
ROBOT WORKS/KINDA WORKS/DOESN'T WORK	/15
USE OF CONTROL ALGORITHM	/15
TECHNICAL REPORT WRITING STYLE	/20
REPORT FORMATTING, I.E. FIGURES AND REFERENCES, ABSTRACT, ETC.	/30
TOTAL	/100

Figure 16– Grading Rubric for Robot White Paper Technical Report

The overall course grade awarded to the students in the ME 499 "Mechatronics" class consisted of a series of homework sets, comprising 30% of the grade, the robot project/competition which was 35% of the grade, and the final report which was 35% of the grade. These statistics are limited to the students who actually enrolled in the class and completed the maze. This is because of the industry based deadline oriented fashion in which the course was managed, i.e. deadlines such as successful completion of the maze were emphasized from the onset of the course. This approach was adopted in mind that students entering the work force and/or graduate school must be cognizant of deadlines and/or budget constraints when working on real-world projects, and they must also recognize that there are penalties when deadlines are missed. This type of instruction helps to address the life-long learning skills need to acquire in order to be successful in private industry and/or graduate school.

Summary

This paper has summarized the use of ARDUINO based projects in teaching a senior level Mechatronics course. The robot project selected was an autonomous navigation of a maze. This project posed a great challenge in terms of navigation, object avoidance, and sensing. Thus, it made for a synergy of subject typically found under the umbrella of a Mechatronics course project. A simple array of infrared sensors and bump switches used in robot outlined in this paper proved to be a viable solution to this project and was proven sufficient for navigating a maze by following a wall. Getting the color sensor to read correctly under all lighting conditions was difficult; ultimately, it required clever programming by the students and a dedicated array of bright white LEDs. It is worth mentioning, however, that this system is severely limited by its need to stay close to the boundaries of the maze. It can only traverse walls which are connected to one another, and therefore cannot reach walls or barriers which stand freely on their own ("islands"). If the end flag for the maze is placed on an island, this robot, in its current configuration, will never get there. Solving this problem requires camera imaging with object recognition, which requires far more computing power than an Arduino alone can deliver. Therefore, any significant improvement upon this robot would require either a much more powerful onboard computer or wireless communication with a laptop or other

high level computing device, neither of which could be implemented within the project budget. As it is, this robot can solve most conventional mazes with little trouble, and the process of its design and construction has been challenging and insightful. Assessment of the robot was culminating in the form of a maze competition.

Bibliography

1. Bolton, W., "Mechatronics", 4th Ed. Pearson, 2008, London.
2. Anderson, K., Jones, C. Autonomous Robotic Vehicle Design. *Proc. of the ASEE PSW Regional Conference, April 7-8, 2005, Loyola Marymount University.*
3. Consi, T., A Versatile Platform for Teaching Mechatronics. *Proc. of ASEE Annual Conference 2012.*
4. Wang, Y., Ault, C., Nakra, T., Salgian, A., Stone, M. The Outcomes of an Interdisciplinary Undergraduate Course Involving Engineering, Science and Arts, *Proc. of ASEE Annual Conference and Exposition 2011.*
5. Warren, J., Adams, J., and Molle, H. "Arduino Robotics", Springer Science and Media, New York, NY. 2011.
6. McComb, G. "Robot Builder's Bonanza," 4th. Ed., Mc-Graw-Hill, New York, NY, 2011.
7. The Robot Shop, http://www.robotshop.com/ last accessed 1/28/13
8. Trossen Robotics, http://www.trossenrobotics.com/c/arduino-robotics.aspx last accessed 1/28/13
9. Margolis, M. "Arduino Cookbook", O'Reily Media, Inc., Sebastopool, CA, 2011.