Development of a Fundamentals of Electrical and Computing Systems course for in-service K-12 Teachers.

Prof. Kundan Nepal, University of St. Thomas

Kundan Nepal is currently an Assistant Professor in the School of Engineering at the University of St.Thomas (MN). His research interests span the areas of reliable nanoscale digital systems, mobile robotics and recongurable computing

Mr. Andrew Tubesing, University of St. Thomas

Andrew Tubesing is Laboratory Manager for the Electrical Engineering program at University of St Thomas in St. Paul, MN. He also serves on the faculty of the UST Center for Pre-Collegiate Engineering Education. Andrew has taught university courses in circuits, electronics, and engineering design for more than a decade. Prior to his academic career, Andrew spent 12 years as an engineer in the broadcast and telecommunications fields. Andrew holds a BA from St. Olaf College and a MS in Electrical Engineering from New Mexico Institute of Mining and Technology.

Development of Fundamentals of Electrical and Computing Systems course for in-service K-12 Teachers.

Abstract

The Science, Technology and Mathematics portion of STEM have been well covered in K-12 education for a number of years. With the adoption of Engineering into the Science Standards in the state of Minnesota, the number of schools that actively offer engineering either as an institute or embedded throughout their K-12 science, math and arts curriculum is increasing[1]. To better equip in-service and pre-service teachers to handle the needs of the engineering curriculum, the University of St. Thomas offers a Graduate Certificate in Engineering Education through its Center for Engineering Education. As a part of the certificate, in-service teachers go through a series of courses that expose them to the fundamentals of the engineering discipline, the engineering design process as well as a thorough review of the curriculum and ways of effectively integrating engineering into their existing courses. Teachers are also required to take one technical elective. These technical electives are typically offered as a part of the "Summer Academy". In 2013, two electives were offered - one in the area of energy and the other in the fundamentals of the electrical and computer engineering disciplines. This paper presents an overview of the "Fundamentals of Electrical and Computing Systems" course, the topics covered, feedback received and some reflections based on the first offering of the course. A total of seven in-service middle and high school teachers took the course during the first offering of the course in summer 2013.

Background

At the University of St. Thomas, the engineering education programs for in-service teachers are offered through its Center for Engineering Education – a partnership between the schools of engineering and education[2]. The programs available for P-12 teachers include:

a) customized professional development,
b) a 12-credit graduate certificate in engineering education for in-service teachers, and

The graduate certificate program is designed for in-service P-12 educators who want hands-on experience in engineering. The certificate program is a rigorous introduction to the engineering content with emphasis placed on the application of the course material in P-12 classrooms. The program helps the teachers develop the knowledge, skills, and competency necessary to effectively implement integrated engineering education in their classroom. The overall goal of this certificate program is to contribute to the engineering skill and efficacy of partner teachers, and in turn, the students with whom they work.

To complete the certificate, P-12 educators are required to take three core courses (total 9 credits) – *Fundamentals of Engineering for Educators*, *Engineering Design* and *Engineering in the P-12 classroom*. A brief course description for the three core courses is provided below:

Core 1: Fundamentals of Engineering for Educators: This practical course provides a broad engineering experience and includes the following topics: electronics, machine design, manufacturing engineering, computer programming, thermodynamics, statics, fluids and mechanics of materials. The class includes hands-on activities, links to academic standards (including Next Generation Science Standards) and discussion of the current/historical importance of the topics.

Core 2: Engineering Design: The focus of this class is on the engineering design process for P-12 educators. Design projects, hands-on computer labs, lectures and field trips will introduce students to how the engineering design process is applied in a variety of fields. Students will learn how to create engineering drawings, apply an engineering design process, use computer-aided-design (CAD) technology, and work with rapid prototyping tools. Strategies for incorporating engineering design projects into the P-12 classroom will be discussed.

Core 3: Engineering in the P-12 classroom: This course provides an overview of current P-12 engineering, exploration of pedagogy and content, links to academic standards, and assessment of classroom initiatives. Educators will learn of programs, methods and other educators who have introduced engineering into P-12 classes across several disciplines. Engineering resources for teachers will be presented. Educators will create a unit or module focused on a hand-on engineering activity for P-12 students in their licensure area.

After completing these three core courses, students are familiar with the fundamentals of engineering, the concept of engineering design and have explored the integration of engineering into their P-12 classrooms. The fourth course requirement of the certificate program is a technical elective in an area of their interest. This course titled *Fundamentals of Electrical and Computing Systems*, is designed to serve as a technical elective for teachers enrolled in the certificate program. The course description, the learning objectives of the course and the course contents are described in the sections below.

Course Description and Learning objectives

The goal of this course is not to train or produce electrical/computer engineers, but to introduce in-service K-12 science and math teachers to the vocabulary, language and concepts of electrical and computer systems. This technical elective builds on the experience and concepts learned by the students in the three core courses related to the fundamentals of the engineering discipline and the engineering design process. In designing this course, our motivation was to help the K-12 educators understand and integrate electrical/computer engineering topics in their science and math curricula. Each unit explored in this course were thus selected also to make the learning interactive, hands-on and economical so that the students can directly find a way to integrate the topics in their respective courses without having to overcome steep financial barriers.

The catalog description for the course is provided below:

> *"This course explores the fundamentals of both computer and electrical engineering disciplines and focuses primarily on the fundamentals principles that have enabled the "digital computing revolution" in*

communications, computing, entertainment, and monitoring systems. Educators will learn about the architecture of a microprocessor, interfacing it with external electronic circuitry/sensors and communicating with the processor through a computer programming language. The course consists of lectures, demonstrations, discussions, numerous in-class exercises and a course project."

The learning objectives for the course were specified as:

Upon successful completion of the course, the student will have:

1. *Gained factual knowledge (terminology, classifications, methods, trends) of electrical and computer systems.*
2. *Learned fundamental principles, generalizations, or theories related to circuits and computer systems and developed an ability to apply those principles to the design and analysis of practical circuits as demonstrated in laboratory experiments and student design projects.*
3. *Developed specific skills and competencies related to the use of software tools and lab instruments to design, simulate, build, test and document embedded systems using the Arduino platform.*
4. *Learned how to find and use resources for answering questions and solving problems in regards to the course project.*

Course Content

This course was designed to provide teachers with the fundamentals of electrical engineering and computing systems. The course introduced students to both the electrical hardware as well as software for interfacing with the hardware. In this section, we briefly describe the topics covered in the major units of the course. Each major unit typically consisted of a lecture session, discussion and lab exercises.

Unit 1: What can the *iPhone* teach us about the Electrical Engineering discipline? (1hr)

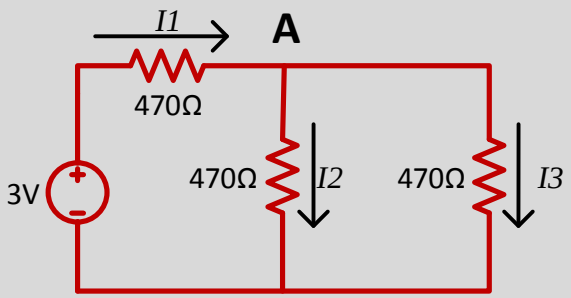
This unit was a survey of the Electrical Engineering discipline. The unit started with the discussion of how electrical engineering has changed our lives and moved into the discussion of the different sub-areas within the vast area of electrical engineering and *new* areas that are off-shoots of the electrical engineering discipline[3]. The unit introduced the *iPhone* and what is inside the beautiful looking case to discuss the interaction between and integration of the different electrical sub-disciplines. By looking at the iPhone students were able to quickly recognize the *computing system* (i.e. the microprocessor) and the *power system (*i.e. the battery) as well as components associated with *communication systems* (i.e. the antenna). Using different components and subsystems from the iPhone, over the course of an hour, students were also introduced to the Digital and Analog Electronics systems, Signal Processing, Electromagnetics and Control Systems as different sub-areas of the electrical engineering discipline.

Unit 2: Fundamentals of Electrical Systems (3hrs)

This unit provided students with the basic circuit terminologies such as charge, current, voltage, electrical power, ground and AC/DC voltage sources followed by introduction to circuit components such as resistors, capacitors, inductors, wires, switches, light emitting diodes etc. Students built circuits consisting of components in series and parallel and measured voltage and current using a multi-meter. Students were also introduced to simple circuit analysis techniques (Ohm's Law, Kirchhoff's Voltage/Current Laws, Voltage Divider, and Nodal Analysis). Students built circuits and compared the voltage and current readings from the multi-meter to the predicted results from circuit analysis techniques. In all cases students found that the measured results deviated from the calculated results. This discrepancy led to exploration of statistical variation in component values and measurement errors. Here is a summary of one of the exercises:

Circuit Analysis Exercise:

For the following circuit:



- Using your favorite circuit analysis technique (Kirchhoff's Current/Voltage Laws, Ohm's Law, Voltage Divider, Current Divider etc) calculate the voltage at Node A.
- Calculate the voltage across and currents I1, I2 and I3 through each of the three resistors.
- Build the circuit.
- Using the multi-meter measure the currents and the voltages.
- Do the values match with the calculated values? Why?

Unit 3: Fundamentals of Digital Logic and Electronics – How do digital systems work? (8hrs)

This unit introduced students to the basic concepts of digital systems—from mathematical building blocks to electronic logic circuits. It started with the binary numbering system, illustrating how this concept of ones and zeros becomes the basis for interpreting and communicating information digitally. Logic concepts of AND, OR, and NOT were explored conceptually at first, then by using logic symbols, Boolean algebra, and truth tables. Theoretical study concluded with system reduction using Boolean algebra and Karnaugh maps. Students

practiced these concepts with exercises focusing on these individual concepts and bringing them together.

The concept of logic functions and the systems that implement them provided the bridge between theory and real world applications. Students solved word problems by designing a combinational logic system to interpret a set of inputs and generate an output dependent on the conditions specified in each problem. Students then learned about the electronic components that perform logic operations and built circuits to implement their designs using 74XX series integrated circuits. Additional time was spent evaluating the need and means for evaluating and optimizing such systems by various parameters such as speed (propagation delay), power efficiency (number of chips/gates used), cost reduction (quantity of components), and complexity of wiring/construction. The unit concluded with discussion of classroom teaching methods for this subject and a variety of options for equipping a classroom/lab to do so. Here is a sample design problem:

Digital Design Exercise:

An auto company needs a digital system that requires certain safety parameters be met before the car will be allowed to start. You must design this safety check system as described below. Perform and document the following steps:

- Assign variables and define their values
- Construct a truth table for the "OK-to-start" conditions.
- Generate a reduced function using a K-map
- Draw a logic diagram and fully label it as a schematic
- Build and test the circuit.

The system has four sensors: An airbag status sensor sends a ready signal if the airbag is functioning properly and ready to deploy if needed. The driver's seat belt sensor indicates whether it is fastened or not. The passenger seat weight sensor determines whether there is a passenger in the seat or not. The passenger's seat belt sensor indicates whether it is fastened or not.

In order for the system to signal the car to start, the airbag status must be ready, the driver's seatbelt must be fastened, and when the passenger seat is occupied its belt must also be fastened.

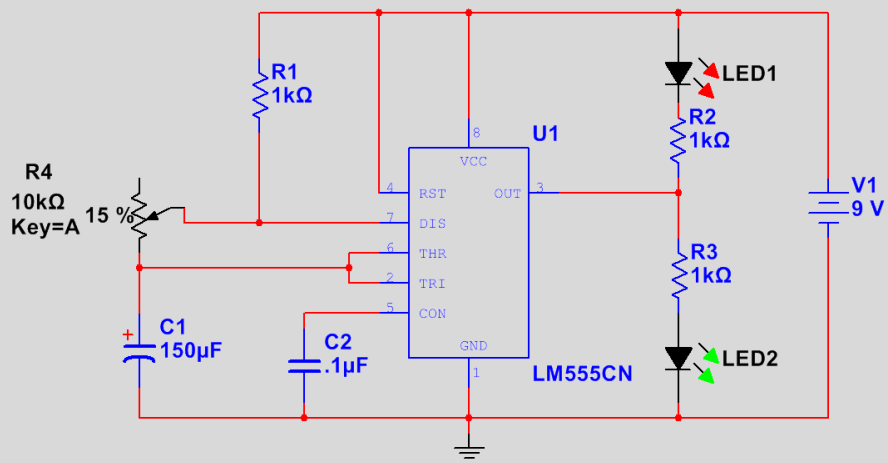
Unit 4: Bridging the Analog and Digital Worlds – The 555 Timer as a TTL Clock Source (4hrs)

This unit introduced the 555 timer chip as a multifunctional device and used it to build a function generator that outputs a TTL clock signal with adjustable frequency. The circuit includes a built-in logic probe so the output states can be observed with LEDs. The 555 timer is a handy tool to teach integrated concepts of both analog and digital electronics. It provided a means to study the charge/discharge characteristics of a capacitor in a R/C circuit and how it can be used to generate a digital signal, thus creating the clock signal and offering a bridge to the concept of sequential

logic systems (and opening a door to discuss the internal components that make up a 555 timer as well). After designing the circuit together as a class, students each built their own function generator with electronic components. They used an oscilloscope to observe the behavior by plotting the capacitor and output voltages over time.

Function Generator Design Exercise:

Using the general layout in the schematic below, design a circuit, which will produce a TTL compatible signal with an adjustable frequency range from 1Hz to 10Hz. A TTL signal is a DC voltage that turns on and off at a regular rate. This signal will drive two LED's which will alternately flash to indicate the high and low (on and off) states of the output (at Pin 3 on the 555 timer component "LM555CN"). Design according to the following parameters:



- Your manager has specified that the company has a surplus of 10 KΩ potentiometers and 150uF capacitors on hand. Therefore, select R_4 and C_1 so that you can make use of these parts.
- When designing your circuit for R_1, R_2, R_3, LED_1, and LED_2, take note that:
  - The LED's have a nominal current rating of 20mA, but we would like to be significantly under that. You must choose R_2 and R_3 values such that the LED current does not exceed the recommended maximum, in this case we will aim for 9 mA.
  - An acceptable design is one in which the lower and upper frequency limits are within 10% of the specified values.
  - Specific resistor values between 1 KΩ and 1MΩ are available.
- The TTL output signal frequency in Hertz is defined by the following equation:

$$f = \frac{1.44}{(R_1 + 2R_4)C_1}$$

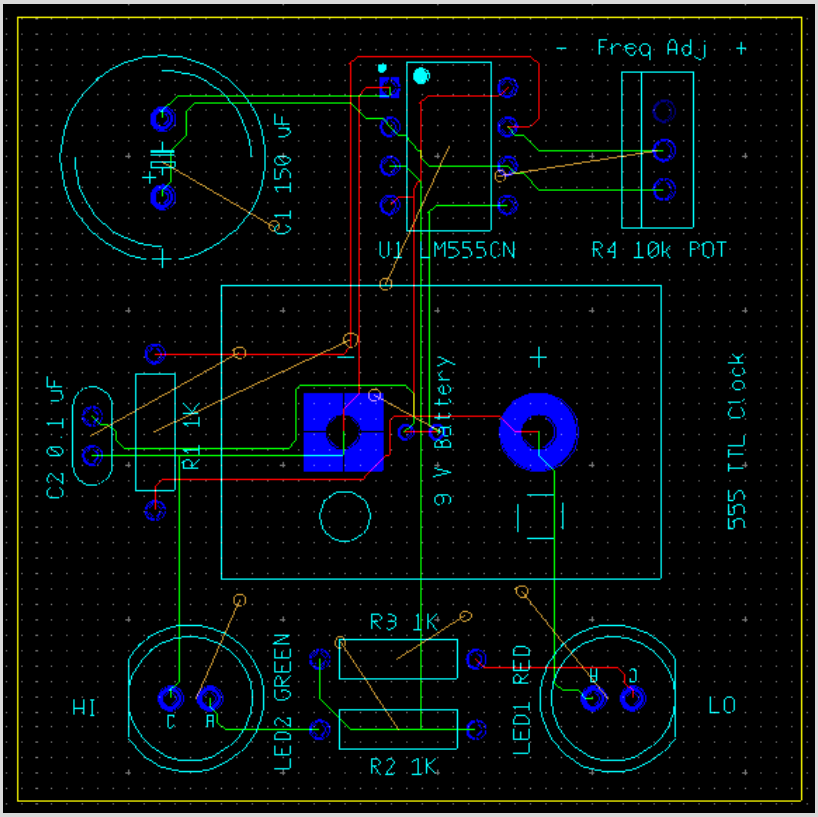
Once we have designed your circuit together in the lab introduction, on your own you need to perform some calculations. Determine the output frequency for each case where R_4 = 0Ω, 3KΩ, 6KΩ, and 10KΩ. The max and min values for R_4 should correspond to your low and high values for the frequency range, respectively.

EXTRA CREDIT: In the lab introduction we discussed how the frequency range could be implemented more exactly by using a parallel resistor to limit the maximum value of the potentiometer. Calculate the resistor needed, add it to your circuit, and test one of the frequency ranges to see if it helped. Calculate your pre-lab values using this data and the resistor when you build your schematic and model it for simulation.

Unit 5: Circuit Simulation and Printed Circuit Board (PCB) Design (4hrs)

Using the circuit designed in the previous unit, this unit introduced students to computer modeling and PCB design. Multisim software from National Instruments (NI) was used to model the 555 timer circuit and simulate its dynamic behavior. Students then compared simulation data with the experimental results they had previously seen on the oscilloscope. Bridging to NI's Ultiboard software, they used their Multisim design, along with packaging and footprint information for real components, to generate PCB design artwork that could be sent out for manufacture. Due to the compressed time scale of the course the boards were not produced, but students were introduced to the process of compiling the necessary files that would be needed to place a custom PCB order. A sample PCB design is shown below.

PCB Design for TTL Function Generator



Unit 6: Fundamentals of Computing Systems – What exactly is a computer? (1hr)

This unit started by asking students what exactly was a computer. Most students identified the traditional computing platforms as a definition of a computer. Students were then introduced to the idea of embedded computers such as a digital cameras, different systems with in a car (e.g. vehicle control unit) that led into the discussion of general purpose and application specific computing. The information flow within a computing platform was used to explain the different components within a computing system such as: memory, arithmetic/logic units, input/output interfaces etc. Students were also introduced to the transistor and how modern computing systems are built all the way from sand to silicon and to integrated circuits[4].

Unit 7: Fundamentals of a Microprocessor – the Arduino and Hardware/Software Interface (5hrs)

This unit focused on the heart of the embedded system – a microprocessor. The goal of this course is to make in-service teachers (i.e. the students in the class) comfortable using a processor system in different applications. The course instructors have familiarity with a wide array of microprocessor platforms, and based on experience with non-electrical engineering students, have found the Arduino platform to have the least barrier to entry. As per the designers of the platform, Arduino was intended for "artists, designers, hobbyists, and anyone interested in creating interactive objects or environments."[5]
Arduino boards have also been successfully used in our first year Introduction to Engineering course to introduce computing systems to students interested in the field of engineering but without much prior experience in programming or using processors. They have also been adopted by many instructors as the choice of processor in mechatronics[6], and microprocessor/embedded system courses[7].

This unit familiarized the students to the different Analog and digital ports, the serial interface, the Arduino programming interface. Since most of the students had very little or no prior programming experience, the unit also focused on the following programming concepts – local and global variable, declaring and calling functions, control structures using if/if-else/for/while/case statements. Students built a simple hardware interface (LED and push-button) to the Arduino and controlled it using software in a lab exercise. The description of the exercise is provided below:

Hardware/Software interface Exercise:

Connect an LED to the Arduino.

- Write code to turn ON the LED.
- Write code to turn OFF the LED.
- Write code to BLINK the LED every 1 second.

Now wire a pushbutton and connect it to the Arduino.

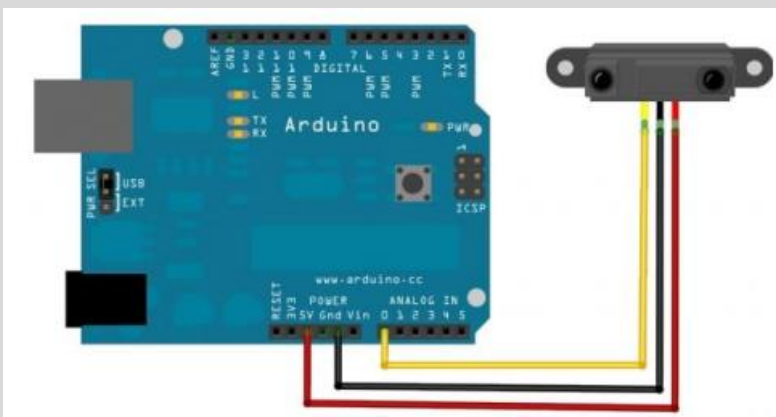
- Write code to read the state of the Pushbutton.
- Add code so that you turn ON the LED when the button is pressed.
- Change your code so that the LED toggles when a button is pressed.

Unit 8: Sensors and Sensor Characterization (5hrs)

A number of sensors were introduced to the students: an ultrasonic distance sensor, infrared proximity sensor, photo-sensor and a passive infra-red motion-sensor. Using datasheets of sensors, the necessary electrical interfaces were determined. For example, the infrared proximity sensor (Sharp IR sensor) did not need any special interface, it need a 5 volt power, a ground connection and a voltage reading corresponding to a particular proximity was provided by the sensor in the third pin. For the passive infra-red motion sensor, the output pin was an *open collector* meaning students had to add a *pull-up resistor* in that pin to read that sensor. The *photo-sensor* was a variable resistor and provided a resistance proportional to the amount of light sensed. Most students had a difficult time figuring out how to read a change in resistance using the microprocessor. Students were then re-introduced to the voltage divider from the previous unit.

This unit also required students to be able to record the sensor readings and do simple averaging using the Arduino processor. To accomplish this, students were introduced to the concept of creating, reading from and writing to arrays.

Sensor Characterization Exercise:



- Connect the Sharp IR sensor to the Arduino as shown above.
- Start a Serial Monitor.
- Read one value and display the value on the screen every time the RESET button is pressed.
- Modify the code to read TEN values from the sensor and display the AVERAGE value on the screen every time the RESET button is pressed. You may want to store the sensor readings in an *array*. The average $\bar{x}$ can be calculated using:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Modify the code to read TEN values from the sensor and display the AVERAGE and STANDARD DEVIATION values on the screen every time the RESET button is pressed. The standard deviation σ can be calculated as:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- Repeat this measurement for a 8 different distances between 10cm and 80 cm. Plot the average sensor reading vs distance. How does your plot match with the one provided in the datasheet?
- Now replace the Sharp IR sensor with a photo-resistor. Using a multi-meter find out how the resistance of your sensor changes with light. Is it high or low when it is bright?
- Using the concept of a voltage divider, connect the sensor to the Arduino analog port. Connect an LED to an output port. The LED should light up every time the light sensor detects a bright source of light.

Unit 9: Pulse Width Modulation and Motors (4hrs)

This unit covered the following topics: Pulse Width Modulation (PWM), H-bridge, using feedback from DC motors, and control of DC servo-motors. Students were introduced to how varying the duty cycle of a PWM signal changes the average analog voltage seen by the load and hence can be used to dim lights or change the speed of DC motors. Students were also introduced to how transistors can be set in an H configuration to create a H-bridge and how the H-bridge can be used to switch the direction of a DC motor. Feedback from motors based on back-emf and rotary encoders were discussed. A servo-motor was used as an example of a DC motor with built in feedback mechanism and the Arduino was used to control the position of the servo. Students were able to integrate the knowledge from the sensors unit with this unit to perform a motor control exercise.

Motor Control Exercise:

- Connect a PUSHBUTTON to pin 2 of the Arduino.
- Connect SHARP IR sensor to ANALOG PORT 1
- Connect the LEFT MOTOR control to pin 9 of the Arduino
- Connect the RIGHT MOTOR control to pin 10

Write a program that will:

- DO NOTHING and wait until the PUSHBUTTON has been pressed.
- Once the PUSHBUTTON is pressed, using the IR sensor go forward until the sensor sees something at least 20cm in front of it.
- Then back up.

Unit 10: Operational Amplifiers as Analog Computers (1hr)

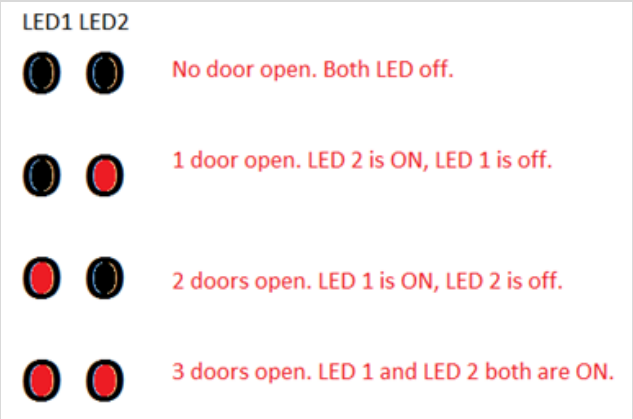
The operational amplifier (op-amp) was introduced as a simple black box (or a black triangle) that could be used either as an amplifier or as a comparator. The comparator aspect of the amplifier was used to highlight how one could design a simple analog computer that performed the same task as the once students had performed earlier with the Arduino. The simple analog computer turned on a LED if a photo sensor connected to the op-amp detected bright light.

Unit 11: Design Problems (4hrs)

This unit was meant to allow students to showcase their understanding of the hardware software interface using two design problems. The first problem included the design of a simple "door counter" (description provided below) required them to wire switches, LEDs and seven-segment displays to the Arduino and write software code.

Design Lab Exercise 1: Door Counter

The Fork Motor Co has hired you and your team to create a system for its latest model EGED699 hatchback car. The hatchback car has three doors – one on the driver side, one on the passenger side and the third door at the back (the hatch). You are to design a circuit that counts the number of open doors on the car at any given time. The total number of open doors is represented in a 2-bit binary format with two LEDs. Each of the doors has a switch that will provide you with logic 1 (5V) if the door is open and logic 0 (0V) if the door is closed.



While designing your system, your engineering team realized that people are not quite familiar with the binary number system. So you decided to make it easier by changing the binary numbers to a digital display format using seven-segment display. A seven-segment display has "7" separate LED segments. If you want to display the number **1**, you light up segments **b** and **c** while turning off all others. **For our 7-segment display, to light up any segment you assign that segment a LOGIC 0.**

This second design problem required students to wire a number of sensors and control a lamp output. The details of the design problem is presented below. This design exercise also introduced the students to transistors to drive larger loads and the use of relays to control a regular 110V AC bulb.

Design Lab Exercise 2: Motion Activated Garage Floodlight

- Design a system using a light sensor, a passive IR motion sensor and a "LED" to simulate the Garage Floodlight such that the floodlight turns ON only when the light sensor detects darkness and motion sensor detects motion.
- Add a two-way switch to your system. Modify the code from the previous part such that your LED "garage floodlight" turns ON only when the light sensor detects darkness and motion sensor detects motion. However, if I am working outside the garage and want the light to ignore the sensors I should be able to put the two-way switch in the ON position. When I have the switch to the ON position the floodlight should stay ON regardless of the sensor readings.
- How would I modify my system if I really wanted to use a 24 DC controlled lamp instead of a LED for the floodlight?
- What would I need to do if I really wanted to use a "real" bulb connected to my real garage switch?

Course Projects

At the end of the course, the students were asked to propose their own project that covered one or more concepts learned in the course. The final project, which spanned one summer month, could be either technical in nature (using the Arduino Hardware/software interface, a digital design project using discrete integrated circuits) or the design of an entire curriculum (multiple lessons) around the Arduino/Electrical Systems. The final projects were evenly split between thorough lesson plans and Arduino based systems. The three technical projects are briefly described below:

a) *Design of an Arduino controlled adjustable power supply.*
This project entailed the designing and building of an Arduino controlled adjustable power supply that can deliver anywhere from 0 to +15V output at 0 to 5A of current from a unit plugged into a household AC outlet. The circuit consisted of AC to DC converter with a step-down transformer, a full-wave rectifier and large capacitors for smoothing. This DC signal was then used to produce a 5V output (1A max) - regulated by a 7805, which provides voltage for digital circuitry & the Arduino microcontroller, a 15V output (5A max) - regulated by a LM338. The third voltage output was created using a differential amplifier and a programmable power resistor array. The value of the output voltage was controlled by 2 pushbutton switches, connected to digital ports of the Arduino - one pushbutton was used to raise the voltage and the other to lower it.

b) *Design of an Arduino controlled car.*
This project involved the redesign of a remote controlled car. The student took-apart a toy car and replaced the RC brain with the Arduino. The Arduino powered vehicle runs forward until a front sensor detects an obstruction in its path directly in front. The vehicle stops with brake lights and runs reverse.

c) *Design of an Arduino virtual Water Ski Slalom course*
The Arduino virtual slalom course removes the need for a physical slalom waterski course for training purposes. The device is attached to the waterski pylon in the boat and tracks waterski rope from left to right as the skier cuts back and forth. When the rope reaches the proper angle that is needed to 'round a buoy' in a physical course, a tone is played to alert the skier and the program is triggered. After each buoy trigger the program measures the elapsed time and displays whether the skier made it 'in time' or 'over time' based on the minimum time needed to round the buoy in a real course. The total number of buoys rounded in time is also recorded and displayed on the LCD. A design mounted on a boat and used for a slalom course is shown in **Figure *1***.

Figure 1 Completed Virtual Slalom project mounted on a boat and used for a slalom course.

Course Assessment

At the end of the semester, we asked students to fill out an anonymous survey regarding the course and the projects. All seven students who took the course responded and rated a number of things including their progress on course objectives. For the four learning objectives presented in earlier section, a rating of 1 indicates that the students made no apparent progress on the learning objective while a rating of 5 indicates that the student made outstanding progress towards those learning objectives. The instrument used for the survey was the IDEA center's *Diagnostic Survey Form*[8]. The results are summarized below in Table 1.

Table 1 Results of End-of-semester survey about the course.

	Average	**Std. Deviation**
Learning Objective 1: Gained factual knowledge (terminology, classifications, methods, trends) of electrical and computer systems.	4.7	0.5
Learning Objective 2 (a): Learned fundamental principles, generalizations, or theories related to circuits and computer systems	4.9	0.4
Learning Objective 2(b): Developed an ability to apply those principles to the design and analysis of practical circuits as demonstrated in laboratory experiments and student design projects.	4.7	0.5
Learning Objective 3: Developed specific skills and competencies related to the use of software tools and lab instruments to design, simulate, build, test and document embedded systems using the Arduino platform.	4.6	0.5
Learning Objective 4: Learned how to find and use resources for answering questions and solving problems in regards to the course project.	4.9	0.4
As a result of taking this course, I feel more comfortable with the area of Electrical and Computing Systems	4.7	0.5
This was an Excellent Course.	4.9	0.4

Based on the results of Table 1, the students made good progress on the learning objectives and the course was well received by the students. The student comments that accompanied the course evaluation mentioned that the course made them feel more comfortable and confident in undertaking an electrical and computer systems project with their students. The comments also highlighted the usefulness of the materials learned in the class to the courses that the teachers teach or the student clubs they advise. Here are a few sample comments:

> *"Very useful course for me to prepare to develop a STEM program."*
>
> *"I had limited electronics and computer background and the course helped me fill in the gaps."*
>
> *"Very good information for me to take to my own classroom".*

At the conclusion of the year 2013, we reached out to all students to find out how they have or are planning on integrating the knowledge from this course into their own courses and activities. A summary of the responses is provided below:

- Two students are integrating modules from this course into a three-year high-school engineering institute. The students have designed and taught an introductory unit within the first course (designed for high school sophomores) that would expose students to the basics of computer and electrical engineering and set the groundwork for a much more in-depth unit (4-5 weeks) during the second-level course the students will take during their junior year next year.
- Another student is using the materials in his Physics course. Similar to the concept of *"Physics First"* which focuses on the introduction of Physics as the first science course in high-school before biology or chemistry, the teacher has adopted an *Arduino First* concept and started the students on the Arduino after teaching a chapter on electronics. He has found the Arduino projects as a great way to keep the students engaged and plans on using it in his upper level Physical Science course.
- A student who is spearheading the creation of a STEM program that launches in the Fall 2014 together with his teaching colleagues is using the lessons in a beginning robotics course (including taking apart the toy car and making it function differently with an Arduino brain).

Reflection

Based on our observation of the course, several changes are warranted before the next offering of the course. While we intended the course to be open to K-12 instructors, we found only 8-12 grade teachers enrolled in the class. We also found that the science teachers have varying background and comfort with hardware and software systems; and that that students struggle more with software than hardware and sometimes lose interest in the material because of the frustrations of programming. Our goal in this course was to provide a flavor of both electrical and computer engineering disciplines by using an embedded platform as the common vehicle for content delivery.

Since the course covered a little bit of analog electronics, digital electronics, computing systems and programming, we found that the success of this course was primarily due to the motivation and commitment shown by the students. To allow students more time to dig deeper into these subject areas and master them with more hands-on examples, for the next offering of the course we plan to separate the course into two courses - one covering basic electrical concepts and analog devices and the second one covering digital and computing systems.

- Fundamentals of Electricity and Electronics
- Digital Electronics and Computing Systems

Our hope is that students will either take both courses as a series or take one of the two depending on their comfort level and their expected use of the material in their own curriculum.

Conclusion

This paper has presented an overview of a new course on the fundamentals of electrical and computer system designed for in-service K-12 teachers. Analysis of the end of semester survey shows that the course met its learning objectives. Our own assessment based on feedback from students suggests that the use of the Arduino among many other embedded platform was the best one for our needs. The platform was straight-forward for students with little to no programming language to use and students have found wealth of information online when they have needed it. The low cost of the platforms have also allowed the students to easily and quickly integrate the units from this course into their own curriculums.

References

[1] Minnesota Academic Standards. http://education.state.mn.us/mde/index.html
[2] Center for Engineering Education. http://www.stthomas.edu/cee
[3] U. Roy, "Preparing engineers for future with collaborative technology," *Journal of computer applications in engineering education*, vol. 2, pp. pp. 99-104, 1998.
[4] Global Foundries, "Sand to Silicon." Online video clip. Youtube. https://www.youtube.com/watch?v=UvluuAIiA50
[5] Arduino website. http://arduino.cc
[6] J. A. Riofrio, S. Northrup, "Teaching Undergraduate Introductory Course to Mechatronics in the Mechanical Engineering Curriculum Using Arduino," *Proceedings of the 2013 American Society for Engineering Education Annual Conference*, Atlanta, 2013.
[7] N. Bird, "Use of the Arduino platform for a junior-level undergraduate microprocessors course *Proceedings of the 2011 American Society for Engineering Education Annual Conference*, Vancouver, Canada, 2011.
[8] http://www.theideacenter.org/