Computational Instruction through PLCs in a Multi-Disciplinary Introduction to Engineering Course

Mr. Nicholas Hawkins, University of Louisville

Nicholas Hawkins is a Graduate Teaching Assistance in the Engineering Fundamentals Department at the University of Louisville. A PhD student in Electrical and Computer Engineering, he received both his B.S. and M. Eng. from the University of Louisville in the same field. His research interests include power electronics and controls, as well as engineering education for first-year students.

Dr. James E. Lewis, University of Louisville

James E. Lewis, Ph.D. is an Assistant Professor in the Department of Engineering Fundamentals in the J. B. Speed School of Engineering at the University of Louisville. His research interests include parallel and distributed computer systems, cryptography, engineering education, undergraduate retention and technology (Tablet PCs) used in the classroom.

Dr. Brian Scott Robinson, University of Louisville
Dr. James Christopher Foreman, University of Louisville

Asst. Professor at University of Louisville, previous appointment at Purdue University. Teaching calculus, power and energy, and industrial control systems related courses. Research in artificial neural networks, expert systems, and new methods of teaching math/calculus. 15 years in industry control systems and power generation industry prior to academic career.

Computational Instruction through PLCs in a Multi-Disciplinary Introduction to Engineering Course

Abstract

This paper is focused on the inclusion of hardware-based Programmable Logic Controllers (PLCs) in a first-year engineering introduction course, *Engineering Methods, Tools, & Practice II*, at the University of Louisville's JB Speed School of Engineering. Each academic year, more than 500 first-year engineering students are exposed to this interactive course, which introduces students to fundamental engineering skills – including teamwork, design, project management, technical writing, critical thinking, programming, communication (including written, oral, and graphical), 3D printing, and an introduction to engineering research.

Previously, this course culminated in a final project that combined construction and mechanical design of a windmill system utilizing data acquisition driven solely by Arduino programming and circuitry. The most recent iteration of the course added the instruction of PLCs through the collaboration with an industry partner prior to beginning the final project. The PLCs provide the students with a secondary form of computational methodology instruction.

A key benefit of the PLC addition is the variation in logic and programming approach that is exposed to students. The inclusion of hardware based PLCs allows for the instruction of ladder logic, which is an alternative form of programming that gives students a different perspective. PLC control logic provides a sequential, real-time, and loop-based perspective versus the typical procedural and object-oriented approaches to which most students are limited in engineering education. The PLC addition is also beneficial in that it provides student exposure to systems and programming principles that are applicable within numerous industrial settings, which can also be useful when interviewing for their first co-operative internship during their sophomore year as many engineering disciplines rely on controlling systems and processes. Since the course continues to employ Arduino programming in addition to the PLC ladder logic programing, students are provided with a greater breadth of programming exposure and experience in integrating multiple hardware systems (PLC and Arduino) within their project.

Introduction

Makerspace pedagogy is an increasingly utilized methodology for teaching undergraduate engineering students about the fundamental tools and practices of engineers within an industrial setting. The second component of a first-year introduction to engineering sequence at the University of Louisville's JB Speed School of Engineering utilizes a makerspace setting and is focused on integration and application of institution-identified fundamental engineering skills and all first-year engineering students are required to take this sequence. This paper is predominantly focused on the addition of Programmable Logic Controllers (PLCs) to the programming aspect of course curriculum. Though not all engineering careers mandate understanding of the electronics instruction discussed within this paper, many employers utilize analogous hardware and programming such that the need for familiarity with these systems has become increasingly multidisciplinary. Examples include process control in Chemical Engineering, Electrical Engineering, and Industrial Engineering as well as control at the systems level in Mechanical and Electrical Engineering, as well as other engineering disciplines that include control of small-scale devices, e.g., Bioengineering. Knowledge of this was a key reason the decision was made to include PLC programming instruction in addition to already-utilized Arduino platform within the course.

Instruction of industrial control systems (such as PLCs) are typically experienced in later coursework of an engineering student's undergraduate degree program, after theory and introductory subjects have been explored. Laboratory experiences are costly both in terms of instructor time and money, especially in the case of damaged equipment [1]. Thus, to overcome this, various courses have attempted to use web-based laboratories to educate engineering students on electronics such as PLCs [2, 3]. However, this approach waives the inclusion of experience-based hands-on education, which is considered to be a crucial part of the laboratory experience. As an alternative and/or supplement, asynchronous labs have been developed [4] to allow for the hands-on experience while maintaining the flexibility and low-cost of doing so external to a traditional lab.

However, instruction through longer-term projects, which span multiple lab sessions versus individual labs, is quite advantageous because it is similar to how the engineering profession functions in industry [5]. Not only does it involve hands-on learning, it utilizes the advantage of an instructor being present to assist the student(s) [6, 7].

Course Structure

The course discussed in this paper is *Engineering Methods, Tools, & Practice II* (ENGR 111), the second component of a required first-year introductory sequence that typically enrolls over 500 students each academic year from across all engineering disciplines. This course takes place at the university's Engineering Garage, a 15,000 ft^2 makerspace that provides the course with individual classrooms in addition to a laboratory-analogous makerspace.

ENGR 111 covers a wide range of topics meant to prepare students for the laboratory and field work they might experience throughout their academic and professional careers. The university has a mandatory co-op program with industry engagement that begins after the first year, and feedback from partnered employers informs and motivates many of the topics taught in this preliminary course. These topics include teamwork, design, project management, technical writing, critical thinking, programming, communication (including written, oral, and graphical), 3D printing, and an introduction to engineering research. The culminating feature of this course is a final project, called the Cornerstone Project, which includes the construction, optimization, and mechanical design of a windmill system. This system includes the integration of student-built AC motors, DC motors, and data acquisition systems that students use to discern quantifiable results for their windmill, such as efficiencies, power output, and windmill speed – accomplished via the construction of a tachometer, which the students design and build.

The first implementation of this course was in Spring 2017, in which the programming aspects of the Cornerstone Project were executed exclusively by Arduinos. The following iteration of the course, in Spring 2018, included the use of hardware-based Programmable Logic Controllers (PLCs) in addition to the Arduinos platform.

Arduino-Driven Cornerstone Project

As mentioned, the original Cornerstone Project data acquisition system was solely driven by the Arduino Uno microcontroller, illustrated in Figure 1 [8], for both circuitry and programming. The Arduino is an excellent microcontroller for teaching basic circuitry and programming to aspiring engineers, as it has easily accessible digital and analog inputs/outputs (I/O) and is run through a variant of C programming.

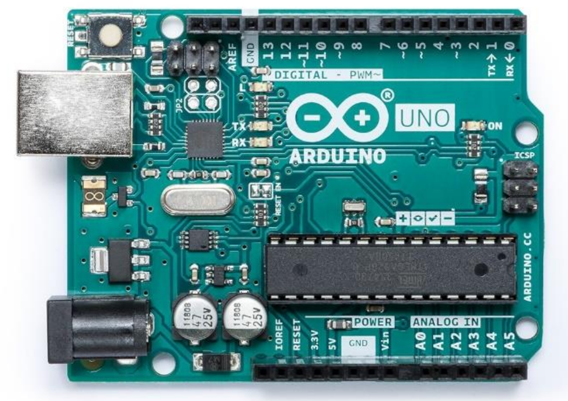

Figure 1: Arduino Uno microcontrollerr

The tachometer is built using two infrared components: an infrared LED and a phototransistor. The LED shines an infrared light outward, which the phototransistor can detect. Any break that occurs within the line of sight between these two components can be measured and accounted for via programmed algorithms. Accordingly, there are two areas within the windmill system in which these sensors can be installed: 1) on opposing sides of the windmill gearing (gears utilized in this course are fabricated with equally spaced holes along the medial circumference), or 2) opposing sides of the windmill blades. It is up to the student(s) to figure this out. They will then make a team-based decision on which approach they prefer, followed by design, modelling, and 3D-printing a bracket that houses the sensors. The students include within their program the number of breaks that occur per revolution and must use this to obtain the rotational velocity of the windmill blades in revolutions per minute (RPM). After the students construct and program their tachometers, they are then able to calculate other quantities for their windmills, such as power and efficiency.

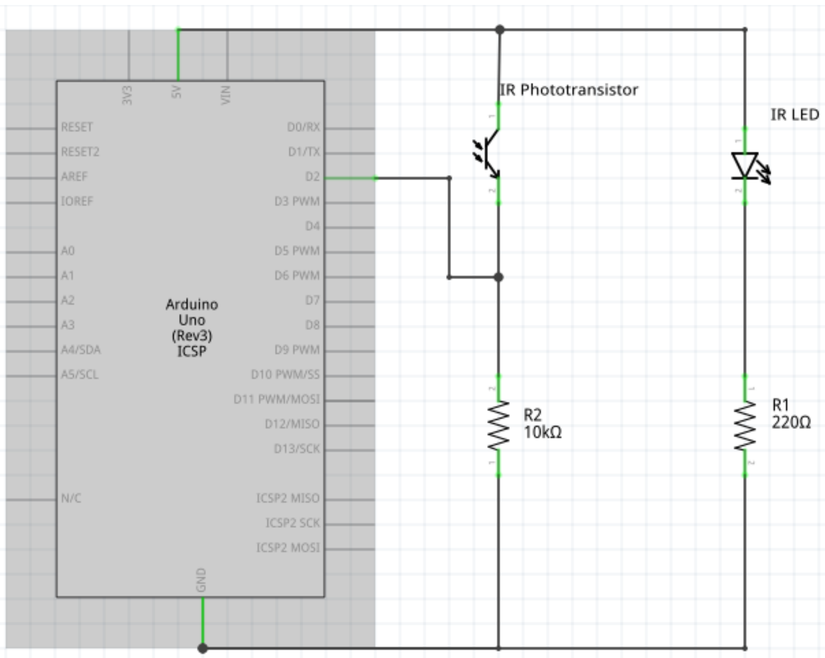

Figure 2: Tachometer schematic provided to students for Arduino circuit

The hardware component of this task is provided to the students in the form of a schematic, as shown in Figure 2. Earlier in the course, students were taught how to read basic schematics and build them on breadboards.

PLC-Arduino Combined Cornerstone Project

To account for the new PLC inclusion, two additional course sessions were created for providing students instruction on PLC programming and hardware. This was done by removal of an open session initially used for working on intermediate course deliverables, and by removal of an open session previously reserved for working on final course deliverables (such as the Final Report). Accordingly, previous instruction related exclusively to Arduino programming remained unchanged, thus alleviating concerns related to students becoming less versed in algorithmic programming due to the PLC instruction inclusion. These two added sessions were primarily focused on teaching students PLC basics while at the same time building and programming towards the tachometer assembly for the Cornerstone Project.

The design parameters and overall outcomes of the Cornerstone Project were kept intact with the inclusion of PLCs. Students continued to use data acquisition and programming to analyze their windmills, now employing the integration of two systems, PLC and Arduino, for a more diverse experience. Beyond this, the primary functional change was using a PLC (instead of Arduino) for RPM data acquisition. The PLC used for this course is an Allen Bradley MicroLogix 830 Programmable Logic Controller [9], illustrated in Figure 3. Like the Arduino, it includes several input and output ports. Unlike the Arduino, this PLC does not include analog ports, only digital I/O. The software that is used to program the PLC is Rockwell Automation’s Connected Components Workbench (CCW) [10], a PLC ladder logic programming platform.

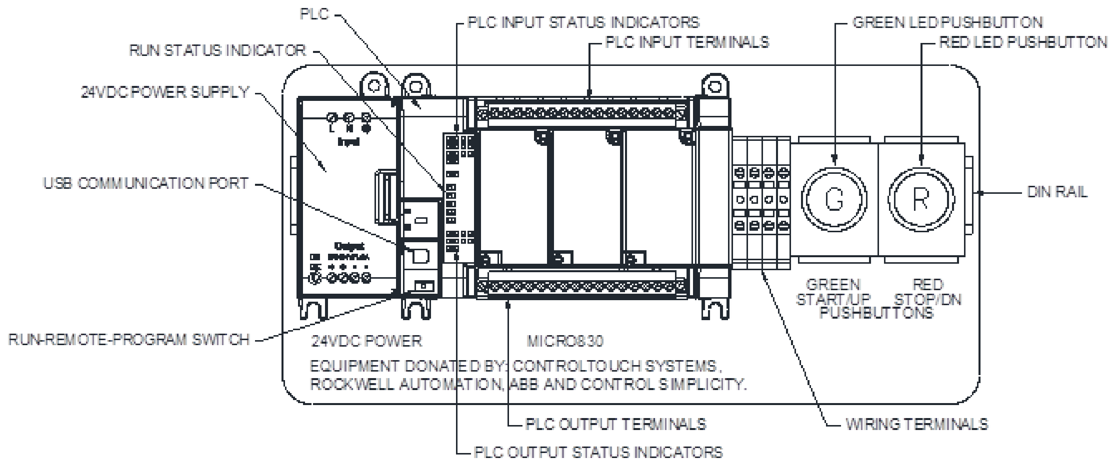

Figure 3: Allen Bradley PLC Diagram

The hardware consists of similar components, but the tachometer circuit was altered to connect to the PLC instead of the Arduino. As seen in Figure 4, the LED remains powered by the Arduino, but the phototransistor is now connected to the PLC. Both would have been connected to the PLC, but the LED isn't rated for the 24V supply that the PLC uses. Otherwise, the hardware changes are minimal.

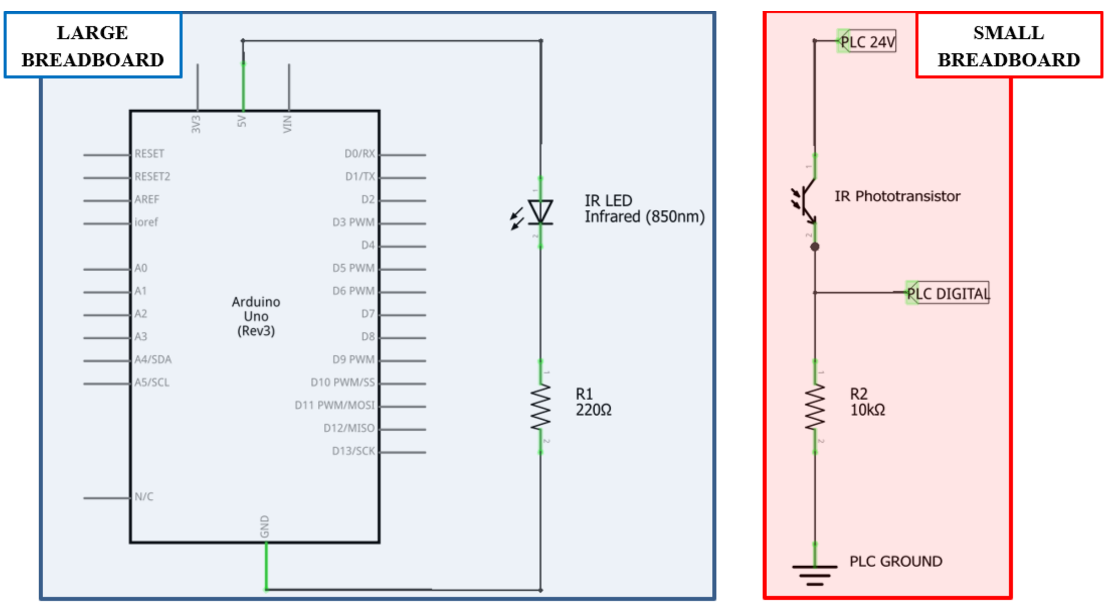

Figure 4: Tachometer circuit configured for use with PLC

With respect to the addition of the PLC, the programming aspect represents the most significant change from the original tachometer-reading methodology. CCW programs the PLC using ladder logic, which is a more visual form of coding. Ladder logic uses the premise that electricity always wants to travel from a power source to ground. As shown in Figure 5 [10], each "rung" of the ladder, by default, connects the left (power) and right (ground) sides together. However, contacts can be added to interrupt the flow of electricity (current), and coils can be added to detect if the electricity is flowing, i.e. the circuit is complete. Normally-open contacts prevent flow unless a certain condition is met, whereas normally-closed contacts allow flow unless a condition is met. By programming these contacts, the flow of electricity can be started and stopped in a logical fashion. To accomplish the needs of the tachometer circuit, students are introduced to a timer functional block that can be used in ladder logic to count inputs over a set length of time.

Instructor-Identified Advantages of the PLC-Arduino Combined Cornerstone Project

Assessments related to the PLC implementation from student perspective(s) are not applicable since there is no basis for student comparison to the course experience when PLCs were not utilized. In other words, course iterations in which Arduino was solely used and both Arduino and PLCs used, respectively, were experienced by completely different cohorts. Previous student surveys administered upon conclusion of the course have been focused primarily on critical thinking and teamwork development. For future course iterations, administrators plan to include student survey questions that inquire upon thoughts and comparison between the two different programming interfaces. Since the course was only recently launched, it is too early to assess changes made in retention and/or student improvements in later disciplinary courses focused on programming. Nevertheless, course instructors that have been present for both course iterations have noted some instructional advantages discussed in the following text.

In addition to the aforementioned reasons that motivated the inclusion of PLCs, another significant difference the inclusion brought about was changing the hardware students use while constructing their tachometer circuits. Upon the PLC implementation, the original, somewhat

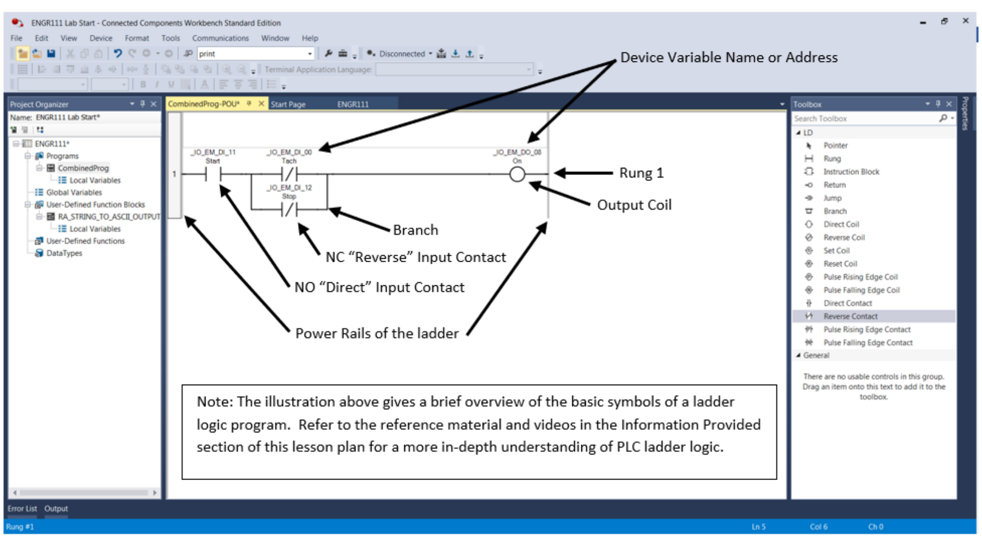

Figure 5: An example of PLC ladder logic in the CCW platform

robust circuit was split into two smaller circuits, which enhanced the opportunities for troubleshooting related issues. Designs in circuitry, especially those created by aspiring engineers, are often physically disorganized, making it difficult for instructors to adequately teach the students how to solve related problems. Moving half of the circuit to a separate prototyping area creates distinct, discrete parts that become easier to analyze.

Another advantage for students is the variation in software development. The addition of ladder logic allows students, who may not possess the skills of proficient programmers, an alternative programming approach to which it may be easier for them to adapt. Algorithm-based programming requires the ability to analyze abstract concepts that are often intangible for new and inexperienced engineering students. PLC ladder logic provides a conceptual alternative, where students of all disciplines can learn to associate programming with the flow of electrical current in control devices, providing students with a window for visualizing the hardware component of software-based control. This is designed to allow students to experience the systems on which they work more completely. In comparison with the original, Arduino-only iteration, course instructors observed a larger portion of the ENGR 111 student body that were much more engaged with this programming component of the Cornerstone Project under the current, PLC-Arduino combined iteration.

Conclusions and Future Work

The authors of this paper strongly adhere to the belief that effective engineering education requires some hands-on experience, regardless of discipline. The JB Speed School of Engineering at the University of Louisville teaches a first-year, makerspace-based course aimed at instructing all engineering students in the fundamentals of the profession, including an introduction of programming and electronics that may be encountered within industry. The addition of Programmable Logic Controllers to the final project of this course improves the realism of the systems students work with and provides a variety of beneficial hardware and software experiences, e.g., systems integration. These experiences help prepare students for a competitive co-operative internship program, as well as their experiences beyond academia.

Partnership with industry is another key aspect of engineering education as this type of experience serves to both demonstrate the engineering profession to students, as well as to build relationships between industry and academia. Future work in this course will include fostering the collaboration of more industrial partners in order to produce more Cornerstone Project ideas, beyond this windmill, and engage students in an ever increasing diversity of industry applications. Some examples of this collaborative outreach are a water filtration system with the local wastewater treatment utility, and potential projects with a local material handling company and the local power utility. Such outreach increases the visibility of the engineering program to aid in the recruitment of new students and exposes current students to more industrial companies who may hire them upon graduation.

References

[1] L. Gomes, J. J. Rodriguez-Andina and S. Bogosyan, "Current Trends in Industrial Electronics Education," *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* pp. VOL. 57, 3245 - 3252, 2010.

[2] F. K. Can Saygin, "A Web-based programmable logic controller laboratory for manufacturing engineering education," *The International Journal of Advanced Manufacturing Technology,* vol. 24, no. 7 - 8, p. 590–598, 2004.

[3] P. Y. H. SHENG-JEN (‘‘TONY’’) HSIEH, "Web-Based Modules for Programmable Logic Controller Education," *Wiley Periodicals,* pp. 266 - 279, 2005.

[4] G. P. H. Burford J Furman, "Asynchronous hands-on experiments for Mechatronics education," *Mechatronics Education in Europe and the United States,* vol. 12, no. 2, pp. 251-260, 2002.

[5] V. G. Agelidis, "The future of power electronics/power engineering education: challenges and opportunities," in *IEEE Workshop Power Electronics Education*, Recife, Brazil, 2005.

[6] L. Guo, "Design Projects in a Programmable Logic Controller (PLC) Course in Electrical Engineering Technology," *Technology Interface Journal,* vol. 10, no. 1, 2009.

[7] J. K. Romanas V. Krivickas, "Laboratory Instruction in Engineering Education," in *11th Baltic Region Seminar on Engineering Education*, Tallinn, Estonia, 2007.

[8] Arduino, "Arduino Uno Rev3," Arduino, [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3. [Accessed 2019].

[9] Allen-Bradley, "User Manual - Micro830 Programmable Logic Controller Systems," February 2018. [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/um/2080-um002_-en-e.pdf. [Accessed 2019].

[10] R. Automation, "Connected Components Workbench Software Overview," Rockwell Automation, [Online]. Available: https://www.rockwellautomation.com/global/detail.page?content_type=tech_data&docid=4740cbbe63a3b91551682bc7b0c6352d&pagetitle=Connected-Components-Workbench-Software-Overview. [Accessed 2019].