Teaching Industrial Control with Open-Source Software

Dr. Hugh Jack P. Eng., Western Carolina University

Dr. Jack is the Cass Ballenger Distinguished Professor of Engineering in the School of Engineering and Technology within Western Carolina University. His interests include robotics, automation, and product design.

Dr. Scott C. Rowe, Western Carolina University

Scott Rowe is an Assistant Professor in Western Carolina University's School of Engineering + Technology. He joined Western Carolina University in 2021 after studies in concentrated solar power and controls engineering at the University of Colorado Boulder. Scott's research relates to accessible and inexpensive engineering equipment for laboratory education.

Teaching Industrial Control with Open-Source Software

Abstract

This paper presents an innovative approach to teaching Programmable Logic Controllers (PLCs) using open-source software and low-cost hardware in an engineering curriculum. The OpenPLC software and a variety of affordable hardware platforms, such as Arduino and Raspberry Pi, are employed to provide students with hands-on experience in programming PLCs. The incorporation of PLC content in the second year of the curriculum prepares students for summer internships, better satisfying industry demands and enhancing their careers. This approach is also beneficial for multidisciplinary project-based learning courses throughout the engineering program. Although a formal assessment of the approach's effectiveness is yet to be conducted, anecdotal evidence suggests positive outcomes. Overall, this paper demonstrates the value of using free software and low-cost hardware in teaching PLC concepts, paving the way for more accessible and cost-effective education in this crucial area of engineering.

Introduction

Industrial control systems are heavily reliant on Programmable Logic Controllers (PLCs). These controllers are specialized computer systems with inputs and outputs designed for high voltages and currents. Moreover, they utilize programming languages that support real-time systems in mission-critical environments. In simpler applications, a PLC can monitor a sensor and actuate motors or pneumatic cylinders. In advanced applications, the controllers can interact with databases, use touch screens for operator input and output, network, interface with vision systems, etc. In a manufacturing environment, PLCs can be found on simple machines for tasks like pushing caps onto bottles. In more advanced applications they can coordinate the operation of massive facilities, like steel mills. As a result there is a high industrial demand for graduates with knowledge and skills in these areas. Most 2-year community colleges offer some courses and degrees related to PLCs. Their graduates can be well prepared to build and maintain PLC based systems in industry. Some 4-year schools teach PLCs in their Engineering Technology and Engineering programs. Students that graduate from these programs are able to move into system design and planning roles. Students that learn to design and implement PLC-based systems are easily employed in some of the highest-paying jobs.

PLC courses normally include laboratory and/or project components. Students learn to use hardware and software from a particular vendor. While PLC vendors often offer generous discounts for education, the cost to outfit a laboratory is still very high. And, licensing limitations often prevent students from installing software on their own computers.

The OpenPLC open source PLC software [1] was chosen to support a PLC course for second-year students in an Electrical and Computer Engineering Technology program at Western Carolina University. The Open-Source project provides a programming environment with standard PLC programming languages like Ladder Logic, Structured Text, Sequential Function Charts, Instruction List, and Function Block Diagrams. Students can write programs and download the results to a wide variety of low-cost hardware platforms. Some of the lower-cost options include Arduino [2] and Raspberry Pi [3] boards. For the purpose of the paper some specialty hardware was purchased that was Arduino based but included numerous input and output cards used in commercial PLCs. Using OpenPLC expanded the students' access to programming software, without compromising the support for low and intermediate-level education.

The Course

The Electrical and Computer Engineering Technology (ECET) program at Western Carolina University is focused on preparation for industry. The regional employers are typically discrete part and product producers that make extensive use of industrial control systems. Graduates of the schools typically go to work for aerospace, automotive, medical, and similar industries.

The school receives annual feedback from external sources including an Industrial Advisory Board (IAB), capstone project sponsors, alumni, and industrial partners. They all share a common message that there is increasing demand for mechatronics knowledge that is not being satisfied by higher education. More notably, they are focused on Programmable Logic Controllers (PLCs). Companies need well-trained technicians with 2-year degrees to build and maintain the equipment. 4-year schools are expected to provide engineers that can do design work and higher-level systems integration.

The ECET program has embraced the call for PLC skills and is deploying the content in the curriculum. An introductory course ECET 290 - Computer Engineering Fundamentals was chosen as the entry-level PLC course. The current course description is provided below. In a future semester, the course description will be updated, but for now it is adequately addressed by the PLC content.

> ECET 290 - Computer Engineering Fundamentals - An introductory course in the study of computer engineering technology. Operating systems, CPU, memory, networking, user interfacing, programming, and basic signal processing and associated hardware. 2 Lecture, 2 Lab. Credits 3

The topics in the PLC version of the course are listed. For illustration, the lecture topics in the current course description are indicated with square brackets *[]*. The course is using a PLC textbook, with a focus on structured design, was written for students in 4-year programs [4].

- The basic concepts of PLCs and programming in ladder logic
- Inputs and outputs including sensors and actuators *[signal processing, hardware]*
- Wiring and system layout *[user Interfacing]*
- Boolean algebra, numbering systems, and data representation
- PLC structure and operation *[Operating systems]*
- Sequential logic using timers and counters
- State-based programming
- Memory types *[CPU, memory]*
- Complex data functions
- Structured Text programming
- C Programming introduction
- C programming for real-time applications

The lecture topics were complemented by laboratory work, for Spring 2023 topics are listed below. The open-source hardware and software platforms played a crucial role in the laboratory work, as discussed in further detail later in the paper.

1. Unboxing the hardware, assembly on a DIN rail, and basic wiring
2. Openplc software installation and testing with PLCs
3. Addition of sensors and buttons
4. Simplified project (a burglar alarm)
5. Autocad Electrical - basic wiring practices
6. Developing a full control wiring diagram
7. Component selection
8. Sequential programming with timers and counters
9. Programming with timing diagrams
10. Programming with state diagrams
11. Programming with basic functions
12. Programming with intermediate functions
13. Structured text programming
14. C Programming with the Arduino IDE
15. Real-time control with C

The Software

The OpenPLC software is still in development and not suited for use in mission-critical systems (yet.) But it has some distinct advantages over other platforms.

- It is free. Industrial software for programming PLCs is very expensive, even when discounted.
- It supports all five IEC 61131-3 languages [5].
- It runs well on Linux and Windows (the Mac version is in a Beta stage.)
- It includes a simulator in the programming environment.
- The hardware it supports can be very low cost.

Some of the disadvantages are:

- The debugger does not allow PLC monitoring directly.
- It supports a very limited set of hardware.

Figure 1 shows the basic user interface with a simple ladder logic program. In the upper center, the variables (tags) are defined. In this case, since the physical locations are not defined, the memory used is virtual or internal. The program is simple combinatorial logic where X = (A + B) * C. As shown, the simulator is running. On the right, the debugger tab shows variable values. Students can change these values to test and verify the system without physical hardware. This can be helpful when testing small programs during lectures.

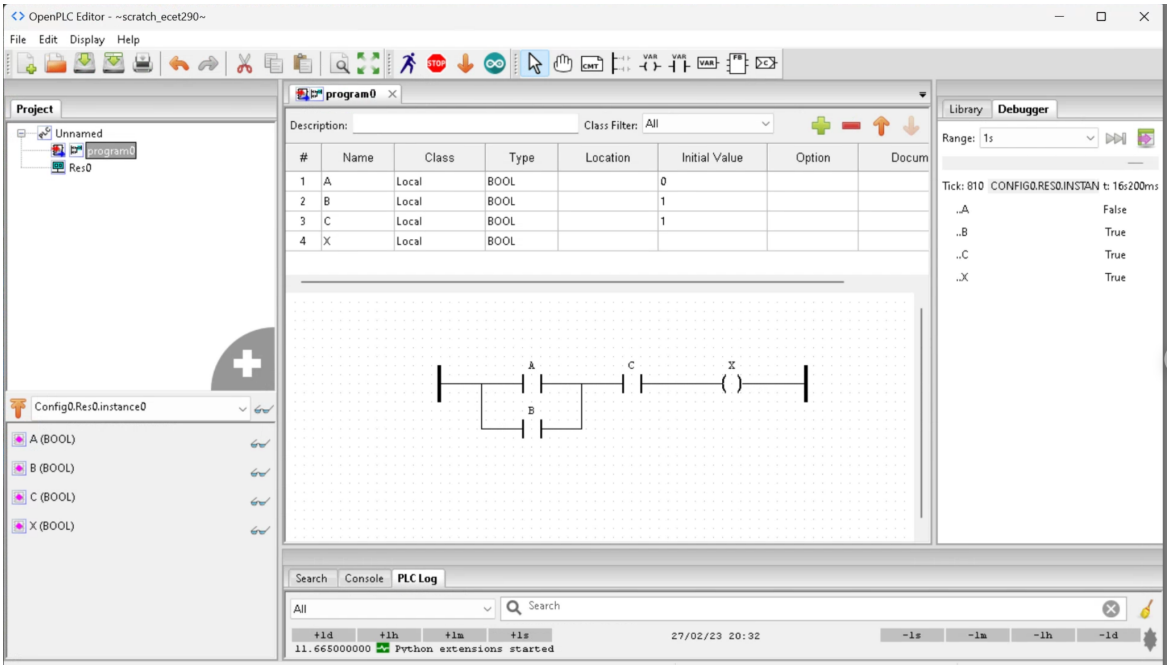

Figure 1 - OpenPLC with a Simple Project

Programs in the other IEC programming languages are shown in Figures 2 through 5.

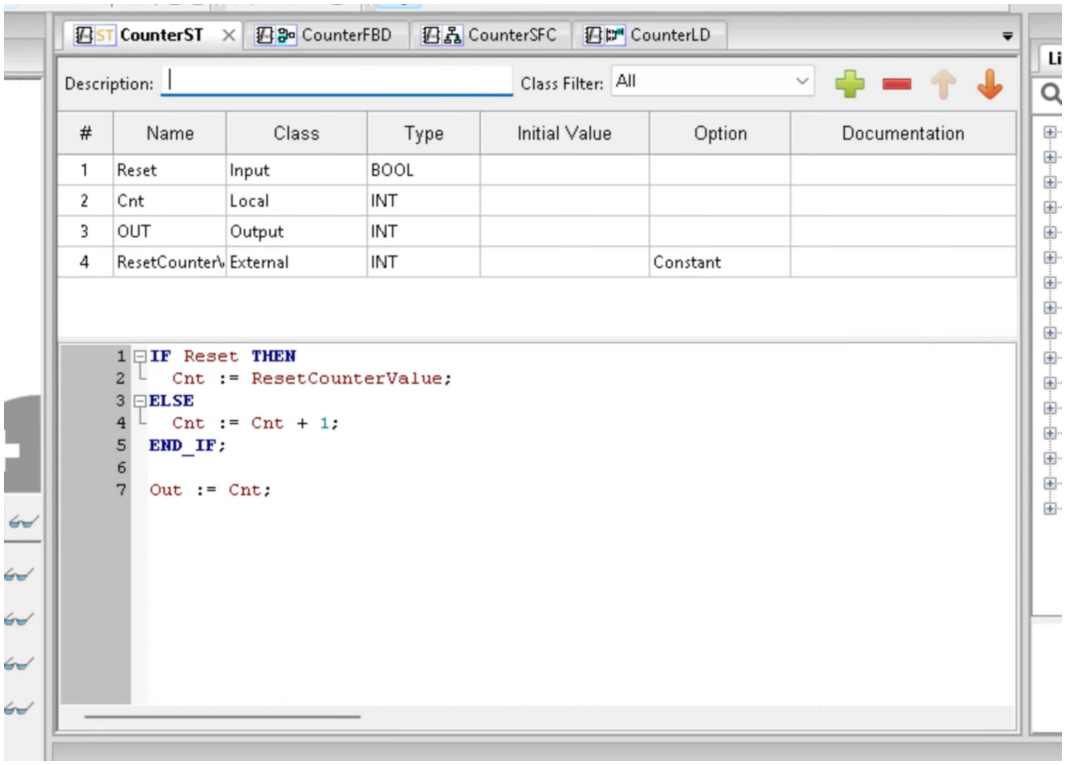

Figure 2 - A Structured Text Program in OpenPLC

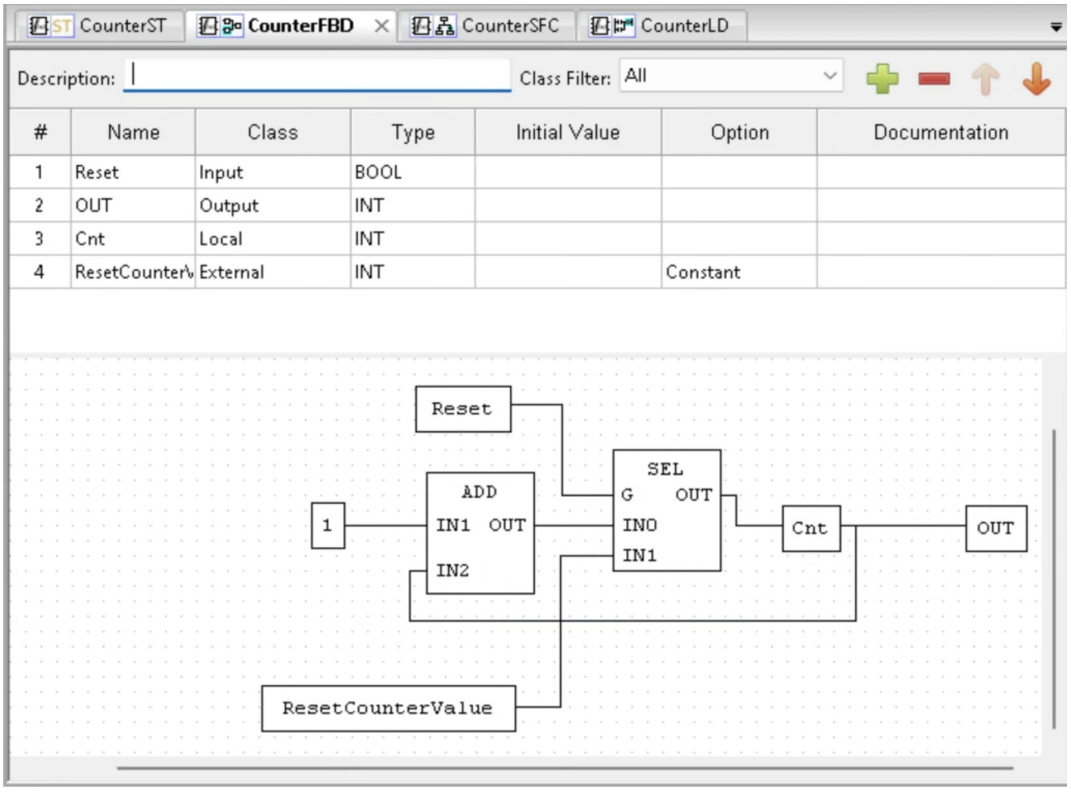

Figure 3 - A Function Block Program in OpenPLC

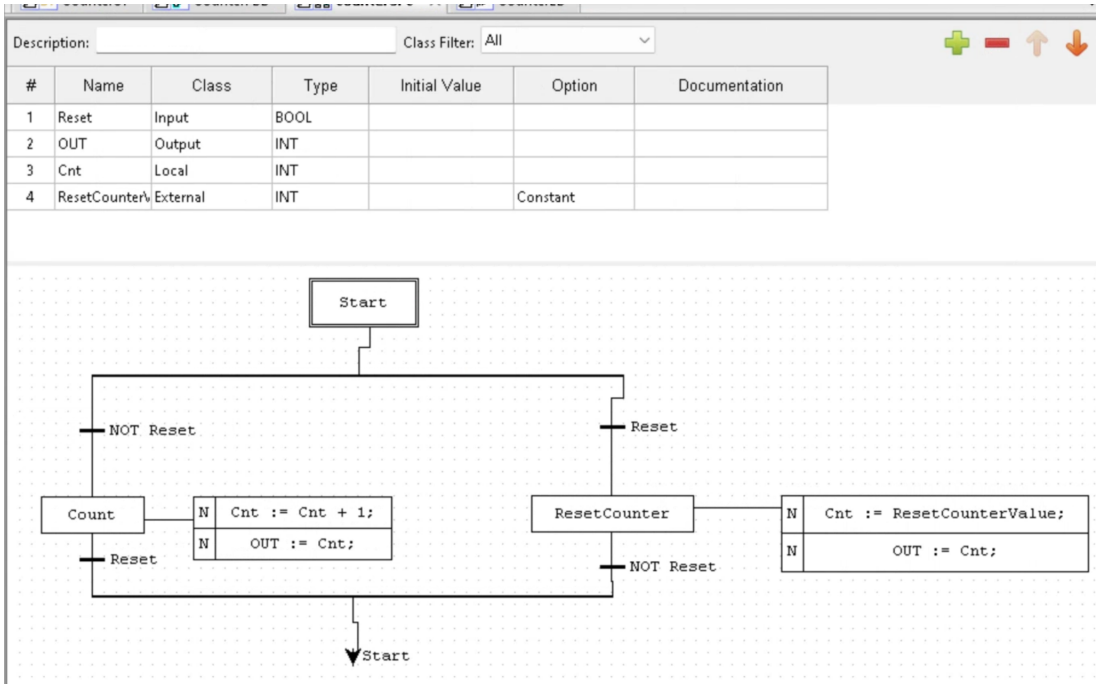

Figure 4 - A Sequential Function Chart Program in OpenPLC

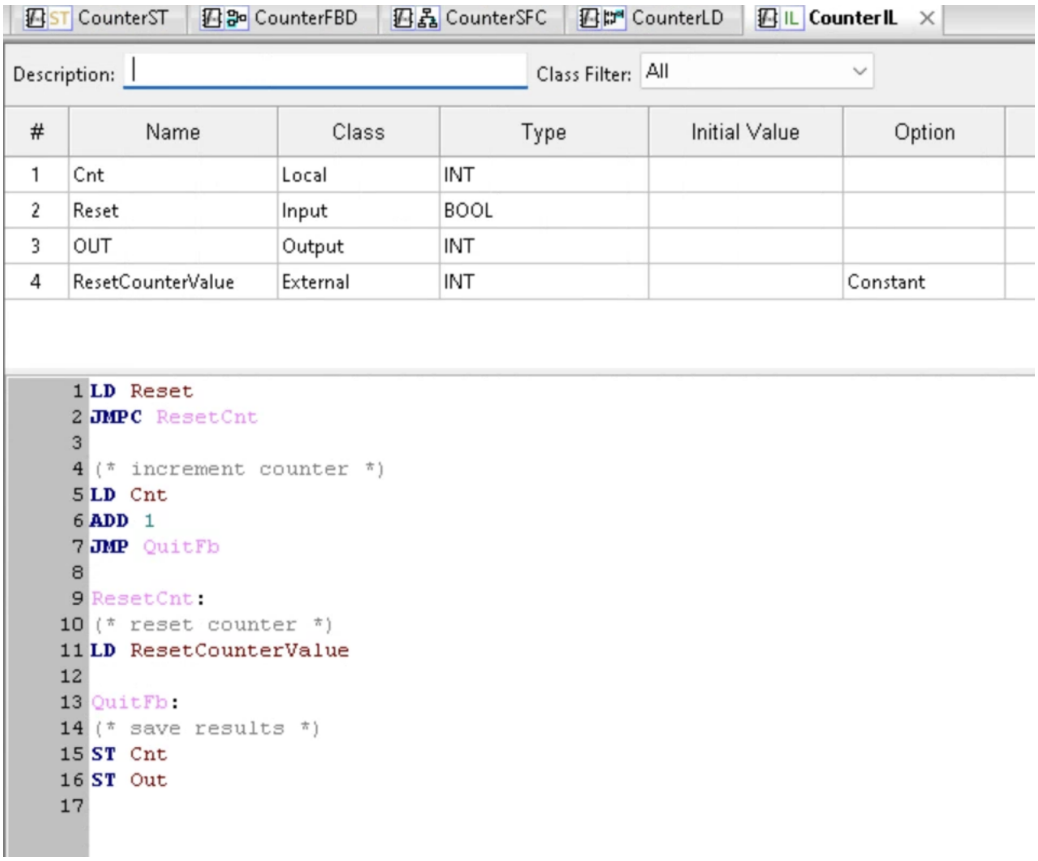

Figure 5 - An Instruction List Program in OpenPLC

The OpenPLC software supports a variety of low-cost hardware platforms, an abbreviated list follows.

- Arduino Uno / Nano / Leonardo / Micro / Mega / Due / Nano Every / IoT / BLE / RB2040 / MkrZero / WiFi / Pro
- Productivity Open P1AM [6]
- ESP8266, ESP32 [7]
- Raspberry Pi 2 / 3 / 4 / Zero
- Windows or Linux (generic target as a soft-PLC)
- Other platforms are also supported

The author has used the software with multiple platforms. One included the $15 Raspberry Pi Zero 2W with the OpenPLC Runtime software installed on Linux. The non-Windows/Linux systems, like the Arduino Uno, have less functionality for communication but they are still capable of limited MODBUS communication [8] over serial connections.

The Hardware

PLC hardware can be expensive and easily reach thousands of dollars per station for reasonable complexity for teaching. A number of manufacturers do offer educational discounts, but the cost is still high. But, for an introductory course, it is possible to use very low-cost options. For example, an Arduino Uno board can be purchased for less than $20 and allow a user to actuate pushbuttons and activate LEDs.

The author has chosen to use a platform that is more industrial in nature. However, it is based on open-source hardware, the P1AM-100 [8], and the hardware is produced by a common PLC manufacturer, Automation Direct [9]. In Figure 6, the P1AM processor is the third card from the left with the USB programming cable attached. All the PLC hardware shown in Figure 6 was purchased from Automation direct for approximately $500. The cards to the right of the CPU are standard PLC cards that are normally used with the industrial PLC 1000 productivity series. The cards to the left are essentially Arduino shields and not meant to work with regular industrial PLCs. The P1AM-100 is based on an Arduino MKRZERO.

The OpenPLC software provides access to the cards to the right using special functions to read and write to variables. But the OpenPLC software does not currently offer access to the cards to the left. Figure 7 shows an example of ladder logic to use the hardware. The first function ‘P1AM_INIT0’ initiates the communication with the cards. The ‘P1_16CCRD0’ function polls inputs and updates outputs for a combination card with 8 inputs and 8 outputs.

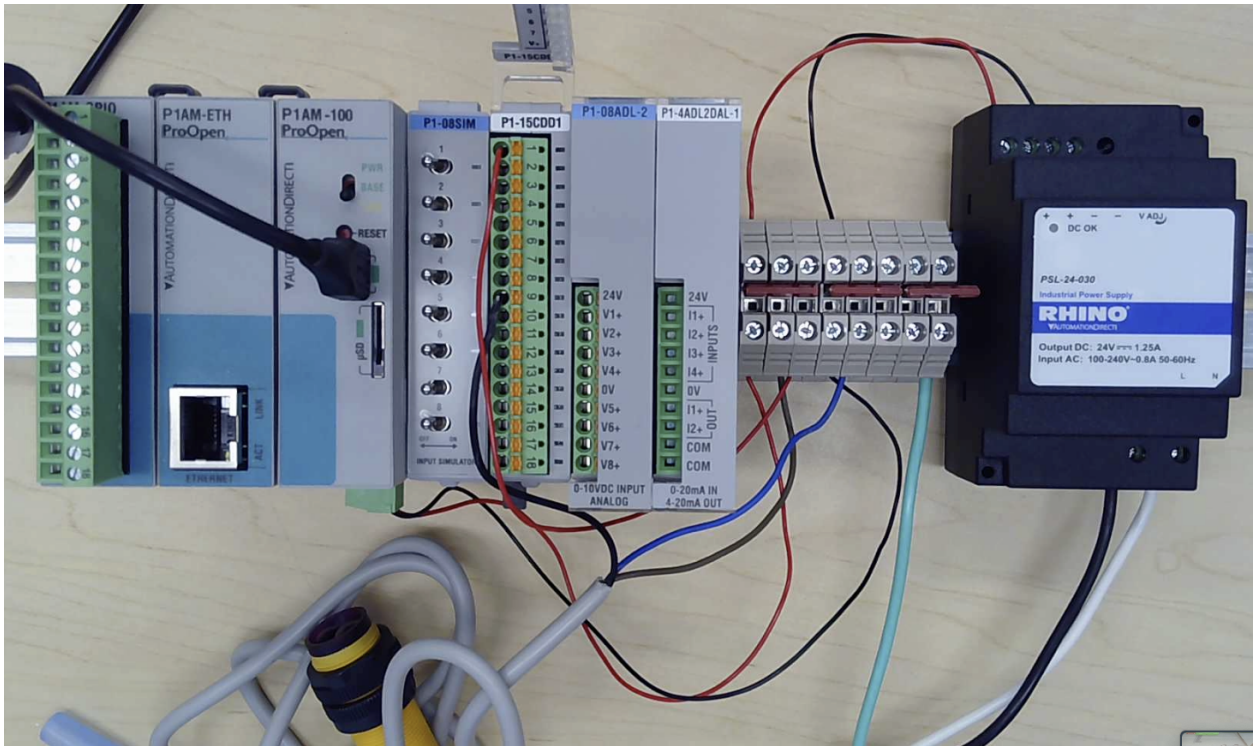

Figure 6 - Basic PLC Assembled on a DIN Rail

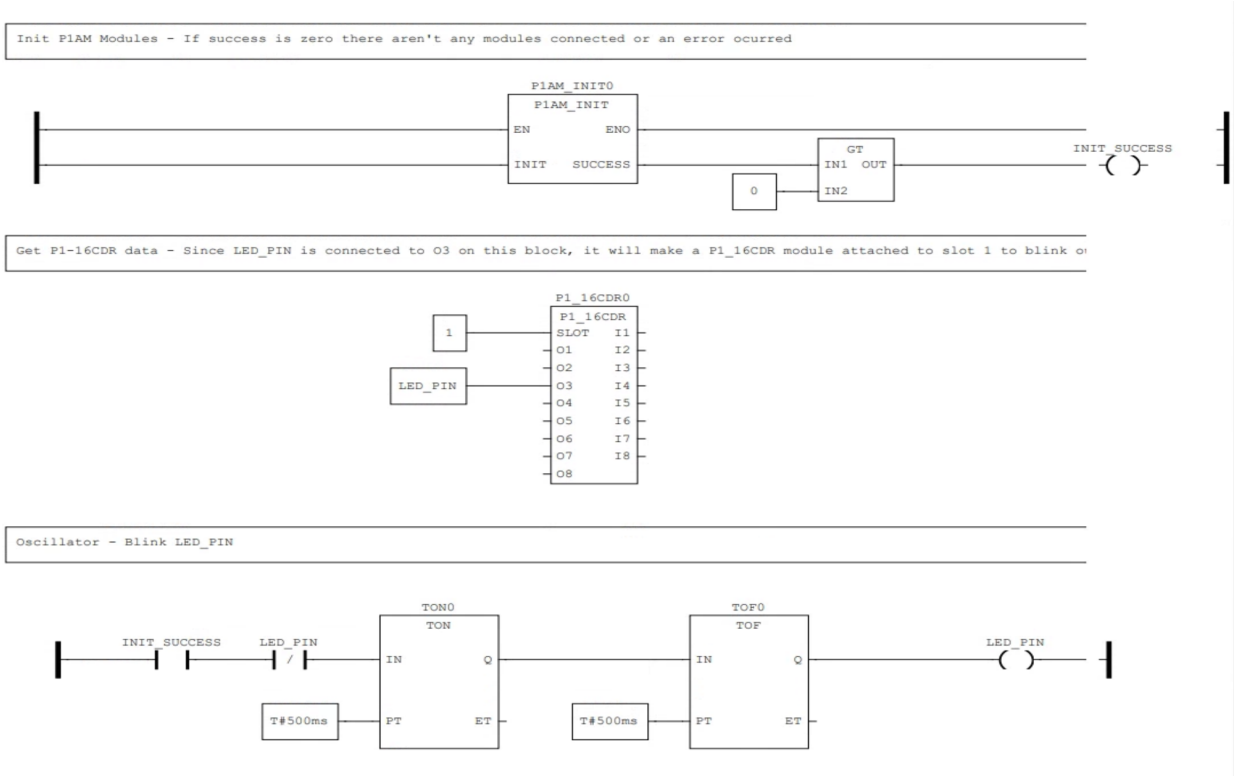

Figure 7 - A Ladder Logic Program Using a PLC Card

Another good alternative is to use the Arduino IDE to program the system in C, which gives access to all of the cards and functionality, including ethernet. However, C programs make software development much more complex, adding demands for liveliness and non-blocking operations. Students must be taught to avoid loops and functions that wait. For the purposes of the course, that topic was pushed to the end of the semester. Students have a following course in microcontroller hardware and programming that will expand their knowledge and skill with C programming.

Overall, the OpenPLC software and the chosen hardware platform, P1AM-100, offer a cost-effective solution for teaching PLC concepts. While students may face challenges in programming with C, the course structure allows them to gradually develop their skills with subsequent courses focused on microcontroller hardware and programming.

Conclusions

The use of free software and low-cost hardware went very well. For about $500 per station, students were able to use real hardware with high-end programming languages. Regular industrial hardware and software that supports the full suite of IEC 61131 programming languages would have cost multiple thousands per station. Students were able to use the OpenPLC software effectively on their own laptops to program the PLC hardware. The lack of a real-time debugger was surprisingly useful, as it encouraged students to review their programs thoroughly rather than making semi-random changes and frequently recompiling.

By the end of the course, the students were exposed to the concepts required for an introduction to computers course, albeit from a non-traditional perspective. And, students were better prepared to support industry and thrive in their careers.

A benefit of introducing PLC content in the second year is that students can leverage this knowledge to secure summer internships in the field, ultimately satisfying industry demand in the long run. While it is still too early to formally assess the effectiveness of this approach, anecdotal evidence suggests it is promising.

A second advantage of PLC knowledge is for use in the following courses. The school has a strong Project Based Learning (PBL) core. The courses blend students from Engineering Technology and Engineering disciplines including Mechanical, Electrical, Computer, and Manufacturing specialties. There is a dedicated multi-disciplinary PBL course in each of the first three years. In the fourth year, there is a 2-semester capstone project course. The courses introduce project management, teamwork, design skills, communications, and more. The courses in the first two years focus on the fundamentals with projects that have some elements of each

discipline but do not require much technical depth. In the third year PBL course, the project is an innovative product design. The teams are comprised of multiple disciplines and students are expected to apply discipline knowledge and skills to execute designs from customer needs to working prototypes. The fourth-year capstone project is done for industry and normally comprises new product design, test equipment, production equipment, or product/production research.

References

[1] “OpenPLC Open Source PLC Software”, https://openplcproject.com/, accessed Feb., 28, 2023.
[2] “Arduino - Home”, https://www.arduino.cc/, accessed Feb., 28, 2023.
[3] “Raspberry Pi Foundation”, https://www.raspberrypi.org/, accessed Feb., 28, 2023.
[4] Jack, H., Automating Manufacturing Systems with PLCs, Lulu, 2015
[5] "IEC 61131-3:2013". International Electrotechnical Commission (IEC), September 2021.
[6] “P1AM-100 - Documentation”, https://facts-engineering.github.io/modules/P1AM-100/P1AM-100.html, accessed Feb., 28, 2023.
[7] “Espressif Hardware” , https://www.espressif.com/en/products/hardware, accessed Feb., 28, 2023.
[8] "Modbus Application Protocol V1.1b3" (PDF). Modbus. Modbus Organization, Inc., https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf, accessed Feb., 28, 2023.
[9] “Automation Direct”, https://www.automationdirect.com/adc/home/home, accessed Feb., 28, 2023.