Introducing PLC Programming in the Digital Logic Course

Firas Hassan
ECCS Department
Ohio Northern University
Ada, Ohio 45810
Email: f-hassan@onu.edu

Heath LeBlanc
ECCS Department
Ohio Northern University
Ada, Ohio 45810
Email: h-leblanc@onu.edu

Michael J. Rider
Department of Mechanical Engineering
Ohio Northern University
Ada, Ohio 45810
Email: m-rider@onu.edu

Juliet K. Hurtig
Academic Affairs Office
ECCS Department
Ohio Northern University
Ada, Ohio 45810
Email: j-hurtig@onu.edu

Abstract—This paper describes a Programmable Logic Controller (PLC) lab component that has been added to the last three weeks of a digital logic semester-based course. First students work through an introductory lab on PLC programming to introduce them to the PLC hardware, the PLC programming environment and the basics of ladder logic. Next, students are given a request for proposals by a fictitious local government who is interested in installing and controlling a four-way traffic light. Students work on the PLC program to implement different traffic light modes, wire a traffic light prototype to the PLC, find the specifications of a typical PLC that can do the job, and write a reply to the request for proposals that includes the following sections: introduction, design details, prototyping and testing results, specifications and bill of material, budget and justification including hourly rate, estimated delivery time and conclusion.

1. Introduction

Digital logic is a foundational course typically offered to electrical engineering, computer engineering and computer science students early in the curriculum. This class typically covers the analysis and design of combinational and sequential circuits. The laboratory portion of the course usually includes design of digital systems using standard combinational and sequential modules. The designed digital systems are described using a hardware description language such as VHDL, simulated using Computer-Aided Design (CAD) tools such as ModelSim, synthesized using CAD tools such as Xilinx, and verified/tested on FPGA boards.

Programmable Logic Controller (PLC) programming is typically covered within a control systems course or an elective later in the curriculum, if at all. Feedback from industrial partners has identified a need to introduce PLC programming as early as possible in the curriculum. For students who are seeking summer internships or co-ops in industry, their training experience will be richer if they have the necessary background in PLC programming beforehand.

Although introducing PLC programming in a digital logic course is unusual, it has been done very smoothly in this course due to the similarities between the two topics. It also benefits students who develop a digital system that interacts with sensors and actuators. A PLC program

Proceedings of the 2018 ASEE North Central Section Conference

can be modeled using Finite State Machines (FSM), which is one of the topics that is traditionally covered in the digital logic course.

In this paper, we discuss a course activity that can be added to the end of the digital logic course to introduce PLC programming. The course activity requires only three laboratory periods. The paper summarizes in Section 2 the laboratory and testing equipment needed to perform the course activity. Section 3 provides the help tips that should be given to the students to complete their tasks successfully. Section 4 describes the different tasks performed by the students in the three different laboratory periods. Section 5 proposes a grading policy for the activity. Finally, some concluding remarks are given in Section 6.

2. Laboratory and testing equipment

Our PLC lab is equipped with a line of PLCs from AutomationDirect.com's Direct Logic series [1]. Each station in the PLC lab has the following equipment:

- DL-405 series, D4-450 CPU 110/220V
- H4-ECOM Ethernet Communication Card for communicating with the programming computer
- D4-16SIM Simulator Card with 16 switches
- D4-16ND2 12-24VDC Digital Input Card
- D4-16TR Relay Output Card that can be used with either AC or DC voltages

Each station also has a Windows PC with the PLC programming software DirectSOFT 6, which allows for programming and monitoring of the PLC and its I/O.

For the project, students are given an emulator of a four direction traffic light, shown in Figure 1. Each side of the traffic light emulator has three Light Emitting Diodes (LEDs) with red, green and yellow colors. All LEDs have their anodes connected together. The cathodes of the three different LEDs on both sides of the south/north and east/west traffic direction are also connected together.

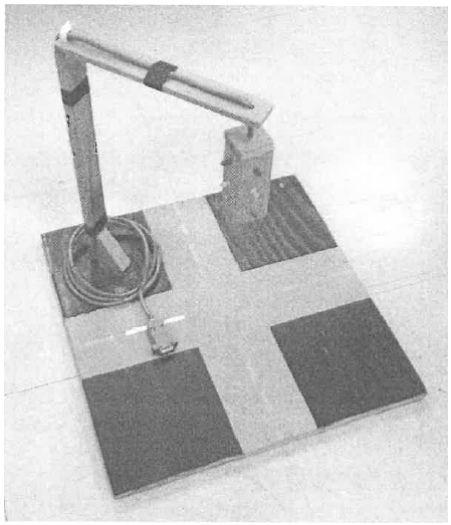

Figure 1. Four way traffic light emulator

3. Help tips

The following help tips are given to students before the introductory laboratory:

To create a new project in DirectSOFT 6, double-click on the "DirectSOFT 6 Programming" icon under Applications. Select the correct family and CPU type as shown in Figure 2. Also, name the new project, then pick the OK button.

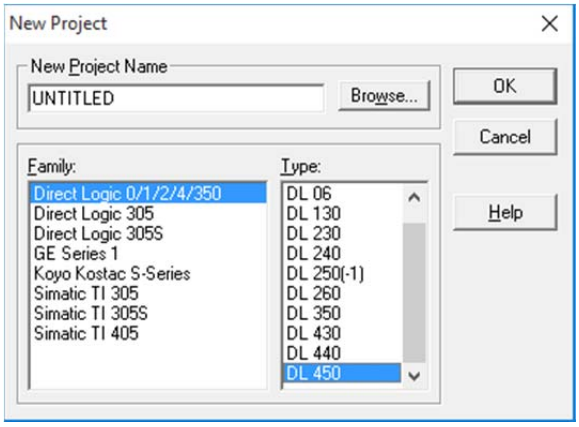

Figure 2. How to create a new project in Direct SOFT 6

The programming environment is initially in View Mode. Figure 3 shows an example with some ladder logic that explains much of the user interface:

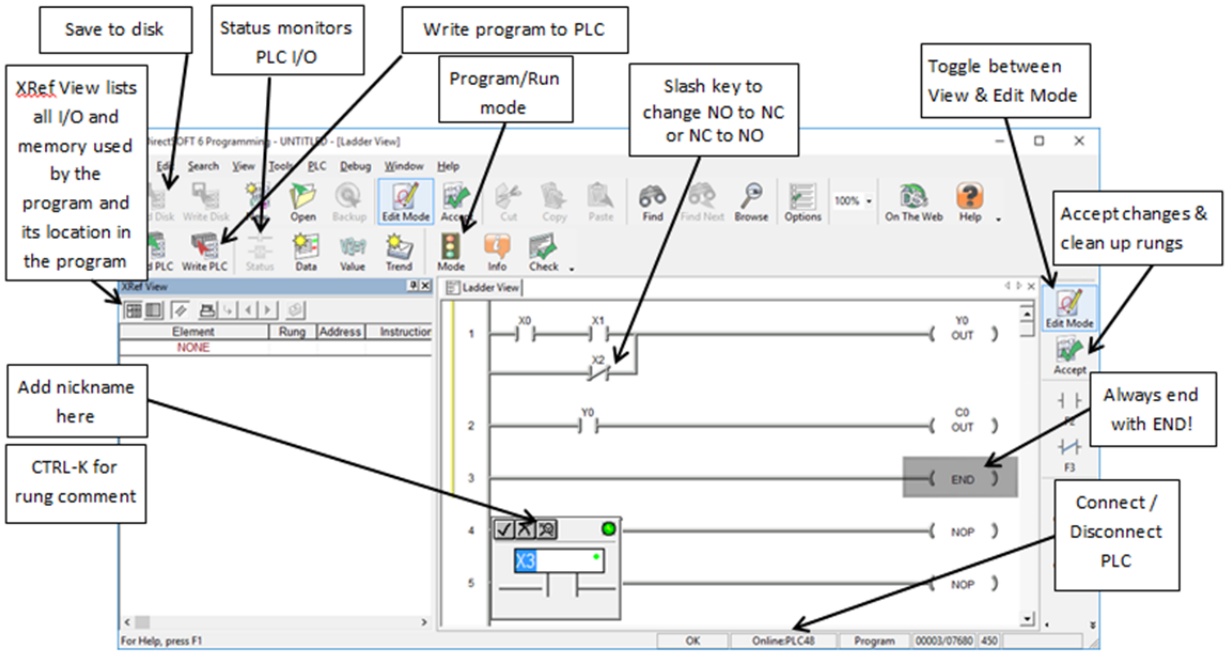

Figure 3. An example with some ladder logic that explains much of the user interface

Each PLC is labeled with an address. Under Comm Links in the DSLaunch window find the link with that address and double-click on it to connect. Alternatively, if you already have code you can use PLC->Connect, choose the correct link, and select "use disk". Note that there are two memory spaces you are working with: the PLC memory and the PC memory. When uploading code to the PLC, the PLC should be in program mode with the keyed selector on the PLC set to Terminal. Once the code is uploaded, activate it on the PLC by changing to Run mode by using the traffic light icon. Turn on Debug mode by clicking on the Status icon, which allows you to see which contacts and coils are energized so you can see the code that is being executed in the PLC. Table 1 below is given to the students summarizing the wiring specification, as well as the PLC outputs that are used to control the LEDs of the traffic light emulator.

Table 1. Wiring specification of the traffic light

	Wire	**Connection**	**Reason**
	Blue	+24 VDC	needed all the time
	White/Blue	NC	not used
Y10	Orange	Ground	Red light (N-S)
Y11	White/Orange	Ground	Red light (E-W)
Y12	Brown	Ground	Yellow light (N-S)
Y13	White/Brown	Groun	Yellow light (E-W)
Y14	Green	Groud	Green light (N-S)
Y15	White/Green	Grond	Green light (E-W)

Students are also given a detailed hand sketch of the required connections between the PLC outputs and the control wires of the LEDs, as shown in Figure 4.

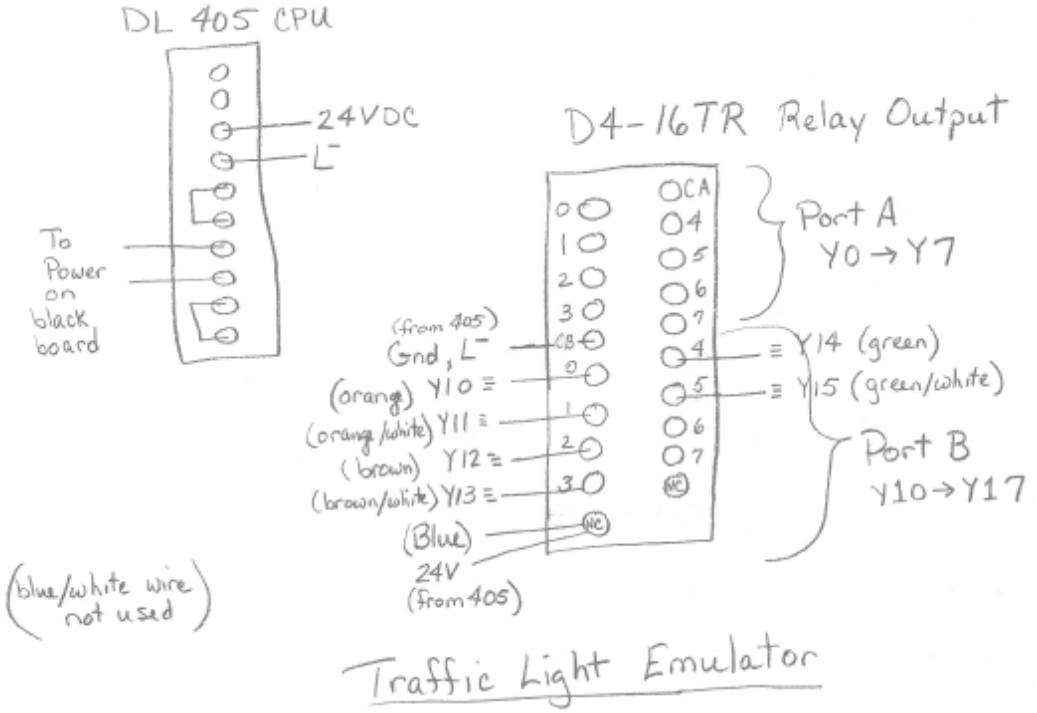

Figure 4. Hand sketch of the wiring diagram

Finally, students are advised to

- Use the "Nickname" feature of DirectSOFT 6 to label all contacts and coils with realistic names based on their purpose.
- Comment each rung of your ladder logic program (CTRL+k) with meaningful phrases.
- Have the PLC rack and power supply unplugged when wiring the cable to connect to the traffic light.
- **Introductory Laboratory and Project Tasks**

In the introductory laboratory students are given the following tasks to help them become familiar with the programming environment of PLCs:

- Using X0 and X1 as inputs and Y0 as output, program the PLC to emulate the following gates:
- 2 input And gate
- 2 input OR gate
- 2 input NAND gate
- 2 input NOR gate
- 2 input XOR gate
- 2 input XNOR gate
- 1 input invertor.

1. Program the PLC to flash the Y0 output between on and off every 1 second.

2. Program the PLC to create a traffic light controller that cycles through the following sequence (shown in Table 2 below).

Table 2. The normal sequence of the traffic light emulator

time	Y0	Y1	Y2	Y00	Y01	Y02
	NS Red	NS Yel	NS Grn	EW Red	EW Yel	EW Grn
1 sec	1	0	0	1	0	0
7 sec	0	0	1	1	0	0
2 sec	0	1	0	1	0	0
1 sec	1	0	0	1	0	0
7 sec	1	0	0	0	0	1
2 sec	1	0	0	0	1	0

All these tasks are done in one three hour lab period.

For their project assignment, students are given the following request of proposals:
A town purchased a traffic light to help prevent collisions at a given intersection. Your job as a control engineer is to pick the controller for the traffic light and design its ladder logic program.

Your prototype implementation will use the in-house DL 405 PLC unit. Due to the traffic light's location, several different lighting patterns must be programmed.

Inputs True/ON	Description
X0	Standard cycle through lights every 20 seconds (Red – 11 seconds, green – 7 seconds, yellow – 2 seconds)
X1	All four directions see flashing red only (1 second ON, 1 second OFF)
X2	All 4 directions see red only (always ON)
X3	North-South see (1 second ON, 1 second OFF) flashing red East-West see (1 second OFF, 1 second ON) flashing yellow
X4	North-South see (1 second ON, 1 second OFF) flashing yellow East-West see (1 second OFF, 1 second ON) flashing red

There is a priority on the inputs such that X1 takes precedence over X2, which takes precedence over X3, which takes precedence over X4, which take precedence over X0.

Note: You are responsible for making the system fail-safe….that is, what happens if more than one switch is thrown? Or if no switches are thrown? Include with your PLC ladder logic a brief description (several sentences) describing how your code achieves the requirement of being fail-safe.

The tasks of the project are then divided into two weeks as follows:
Tasks for week 1:

1. Download the traffic light controller project that was created in the introductory laboratory.
2. Edit the program to run a traffic light with the standard mode. N.B. you mainly need to edit the names of the outputs.
3. Connect the cables to the PLC following the wiring diagram.
4. Test the code on the available traffic light in the lab.
5. Start working on the remaining tasks of the project.

Tasks for week 2:

1. Finish working on the project and test it.
2. Find the specifications for the best possible PLC unit and extra input/output modules that can do the job.
3. Write a bill of materials (BOM) for everything needed.
4. Write a report that will answer the customers' request for proposals and include the following:
   - Design details
   - Prototyping and testing results
   - Specs and BOM
   - Budget including your hourly rate.
   - Delivery time

5. Grading policy

As a whole, all PLC related activities are done in three laboratory periods. Students are divided into groups of two. After demonstrating all the tasks of the introductory laboratory and project successfully, each group submits one report. The grade of the final report is divided based on the different report sections using the following percentages, shown in Table 3.

Table 3. Grading percentages of the final report sections

Intro.	Design details	Prototyping & testing results	Specs & BOM	Budget including your hourly rate	Delivery time	Conclusions
5	35	30	10	10	5	5

6. Conclusion

In this paper we have shown that ladder logic and PLC programming naturally fit in a Digital Logic course. The proposed project introduces students to industrial control concepts early in the curriculum. We can also claim the following added advantages of having this project in the digital logic course. First, it enhances students' curiosity in current technical trends. The main purpose of the project is to introduce students very early to PLCs, which is extensively used in industry. Second, it allows students to understand the ramifications of design decisions. The fact that students are controlling a traffic light that could cause accidents if it is wrongly controlled exposes students to the ramifications of their design decisions. Third, it allows students to create solutions that meet customer needs. Students begin with customer needs and must find solutions for them. Fourth, it teaches students to collaborate by forming and working in teams. Students work with multidisciplinary groups including computer science, electrical and computer engineering students. Fifth, it improves students' communication skills by producing effective written reports. All groups are required to respond to a request of proposals. Finally, because the project has a restricted deadline, it requires students to meet this commitment. All these added advantages target the entrepreneurial mindset [2, 3] of engineering graduates.

References

[1] R. J. Michael , "Principles of the DL405 Family Programmable Logic Controllers", 10th edition, ONU Print Shop, 2017.
[2] D. E. Melton, "Staking Entrepreneurially Minded Learning Alongside Other Pedagogies" , KEEN'zine –Intrapreneurship Edition, http://www.keennetwork.org/, 2015.
[3] A. L. Gerhart, "Entrepreneurially Minded Learning: Incorporating Stakeholders, Discovery, Opportunity, Identification, and Value Creation into Problem-Based Learning Modules with Examples and Assessment Specific to Fluid Mechanics", Proceedings of the 123rd ASEE annual conference and exposition, 2016.