Real World Learning in Distance Education

John C. Anderson, PE
Mechanical Engineering Technology Dept., Purdue University,
1417 Knoy Hall, W. Lafayette, IN 47909-6219
Ph. (765) 494-7526 email: jcanderson@tech.purdue.edu

Abstract

Obtaining and maintaining laboratory equipment in technology courses is a constant problem. This paper proposes an alternative to the use of simulation or to buying laboratory equipment for a school laboratory, and uses a course in Programmable Logic Controllers (PLC) to illustrate the concept.

With the reduction in pricing of electronics equipment and Internet direct sales companies servicing the industrial automation market, it is possible to purchase industrial grade automation equipment at very low prices. This allows students to purchase their own equipment to perform the experiments necessary to learn PLC applications and programming.

I. Introduction

The problem of obtaining and maintaining laboratory equipment in technology courses is an old and familiar issue. In areas such as that of manufacturing automation this problem is compounded by rapidly changing equipment and standards.

In addition there has been a great deal of attention paid to the subject of making technology labs more flexible in terms of capital expenditure and to facilitate distance learning. Much of this work has been focused on virtual laboratory software, that is, simulations of the actual equipment. Although simulations do address the issues of distance learning and to a large extent, the problem of capital expenditure on equipment with a relatively short useful life, it’s use can cause a loss of some of the practical knowledge that a student picks up in actually working with the equipment.

Simulations can leave some fairly large gaps that hinder the students understanding of actual commercial installations. In the case of discrete event controllers, such as Programmable Logic Controller (PLC’s) simulators don’t support the development of the concepts of peripheral equipment and an understanding of how the Input and Output (I/O) devices interface with the PLC.
Educators have wrestled with this problem for many years, but current technology offers some new avenues of approach. In particular the continued developments in electronics have lowered the cost of components and the availability of the Internet offers the opportunities for students to interact in a timely fashion, not only with their instructors, but also with equipment manufacturers.

In this paper a Programmable Logic Controllers (PLC) course is used as an example of this approach. In a traditional course the student would travel to the instructor for lecture sessions and to a laboratory for access to equipment to practice the principles learned in the lecture session. The costs to the student would include travel, textbooks, lab manuals, and tuition.

As an alternative approach the student could purchase a commercial level PLC and programming software, download the manufacturers operating manuals from the Internet, and participate in lecture sessions either on the Internet or in a classroom setting using their own equipment. Manufacturers literature combined with the lecture presentation and notes would substitute for the textbook. Recent reductions in the cost of PLC’s and new marketing approaches have allowed the cost to the student of this approach to be similar to the traditional approach.

The advantages of using an actual PLC are

- Allows the student to wire up the PLC to inputs and outputs,
- Allows the student to program using an actual commercial version of the PLC programming software,
- Allows the student to use the PLC in actual applications after the class.

In addition this greatly facilitates distance learning in a subject that has been traditionally tied to expensive laboratory facilities.

II. Equipment

For this project a Model DL05DD PLC manufactured by Koyo and distributed by Automationdirect.com is used (Fig. 1). This PLC is part of a broad family of PLC’s that use the same programming interface, so that the programming language for the DL05 is a subset of the language used for the top of the line DL405 and has the same programming screens and file organization.

The great advantage of the DL05 for teaching is the size and the price ($99). The PLC offers the following features;

- 8 Input points (10 to 26 volt input for the model selected)
- 6 Output points (5-30 VDC, .6A)
- External power 10-30 VDC (allows safe operation by students)
- Internal high speed counter and output pulse generator for motion control applications (counting pulses from an encoder or driving stepper motors)
- Drum switch instruction set (allows the PLC to simulate a mechanical drum switch for sequential operations)
Both relay ladder logic and functional block styles of programming.

**Figure 1 – Automationdirect DirectLogic 05 PLC**

The input devices chosen are momentary closure pushbutton switches available from Radio Shack (Figures 2 & 3). Momentary closure pushbuttons are used because they are the most common input devices encountered in the field, and they can be used to simulate other momentary closure inputs, such as limit switches. The Radio Shack switches are available with either black or red actuators (push buttons) that are two of the most common actuator colors encountered in the field.

**Figure 2 PB Switches**

**Figure 3 PB Switch Detail**
The output devices are incandescent pilot light fixtures also available from Radio Shack. These have lenses that are available in a number of colors to simulate industrial pilot lights. An interesting series of experiments in programming PLC’s is programming traffic light situations using red, green, and amber lenses on the lights. This can utilize not only relay replacement instructions, but also timers, counters, and drum sequencer instructions.

Figure 4 – Pilot Light w/ Red Lens

Although this PLC is available in versions powered by 115 VAC, a model powered by DC voltage (10 to 30 VDC) was selected. A small sealed power supply that converts 115 VAC to 12 or 24 volts DC is used to power the PLC and the output devices (pilot lights). This provides an intrinsically safe circuit for students to work with and 24 VDC control voltage is an accepted standard in industry.

In addition to the above, one could round out a student equipment kit by including a small section of DIN rail to mount the PLC and power supply, a small control box (available from Radio Shack) to simulate a pushbutton panel, a terminal strip, and wires.

### Parts Cost Summary For PLC Trainer

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Cost</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koyo PLC, Model DO-05</td>
<td>1</td>
<td>$99.00</td>
<td>$99.00</td>
</tr>
<tr>
<td>DiretSOFT Programming Software</td>
<td>1</td>
<td>$99.00</td>
<td>$99.00</td>
</tr>
<tr>
<td>12 to 24 volt Sealed Pwr. Supply</td>
<td>1</td>
<td>$25.00</td>
<td>$25.00</td>
</tr>
<tr>
<td>Toggles Switches</td>
<td>8</td>
<td>$1.00</td>
<td>$8.00</td>
</tr>
<tr>
<td>Pilot Lamps</td>
<td>6</td>
<td>$2.00</td>
<td>$12.00</td>
</tr>
<tr>
<td>Miscellaneous (DIN Rail, Enclosure, etc.)</td>
<td></td>
<td>$10.00</td>
<td>$10.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$253.00</strong></td>
</tr>
</tbody>
</table>

Although the price of the parts for the trainer seem high at first glance they must be balanced against the costs a student would incur in a traditional format class. The student may forego purchasing a textbook, thus saving approximately $80. Typically there is a lab fee that is charged which might be as much as $25. There is also the convenience of not having to
come to a laboratory and the travel costs. These partially offset the cost of materials to the students. Another advantage to the student in this type course, as opposed to the use of a computer based simulation, is that the PLC is an actual industrial grade PLC and can be used in other applications, and will have a resale value.

III. Course Materials

The information available from manufacturers on the Internet allows some new approaches to the course reference materials supplied to the student.

Traditionally a text is used that provides the bulk of the material that the student needs to work through the course. This includes descriptions of the physical installation (wiring, safety, construction of input and output devices), examples of catalog descriptions of parts, descriptions of programming instructions, as well as programming techniques and strategies. Much of this material is available at the Automationdirect.com1 web site (Figure 5)

Figure 5 – Contents of Automationdirect Web Site

The site includes marketing oriented information such as product brochures and selection guides, as well as technical information. An excellent resource is the DL05 User Manual. The manual is on the web site in PDF format and can be downloaded to a local disk drive, or stored on a CD. As shown in the table of contents below (Figure 6), the manual covers many of the topics normally covered in a text book.

For example, the manual explains the concept of the CPU scan (Figure 7) and how that relates the program to the physical state of the inputs and outputs. The explanation and graphics are very good and can be augmented by lecture, or a computer based presentation. The manual also offers specific wiring instructions and practical tips on wiring the PLC system in adverse conditions.
Figure 6 - Automationdirect.com DL05 Table of Contents

Chapters

1. Getting Started
   The main contents of this manual are organized into the following eight chapters:
   - Introduces the six models in the DL05 Micro PLC family and the available I/O types. Also includes tips on getting started and how to design a successful system.

2. Installation, Wiring, and Specifications
   - Shows how to prepare for system installation, and gives safety guidelines to help protect your personnel and machinery. It includes system and I/O wiring diagrams, including specifications for each of the six DL05 versions.

3. High-Speed Input and Pulse Output Features
   - Teaches you how to configure the High-Speed I/O operation to provide counting functions, motion control profiles, input filtering, interrupt capability, and much more. We recommend studying Chapters 2 and 3 together.

4. CPU Specifications and Operation
   - Explains how the DL05 internal CPU controls the PLC system operation. This includes information on I/O point updates, application program execution, and memory structure.

5. Standard RLL Instructions
   - Describes how each of the DL05's standard instructions operate. Example programs for both DirectSOFT and the Handheld Programmer are included.

6. Drum Instruction Programming
   - Describes the powerful drum sequencer operation, including detailed theory of operation and control techniques. It reveals how drums solve sequential processes problems.

7. RLL Plus Stage Programming
   - Shows how to develop state transition diagrams for your process and convert them to stage programs. This method is a real time-saver for programming and debugging.

8. PID Loop Operation
   - Explains how the DL05 PID Loop Capability can control up to 4-loops. Discusses several topics, including PID variable data, loop tuning, alarms, ramp/leak profiles, and more.

9. Maintenance and Troubleshooting
   - Is a guide designed to aid you in diagnosing, repairing and avoiding system problems. It includes tips for finding I/O circuit problems, ladder program errors, etc.

10. F0-04AD-1 4-Channel Analog Current Input
    - Describes the use of the F0-04AD-1 analog option card. It includes a wiring diagram, information about ladder logic necessary for using analog data, and new Special Relays.

11. Memory Cartridge / Real Time Clock
    - Shows how to transfer program from PLC to MC or vice versa. It explains how to set the time and date, set the jumper, and it provides the module's memory map.

Figure 7 - CPU Scan Example

- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
- The CPU reads the inputs from the local base and stores the status in an input image register.
- The CPU scans the inputs from the local base and stores the status in an input image register.
There are other resources available at the automationdirect.com website, in addition to the PLC manual, to enrich the students educational experience. In the marketing (Product Information) section there are case studies of contemporary applications of this particular PLC. These may be used as case studies, or examples, of actual successful implementations of the same type PLC that the student is using.

Since there is information on other devices at this web site there is also the opportunity to integrate problems in design and specification of entire systems. Automationdirect.com sells not only PLC's, but also a wide range of industrial controls, including contactors, relays, variable frequency AC drives, pushbuttons, and limit switches. Figure 8 is an illustration from the catalog section on switches and offers the student a real image to use in selecting components.

One example of the value of this broad range of information might be an assignment to automate a conveyor based material handling work cell. The assignment may include working up a material list of actual components (including the PLC). Since the students would be using an actual catalog this adds some realism to the exercise and offer familiarization to using catalogs.

Figure 8- Catalog Entry for Momentary Closure Pushbuttons

The web site also offers Cad files for many of their components allowing exercises to design and layout control panels.

Figure 9 below is an outline for a 3 credit hour course using the concepts proposed in this paper. This outline relies completely on the information contained in the Automationdirect.com web site to provide reference material. This must, of course be augmented with a strong set of lectures, preferably with published lecture notes.

The course, as outlined, could be functionally graded, ie, the students grades would depend on their lab results, or exams could be developed and inserted into the schedule. Since the
students have their own lab equipment, and assuming they have Internet access, the program files can be emailed to the instructor and functionally tested.

Figure 9 – Typical Syllabus for 3 Credit Hour PLC Course

<table>
<thead>
<tr>
<th>Lecture #</th>
<th>Topic</th>
<th>References</th>
<th>Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to PLC</td>
<td>DL05 Chap 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PLC Hardware</td>
<td>DL05 Chap 2</td>
<td>Plan and Assemble a PLC system</td>
</tr>
<tr>
<td>3</td>
<td>PLC Hardware contd.</td>
<td>DL05 pg 4-1 to 20 AD</td>
<td>Sensors &amp; Encoders</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Number Systems</td>
<td>DL05 pg 4-20 to 25</td>
<td>Test PLC Electrical System</td>
</tr>
<tr>
<td>5</td>
<td>Intro to PC / PLC software</td>
<td>DirectSOFT Quick Start Manual</td>
<td>Program 1</td>
</tr>
<tr>
<td>6</td>
<td>Relay Replace Commands</td>
<td>DL05 pg 5-1 to 25</td>
<td>Program 2</td>
</tr>
<tr>
<td>7</td>
<td>Relay Replace Commands</td>
<td>DL05 pg 5-1 to 25</td>
<td>Program 3 - Conveyor 1</td>
</tr>
<tr>
<td>8</td>
<td>Relay Replace Commands</td>
<td>DL05 pg 5-1 to 25</td>
<td>Program 4 - Conveyor 2</td>
</tr>
<tr>
<td>9</td>
<td>Programming Timers</td>
<td>DL05 pg 5-30 to 34</td>
<td>Program 5 - Delayed I/P</td>
</tr>
<tr>
<td>10</td>
<td>Timers contd.</td>
<td>DL05 pg 5-30 to 34</td>
<td>Program 6 - Traffic Light 1</td>
</tr>
<tr>
<td>11</td>
<td>Counters</td>
<td>DL05 pg 5-34 to 42</td>
<td>Program 7 - Counting I/P</td>
</tr>
<tr>
<td>12</td>
<td>Counters, contd.</td>
<td>DL05 pg 5-34 to 42</td>
<td>Program 8 - Traffic Light 2</td>
</tr>
<tr>
<td>13</td>
<td>Program Control</td>
<td>DL05 pg 5-96 to 102</td>
<td>Program 9 - Conveyor 3</td>
</tr>
<tr>
<td>14</td>
<td>Data Manipulation</td>
<td>DL05 pg 5-43 to 62</td>
<td>Program 10 - Conveyor 4</td>
</tr>
<tr>
<td>15</td>
<td>Math Instructions</td>
<td>DL05 pg 5-63 to 76</td>
<td>Program 11 - Traffic Light 3</td>
</tr>
<tr>
<td>16</td>
<td>Sequencers</td>
<td>DL05 Chap 6</td>
<td>Program 12 - Automated Machine 1</td>
</tr>
<tr>
<td>17</td>
<td>High Speed Counters</td>
<td>DL05 Chap 3</td>
<td>Program 12 - Automated Machine 2</td>
</tr>
</tbody>
</table>

IV. Conclusion

The approach described in this paper offers several important advantages over other traditional and non-traditional approaches.

First, it conserves capital resources for the educational institution by either shifting the lab to the student, or at least minimizing the cost and space necessary for a lab. If the school wished to provide a traditional lab, this equipment is small and light, and requires only 115 volt AC power. The cost per workstation is low (approximately $250 per student) and the PC used to program can be used for other purposes.

Second, it enriches the students experience by adding the dimension of specifying and assembling the input and output devices. The student also gains experience at using commercially available literature in the design process.
References

1. URL:http://automationdirect.com, Automationdirect.com web site

JOHN ANDERSON
John Anderson is an Assistant Professor of Mechanical Engineering Technology at Purdue University. He is a registered Professional Mechanical Engineer in Nevada and received B.S. and M.S. degrees in Mechanical Engineering from the University of South Carolina. Prof. Anderson is currently teaching in the areas of manufacturing automation and polymer manufacturing and is active in working with industry.