Session 1647

Web-based Programming Guide for Allen Bradley PLCs

James A. Rehg
Penn State Altoona

Abstract

Programmable logic controllers (PLC) concepts are taught in many Electronic Engineering Technology programs. However, the task is made more difficult by current industry practices. For example, every PLC vendor has a different programming environment, a different command set, and uses documentation that is not addressed to new users of the technology. This variation in PLCs is a difficult instructional problem to solve. However, this paper describes a web site developed to overcome the third problem, availability of good resource material for teaching laboratories equipped with Allen Bradley SLC 500 PLCs. The web site provides the following program development support: 1) an introduction to the SLC 500; 2) an overview of the command structure used in PLCs and the SLC 500 in particular; 3) discussion of the hardware interface for the SLC family of modules; 4) SLC 500 command reference and definitions accessible by symbol, name, or command group; 5) command syntax and example ladder logic programming applications; 6) introduction to programming; 7) introduction to development of human machine interfaces; and 8) laboratory exercises. The paper describes the development and use of this site in teaching introductory and advanced PLC laboratories.

Introduction

The programmable logic controller (PLC) has become the most frequently used computer for industrial control. Since its introduction in the early 70s, the technology has evolved from an electronic replacement for relay logic to a versatile shop-floor computer capable of handing a number of control problems. The discrete control solutions range from simple start/stop logic to systems including thousands of discrete signals. The closed-loop control applications include control of position and velocity, as well as a full range of continuous process parameters such as temperature, flow, level, and pressure. As a result of the rapid growth of PLC applications in industry, individuals with PLC ladder logic program design, implementation, and system troubleshooting skills are in high demand by manufacturers using the PLC technology and by systems houses doing turn-key system development. Recognizing the need for graduates with PLC experience, two- and four-year engineering technology programs across the United States have added PLC courses and laboratories.

The issues associated with the development of a PLC course and laboratory include: absence of a standard for PLC languages, dearth of vendor-specific instructional materials, cost of the student laboratory stations, breadth of the applications, and magnitude of the learning requirements. The first two issues, PLC language standards and the scarcity of good PLC instructional materials, are addressed in this paper.

Teaching PLC Programming Languages

The absence of a standard for PLC languages creates a number of problems when instruction in PLC concepts is planned. Two options are available to overcome this issue: teach PLC programming using a generic set of PLC commands or select an industrial PLC and teach PLC programming using the command set from the selected vendor. The first option is often used in PLC textbooks so that the text appeals to a wide range of schools using different PLC systems in the laboratory. A major disadvantage to option one is that a simulator or virtual generic PLC is not available to test the generic programs. Therefore, students must still learn the language for PLCs used in the laboratory. In addition, it is difficult to develop a generic language that represents the more complex, higher-end commands because the approaches used by different PLC vendors are quite varied.

The second option, selection of a specific PLC language, is usually the technique chosen for PLC course development and is the choice for the Penn State Altoona courses. A major advantage of this strategy is that the language instruction delivered in the lecture and laboratory is focused on the same command set. As a result, both basic and advanced language commands can be covered giving students broader programming skills. In addition, the ability to use a specific PLC language enhances the graduate's opportunities in the job market. After a number of semester of experience using this approach, two problems associated with this strategy became apparent: the students learn a single PLC model from one vendor, and the availability of vendor specific instructional material targeted for beginning PLC uses is limited. While the first problem may seem severe, students taught with this approach report that learning a second set of PLC commands from a different vendor is not a problem[1]. In addition, many indicate that the learning curve on a different system is significantly shortened because of the initial PLC knowledge.

Experience in the Penn State Altoona courses indicates that overcoming the second problem presents a stiffer challenge. The amount of PLC programming and operational reference material available to the instructor and to students is a function of the selected vendor. The larger the market share of the PLC vendor, the greater the availability of training material. However, the vendor with the leading market share in the United States, Allen Bradley (AB), does not have a broad selection of good support documentation across the complete PLC command set. At the introductory level, textbooks provide the best support for Allen Bradley with little or no coverage of GE/Fanuc, Seimens/TI, Omron, Modicon, and Square D systems. For a discussion of the more advanced commands, the only material available is from the PLC vendors, and that material is often presented at a level that only experienced PLC programmers can understand.

The web-based programming guide for Allen Bradley PLCs described in this paper serves two functions: the guide is a detailed on-line reference available to schools using the AB SLC 500 model PLC in a course and laboratory environment, and the web-based site is a model for the development of programming guides for PLCs from other vendors.

PLC On-line Programming Guide

When fully implemented, the web site will provide the following program development support: 1) an introduction to the SLC 500; 2) an overview of the command structure used in PLCs and in the SLC 500; 3) a description of the SLC 500 hardware interface; 4) an SLC 500 command reference accessible by symbol, name, or command group; 5) a reference for command syntax and example ladder logic programming applications; 6) an introduction to programming, 7) an introduction to development of human machine interfaces; and 8) laboratory exercises. All of the site elements will be activated in the Spring 1999 semester except number 7 development of a human machine interface. The site, located at *http://www.courses.psu.edu/Altoona/fall1998/e_met/e_met430.001/main.html* uses the two-frame opening screen displayed in Figure 1.

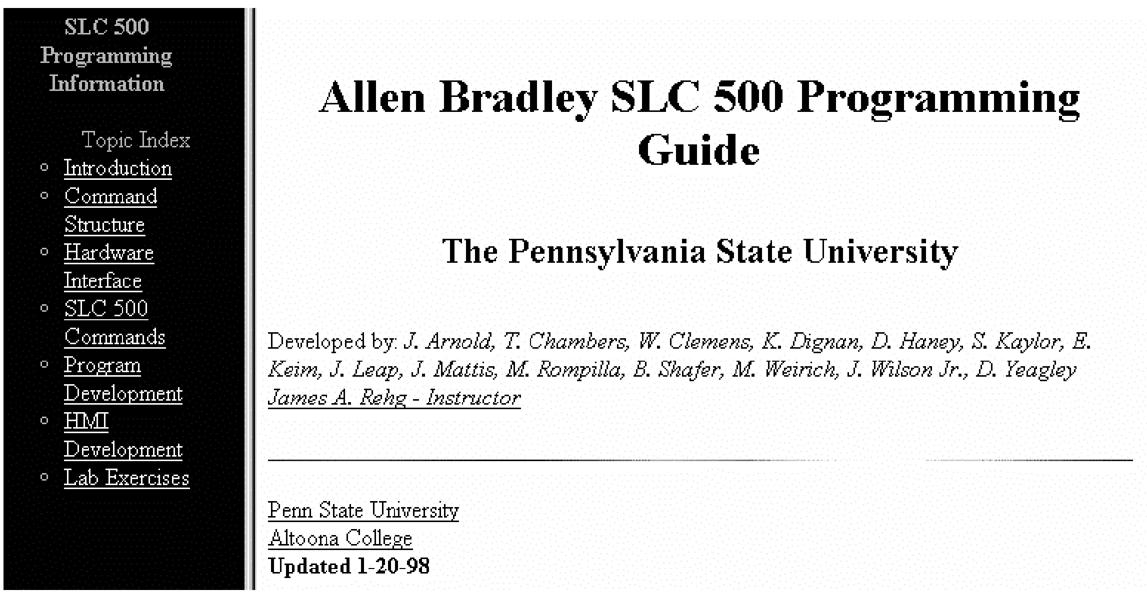


Figure 1 PLC Programming Guide Splash Screen

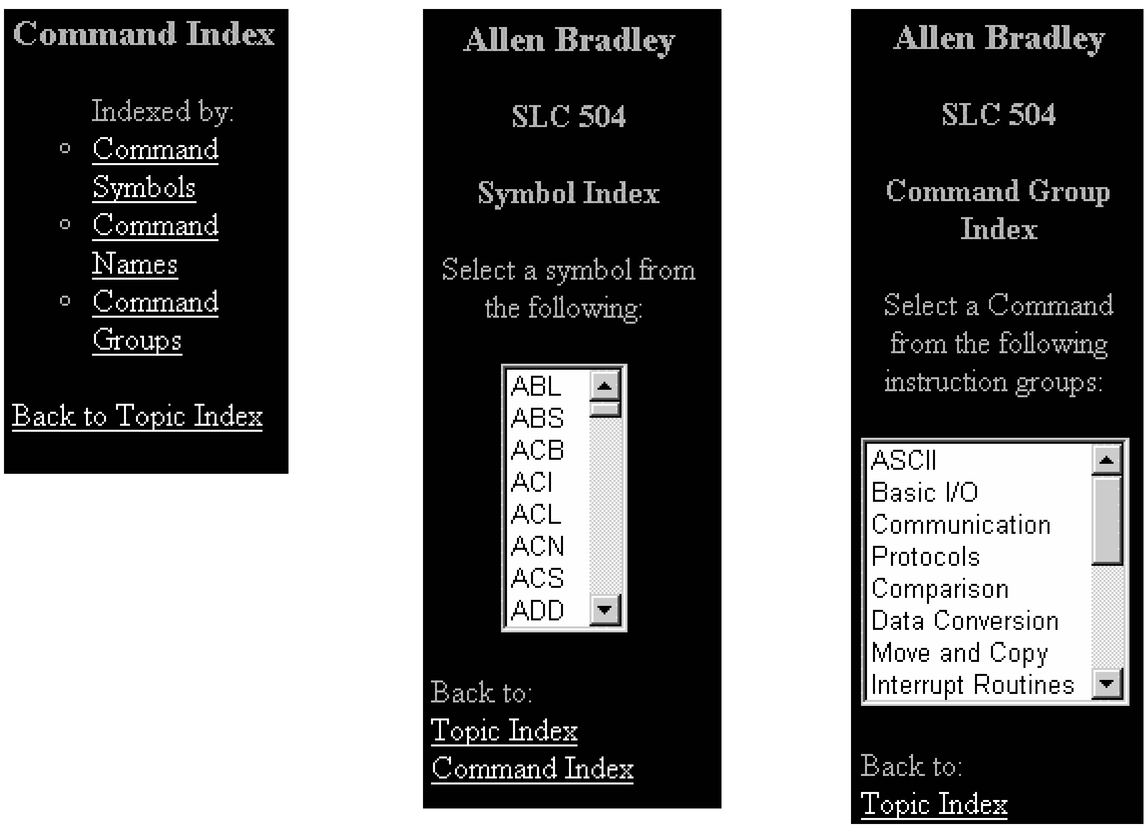
Brief descriptions of the menu elements in the left frame are provided below. Example screens are included for all completed sections of this developing web site.

Introduction to the SLC 500 - This section provides an introduction to PLCs in general and the SLC family of controllers in particular.

Overview of the command structure used in the SLC 500 - This section provides an overview of the SLC 500 family command and programming structure, a description of the variables, assignment of variables to command functions, and an overview of the Allen Bradley Logics 500 and Links Windows programming environment.

Description of the SLC 500 hardware interface - This section provides a description of the SLC 504 and 505 PLC hardware, including the processor and all the basic modules used with the system. In addition, the interface hardware used in the Penn State Altoona laboratories is also covered.

SLC 500 command reference accessible by symbol, name, or command group - This section provides a link to SLC 504 commands using three types of listings. One grouping, symbols, lists PLC ladder logic commands by the AB three-letter symbol. The lists the commands by name. The third lists commands in logical groups. For example, the group *timers and counters* has all the commands associated with timers and counters. The command groupings reflect the section titles in the AB program reference manual. The screen captures in Figure 2 illustrate the technique to select commands using the symbol and group listings.



(a) (b) (c)

Figure 2 (a) Command Index, (b) Symbol Index, (c) Command Group Index

The Command Index view is obtained by selecting *SLC 500 Commands* from the main menu, and the two remaining indexes illustrated in Figure 2 are selected from the listing on the Command Index.

Command syntax and example ladder logic programming applications - The content and structure for the pages describing each of the commands was developed using the criteria that the site should provide all of the support that a new PLC user needs to develop an application program. In addition, it should provide experienced users with a source to check the syntax on less frequently used commands. Therefore, the content has to include all of the information available from the numerous AB manuals for PLC operation, programming, and reference. In addition, it has to be organized to support the needs of both new and experienced users. Another requirement is that references to other commands or AB information should be hyper-linked to that page in the on-line guide. The screen in Figure 3 illustrates how a command is introduced. Note that the graphic of the button used in the Rockwell software is displayed, along with an introduction to the command. Sufficient detail was added to make the command operation and programming syntax as clear as possible.

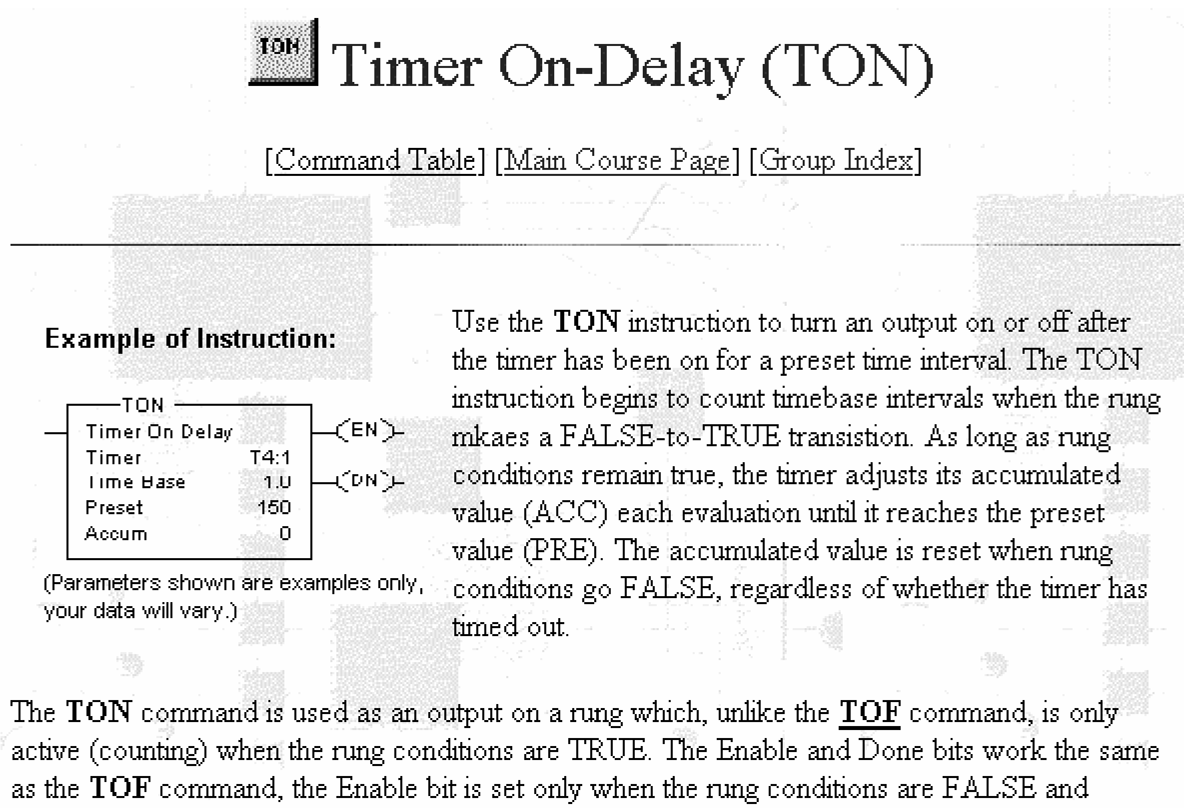
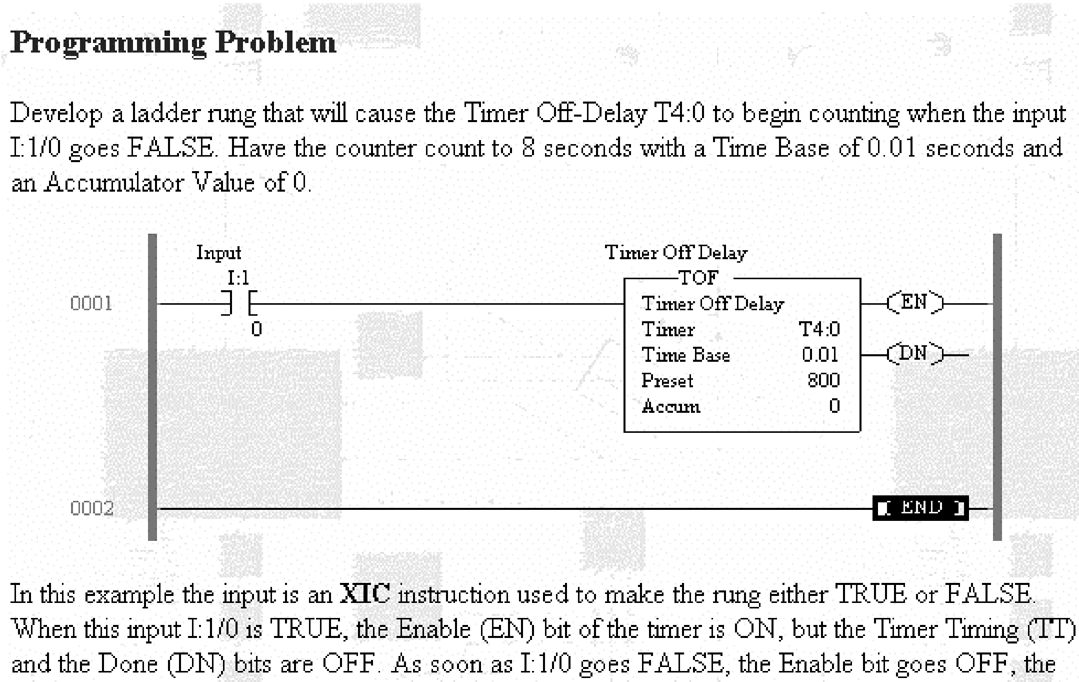


Figure 3 Command Description

In the body of the description there is a hot-link reference to a Time Off Timer, **TOF**, that provides information about that command. Another part of each command is an example ladder logic solution illustrating the syntax for the command. Figure 4 illustrates one of these examples for the Time Off Timer command. Commands are also organized using the groups of commands that have common syntax or operation. The groups are accessed using the index pictured in Figure 2c. The group pages feature an overview of the group function (Figure 5) followed by a graphic that illustrates where the group name appears in the programming environment (Figure 6). In addition, Figure 6 provides a list of all the commands in the group and a brief description of each command with a link to the command page.

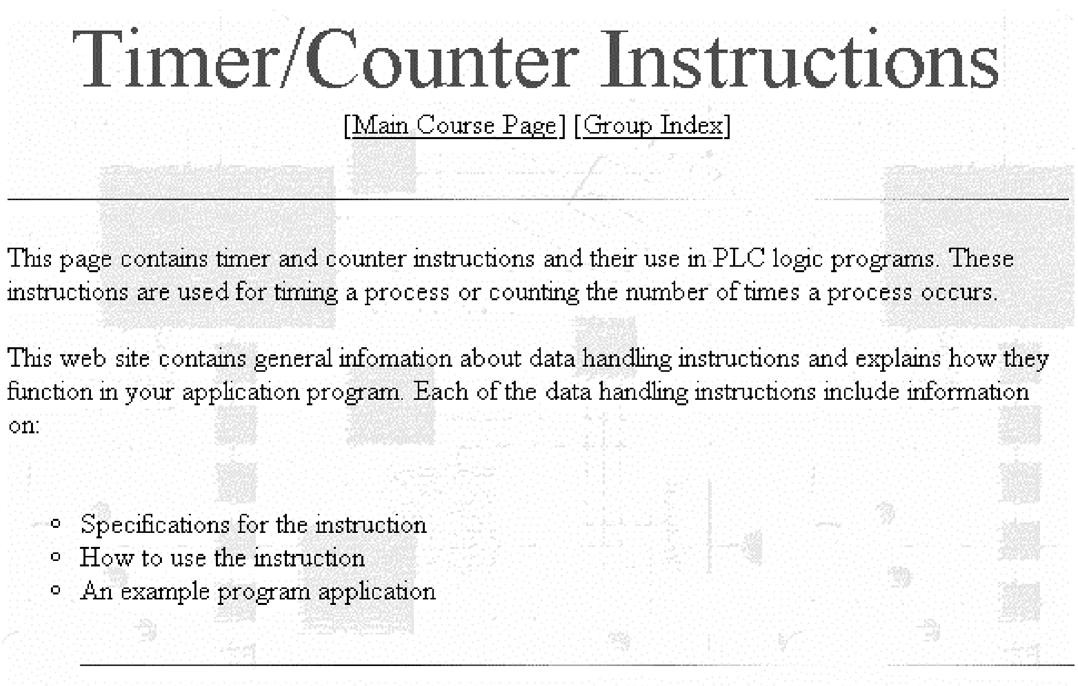


Programming Problem

Develop a ladder rung that will cause the Timer Off-Delay T4:0 to begin counting when the input I:1/0 goes FALSE. Have the counter count to 8 seconds with a Time Base of 0.01 seconds and an Accumulator Value of 0.

In this example the input is an **XIC** instruction used to make the rung either TRUE or FALSE. When this input I:1/0 is TRUE, the Enable (EN) bit of the timer is ON, but the Timer Timing (TT) and the Done (DN) bits are OFF. As soon as I:1/0 goes FALSE, the Enable bit goes OFF, the

Figure 4 Example Ladder Logic Program



Timer/Counter Instructions

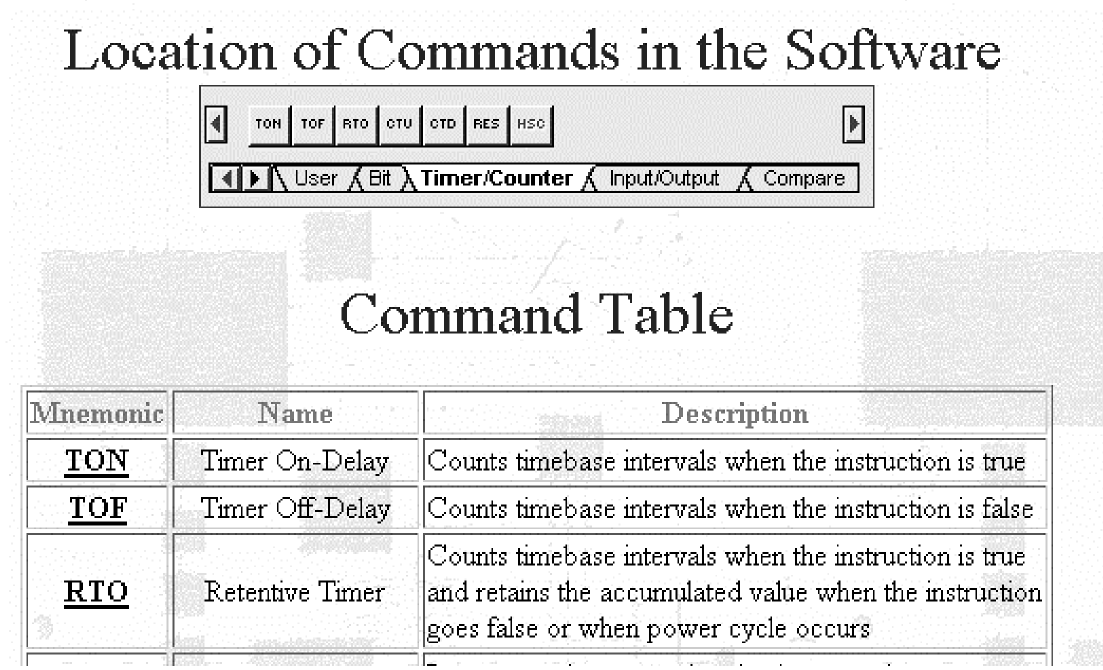
[Main Course Page] [Group Index]

This page contains timer and counter instructions and their use in PLC logic programs. These instructions are used for timing a process or counting the number of times a process occurs.

This web site contains general infomation about data handling instructions and explains how they function in your application program. Each of the data handling instructions include information on:

- Specifications for the instruction
- How to use the instruction
- An example program application

Figure 5 Example of Group Command Splash Page



Mnemonic	Name	Description
TON	Timer On-Delay	Counts timebase intervals when the instruction is true
TOF	Timer Off-Delay	Counts timebase intervals when the instruction is false
RTO	Retentive Timer	Counts timebase intervals when the instruction is true and retains the accumulated value when the instruction goes false or when power cycle occurs

Figure 6 Description and Links to Commands in the Timer Group

Introduction to programming - This section describes the programming environment in the AB Logics 500 and Links software. Students new to the Logics 500 software could use this section to learn how to logon, develop ladder logic programs, and upload those programs to the target PLC.

Introduction to development of human machine interfaces (HMI) (under development) - This section covers the development of HMI screens using the Visual Basic Active-X controls provided in the Logics 500 software. In addition, other HMI software will be added to build more robust interface screens.

Laboratory exercises - The final section includes copies of the laboratory exercises required for two PLC courses in the Penn State Altoona BS in Electromechanical Engineering Technology course sequence. The experiments are provided in Word 97 and Adobe Reader format so that students can view the procedures on-line and also print out any portion of the exercise needed for the final report.

Conclusion

This work-in-progress web site is designed to be a one-stop source for AB SLC 500 program development information, both for the student just learning PLCs and for the experienced PLC user. In addition, it should serve as a template for other sites supporting PLCs from other vendors.

Acknowledgment

The initial work on this project was started as a team project in the EMET 430 Automation Machines course taught in Fall 1997. The students listed on the splash page, Figure 1, developed the descriptions of the commands listed on the site. Allen Bradley did not support the development of site material.

References

1. *Penn State Altoona Graduate Survey, 1998.*
2. *Reference Manual*, Rockwell Automation, Inc., Milwaukee, WI, 1996.
3. *Discrete I/O Manual*, Rockwell Automation, Inc., Milwaukee, WI, 1996.
4. *Step-by-step Guide to Project Development*, Rockwell Automation, Inc., Milwaukee, WI, 1996.
5. *Data Highway/Data Highway Plus*, Rockwell Automation, Inc., Milwaukee, WI, 1996.

BIOGRAPHY

JAMES A. REHG – James Rehg received a B. S. and M. S. in Electrical Engineering from St. Louis University and has completed additional graduate work at the University of South Carolina and Clemson University. Since August 1995, Jim has been working as an assistant professor of engineering and as Program Coordinator of the B. S. program in Electro-mechanical Engineering Technology at Penn State Altoona. He is the author of five texts, including the following books published by Prentice Hall: *Introduction to Robotics in CIM Systems 4th ed.* and *Computer Integrated Manufacturing*.