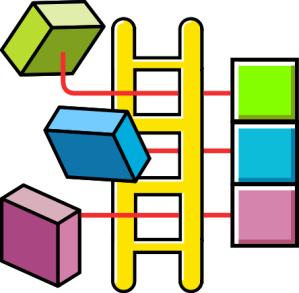
**A SELF-SELECTED HOMEWORK  
DOCUMENTING THE USES, BASICS,  
AND IMPORTANCE OF LOGIXPRO 500**

*By Luke Walmer*



SUMMARY

With the rapid advances in industrial robotics and automation, many companies are racing to establish themselves as the leading supplier of automation components, assemblies, services. One of the largest companies that has managed to stand out from its competitors is Allen-Bradley. Manufactured by Rockwell-Automation, Allen-Bradley provides a broad array of Programmable Logic Controllers (PLC), automation sensors, motor-control programs, and a number of other key automation components. This paper documents a study of LogixPro; a PLC simulator designed to teach a user the basics of PLC usage, language, and processes.

*MEEG453 – Manufacturing Processes and Systems  
Dr. Fagerstrom  
Due: November 15, 2011*

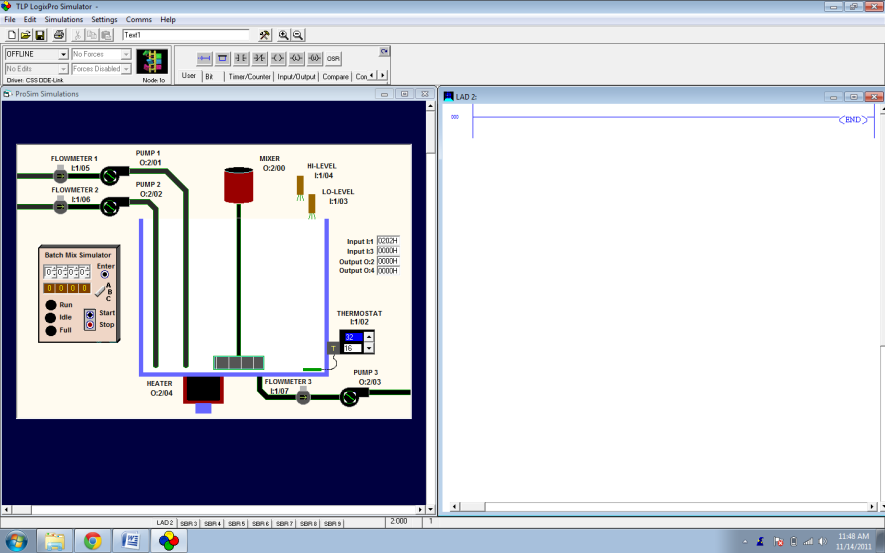
**SECTION I - MAIN DISCUSSION**

The knowledge of programming in a production environment is extremely valuable. Whether a company or process requires hard coding, machine code, programmable logic controllers (PLC’s), or robots, an employee who understands and implicates computer programming and controls has the ability to substantially contribute to their team. Because of the value associated with programming and direction in which many manufacturing processes are headed, I decided to learn the basics of programming, using and implementing PLC’s.   
 So what exactly is a PLC? A PLC is a device that contains both the hardware and software necessary to operate an automated industrial controls system. To the user, a PLC consists of two main components, a computer or processer and an interface that displays inputs and outputs. In an automated setting, the PLC is connected to various additional components including, but not limited to, sensors, lasers, switches, and displays which show outputs and changeable inputs. The basic operating process that a PLC follows is: read the inputs, process them into programmed commands, and execute the commands, thus producing an output of actions or data. Many times, a PLC is programmed for continuous scanning of the code and runs infinitely until a user indicates otherwise or changes the program.

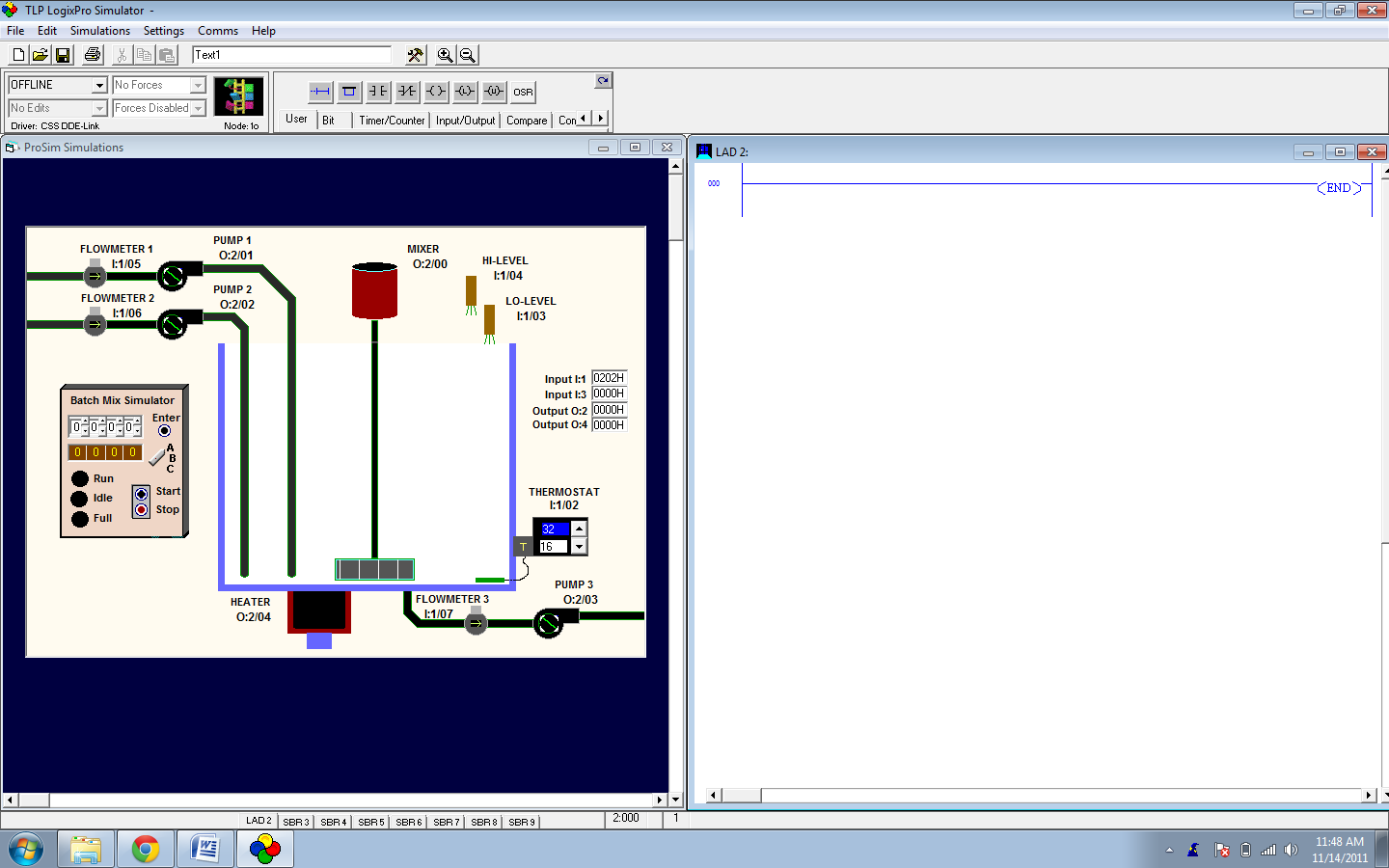
  
*Figure 1. Omron CP1L Programmable Logic Controller*[[1]](#endnote-1)

PLC’s have been used in various manufacturing industries since the late 1970’s when a company called Bedford Associates, located in Massachusetts, developed a device to replace the relay control systems used by General Motors. Relay control systems are notorious for limited lifetime as their mechanically operated parts wear down over time, while a PLC digitally processes and commands information. [[2]](#endnote-2) In today’s industry, PLC’s are commonly used for relay control, motor control, networking, and calculations/data managing.[[3]](#endnote-3)  
 A PLC works by receiving user inputs, converting them to Boolean values and processing them through a sequence of time delays, counters, and motion commands. The data or results from the process can then be saved or fed back through the program as new inputs.[[4]](#endnote-4) Most PLC suppliers provide the buyer with software than can be used to program the device. Some of the major suppliers of PLC’s and PLC components are Allen-Bradley (who interestingly enough owns the rights to the term PLC, even though it is commonly used), FANUC, Rockwell Automation, General Electric and Omron.[[5]](#endnote-5) The major issue with learning how to program PLC’s is that an actual PLC is needed to complete the commands. This can be a costly problem as, for example, the basic PLC’s made by Allen-Bradley carry a retail price of $500-$1200. To solve this problem, certain companies have developed a program which hosts a virtual PLC. The virtual PLC simulates the action of the actual hardware, while allowing the user to learn and understand the entire programming process without having to purchase and actual PLC.   
 One of the most commonly used PLC simulation programs is called LogixPro and is jointly developed by The Learning Pit and Allen-Bradley. It provides users with an excellent training resource that moves though basic and into advanced logic control programming. The interface is user friendly and splits the screen between a programming mode and animated run mode, complete with moving parts, indicators, sounds, and tabulated data results. The basic tutorials thoroughly explain the process of relays and how programmable logic can be utilized to count and alternate commands in a system. As the tutorials increase in difficulty, more applicable and specific example are given, including compressors, batch mixing, and garage door opening commands. Once the user has comfortably completed these middle level examples, they have the option to program more advanced processes like bottle capping lines and elevator controls. In my own learning procedure, I completed the Silo and Compressor tutorials, and used the knowledge of those two simulations to complete the Batch Mixing project.

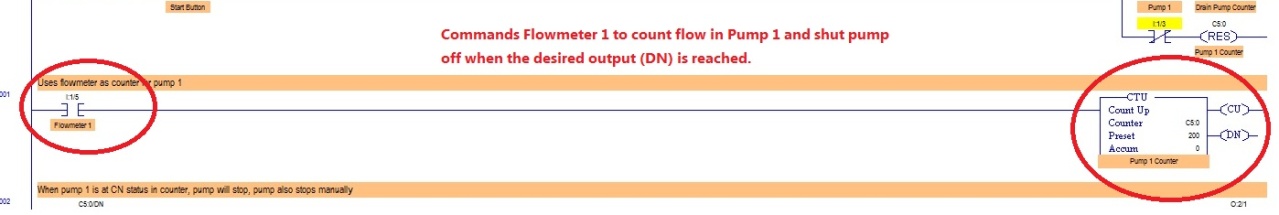
When the Batch Simulator is first opened, the screen looks like the image displayed in *Figure 2*. On the left an animated setup is displayed and on the right is a blank sheet for the programming lines.

  
*Figure 2.*

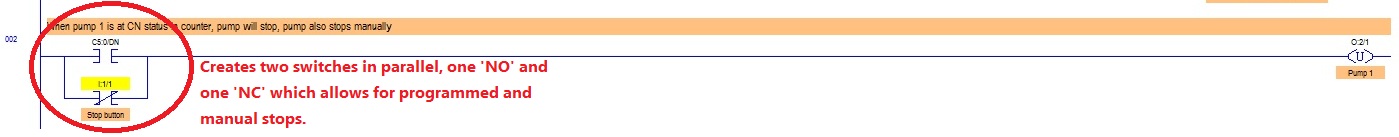
LogixPro uses a ‘ladder-like’ concept to process the inputs. The code moves sequentially through a series of ‘rungs’, which each contain linked commands and branches. The program automatically organizes the inputs to left of each rung, while moving the outputs to the right side. LogixPro also has the ability to name and reference commands, which creates a much cleaner working environment, i.e. if a command is properly named, the next time the same type of command is identified, all of the labels and settings will be copied to the new command. The user is provided with easily accessible command icons shown in *Figure 3*. The entire program adopts a ‘drag-and-drop’ principle, in that the user clicks on and drags the icons to the ladder rungs, rather than typing in code for each individual command.

  
*Figure 3.*

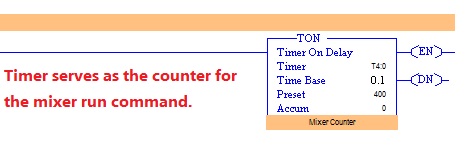
To begin writing the program, the user starts by adding an input and a corresponding output. The output does not necessarily need to be the final operation that is run, it simply needs to ‘close out’ the input so the data is ready for the next command. In this example, I added two normally open reset commands in series followed by a normally open switch for the start button. The reset commands tares the inputs to a preset value, zero. The start button is normally open or in the ‘true’ position once the program is initiated by the user. The corresponding outputs selected are a series of parallel pump controls and counters. Pump 1 will run until the predetermined value is reached (when the tank is full). The way that I set the counter to work is to view the true:false switches as gallons pumped. Once the Pump 1 counter reaches the maximum volume, Pump 1 shuts off, and the drain pump counter begins recording data until the preset value and time is reached. I used the flowmeters for Pump 1 and the drain pump as the hub to which I attached the counter. This layout is shown in *Figure 4*.

 *Figure 4.*

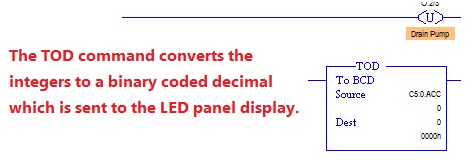
One necessary line of code, which is also important in any operation, is an E-Stop (or Emergency Stop). To do this, I added two switches in parallel for the stop command. The top switch is normally open and serves as the automatic stop command linked to the Pump 1 counter. The bottom switch is normally closed and serves as the manual stop of the entire process. This command is shown in *Figure 5*.

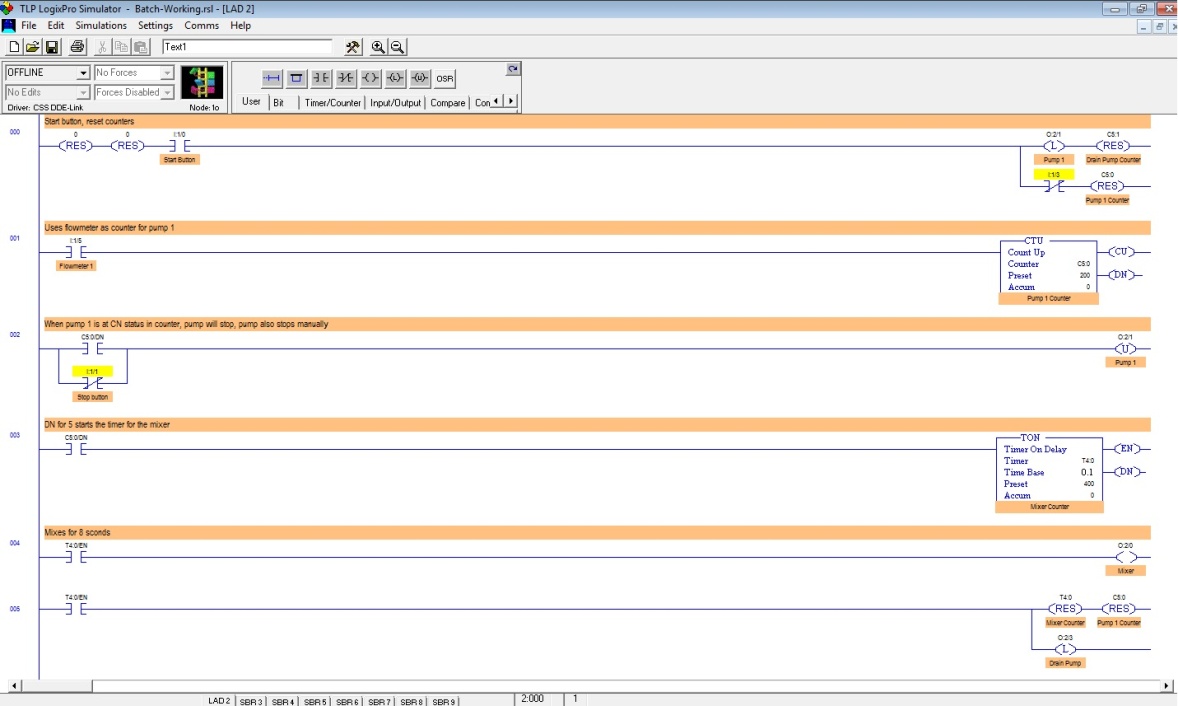
*Figure 5.*

The next step is a good example of the relays that are replaced virtually by the timer delay. This is included so that the mixer runs while the tank is held full. In this program, the timer delay period is hardcoded into the ladder. However, it can be coded as an input for the user if different batches require different mix times. The timer that I added is shown in *Figure 6* and is a process connected to the stopping of Pump 1. Once the timer reaches its full count, the mixer runs.

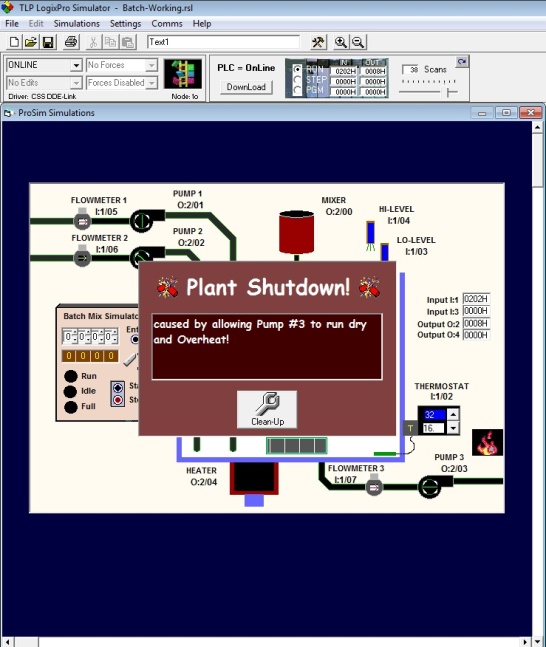
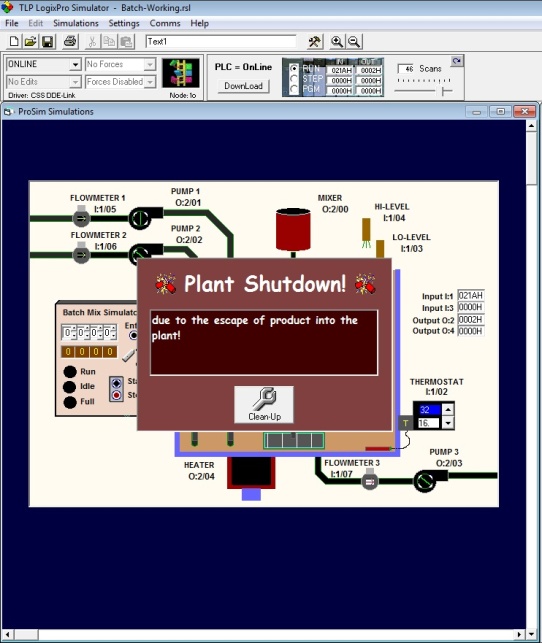
  
*Figure 6.*

One uncommonly ignored factor is that the language that humans and machines/computers read is very different. To account for this, I added a TOD command (To Binary Coded Decimal) which converts the incoming integer value to a digitally readable form which is displayed on the LED panel. This command is a good example of the command linking that takes place. The source is listed as the counter. In this case, the TOD output is the last command in the program and is shown in *Figure 7.* A screen capture of the final layout of the program can be seen in *Figure 8*. The image also clearly illustrates the ladder rung layout of that is used by LogixPro,

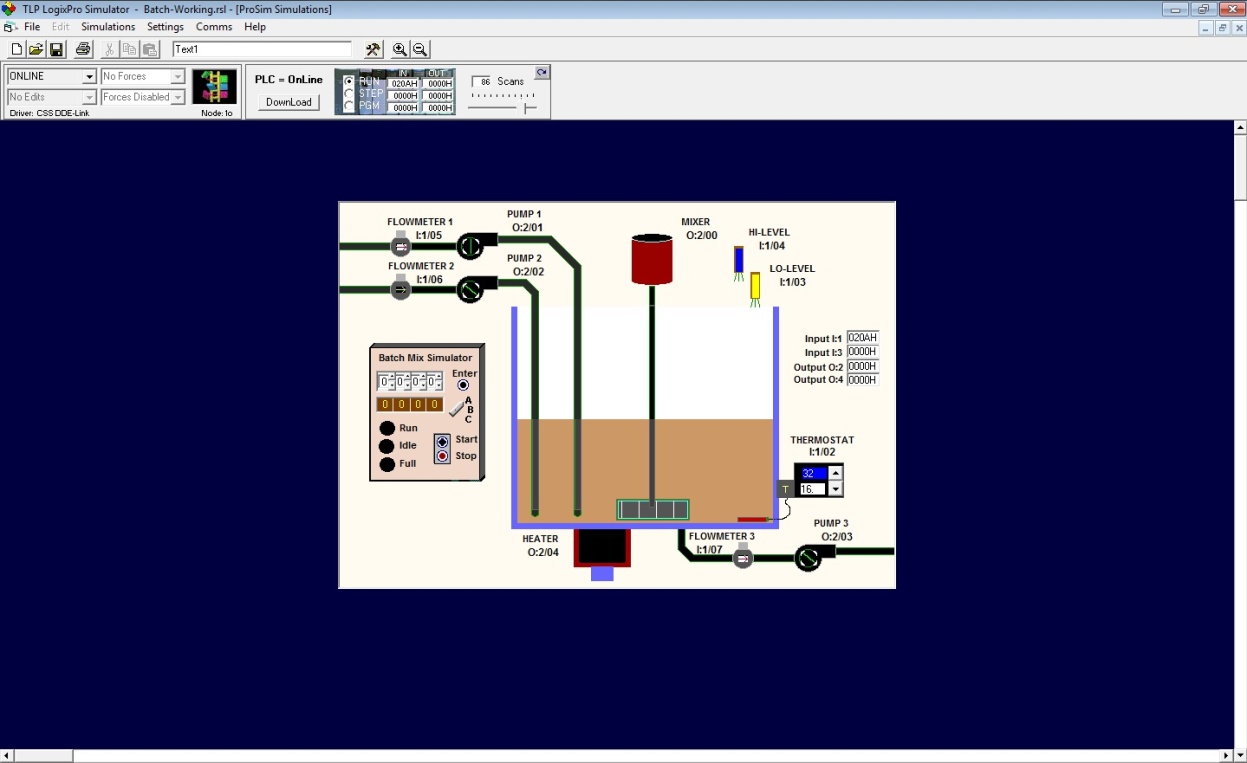
  
*Figure 7.*

  
*Figure 8.*

The final step in program creation is to update the PLC driver. To do so, the user must click the ‘Download to PLC’ button. This simulates downloading to an actual PLC for operation. This is where PLC simulation programs come in handy. Just as a 3D modeling program and rapid prototyping can save time and money, PLC simulation allows a programmer to work out errors, fine tune the process, and effectively select the appropriate PLC to purchase by providing accurate results and data. LogixPro is also user friendly and ‘smart’ in a sense. Using the batch mixer as an example, if the program is coded improperly and the drain pump continues to run after the material has exited the system, the user will be notified of a ‘Plant Shutdown’, citing a fire due to the drain pump overheating. The same occurs when the material flowing into the tank exceeds the maximum capacity; the plant shutdown error is displayed, citing an overflow of chemical substance. These instances are shown in *Figures 9 and 10*.

   
*Figures 9 & 10.*

A final rendering of the program successfully running (filling, mixing, and draining) is displayed in *Figure 11*. *(A video of this process will be shown during the presentation)*.

  
*Figure 11.*

**SECTION II – DISCUSSION OF ECONOMIC ASPECTS**

PLC’s serve as the ‘brain’ in an automated system, thus with the growth and inclusion of automation in manufacturing, including PLC’s in a process is, in most cases, a financially smart move. Although the automation industry takes a heavy hit every time the economy plummets (as people look for things they don’t ‘need’ to cut down spending), the direction of manufacturing is expanding every day towards automation. More processes are becoming automated because the speed, accuracy, and flexibility cannot be matched. In many cases, human labor simply cannot meet the product delivery demands. Automation is also being heavily integrated into the research and development setting as experiments and tests can be efficiently completed to perfection. These listed demands are directly related to the need to justify purchasing automation components, primarily PLC’s. A starter PLC can be purchased for as low as $70.00. Granted, these models are very limited in their abilities, however, a moderately capable PLC can be purchased for approximately $300.[[6]](#endnote-6) The cost of an automated system can be easily offset by the increase production rate, which in turn increases product revenue. Most of today’s PLC’s come with a program guide that presents an easy-to-use interface, so little time is spent becoming accustomed to and implementing the software. Moreover, PLC simulation software like LogixPro(retails for approximately $40) allows for even more cost saving as it enables a company to train employees without having to purchase additional PLC’s for the training sessions.

**SECTION III – RELEVANCE FOR THE BEGINNING AND SENIOR ENGINEER**

The significance of this topic for a beginning engineer is to understand the ease of learning PLC programming and the importance of automation integration into today’s manufacturing industries. As a young engineer, the ability to contribute cutting edge skills and hard-to-find knowledge can make the difference between you or another candidate securing a job opening. Understanding PLC’s and their applications is another excellent tool to have in the engineering toolbox.  
 The significance of this topic for a senior engineer is that it allows him/her to keep up with the constantly changing available technologies. As a senior engineer, staying up-to-date with all of the newest products can be difficult. It is also important because as an experienced engineer, he/she may find themselves in a decision-making position on whether or not to incorporate automation or make an expensive purchase. Understanding the value of PLC’s can help them make the right decision.

**SECTION IV – WHAT I LEARNED**

By using LogixPro to teach myself the basics of PLC programming, I now understand the value in incorporating PLC’s and automation into a process. Programming and coding has always seemed like a foreign language from which I kept my distance. By completed this self-selected homework, I have been able to overcome that fear and actual find enjoyment in designing programmable logic systems. I’ve discovered that it is very rewarding to fully write a program and have it work. In the process, I have also become much better versed in the products currently on the market and their respective costs. This will help me work more efficiently with others if my future job position requires use of PLC’s. I also have gained a high level of respect for experienced engineers who have the ability to write advanced automation programs for PLC’s. I strive to be able to contribute similar high-level programs to my future employer.

1. Omron CP1L Compact Machine Controller. <http://industrial.omron.eu/en/products/catalogue/automation\_systems/programmable\_logic\_controllers/compact\_plc\_series/cp1l/default.html> [↑](#endnote-ref-1)
2. “What Is A PLC?” AutomationDirect.com. November 14, 2011 <support.automationdirect.com/docs/whatisaplc.html> [↑](#endnote-ref-2)
3. “What Is a Programmable Logic Controller [PLC]?” Wisegeek.com 2003. November 14, 2011. <www.wisegeek.com/what-is-a-programmable-logic-controller.htm> [↑](#endnote-ref-3)
4. “Programmable Logic Controller – What’s one of those?” Machine-Information-Systems.com 2007. November 14, 2011. <www.machine-information-systems.com/PLC.html> [↑](#endnote-ref-4)
5. www.pdfsupply.com [↑](#endnote-ref-5)
6. Industrial PACs and PLCs. AutomationDirect.com 2011. November 14, 2011. <http://www.automationdirect.com/adc/Overview/Catalog/Programmable\_Controllers?source=google&keyword=cheap%20PLC&type=search&gclid=CPfmrKf2tqwCFYnd4AodMgHJFg> [↑](#endnote-ref-6)