Session 1620

Development of a Virtual Production Machine for a PLC Laboratory

Don Lambing, James A. Rehg, Jamie Rowe
Cambria Systems Inc./Penn State University Altoona College/ABB Robotics Inc.

Abstract

The programmable logic controller (PLC) as become the de facto standard for control of production machines, work cells, and flexible manufacturing systems. As a result, colleges and universities have added PLC systems in manufacturing and automation laboratories to teach machine control using PLCs. In most implementations the laboratory has multiple student stations, each equipped with a PLC system and computer for programming. This paper addresses a critical component, the target manufacturing system, which is often missing in the integration of PLCs into the automation laboratory. The laboratory exercises associated with the PLC laboratory component usually start with projects that introduce students to the PLC operating system, input/output modules, electrical interface, and the computer based programming language to create ladder logic. These exercises are followed by a series of experiments that cover how to use the PLC ladder elements, such as inputs, outputs, timers, and counters, and the PLC program ladder structure. Up to this point the PLC is usually interfaced to a set of lights and switches in order to display outputs and provide input conditions. The most important part of PLC programming, applying the PLC to the control of an industrial machine or system, is not an option at many institutions for the following reasons: replicating manufacturing systems for each PLC station is too costly; the systems have too large a footprint for the space available; or the number of different types of systems required limits their use.

This paper describes a system in use at Penn State Altoona that supports the development of machine control programs at eight PLC student stations and the testing of the solutions at a single common bench where a variety of manufacturing systems can be introduced. The use of a common manufacturing system would not be an adequate solution to the problem without the ability to develop the machine control program at each PLC station on a virtual manufacturing machine. Wonderware software, a human machine interface software package with a broad industrial base, is used to simulate the manufacturing machine and provide students with a virtual model of the machine in order to measure the success of their control programs. After the machine program is developed and tested on the virtual manufacturing machine at the PLC student station, the program is passed over a network to a PLC on a common bench with the

actual hardware system. This laboratory configuration has the following advantages: 1) a single hardware system supports multiple PLC stations; 2) the virtual machine is sufficiently robust to verify a successful program; 3) students see their program drive an actual production machine; 4) verification of the program on the actual production machine is usually fast; and 5) a large number of different types of production machines can be used with the multiple student stations. The paper describes the system elements and architecture, development of the virtual machine and simulation software, typical manufacturing machines, and laboratory procedures.

The PLC Laboratory System

The automation laboratory at Penn State Altoona includes eight student stations one of which is illustrated in Figure 1. The stations include an SLC 504 programmable logic controller (PLC) from Allen Bradley (AB); a Windows 95-based microcomputer with ethernet capability to the college backbone and to a proprietary AB data-highway linking all the PLCs on the student benches; an input/output panel with lights and switches; and several rows of terminal blocks for interfacing with the PLC, lights and switches, and other external systems. The laboratory supports three courses that cover PLC programming for two associate degrees and the Electro-mechanical bachelor's degree programs. The last course covers programming and control of automation systems; as a result, more complex automation systems were required for each of the eight laboratory stations. Initially, four small table-top assembly systems were added to the laboratory to satisfy this training need. Each of the four trainers is shared by two student stations through the use of an interface system between the benches and the trainer. However, when the need for additional target control systems was identified, it became apparent that adding additional systems presented the following problems: the space available around the benches do not permit additional systems to be permanently added; bench top space would not allow the new desired systems to be placed on the bench; storing four sets of trainers that would be moved into the laboratory when needed was not a valid option due to space limitations; and the cost of purchasing four system for each new target application was not a cost that the department could justify.

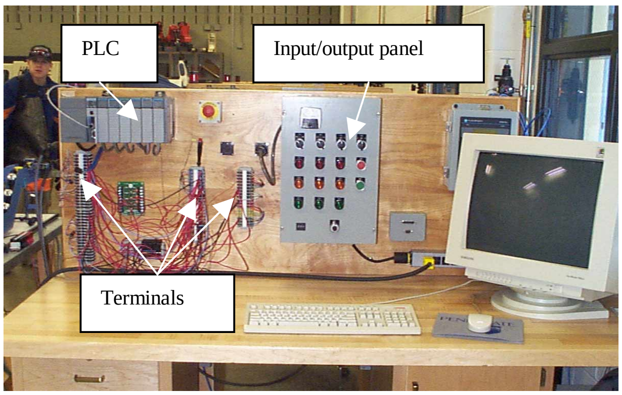


Figure 1

A Virtual Solution

The need for multiple different manufacturing systems at each of the student stations was satisfied by developing virtual manufacturing systems capability at the eight student stations. The virtual system strategy includes the following elements: 1) students are presented a virtual manufacturing system (VMS) on the computer screen that was developed using Wonderware, a human machine interface (HMI) software used in many industries; 2) the students develop a PLC program to control the virtual machine just as they would for a real hardware system; 3) the program is tested using the VMS on the computer with the dynamic data exchange (DDE) feature of the Windows environment linking the PLC solution to the virtual machine in Wonderware; 4) after student demonstrations of a successful solution on the VMS, the program is moved over the network to a PLC at a station that has a single hardware implementation of the manufacturing system; and 5) the students verify that the program developed for the VMS works equally well on the real manufacturing system.

The Virtual Manufacturing System

The VMS was developed using Wonderware's capability to link data generated in the PLC to images displayed on the computer screen. Using DDE, a change to an input, output, binary bit, integer, or analog value in the PLC causes a variable in Wonderware to reflect the same change. The Wonderware variables are linked to properties of the screen images, causing the images to change their visiblity, color, orientation, or position. The Properties Dialog box illustrated in Figure 2 shows the image properties that can be controlled. Note the broad range of parameters that can be controlled (the grayed-out parameters can be active as well). The target manufacturing system selected for the initial virtual machine was a two-axis pneumatic robot used to assembly two parts. A picture of the system is provided in Figure 3. The system has two pneumatically-actuated magazine type parts feeders (the vertical tubes in Figure 3), vertical and horizontal axes, and a series of sensors that are used to detect the presence of parts and the location of the robot axis. In the upper right corner of the figure, one can see an SLC 505 PLC that uses an ethernet connection to link it with the student station computers.

The development of the virtual machine for the system in Figure 3 started with the development of bitmap images to represent all of the parts and axis positions of the actual hardware. The virtual system parts are built using the draw features of Wonderware in the Windowmaker screen illustrated in Figure 4. Using the drawing tools the robot profile is created with all of the necessary permutation to reflect actual operation. The images used in the VMS are illustrated in Figure 5. Wonderware offers two options for creating the many different views present in Figure 5. One method starts with an individual picture for all of the possible views and uses on/off logical conditions of variables to make a specific view visible and all others views hidden. For example, turning off Figure 5b and turning on Figure 5c to animates the robot in the vertical

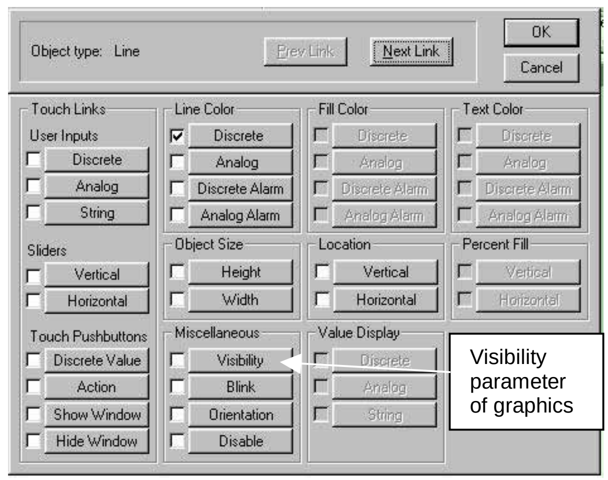


Figure 2

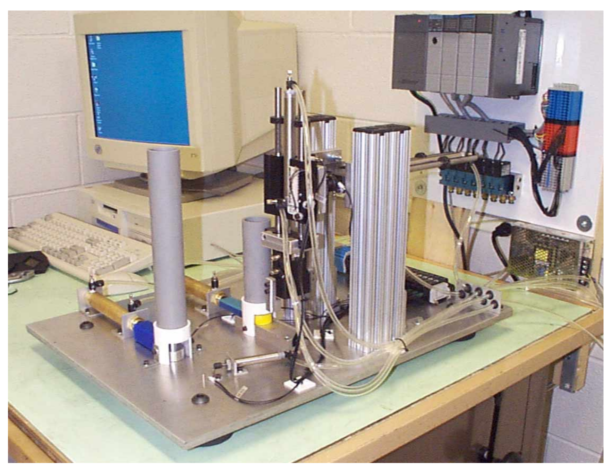


Figure 3

direction, making appear that the robot is picking up a part. A second approach uses the position command to move parts of the object, like the vertical axis, based on changes in the PLC program or in the on/off condition of variables. For example, with Figure 5b visible the gripper and vertical axis bar would be moved in a downward direction to simulate the change in the vertical axis that is required to acquire a part. As a result, the screen would change from the view in Figure 5b to that in Figure 5c. Both techniques were tested and both produced equally good results. The first technique is more drawing intensive and the second requires some additional scripting and careful grouping of drawn objects to move only specific parts of the figure.

The parts feeders visible in the system in Figure 3 were not represented in the VMS graphically. It was decided that part feeder action would be shown in the VMS by dis-playing a part when the feeder was activated in the PLC ladder. This action is illustrated by the graphics in Figure 5a (no part present) and Figure 5b (the top and base parts present). The goal of the VMS was to test the operation of the PLC program and not to model the manu-facturing system in detail.

The manufacturing system has eight sensors located at strategic locations to mon-itor the system operation and to provide feed-back to the control program (Figure 6).

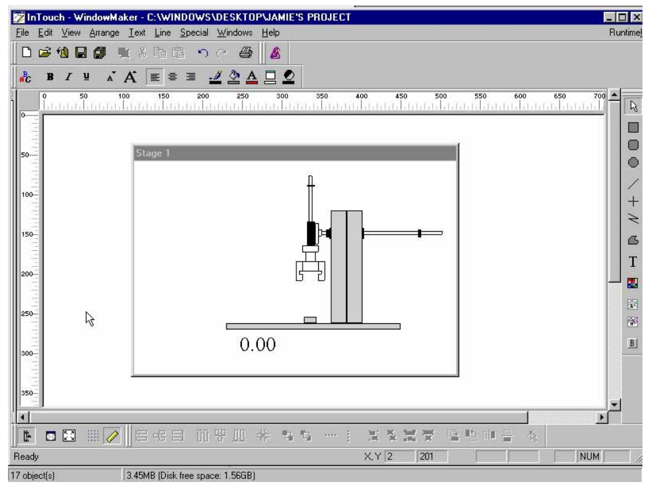
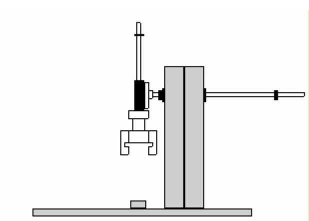
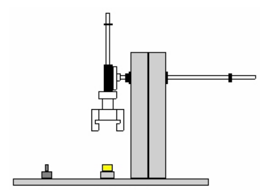


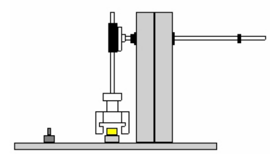
Figure 4



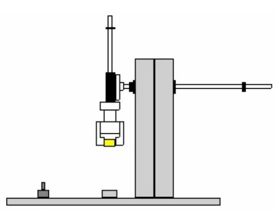
(a)



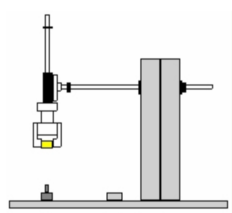
(b)



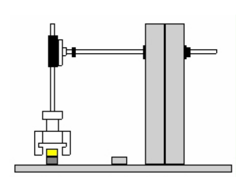
(c)



(d)



(e)



(f)

The two robotic actuators have limit sensing that indicates when an axis reaches one of the two limits or extremes of travel. In addition, sensors indicate when a part is in the pickup and assembly areas and if the parts feeders have a part ready to be injected. When the PLC program is driving the actual manufacturing system, the sensor outputs are used to move the PLC ladders through the programmed sequential function rungs. In the VMS the sensor signals are generated with script code that simulates the transition time for the actual actuators.

The PLC and VMS Link

The link between the PLC and the computer running the VMS Wonderware program is built using the dynamic data exchange (DDE) function in the Windows operating system. The DDE can work locally with programs in the same computer sharing data, or between nodes, with a proprietary network like the Allen Bradley (AB) data highway. The DDE link can also work over the network using netDDE options. In the VMS project the DDE link is local between the PLC processor and the Wonderware software application. In the automation laboratory, the PLCs are a node on the proprietary network, along with all of the computers at the student stations. As a result, the data is exchanged between the VMS and PLC through a PLC interface card (KT type AB card)

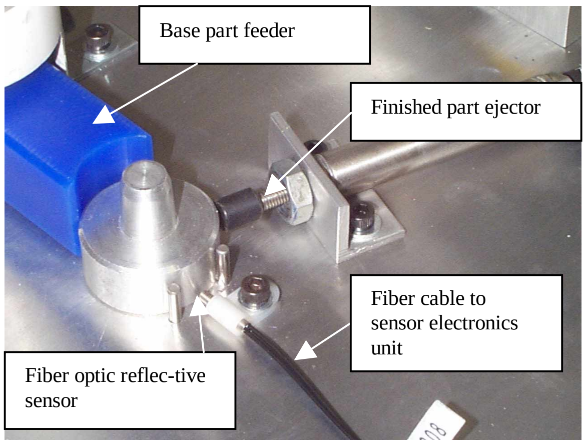


Figure 6

The DDE link between Wonderware and an Allen Bardley PLC is built by adding a common topic name to dialog boxes in both systems. The link in Wonderware is created using the dialog box in Figure 7. Note in the figure that the topic name is *ladpro,* and the access name has the same identifications. The access name is the name of the PLC program, and the topic name is the DDE name used to identify the two Windows applications that will be exchanging data. The access name and the topic name can be different. The application name is the name of the software application that provides the DDE link capability. In the Allen Bradley PLC, the *rslinx.exe* program provides this capability.

On the Allen Bradley side the link to Wonderware is created inside of the rslinx application by referencing that same topic name. With the link established, changes in PLC inputs and outputs (both discrete and analog) and all types of registers (bits, integers, and time/counter accumulators) can be used to change object characteristics directly in the Wonderware application or can be passed to variables for other types of object and programmed control.

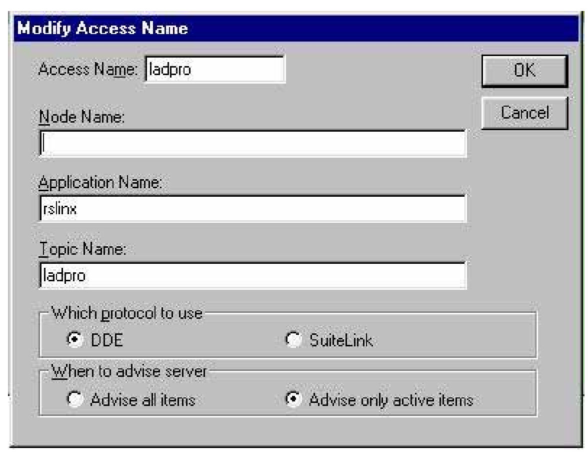


Figure 7

A second dialog box in Wonderware is used to link to the variable in the PLC that is changing. The dialog box in Figure 8 illustrates how that variable is specified. The dialog box is called the Tag Name Dictionary and includes a number of items. The variable in Wonderware

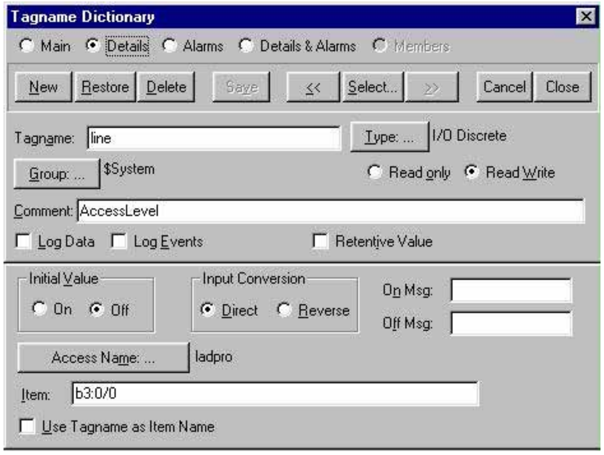


Figure 8

that will contain the PLC data is specified by the Tag Name, while the name of the variable in the PLC where the data originates is specified in the text box labeled item. Note that the tag name is line while the item name is *B3:0/0*. This means that the bit value in the binary register B3:0/0 in the PLC will be represented by the variable *line* in Wonderware. The names used for the tag name and item can be identical if it makes tracking and identifying the variables easier. Additional parameters are controlled by this dialog box, such as the initial value and the specification of the type of variable, which in this example is I/O Discrete.

With communications between the applications established, it is possible for either application to change parameter values in the other application. For example, a change in the PLC value of B3:0/0 from off to on will cause the value of the variable line in Wonderware to change from 0 to 1. Likewise, if Wonderware changes the value of a line from 0 to 1, then the binary bit B3:0/0 in the PLC will change accordingly. This bi-directional control capability is used, along with the scripting language options in Wonderware, to develop programs that simulate the operation of the actual manufacturing system. For example, in the actual manufacturing system the movement of an actuator is verified by the limit sensors attached to each extreme of the actuator's travel. As a result, the program is sequenced by using the change in a limit sensor to indicate that an axis has reached its destination, and the next step in the sequence can be initiated. However, in the VMS these sensors are not physically present, and so they must be created through the use of scripts. The condition script offers one of several options for the creation of this type of simulation. Using the condition script, a software delay can be created that is initiated when Wonderware senses that an output of the PLC has activated an axis pneumatic solenoid. The delay simulates the transition time required for the axis to reach the other end of travel. After the delay, the script changes a variable to indicate that the limit sensor at the end of travel has been turned on. This change in the Wonderware variable is communicated to the PLC as a change a binary bit through the DDE interface, and PLC responds just as it would if an actual sensor was connected to that bit. The condition script dialog box is illustrated in Figure 9. Note that the condition types that are support include on true, while true, on false, and while false.

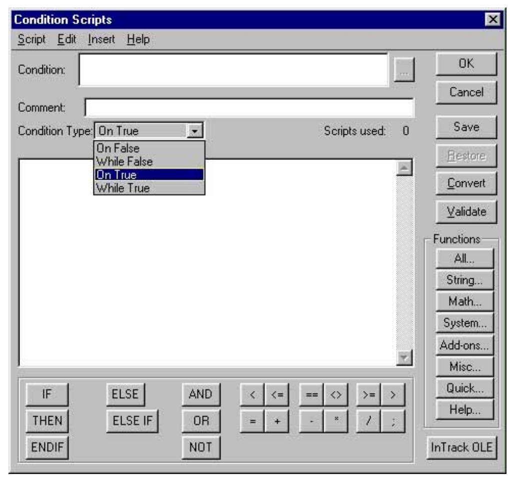


Figure 9

System Operation

The laboratory procedure to use the VMS in the advanced automation laboratory is:

1. Students are given a logical control problem and asked to develop a PLC program to satisfy the control need described.
2. Students are instructed to load Wonderware at the same time that they load the PLC RSLogics software. They are instructed to run the VMS application in Wonderware.
3. Students first develop PLC software that actuates the two-axis robot and assembly system using push buttons, so that they can see how the VMS responds to PLC control outputs.
4. Students develop the full control program in the PLC and observe how the VMS responds to their solution.
5. When the students and instructor are satisfied that the solution appears to be correct, the PLC program is passed over the network to the PLC connected to the single station in the laboratory with the actual manufacturing system (Figure 3).
6. The students test the program at the actual system to verify that operations meet all specifications.

Use of the system indicates that verification of the final program or identification of problems at the actual systems usually takes less than ten minutes. Therefore, it is possible for one actual system to easily support a laboratory with eight student stations, all working on the same control problem.

Conclusion

The virtual manufacturing system concept provides a number of advantages: 1) an eight-student station laboratory requires only one actual system trainer, which reduces acquisition cost; 2) the reduction in the number of trainers translates into reduced storage and laboratory space; 3) the interface to the VMS is less complex than the operation of the actual equipment, and so the students are not distracted with the hardware as they develop the control program; 4) a larger variety of automation systems can be used since the cost of only one station is incurred; 5) student experience with the actual hardware is not compromised with the use of the simulation. The singular disadvantage is that some systems are difficult to model with the limited programming commands present in the HMI software.

The next step in the development of this application is the development of additional trainers including a material handling system and a batch material process control system.

References

.

1. *Reference Manual*, Rockwell Automation, Inc., Milwaukee, WI, 1996.
2. *Discrete I/O Manual*, Rockwell Automation, Inc., Milwaukee, WI, 1996.
3. *Intouch Users Guide, Wonderware Corp., Irvine, CA, 1999*
4. *Data Highway/Data Highway Plus*, Rockwell Automation, Inc., Milwaukee, WI, 1996.

BIOGRAPHY

DONALD LAMBING - Donald Lambing is a Systems Engineer at Cambria System Inc. in Sidman, PA. Cambria Systems specializes in the design and implementation of industrial automation systems.

JAMES 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 2nd ed*.

JAMIE ROWE – Jamie Rowe works for ABB Robotics in the Field Service Unit. When he work on the project he was a senior student in the BS in Electro-mechanical Engineering Technology program at Penn State Altoona.