

---

## **AC 2011-1194: MODELING AND EXPERIMENTAL VERIFICATION OF PLC CODES IN A ROBOTICS AND MECHATRONICS COURSE**

### **Richard Chiou, Drexel University**

Dr. Richard Chiou's background is in mechanical engineering with an emphasis on manufacturing. Dr. Chiou is currently an associate professor in the Goodwin School of Technology and Professional Studies at Drexel University. His areas of research include machining, mechatronics, and internet based robotics and automation. He has secured many research and education grants from the NSF, the SME Education Foundation, and industries.

### **Yongjin (James) Kwon, Ajou University**

### **Robin Kizirian, Drexel University**

Robin Kizirian completed his M.S. degree in Computer Engineering at Drexel University in Philadelphia and his B.S. degree in Computer Engineering with a minor in Computer Science at the University of Hartford in Connecticut. He is currently serving as a research assistant at the Engineering Technology department at Drexel University. Robin has been involved in various projects funded by Pfizer, NASA, NSF and Department of Education. His areas of research include Embedded Systems, Mechatronics, Efficient Solar Energy Systems, Internet-based Quality Control and 3-D Online Education.

### **Matthew Dordai**

### **Bret Alan Davis, Drexel University**

Bret Davis is pursuing his B.S. degree in Engineering Technology at Drexel University in Philadelphia. He is currently serving as a research assistant at the Engineering Technology department. Bret's research interests involve Digital Electronics, Mechatronics, and Automation Systems.

# **Modeling and Experimental Verification of PLC Code in a Robotics and Mechatronics Course**

## **Abstract**

We have developed and implemented a discrete event simulation approach for introducing programmable logic controller (PLC) code to undergraduate and graduate industrial and manufacturing engineering and technology students at Drexel University. Students learn the simulation methods necessary to support the modeling and verification of automated processes in a robotics and mechatronics course. Experimental case studies are derived from topics of interest to train manufacturing engineers for performing virtual simulations on PLC modeling systems. The virtual design process is split into three main sub-processes that are recommended to be performed in a hierarchical fashion. These design process structures include the component 3D modeling stage, logical I/O modeling stage, and the cell 3D modeling stage. Both the virtual and physical models provide equivalent results. Course instructions are based on active learning. Tutorials and laboratories assist students in comprehending the simulation methods. The course is taught with the computer-aided simulation software, PLCStudio, to help enhance the combination of passive and active learning in the class.

## **Introduction**

An effort has been initiated to modernize and add a wider variety of PLC teaching innovations to the Robotics and Mechatronics Laboratory at Drexel University since 2009. The approach to modernize the laboratory involved the participation of undergraduate and graduate students for redesigning experimental setups and upgrading the automated equipment as part of their Co-op project. This approach kept the cost low and provided enhanced learning opportunities for the students who developed the experiments for MET 205 Robotics and Mechatronics. They gained valuable real world experience in the design and integration of automated equipment and PLCStudio Software. Additionally, requiring the students to integrate the components and wiring into professional looking enclosures provided them an insight into the skills required by mechanical and electrical engineers who maintain equipment in manufacturing plants.

This paper explores how PLC simulation can be adopted to the classroom in conjunction with a physical setup to enhance the students' learning experience. Students in the Engineering Technology program are required to complete a capstone course MET 205 Robotics and Mechatronics that increases the student's mechatronics design-for-manufacturability knowledge. A novel digital manufacturing technique is implemented in the course for students to learn the new approach that reduces cost and compresses time for delivery of products produced in batches. The technique is composed of three stages – digital prototype, physical prototype and robotic system. The technique effectively integrates the contemporary advanced manufacturing technologies such as solid modeling, CAD, and PLC in improving the product design, analysis, and production process. Implementation of simulation software, PLCStudio, provides students with the necessary tools to create and test a virtual representation of a physical work cell. Students would be able to use PLCStudio to develop a 3D work cell, define I/O, and establish robot motion paths. The use of STEP 7, a Siemens ladder logic and PLC emulating software

suite, allows students to write their own ladder logic which is then run on an emulated PLC, to read and write to and from the I/O model in the virtual cell<sup>1-5</sup>.

The level of complexity of the models that the PLCStudio software can simulate provides the students with the opportunity to develop sophisticated logical control systems for real-world applications. Such considerable capability makes the software ideal for carrying out experiments of varying difficulty levels. Two simulations have been executed to provide useful information about how PLCStudio could be adapted for beneficial use in an academic setting. The first experiment is a simple comprehensive demonstration of how a complete system can be built from scratch and simulated using PLCStudio. This demonstration would allow students to grasp the essential concepts of the software's functionality within the time constraints of the course. The students then verify their simulation and PLC code by testing it on an analogous physical system. The second experiment involves both a real-world and a virtual quality control work cell. Having both the physical cell and the virtual cell allows students to verify their PLC code and analyze it before implementing it in a real PLC<sup>6-7</sup>.

### Simulation software

The methodology used by PLCStudio for its simulation is based on state-based object models and their interconnection to a graphical 3D model. Components built in the virtual platform are a close approximation of the physical laboratory workcell. In addition to the 3D virtual environment, the I/O models of the PLC need to be developed properly to complete the simulation. Furthermore, the workflow to implement virtual commissioning in physical laboratories or shops is discussed<sup>8-9</sup>. This project demonstrates the result of simulation to evaluate the functionality of the virtual workcell. The physical workcell in the laboratory is also compared side by side with the 3D graphic animation. Through the standard procedure of workflows, robust simulation results can then be obtained for industrial or academic reference. PLCStudio works in conjunction with STEP 7 to allow the user to not only fabricate a virtual 3D representation of the factory floor, but also write and test the PLC ladder logic in a virtual environment before it is run on a physical PLC. PLCStudio's design process can be broken down into three main steps; the component model, logical I/O model and cell model. These three steps are performed in hierarchical phases to produce the final virtual model.

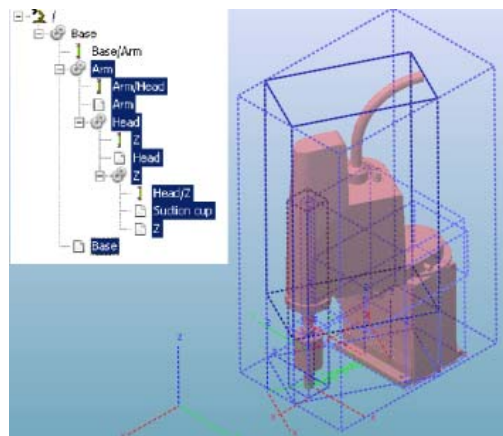


Figure 1: Component Model

## Component 3D Model

The first step in building a virtual cell is to design 3D solids to be used in a component model. The 3D solids can either be generated with PLCStudio's CAD tools or imported from other CAD applications such as AutoCAD or SolidWorks. For devices with movable parts, e.g. robots and pneumatic pistons, axes of motion must be defined and linked to the 3D solids to form joints. Relative or absolute motions involving only one joint can also be defined in the component model. Figure 1 shows an example of the component model representing a Yamaha SCARA robot.

## Logical I/O Model

The logical I/O model in PLCStudio describes the logical operation of input/output signals between the PLC I/O model and the external devices. This is performed by building a symbol table and state diagram for each device in the simulation with inputs and outputs. Output signals from the PLC to the device are defined as state transitions. Input signals from the device to the PLC are defined as internal events of a state. In this way, the PLC gives commands to the device which causes it to make the transition from one state to another while the device gives feedback to the PLC with the device's current state. An example of a state diagram is shown in Figure 2. The state diagram describes a device with two states, A and B. A transition of signal "PLC\_Output\_1" from low to high causes the state machine to transition from state A to state B and perform any actions in the simulation defined by that transition. The system is then at state B. A transition of signal "PLC\_Output\_2" from low to high causes the state machine to transition from state B to state A and perform any actions in the simulation defined by that transition.

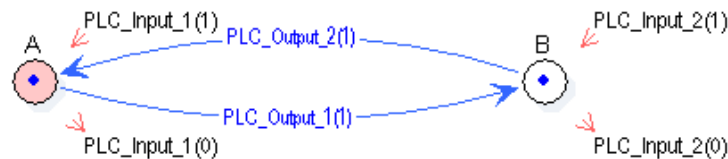


Figure 2: PLCStudio state diagram

The PLC inputs, denoted by arrows pointing into and out of the states, are entering and exiting events. These signals from the device to the PLC are used to communicate which state the device is in. Input signals pointing into a state are entering events, while those pointing out of a state are exiting events. In this diagram, "PLC\_Input\_1" is initially high, set by the entering event of the initial state. When "PLC\_Output\_1" is set high by the PLC, "PLC\_Input\_1" is set low by the exiting event of state A and the system makes the transition from state A to state B. When the transition is complete, the entering event of state B sets "PLC\_Input\_2" high. When "PLC\_Output\_2" is set high, "PLC\_Input\_2" is set low. In other words, when "PLC\_Input\_1" is set high, the device returns to its initial state.

For a simulation, a state diagram is built for each individual device in the system. These state diagrams can either be continuous or non-continuous, determined by whether or not a complete path is formed between two or more states. State diagrams can also contain conditional

statements by having multiple transition statements coming from a state. These fundamental components are combined to describe the reactions of a device to signal inputs from the PLC.

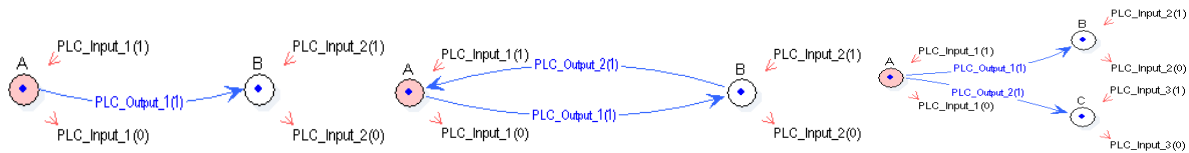


Figure 3: Cell model integrated with state diagram

### Cell 3D Model

The cell model is where the individual component models are combined to form the virtual model of the workcell. This requires all of the components to be imported into the cell model and positioned in the same space as in the physical workcell. If the user is designing a new workcell, the components must be positioned correctly relative to each other. The sensors used in the simulation are also defined in the cell model. There are two types of sensors available: a photo sensor and a limit sensor. The photo sensor detects if an object intersects a line segment defined by two points, direction vector and length. A touch sensor detects if an object is within a 3D area defined by a rectangular prism. In Figure 4, the diagram shows examples of both types of sensors.

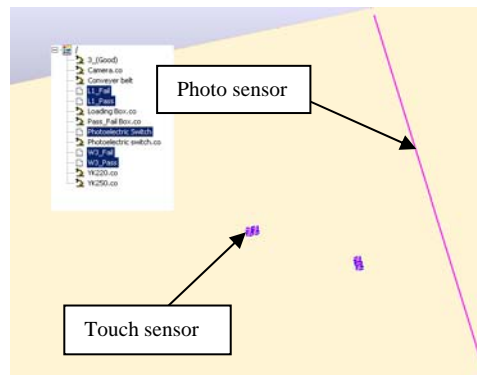


Figure 4: Simulated touch sensors in a cell model

### An Experimental Case Study

To familiarize students with the process of producing a complete simulation in PLCStudio, the students are asked to perform a simple tutorial experiment. The resulting product of this experiment is a system consisting of an actuator that pushes a block back and forth in a cyclic pattern. The actuator is triggered through the emulated PLC by two inputs. To complete this project, the students are required to have fundamental knowledge of the three phases of simulation. The components are designed in the component modeling phase after they are added to the cell model. The actuator and the block are all composed in the PLCStudio program using basic geometric shapes. The sensor mount CAD model can be constructed by the other CAD software programs and then imported into the program as a component model. Both the block

and the sensor mount are static objects, as they have no kinematic function. The actuator is composed of a cylinder housed in a box which extends and retracts. To create the actuator, all 3D components of the actuator are produced and linked together. An axis is created allowing the cylinder to jog in and out of its housing. The students then define the actuator's motions that are later used in the final simulation to shift the block back and forth. Figure 5 shows the final actuator and the connections between the piston and the body.

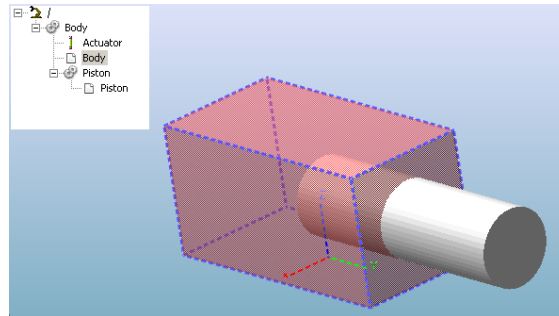


Figure 5: Actuator and the connections between the piston and the body

The cell model or the I/O model could be constructed at this point in the process. However, producing the cell model first simplifies the task to visualize the interactions that are programmed in the I/O modeling phase. The students create a new cell model and import all the previously created 3D components. The students place each component in the cell to create the desired design. Once the layout is completed, the students then place and program the two photo sensors whose length and position are defined. The sensors can be used as inputs to the PLC, but for this simulation, the inputs are toggled by the state transitions that are defined in the I/O modeling phase. Figure 6 portrays the final cell model.

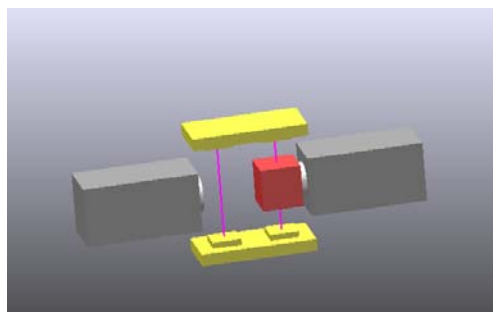


Figure 6: Final cell model

The final step in this process is for the students to create the I/O model to display how the components in the cell interact. A symbol table is first constructed to contain all of the symbol names and the associated hardware addresses of the inputs and outputs in the emulated PLC. Two I/O devices are created; one containing the state diagram for the actuators and the second being a toggle button that starts the simulation. The state diagram for the actuators consists of a simple left and right state which indicates the displacement of the block. State transitions are then defined between the left and right state and another between the right and left state. Each state transition is assigned a symbol (Y1 or Y2 in this case) which is used in the PLC ladder

logic to transition between states. After defining the state transitions as inputs to the PLC, the event properties for each state have to be specified. The state transitions trigger the event properties to send an input to the PLC. Once the inputs and outputs have been set, the effects of the state transition on the cell model are described. This is analogous to constructing a program in a robot controller. For the left to right transition, the left actuator extends while the right actuator retracts. The box is set in the program to attach to the piston as it starts its motion and detach after its motion is completed. The same process is completed for the right to left transition, except that the actuators' motions are switched. Figure 7 shows the symbol table, the device state, and the start button state.

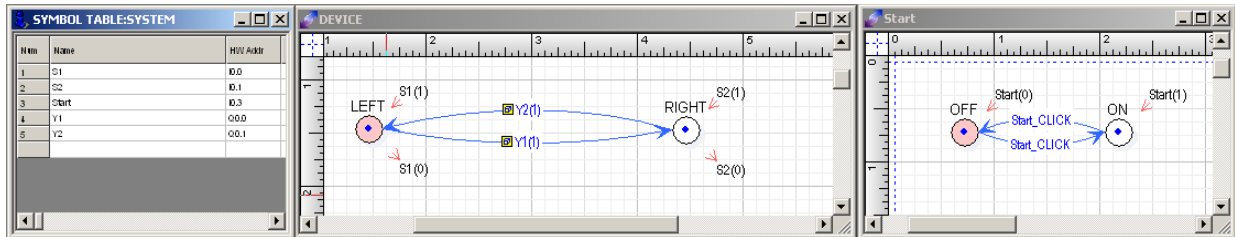


Figure 7: Symbol table, device state, and start button state

After the students have completed the model using the PLCStudio software, they need to setup the STEP 7 PLC software to run the simulation. To configure the emulated PLC, a new project in the STEP 7 program has to be created. The PLC code is later run in an emulation program called PLCSIM found in the STEP 7 software. The symbol table that was initially created in PLCStudio must now be transferred into the STEP 7 project so that the ladder logic can be written using the I/O table that was designed in PLCStudio. The ladder logic program as shown in Figure 8 is an example of what the students would create for the tutorial simulation and download to PLCSIM.

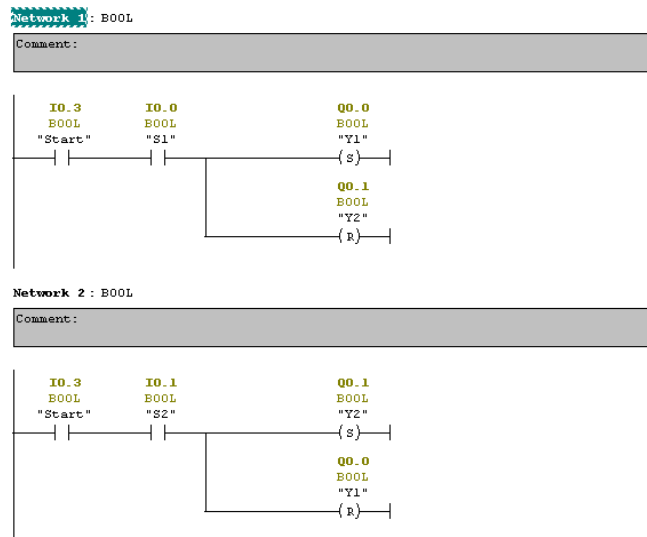


Figure 8: PLC ladder logic diagram downloaded to PLCSIM

Once the students have completed this process, the simulation is ready to run. This simulation introduces the notion of incorporating different components to create a fully integrated robotic system. The standard curriculum in the robotics course focuses on the programming of individual robots, but the students never use the PLC to setup the I/O configuration. By completing this tutorial experiment, the students can gain a better understanding of how an integrated system is developed and controlled.

### Advanced application – Quality control

This experiment performs quality control testing on machined parts and sorts them based on preset tolerances. The devices used in the physical part of the experiment are a YK220X SCARA robot, a YK250X SCARA robot, a machine vision camera, a conveyor belt, and a photoelectric sensor. The layout of the components in the workcell can be observed in Figure 9.

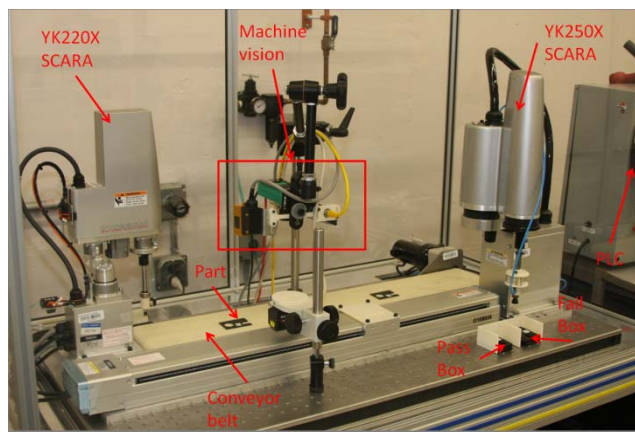


Figure 9: Robotic cell equivalent to PLCStudio virtual system

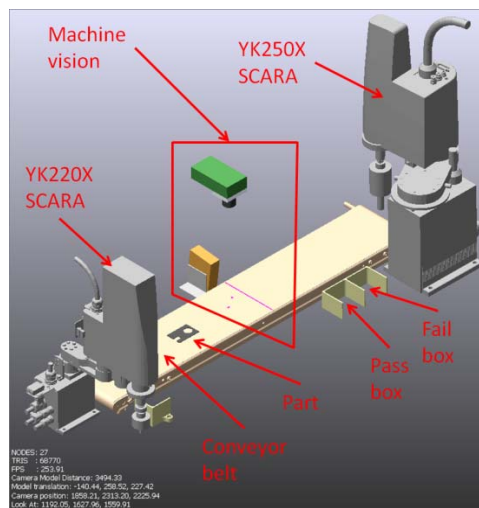


Figure 10: Quality Measurement Cell



As opposed to the first experiment, students are responsible for only constructing and verifying the PLC logic in the virtual cell without designing the components. This is due to the fact that the complexity of constructing the virtual cell and creating the I/O model is beyond what could be expected of students in a limited class time. Figure 10 shows the assembled cell model.

Both the virtual and the physical cells are equivalent in their layouts and components. Same as the physical cell, the virtual cell contains models of the two SCARA robots, the photoelectric sensor, and the machine vision camera. The difference between the two cells is the implementation of the machine vision camera. The left image in Figure 11 shows the parameters used in the machine vision camera's program to measure parts. The machine vision camera can take a picture and analyze it using an on-board processor and then transmit a pass or fail signal to the PLC. In Figure 11, the diagram on the right shows how the machine vision camera sensor is simulated in PLCStudio.

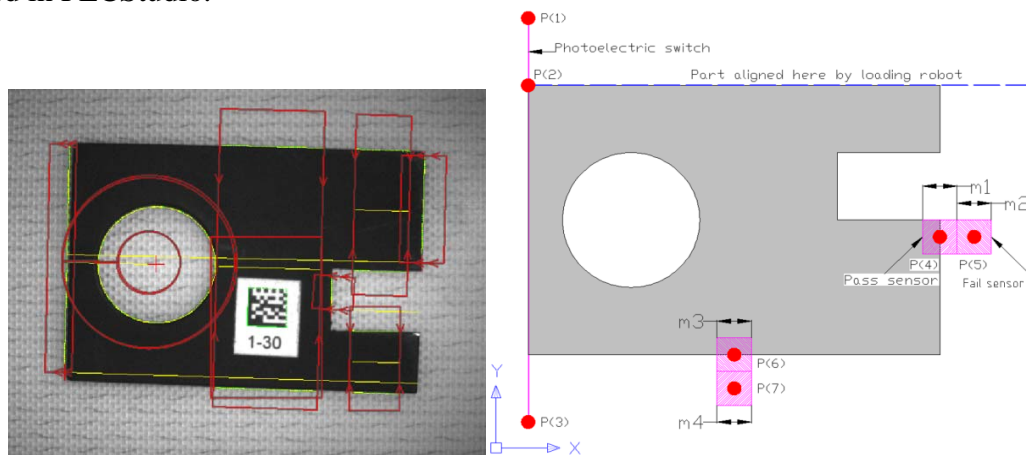


Figure 11: Simulated machine camera in PLCStudio

The virtual machine vision camera is a combination of photo and touch sensors that are used to measure dimensions of a part. To simulate the machine vision camera, four touch sensors and one photoelectric sensor are used. Sensors m1 and m3, shown in Figure 11, are the pass sensors. Sensors m2 and m4 are the fail sensors. The pass or fail signal sent to the PLC determines where the YK250X robot places the measured part.

Students are given the cell model and I/O model for the simulation. To run this simulation the students must create the ladder logic program that uses the input data from sensors and sends the output data to the robots. Figure 12 shows the ladder logic utilized by PLCStudio on the left and the logic used by RSLogix on the right. It is necessary to use RSLogix because PLCStudio is only compatible with STEP 7 software which works with Siemens PLCs while the PLC used in the lab is made by Allen Bradley which utilizes RSLogix.

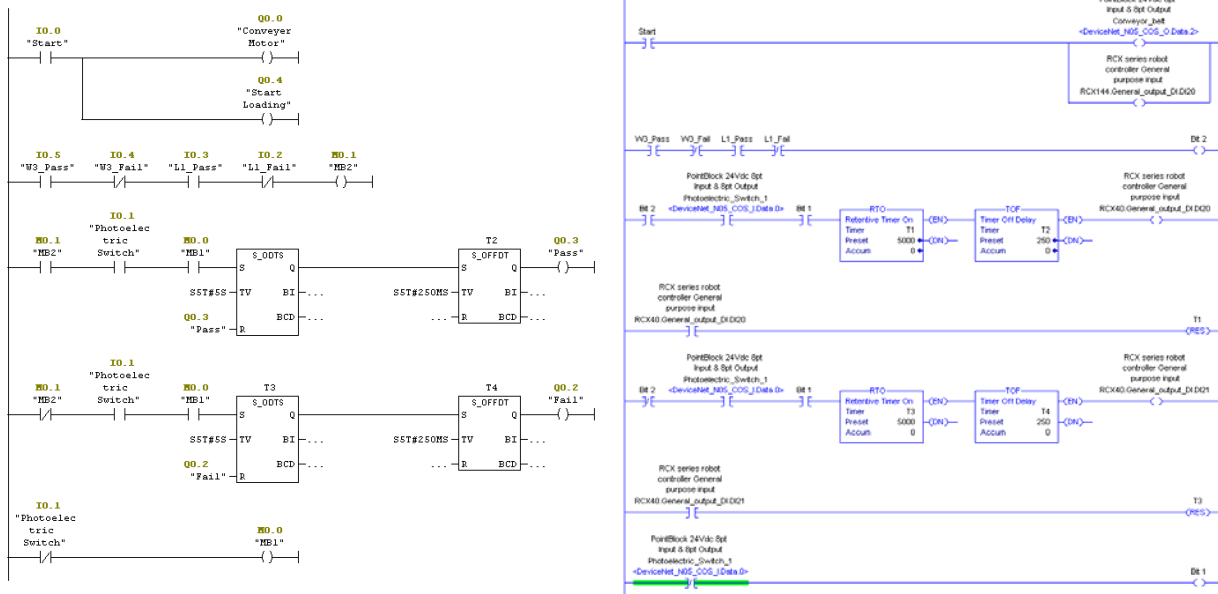


Figure 12: Ladder logic

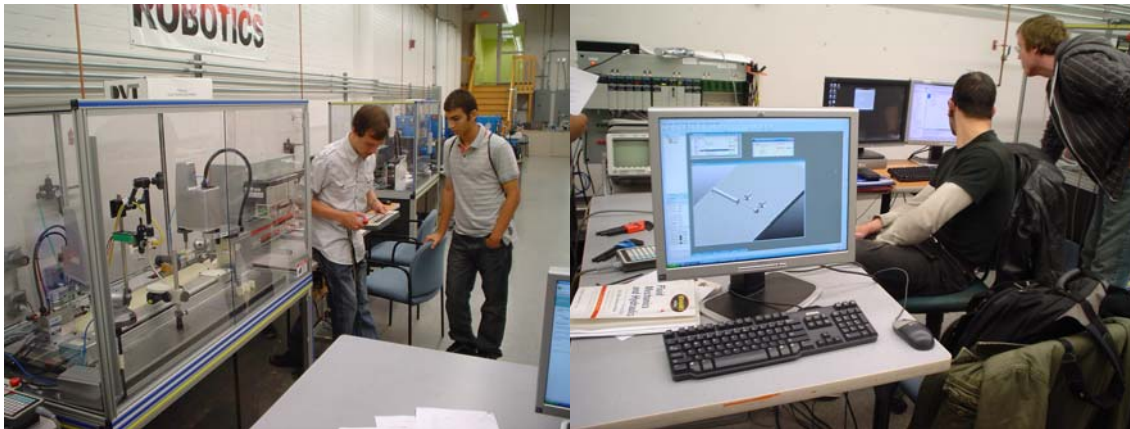


Figure 13: Students building a robotic cell and programming automation by PLCStudio

PLC programs can be tested in PLCStudio and implemented in the robotics and mechatronics laboratory. The robot sequence timing is measured in the laboratory experiment as shown in Figure 13. The students collect data to measure the cycle time of the robotic cell. If the robot receives a pass signal, then it runs the “pass” routine which sorts the part into the pass bin. If it receives a fail signal input then it runs its “fail” routine which sorts the part into the fail bin. A timing diagram of the sequence of events performed using PLCStudio is displayed in Figure 14.

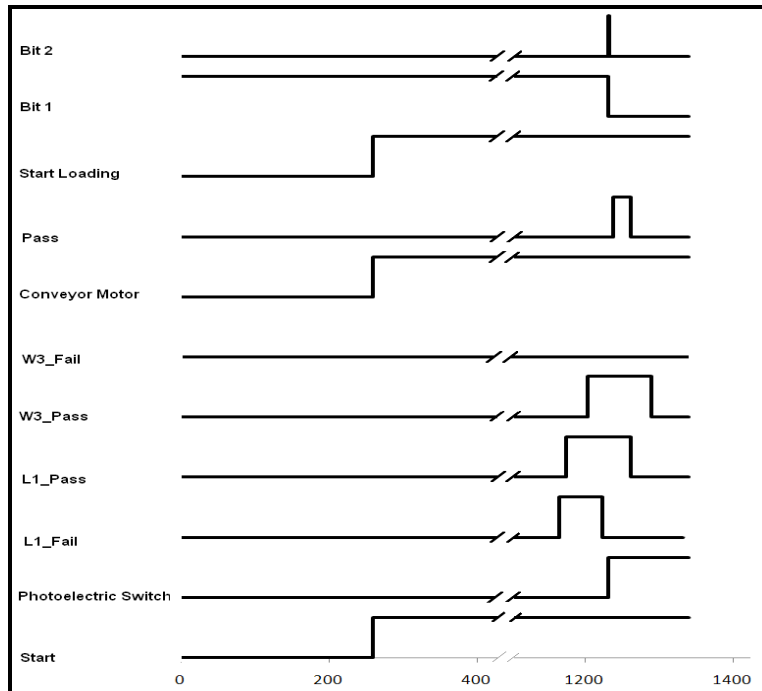


Figure 14: PLCStudio timing diagram

## Conclusion

As technology continuously advances, it is essential that students stay up-to-date with these advancements. PLCStudio software is an enhancement in industrial robotic cell simulation. It allows the integration of kinematic models and enables the user to write and test PLC code that can be utilized in a manufacturing environment. The experiments described in this paper allow the students to gather a firm understanding of how this new software works and its possible applications to a modern production environment. PLCStudio offers a unique way for students to gain hands-on experience through producing and simulating their own work cells. It is a proposed tool for educating tomorrow's industrial and manufacturing engineers.

## Acknowledgement

This work has been partially supported by the Korea Association of Small Business Innovation Research (KASBIR) under the International Joint Research and Development Program, Contract Number #00035088. The authors wish to express sincere gratitude for their financial support.

## Bibliography

1. Hyeong-Tae Park, Jong-Geun Kwak, Gi-Nam Wang, and Sang C. Park, "Plant model generation for PLC simulation," *International Journal of Production Research*, Volume 48, Issue 5, pp. 1517 – 1529, 2010.

2. Jin-Shyan Lee and Pau-Lo Hsu, "A systematic approach for the sequence controller design in manufacturing systems," *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 7-8 / April, pp. 754–760, 2005.
3. Klingstam P and Gullander P, "Overview of simulation tools for computer-aided production engineering," *Computers in Industry*, Vol. 38, Issue 2, pp. 173–186, 1999.
4. Elisabet Estévez, Marga Marcos, and Darío Orive, "Automatic generation of PLC automation projects from component-based models," *Int J of Adv Manuf Technol*, 35:527–540, 2007.
5. Iwata K, Onosato M, Teramoto K, and Osaki S, "A modeling and simulation architecture for virtual manufacturing systems," *ANN CIRP*, 44(1):399–402, 1995.
6. Ye L. and Lin F., "Virtual system simulation – a step beyond the conventional simulation," *22nd Int. Conf. on Computer and Industrial Engineering*, pp 304–306, 1997.
7. Park, Chang Mok, Park, Sangchul, and Wang, Gi-Nam, "Control logic verification for an automotive body assembly line using simulation," *International Journal of Production Research*, pp. 1-19, 2008.
8. Chang Mok Park, Sangchul Park, and Gi-Nam Wang, "A PLC programming environment based on a virtual plant," *International Journal of Adv Manufacturing Technology*, Vol. 39, pp. 1262–1270, 2008.
9. Richard Chiou, Yongjin Kwon, and Prathaban Mookiah, "Manufacturing E-Quality Through Integrated Web-enabled Computer Vision and Robotics," *International Journal of Advanced Manufacturing Technology*, Volume 43, Numbers 7-8, pp. 720-730, August, 2009.