Visual validation of PLC program using virtual simulator

R Badarudin¹, and D Hariyanto¹

¹Dept. of Electrical Engineering, Universitas Negeri Yogyakarta, Indonesia

E-mail: rohjai.badarudin@uny.ac.id

Abstract. This article discusses the programmable access interface between visual studio and the Siemens PLC simulator (S7-PLCSim). The communication interface adopts S7-ProSim COM object control. By using the S7-ProSim, we can write Microsoft Visual Studio-based software application code to change the PLC input value, read the PLC output value, change the position of the PLC control mode switch, and other controller tasks. Visual studio-based virtual modelling software applications can be used to validate PLC programs in real time. So that virtual lab-based PLC program validation is possible to develop based on Microsoft Visual Studio with a PLC interface using the S7-ProSim COM object.

1. Introduction

During the Covid-19 pandemic, hands-on learning was impossible. Learning during the Covid-19 pandemic was carried out online through a digital platform. One of the most important factors in the vocational learning process is the practice of skills [1]. The right media and appropriate learning methods must be applied to support the skills of students. Remote practical learning requires virtual tools used to validate the students home works. PLC programming training can be carried out by simulation using a virtual environment. Programming logic planning, program checking, and sample program modification can easily be done based on virtual environment [2]. Simple PLC program validation can also be performed using the simulation tools provided by the PLC programming software without being connected to the PLC hardware. Several types of PLC programming tools are equipped with simulation menus.

This paper presents the solution of validation plc program using visual simulation hardware. Simulation hardware has been development using windows form development software. Using 2D graphic simulation software, this plant can movement based on commands from PLC output and the sensor will activate automatically when it is triggered and for the push button will be activated manually by mouse click event. S7-PLCSim is a PLC simulator tool for validating Siemens PLC programs without connecting to Siemens PLC hardware [3]. S7-PLCSim provides a simple interface of displaying CPU operating modes and changeable control variables, such as input and output variables. Simple PLC programming may be solved using the S7-PLCSim interface, there are several complex PLC programming experiments where program validation using S7-PLCSim is no longer possible. Program validation is carried out with the real hardware and using similar plant hardware according to the program being made. This condition requires materials that are very expensive and also take up a lot of space [4].

PLC program validation techniques with virtual plant-based simulations are increasingly developing and becoming the latest trend. One of the advantages of the virtual plant model is simplicity, ease of modelling, easy model changes and easy to do with other software such as PLC software and 2D or 3D
graphical simulation software [5]. So that the development of a virtual plant based on visual studio graphic design with PLC-based control can be realized with the S7-Prosim and S7-PLCSim object COM interfaces. The simulation is fully executed on a PC device or a Siemens PLC programming. S7-PLCSim can be used to simulate Siemens PLC programs for S7-300, S7-400, and WinCC controllers. S7-PLCSim provides a simple interface to the Siemens PLC user program for monitoring and modifying different objects such as input and output variables. S7-PLCSim provides an interface for graphic design to view and modify program variables, run S7-PLCSim programs in single or continuous scanning mode, and change the operation mode of the simulation controller. S7-PLCSim also includes a COM object called S7-ProSim which provides programmable access to S7-PLCSim. With the S7-ProSim, we can write windows form-based software application code to perform tasks such as changing input or output variables, the position of the key switch mode of the S7-PLCSim, running control programs in single scan mode, reading or writing values of controller values, and many more [6] [7][8]. S7-ProSim Com Object can be linked to a visual-studio based windows form application using a specific programming code. The windows form application in visual basic can be used to change PLC input variables or read PLC output variables. S7-ProSim is used for two-way communication interface between windows form application in visual studio and S7-PLCSim simulator device.

2. Proposed System

Elements built based on Microsoft Visual Basic software include switches, actuators, sensors, and moving parts. A simple model of the switch model is shown in Figure 1. with the action of manual operation using the mouse click event. Switch is a variable that can be changed the value of the variable, for example the switch variable is S1. The initial or normal position of a switch is the first position (P1), the value of the variable S1 is zero, the variable S1 will remain equal to zero until there is an action of change. If the change action is applied to switch S1, then the change position is indicated by another position (P2) with the variable value S1 is one.

![Figure 1. Switch model system architecture](image)

A simple model of the actuator model is shown in Figure 2 with the change action calls the actuator variable name. The actuator is a moving part that is named to differentiate between one moving part and another. The movable part name is a variable so that it can be called upon to give the change action. For example, the PLC output signal is stored in variables ‘A’ and ‘B’. The initial or normal position of an actuator is the first position (P1 ’), and the final position of the actuator is the second position (P2’). The change action is carried out by calling and executing the name of the moving part, so that the value of the variable ‘A’ must be compared with the actuator position variable. The actuator position variable is to compare the position of the last actuator with the change action to be given, so that there is no conflict in the coding of the program.

A simple model of the sensor model is shown in Figure 3 with the change in action in the form of variable values on a graphical examination. In actual conditions, the part detected by the sensor is able to sense light and touch. Sensing models in software use graphical checks to replicate physical sensing. The state of the sensor is changed by the state of a certain graphic part. Assuming a sensor checks the state of a moving part, for example the movement of an actuator. The minimum and maximum positions
of double acting cylinders are detected by sensor variables 'a' and 'b'. Graphically, a double acting cylinder has two end positions, namely the minimum position and the maximum position. The variable value 'a' equal to one when the double acting cylinder frame is at a minimum position, and the variable value 'a' equal to zero when the double acting cylinder frame is at a position other than the minimum. The variable value 'b' equal to one when the double acting cylinder frame is at the maximum position, and the variable value 'b' equal to zero when the double acting cylinder frame is at a position other than the maximum.

![Diagram](image1)

**Figure 2.** Actuator model system architecture

3. Finding and discussions

3.1 PLC Interface
The PLC interface and virtual model are developed using commands from the S7-Prosim function library. Input variable values, namely sensors and switches, are sent to S7-PLCSim with write input point command. Output variable values from S7-PLCSim are read the output values to manipulate the actuator variable values on the model with read input point the command. The command codes are shown in Table 1.

| Command                          | Coding                                                                 |
|---------------------------------|------------------------------------------------------------------------|
| Write input point from visual    | [Sub Write Input Point (Byte Index as long, Bit Index as long, Data) |
| basic to S7-PLCSim              | as long:].                                                             |
| Read output point from S7-PLCSim| [Sub Read Output Point (Byte Index As Long, Bit Index As Long,          |
| to visual basic.                | Data Type As Point Data Type Constants, Data) As Long:].             |

The data sent to S7-PLCSim is always updated when there is a change in the variable values of switches and sensors in the virtual model. The reading of the output value from S7-PLCSim is also done repeatedly. The reading interval is done as quickly as possible so that the response when there is a change in the output of the S7-PLCSim can be immediately responded to by the model.

3.2 Architecture and software implementation
Program communication channels are created between the Simatic manager-S7 program as a PLC programming device, S7-PLCSim as a virtual PLC hardware device and a virtual model as a controlled object or plant. This allows user programs to be validated and errors detected. Variable data values are continuously exchanged between the S7-PLCSim and the virtual model because the output value affects the actuators in the virtual model, and the switch or sensor affects the program input. Figure 4 shows the data flow between the PLC programming device, the S7-PLCSim module and the virtual model.
Examples and illustrations are shown in Figure 5. The software is used to describe the performance of a virtual model, by adopting an electro-pneumatic-based moving machine that is controlled using a PLC. There are three double acting cylinder actuators with electro-pneumatic valve control. The moving machine can virtualize the process of moving its parts so that the user program can be validated.

**Figure 4.** Data flow between the PLC programming device, the S7-PLCSim module and the virtual model.

**Figure 5.** Virtual process model of moving machine
4. Conclusion

PLC program validation can be done by real time simulation between S7-PLCSim and visual studio based virtual models. The S7-PLCSim communication interface and virtual model use the S7-ProSim COM object control in the visual studio. The PLC programming device uses the Simatic Manager Step-7 to program the Siemens PLC module. Virtual process models with various types of plant models can be developed based on a visual studio to validate PLC programs in real time.

5. References

[1] Huang, X. (2011, August). A mechatronics educational laboratory platform for manufacturing automation based on Googol FMS. In Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology (Vol. 1, pp. 389-392). IEEE.
[2] Park, C. M., Bajimaya, S. M., Park, S. C., Wang, G. N., Kwak, J. G., & Han, K. H. (2006, November). Development of virtual simulator for visual validation of PLC program. In Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on (pp. 32-32). IEEE.
[3] Amida, A. (2013). Programming and testing of storage conveyor in a virtual.
[4] Damas, M., Pomares, H., Tarifa, J. A., Jiménez Mesa, G., & Roldán Herencia, M. V. (2006). Virtual laboratory for the supervision and control of scale models of industrial processes. Current Developments in Technology-Assisted Education. Badajoz: Formatex, 1-342.
[5] Park, C. M., Bajimaya, S. M., Park, S. C., Wang, G. N., Kwak, J. G., & Han, K. H. (2006, November). Development of virtual simulator for visual validation of PLC program. In Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on (pp. 32-32). IEEE.
[6] Fang, H., & Tang, W. (2012, April). Research on communication between computer and PLC based on VB. In Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on (pp. 2795-2797). IEEE.
[7] S7-PLCSIM Version: 5.4, Operating Manual, Edition 07/2011.
[8] Wischniowski, R., & Freund, E. (2004, April). COSIMIR Transport: modeling, simulation and emulation of modular carrier based transport systems. In Robotics and Automation. 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on (Vol. 5, pp. 5171-5176). IEEE.