

## USING EMULATION TO DEBUG CONTROL LOGIC CODE

Richard Phillips  
Bonnie Montalvo

E<sup>2</sup>M/Polytron  
3300 Breckinridge Blvd.  
Duluth, GA 30096, USA

### ABSTRACT

Debugging the custom code developed for a logic controller is a crucial and high risk step for any production line startup. Emulation, the process of building a virtual 3D production line responsive in real-time to a logic controller, provides the controls engineer early access to the line. Using this safe, easy, emulated testing environment reduces startup time by up to 50%. This case study will examine our strategy for implementing emulation as applied to a consumer product packaging line.

### 1 INTRODUCTION

During Detailed Design of a material handling or production line, the controls engineer must write new control logic to operate the automated production line. This process requires a high risk, critical stage of debugging which commonly takes place on the factory floor – at a time when the manufacturer is eager for production. Emulation, the process of building a virtual 3D production line that communicates with a Programmable Logic Controller (PLC), allows the controls engineer(s) to debug code in parallel with the installation process. For this reason, emulation has the ability to facilitate rapid startup.

In this case study, a 3D model was constructed using the physics-based software, Emulate3D Controls Testing 2009 (Demo3D; Emulate3D Ltd. 2010). Demo3D allows the programmer to customize life-like manufacturing lines with equipment that is responsive to the control logic in real-time. In this paper, we discuss an implementation strategy for constructing a Demo3D model for the purpose of debugging a logic controller. We will also present a case study in which our implementation strategy was directly applied to emulate a consumer product packaging line.

### 2 BACKGROUND

Our case study will examine the installation of a new state-of-the-art bundler. The layout allows for three different lines to feed into the bundler with ten different package sizes. Depending on the size and feeding lane, the PLC activates recipes that set conveyor speeds and enable equipment, allowing each package to reach the bundler and warehouse with the correct orientation.

The line consists of the following equipment:

- Static Merge: Merges product from two lanes into one lane
- Orienter: Orients packs by 90 degrees
- Gripper Upender Elevator: Orients packs by 90 degrees while raising to adjoining conveyor
- Laner: Sends product down one of multiple lanes
- Bundler: Creates a variety of bundled packs depending on PLC recipe

When asked why emulation was included in this project, the project manager stated: “The project team had heard of other recent case studies within their company where a successful start-up was largely attributed to the use of emulation. Emulation allowed the logic program that controlled the line operation and integration to be debugged prior to the startup phase. This reduced the risk during startup for increased costs and schedule delays, in addition to allowing the project team to virtually “see” the system in operation and provide their feedback without the challenges present during the startup phase.”

### **3 IMPLEMENTING AN EMULATION**

As in this case study and in any emulation project, the simulation engineer performs seven key steps working alongside the controls and mechanical engineers and project manager, as well as client representatives. Below, we present a strategy for the successful completion of an emulation.

#### **3.1 Scope Definition**

First, the project team determines which aspects of the model will be emulated and which will be simulated. Most often, the controls engineer needs to test the overall functionality of the line and not individual equipment. In this instance, a hybrid approach is ideal. A hybrid model consists of emulated conveyors and simulated unit ops with PLC handshaking.

#### **3.2 Data Collection**

To construct the model in Demo3D, the simulation engineer begins by gathering data that will assist in the development of the model:

- **Line Layout:** Once imported into the Demo3D, an AutoCAD layout facilitates equipment placement.
- **Functional Description:** A mechanical description of each piece of equipment on the line is necessary to construct physics-based replicas in the model.
- **Control Logic Tags:** To communicate with the line equipment, the logic controller sends and receives data and commands (or I/O bits) by way of addresses called tags. The simulation engineer must know what information will be sent and received and its corresponding tag
- **Product Specifications and Rates:** Necessary for realistic product flow throughout the model

#### **3.3 Equipment Construction**

Demo3D offers catalog items such as conveyors, transfers, and turntables that are easily adapted to meet equipment specifications. Demo3D also offers QuickStart items such as Forklifts, AS/RS, Palletizers, and human workers that contain more complex logic. The task of the simulation engineer is to modify these items and create life-like replicas. Demo3D is a physics-based, discrete modeling environment, so the physical dimensions and properties of each modeling element, or visual, must match those of their actual counterpart. Demo3D scripting is a variation of Microsoft’s JScript language that incorporates visuals with independent properties. Each property has attached script that is responsive to PLC changes. Each visual may also require additional coding to meet the individual equipment requirements. The coding process is noticeably different for simulated and emulated equipment.

##### **3.3.1 Simulation**

In simulation, the simulation engineer writes logic to control the activation of equipment and movement of loads. To create a visual that performs like the real equipment, the simulation engineer must have a

complete understanding of the logic narrative that guides the equipment. Every instance where the PLC receives and sends signals must be simulated with script.

### **3.3.2 Emulation**

In emulation, the simulation engineer writes logic to bridge the model visuals and PLC. Every Demo3D visual contains default properties that are available for communication with the PLC. What happens when the default property does not meet the tag-specific requirements? For example, a conveyor has a default property that activates the conveyor motor but the PLC requires the conveyor motor to turn on and run forward or reverse. In this case, the simulation engineer must create a custom property. A custom property allows the simulation engineer to write script that directs the conveyor according to the signals received from the PLC. The simulation engineer programs the model to interact realistically with the PLC.

### **3.4 Model Construction**

Once equipment visuals are constructed, the emulation model may be assembled. Using the AutoCAD layout and Demo3D's drag-and-drop functionality, a simulation engineer can quickly construct a layout from his catalog. By placing "load creators" at the start of each line and "load deleters" at the end, product is introduced at the beginning, and removed from the end, of the line. Each load creator has a customizable release frequency and allows for the creation of multiple distinct loads.

### **3.5 Communication**

To establish communication, it is necessary that the PLC run as it would on the factory floor. If hardware is not available, the PLC may be emulated. For Rockwell PLCs using RSLinx Gateway (Rockwell Software 2010) or another communication server, the tag address must point to the PLC address. Within Demo3D, the tag browser links PLC tags to their corresponding visual and its property. Once the logic is running on the controller hardware or emulator and connection is established with Demo3D, the PLC is ready to run its virtual factory.

Correctly configuring communications requires about a day of work depending on the size of the model and complexity of the logic. However, communication between the PLC and model often causes many larger problems within the model. This is because communication equipment is susceptible to interference and requires a variety of case specific adjustments.

While not applied in this case study, at this time it is possible to establish communication between the Human Machine Interface (HMI) and PLC and operate the model exclusively by HMI. Although the addition of an HMI adds an extra layer of complexity, the HMI allows for emulation to fully mimic the operator experience.

### **3.6 Debug**

During the debug phase, the controls engineer and simulation engineer begin testing their logic simultaneously. It is often necessary to determine the source of logic problems because they may stem from the PLC, model, hardware, or communication. The debug phase may take multiple weeks depending on the size and complexity of the model and controller. Once all problems have been resolved, the controls engineer can run test scenarios and debug the controller as he would on the factory floor.

### **3.7 Logic Review**

Finally, the controls and simulation engineers demonstrate each scenario to confirm that the controller meets factory floor standards. At this time it is possible to see how the future line will operate while verifying the robustness and capability of the logic controller. If any problems are found, the debug and review operations are repeated until the logic is up to operational standards.

## **4 APPLICATION**

At the start of our case study, the project team already had a robust Demo3D catalog of simulated equipment, including a Static Merge, Bundler, Laner, and Gripper Upender Elevator. This greatly reduced the time required to construct equipment as the simulated catalog items were easily tailored for the emulation. Equipment was tailored by adjusting simulation logic for correctness and then coding PLC handshaking bits. An additional requirement included that the model be quickly re-configured for each product and line change. This was achieved by creating selection buttons within Demo3D for both the product and the line. When selected, the corresponding script would configure any manual operations within the model. The selection buttons also facilitated communication with the PLC, sending the recipe number to the PLC. This process allowed for 30 different line configurations to be tested with negligible time between product and line changes.

The simulation engineer also created scripting to simulate jam, emergency stop, and operator error scenarios – all activated by buttons within Demo3D. Jam scenarios were especially painless compared to testing on the factory floor. Real-time interaction with the model allowed for the removal of jammed product by mouse selection. As for accumulated product, it vanished by resetting the model. By testing each product and line configuration as well as all scenarios, the controls engineer could check the robustness of the PLC logic over the course of a week. The logic review was performed with project team personnel within a few hours.

For this emulation, the control logic was written in RSLogix 5000 (Rockwell Automation 2010a), emulated in a virtual chassis using RSLogix Emulate 5000 (Rockwell Automation 2010b) and connected to Demo3D through RSLinx Gateway (Rockwell Software 2010).

## **5 BENEFITS**

The project manager saw many benefits to having a virtual start-up for the line: “The Emulation model allowed the project team to see the line controls in operation on a virtual line, and provide their feedback. It was much easier to discuss how to best control each scenario in a meeting room environment than on the factory floor where noise, logistics, and distractions made such discussions difficult. The model allowed the team to run all of the scenarios as many times as required, make the agreed to changes, and reach consensus that the controls system operated as intended.”

Emulation has the ability to reduce time, risk, and costs associated with the startup of a production line. By providing the controls engineers early access to the virtual manufacturing system, they can work in parallel with physical installation rather than after it is complete. We have often found that this results in a 50% reduction of the time and resources required for a startup on-site – as well as mitigating many risks. These benefits translate to less costly projects that provide a quicker time-to-market, and a steeper startup curve. Emulation also provides an environment which is ideal for optimizing and modifying logic without the worry of lengthened projects or expensive change orders.

Finally, emulation offers a chance to carry out more sustainable practices. By virtually testing scenarios, fewer products are wasted during installation. Also, a 50% reduction in startup time reduces travel expenses and associated carbon emissions.

## **6 CONCLUSION**

In this case study, we proposed an implementation strategy for constructing a Demo3D model to debug a logic controller. We also demonstrated that the application of this strategy to an automated production line, using emulation to debug the control logic, allowed for debug and acceptance of the line controls prior to startup. This translated to cost savings, reduction in schedule, rapid startup at a reduced cost, and ultimately faster time to market for our customer.

## REFERENCES

- Emulate3D Ltd. 2010. Demo3D – Physics-Based Industrial Demonstration, Controls Testing & Machine Simulation Products. Available via <<http://www.demo3d.com/>> [accessed April 15, 2010].
- Rockwell Automation. 2010a. Design & Configuration: RSLogic 5000. Available via: <<http://www.rockwellautomation.com/rockwellsoftware/design/rslogix5000/>> [accessed April 15, 2010].
- Rockwell Automation. 2010b. Design & Configuration: RSLogic Emulate. Available via: <<http://www.rockwellautomation.com/rockwellsoftware/design/rslogixemulate/>> [accessed April 15, 2010].
- Rockwell Software. 2010. RSLinx Technical Data. Available via: <[http://literature.rockwellautomation.com/idc/groups/literature/documents/td/linx-td001\\_-en-p.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/td/linx-td001_-en-p.pdf)> [accessed April 15, 2010].

## AUTHOR BIOGRAPHIES

**RICHARD PHILLIPS** is a Project Manager at E<sup>2</sup>M/Polytron, Duluth, GA. He is a Project Management Professional (PMP) and Professional Engineer (PE) with over 15 years experience delivering system integration and automation services to manufacturing clients. He received his Bachelor and Master of Science degrees from Auburn University in Electrical Engineering. His email address is <[rphillips@polytron.com](mailto:rphillips@polytron.com)>.

**BONNIE MONTALVO** is a System Analytics Simulation Engineer at E<sup>2</sup>M/Polytron, Duluth, GA. She received her Bachelor of Science from Georgia Institute of Technology in Industrial and Systems Engineering. She has created simulations and emulations in both Automod and Demo3D for various industry clients. Her email address is <[bmontalvo@e2m.com](mailto:bmontalvo@e2m.com)>.