White Paper on How to Implement a Moore State Machine (MSM) on a Programmable Logic Controller (PLC).

By Eric Rouse

A state machine is a powerful tool for controlling sequenced action. In fact, any system (a collection of devices with input and/or output datasets) can be modeled by a collection of superimposed, asynchronous, finite state machines.

More specifically, a Moore State Machine (MSM) is a particular state machine model comprised of a finite amount of states. What makes an MSM unique is that its outputs are based on the current state, not by input values. This makes it a very useful model in PLC programming.

Inputs are used to determine state transitions and outputs are activated based on state. This makes programming a process as easy as breaking it into parts, diagramming it and implementing the diagram in PLC code.

For purposes of illustration, let's take a simple process of fixturing a part. The customer has asked that after an operator loads the part the machine should see that it is datumed properly and then clamped in place. Once clamped, a robot executes a drill/tap/debur operation. After the robot is done the part needs to be ejected to the finished parts bin.

Step 1 – break into parts: This simple process must be broken into its smallest actionable parts.

1. We detect a part is loaded. This will require a part-present input.
2. We push the part against its datums. This will require air cylinder tappers, solenoid valve outputs and cylinder position inputs.
3. We clamp the part. This will require a clamp cylinder, solenoid valve outputs and cylinder position inputs.
4. We need to release the datum holds. Already have the required inputs/outputs from step 2.
5. We need to communicate with the robot that we are ready, and it needs to communicate with us when it is done. This requires data passed over an Ethernet network.
6. We need to unclamp the part. Already have the required inputs/outputs from step 3.
7. We need to release put the part in the eject bin, this requires an eject conveyor. Requires motor starter.

Step 2 – create a diagram: The state diagram is a simple way yet powerful way to document and define the process. It is helpful to list the inputs that will transition from each state and the outputs that each state will execute. See Figure 1 below.

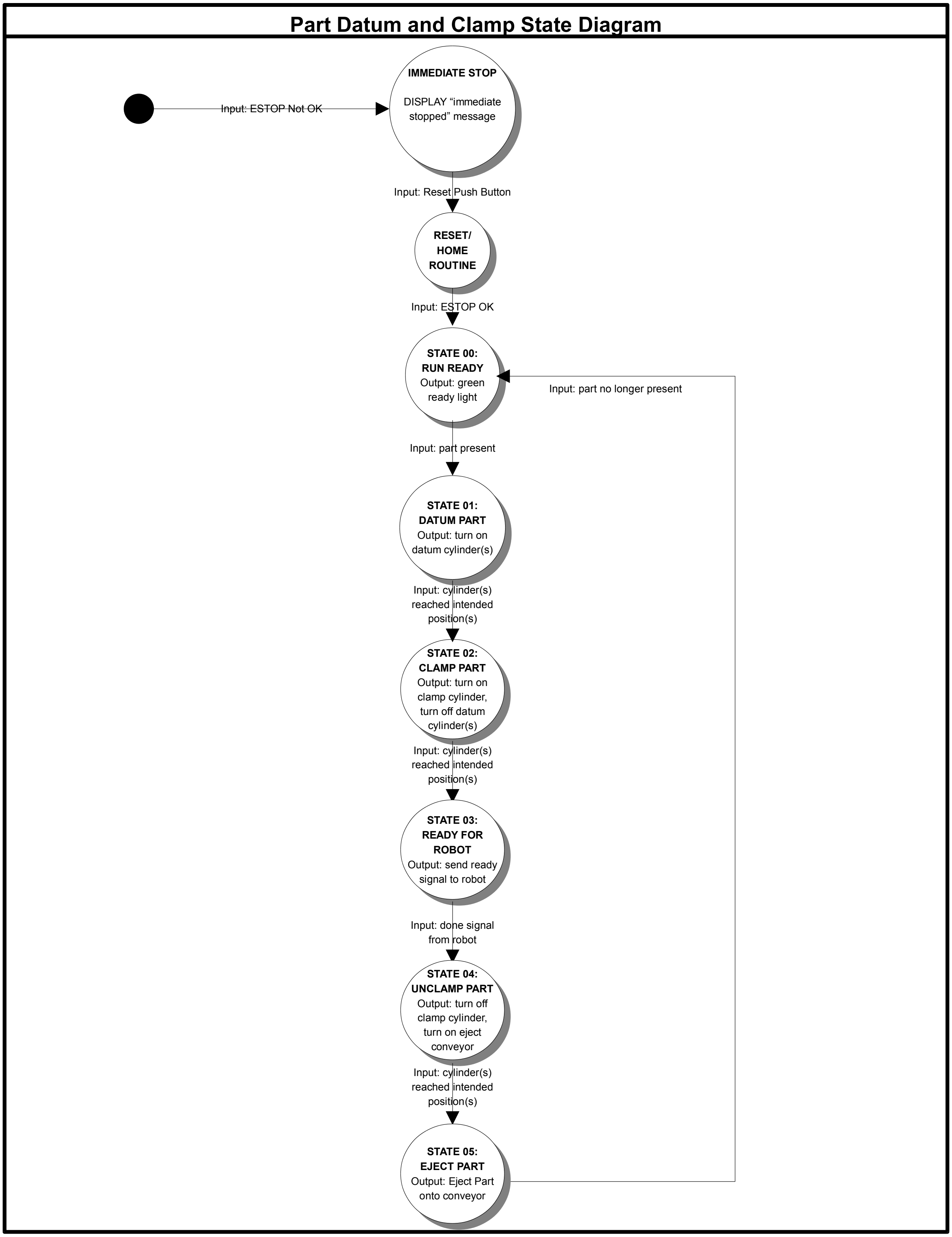

Figure 1 - Moore State Diagram, Graphically Represented.

Step 3 – implement the diagram in code. The PLC code implementation is what makes this diagram so powerful. It is a simple latch/unlatch of transitions (Process_NewStates) based on inputs. Then, a simple latch/unlatch of outputs based on states (Process_CurrentState). See Figure 2.

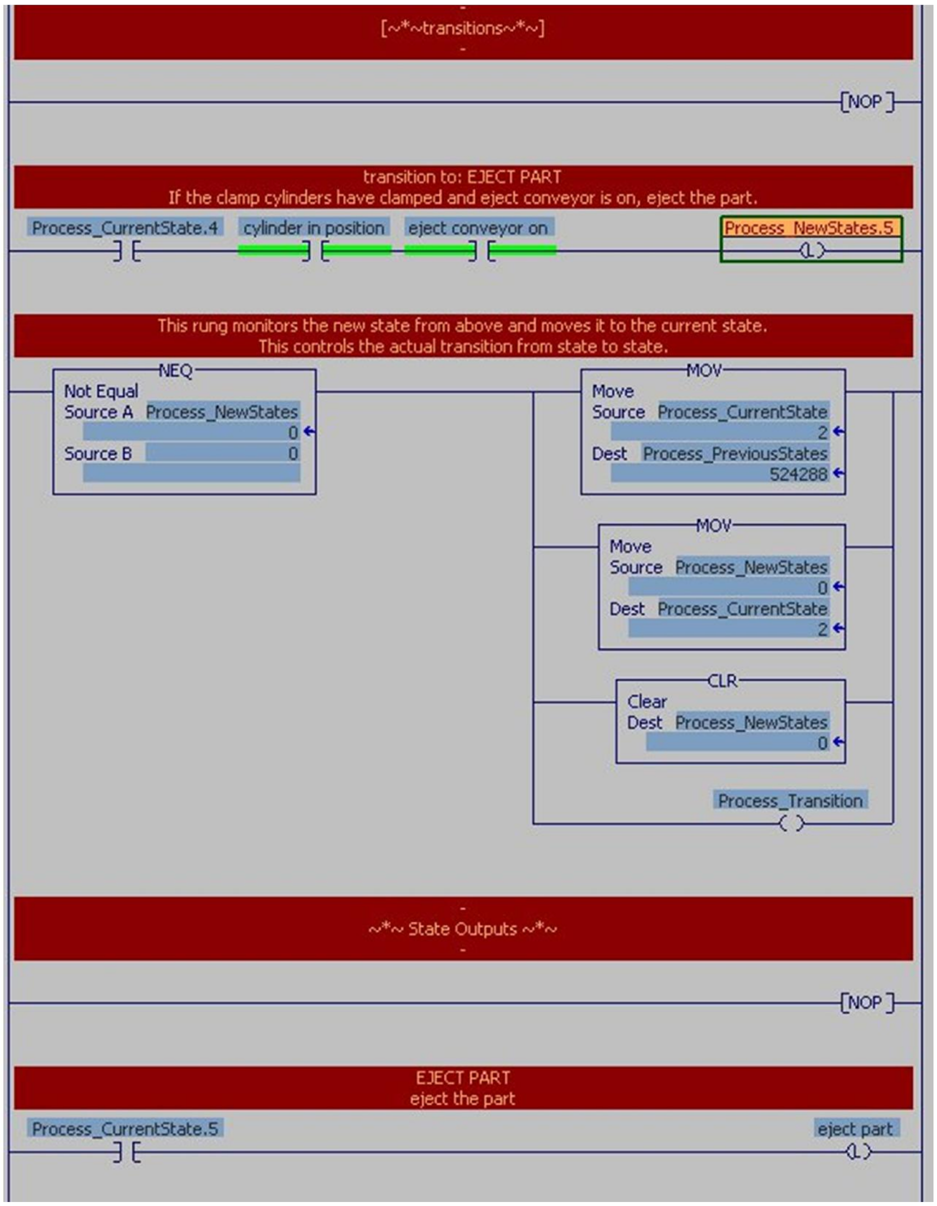

Figure 2 - Part of MSM Implemented in Ladder Logic

The real power of this implementation is the middle rung; it controls the actual transition, allowing only one transition at a time. It is the gatekeeper that maintains the state machine structure and order.

Thus, the structure and simplicity of a Moore State Machine is quick to design and implement. And yet its use and structure are dynamic and powerful. Break the process down, diagram and then code.