

PLC Programming Techniques 2

PLC Programming Techniques – Part 2

PLC Programming Techniques 2

Note:

Ideas presented here are just some thought starters on programming and are not meant to be all-inclusive or critical of others. Coding has many common elements; however, programming techniques are unique to each individual programmer and therefore no one “right” answer exists.

Program use such as user acceptance testing and validation is the true test of code, not how the code is written.

However, if other factors or quality attributes are specified or otherwise required such as, for example, modifiability and maintainability, program structure becomes important.

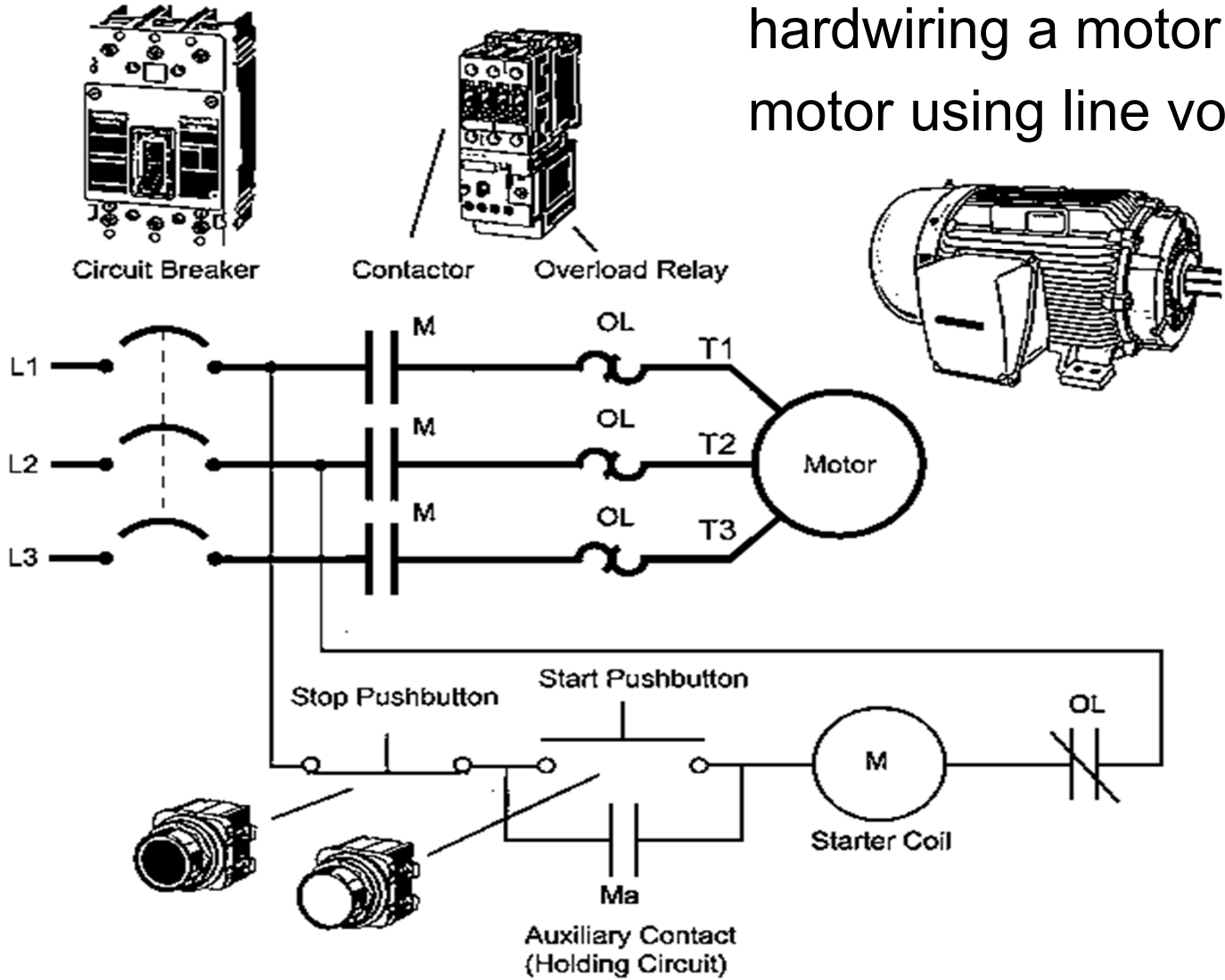
PLC Programming Techniques 2

Lesson Objectives

- Review basic Motor Control Program
- Understand the how that both Hardware and Software safety systems are sometimes required
- Look at how some programming techniques can be used in the Simulated Environment
- Discuss how limitations in the Simulated Factory are reflections of actual systems in Real Life.

PLC Programming Techniques 2

Typical arrangement for hardwiring a motor starter & motor using line voltage

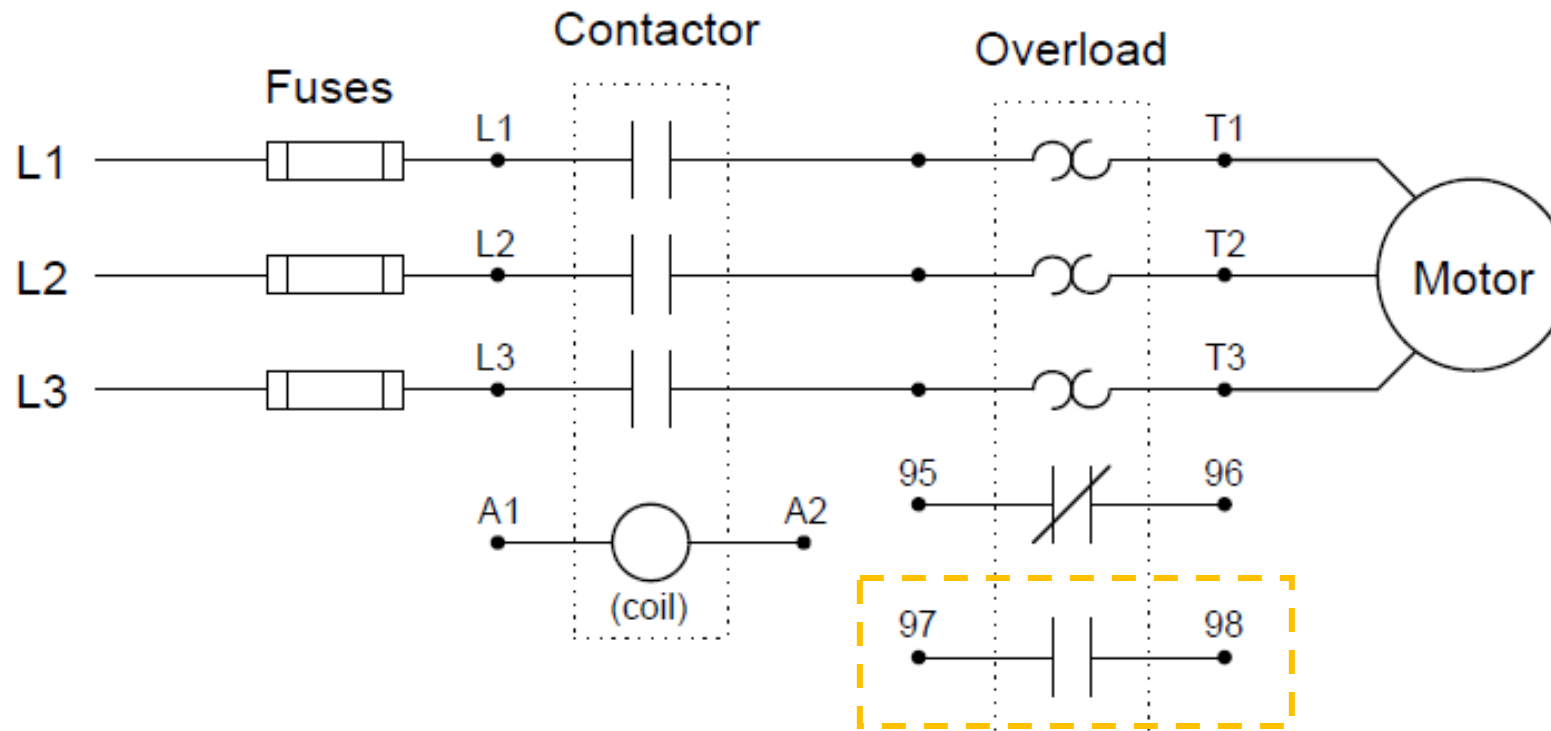


PLC Programming Techniques 2



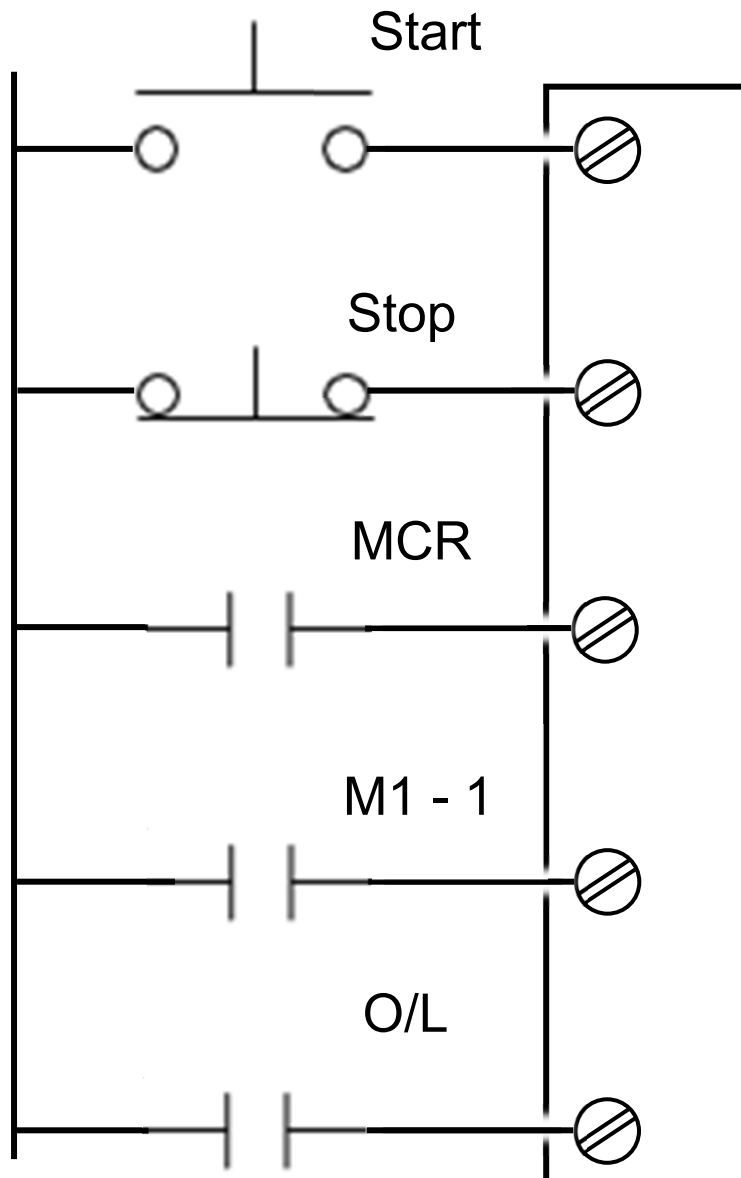
- Standard Starter consisting of a contactor coupled with an electronic overload block.
- The contactor and overload both have one NO and one NC contact for monitoring purposes

PLC Programming Techniques 2



Contactors with overload blocks (previous slide) may have an extra set of contacts (term 97, 98) that can be utilized as a monitor input to the PLC.

PLC Programming Techniques 2

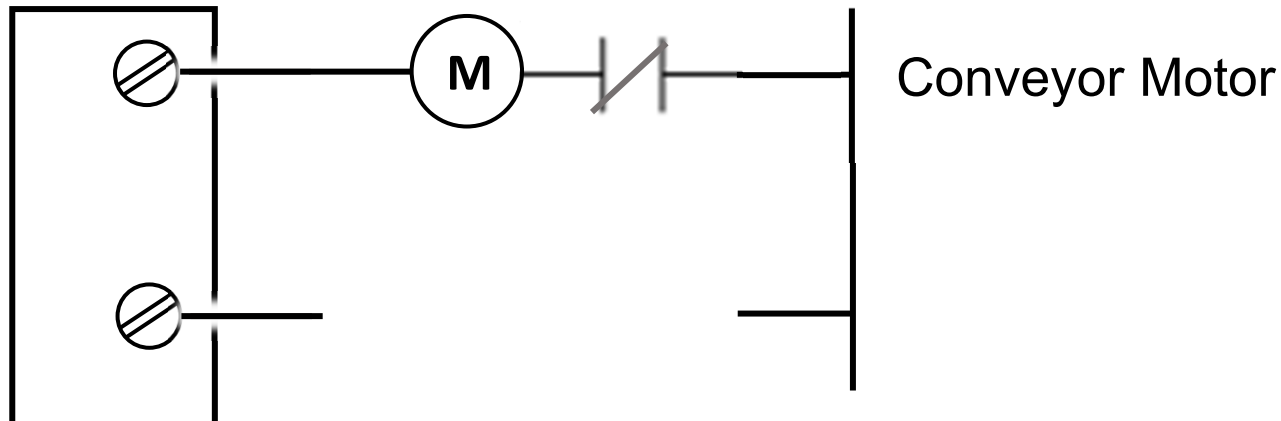


- Begin by defining some standard PLC inputs.
- Now the Stop Button is N.C.)
- All other inputs are N.O. for simplicity (Notes: Could be Physical PB or even inputs from HMI)

For this example, addressing is not necessary and will not be discussed

N.O. = Normally Open
N.C. = Normally Closed

PLC Programming Techniques 2



Shown here, the motor starter is wired to a PLC Output with a series connection to the Overload Monitor Contact. There are some examples deviating from this traditional wiring, however, the hardwired O/L contact provides an extra layer of protection.

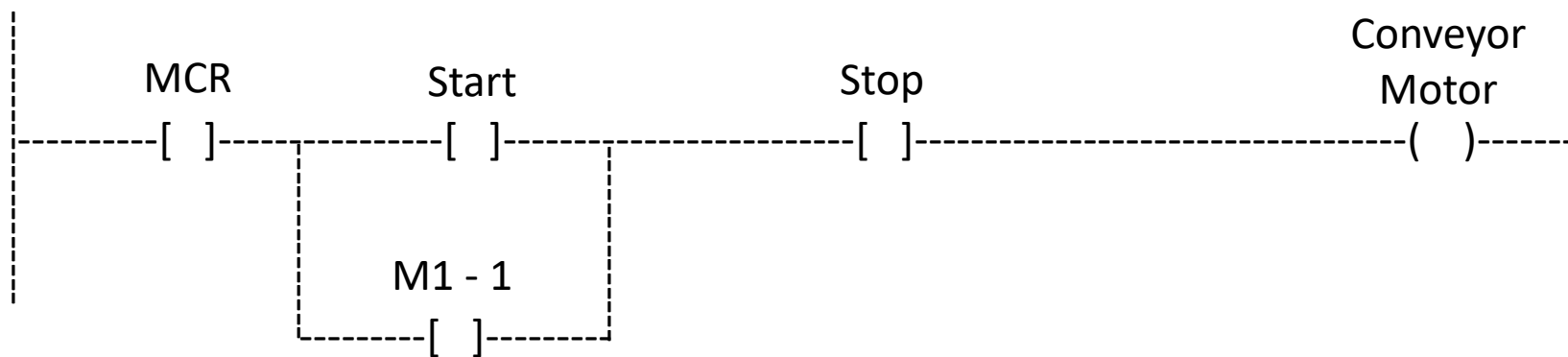
PLC Programming Techniques 2

Complication Factors not Addressed:

Programming Techniques is a very difficult subject to discuss since how the program is written will depend heavily on many other mitigating factors such as:

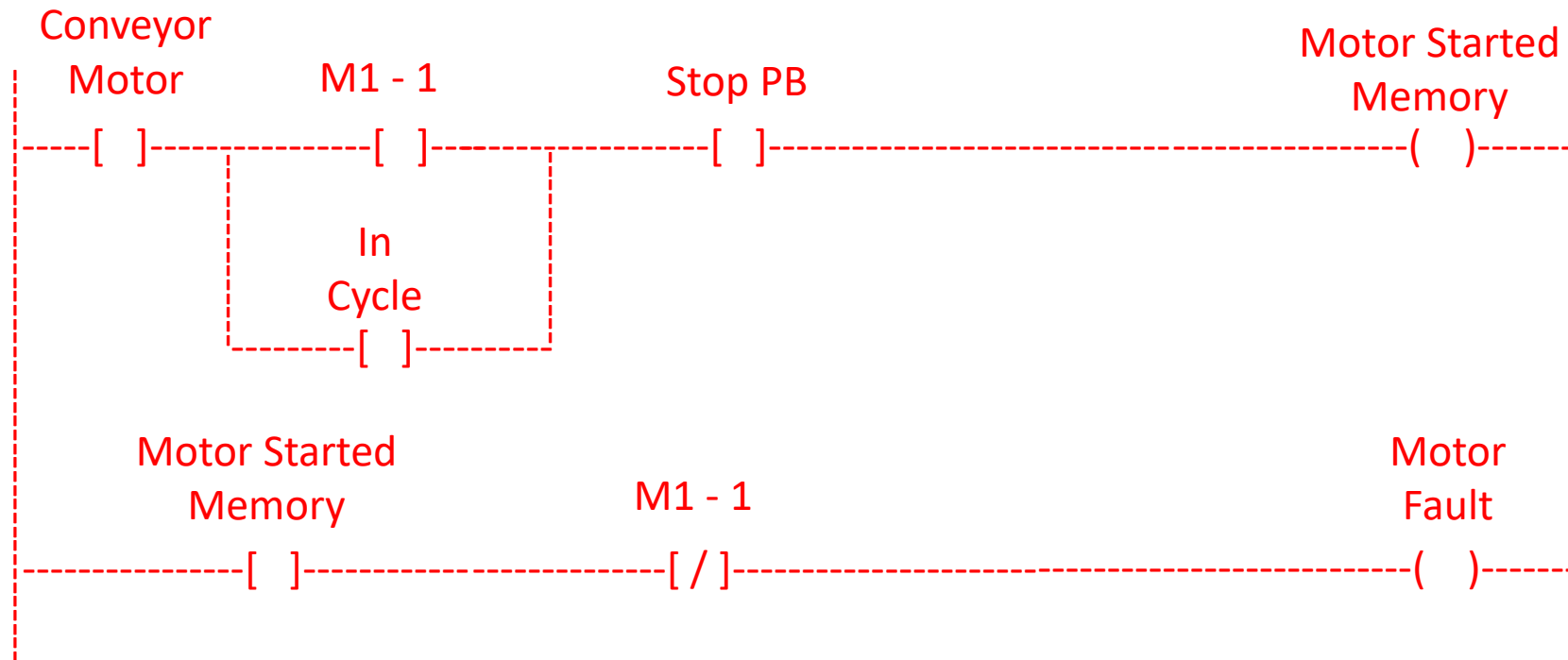
- Network Connected “Smart” Contactors and Starters
- Integrated Safety Systems
- Contactors monitored by a second O/L contact
- Motor systems utilizing current sensing to detect errors before they lead to a catastrophic failure.

PLC Programming Techniques 2



Seems rather simplistic but the idea here is that the Auxiliary Contract from the Starter is used as the Seal, NOT the output to the Starter. In this way, if the Starter fails to engage or is de-energized due to can Overload, the Software Seal is disrupted providing a second level of safety to accidental restart in case of Auto or Manual Reset prior to the Stop Button is Pressed.

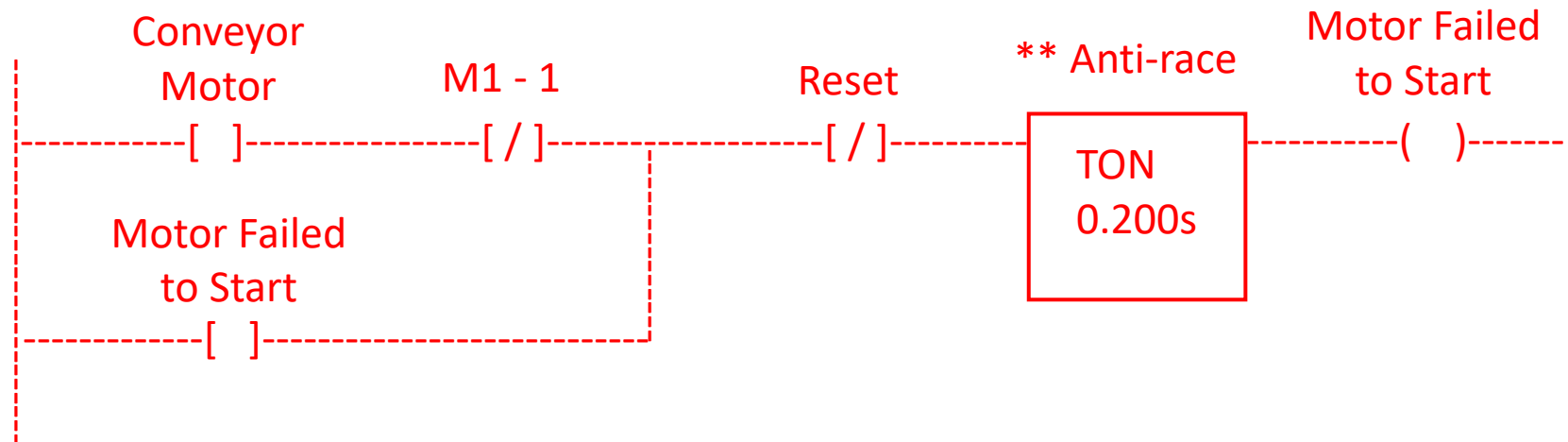
PLC Programming Techniques 2



In the case of NOT having a second contact on the Overload block, a simple memory can be used to detect a motor has tripped.

Fault detecting and messaging need not be complicated, but it can be a time-consuming endeavor during program development.

PLC Programming Techniques 2



The previous network dealt with the situation in which the motor started and then suffered a fault due to overload. However, if the motor contactor fails to engage, a direct method of detecting the failure is shown here.

An indirect or secondary method such as zero speed, pressure or vibration sensing should also be used in case of mechanic failure

** Anti-race -- see next page

PLC Programming Techniques 2



We must always take into account the electro-mechanical delay inherent in the system when creating fault detection programming. There will always be a slight delay between energizing the contactor or solenoid and seeing a monitoring input turn on so an “anti-race” timer needs to be employed to prevent nuisance faults.

As obvious as this seems, the response delay duration is so short that the fault bit will seal in without the programmer ever seeing the event on a programming terminal.

PLC Programming Techniques 2

Note that:

Using sealing or latching programming is up to the discretion of the programmer or may even be part of system specifications, PLC architecture or other design need.

PLC Programming Techniques 2

Applying to Simulation Programming

The same concept employed in Fault Detection can be applied to monitoring or creating Advanced and Returned Status in the Simulated Environment. These technique can also be carried over to real world machines and other automatic equipment.

PLC Programming Techniques 2

Most of the simulations do not include monitoring off all positions which means we need to be creative in determining how to best mimic this type of detection.

Factoryio has a least 3 different Command (C) / Status (S) combinations:

- Move (C), Moving (S)
- Move (C), In Position (S)
- Move (C), Actual Position (ie: Clamped) (S)

The next few slides provide some ideas on how to program using limited information.

PLC Programming Techniques 2

Move (C), Moving (S) [Initial condition 0, 0]

When the Move is set to 1, Moving Status = 1 is returned until the object in the Simulation Stops moving. Once the object stops, Moving Status is set back = 0. The same thing occurs when the Move Command is set back to 0 and the Object moves back to its original position.

Move (C), In_Position (S) [Initial condition 0, 1]

When the Move is set to 1, In_Position Status = 0 is returned until the object in the Simulation Stops moving. Once the object stops, In Position is set back = 1. The same thing occurs when the Move Command is set back to 0 and the Object moves back to its original position.

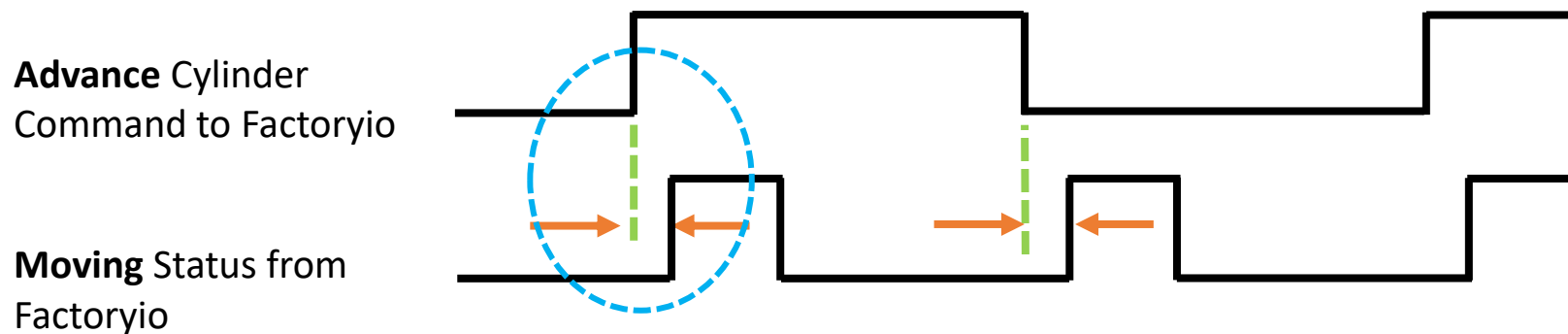
Move (C), Actual Position (ie: Clamped) (S)

When Move is set to 1, there is no change in Position Status until a Part is Captured by the Clamping Mechanism. Once a part is in position, Clamped is set = 1. Clamp position can vary depending on part size.

When Move is set to 0, Clamped Status changes back to = 0 as the Clamp starts to open. There is no change in Status after this point in the operation.

PLC Programming Techniques 2

Without direct positional information, we need to be creative in determining how to best mimic this type of detection. At the same time, the delay between Issuing a Command and detecting a change in Status via an Input needs to be addressed.



As discussed with the Contactor example earlier, the response delay duration is very short. Status memories, such as:

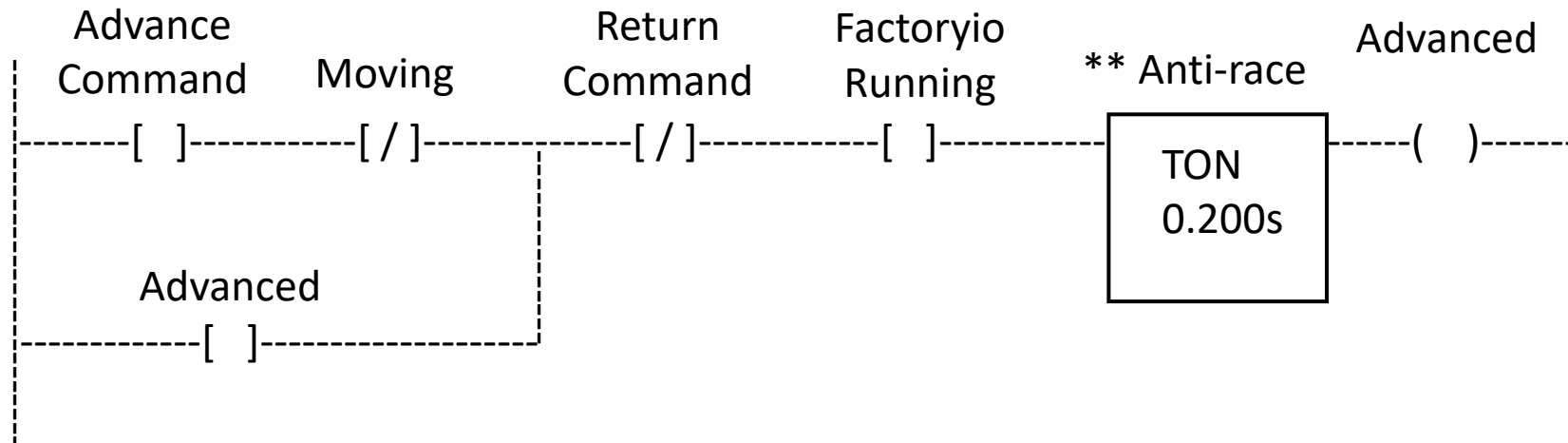
Movement Complete = Advance Command AND NOT Moving

will seal in without the programmer ever seeing the event on a programming terminal.

PLC Programming Techniques 2

Move (C), Moving (S) [Initial condition 0, 0]

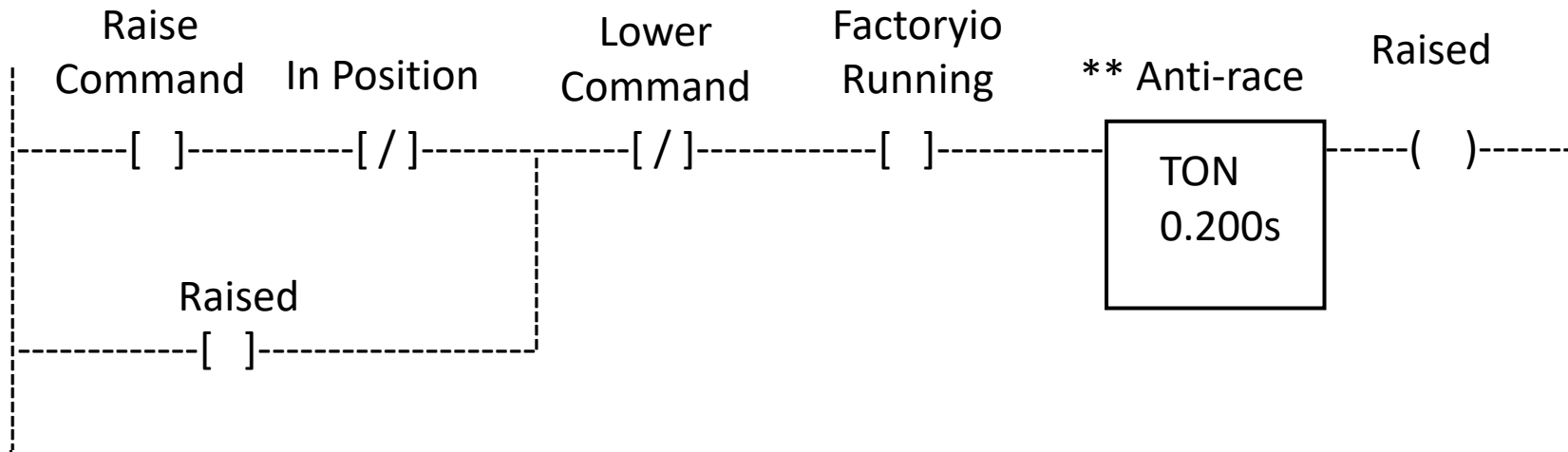
Since many of the simulations do not include full monitoring off all positions which means we need to be creative in determining how to best mimic this type of detection.



A simple timer can be utilized to ignore any set of conditions that is short in duration. For this example, 200ms is used.

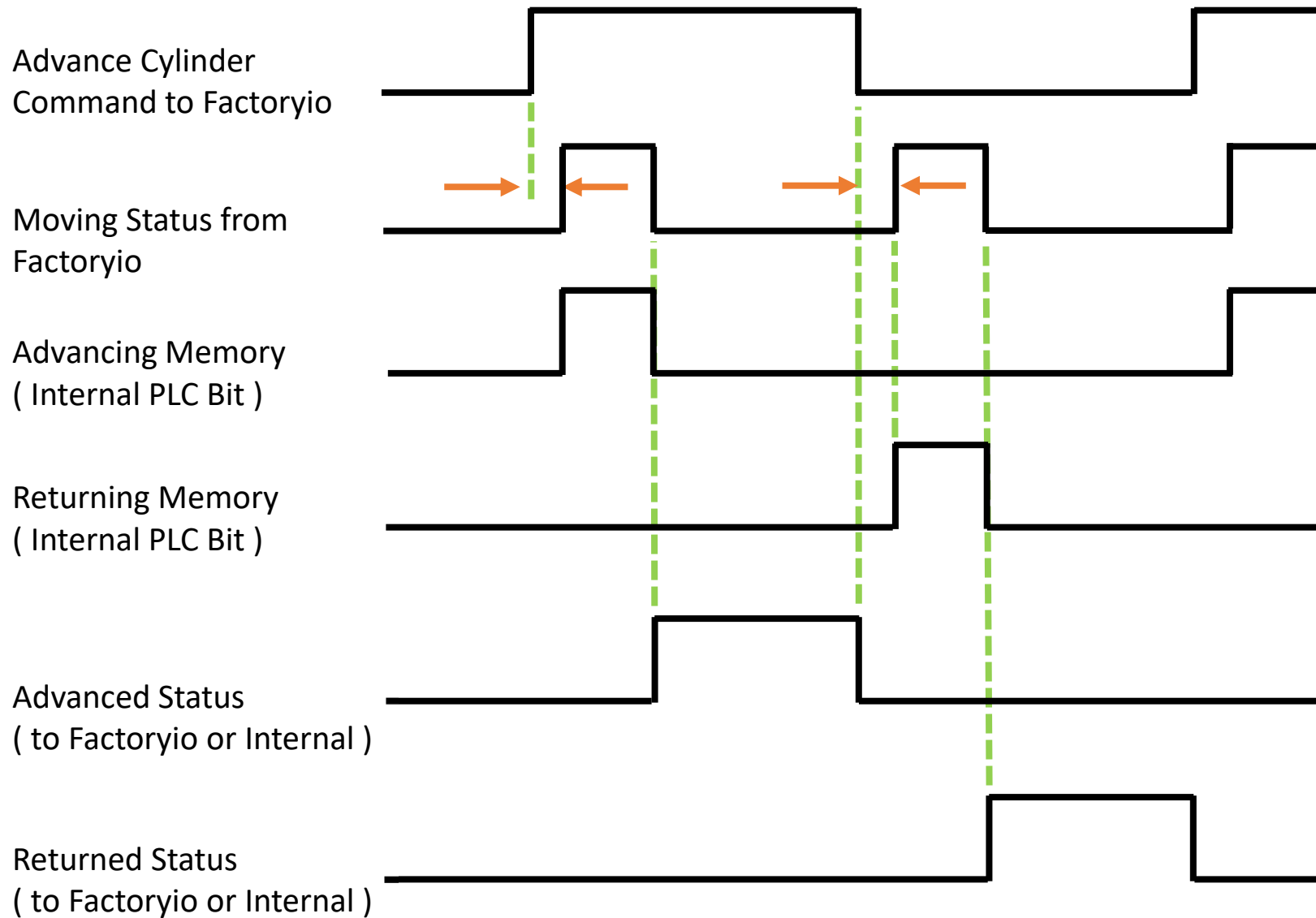
PLC Programming Techniques 2

Move (C), In_Position (S) [Initial condition 0, 1]



In both these examples, it would certainly be better if an Advanced or Raised Proximity is used, however, this relates to real Machines in that there are some situation where direct detection is not possible and a secondary method must be used. Timers can work but rely on the assumption that a motion has completed.

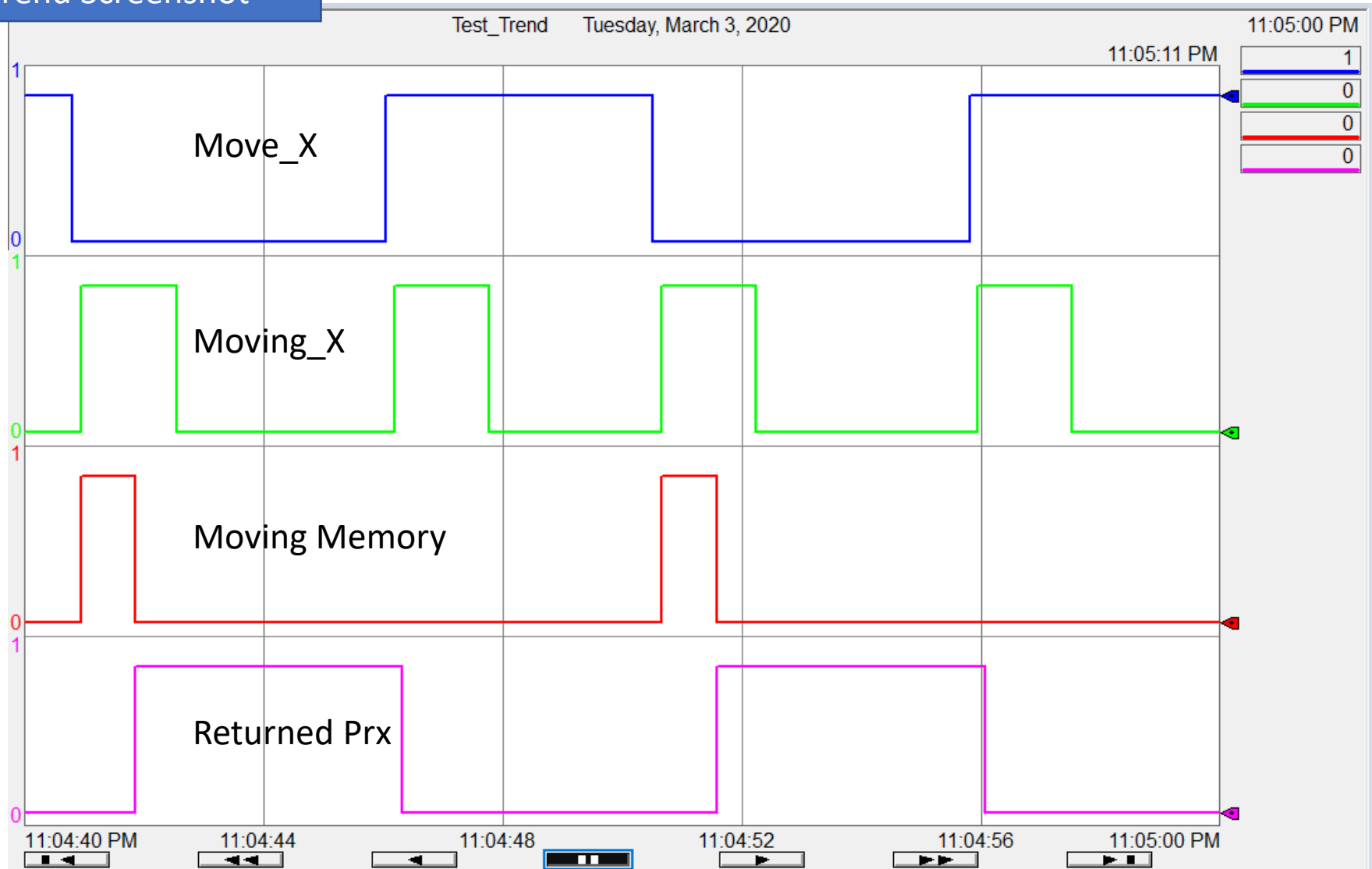
PLC Programming Techniques 2



PLC Programming Techniques 2

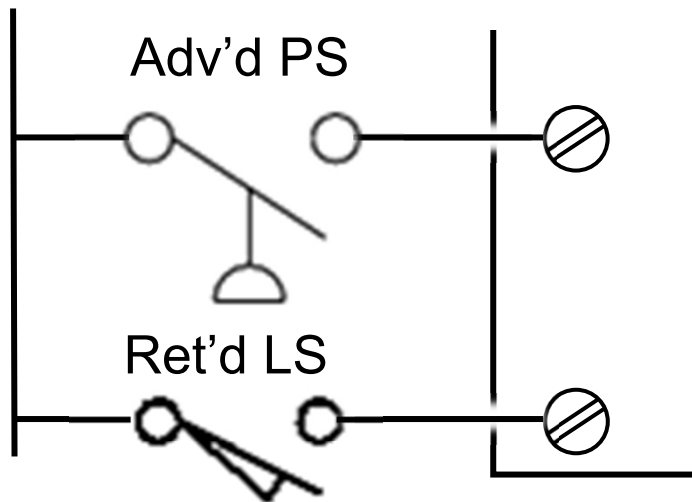
You can employ the Rockwell Trending tool to verify the predicted timing.

Trend Screenshot

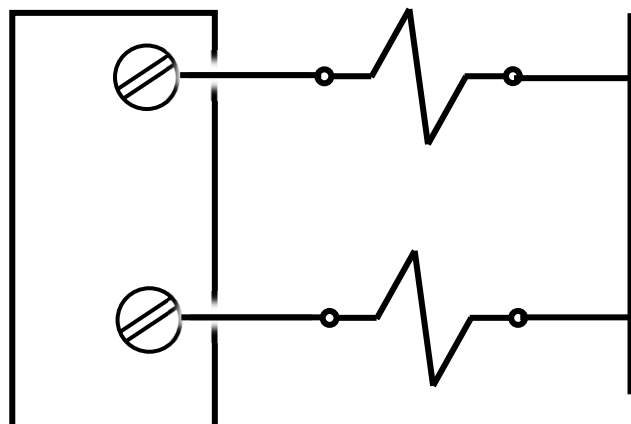


PLC Programming Techniques 2

Variation for Discussion.

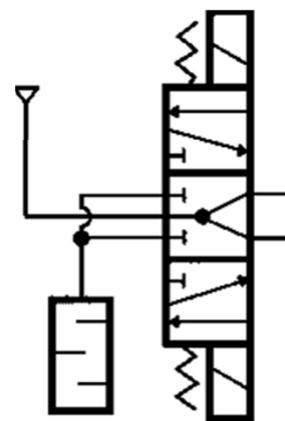


- Two outputs move cylinder to an Advanced Position and then Return it (Home) again.
- Only one direct position detection and one pressure switch to detect build up of pressure in the Advance Air Line.



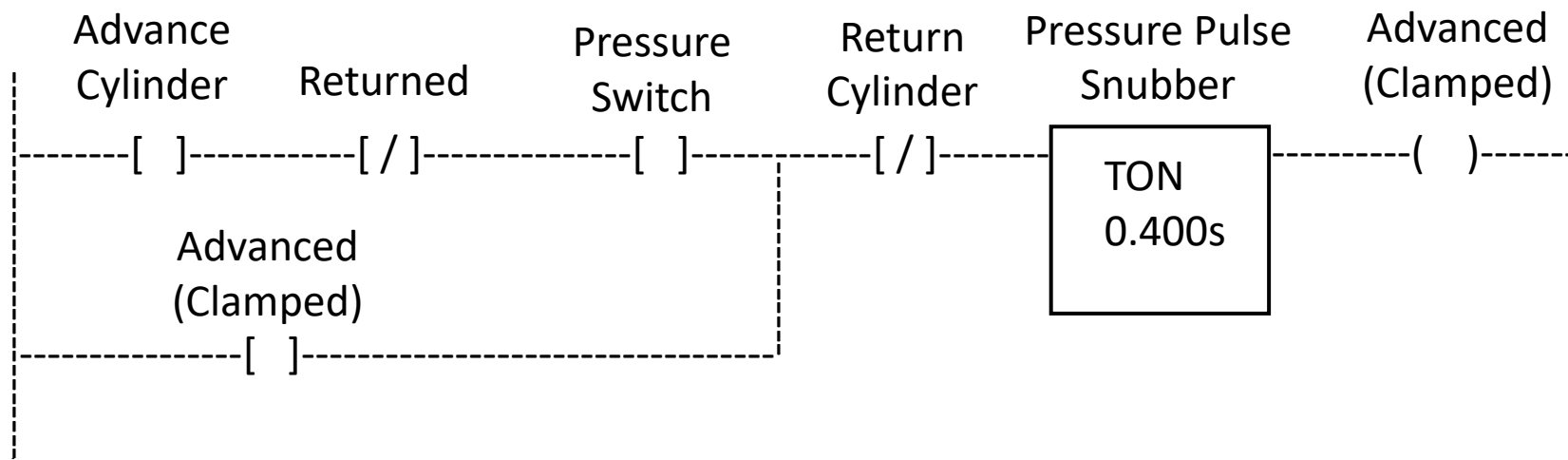
Advance

Return



PLC Programming Techniques 2

One reason for this configuration could involve a clamping mechanism with no fixed advanced position due to part variances.



There is always a chance, depending on the pressure setting and sensitivity of the Pressure Switch, that pressure build up at beginning of a stroke or during engagement could trigger the switch. The timer is added to ensure the pressure is steady before signaling an Advanced State has been reached.

PLC Programming Techniques 2

These slides did not address all possibilities but should give you some better insight into programming and how to apply various techniques to the Simulated equipment.