**SOLANO COMMUNITY COLLEGE**

**MT 164, PROGRAMMABLE LOGIC CONTROLLERS**

Spring, 2018

PLC PROGRAMMING LAB 9 - FESTO AUTOMATION TRAINER: PT 1

**Catharine Crayne**

**04/25/2018**

**OBJECTIVE**:

1. To write a ladder logic program the will incorporate the use of subroutines and timers.

**Group Members:**

**-**

**-**

**-**

**-**

**-**

**-**

**MATERIALS**: Micrologix 1100 Trainer Unit, Festo Automation Trainer

**SAFETY AND EQUIPMENT NOTES:**

1. Turn off trainer when attaching wires

**PROCEDURE**:

**Part 1: Research**

1. Investigate one section of the Festo Automation Trainer. Trainer Station: Distribution
2. Write down the steps and sequence of its operation. Use the animation found on the course’s Google Drive.
3. Apply 24V power to the Input Sensor wiring block and identify and locate each sensor.
4. List out the Input Sensors and describe what they do.

**1B1** – Limit Switch – Senses when the cylinder is extended

**1B2** – Limit Switch – Senses when the cylinder is retracted

**2B1** – Vacuum Sensor – Detects when air flow is interrupted in the venturi (when puck is picked up)

**3B1** – Limit Switch – Detects when the arm is extended fully on the outside of the terminal

**3B2** – Limit Switch – Detects when the arm is extended fully to the platform holding the puck

**B4** – Break Beam – Detects when either a puck is in the bottom of the cannister or the cylinder is retracted to push a puck into position to be picked up.

**IPF1** – Optical sensor – Detects a connection to a corresponding Festo terminal

1. List out the Outputs and describe what they do.

**1M1** – Pushes cylinder into the retracted position

**2M1** – Venturi uses air pressure to pick up the puck

**2M2** – Venturi reverses air flow/pressure direction to release/blow puck into the next terminal

**3M1** – Arm moves from the outside of the terminal to the interior to pick up the puck

**3M2** – Arm moves to drop the puck into the next terminal

**Part 2: Testing outputs**

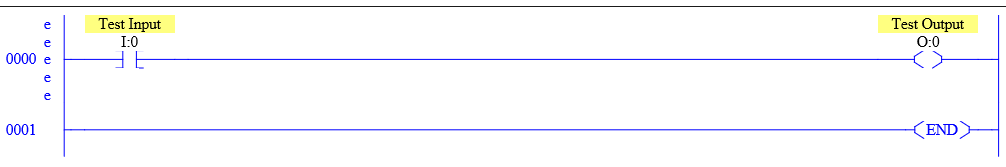
1. Write a ladder logic program that will energize an output when a push button is pressed.
2. Download your program to your Micrologix PLC trainer and wire up the push button and the output.
3. Connect Ground to your output block on the FAT. Use the output from your PLC to power various output components on the FAT to see how they will work. You may need to provide air to your FAT.

**Part 3: Writing the control program**

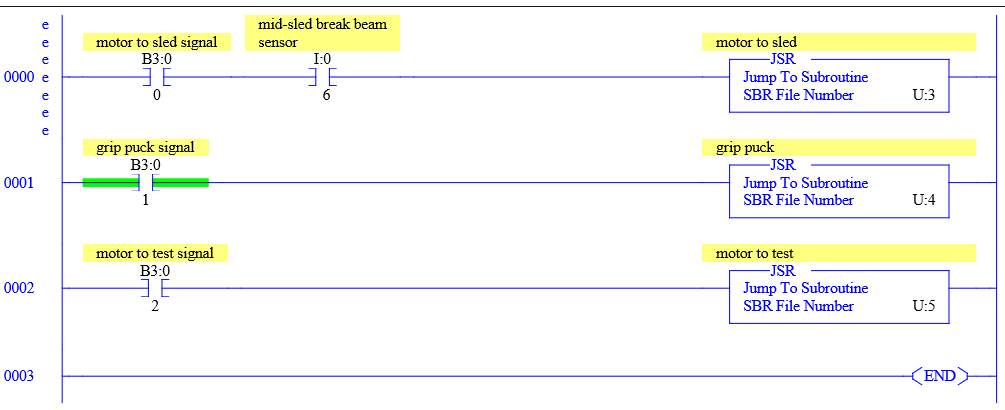
1. Now begin to write the control program for the FAT. It should use a push button to start it, and a safety stop to stop it. Use these buttons on your trainer. Think about how to use subroutines to make your program easier to read and troubleshoot. Program it in stages. Keep in mind if the start conditions or reset conditions that may be needed for it to work. If your FAT uses more than 10 Inputs, devise a way to split the program between two PLCs.
2. Verify that it works. Print out your program(s).

**RESULTS - DATA**

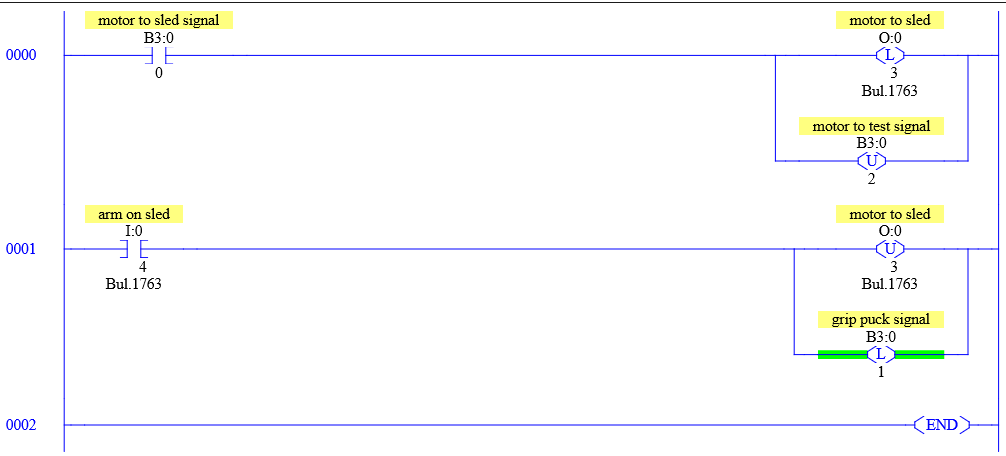
Part 2: Snapshot of our program.



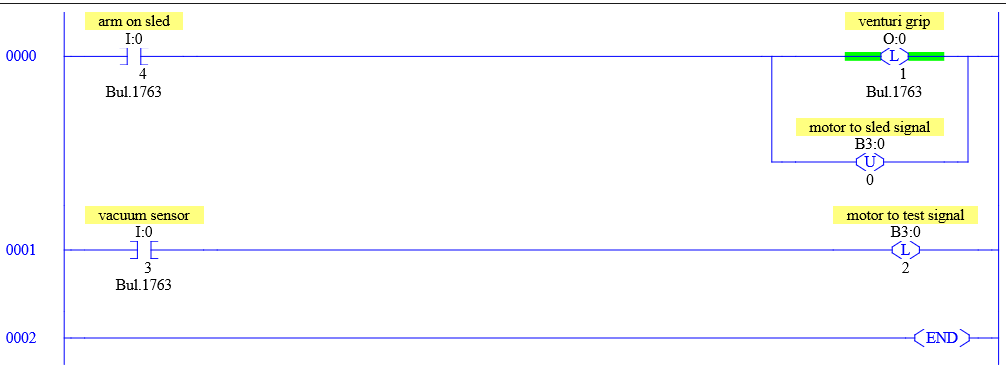
Part 3: Snapshot of our program.



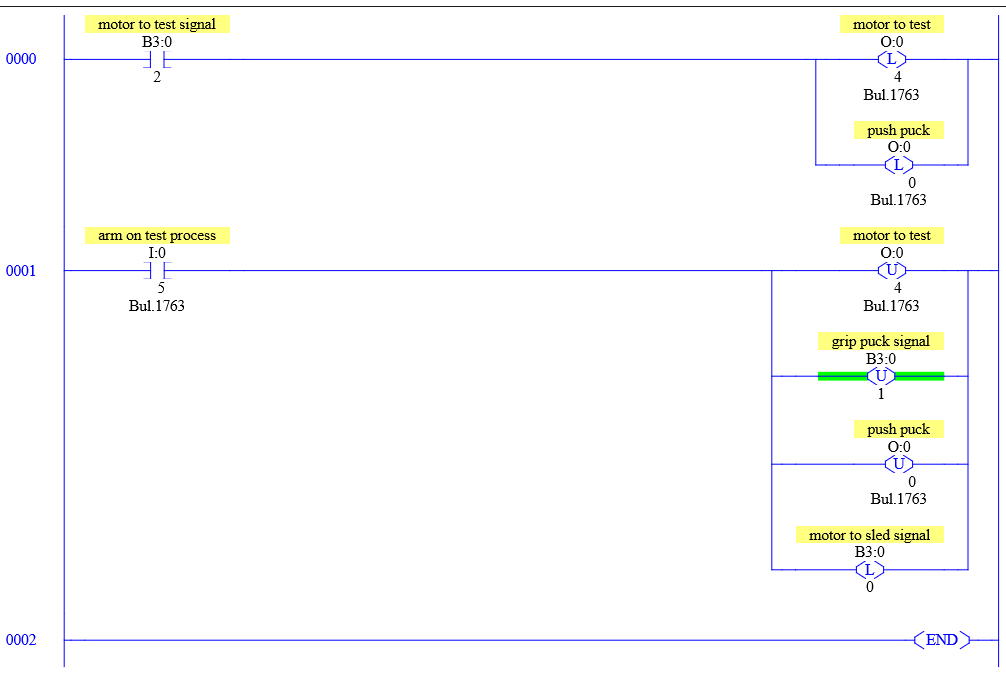
Motor to Sled Subroutine (U:3)



“Grip Puck” Subroutine (U:4)

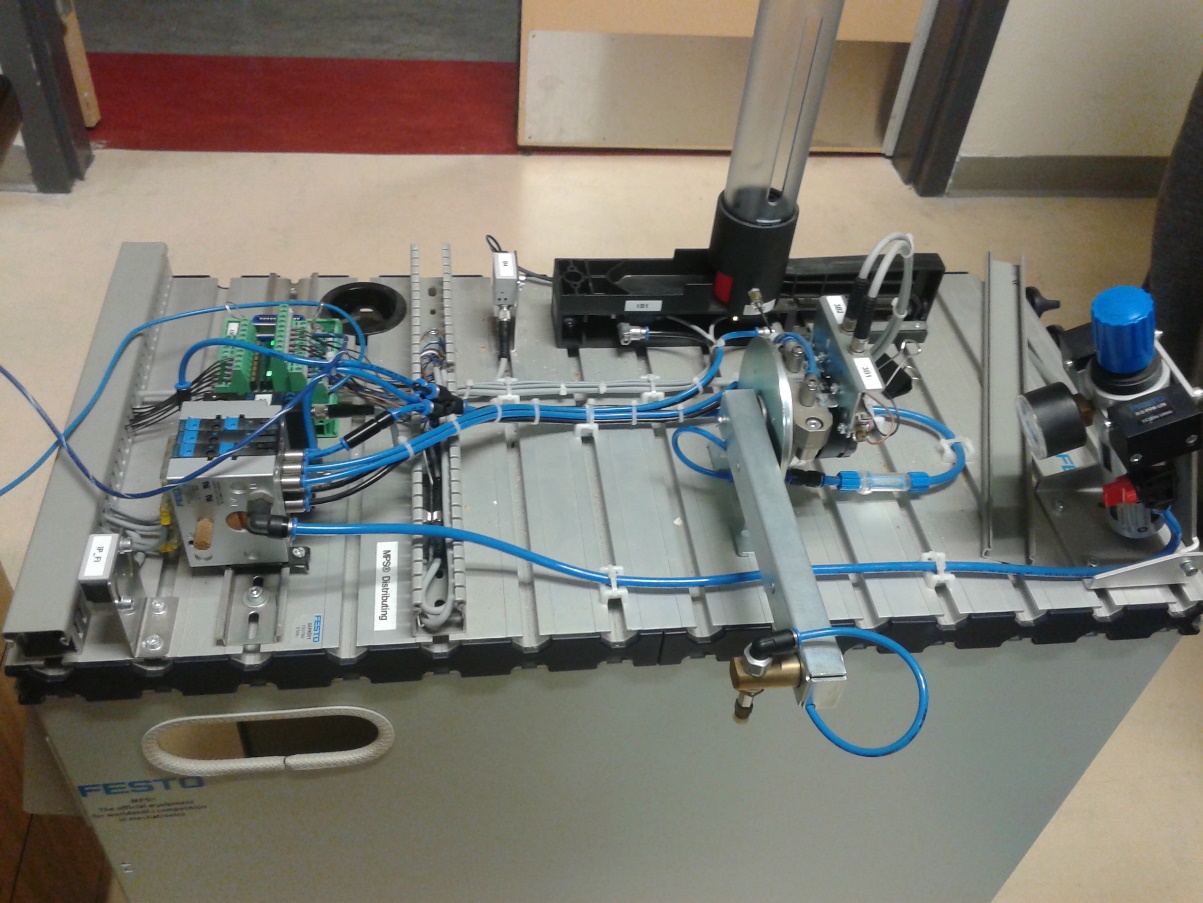


Motor to test process (U:5)



**OBSERVATIONS**

Part 1: Picture of our setup



**ANALYSIS QUESTIONS:**

1. Why is the Control Logix 1500 PLC more suited for this Automation Trainer?

The ControlLogix 1500 PLC is more suited for the Automation Trainer because it has more input/output slots.

1. What was the most difficult part to program and what can you do differently to mitigate the problems or issues with this part?

Connecting the sub-routine functions using flashing bits.

1. How does this station communicate with the previous or next station in the automation line?

They communicate via optical sensors. This station uses IPF1

1. What could you do or change about your program if initial conditions aren’t met, like an actuator in the wrong position or a puck is jammed? (don’t program it, just describe how it could be changed).

Program an emergency stop that de-energized all outputs if all required input conditions are not met.