**SOLANO COMMUNITY COLLEGE**

**MT 164, PROGRAMMABLE LOGIC CONTROLLERS**

Spring, 2018

PLC PROGRAMMING LAB 10 - FESTO AUTOMATION TRAINER: PT 2

**Catharine Crayne**

**05/17/2018**

**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
2. 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.

**PROCEDURE​:**

**Part 1: Research**

1. Investigate one section of the Festo Automation Trainer. Trainer Station: \_Testing \_\_\_\_

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.

I:0/0 – Part\_AV – Blue capacitive sensor – Detects if a puck is on the platform

I:0/1 – B2 – Inductive sensor – Detects if a metal puck is on the platform

I:0/2 – B4 – Reflective Break Beam: Determines if there is an obstruction in the path of the lift

I:0/3 – R1 – Compariter: Measures puck size once lifted

I:0/4 – 1B2 – Limit Switch: Upper limit switch on lift

I:0/5 – 1B1 - Limit Switch: Lower limit switch on lift

I:0/6 – 2B1 – Sensor: Checks if cylinder is empty or has a puck in it

I:0/7 – Push Button: Stops process

IP\_N\_FO - Optical sensor – Detects a connection to a corresponding Festo terminal

B5 – 0-10V Analog Compariter – used to adjust height value accepted as true by R1

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

O:0 – Lower Lift

O:1 – Raise lift

O:2 – 2M1 – Pushes puck off the platform, onto the slide

O:3 - Blows air at puck to push it all the way down the slide

**Part 2: Testing outputs**

6. Write a ladder logic program that will energize an output when a push button is pressed.

7. Download your program to your Micrologix PLC trainer and wire up the push button and the output.

8. 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.

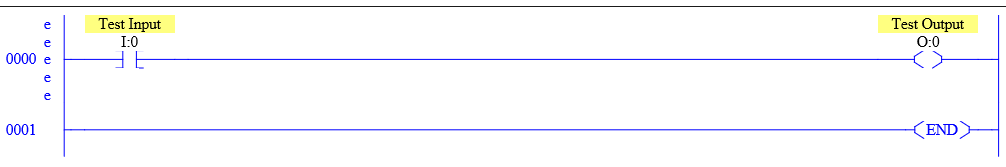
**Part 3: Writing the control program**

9. 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.

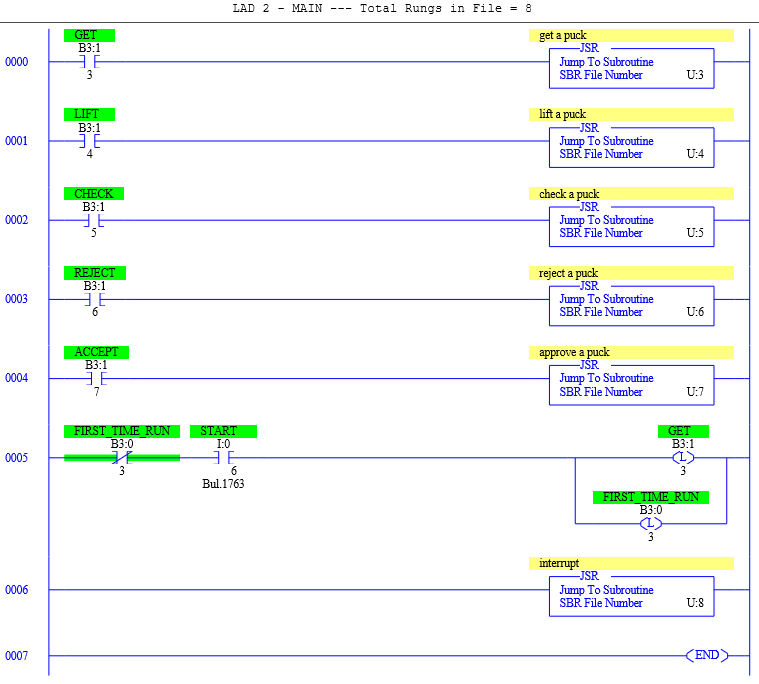
10.Verify that it works. Print out your program(s).

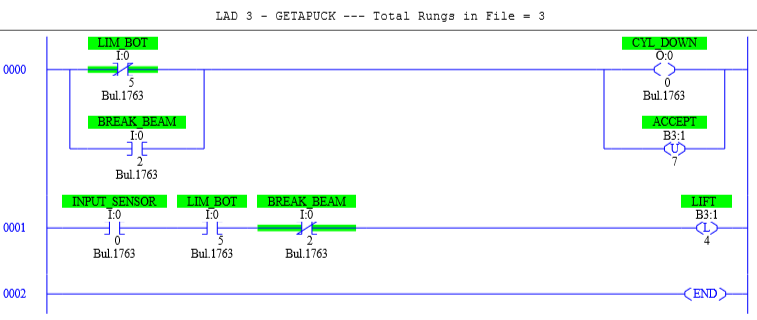
**RESULTS - DATA**

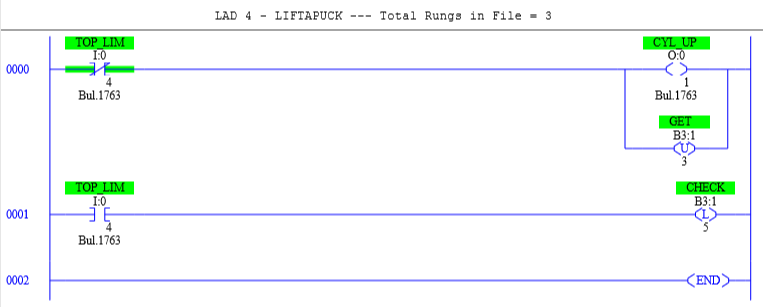
Part 2: Snapshot of our program.

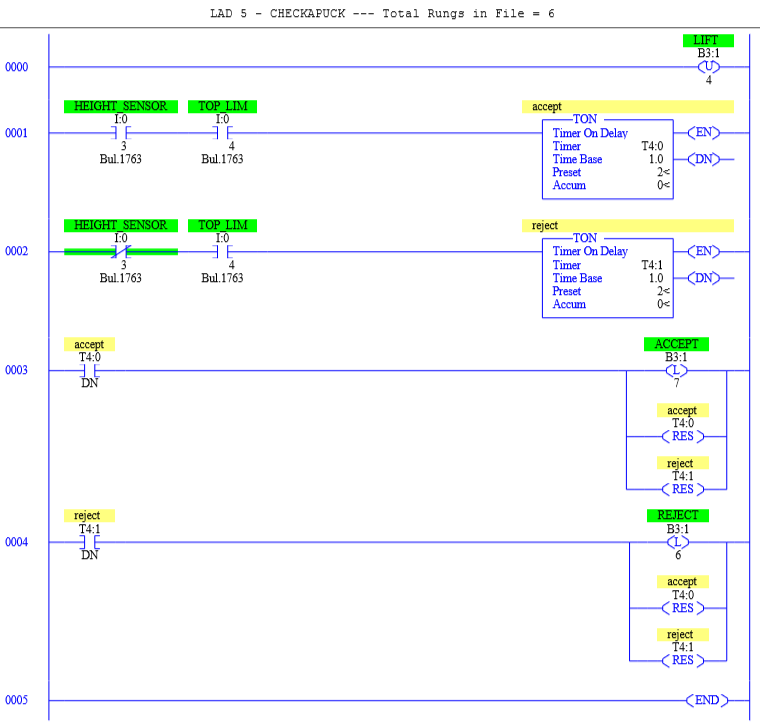


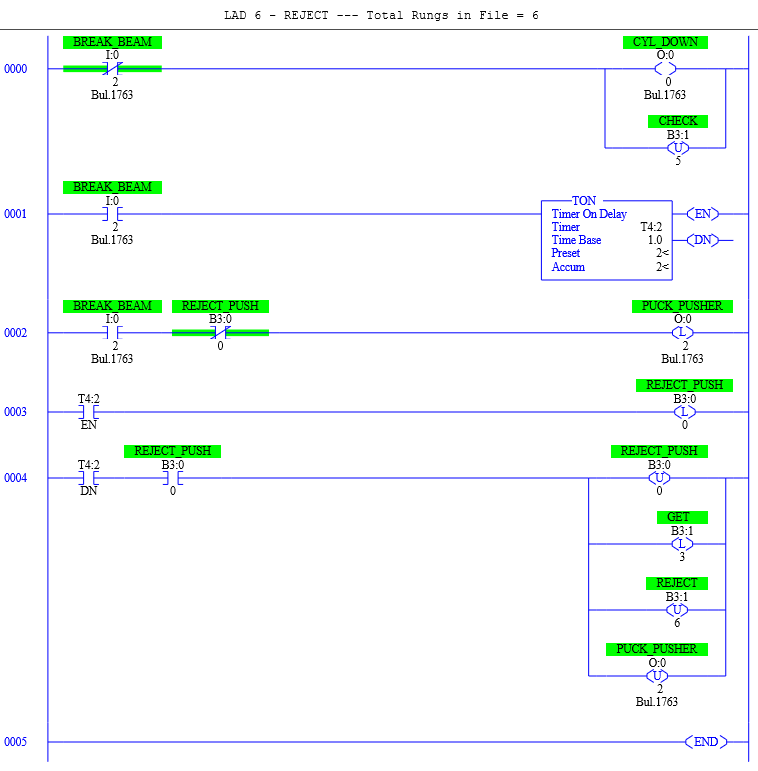
Part 3: Snapshot of our program

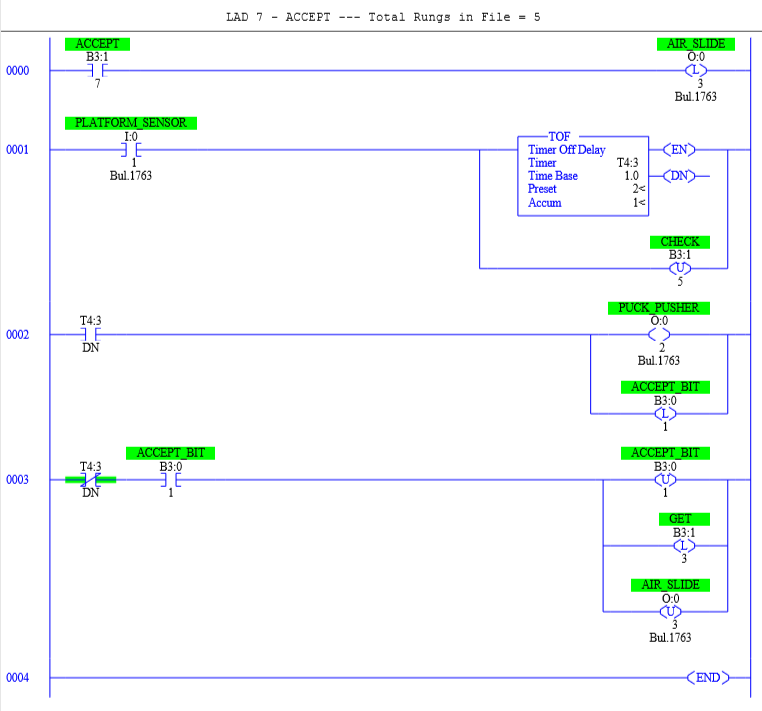


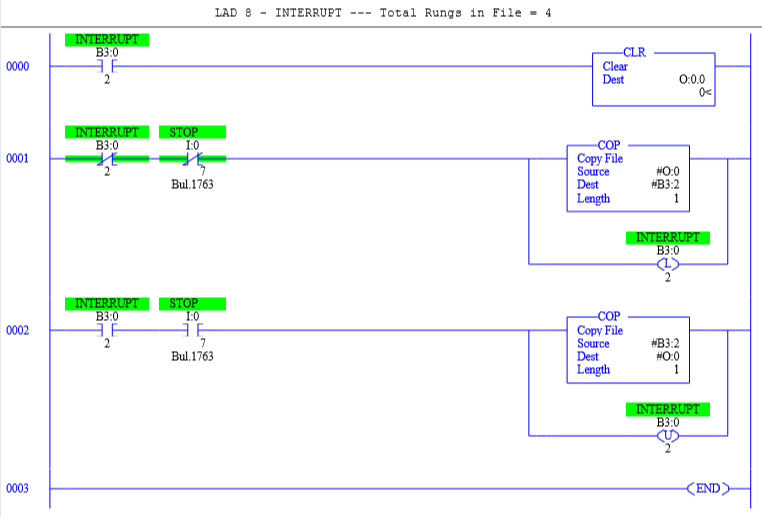






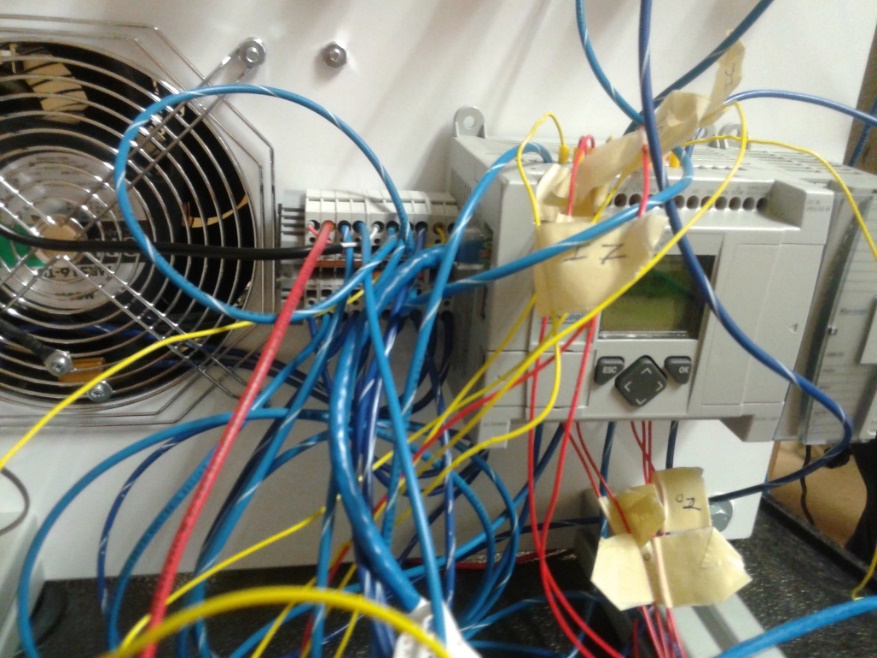
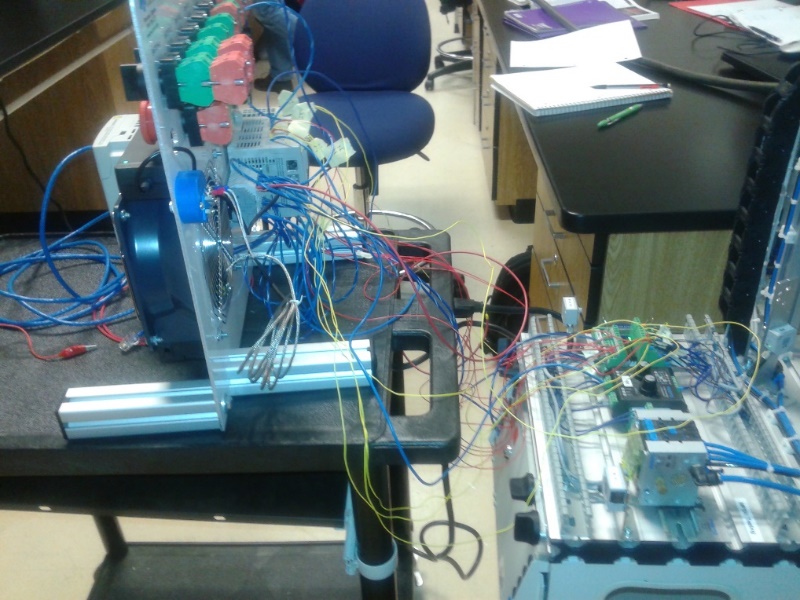


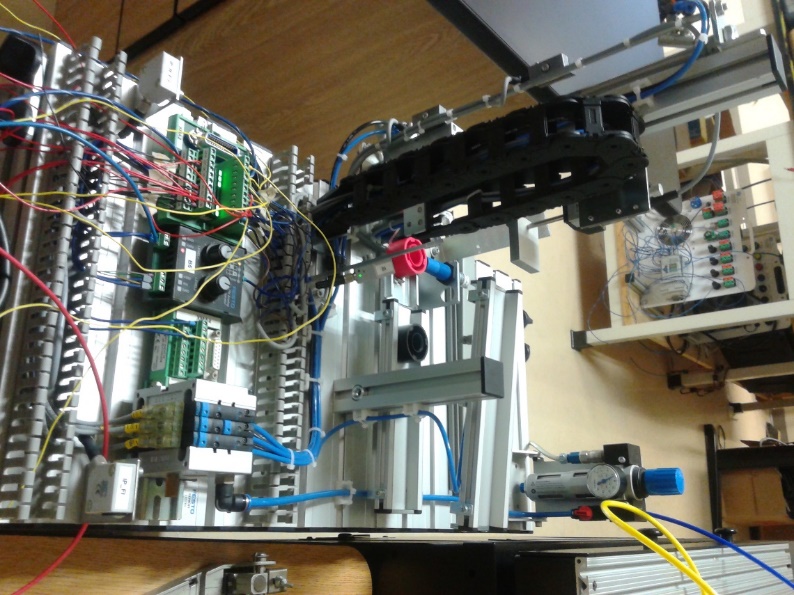




**OBSERVATIONS**

Part 1: Picture of our setup



**ANALYSIS QUESTIONS:**

1. What similarities does this Trainer station have in common with the previous one you worked on?

*They are both Festo units. They both use limit switches to determine position of the parts in relation to where the unit is in the programmed process. Both stations use air pressure outputs to move the pucks. Both stations use break beams to determine if a puck is in place to continue operating the program. Both stations use an optical sensor to communicate with other Festo trainer stations.*

1. What did you learn from doing the last lab that was useful for programming this new one?

*Using sub-routines and latching bits to keep code organized and reduce the length of the main program. Without breaking the main program into subroutines the main program would span several pages and would be cumbersome to manage and troubleshoot.*