**SOLANO COMMUNITY COLLEGE**

**MT 162, ROBOTICS**

Fall, 2018

SOLANO COMMUNITY COLLEGE

ROBOTIC SYSTEMS MT-162

**LAB 7: BRANCHING**

**Catharine Crayne**

11/7/18

**OBJECTIVE**:

1. To learn how to create motion program that use various methods and types of branching.

**GROUP MEMBERS**

*Dianne*

**MATERIALS**: RoboGuide and Laptop, Fanuc Robot, Box

**PROCEDURE**:

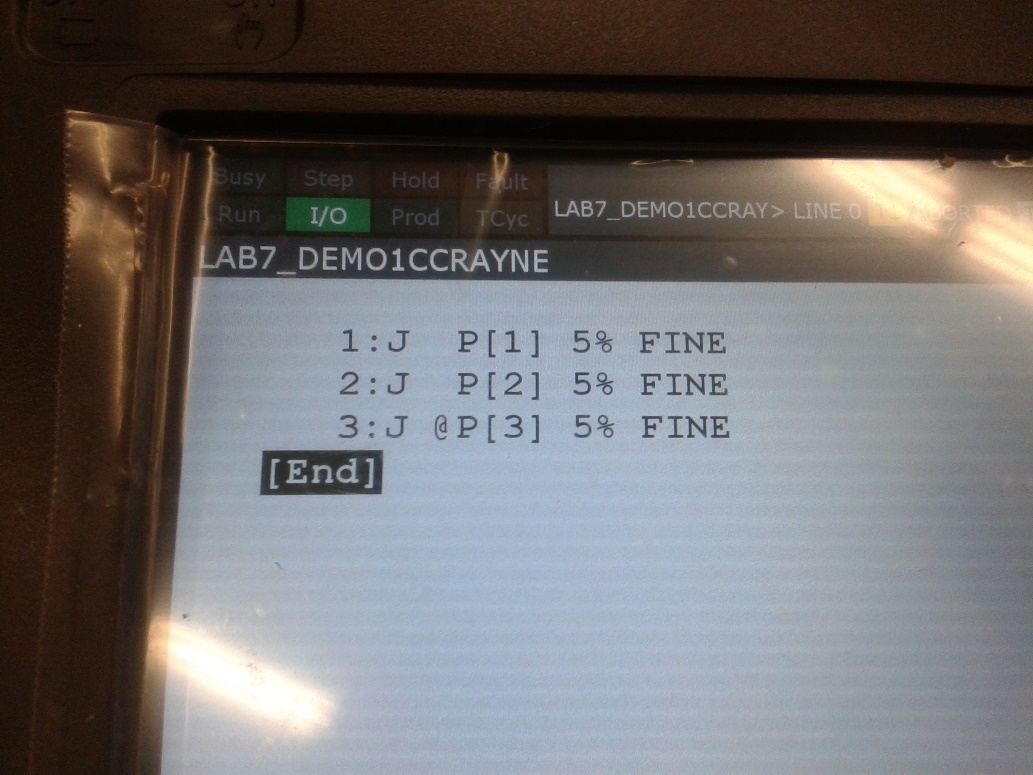
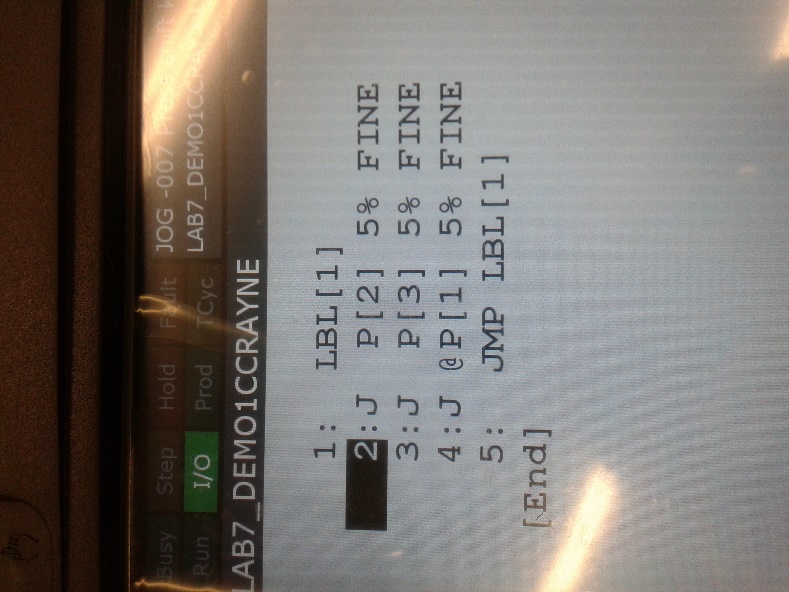
1. Power up the robot controller. Place the robot and pendant in Teach Mode.
2. Create a new program called “Lab7\_DEMO1YourName”.
3. Teach at least 3 points and motion between them.
4. Use the LBL and JMP LBL instructions to have the program repeat in an infinite loop.
5. Test it in STEP and RUN modes.
6. Create a new program called “LAB7\_TRIANGLEYourName”.
7. Teach the robot to start in a “home position”, trace out a triangle shape, and then return back to a “home position”.
8. Create a program called “LAB7\_CIRCLEYourName”.
9. Teach the robot to start in a “home position”, trace out a circle shape, and then return back to a “home position”.
10. Verify there is a program already written for a BOX shape.
11. Now create a new program called “LAB7\_MAIN1YourName”.
12. Have this program loop forever calling the other programs for BOX, TRIANGLE, and CIRCLE.
13. Create a new program called “LAB7\_MAIN2YourName”.
14. Have this program loop forever until 1, 2, or 3 is entered into data register 5. When 1, 2, or 3 is entered, the program will CALL the other programs BOX, TRIANGLE, or CIRCLE. (use an IF instruction)
15. Create a new program called “LAB7\_MAIN3YourName”.
16. Again have this program loop forever until 1, 2, or 3 is entered into data register 5. When 1, 2, or 3 is entered, the program will CALL the other programs BOX, TRIANGLE, or CIRCLE. (this time use the SELECT instruction)

**RESULTS - DATA**

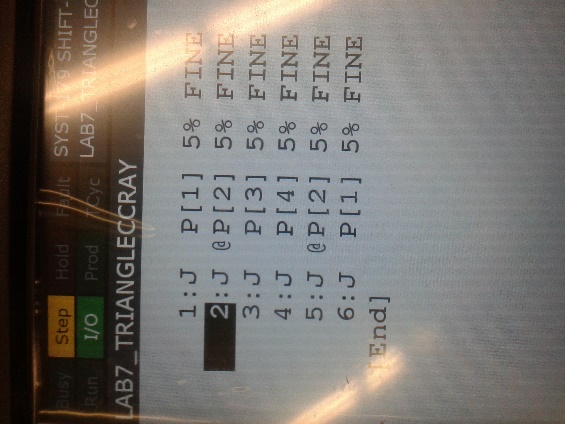
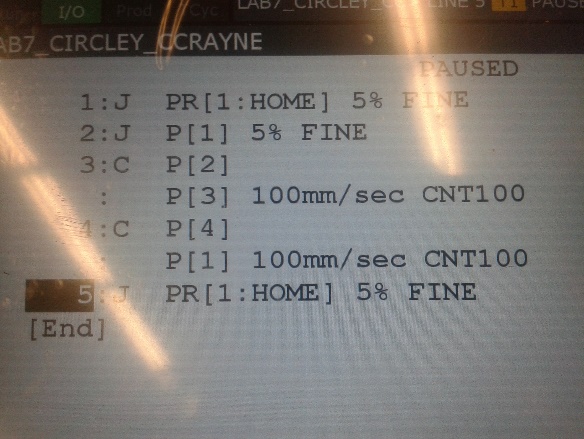
Not applicable for this lab.

**OBSERVATIONS**

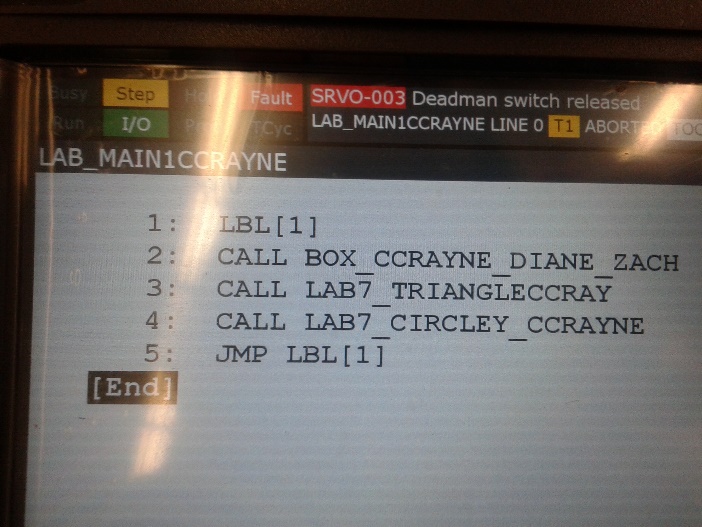
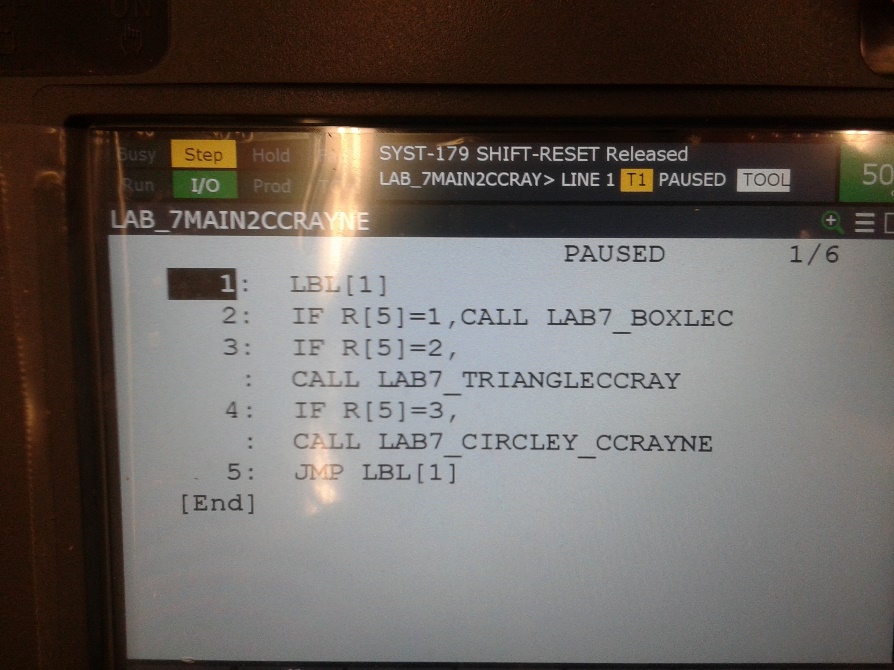
Step 2 - 3 Step 4-5

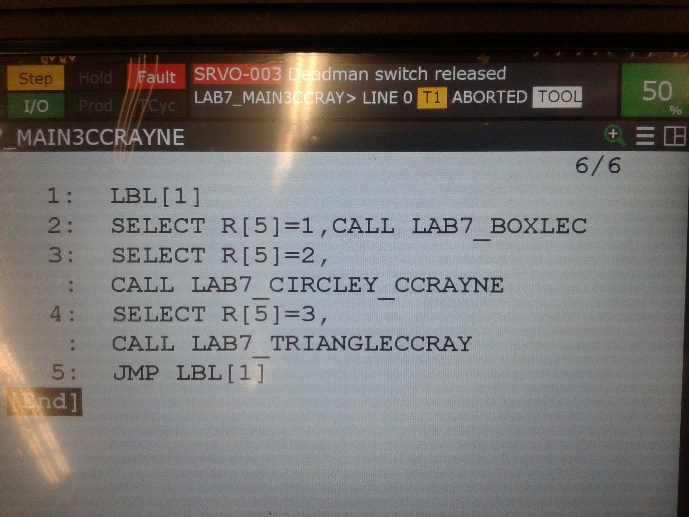
Step 6-7 Step 8-9

Step 10-12 Step 13-14

Step 15-16



**Analysis Questions:**

1. What would happen if the subprograms for BOX, TRIANGLE, or CIRCLE have an infinite loop within them and you called them from your MAIN program?

*The program would get stuck in the infinite loop in the sub-program and would not cycle through the remainder of the MAIN program.*

1. How would change one of these programs so that it would pause for half a second before repeating the loop?

*Add a ‘WAIT = 1.00 (sec)’ instruction in between the last SELECT/CALL instruction and the JMP LBL[1] instruction.*

1. How would adding the instruction “R[5] = R[5] + 1” to the top of your program in step 16 (right before the SELECT instruction), change the resulting program? (Assume R[5] = 0 in the beginning).

*It would increment R[5] by 1 each time the program runs. The program would start, R[5]=0 would increment to R[5]=1, the SELECT R[5]=1 condition would evaluate to true and the corresponding CALL instruction would execute the sub-program, the program returns to MAIN (in the event that the sub-program does not create an infinite loop), the program evaluates SELECT R[5]=2 and SELECT R[5]=3 to false so the corresponding sub-program does not execute, the MAIN program executes the JMP LBL[1] instruction, and R[5]=1 retains its value when the program starts over. R[5]=1 gets incremented again, the SELECT R[5]=1 condition would evaluate to false, the SELECT R[5]=2 condition would evaluate to true and the conditional instruction executes the sub-program and returns to MAIN, the SELECT R[5]=3 condition would evaluate to false, JMP LBL[1] is executed and the process repeats, this time executing the SELECT R[5]=3 condition to true. Once the program starts over again, the R[5] value increments to 4, and none of the SELECT conditional statements execute, though the program will continue to run.*