Robot Programming Lab #10 Registers JD Jones and John Nelson

Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Point Value = \_\_\_\_\_\_\_\_\_\_/100 points

In this lab you will be learning how registers work. Use your program for lab 9 and rename it for Lab 10.

Lets understand the Registers command.

Simple terms a REGISTER R[1] is a storage location or memory location in the robot capable of storing a number.

Registers are used for many types of controls on a robot. They can be set similar to a bit in a PLC, On or Off to tell a state of something. They can be used to store a larger number. For this example we are going to use a register to control how many times we run a loop. We will start with the program.

1: R[1] = 0

2: Lbl [1: start of program]

3: J P[1] 100% fine

4: J P[2] 100% fine

5: J P[3] 100% fine

6: J P[4] 100% fine

7: R[1] = R[1] + 1

8: If R[1]>= 5, Jmp Lbl [2]

9: Jmp Lbl [1]

10: Lbl [2]

END

This is what is happening.

Line 1: is resetting the register [1] to zero. (PLC folks = reset of a counter command)

Line 2: you should know this.

Lines 3-6: Moves for the robot to perform.

Line 7: is an incrementing counter. (PLC folks = CTU Up counter)

Line 8: is telling the robot to make a decision. **If the statement is False it goes straight to the next line which says to jump to Lbl [1]. If it is TRUE then it completes the rest of the line after the comma and jumps to Lbl [2]. I really want you to understand this!!!!!!!!**

|  |  |
| --- | --- |
|  |  |



This program will run for a total of 5 cycles.

**TASK:**

In this lab take your lab 9 program and save it as lab 10. Then change the program to run the pick and place portion then at the end have it clap for itself for 3 times. ! Hint: Don’t put in lines and lines of open and close gripper. You will loose points. **Make sure when you print it out and do a written explanation. It is worth 20 points.**

This is great time to show you how to have 2 screens open at one time.

Press the Shift and Disp key.

A menu will pop up choose double press enter

Now your screen will be split in the middle from top to bottom. You will notice the top left screen will be dark blue and the right screen will be white. Wherever the dark blue is located is what you have control of. To move the cursor to the right screen.

Press Disp key WITHOUT the shift key. This will toggle you between screens.

When you run this program it is great to see the registers increment while the program is running.

Put the cursor on the right screen.

Press the Data button

Choose registers if it is not on it already.

Move the cursor to the register you have used in your program.

Now move the cursor to the left screen and step your program from the FIRST line.

You will see the register zero out when you told it to. You will see it increment when you used the counter command line.

NOTE: The program you want to run MUST be in the left screen. It will only run in the left screen!

Be ready to demonstrate the following to the instructor.

**Points for:**

A) Describing the LBL INSIDE THE LBL instruction. Not just a remark! Use your name in it plus brief description. Ex. LBL[1:JD clapping] 5 pts

B) Describing the Register INSIDE THE R. Using your name included. Ex. R[1:JD P & P counter] 5 pts

C) Placement of the LBL and JMP LBL commands 10 pts

D) Having it clap 3 times only. 10 pts

E) Run in automatic. 5 pts

F) Show the instructor multiple screens. 5 pts

G) Saving your program onto a jump drive. 10 pts

H) Printing the program. 10 pts

**I)** **Written explanation of your program. 20 pts.**

J) The programs below will do what? 10 pts each total of 20 pts.

INSTRUCTOR’S INITIAL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What would happen when the program below was ran? Please use the data table for the R[1] and walk line for line through the program. HINT: Read the program for what it is not what you think it should be.

1: Lbl [1: start of program]

2: R[1] = 0

3: J P[1] 100% fine

4: J P[2] 100% fine

5: J P[3] 100% fine **How many times will the loop run? 10 pts.**

6: J P[4] 100% fine

7: R[1] = R[1] + 1

8: If R[1]>= 8, Jmp Lbl [2]

9: Jmp Lbl [1]

10: Lbl [2]

11: END

1: R[1] = 0

2: Lbl [1: start of program]

3: J P[1] 100% fine

4: J P[2] 100% fine

5: J P[3] 100% fine **How many times will the loop run? 10 pts.**

6: J P[4] 100% fine

7: R[1] = R[1] + 1

8: If R[1]<= 10, Jmp Lbl [1]

9: END

