Robotics, Integration, and Automation

THE MAIN JOB

Name	Class/Period	Date

1. Overview

In this lab activity, you will create a job which will run throughout the execution of the manufacturing application (project). When prompted by the PLC, this job will call the other jobs that you have already created.

> ***Note:*** *In order to differentiate between the programmable logic controller and the robot controller, the two devices are referred to as the PLC and the robot, respectively.*

2. Performance Objectives

After completing this lab activity, you will be able to:

- Create the main job for the manufacturing application.
- Zero output group values.
- Call other jobs from the main job.
- Loop the main job.
- Add comments to the main job.
- Adjust the jobs that energize digital output devices.

3. Required Materials

You need the following materials to complete the lab activity:

- SmartCart 4.0
- Computer
- Ethernet cables

4. Required Software

Logix Designer is required for this lab activity. It is included in the Studio 5000 suite. Ensure that the software is installed on your PC and has a valid license. If you are having problems installing or licensing the software, contact your instructor or IT manager.

5. Inventory and Safety

Before beginning the lab activity, review this checklist and mark off each item as you complete it.

- ☐ All hardware components are available for this lab activity.
- ☐ Hands, hair, and clothing are securely away from the work area.
- ☐ The work area is clean and devoid of food or drink.

- ☐ Review the SmartCart safety guidelines.
- ☐ Read through the entirety of this lab activity to familiarize yourself with the requirements.

6. Additional Background Knowledge

6.1. Zeroing Output Groups

In the lab activity tasks, you will adjust your robot jobs to turn output group (OG) bits on or off. This will allow the PLC to know when a particular job is in progress or has been completed.

Before the main job begins to call the other jobs, all of the OG bits should be reset to 0. The easiest way to do this is to zero all of the OGs at the top of the main job. You will be shown how to do this using the Inform List.

6.2. The "Alive" Bit

Since the PLC is the brain of the SmartCart and is responsible for overall project performance, the PLC will be in operation even before the main robot job is executing. The PLC, therefore, needs a way to know whether or not the robot is ready to start job execution. To accomplish this, the robot will send a signal to the PLC – it will turn on a specific bit to tell the PLC that it is ready. This bit is called the "alive" bit.

Based on the PLC logic that you will program, only if the PLC has received the alive bit signal will it send a start signal to the robot.

6.3. Other Notes About the Robot Jobs

Here are some other relevant ideas to note before starting the lab activity:

- The main job will have multiple CALL instructions and will be looped.
- The CALL instructions will be conditional. They will only execute if the specified input bits are turned on.
- Other jobs will turn a shared bit on at the beginning of the job execution and will turn the same bit off at the end.
- Jobs that turn on digital output devices (with the exception of the gripper jobs) will no longer use timers. Instead, the PLC will have complete control over these jobs. The job execution will loop until it receives a signal from the PLC.
- Make sure that you document your jobs and PLC routines (using comments) as well as add any descriptions to relevant PLC tags. If it is your preference, you may use alias tags in place of the robot module base tags.

7. Lab Activity

7.1. Adjusting Robot Pick-and-Place Jobs

In this task, you will add DOUT instructions to the beginning and end of two pick-and-place jobs so that the PLC will be able to know when the program execution has started or finished for those jobs.

Perform these steps:

1. Power up the I/O box. Wait for the PLC to power on.
2. Power up the robot. Ensure that the mode key on the pendant is set to Teach mode.
3. Power up the air compressor.
4. Using the pendant, navigate to **JOB > SELECT JOB**.
5. Select a pick-and-place job that you have previously created.

   The job opens. An example is shown here.

```
//NAME ROTARY2CONVEYOR
0000 NOP
0001 MOVJ P001 VJ=15.00
0002 MOVJ P005 VJ=15.00
0003 CALL JOB:GRIP_OPEN
0004 MOVL P006 V=60.0
0005 CALL JOB:GRIP_CLOSE
0006 MOVL P005 V=60.0
0007 MOVJ P007 VJ=15.00
0008 MOVJ P003 VJ=15.00
0009 MOVL P004 V=60.0
0010 CALL JOB:GRIP_OPEN
0011 MOVL P003 V=60.0
0012 MOVJ P007 VJ=15.00
0013 MOVJ P001 VJ=15.00
0014 CALL JOB:GRIP_CLOSE
0015 END
```

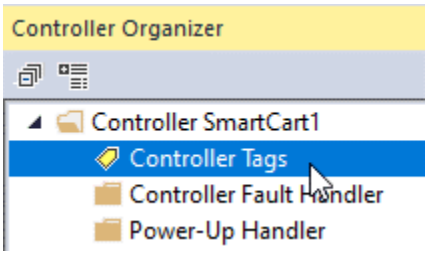
6. At the beginning of the job, add an instruction to turn on one of the bits on OG#3. In this example, OT#0019 is used, and it is recommended that you use the same bit. This bit corresponds to Robot:I.Data[0].2. When Robot:I.Data[0].2 is on, the PLC knows that this job is executing.

```
0000 NOP
0001 DOUT OT#(19) ON
0002 MOVJ P001 VJ=15.00
0003 MOVJ P005 VJ=15.00
0004 CALL JOB:GRIP_OPEN
0005 MOVL P006 V=60.0
…
```

7. At the end of job, add an instruction to turn off the same bit. When Robot:I.Data[0].2 turns back off, the PLC knows that the job is complete.

```
…
0013 MOVJ P007 VJ=15.00
0014 MOVJ P001 VJ=15.00
0015 CALL JOB:GRIP_CLOSE
0016 DOUT OT#(19) OFF
0017 END
```

8. Open your Logix Designer PLC programming project from the previous lab activity.
9. Remove all program logic. Delete all subroutines and delete the rungs in the main routine.
10. Download the project to the PLC and go online in **Rem Run** mode. Ensure that the *IO OK* LED is displayed.
11. Open the Data Monitor (Controller Tags list > Monitor tab).



12. Open the robot module input tags and expand the SINT tag for OG#3.

Name	Value	Data Type	Descriptio
Robot:I.Data	{...}	SINT[32]	
Robot:I.Data[0]	0	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	0	BOOL	OUT#0018
Robot:I.Data[0].2	0	BOOL	OUT#0019
Robot:I.Data[0].3	0	BOOL	OUT#0020
Robot:I.Data[0].4	0	BOOL	OUT#0021
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

ⓘ ***Note:*** *In the next step, you will play the job. Ensure all safety measures are in place. There is no need to use workpieces in this task or in the rest of the tasks of this lab activity.*

13. Using the pendant, switch the mode key to Play mode, and with servo power on, run the job that you just modified. When running the job, note the value of Robot:I.Data[0].2. The tag should turn on when the job starts running and should turn off when the job is complete.

Name	Value	Data Type	Descriptio
Robot:I.Data	{...}	SINT[32]	
Robot:I.Data[0]	4	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	0	BOOL	OUT#0018
Robot:I.Data[0].2	1	BOOL	OUT#0019
Robot:I.Data[0].3	0	BOOL	OUT#0020
Robot:I.Data[0].4	0	BOOL	OUT#0021
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

14. Switch the pendant's mode key back to Teach mode.

15. Repeat this task for a second pick-and-place job. Use bit **OUT#0021** instead of OUT#0019.

7.2. Adjusting a Digital Output Job

In this task, you will take a job used to operate the rotary table and modify it so that data can be exchanged between the job and the PLC.

Perform these steps:

1. Using the pendant, navigate to **JOB > SELECT JOB**.
2. Select the job that operates the rotary table. (If you have not yet created this job, create it now.)

   The job opens. An example is shown here.

```
//NAME ROTARY_TABLE
0000 NOP
0001 DOUT OT#(8) OFF
0002 DOUT OT#(4) ON
0003 TIMER T=10.000
0004 DOUT OT#(4) OFF
0005 END
```

ⓘ ***Note:*** *The above job is undocumented. Recall that OUT#0008 controls the direction of the rotary table, while OUT#0004 controls its on/off status.*

3. As in the previous task, add a DOUT instruction at the beginning of job to turn on a bit in OG#3. At the end of the job, add a DOUT OFF instruction for that same bit. Do not use any of the bits that you used previously. In this example, OUT#0020 is used, and it is recommended that you use the same bit.

```
0000 NOP
0001 DOUT OT#(20) ON
0002 DOUT OT#(8) OFF
0003 DOUT OT#(4) ON
0004 TIMER T=10.000
0005 DOUT OT#(4) OFF
0006 DOUT OT#(20) OFF
0007 END
```

4. We do not want the turn table to be operated by a timer. Rather, we want it to be controlled by the PLC. Therefore, delete the TIMER instruction…

```
0000 NOP
0001 DOUT OT#(20) ON
0002 DOUT OT#(8) OFF
0003 DOUT OT#(4) ON
0004 TIMER T=10.000
0005 DOUT OT#(4) OFF
0006 DOUT OT#(20) OFF
0007 END
```

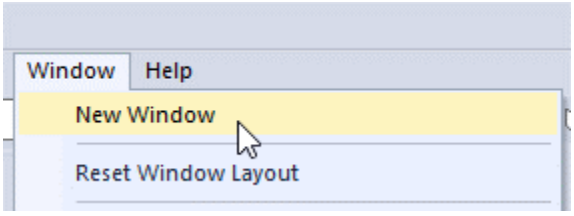
5. …and replace it with a **WAIT IN#(20)=OFF** instruction.

```
0000 NOP
0001 DOUT OT#(20) ON
0002 DOUT OT#(8) OFF
0003 DOUT OT#(4) ON
0004 WAIT IN#(20)=OFF
0005 DOUT OT#(4) OFF
0006 DOUT OT#(20) OFF
0007 END
```

This instruction will cause the job to wait until the robot receives a signal from the PLC when Robot:O.Data[0].3, the corresponding BOOL tag, turns off. The rotary table will then stop turning and the job will execute to the end line.

The PLC will initially have to energize Robot:O.Data[0].3, an action that will be recognized by the main job which you will program later.

6. In Logix Designer, return to the input module tags. In the top menu, navigate to **Window > New Window**. This opens a second instance of the PLC tags.



7. In the new window, collapse the input tags. Navigate to the output modules tags and expand the IG#3 SINT tag (Robot:O.Data[0]).

Name	Value	Data Type	Descript
Robot:O	{...}	AB:ETHER...	
Robot:O.Data	{...}	SINT[32]	
Robot:O.Data[0]	0	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	0	BOOL	IN#0018
Robot:O.Data[0].2	0	BOOL	IN#0019
Robot:O.Data[0].3	0	BOOL	IN#0020
Robot:O.Data[0].4	0	BOOL	IN#0021
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

8. Energize Robot:O.Data[0].3 by changing its value to **1**.

Name	Value	Data Type	Descript
Robot:O.Data[0]	8	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	0	BOOL	IN#0018
Robot:O.Data[0].2	0	BOOL	IN#0019
Robot:O.Data[0].3	1	BOOL	IN#0020
Robot:O.Data[0].4	0	BOOL	IN#0021
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

9. Select the other Controller Tags window and display the input tags from OG#3.

Name	Value	Data Type	Descriptio
Robot:I.Data[0]	0	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	0	BOOL	OUT#0018
Robot:I.Data[0].2	0	BOOL	OUT#0019
Robot:I.Data[0].3	0	BOOL	OUT#0020
Robot:I.Data[0].4	0	BOOL	OUT#0021
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

10. On the pendant, turn the mode key to Play mode. With servo power on, run the job.

The rotary table starts rotating. Note the value of Robot:I.Data[0].3, which is now 1.

Name	Value	Data Type	Descriptio
Robot:I.Data[0]	8	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	0	BOOL	OUT#0018
Robot:I.Data[0].2	0	BOOL	OUT#0019
Robot:I.Data[0].3	1	BOOL	OUT#0020
Robot:I.Data[0].4	0	BOOL	OUT#0021
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

11. Select the window that has the output module tags open. Change the value of **Robot:O.Data[0].3** from 1 to **0**. This stops the rotary table.

Name	Value	Data Type	Descript
Robot:O	{...}	AB:ETHER...	
Robot:O.Data	{...}	SINT[32]	
Robot:O.Data[0]	0	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	0	BOOL	IN#0018
Robot:O.Data[0].2	0	BOOL	IN#0019
Robot:O.Data[0].3	0	BOOL	IN#0020
Robot:O.Data[0].4	0	BOOL	IN#0021
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

12. Return to the input tags. Note that Robot:I.Data[0].3 is now off.

Name	Value	Data Type	Descriptio
Robot:I.Data[0]	0	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	0	BOOL	OUT#0018
Robot:I.Data[0].2	0	BOOL	OUT#0019
Robot:I.Data[0].3	0	BOOL	OUT#0020
Robot:I.Data[0].4	0	BOOL	OUT#0021
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

13. Add additional detail to the descriptions of the input tags so that you can easily recognize which jobs the tags correspond to.

Robot:I.Data[0].2	0	BOOL	OUT#0019 Rotary2Conveyor
Robot:I.Data[0].3	0	BOOL	OUT#0020 Rotary_Table
Robot:I.Data[0].4	0	BOOL	OUT#0021 Conveyor2Rotary

14. Return the pendant's mode key to Teach mode.

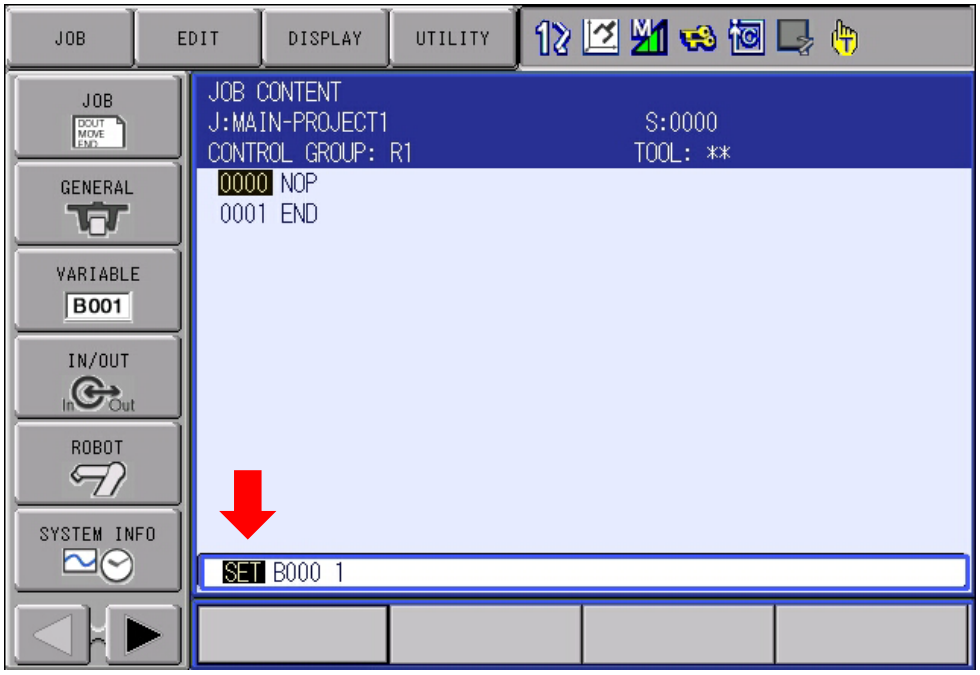
7.3. Building the Main Job

In this task, you will create the main job. This job will be initiated from the pendant, and its objective will be to call the other jobs when prompted by the PLC.

Currently, you should have three jobs that exchange data with the PLC: two pick-and-place jobs and one job that controls the rotary table.

Perform these steps:

1. On the pendant, navigate to **JOB > CREATE NEW JOB**.
2. Create a new job. Name it MAIN-PROJECT1 or similar.
3. In the first program line, add an instruction to set all of the bits in OG#3 to 0. To do so:
   a. Open the Inform List and navigate to **ARITH > SET**.
   b. In the input buffer line at the bottom of the screen, move the cursor to **SET** and then press SELECT.



c. The instruction's DETAIL EDIT screen is displayed. Select the double arrow button to the right of B000.

d. From the dropdown list, select **OG#()**.

e. Select the **1** to the right of OG#().

f. Change the value to **3**.

g. Press ENTER.

h. In the input buffer line, change the 1 to **0**.

i. Press ENTER to add the instruction to the job.

Your job should now look like this:

```
0000 NOP
0001 Set OG#(3) 0
0002 END
```

ⓘ ***Note:*** *For the purpose of brevity, the lines of code shown here are undocumented. You should be adding comments where relevant.*

4. Add a DOUT instruction which will turn on the "alive" bit. For the alive bit, use OUT #0018, which corresponds to Robot:I.Data[0].1.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 END
```

5. Add a WAIT IN#(18) ON instruction. This makes the job wait for a signal from the PLC (Robot:O.Data[0].1) before continuing.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 WAIT IN#18 ON
0004 END
```

ⓘ *Note: For simplicity, we have used the same bit number. However, any unused input bit from IG #3 onwards be used (and any unused output bit from OG #3 onwards can be used as the alive bit).*

6. In the next instruction line, add a label (INFORM LIST > CONTROL > LABEL). At the end of program, you will add a conditional jump instruction to return to this label.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 WAIT IN#18 ON
0004 *TOP
0005 END
```

7. On the next instruction line, add an instruction to CALL the first pick-and-place job. The instruction should be conditional: only call the job if a specific input bit is turned on. In this example, IN #0019 is used, which corresponds to Robot:O.Data[0].2. Recall that to add a condition, use the DETAIL EDIT page of the instruction, which is similar to what you did in step 3.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 WAIT IN#18 ON
0004 *TOP
0005 CALL JOB:ROTARY2CONVEYOR IF IN#(19)=ON
0006 END
```

8. For the next two lines, add conditional call instructions for the job that controls the rotary table and the second pick-and-place job. Use different input bits for each job. Again, for simplicity, we have chosen the same bit numbers for the jobs that we used for the output bits within the jobs themselves.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 WAIT IN#18 ON
0004 *TOP
0005 CALL JOB:ROTARY2CONVEYOR IF IN#(19)=ON
0006 CALL JOB:ROTARY_TABLE IF IN#(20)=ON
0007 CALL JOB: CONVEYOR2ROTARY IF IN#(21)=ON
0008 END
```

9. Add a conditional jump command (INFORM LIST > CONTROL > LABEL) to return to the label if input #18 is still on.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 WAIT IN#18 ON
0004 *TOP
0005 CALL JOB:ROTARY2CONVEYOR IF IN#(19)=ON
0006 CALL JOB:ROTARY_TABLE IF IN#(20)=ON
0007 CALL JOB: CONVEYOR2ROTARY IF IN#(21)=ON
0008 JUMP *TOP IF IN#(18) ON
0009 END
```

10. Add an instruction to clear OG #3 (This is the same instruction in line 0001 and may be copied from there). Note that this instruction also turns off the alive bit.

```
0000 NOP
0001 Set OG#(3) 0
0002 DOUT OT#(18) ON
0003 WAIT IN#18 ON
0004 *TOP
0005 CALL JOB:ROTARY2CONVEYOR IF IN#(19)=ON
0006 CALL JOB:ROTARY_TABLE IF IN#(20)=ON
0007 CALL JOB: CONVEYOR2ROTARY IF IN#(21)=ON
0008 JUMP *TOP IF IN#(18) ON
0009 Set OG#(3) 0
0010 END
```

11. Add comments to the job if you have not done so already.

12. Move the cursor to job line 0000.

13. Turn the pendant's mode key to Play mode.

Warning: *You will play the job in the next step. Ensure that all safety measures are in place and that you are ready to press the emergency stop button if necessary.*

Note: *Do not use any workpieces for this task. Store all workpieces before continuing.*

14. Turn servo power on and run the job. The job execution waits for input from the PLC (at the WAIT IN#18 ON instruction line).

15. In Logix Designer, open the input tags. Note that Robot:I.Data[0].1 is on. This is the alive bit. Note it in the description.

Name	Value	Data Type	Description
Robot:I.Data	{...}	SINT[32]	
Robot:I.Data[0]	2	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	1	BOOL	OUT#0018 - Alive Bit
Robot:I.Data[0].2	0	BOOL	OUT#0019 Rotary2Conveyor
Robot:I.Data[0].3	0	BOOL	OUT#0020 Rotary_Table
Robot:I.Data[0].4	0	BOOL	OUT#0021 Conveyor2Rotary
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

16. Open the *output* tags. Change the value of Robot:O.Data[0].1 to **1**.

Name	Value	Data Type	Descript
Robot:O	{...}	AB:ETHER...	
Robot:O.Data	{...}	SINT[32]	
Robot:O.Data[0]	2	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	1	BOOL	IN#0018
Robot:O.Data[0].2	0	BOOL	IN#0019
Robot:O.Data[0].3	0	BOOL	IN#0020
Robot:O.Data[0].4	0	BOOL	IN#0021
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

Job execution progresses to the line with the label. Note, however, that in reality, the job is on a loop and is polling for additional input.

17. Turn Robot:O.Data[0].2 on. The robot begins executing the first pick-and-place job.

Name	Value	Data Type	Descript
◢ Robot:O	{...}	AB:ETHER...	
◢ Robot:O.Data	{...}	SINT[32]	
◢ Robot:O.Data[0]	6	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	1	BOOL	IN#0018
Robot:O.Data[0].2	1	BOOL	IN#0019
Robot:O.Data[0].3	0	BOOL	IN#0020
Robot:O.Data[0].4	0	BOOL	IN#0021
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

18. The robot will continuously execute the job until Robot:O.Data[0].2 is turned off. Turn off the tag. The robot completes the job and then stops.

◢ Robot:O.Data[0]	2	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	1	BOOL	IN#0018
Robot:O.Data[0].2	0	BOOL	IN#0019
Robot:O.Data[0].3	0	BOOL	IN#0020
Robot:O.Data[0].4	0	BOOL	IN#0021
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

19. Turn **Robot:O.Data[0].3** on (change its value to 1).

The rotary table turns on. It will run continuously until the Robot:O.Data[0].3 is turned off.

20. Turn Robot:O.Data[0].3 **off (0)**. The rotary table stops.

21. Turn Robot:O.Data[0].4 **on (1)**. This initiates the second pick-and-place job.

22. Allow the job to run several times, then turn Robot:O.Data[0].4 **off (0)**. The robot completes the job and then stops its motion.

23. Add a more detailed description to the output tags to include the jobs being executed by each tag.

Name	Value	Data Type	Description
Robot:O.Data[0]	0	SINT	IG#003
Robot:O.Data[0].0	0	BOOL	IN#0017
Robot:O.Data[0].1	0	BOOL	IN#0018 Start Main Job
Robot:O.Data[0].2	0	BOOL	IN#0019 Start Rotary2Conveyor
Robot:O.Data[0].3	0	BOOL	IN#0020 Start Rotary_Table
Robot:O.Data[0].4	0	BOOL	IN#0021 Start Conveyor2Rotary
Robot:O.Data[0].5	0	BOOL	IN#0022
Robot:O.Data[0].6	0	BOOL	IN#0023
Robot:O.Data[0].7	0	BOOL	IN#0024

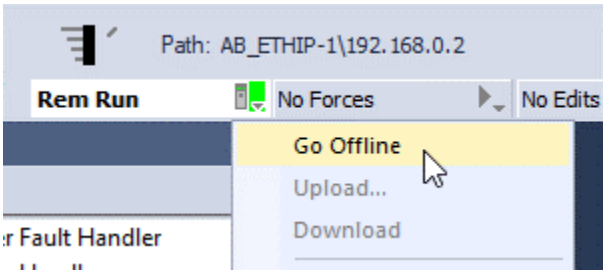
Note that the job is still polling. You can turn the tags Robot:O.Data[0].2, Robot:O.Data[0].3, and Robot:O.Data[0].4 on again (one at a time) to have the program call their respective jobs.

24. Turn Robot:O.Data[0].1 off. The job completes execution. Note that (in the input tags list) that the alive bit is now off.

Name	Value	Data Type	Description
Robot:I.Data	{...}	SINT[32]	
Robot:I.Data[0]	0	SINT	OG#003
Robot:I.Data[0].0	0	BOOL	OUT#0017
Robot:I.Data[0].1	0	BOOL	OUT#0018 - Alive Bit
Robot:I.Data[0].2	0	BOOL	OUT#0019 Rotary2Conveyor
Robot:I.Data[0].3	0	BOOL	OUT#0020 Rotary_Table
Robot:I.Data[0].4	0	BOOL	OUT#0021 Conveyor2Rotary
Robot:I.Data[0].5	0	BOOL	OUT#0022
Robot:I.Data[0].6	0	BOOL	OUT#0023
Robot:I.Data[0].7	0	BOOL	OUT#0024

25. Switch the pendant mode key to Teach mode.

26. Take the Logix project offline.



27. Save the project. You will use it in the next lab activity.

    In this activity, you manually changed the values of PLC tags to control the jobs being executed by the robot. In the next lab activity, you will add a PLC logic routine that will control the *sequence* of the jobs that are executed.

8. Authentic Skill Assessment

Have your instructor verify that your work meets the requirements in the performance objectives and sign below. Keep this lab activity sheet for future reference.

Instructor Signature	Date

9. Reset Steps

This lab activity does not have any reset steps.

10. Shutdown

Unless instructed otherwise by your instructor, review and complete each of the items on the checklist below.

- ☐ Jog the robot to a safe position with the gripper jaws pointing downwards.
- ☐ Return the pendant to its storage hook on the side of the SmartCart.
- ☐ Power down the robot.
- ☐ Power down the air compressor.
- ☐ Power down the I/O box.
- ☐ Close Logix Designer.