Robotics, Integration, and Automation

THE LOGIC SEQUENCER

Name	Class/Period	Date

1. Overview

In the previous lab activity, you created a main job which, when prompted by the PLC, called other previously created jobs. Not including any comments that you may have added, the job had these lines of code:

```
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
```

After playing the job, you had to manually adjust the values of the relevant PLC tags.

In this activity, you will program a PLC routine that controls the sequence of the initiation of each of the jobs.

> ***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 a routine that initiates specified robot jobs.
- Use ladder logic to control the order of job operation.
- Build a ladder logic sequencer.

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.

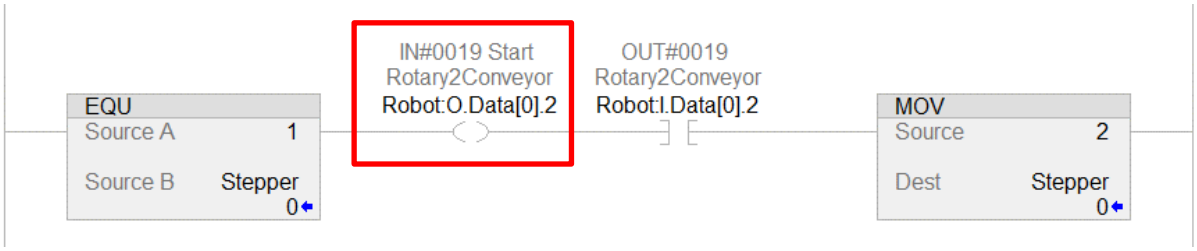
- ☐ All hardware components are available for this lab activity.
- ☐ Hands, hair, and clothing are securely away from the work area.
- ☐ The work area is clean and devoid of food or drink.
- ☐ Review the SmartCart safety guidelines.
- ☐ Read through the entirety of this lab activity to familiarize yourself with the requirements.

6. Lab Activity

6.1. Creating the Routine

In this task, you will create the ladder logic routine that controls the order of execution of the same robot jobs that you worked with in the previous activity. A “stepper”, created using a DINT tag as well as MOV (Move) and EQU (Equal to) instructions, will be used to build the sequence. Most of the rungs in the ladder diagram will begin with an EQU instruction which checks the stepper tag value and will end with a MOV instruction that will update the value of the stepper to the next stage in the sequence. Other instructions that check job progress and initiate jobs will be placed between the EQU and MOV instructions.

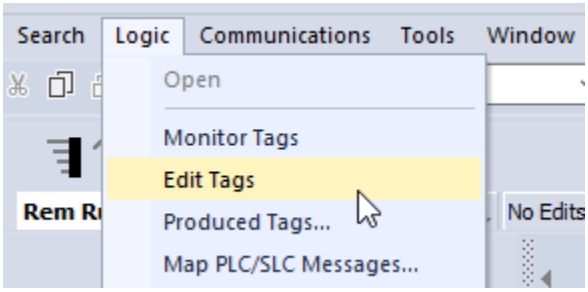
Below is an example rung. Note the location of the OTE (Output Energize) instruction. Normal convention is to always place OTEs at the end of a rung, but when building a sequence it is acceptable to place instructions in locations where they would not normally be found as long as the correct logic sequence is kept. You can see that Logix Designer is alright with this and does not return a compile error here.



An OTE placed to the left of an XIC.

Perform these steps:

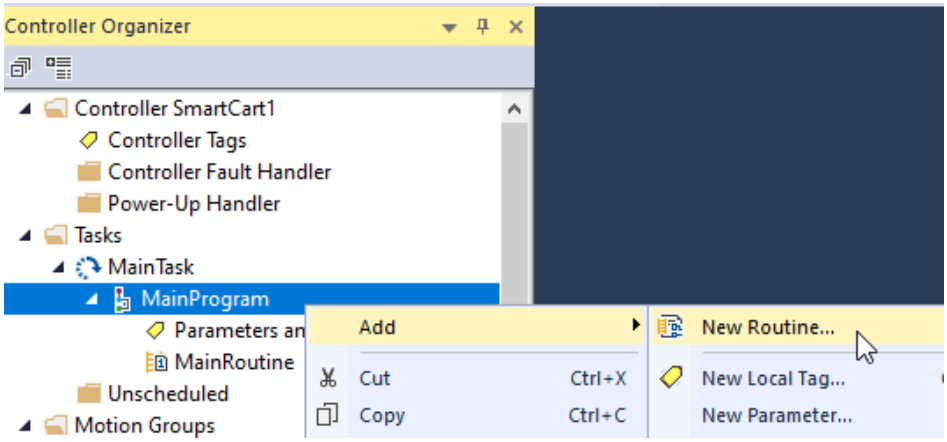
1. Run Studio 5000.
2. Open the project that you saved in the previous lab activity.
3. Navigate to **Logic > Edit Tags** to open the tag editor.



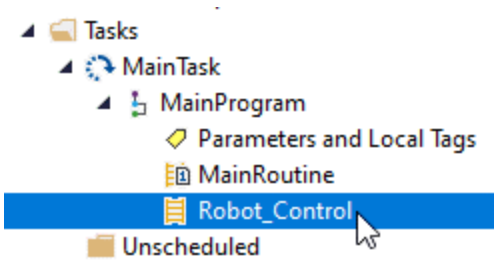
4. Create the following tags:

Name	Data Type	Explanation of Use
Stepper	DINT	Keeps track of the sequence.
Start	BOOL	Starts the sequence.
Reset	BOOL	Returns the stepper value to 0.
Rotary_Table_Timer	TIMER	Controls the duration of the rotary table job.

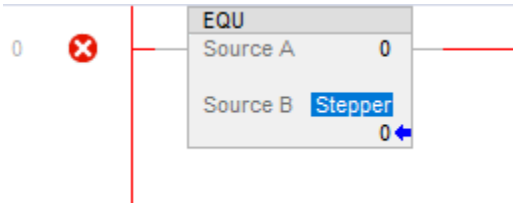
5. In the Controller Organizer, navigate to **Tasks > MainTask > MainProgram** and right-click **MainProgram**. Create a new ladder logic routine named Robot_Control or similar.



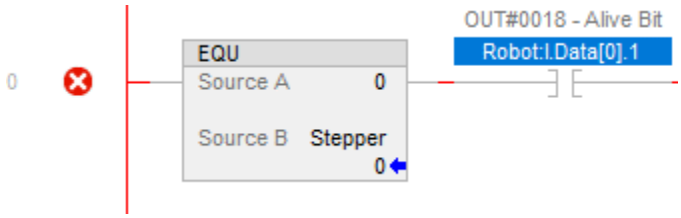
6. Double-click the new subroutine to open its ladder diagram.



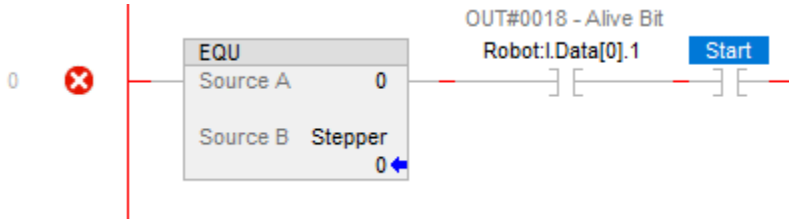
7. The stepper will initially start with a value of 0. This is the first condition for the sequence to start. Therefore, to rung 0 add an **EQU** instruction that compares the **Stepper** tag (Source B) to **0** (Source A) as shown.



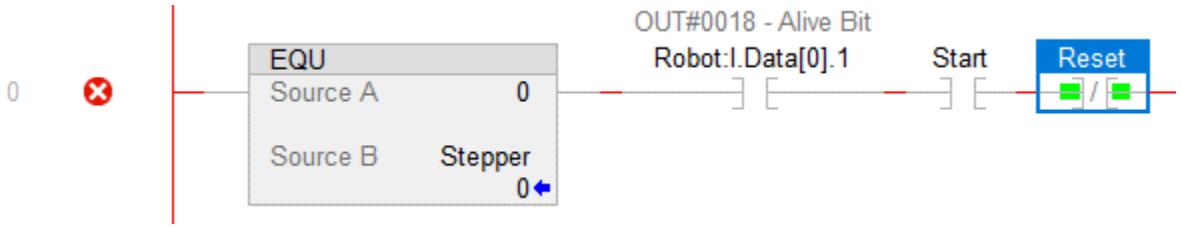
8. Another condition is that the alive bit is on. Add an **XIC** (Examine On) instruction addressed to the alive bit (**Robot:I.Data[0].1**).



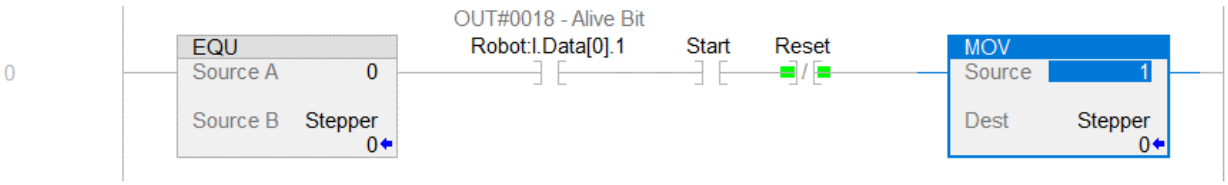
9. A third condition is that the start button is pressed. Add an **XIC** addressed to the **Start** tag. (Right now, you have no start button, but you will have one when you create your HMI screens in the next section. Instead, you will toggle this instruction to start the sequence.)



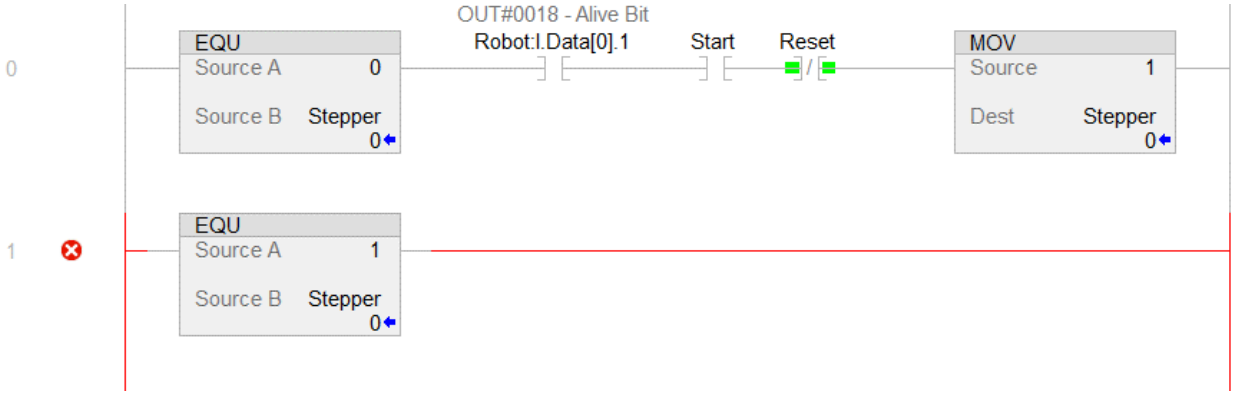
10. A fourth condition to start the sequence is that the reset button is *not* pressed. (It's always good practice to have a reset button). Add an **XIO** (Examine off) instruction addressed to the **Reset** tag.



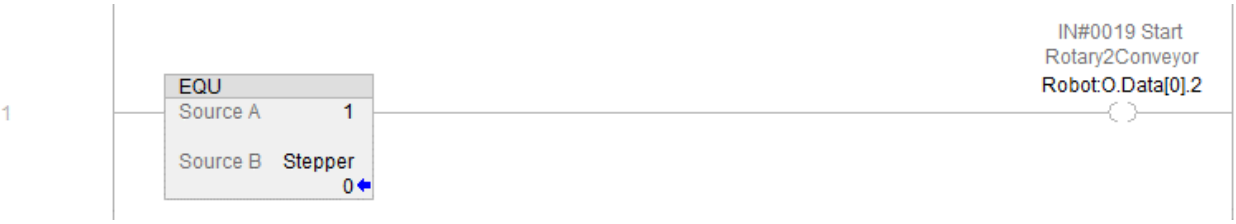
11. That was the last condition for starting the sequence. Complete the next rung with a MOV instruction that changes the stepper value to **1**.



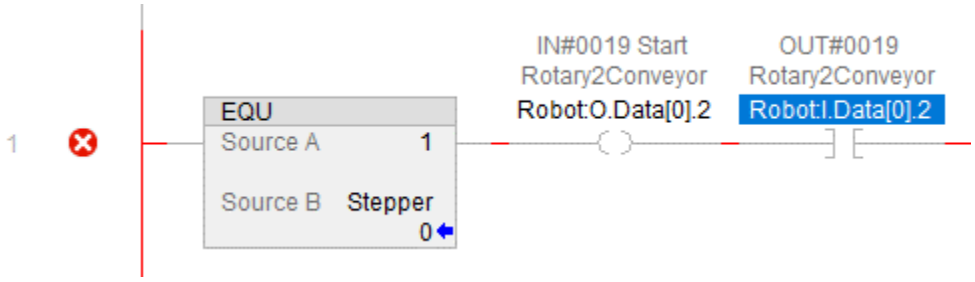
12. Add another rung, this time with an **EQU** instruction comparing the stepper to **1**.



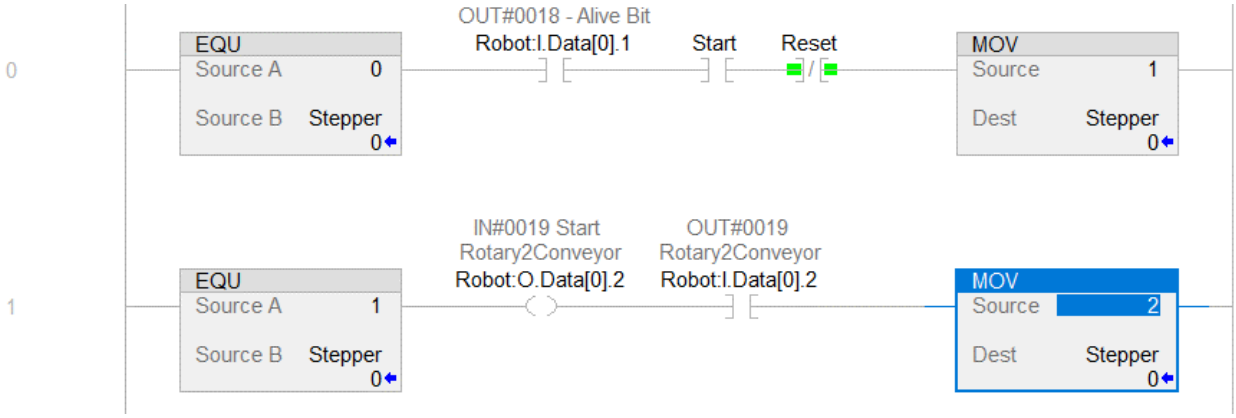
13. At this stage in the sequence, the first job should start. Add an **OTE** to the rung and address it to the output tag that initiates the first pick and place job, **Robot:O.Data[0].2**. (Recall from the previous lab activity that the job has a WAIT IN#19 ON instruction. IN#0019 is linked to Robot:O.Data[0].2.)



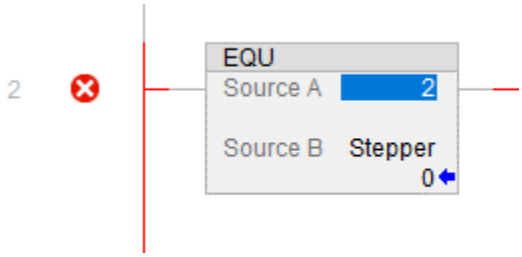
14. Before moving to the next stage of the sequence, the robot must acknowledge that the job has begun. *To the right of the OTE*, add an **XIC** addressed to **Robot:I.Data[0].2**. This tag is linked to the bit that turns on when the job starts (via the DOUT OT#19 ON instruction in the job.)



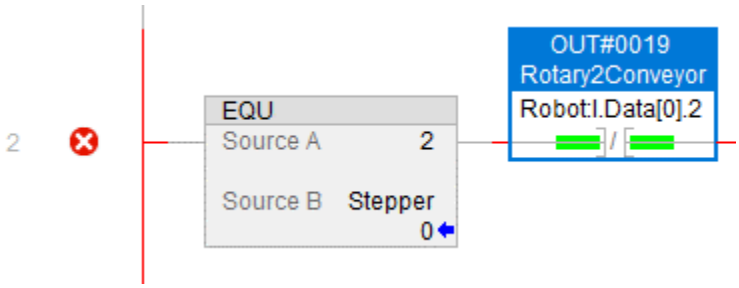
15. That's all for step 1. Add a **MOV** instruction which changes the value of the stepper to **2**.



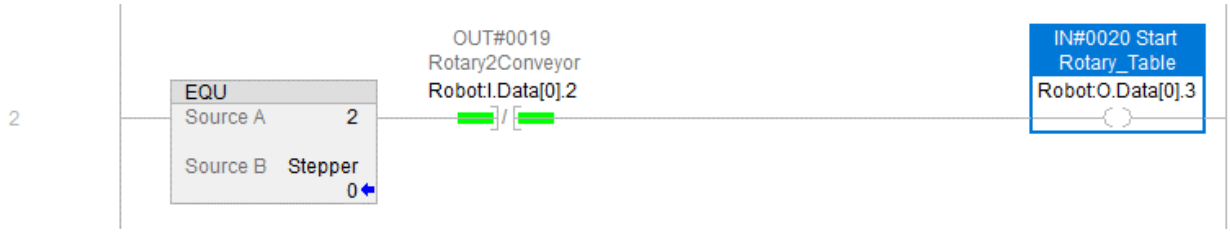
16. To ensure that step 2 has started, add a new rung with an **EQU** instruction that compares the stepper to **2**.



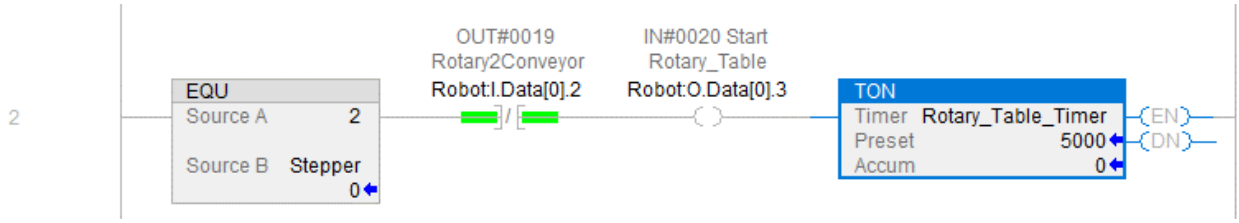
17. In this stage, we want the second job - the job that activates the rotary table – to start. First, however, a condition should be added to ensure that the first job is complete. Recall that Robot:I.Data[0].2 will turn off when the job completes because of the DOUT OT#19 OFF instruction on the last job line. Therefore, add an **XIO** addressed to **Robot:I.Data[0].2** after the EQU instruction.



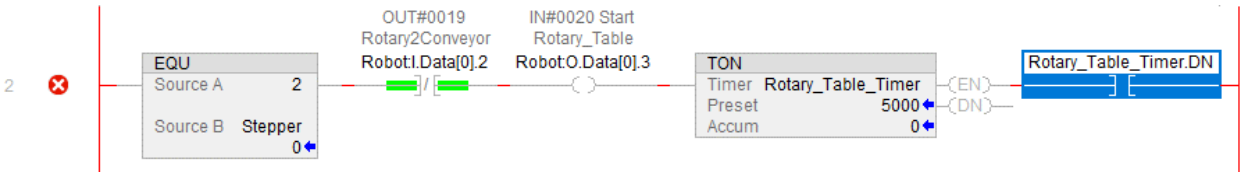
18. Add an **OTE** to start the rotary table job.



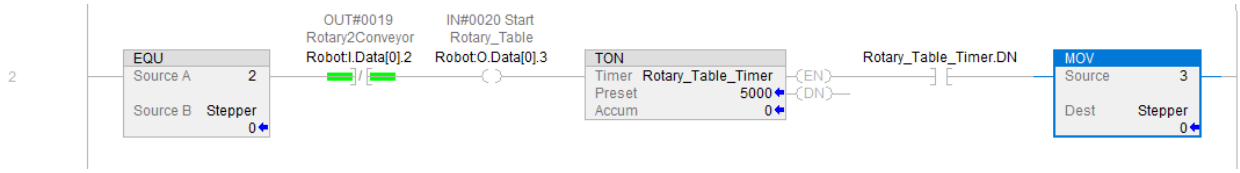
19. To the right of the OTE, add a **TON** addressed to the **Rotary_Table_Timer** tag. Set the preset to **5000** (5 seconds).



20. The timer must be complete before the next stage. Add an **XIC** addressed to the timer's **DN** tag member.

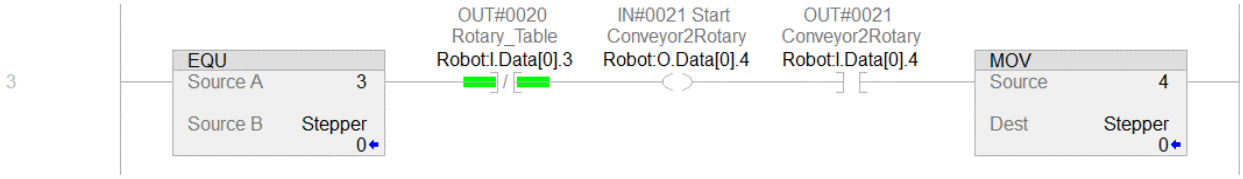


21. Complete the rung with a **MOV** instruction that changes the stepper value to **3**.

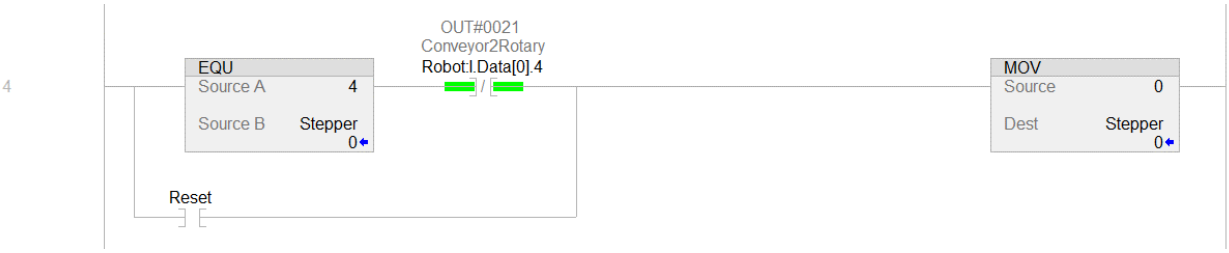


22. Add a new rung with instructions for the next step. Like the rungs preceding it…

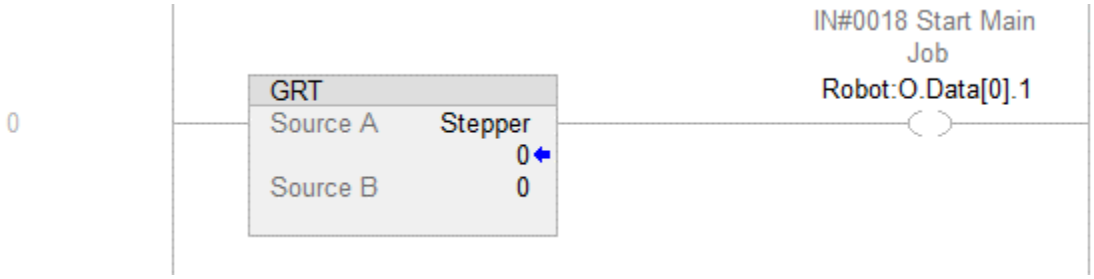
- The first condition is that the stepper is on the correct step (EQU).
- There is a condition that the previous job is complete (XIO).
- The next job (the second pick and place job) has been initiated (OTE).
- There is a condition that the robot has acknowledged that the job is initiated (XIC).
- The sequence moves to the next step (MOV).



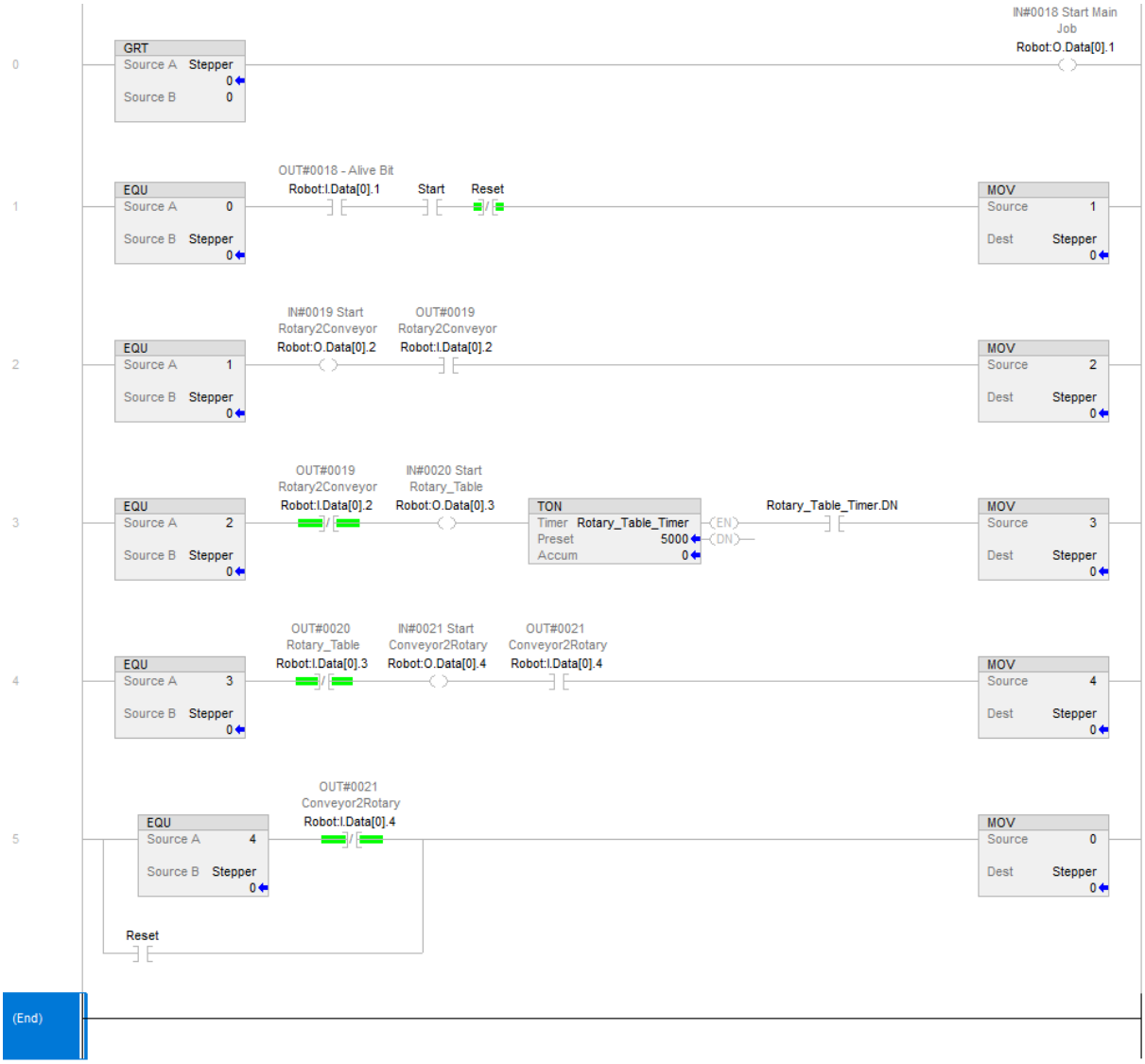
23. The last rung will be used to reset the stepper to 0. The conditions are that the stepper has reached step 4 AND the last job has completed execution OR the Reset button is pressed (the actual reset button will be added to an HMI screen in the next section). Create this rung now.



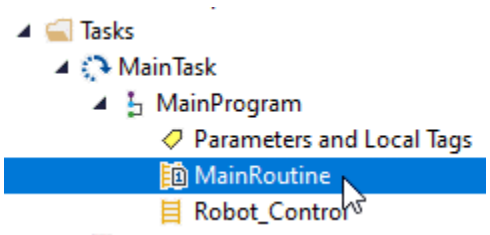
24. Recall that the main job can only be started and can later only poll for input when it receives a positive signal from Robot:O.Data[0].1 (IN# 18). This bit should only be on when the routine has been initiated, i.e., when it is at step 1 or later. Therefore, *at the top of the ladder diagram*, add a new rung that **energizes Robot:O.Data[0].1** if the **stepper** value is **greater than 0** (GRT instruction).



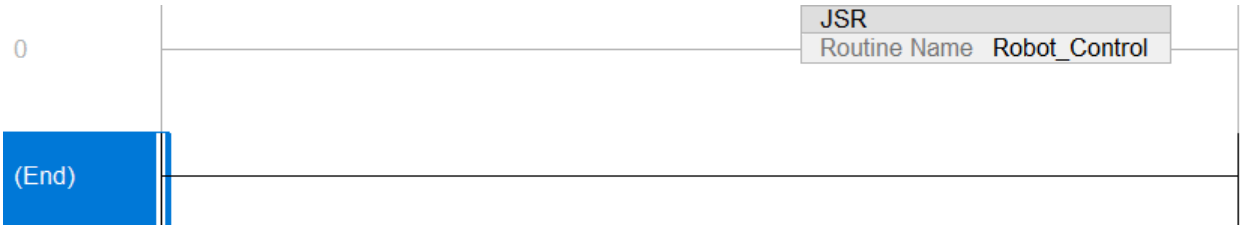
Your finished ladder routine should look like the one below. Note that this is just one way that you could have created this sequence. As in all things PLC programming, there are usually several different ways you can get something done.



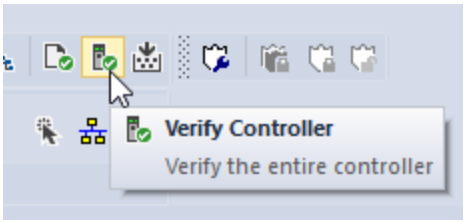
25. Double-click the MainRoutine to open its ladder diagram.



26. Add a **JSR** (Jump to Subroutine) instruction that jumps to the **Robot_Control** routine. Delete the extraneous parameters.



27. Verify the controller to ensure that there are no compile errors.



28. Save the project.

6.2. Testing the Sequence

In this task, you will go online with the controller and test the sequence.

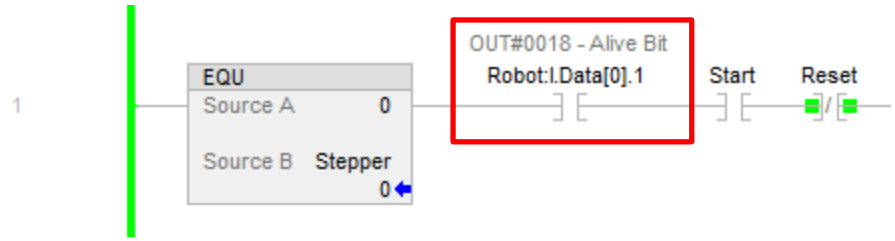
Perform the following steps:

1. Power up the I/O box. Wait for the PLC to power on.
2. Power up the robot and the air compressor.

ⓘ ***Tip:*** *Use Command Prompt to ping the PLC and the robot and confirm network connectivity.*

3. In Logix Designer, download the project to the PLC and stay online in Rem Run mode.

4. Open the Robot_Control routine. Note that the alive bit is off.

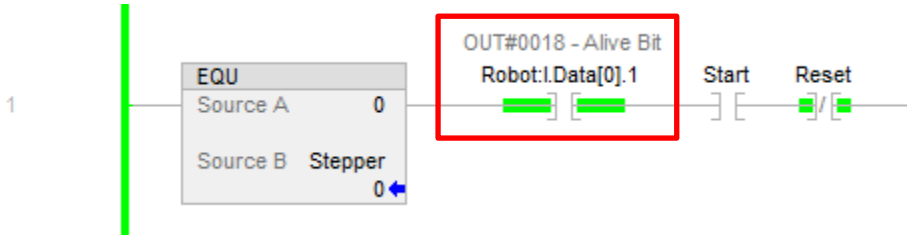


5. Using the robot's programming pendant, navigate to **JOB > SELECT JOB**.
6. Select the main job that you created in the previous activity.
7. Turn the pendant's mode key to Play mode (center position).

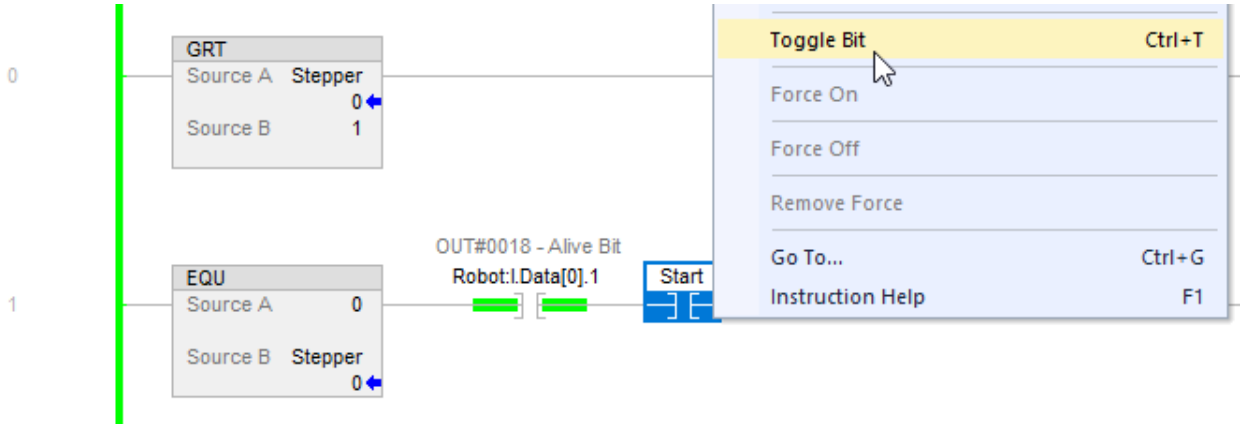
Warning: *You will play the job in the next step. Ensure that all safety measures are in place and that you are ready to press the emergency stop button if necessary.*

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

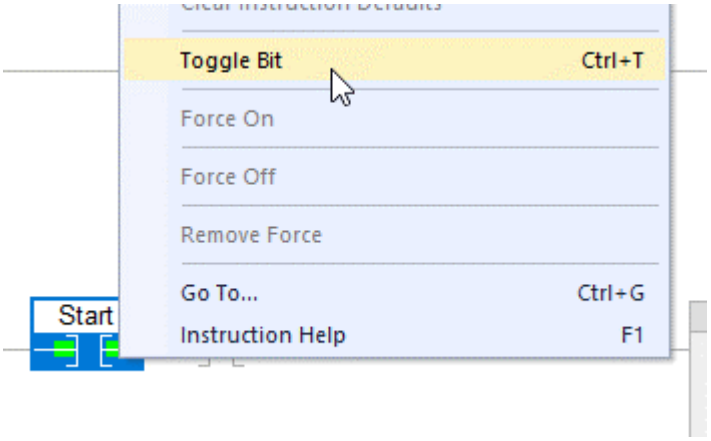
8. Turn servo power on and then press the green button to play the job. The job execution waits for input from the PLC. Note that the alive bit is now on.



9. Right-click the **Start** XIC and select **Toggle Bit**.

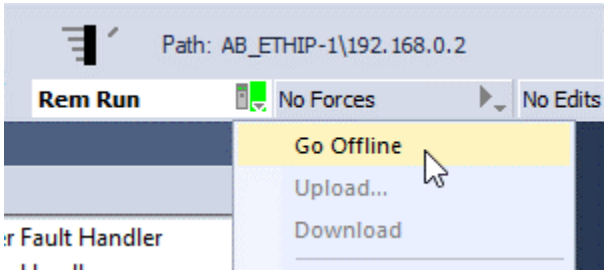


10. The robot begins executing the first job. Toggle the Start bit off.



The robot continues to execute the first job. It completes all three jobs in the specified sequence and then stops. Because the sequence is reset to 0 when it reaches step 4, Robot:O.Data[0].1 turns off and job execution is completed. The alive bit also turns off.

11. Run the job again and toggle the Start XIC. This time, however, do not toggle the XIC off. Wait for several cycles of jobs and then toggle the XIC off. Monitor the values of the tags while the system is running.
12. Test the **Reset** XIC by toggling it during various stages of operation.
13. Troubleshoot your PLC program and robot job if necessary.
14. Switch the pendant mode key to Teach mode.
15. Go offline.



16. Save the project. You will use it in the next lab activity.
17. If time remains, save the project with a different name and experiment by changing the order of the jobs in the sequence.

7. Summary and Next Steps

In this activity, you created a PLC program routine that controlled the order of jobs being executed by the robot. Here are some important notes:

- As mentioned above, there are several different ways that you could have designed this ladder diagram. In the future, feel free to try different programming strategies and techniques, especially when you go about building the project applications in Sections 7 and 8.
- The ladder diagram only controlled a few of the jobs that you created. When working on your project, you will integrate more jobs into the program.
- Also to be integrated into the program (whether in this subroutine or another) are the module tags of the smart device modules and the QC camera.
- Instead of toggling bits on a ladder diagram, you will use the HMI screen to operate the program and monitor the system feedback.

This completes the PLC section of the course, but it by no means concludes your time with the PLC. Good luck in the next section: *HMIs*.

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.
- ☐ Turn off the air compressor.
- ☐ Power down the I/O box.
- ☐ Close Logix Designer.