

Lab 2 Tutorial – Xinlei – 09/26/2024

Key points before start

- What's ladder logic programming? The difference between other programming languages, like C, C++, python...
 - Graphical programming, on/off control (0/1 control)
- Five basic components and format issue
- How to read a one-line program
- How to read a multiple-line program
 - Wait for physical response?
- Most important idea for writing ladder logic programs for automation (task 2)
 - One line should be executed **only once**, and the program should be executed from top to bottom
 - Automation vs intelligence (beyond the course)

1. Setup PLC Communication in RSLinx Classic Lite

1. Open **RSLinx Classic Lite** software
2. Navigate to **Communications > Configure Drivers**
3. **Delete** all currently configured drivers
4. **Create a new driver:**
 - Choose **RS-232**, click **Add New**
 - Assign a name to the driver
 - Select the correct communication port and set the baud rate to **9600**
 - Click **Auto-Configure**. You should see the message: "Auto Configuration Successfully"
 - Click **OK**, but do not close the software.

2. Open RSLogix Micro Starter Lite

1. Launch **RSLogix Micro Starter Lite** software.

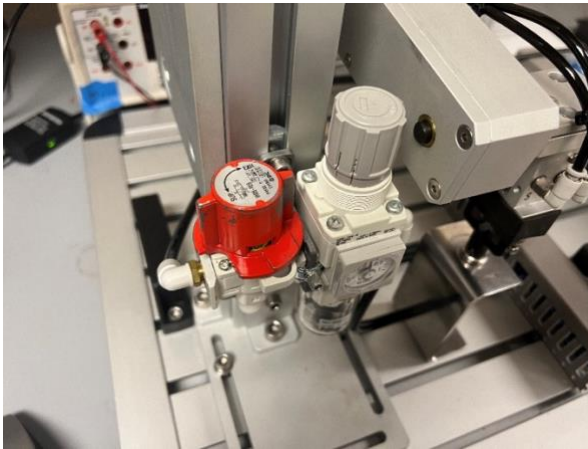
3. Configure the Communication Driver

1. Ensure the driver selected is the one you previously set up
2. Click **File > New** to create a new project

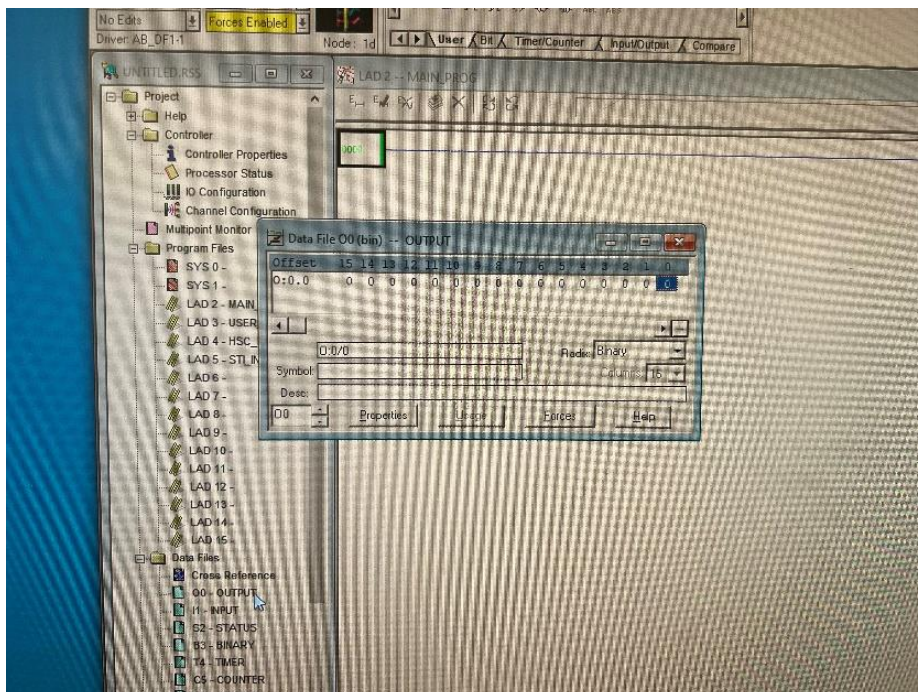
3. Select the correct machine number (the last one), verify the driver, and click **OK**

4. Test Connection to PLC

1. From the dropdown menu, select **Download** to transfer the blank project to the PLC
2. **Yes** to all prompts
3. Go **Online**
4. Enter the Run mode
5. Ensure the **red air valve switch** is connected.



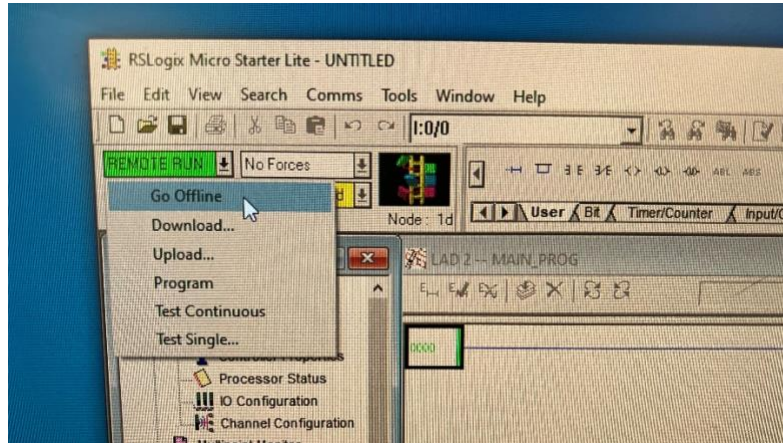
6. In the **Data Files**, double-click the output (mouse position on photo), and change it to test if the PLC is connected successfully, as did in the last lab.



5. Programming in PLC

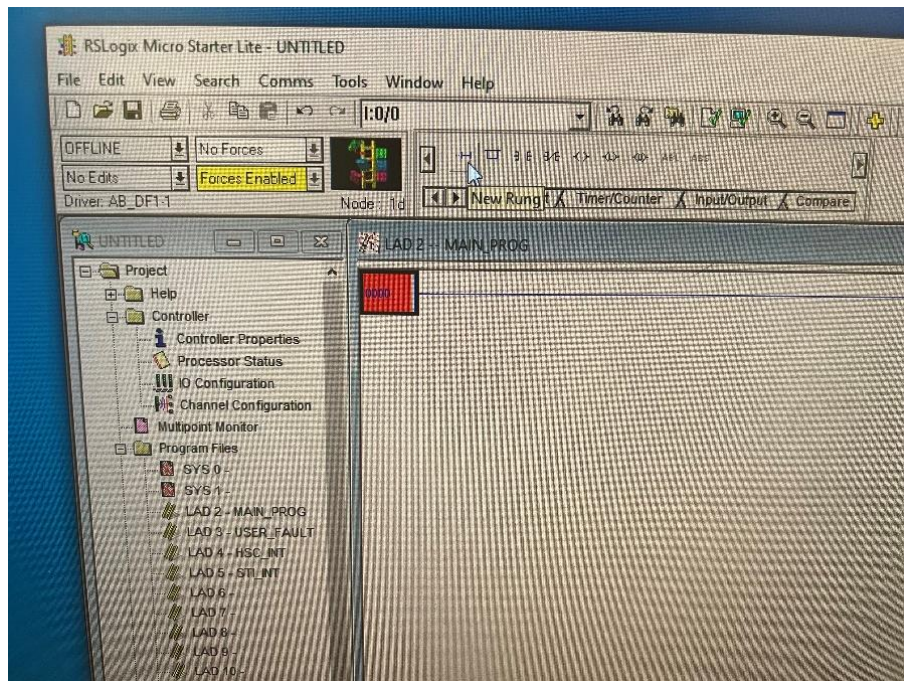
1. Go Offline:

- Always set all outputs to **0** before Go Offline
- Go offline



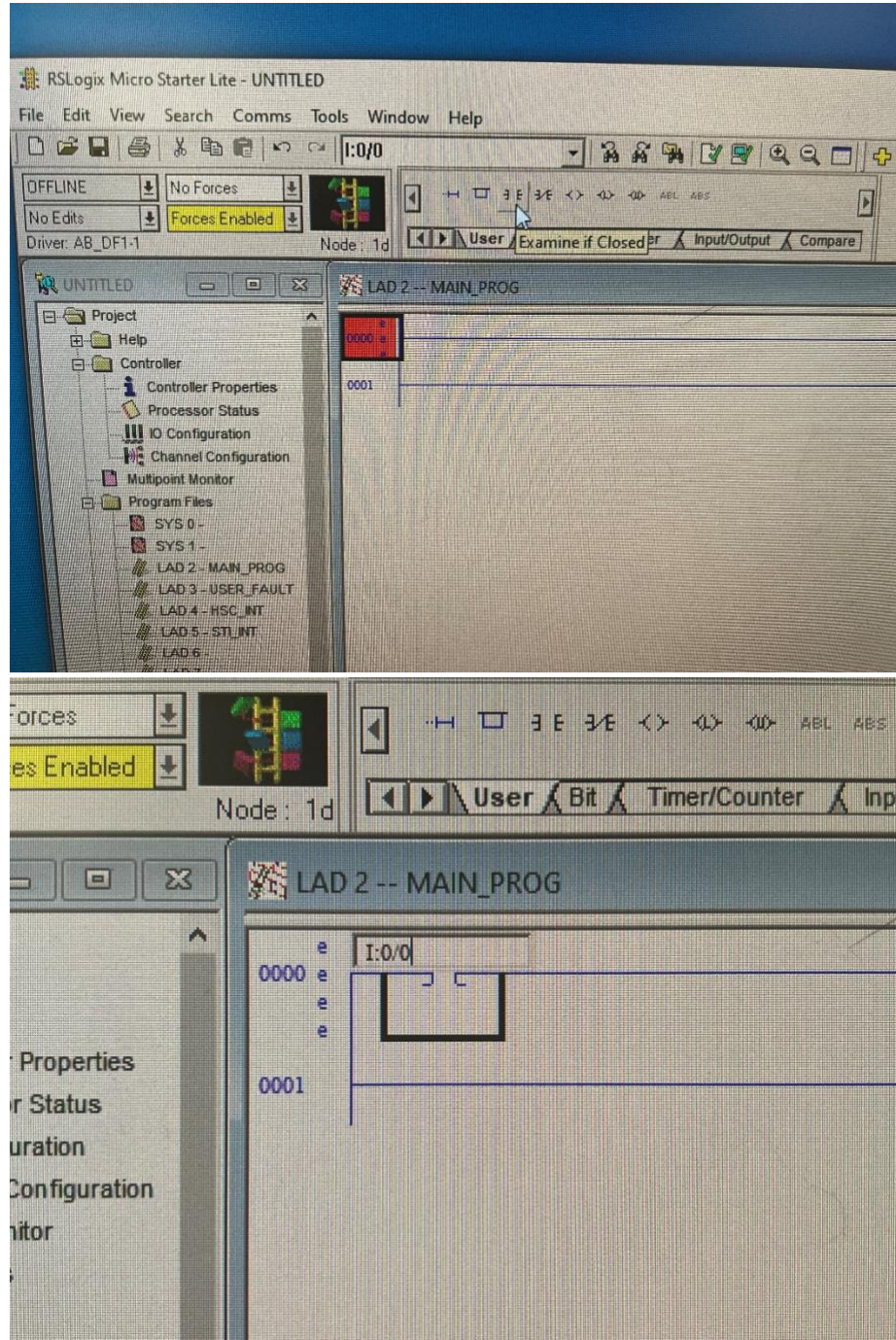
2. Add a New Rung:

- Click the red box first, then click **New Rung (mouse position)** to insert a rung into the program

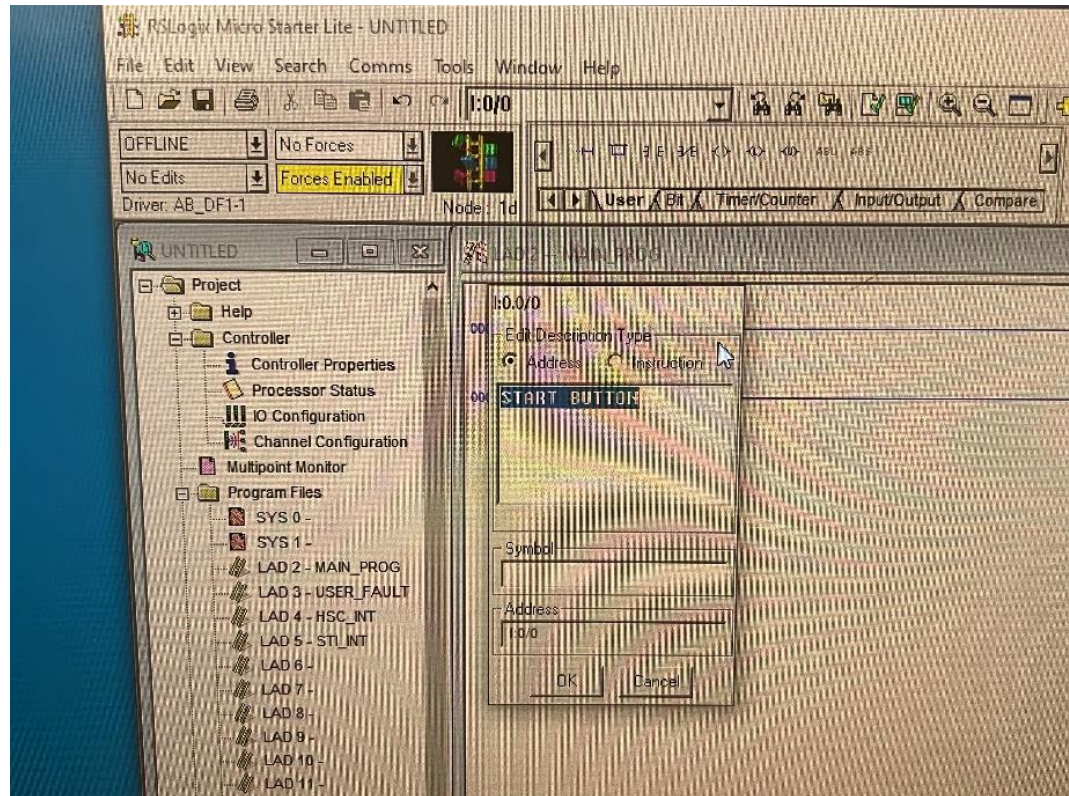


3. One Line Example Code 1:

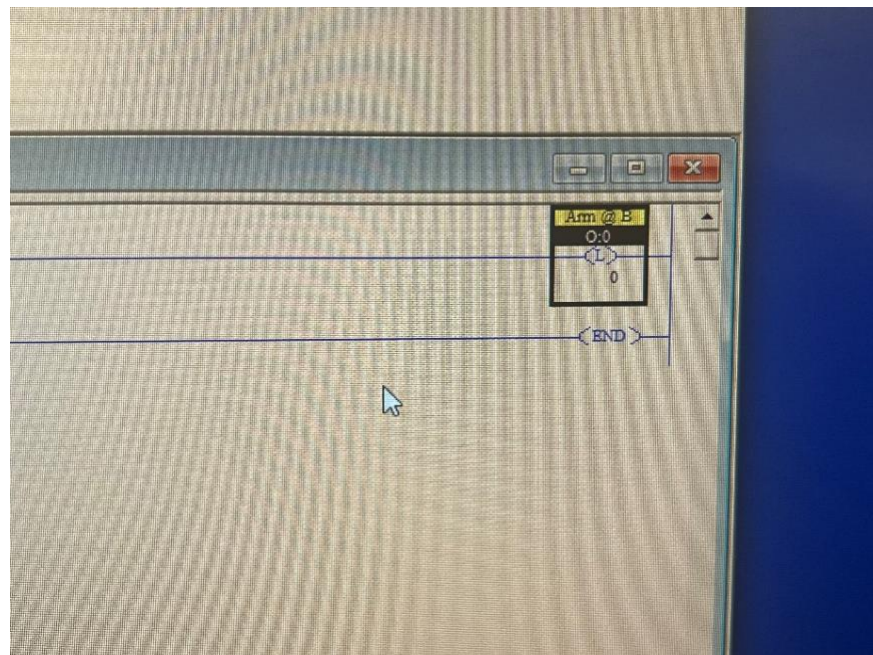
- Add an **Examine If Closed (XIC)** instruction. Double-click to edit and enter **I:0/0** (be careful about the format, the first bit is always 0).



- Press **Enter** and add a description for clarity



- Add a **Latch (OTL)** instruction with **O:0/0**, labeled **Arm to B (latch the bit zero in output will cause the arm moving from position A to position B)**. Typo error on the figure (“Arm @ B” should be “Arm to B”, “Arm @ B” should be the description for “I:0/5”, come to me if you don’t understand this)



4. Download and Run:

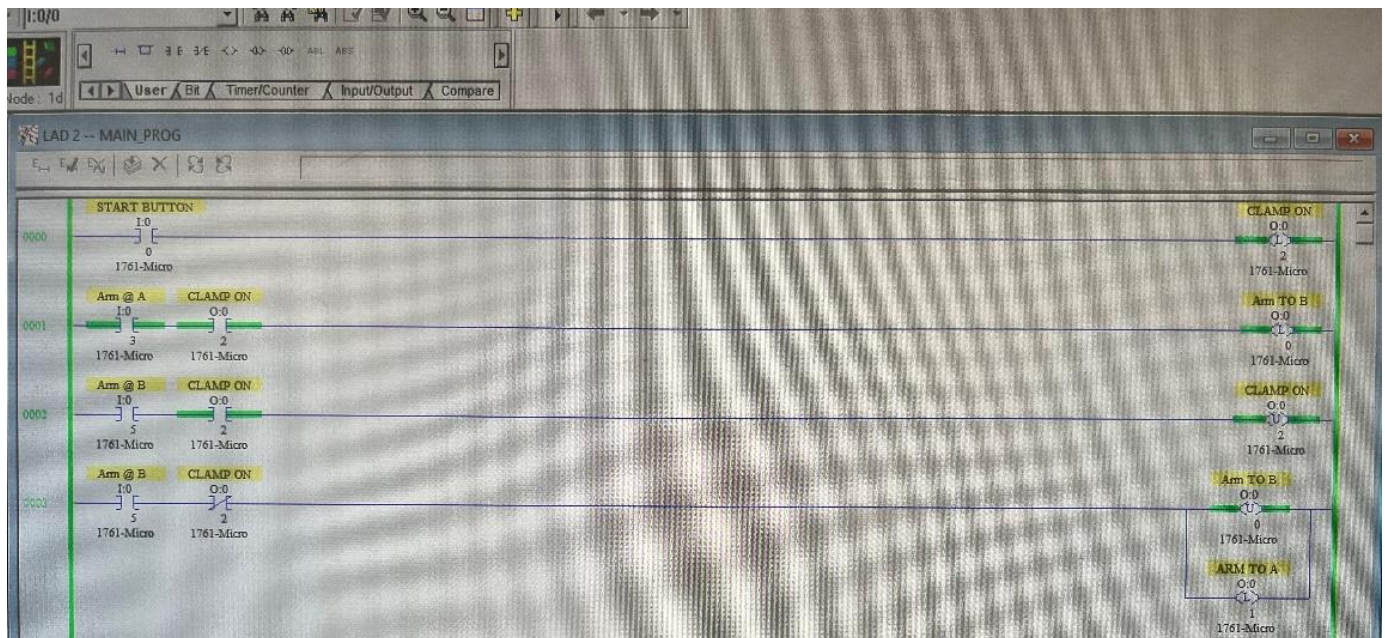
- Click **Download**, then switch to **PROG Mode** when prompted.
 - Switch to **RUN Mode**, and **Go Online**.
 - Press the **Start Button** to test the motion.
 - Adjust the output as mentioned in the previous section.
5. Click **Go Offline** to revise your code, and back to **step 4** to download and run the new code

-----Important-----

Before you start to finish task 1 and 2, please read,

- Only Branch, Examine if closed, Examine if opened, Latch, and Unlatch will be used. Don't need to use other blocks.
 - Try and finish example code 1 before starting task 1
 - Try or think about example code 2 before starting task 2
-

6. Example code 2 for task 2:



Human action	Code Description	
Hit the start button (Latch the input bit 0)	Check if the input bit 0 is 1	Latch the output bit 2 (Open the clamp)

	Check if the input bit 0 is 1 and if the output bit 2 is 1	Latch the output bit 0 (Move arm to position B)
	Check if the input bit 5 is 1 and if the output bit 2 is 1	Unlatch the output bit 2 (Close the clamp)
	Check if the input bit 5 is 1 and if the output bit 2 is 0	Unlatch the output bit 0 and latch the output bit 1 (move arm from position B to A)

Task 1 (50%)

Write a program such that,

- Hit the start button, the arm will move from initial position (normally A) to B
- Hit the stop button, the arm will move from B to A
- Switch the auto/manual, the clamp will open/close

After finish task 1, try the action below and observe the result

Action: Hit the start button first, and then hit the stop button before the arm stops at position B

Result: ?

Conclusion: ?

Task 2 (50%)

Write a program such that, once hit the start button, the machine will automatically finish the following actions in order:

- Pick up the object
- Move the object to the second position
- Release the object at the second position
- Then recollect the object and return it to the first position

Hints:

- Each line of the program should only be executed **once**, since you're programming the action **sequence** of the machine, and you want the machine to behave **in order**
- How to ensure each line only could be executed once?
 - You have redundant outputs 3,4,5 which are only related to LED indicators on PLC control box and won't cause any physical actions of the arm
- How many redundant outputs you need to use as flags to make each line different?