Programmable Logic Control (PLC) Supplemental Material

Interdisciplinary Automatic Controls Laboratory - ME/ECE/CHE 389

February 5, 2016

Contents

1 Ladder Logic

A programmable logic controller, or PLC, can be used to control any process that requires automation, such as processes found in a chemical plant, automated machinery, or laboratory equipment. When used in such a process, the PLC is programmed to perform a set of logical steps in a specific order to complete the desired task. These steps are programmed into the PLC using a ladder logic program, which consists of a specific set of instructions which will be used to define the process and control the machine.

Ladder logic is a graphical programming language that uses diagrams similar to electric circuits to enter a program into the PLC processor (MicroLogix 1000 in our case). The controller takes in inputs from the ladder logic, processes the input data based on existing conditions, and sends outputs to the machine or process. The ladder program entered is a specific set of instructions provided for the controller to follow during control of the process. A ladder diagram (Figure 1) graphically shows each of the specific steps that the controller must follow in the control circuit operation.

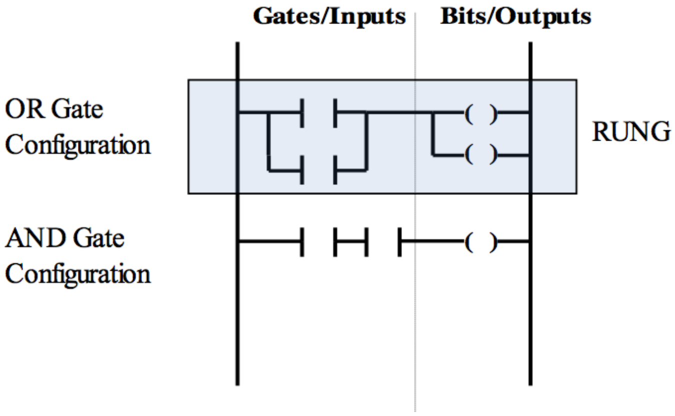


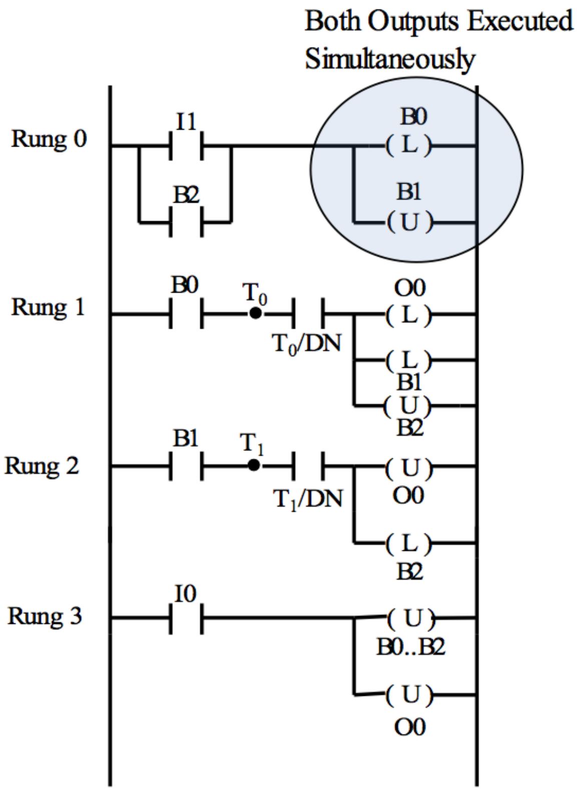
Figure 1: Ladder diagram.

The left side of the ladder represents a live wire through which information flows. Although the controller is able to scan the entire ladder logic program instantaneously, it must still proceed through the control process in the logical pattern. This is accomplished through the use of gates. The gate represents the input on the rung of the ladder diagram. The two main gates used in the laboratory are ”OR” and “AND” gates (Figure 1). The “OR” gate is similar to a parallel circuit, where if either one OR the other switches are closed then the output side of the rung will become “energized”. The “AND” gate is similar to a set of switches connected in series. If the input referring to the first switch AND the input referring to the second switch are both turned on, the gate is closed, and the output side of the diagram becomes energized. These types of gates can be arranged so that the outputs are energized to perform some logical series of operations.

When the gate conditions are satisfied, the output instructions, which are located at the end of the rung, are activated. Only two output instructions are necessary to carry out the labs in this course: initialization of bits and activation of external switches. The initialization of bits enables new rungs to be activated. The activation of external switches activates or deactivates physical operations; i.e. the signal sent from the controller causes a pneumatic valve to open which, in turn, extends a piston in a piston-cylinder device. In ladder logic, these output instructions are accomplished by loading and unloading bits and outputs, which subsequently energize a rung to load or unload an output. Each symbol entered on a rung in ladder logic is given an address which consists of a letter and a number. The letter represents various functions, while the number allows for distinction. In this laboratory, the letters used and their definitions are as follows:

I	→	Input	→	input to the controller from an external source
O	→	Output	→	output from the controller to an external source
T	→	Timer	→	sets time between operations in the program
B	→	Bit	→	used to activate desired rungs in program or store information
L	→	Load	→	loading of desired operation
U	→	Unload	→	unloading of desired operation

1.1 Example Ladder Logic Program



On the first rung (rung 0) of the ladder diagram, either Input 1 (I1) or Bit 2 (B2) must be closed to energize the rung. If the system is experiencing initial start-up, then it will require an external input, such as pressing a button or a switch, in order to begin the control process. If this is the case, then the manual loading of Input 1 will be responsible for energizing the rung and activating the outputs on the right half of the rung. In this case, the output loads bit B0 as designated by (L) and unloads bit B1 as designated by (U). The loading of bit B0 closes the gate on rung 1, which initiates timer T0. A timer waits for some predefined amount time to elapse and then loads a corresponding “Timer-Done-Bit”. In this case, timer T0 closes timer-done-bit T0/DN, which energizes the rung 1 and activates its outputs. Rung 1 loads output 0 to perform some physical operation, loads bit 1, and unloads bit 2. The program proceeds similarly to rung 2 and then loops back to rung 0. The program will continue to loop through rungs 0-2 until the external input I0 is activated, which unloads all the bits.

Rung 3 is included in the program as a means of turning off the machine and as a safety rung. It is not necessary for the actual system movements and controlled steps, but is necessary as a means to stop the system and return all actions to their initial state. Note, the system will continue to loop through rungs 0-2 until this safety rung is activated by input I0. After I0 is initialized, the bits that were loaded during the process are each unloaded so that each is returned to its initial state when the system is turned off. This way, the system can be halted at any stage during the process.

2 Setting up RSLinx and RSLogix

Two programs will be needed for writing and editing ladder logic files and downloading them into the PLC processor. The first is RSLinx which will be used to setup the communications between the PC and PLC processor. RSLinx must be setup properly and remain open while editing and running the ladder logic programs. The second is RSLogix which will be used for editing the ladder logic program and downloading it to the PLC processor.

2.1 Setting up RSLinx

1. Navigate to All Programs, Rockwell Software, RSLinx, and open RSLinx Classic
2. Check that the PC and PLC are communicating (**if not you need to configure drivers**)
   (a) Select Communications drop down

(b) Select RSWho

(c) Confirm that the PC and PLC are talking to each other, the Link is blinking

(d) NOTE: RSwho just brings up the drop down tree which may already be visible

3. Configure drivers (**see TA for help on this step**)

   (a) Select the Communications drop down menu

   (b) Select Configure Drivers

   (c) From Available Drivers Types drop down menu select RS-232 DF1 devices and click Add New

   (d) Click Configure on the right

   (e) Select the appropriate Comm Port (this is the particular USB port that is connected to the PLC processor)

   (f) Click Auto-Configure, click OK, click Close

4. **Do not close RSLinx:** It must remain open in order to download ladder logic programs into the PLC processor with RSLogix.

2.2 Using RSLogix: Editing Downloading and Running Programs

RSLogix is used for editing ladder logic programs and downloading them into the PLC processor so that they can be run on the physical hardware. The extension .RSS is used to designate ladder logic program files to be run with RSLogix. A sample .RSS file is available to help familiarize yourself with the programming functions of RSLogix.

2.2.1 Configuring RSLogix

1. Open RSLogix

   Navigate to All Programs, Rockwell Software, RSLogix Micro English, and open RSLogix Micro English

2. Open the sample .RSS file which can be found on the controls drive.

   ```
   T:\PLC_02_TrafficLight\Ladder_Files\RSLOGIX_TUTORIAL_ME389.RSS
   ```

3. Ensure you are properly connected to the PLC processor

   (a) On left side of RSLogix, the program tree should be visible

   (b) If program tree is not visible you can return to the default view by selecting Window drop down menu, Arrange, Default Project, OK

   (c) In the Program Tree, extend the controller folder, double click Controller Properties

   (d) Select Controller Communications, click Who Active

   (e) Select the MicroLogix 1000 processor

   (f) Click OK, click Apply (in Controller Properties), click OK

4. Now you should be able to download and run the sample file.

2.2.2 Downloading and Running Programs

There are two states to the RSLogix software, Online and Offline. While Offline, the ladder program can be edited and then downloaded into the PLC processor. In the Online mode, RSLogix can be used to observe the current state of each instruction. For example, if bit B0 is currently activated, it will be lit up in the RSLogix ladder diagram while in Online mode. However, editing the program can only be done in Offline mode.

There are also two states to the PLC processor, Program mode and Run mode. The processor state can be controlled from the PC with RSLogix. In Run mode, the processor will carry out the most recently downloaded program. In Program mode, the processor will sit idle, allowing you to make changes to your program (note: edits can only be done with RSLogix in Offline).

1. Download the sample program into the PLC processor (`RSLOGIX_TUTORIAL_ME389.RSS`).
   - (a) Select Comms drop down menu
   - (b) Select Download
   - (c) You may be prompted to switch RSLogix to Online mode, click yes if desired
   - (d) You may also be prompted to switch the Processor to Run Mode, do so if desired
2. Switch RSLogix to Online mode (if not already)
   - (a) Select Comms
   - (b) Select Go Online
   - (c) The ladder in the upper left of RSLogix will be spinning when Online.
3. Switch the Processor to Run mode (if not already)
   - (a) Select Comms
   - (b) Select Mode
   - (c) Select Run
   - (d) When in run mode the left ladder will be highlighted in green.
4. Switch to Offline mode to edit the program
   - (a) Select Comms, Mode, Program
   - (b) Select Comms, Go Offline
5. Print the program to PDF
   - (a) Select File drop down, select Print Report
   - (b) Under Name select Adobe PDF, click OK to print
   - (c) An explorer window should pop up to select where to save the PDF
6. **Note:** For the MicroLogix 1000, you can only edit the program in offline mode.
7. **Note:** You must re-download program to the PLC before with any new edits to the program.

8. **Debugging Tip:** If your program cannot be downloaded because of programming errors

   (a) Programming errors are listed in the results window

   (b) Select View, Check the Results window (if not already checked), the Results window is displayed at the bottom

   (c) Errors are listed in File 2

3 Ladder Logic with RSLogix Tutorial

A quick tutorial on using RSLogix to view and download the ladder logic file into the PLC processor. We consider the example ladder logic file in RSLogix of Figure 2. The program is initiated by pressing Input I1, which energizes rung 0 and loads Bit B0 and initiates Timer T0. Timer T0 counts up to 5 seconds, then loads Bit B3, energizing rung 1 and loading Output 11. Finally the last rung is energized by Input I0, which unloads everything.

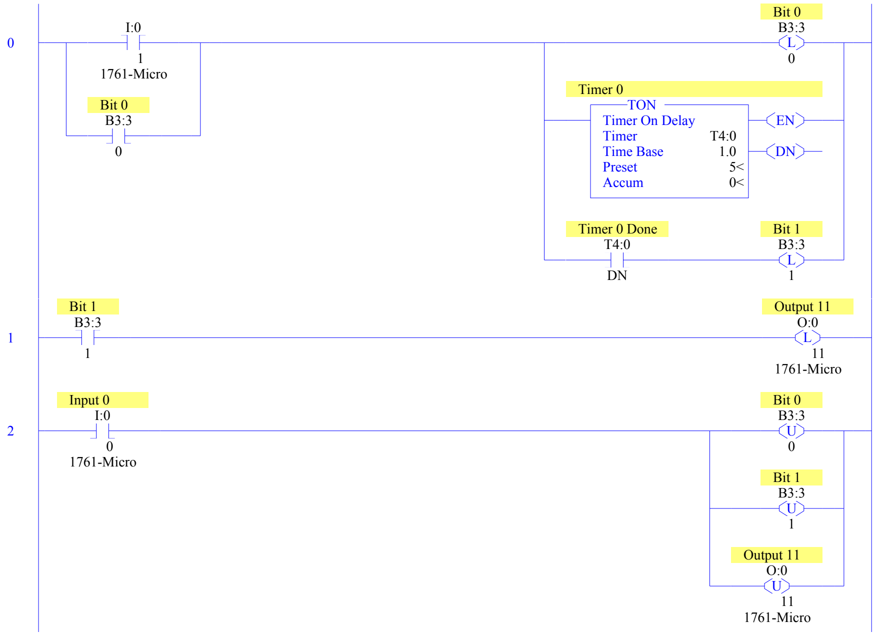


Figure 2: Example ladder logic program (RSLOGIX_TUTORIAL_ME389.RSS).

1. Open the practice file `RSLOGIX_TUTORIAL_ME389.RSS` with RSLogix

2. Download the file in the processor (Comms, Download)

3. Switch RSLogix to Online (Comms, Go Online) if not already Online

4. Change PLC processor to run mode (Comms, Mode, Run) it not already in Run mode

5. Start the file by pressing Input I1

6. You can observe the progress of the program from the RSLogix window

   (a) Notice activated instructions will be lit up in the RSLogix window.

(b) Notice the inputs and outputs can be named to make the code more readable.

(c) Notice how the respective LEDs on the PLC processor (MircroLogix 1000) light up when the corresponding inputs and outputs are loaded.

4 Basic Ladder Logic Instructions with RSLogix

This section briefly covers the necessary ladder logic programming commands needed to carry out the ME389 PLC laboratories using RSLogix. For more information on each instruction see the associated chapter in the MicroLogix Manual. The instructions needed for programming are located in the box displayed in Figure 3, most of instructions necessary for the lab are under the "User" tab and "Timer/Counter" tab.

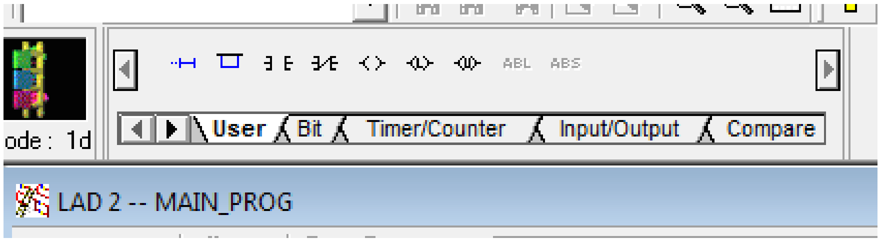


Figure 3: RSLogix Instruction toolbar.

4.1 Recommended Reading

1. Chapter 7: Detailed information entering instructions and functions i. Important bit instructions are Load, AND, OR (7-3)
   (a) Set and Reset (7-8)
   (b) Memory Push and Memory Pop (7-10)
   (c) Timer Instructions: Timer On-Delay (7-16)
   (d) Counter: Count Up (7-24) and Reset (7-27)
2. Chapter 16: Entering and editing your program (For Handheld programmer)
   (a) Particularly (16-1) through (16-4)

4.2 Bits Chapter 6-3

4.3 Inputs

For the MicroLogix 1000 inputs are cataloged in the form I:0/0 - Input 0

4.4 Outputs

For the MicroLogix 1000 outputs are cataloged in the form O:0/1 - Input 1

4.4.1 Output Latch (OTL) and Output Unlatch (OTU), Chapter 6-5

4.5 Examine Open

4.6 Examine Closed

4.7 Count Up (CTU) Chapter 6-18

4.8 Timer On-Delay (TON) Chapter 6-11

An example of a timer in RSLogix is shown in Figure 4. In this case, a Timer-On-Delay (TON) is loaded by bit B1 (B3:0/1). This particular timer, T1 (T4:1), waits for 2 s, then loads the timer done bit (T4:1/DN) which loads Bit 9 (B3:0/9). Don't worry about the specific meaning of each digit in B3:0/1, just understand that this refers to Bit 1, and likewise, T:4/1 refers to timer 1.

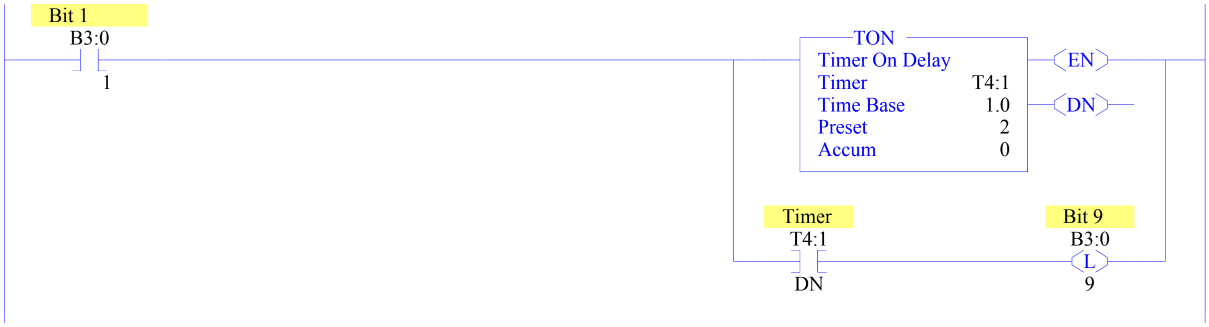


Figure 4: Example of a Timer On Delay (TON) in RSLogix.

4.9 Memory Push and Memory Pop (7-10)