Lab Week 13: State Tables to Ladder Logic

0. Objectives

- Turn your state table into ladder logic

1. Creating your PLC program

Using the State Table in Table 1, let's transform it to Ladder Logic

Table 1 – Example State Table

State	**Trigger**	**Pos_Z_Move**	**Clamp**	**Other**
1	Start	**X**		
2	+Z Limit	**X**	**X**	T1 (3s)
3	T1		**X**	
4	-Z Limit			Program Terminate (State 0)

General Process:

For each state create one *latching* ladder rung (see appendix) as follows:

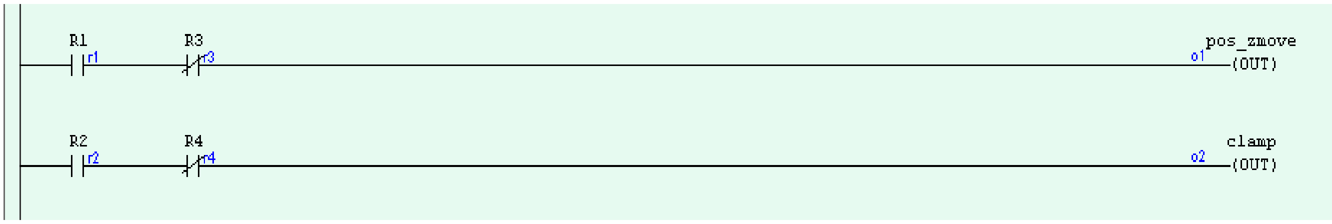
- 1st element: Trigger
- 2nd element: Termination (usually the previous output)
- 3rd element: Timer (if necessary)
- 4th element: Output (internal relay)

For each physical output create one ladder rung as follows:

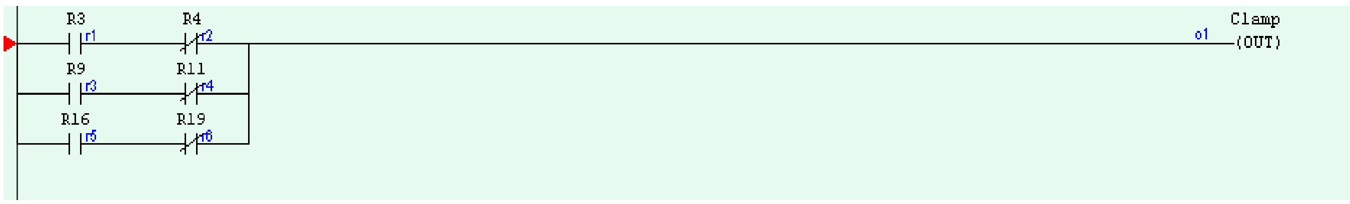
- Use Boolean logic and internal relays on left rail to relate to physical outputs on right rail

For Exercise 1, the only physical outputs are +Z actuation, –Z actuation, and clasp. +Z actuation turns on in State 1 and off in State 3. Clamp turns on in State 2 and off in State 4.

The ladder logic for the physical outputs would be written as:



If clamp also turned on in state 9, off in state 11, and then on in state 16 and off again in state 19, its ladder rung would look like the following:



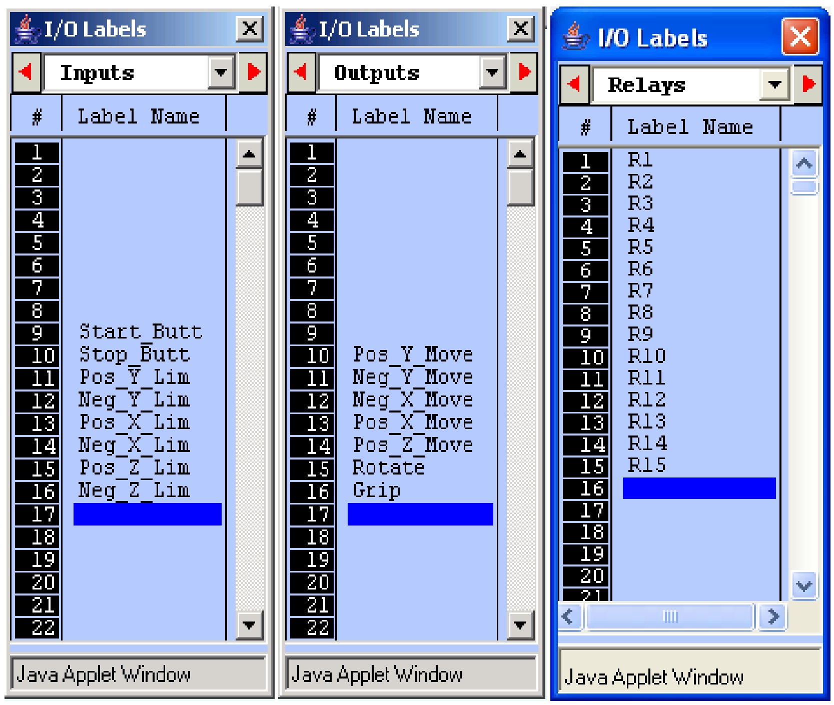
Do this for all states and all outputs and combine all ladder rungs together.

2. Create Ladder Logic from Your State Table

2.1 Fill in input/output table (Inputs, Outputs, Relays, and Timers)

a. Timers are in a tenth of a second
b. Create a Relay for each state that you have (You may have more or less than 15)

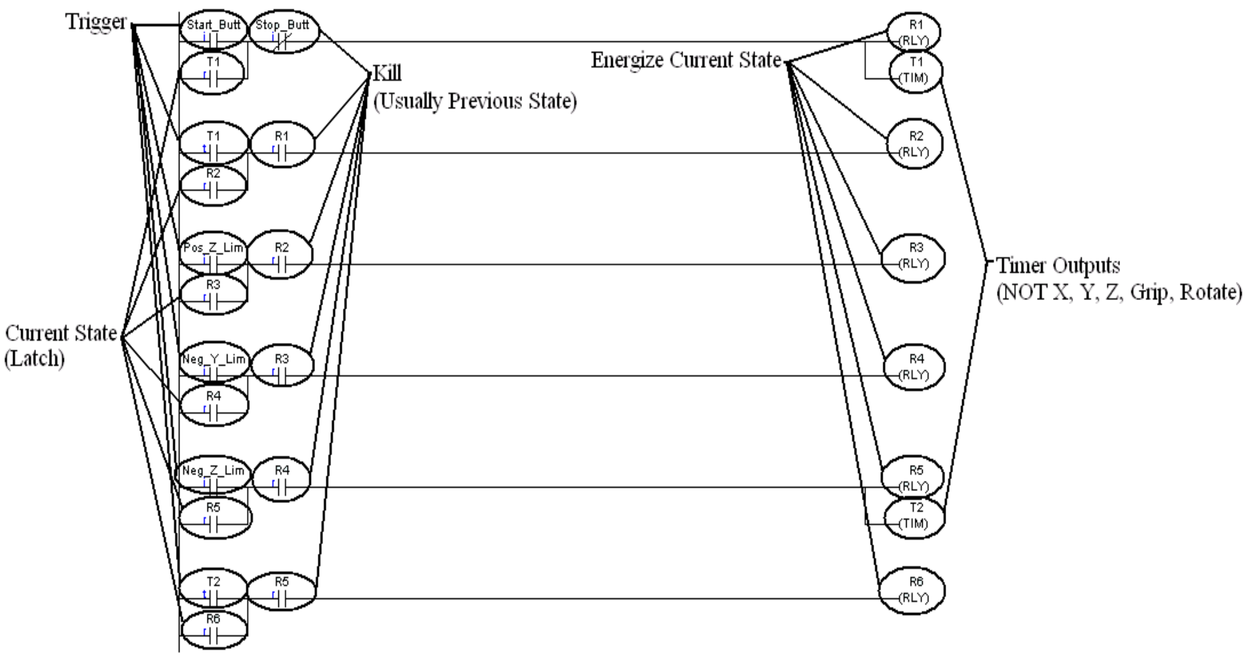
Note: the input and output tables correspond to the wiring on the PLC. Make sure your input and output tables look like the images below.



Note: The number of relays that you have is dependent on your state table. You may have more or you may have less than 15 relays.

2.2 Create a ladder rung for each state

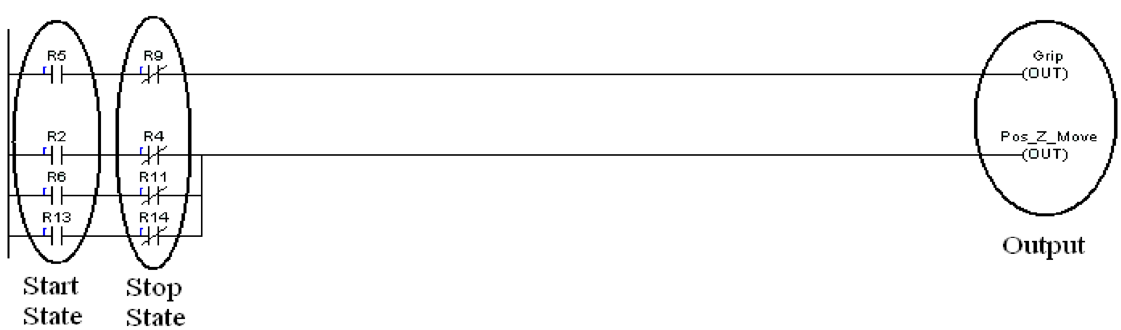
Don’t worry about the outputs yet.



2.3 Create ladder logic (Outputs)

Now that all the state ladder logic is done, now relate the states to the outputs using Boolean logic. For example, the first circuit below turns the grip on in state 5 and kills it in state 9. Or in Boolean logic: R5 and not R9 = Grip. The next circuit determines the positive z movement. R2 +z move, R4-off, R6 +z move, R11-off, R13 +z move, R14 – off.

NOTE: Each output can only be used once.

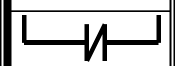
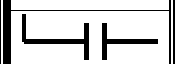
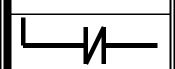
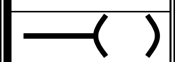
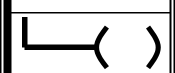
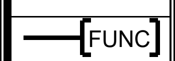
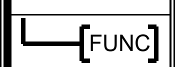


3. Deliverables (due 11/26 or 11/28, depending on your lab section)

Two completed PLC programs based on your state tables

1. Start button sequence
2. Stop button sequence

4. Appendix

Basic Ladder Logic Components	
Symbol	*Function*
	Normally open contact connected in series
	Normally closed contact connected in series
	Normally open contact connected in parallel
	Normally closed contact connected in parallel
	Normally open contact connected in parallel followed by another element
	Normally closed contact connected in parallel followed by another element
	Connect coil (output) to right power rail
	Connect coil (output) to right power rail in parallel
	Run the defined user program when inputs are true
	Run the defined user program in parallel when inputs are true