Hardware Engineer:	Basil Alsulami	Lab:	3
Software Engineer:	Andrew Struthers	Due:	January 24th, 2019

Lab 3: Boolean Logic Implementation

Purpose:

In the first two labs, the majority of the hardware configuration was completed. With the exception of occasional alterations, the final hardware configuration will be completed in this lab. This will focus exclusively on interfacing the five indicators on the front panel to the PLC. Because of the limited number of outputs available, only one of the indicators will be directly interfaced to the PLC while the others are interfaced indirectly using the four general purpose relays.

Upon completion of the hardware component, the lab will require each group to implement a series of logic circuits in the PLC. The first three logic circuits will be expressed as an Algebraic expression, which will require the groups to interpret the expressions as separate rungs in the ladder and create truth tables for the expressions to verify functionality. The two subsequent logic circuits will be expressed as truth tables and will require the students to write an algebraic expression from each truth table, then implement the circuits in the ladder diagram.

Objectives:

Upon completion of this lab, you should be able to:

- Demonstrate momentary contact and relay interfacing techniques
- Implement control panel wiring solutions using NFPA 79
- Establish network connectivity using RSLinx Classic
- Interpret ladder logic diagrams
- Implement Boolean Logic statements in Ladder Logic.

Materials:

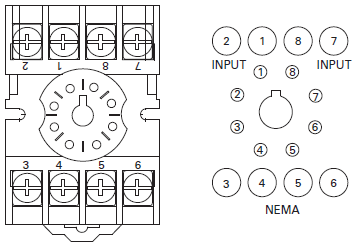
1 *x*	PLC Trainer	1 *x*	Digital Multi-Meter Fluke 179 or equivalent
2 ft	Blue wire		
5 ft	Orange wire		

Procedure:

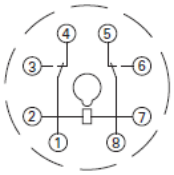
Final Hardware Configuration

The Hardware Engineer should perform following portion of the procedure.

1. Before beginning any lab, always inspect the equipment to verify that all connections, including the chassis ground wires are secure and properly connected and that the equipment is safe to operate.



2. Use the diagram to the right to connect one of the two available poles on each of the four general purpose relays to a 24V DC distribution point. Because the relays are low current devices, the power may be daisy-chained.



3. Interface the green indicator to the leftmost general purpose relay by routing a wire from the corresponding NO contact on the relay base to the terminal block associated with the green indicator.

4. Interface the yellow, blue, and white indicators the NO contact on each of the subsequent relays.

5. Interface the red indicator directly to a terminal block associated with one of the unassigned outputs from the output module.

6. Record each indicator's physical address in the table below. The physical address corresponds to the connection point on the output module of the PLC and should be recorded in the following format:

   Local:<slot#>:<I(nput)/O(utput)>.Data.<bit#>

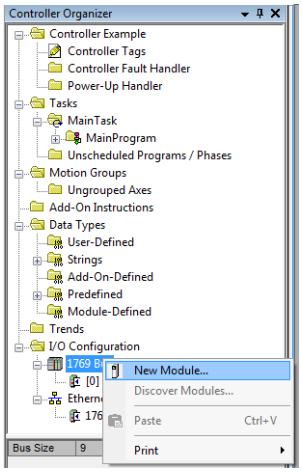
Indicator	Physical Location
Red	Local:2:O.Data.<5>
Yellow	Local:2:O.Data.<2>
Green	Local:2:O.Data.<1>
Blue	Local:2:O.Data.<3>
White	Local:2:O.Data.<4>

7. Secure all wires and install all covers. Apply power to the unit and verify that the input module is detecting all inputs by actuating each input and observing the status of the point on the input module. Demonstrate the proper operation of the circuit to the Instructor or a TA to receive a sign-off.

Software configuration

The Software Engineer should perform following portion of the procedure.

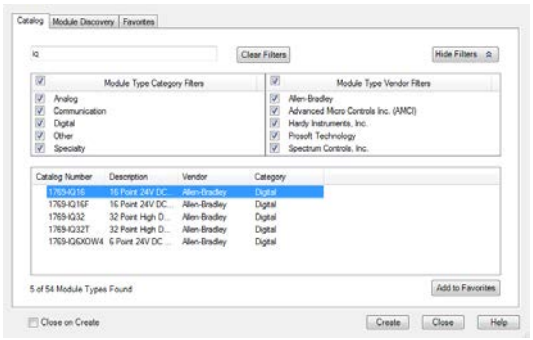
8. If necessary, use **Network Connectivity and Configuration** procedure from Lab 2 to configure the network connection for your PLC from the PC that you plan to use for the remainder of this lab.



9. Launch **Studio 5000** and create a new Project named "Lab3_Unit#" where # corresponds to your Unit number. Make sure to select the the "1769-L30ER" CompactLogix™ 5370 Controller before proceeding. **Save the project to your N: drive.**

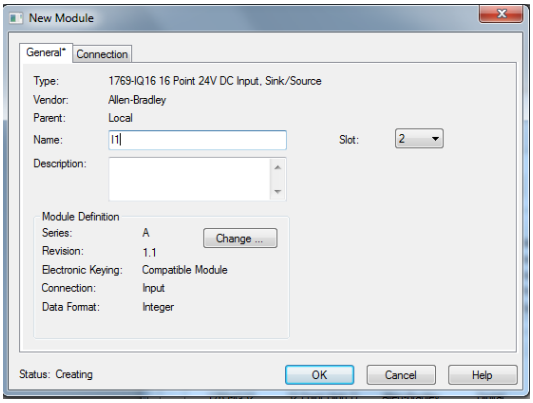
10. Under the Controller Organizer, expand the I/O Configuration and right click on the "1769 Bus".

11. Select "New Module…".



12. Locate the input module from the list. The input module model can be located on the inside cover of the "DC Input" module.

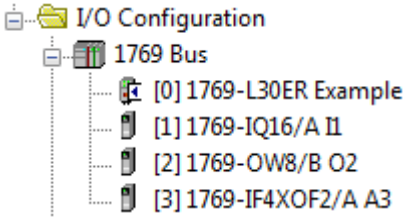
13. Click "Create".



14. Provide the input module the name "I1", referring to the module as an input located in slot 1, and click "OK".

15. The module will show up under the 1769 Bus in the I/O Configuration.

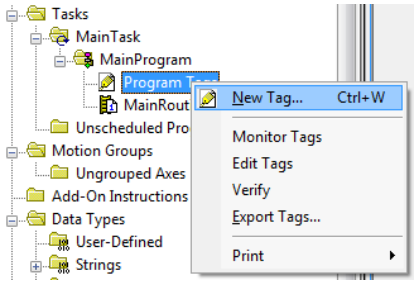
16. Repeat this process for the output module and analog module, labeling these modules "O2" and "A3" Respectively. Once complete, the following devices should be attached to the 1769 Bus.



Program Tag Creation

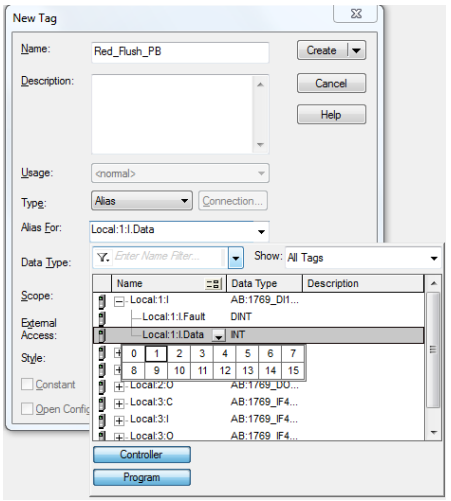
Program Tags serve as variable names that can be easily identified in the program. The program tag can be used instead of the physical address to write programs in the PLC.

17. Verify that the controller recognizes each module by checking that the Controller Tags have been created under Controller tab.



18. Under the MainProgram task, open the Program Tags explorer by double clicking it.

19. Rigth click on the Program Tags label to create a new tag.
    a. Name the Tag "Red_Flush_PB"
    b. Select "Alias" from the Type drop down menu.
    c. Select the bit that the Red Flush Pushbutton is connected to on the input module Alias from the drop down menu.
    d. Click "Create". The program tag should appear in the Program Tag explorer.



20. Repeat the previous step for each pushbutton interfaced to the input module using the following tag names. Record the associated physical address of each pushbutton **in the format specified in step 6.**

Push Button Tag Name	Input Module Terminal
Red_Flush_PB	Local:1:I.Data.<4>
Yellow_Flush_PB	Local:1:I.Data.<3>
Green_Flush_PB	Local:1:I.Data.<2>
Blue_Flush_PB	Local:1:I.Data.<1>
White_Flush_PB	Local:1:I.Data.<0>
Black_Mushroomhead_PB	Local:1:I.Data.<5>
Red_Mushroomhead_PB	Local:1:I.Data.<6>

21. Create the five indicator tags using the same procedure using the following tag names.
    - If you are completing this assignment electronically, the Physical Location will copy over from Step 6. Otherwise, fill in the associated physical address of each tag from step 6:

Indicator Tag Name	Physical Location
Red_Indicator	Local:2:O.Data.<5>
Yellow_Indicator	Local:2:O.Data.<2>
Green_Indicator	Local:2:O.Data.<1>
Blue_Indicator	Local:2:O.Data.<3>
White_Indicator	Local:2:O.Data.<4>

Logic Circuit Implementation

Implement each logic circuit using Studio 5000. A spreadsheet application (Excel) may be used to generate the truth tables on the first three problems. Simplification of the logic circuits is not required. It is advisable to test functionality of each circuit individually before advancing to subsequent circuits.

22. Construct a truth table for the expression:

$$Green\_Indicator = Green\_PB \cdot (Red\_PB + Black\_Mushroomhead\_PB + Yellow\_PB)$$

Implement this expression on the first rung of your ladder logic. Include the truth table with the lab report.

23. Construct a truth table for the expression:

$$Yellow\_Indicator = Yellow\_PB + Red\_PB \cdot White\_PB \cdot \overline{Green\_PB}$$

Implement this expression on the second rung of your program. Include the truth table with the lab report.

24. Construct a truth table for the expression:

$$Red\_Indicator = Red\_PB \cdot \left(Yellow\_PB \cdot \left(\overline{Green\_PB} + White\_PB \cdot \left(\overline{Green\_PB} + Blue\_PB\right)\right)\right)$$

Implement this expression on the third rung of your program. Include the truth table with the lab report.

25. Write an algebraic expression for the truth table below. Implement the expression as the fourth rung of your program. Include the algebraic expression with the lab report.

Green_PB	Yellow_PB	Red_PB	White_Indicator
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

26. Write an algebraic expression for the truth table below. Implement the expression as the fifth rung of your program. Include the algebraic expression with the lab report.

Blue_PB	White_PB	Green_PB	Yellow_PB	Blue_Indicator
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

27. Download the program to the PLC and place the PLC in run mode. Verify that the outputs correspond with the truth tables constructed above.

28. Verify that each logic circuit corresponds to its corresponding truth table.

29. Include a PDF of the program in the lab report.

Note:

Students enrolled in ETSC 522 must implement each of the logic circuits using Structured Text and Ladder Diagrams.