Programmable Logic Controllers

Basic Ladder Logic Programming

Outline

• Boolean statements and ladder logic equivalents
  - Logical AND
  - Logical OR
  - Logical NOT

• Commonly used ladder logic sequences
  - Start-stop-seal circuits
  - Basic interlocks

• Properly formatted outputs
Boolean logic control programs

- Boolean logic control programs examine and control on and off states
  - Boolean here is used interchangeably with the word “discrete”

- Each control program (ladder diagram sequence) can contain one or more conditionals

- Example
  - If (a part is on the conveyor) AND (there is not a box in the chute) THEN (turn the conveyor motor on)

- In terms of sensors and actuators this becomes
  - If (sensor_A is ON) AND (sensor_B is NOT ON) THEN (turn actuator_C ON)

Conveyor motor control system
Logical AND ladder diagram

- The logical AND function is constructed by series combinations of digital (discrete) inputs
  - Two (or more) series components

Logical OR ladder diagram

- The logical OR function is constructed by parallel combinations of digital (discrete) inputs
  - Two (or more) parallel components
**Logical NOT**

- The logical NOT function is constructed by referencing the input signal with a normally closed contact (XIO instruction).

**Complex Boolean expressions**

- More complex Boolean expressions can be formulated with various serial-parallel combinations of XIC and XIO instructions.
  - NAND, NOR, XOR, XNOR
Start-stop-seal circuits

• For PLC systems without latch and unlatch instructions, a circuit is needed that will allow a process to start, continue to run after a start button is released, and stop under control of another button
  – A circuit that implements this functionality is commonly referred to as a **start-stop-seal** circuit

• A feedback path (i.e. a contact) that references the output is normally used to seal around the start contact

---

Start-stop-seal ladder diagram

Initial state  
START pushbutton pressed  
START pushbutton released  
STOP pushbutton pressed
Start-stop-seal variations

- In practice, several start and/or several stop buttons can be used in a process.
- Start buttons (with XIC instructions) can be used:
  - In series if it is required that all be pressed before a process starts.
  - In parallel if pressing any start button is to start a process.
- Stop buttons (with XIO instructions) are normally used in series if pressing any stop button is to stop a process.

Start-stop-seal circuit example
Interlock circuits

- Interlocks can prohibit output(s) from energizing under a certain condition
- Example: O:2/0 should not energize if O:2/1 is energized (and vice versa)

Formatting considerations

- Ladder logic rungs should be formatted so the reader can easily infer the meaning of the intended logic
- One mechanism to help this is the grouping of related signals within an area on a given rung of logic
- For example:
  - Group signals together that have some common intent
    - Start signals
    - Stop signals
    - Emergency stop signals (E-stop)
    - Interlocks
  - Controls that might have greater importance (i.e., E-stop) might be located on the left hand side of the rung if possible
### Formatting considerations

<table>
<thead>
<tr>
<th>E-stop conditions</th>
<th>Normal Stop</th>
<th>Normal Start</th>
<th>Interlocks (if any)</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTOR 1 Control Rung: Specifies all start and stop conditions. An interlock with MOTOR 2 is provided.</td>
<td>MOTOR 2 Control Rung: Specifies all start and stop conditions. An interlock with MOTOR 1 is provided.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is also a good example of instruction and rung documentation.

### Properly formatted outputs

- An output energize instruction (OTE) referencing a specific output bit should appear only once in a ladder logic program.
Properly formatted outputs

- Only one output energize instruction (OTE) should appear in a rung of ladder logic.

- If more than one output is to be controlled by a certain rung of ladder logic, the output energize (OTE) instructions can be placed in parallel.