ECET 4530

Industrial Motor Control

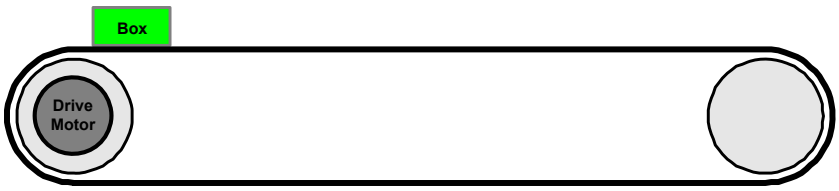
RSLogix – Example Problem

1

RS Logix Problem Statement

Given a simple PLC-controlled conveyor that will be used to transport a box back-and-forth between the conveyor's ends;

Create a Ladder Logic Program for the PLC that will provide the required operational logic for the system based on the specifications stated in the following slides.



2

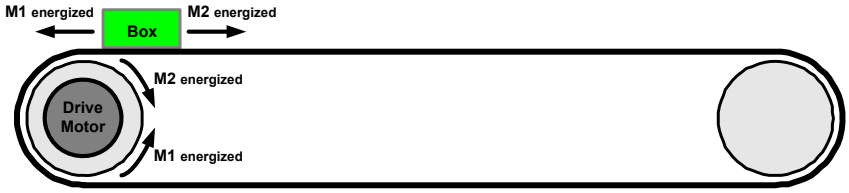
Example Problem - System Description

The conveyor will utilize a single drive motors, the direction of which is determined by a pair of contactors:

- **Energizing contactor <u>M1</u> moves the Box to the <u>left</u>, and**
- **Energizing contactor <u>M2</u> moves the Box to the <u>right</u>.**

Notes – Contactors M1 and M2 cannot be energized simultaneously.

– The PLC's relay outputs will be used to energize the field-coils of the contactors.

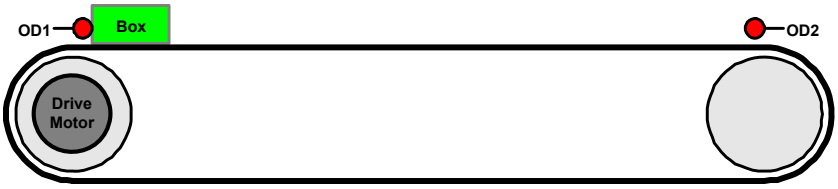


3

Example Problem - System Description

An Optical Beam Detector will be placed at each end of the conveyor in order to detect when the box reaches either end of the conveyor.

Note – when the box breaks one of the beams provided by the optical detectors, a normally-open contact in that beam's detector module will close and provide +24V_{DC} to one input of the PLC.



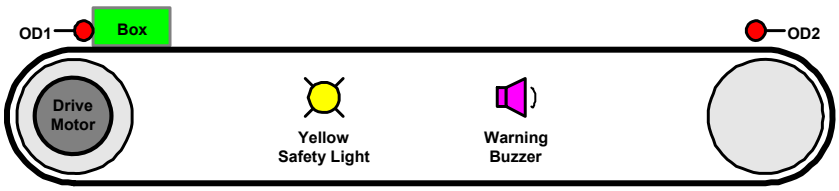
4

Example Problem - System Description

Additionally, a yellow Safety Light and a Warning Buzzer are included with the system.

The safety light must be illuminated whenever the system is operational.

The buzzer will only sound during startup and shutdown.

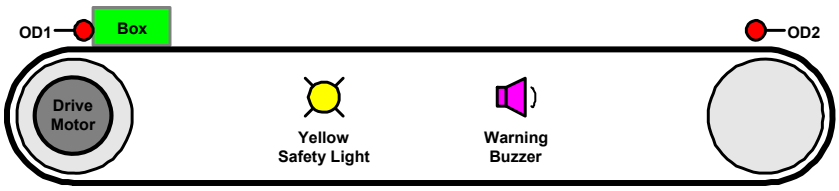


5

Example Problem - System Description

Three pushbuttons will be used to control the system's operation:

- **A Normally-Open START button**
- **A Normally-Open GO button**
- **A Normally-Closed STOP button**

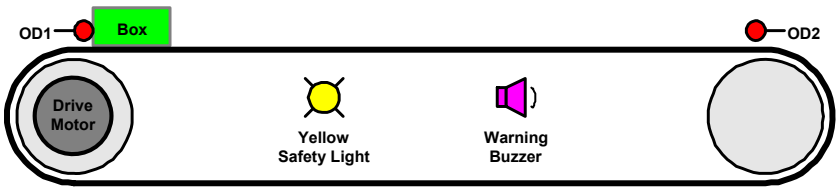


6

Control System – I/O Schedule

The Input Schedule for the PLC is as follows:

Input 0 – START (NO)
Input 1 – GO (NO)
Input 2 – STOP (NC)
Input 3 – OD1 (NO)
Input 4 – OD2 (NO)

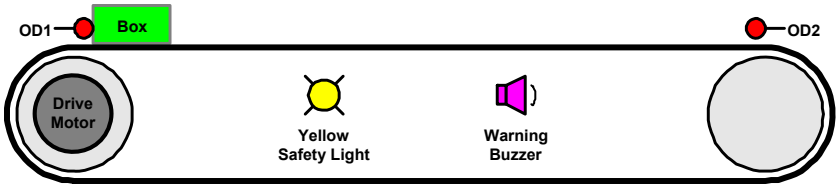


7

Control System – I/O Schedule

The Output Schedule for the PLC is as follows:

Output 0 – Yellow Safety Light
Output 1 – Warning Buzzer (Horn)
Output 2 – M1 (Field Coil – Contactor M1)
Output 3 – M2 (Field Coil – Contactor M2)



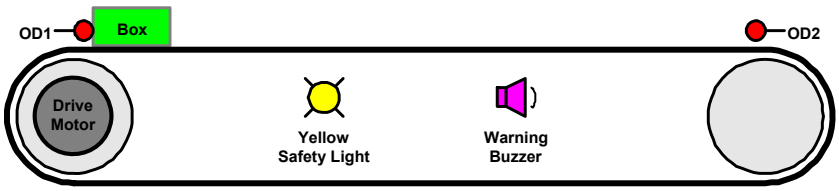
8

Example Problem – Operational Logic

System Operation – Startup

When START is pressed

- **The Yellow Safety Light *Illuminates.***
- **The Warning Buzzer *sounds* for 5 Seconds.**
- **When the 5-second Warning Buzzer is complete, the GO button is enabled.**



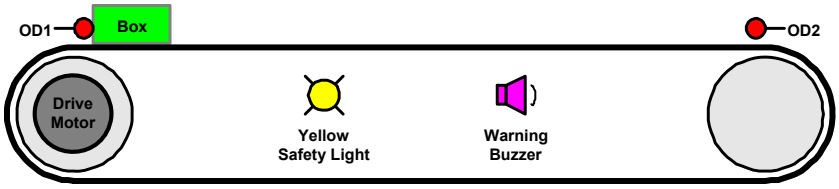
9

Example Problem – Operational Logic

System Operation – Normal Operation

When GO is pressed

- **If the GO button has been enabled, the conveyor will begin continuously moving the box back-and-forth between the optical detectors until the STOP button is pressed.**



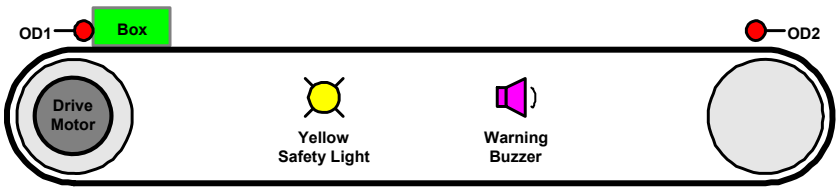
10

Example Problem – Operational Logic

System Operation – Shutdown

Pressing STOP

- **The system operates until the box reaches OD1, at which point the conveyor stops.**
- **When the conveyor stops, the Warning Buzzer sounds for 5 Seconds.**
- **When the 5-second buzzer is complete, the Yellow Safety Light is extinguished.**

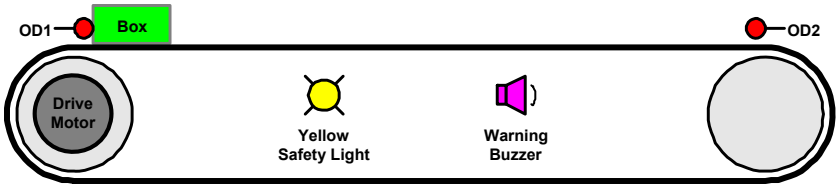


11

Ladder Logic Program– Status Bits

The following Status Bits will be utilized in the PLC's program:

- **Startup – Startup procedure active**
- **Ready – Awaiting Go button**
- **Run – Normal operation**
- **Shutdown – Shutdown procedure active**

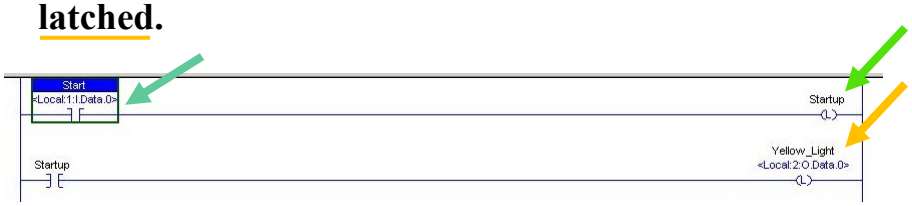


12

Ladder Logic Program – Startup

System Startup

When START (In-0) is pressed, the Startup bit will be latched, in-turn causing the Yellow Light's output (Out-0) to be latched.



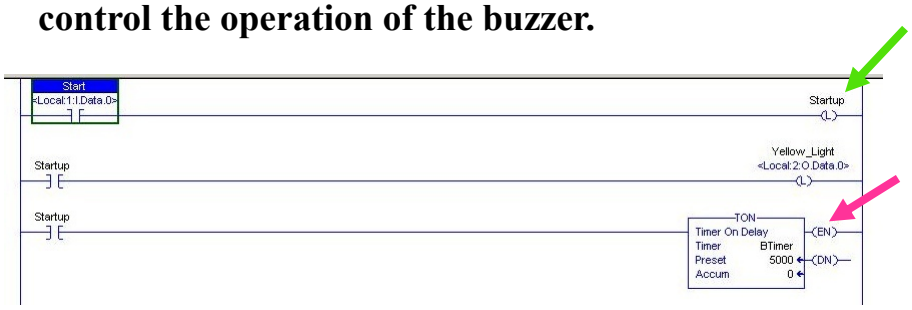
Part_1

13

Ladder Logic Program – Startup

System Startup

Additionally, when the Startup bit is latched, the 5-second timer (BTimer) will be enabled. This timer will be used to control the operation of the buzzer.



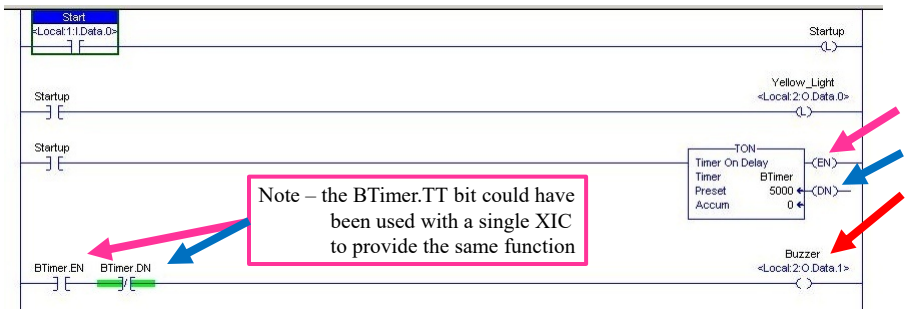
Part_1

14

Ladder Logic Program – Startup

System Startup

When BTimer is enabled but not done (i.e. – it is transitioning), the Warning Buzzer's output (Out-1) will be energized.



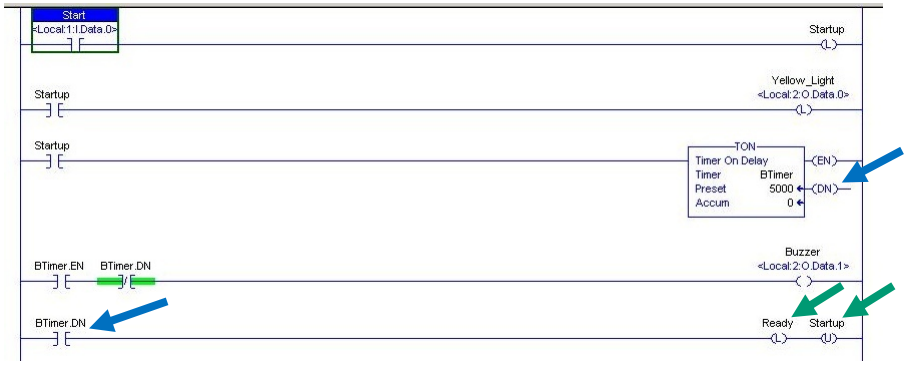
Part_1

15

Ladder Logic Program – Startup

System Startup ➡ Ready Mode

When BTimer is done (i.e. – its accumulator reaches 5000), the Ready bit will be latched and the Startup bit will be unlatched… at this point, system startup is complete.



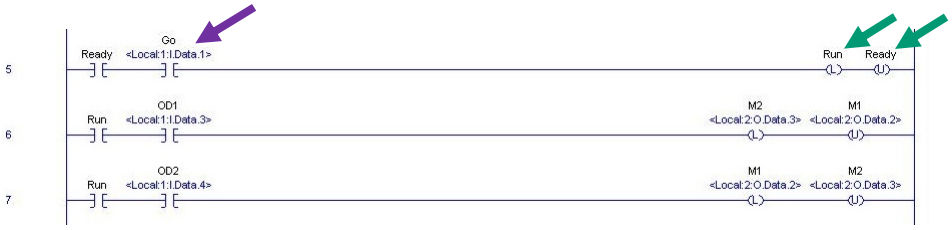
Part_1

16

Ladder Logic Program – Ready Mode

Ready Mode ➡ Run Mode

The system will remain in Ready mode until the GO button is pressed, at which time the Run bit will be latched and the Ready bit will be unlatched.



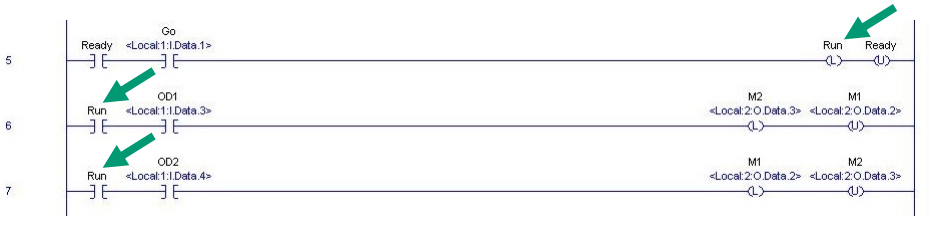
Part_1

17

Ladder Logic Program – Run Mode

Run Mode

When the Run bit is set (1), the conveyor will begin moving, the direction of which is based on the position of the box on the conveyor.



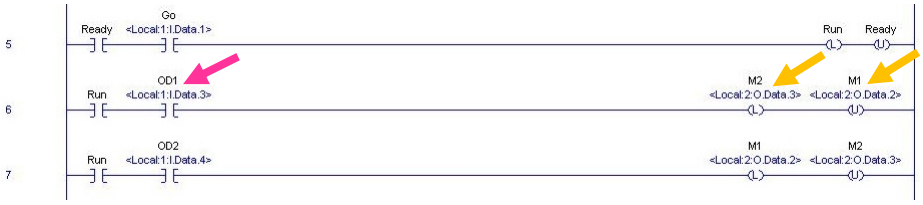
Part_1

18

Ladder Logic Program – Run Mode

Run Mode (Left→Right Motion)

When the box breaks OD1's beam, the output (Out-3) for contactor M2's field coil is latched and the output (Out-2) for contactor M1's field coil is unlatched.



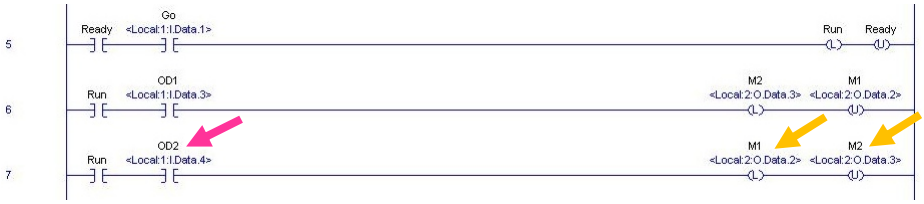
Part_1

19

Ladder Logic Program – Run Mode

Run Mode (Right→Left Motion)

When the box breaks OD2's beam, the output (Out-2) for contactor M1's field coil is latched and the output (Out-3) for contactor M2's field coil is unlatched.



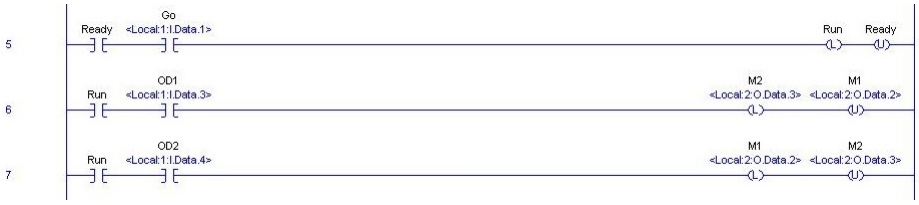
Part_1

20

Ladder Logic Program – Run Mode

Run Mode

In this manner, the box will be moved back-and-forth continuously between the optical detector beams until the STOP button is pressed.



Part_1

21

Ladder Logic Program – Shutdown

Run Mode ➡ System Shutdown

When the STOP button is pressed, the Shutdown bit will be latched.

Note that the Run bit is not unlatched at this time.

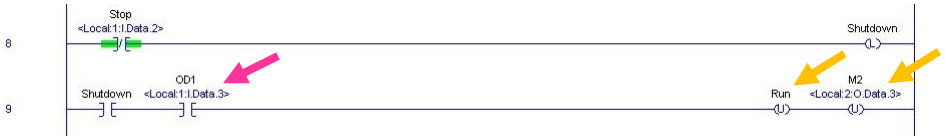
Part_1

22

Ladder Logic Program – Shutdown

System Shutdown

After the **Shutdown** bit is latched, the conveyors will keep running until the box breaks OD1's beam, at which time both the **Run** bit and M2 (Out-3) are unlatched.



Part_1

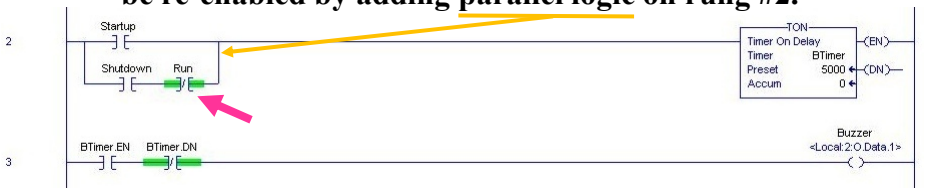
23

Ladder Logic Program – Shutdown

System Shutdown

After **Run** is unlatched, BTimer is enabled again, and the buzzer sounds for 5 seconds.

Note – Instead of creating a 2nd buzzer timer, the original timer can be re-enabled by adding parallel logic on rung #2.



Part_1

24

Ladder Logic Program – Shutdown

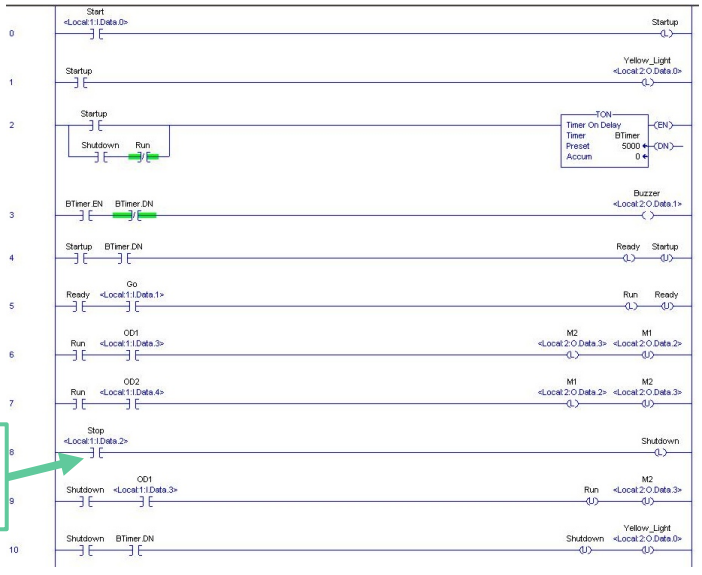
System Shutdown

When BTimer is done the 2nd time (during Shutdown), both the Shutdown bit and Yellow Safety Light (Out-0) are unlatched, at which point system shutdown is complete.

Part_1

25

Ladder Logic Program



Typo – should be an XIO since the Stop button is NC
(It is correct in the previous slides)

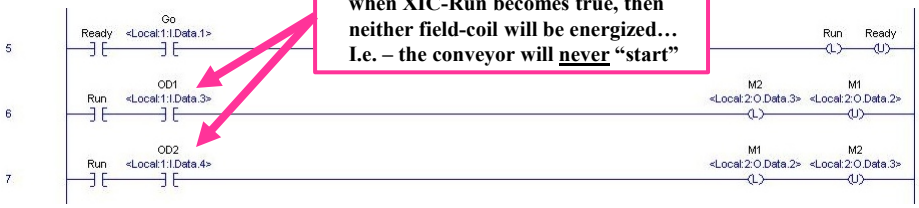
Conveyor_Demo_Part_1.ACD

26

Possible Startup Issue?

What happens if GO is pressed but the box is initially positioned such that it's <u>not</u> breaking the beam from OD1 or OD2?

If XIC-OD1 and XIC-OD2 are <u>both false</u> when XIC-Run becomes true, then neither field-coil will be energized… I.e. – the conveyor will <u>never</u> "start"



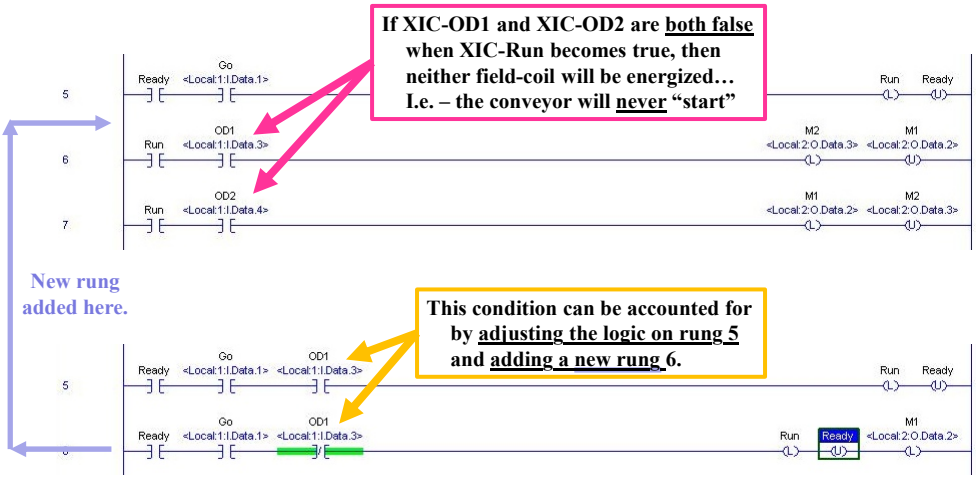
Part_2

27

Possible Startup Issue?

What happens if GO is pressed but the box is initially positioned such that it's <u>not</u> breaking the beam from OD1 or OD2?

If XIC-OD1 and XIC-OD2 are <u>both false</u> when XIC-Run becomes true, then neither field-coil will be energized… I.e. – the conveyor will <u>never</u> "start"



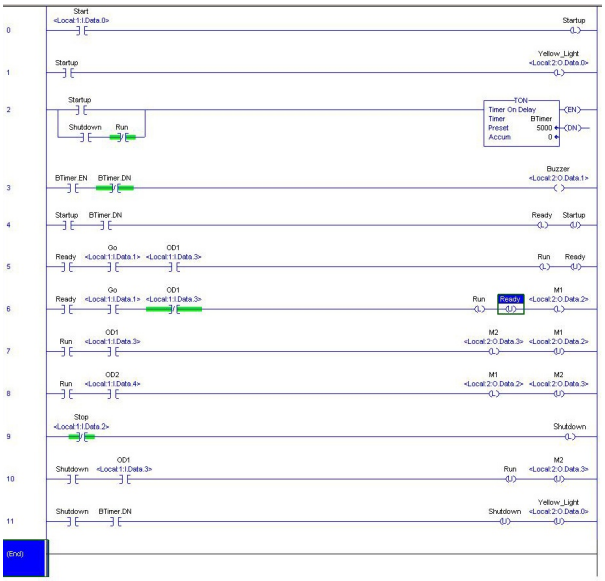
New rung added here.

This condition can be accounted for by <u>adjusting the logic on rung 5</u> and <u>adding a new rung 6</u>.

Part_2

28

Ladder Logic Program



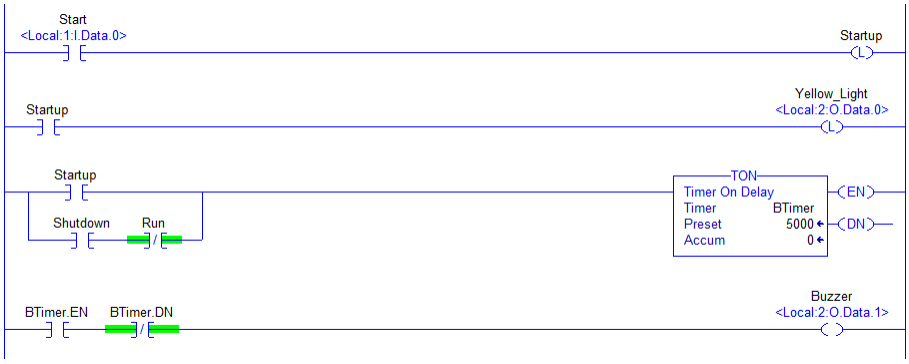
Conveyor_Demo_Part_2.ACD

29

Other Issues to Consider?

IMPROPER BUTTON PRESSES:

What happens if START is pressed after "Startup" is complete and the system is already operating in either the "Ready", "Run", or "Shutdown" modes?



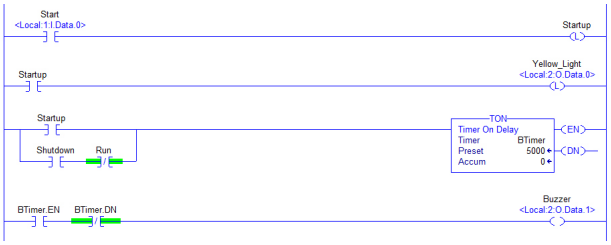
Part_3

30

Other Issues to Consider?

IMPROPER BUTTON PRESSES:

If START is pressed after "Startup" is complete and the system is already operating in any other mode, "Startup" will be re-latched causing the buzzer to sound again for 5 seconds.



This can be prevented by adding additional logic to rung "0":

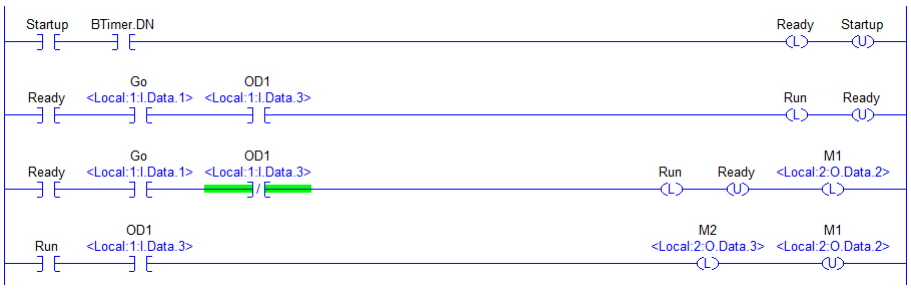
Part_3

31

Other Issues to Consider?

IMPROPER BUTTON PRESSES:

What happens if GO is pressed before "Ready" is complete (i.e. – when the system is shutdown or in "Startup" mode) or while the system is operating in either the "Run" or "Shutdown" modes?



Part_3

32

OTEs vs. OTLs & OTUs

Many experienced PLC programmers strongly recommend that OTL/OTU instructions be used sparingly, instead being replaced by OTE instructions whenever possible.

The primary reason for this is the manner in which the pre-scan of the ladder-logic program affects the instructions when the PLC is initially switched into RUN mode.

Although there are some advantages to this philosophy, beginning programmers often find it difficult to successfully implement programs using only OTE instructions.

Part_3

33

Initial Scan (Pre-Scan) of Ladder

When the PLC is switched from Program ⇨ Run mode, the PLC does pre-scan (initial scan) of the ladder diagram and, during this scan, considers the Rung Conditions FALSE for every rung regardless of the states of the logic instructions.

Since most output instructions are trigged or enabled by a TRUE rung condition, such as OTLs and OTUs, the initial execution of those instructions causes nothing to happen.

But an OTE triggers off both TRUE and FALSE rung conditions:

- **if an OTE is located on a rung, then the bit associated with that OTE's tag will be reset to zero during the pre-scan.**

Part_3

34

Previously Latched or Set Bits

An important aspect of the pre-scan is:

the only bits that are automatically reset to "0" during the pre-scan are those bits assigned to an OTE instruction.

If a bit assigned to OTLs and OTUs was previously set to a "1" and the PLC is switched back into RUN mode, then:

that bit does not automatically reset to "0".

This can be potentially dangerous, especially if the previously-latched bit causes an output to energize, in-turn causing a machine to start moving unexpectedly.

Part_3

35

OTEs vs. OTLs & OTUs

Although this aspect of the pre-scan provides an advantage to the use of only OTEs (and not OTLs and OTUs), beginning programmers often find it difficult to implement programs successfully using only OTE instructions.

To accommodate for the fact that bits assigned to OTL/OTUs are not automatically reset to "0" during the pre-scan, efforts must be taken to actively initialize these bits at the beginning of a ladder-logic program's execution.

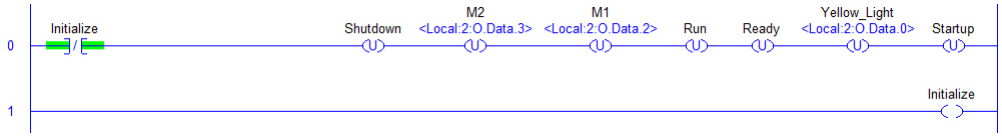
Part_3

36

Initializing Bits

Initializing bits can be tricky due to the nature of the pre-scan.

One method is as follows:



- **Add a rung "0" with an XIO that will unlatch (or latch if needed) any bits that require initialization whenever the program is first executed.**
- **Add a rung "1" with (no logic instructions) only a single OTE that has the same tag as the XIO on rung "0".**

When the program is executed, the OTE's bit will be reset during the pre-scan, causing the rung condition of rung "0" to be TRUE during the next scan, immediately after which the OTE will become TRUE, resulting in a rung "0" FALSE rung condition for the remainder of the execution.

Part_3

37