



# *ECET 4530*

## *Industrial Motor Control*

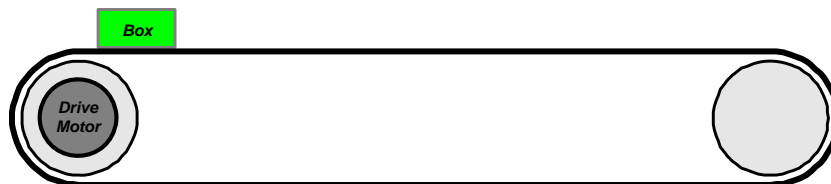
### *RSLogix – Example Problem*



## **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.**



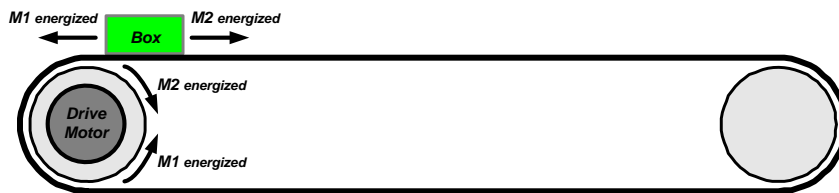


## Example Problem - System Description

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

- Energizing contactor M1 moves the Box to the left, and
- Energizing contactor M2 moves the Box to the right.

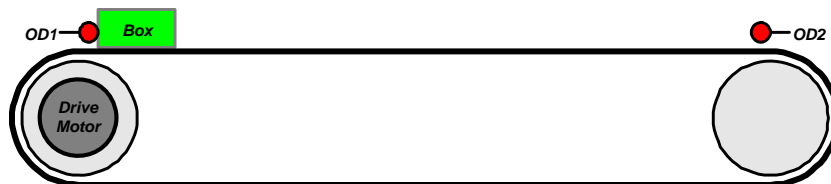
Notes – Contactors M1 and M2 cannot be energized simultaneously.  
– The PLC's relay outputs will be used to energize the field-coils of the contactors.



## 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.



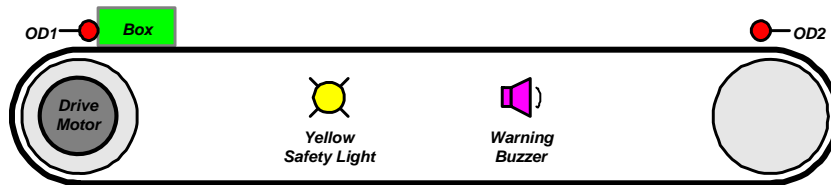


## Example Problem - System Description

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.

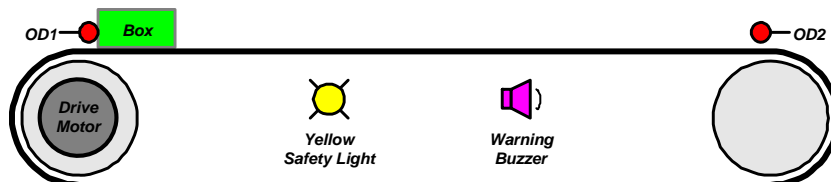
The buzzer will only sound during startup and shutdown.



## 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





## Control System – I/O Schedule

The Input Schedule for the PLC is as follows:

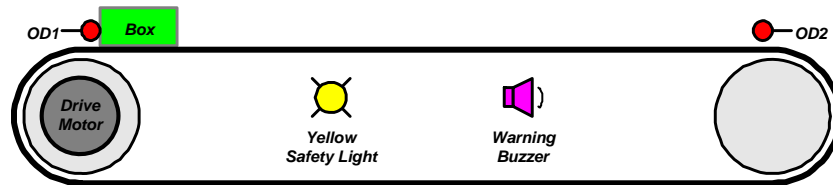
**Input 0 – START (NO)**

**Input 1 – GO (NO)**

**Input 2 – STOP (NC)**

**Input 3 – OD1 (NO)**

**Input 4 – OD2 (NO)**



## Control System – I/O Schedule

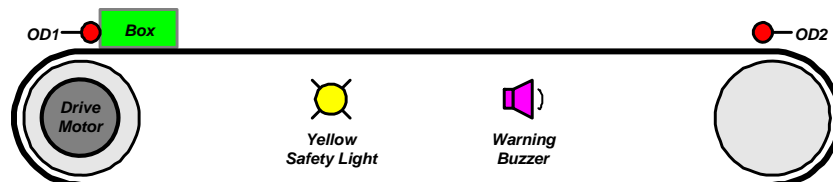
The Output Schedule for the PLC is as follows:

**Output 0 – Yellow Safety Light**

**Output 1 – Warning Buzzer**

**Output 2 – M1 (Field Coil – Contactor M1)**

**Output 3 – M2 (Field Coil – Contactor M2)**

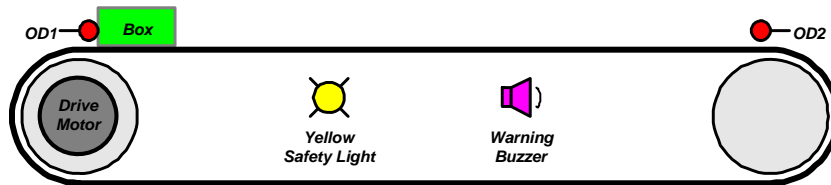




## Example Problem – Operational Logic

### System Operation – Startup

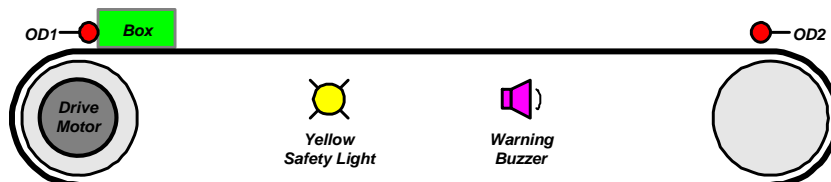
- When START is pressed
- The Yellow Safety Light *Illuminates*.
  - The Warning Buzzer *sounds* for 5 Seconds.
  - When the 5-second Warning Buzzer is complete, the GO button is enabled.



## 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.

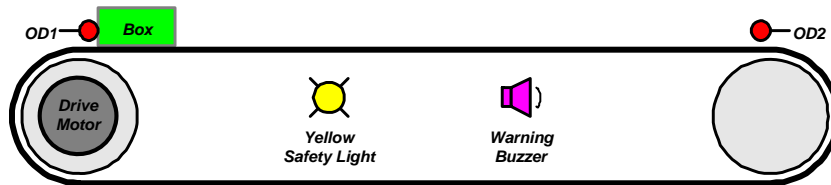




## Example Problem – Operational Logic

### System Operation – Shutdown

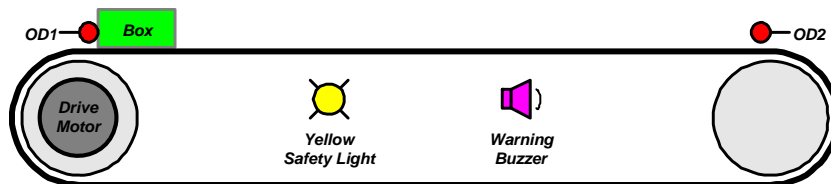
- Pressing **STOP**
- The system operates until the box reaches OD1, at which point the conveyor stops.
  - When the conveyor stops, the Warning Buzzer sounds for 5 Seconds.
  - When the 5-second buzzer is complete, the Yellow Safety Light is extinguished.



## 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

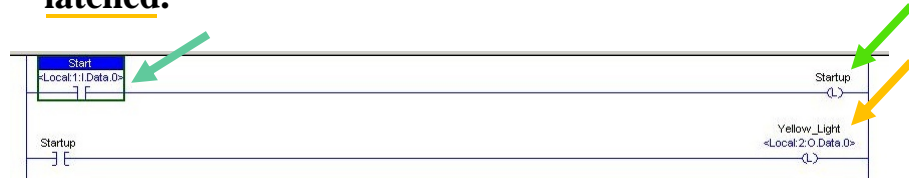




# Ladder Logic Program – Startup

## System Startup

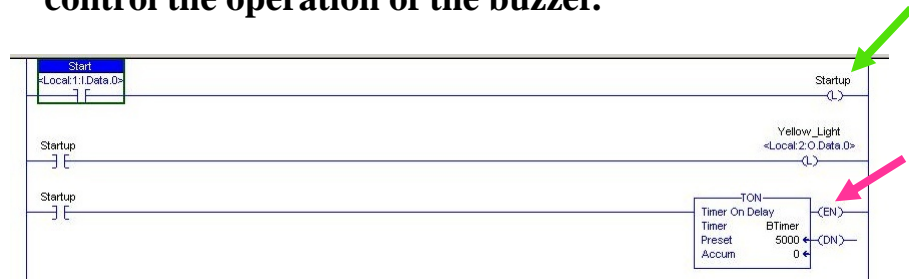
When **START** (In-0) is pressed, the **Startup** bit will be latched, in-turn causing the **Yellow Light's** output (Out-0) to be latched.



# 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.

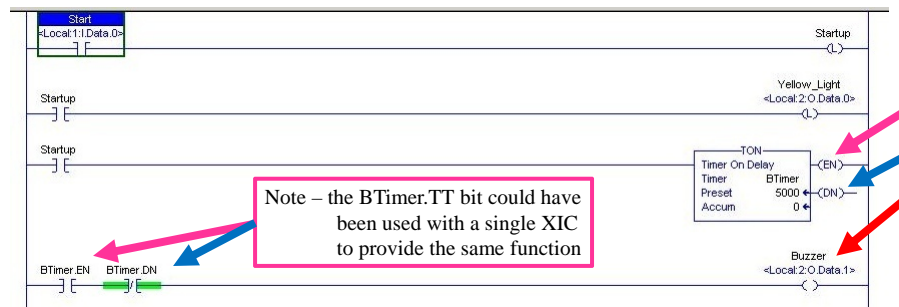




# Ladder Logic Program – Startup

## System Startup

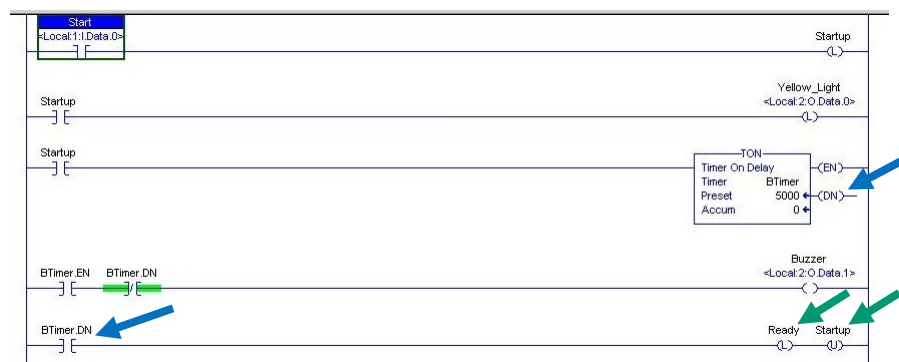
When **BTimer** is enabled but not done (i.e. – it is transitioning), the Warning Buzzer's output (Out-1) will be energized.



# 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.



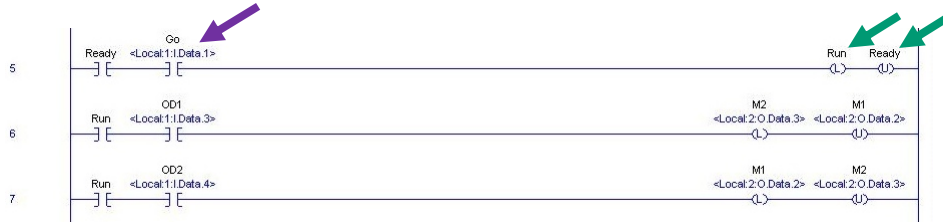




# Ladder Logic Program – Ready Mode

## Ready Mode → Run Mode

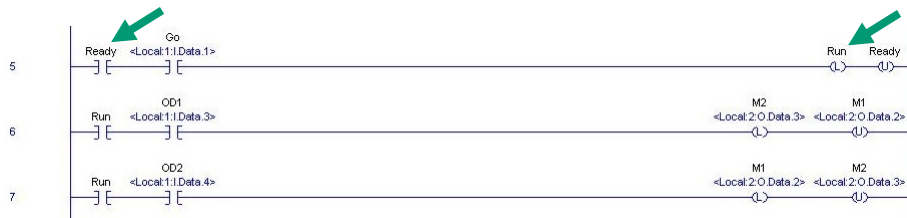
The system will remain in Ready mode until the **GO** button is pressed, at which time the **Run** bit will be latched and the **Ready** bit will be unlatched.



# 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.

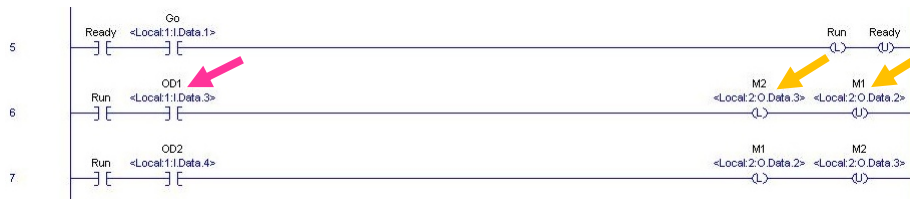




# Ladder Logic Program – Run Mode

## Run Mode (Left→Right Motion)

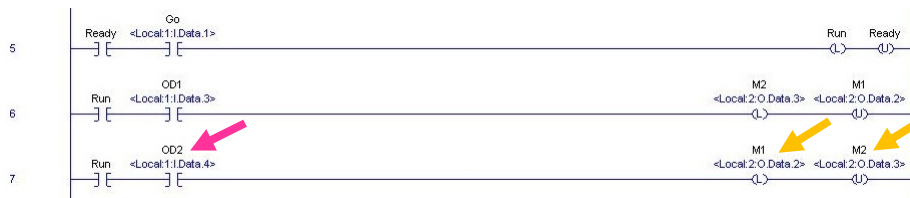
When the box breaks OD1's beam, the output (Out-3) for contactor M2's field coil is latched and the output (Out-2) for contactor M1's field coil is unlatched.



# 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.

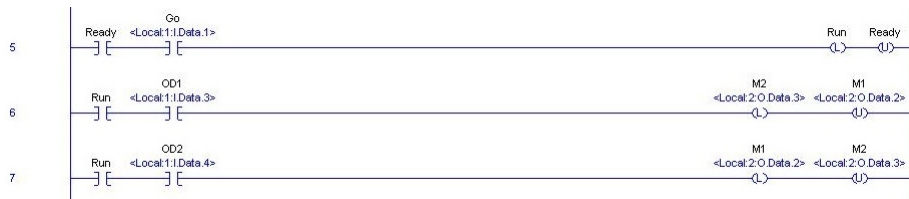




# Ladder Logic Program – Run Mode

## 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.



# 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.





# 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.

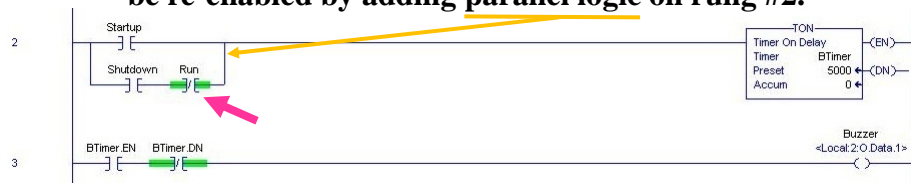


# Ladder Logic Program – Shutdown

## System Shutdown

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

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

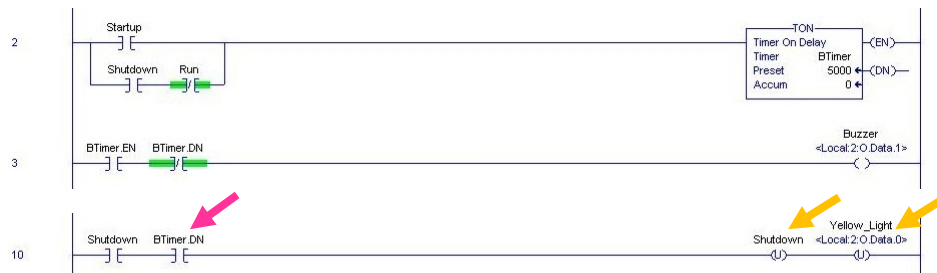




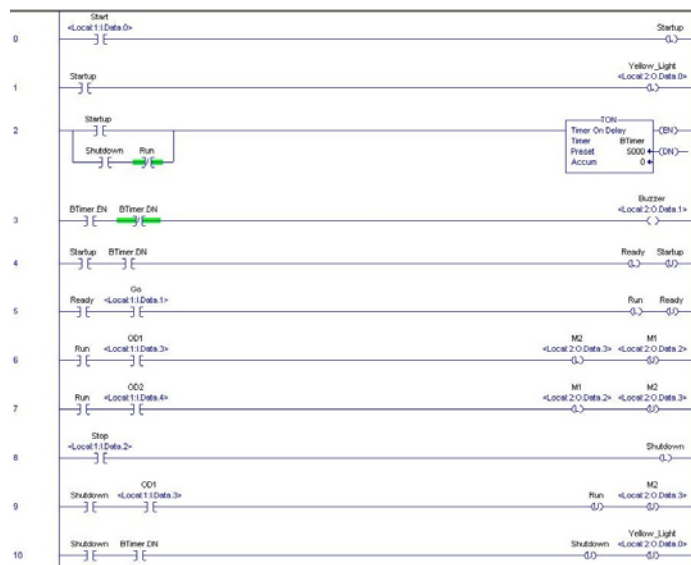
# Ladder Logic Program – Shutdown

## System Shutdown

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



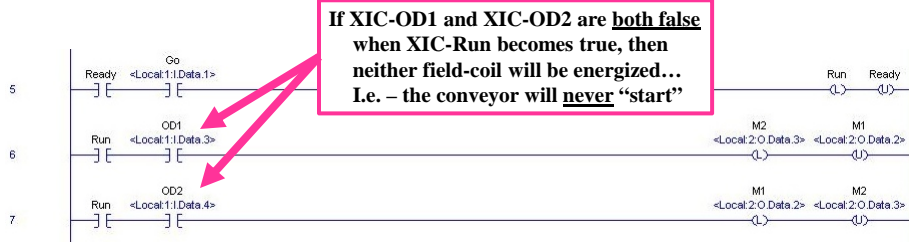
# Ladder Logic Program





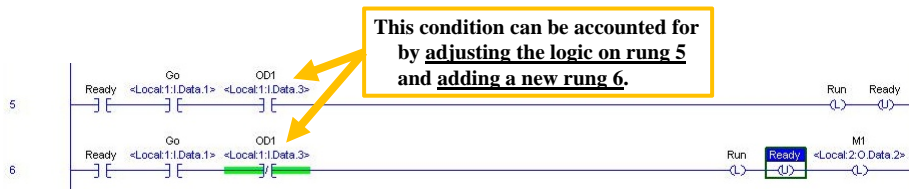
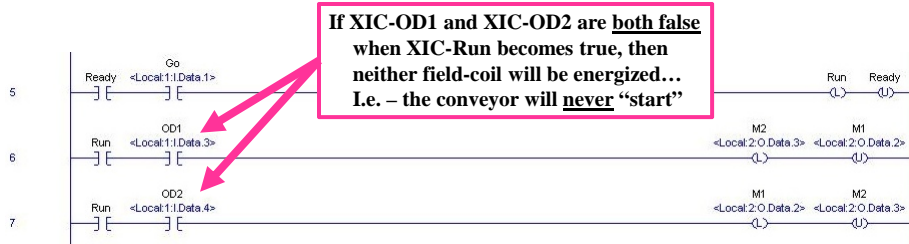
# Possible Startup Issue?

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



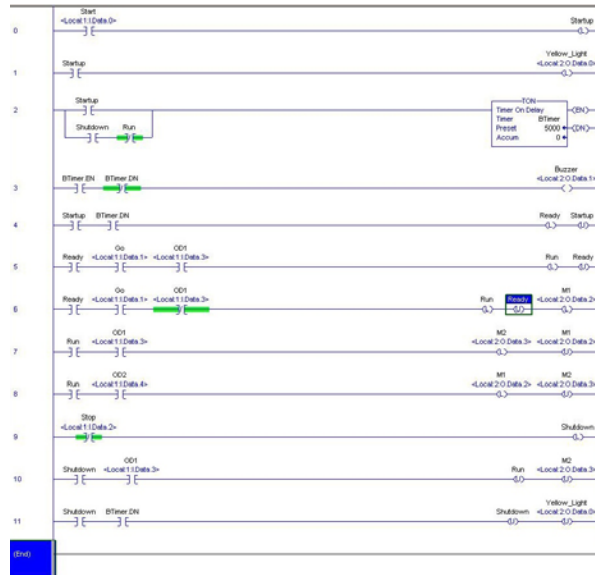
# Possible Startup Issue?

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





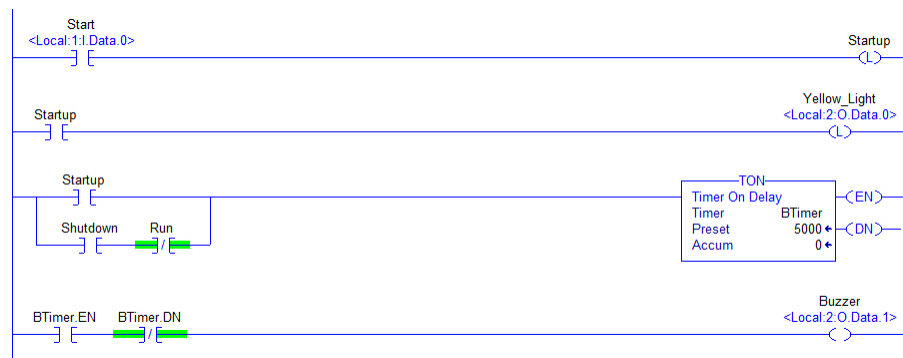
# Ladder Logic Program



## 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?

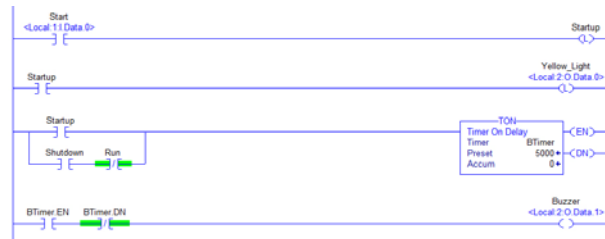




## Other Issues to Consider?

### 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.



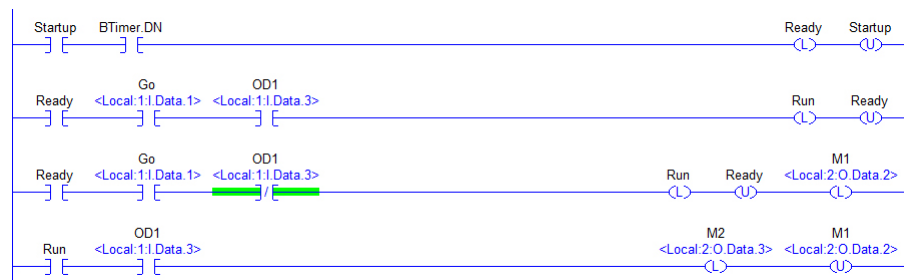
This can be prevented by adding additional logic to rung “0”:



## 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?







## Initial Scan (Pre-Scan) of Ladder

When the PLC is switched from Program  $\Rightarrow$  Run mode, the PLC does pre-scan of the ladder diagram.

During the pre-scan, PLC “looks” at the output instructions on each rung, but the PLC does not execute them.

If an OTE is located on a rung, the bit associated with that OTE’s tag is reset to zero.

But, if an OTL/OTU is located on a rung, the PLC does not change the value of the bit associated with that instruction.



## Previously Latched or Set Bits

Thus, if a bit was previously latched or set to a “1” before the PLC is switched out of Run mode, then the bit does not automatically become unlatched (reset to “0”) unless either:

- that bit is associated with an OTE instruction, or
- careful effort is taken to make sure the bit is initialized at the beginning of the program’s execution.

Failure to do so could cause a system to malfunction and can be potentially dangerous, especially if the previously-latched bit causes an output to energize, in-turn causing a machine to start moving unexpectedly.



## 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.

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

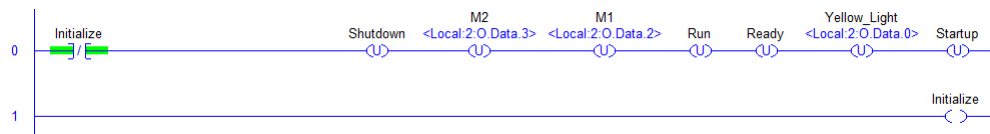
For this reason, any bits controlled by OTL/OTU instructions should be initialized whenever a program is executed.



## Initializing Bits

Initializing bit 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.