

# Advanced Safety Techniques for Logix Controllers



*For Classroom Use Only!*

LISTEN.  
THINK.  
SOLVE.®

# Important User Information

This documentation, whether, illustrative, printed, "online" or electronic (hereinafter "Documentation") is intended for use only as a learning aid when using Rockwell Automation approved demonstration hardware, software and firmware. The Documentation should only be used as a learning tool by qualified professionals.

The variety of uses for the hardware, software and firmware (hereinafter "Products") described in this Documentation, mandates that those responsible for the application and use of those Products must satisfy themselves that all necessary steps have been taken to ensure that each application and actual use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards in addition to any applicable technical documents.

In no event will Rockwell Automation, Inc., or any of its affiliate or subsidiary companies (hereinafter "Rockwell Automation") be responsible or liable for any indirect or consequential damages resulting from the use or application of the Products described in this Documentation. Rockwell Automation does not assume responsibility or liability for damages of any kind based on the alleged use of, or reliance on, this Documentation.

No patent liability is assumed by Rockwell Automation with respect to use of information, circuits, equipment, or software described in the Documentation.

Except as specifically agreed in writing as part of a maintenance or support contract, equipment users are responsible for:

- properly using, calibrating, operating, monitoring and maintaining all Products consistent with all Rockwell Automation or third-party provided instructions, warnings, recommendations and documentation;
- ensuring that only properly trained personnel use, operate and maintain the Products at all times;
- staying informed of all Product updates and alerts and implementing all updates and fixes; and
- all other factors affecting the Products that are outside of the direct control of Rockwell Automation.

Reproduction of the contents of the Documentation, in whole or in part, without written permission of Rockwell Automation is prohibited.

Throughout this manual we use the following notes to make you aware of safety considerations:

---

**WARNING**

Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

---

---

**IMPORTANT**

Identifies information that is critical for successful application and understanding of the product.

---

---

**ATTENTION**

Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:

- identify a hazard
  - avoid a hazard
  - recognize the consequence
- 

---

**SHOCK HAZARD**

Labels may be located on or inside the drive to alert people that dangerous voltage may be present.

---

---

**BURN HAZARD**

Labels may be located on or inside the drive to alert people that surfaces may be dangerous temperatures.

---

---

# Lab 13 – Advanced Safety Techniques for Logix Controllers

---

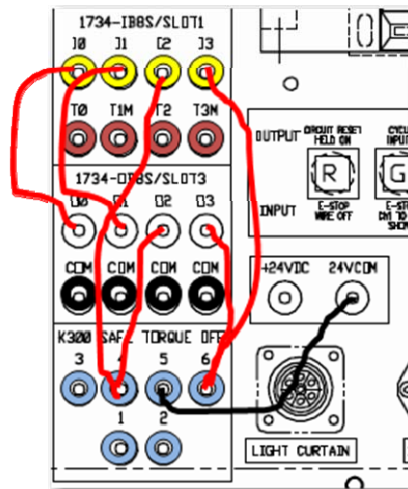
## Contents

- Before you begin** ..... 4
- About this lab ..... 5
- Tools & prerequisites ..... 5
- MSR57 functionality** ..... 6
  - Safe Limited Speed with the MSR57 ..... 6
  - Safe Maximum Speed with the MSR57 ..... 8
  - Safe Torque OFF with standstill checking ..... 9
- Certified Safety Instructions** ..... 10
  - Safety Input Instructions ..... 12
  - Safety Output Instructions..... 22
- Falling Edge Manual Reset**..... 25
- Safety AOs** ..... 27
  - Generate the Signature ID ..... 33
  - Generate the Safety Signature ID ..... 36
- CIP Safety** ..... 38
  - Connection Reaction Time Limit (CRTL) ..... 39

---

## Before you begin

- Download **AF2011 Lab13 GuardLogix Starting Point.ACD** to the CompactGuardLogix controller. This file is located on the desktop.
- The three (3) red/green pushbutton pilot light combo units on the lower right of the demo case are maintained buttons. Verify they are not in the maintained position.
- K300 Drive Power should be in the ON position.
- MSR-57 Safe Limited Speed should be in the RUN position.
- Both Mushroom head buttons should be pulled out to the active state.
- The analog input 1 dial should be between 1 and 9 on the dial. Below 1 or above 9 will cause a maximum speed fault.
- The yellow encoder cable should be plugged into the top RJ45 connector (Encoder 1) on the MSR57P.
- There should be five (5) connections made to the ethernet switch. They should be connected to:
  - Ethernet card in CompactLogix chassis
  - Point I/O adapter (top port)
  - Kinetix 300 drive
  - PanelView Plus 1000
  - PC
- The jumper cables should be attached as shown below



## About this lab

This lab is not a traditional Automation Fair lab, in which the student does the lab in a library like setting, and hopefully understands why there are performing all the steps. Although there is nothing to stop you from simply blasting thru this lab on your own, the instructor is going to break the lab into short consumable components. Each will be introduced, not only from a technical standpoint, but from a commercial point of view when appropriate. Each demo will be done by all students at once, and questions and concerns will be heard by the whole group. The actual demonstrations used in this lab, in some cases, are just simple demos of safety features that have already been completed for you. This allows us time to talk about the benefits of these features for you and your customers.

Taking a look at the demo case, you see that most of the products being used in this lab are safety products. This makes sense since this is a safety lab. The one product that is 'standard' is the Kinetix 300 drive. The Kinetix 300 drive does have redundant Safe Torque OFF inputs, and these are the safety outputs that will be controlled throughout this lab. The general theme of this lab will be to start the motor, and then stop it safely. Starting the motor should be easy. Pull out the Emergency Stop button(s) and then press or cycle any buttons that are flashing. Go ahead and try it.

## Tools & prerequisites

This is not an RSLogix 5000 lab. This lab assumes you are able to download projects, create tags, and perform basic editing. Please advise your instructor if this is not the case.

- The starting point ACD is located on your desktop. You may want to download it now, so that you are starting the lab with the appropriate project. It is called **AF2011 Lab13 GuardLogix Starting Point.ACD**.
- The Compact Machine Safety Solutions (CMSS) demo case is the only required hardware.

---

## MSR57 functionality

### Safe Limited Speed with the MSR57

Functional safety can enable operators or maintenance personnel to request safe access to a machine by simply pressing a button. But some machine processes make it undesirable to completely stop the machine. For example, web breaks due to restarting or materials that harden quickly during stoppages. In these instances, it is more practical to bring the machine to a reduced, yet safe speed, when a request for operator access is made. When the machine slows to the safe speed, the gate solenoid is unlocked, allowing safe access to the machine. The personnel enter the machine, perform routine and repetitive maintenance, leave the machine, close the gate, and manually resume normal speed.

When Safe Limited Speed is requested, a configurable timed delay of 3 seconds begins, and when it is reached, the speed must be under the safe limited speed or a fault is declared. The Safe Limited speed has been configured to 200 RPM (revolutions per minute). This is 3.3 RPS (revolutions per second). Both the meter on the demo and the HMI display RPS. There is an implied decimal point on the meter, so a value of 33 equals 3.3 RPS. After the delay, the gate will unlock because safe limited speed has been reached.

Safe Limited Speed Parameters within the MSR57 Safe Speed Relay

1: 0.52	Lim Speed Input	2NC	
1: 0.53	LimSpd Mon Delay	3.0	Sec
1: 0.54	Enable SW Input	Not Used	
* 1: 0.55	Safe Speed Limit	200.0	RPM
* 1: 0.56	Speed Hysteresis	0	%

1. If you have not already done so, download **AF2011 Lab13 GuardLogix Starting Point.ACD**. It is located on the desktop.

With the motor running and no faults

2. Move analog input dial between 4 and 6 so that the motor speed is below 33 on the meter and 3.3 on the HMI
3. Turn the SLS keyswitch to right (**LIM**ited) position to request SLS on the MSR57.  
SLS requested status appears on the HMI.
4. After the gate switch unlocks, pull out the gate key to simulate opening the gate door.  
SLS Mode status appears on the HMI. If a fault occurs because SLS was not reached, do steps 6-10 and then try again
5. Slowly increase the speed using the dial above 3.3 RPS, until the SLS is exceeded.  
The HMI indicates that the MSR57 has faulted, dropped out of SLS mode, and is now in Safe Stop mode.

6. Reinsert the gate key
7. Turn the SLS keyswitch back to the left (Run) position
8. Cycle the flashing red Safe OFF mushroom head button
9. Press the flashing green button
10. Press the flashing yellow button to start motion

## Safe Maximum Speed with the MSR57

Safe Maximum Speed (SMS) is typically used to protect against speeds that can damage machinery, the results of which could cause injury to personnel. When the MSR57 is configured for Safe Maximum Speed, the motor speed is monitored and compared against a user-configurable maximum. The safe max speed has been configured for 750 RPM which equals 12.5 RPS. If the speed exceeds the Safe Maximum Speed, the safety outputs are dropped out and the motor stops.

Safe Maximum Speed Parameters within the MSR57 Safe Speed Relay

1: 0.61	Max Speed Enable	Enable	
1: 0.62	Safe Max Speed	750	RPM
1: 0.63	Max Spd Stop Typ	Torque Off	

With the motor running and no faults

1. Move the dial labeled analog input 1 to either less than 1 or greater than 9.  
The HMI will indicate that the MSR57 has faulted.
2. Cycle the Safe OFF button (it should be flashing)
3. Move the analog input dial to a value between 1 and 9 so that a fault does not occur upon restarting
4. Press the flashing green button to restart the safety
5. Press the flashing yellow button to start motion

## Safe Torque OFF with standstill checking

Whenever a stop is initiated on this demo case, the MSR57 safety outputs drop the K300 safe OFFs, and the motor coasts to a stop. But since there is no inertia on the motor, the motor stops immediately. The standstill checking detects that the motor has come to standstill, and unlocks the gate. Although it appears that everything simply stopped and the gate unlocks when the Safe OFF button is pressed, in actuality, the standstill checking determined the motor was stopped, and then unlocked the gate.

Safe Torque OFF with standstill checking parameters within the MSR57 Safe Speed Relay

*	1: 0.44	Safe Stop Input	2NC	
*	1: 0.45	Safe Stop Type	Torque Off	
	1: 0.46	Stop Mon Delay	0.0	Sec
	1: 0.47	Max Stop Time	10.0	Sec
	1: 0.48	Standstill Speed	50.000	RPM
	1: 0.49	Standstill Pos	100	Deg
	1: 0.50	Decel Ref Speed	50	RPM
	1: 0.51	Stop Decel Tol	100	%

With motor running and no faults

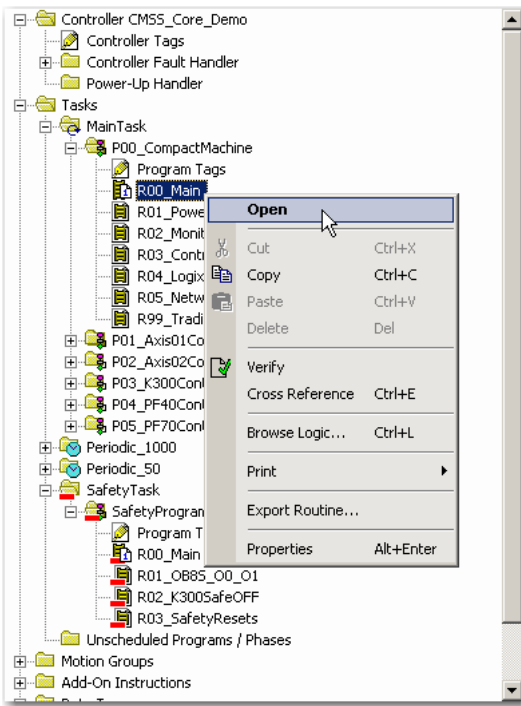
1. Push the Safe OFF button (top E-Stop).  
The HMI indicates that the MSR57 is in Safe Stop mode.
2. Pull out the gate key to simulate opening the gate door
3. Re-insert the gate key to simulate the completion of the routine and repetitive maintenance
4. Pull the Safe OFF (top) button back out
5. Press the flashing green button to restart the safety
6. Press the flashing yellow button to start motion

---

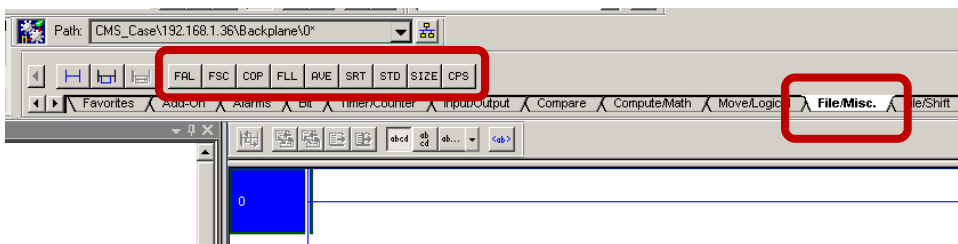
## Certified Safety Instructions

Over 50 certified safety instructions are available for use within the safety task. This is a subset of the total instruction list available in RSLogix 5000. A core safety principle is to keep safety code as simple as possible. Complex math and program control instructions are unavailable in the safety task due to the complexity they could inject into the safety code. For example, there are nine (9) instructions available within the File/Misc tab for standard routines. There is only one (1) available within the same tab for safety routines, the COP instruction.

1. Open the **R00\_Main** routine within the **P00\_CompactMachine** program

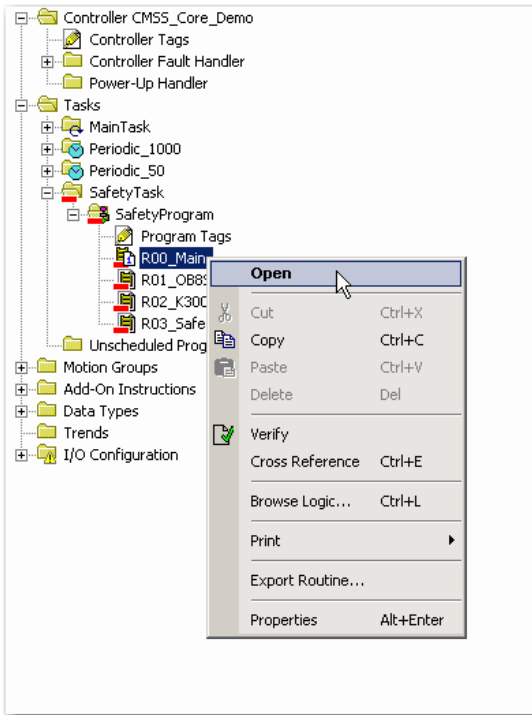


2. On the **Language Elements** toolbar, select the **File/Misc.** tab.

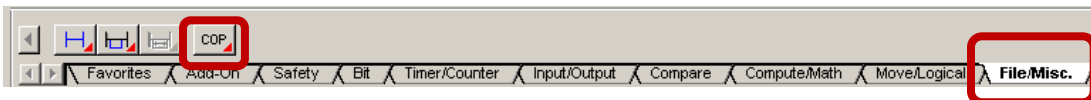


Note that nine (9) instructions are available

3. Close the **R00\_Main** routine
4. Open the **R00\_Main** routine in the SafetyTask



5. On the **Language Elements** toolbar, select the **File/Misc.** tab.



Note that one (1) instruction is available

6. Close the **R00\_Main** routine

## Safety Input Instructions

Under the safety tab, you will see a set of instructions developed to control specific safety functions. We will begin by focusing on the safety input instructions. The base instruction is DCS (Dual Channel Input Stop) is typically used for devices that STOP safety outputs, for example, an Emergency Stop button.

To show the features of this instruction, we need to change the configuration of the EStop channel pair from Equivalent to Single. This allows the DCS instruction to check for discrepancy rather than the safety IO module.

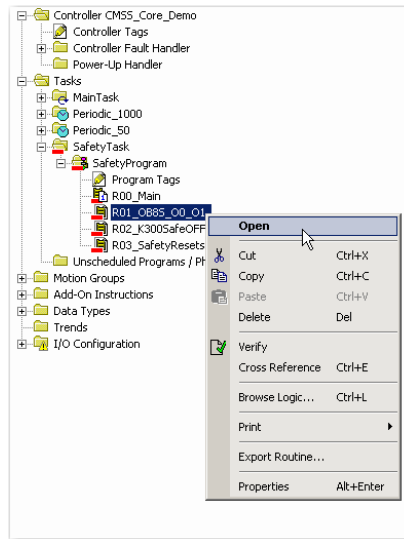
When you cycle the Emergency Stop button on the demo case, notice that the output O1 simply follows the state of the button.

Press the EStop wire OFF button to drop out channel B of the Emergency Stop button (input 3 on the IB8S in slot 2). The channels are now diverse, and if they remain diverse until the 3 second discrepancy timer expires, the DCS declares a fault. The fault remains until the wire OFF is repaired, and the reset button is pressed. Notice that the DCS output O1 does not go HI until the Emergency Stop button is cycled to prove that the fault has been fixed.

To summarize, the DCS instruction monitors dual channel devices and sets the output when both channels are in the active state (HI), and proper restart actions are completed. If the channels are not equivalent for longer than the discrepancy time, a fault is declared.

Many of the other safety input instructions simply build onto this base functionality.

### 1. Open the R01\_OB8S\_00\_01 routine in the Safety Task



### 2. Select the Safety tab on the language elements toolbar

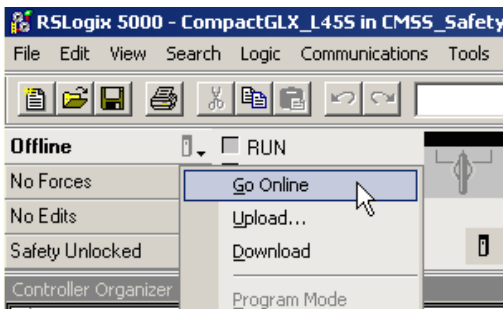


Under the safety tab, you will see a set of instructions developed to control specific safety functions. We will begin by focusing on the safety input instructions. The base instruction is DCS (Dual Channel Input Stop).

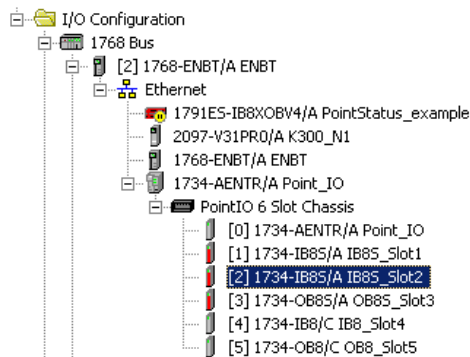
DCS stands for Dual Channel Input Stop. This safety instruction monitors a dual channel safety device whose main function is to stop a machine safely. The instruction enables the output if both channel A and Channel B are in the active (HI) state, with no faults, and a LO to HI transition is seen on the reset button. The instruction drops out the output if either channel goes to the safe (LO) state, or the input status parameter goes LO. The input status parameter is typically the status of either the channels themselves or the module they are wired to. If used in this typical fashion, a LO channel, channel fault, or module fault will drop out the output of the DCS. Based on the settings of the restart parameters, a test of the device can even be required before the output is energized.

To show the features of this instruction, we need to change the configuration of the EStop channel pair from Equivalent to Single. This allows the DCS instruction to check for discrepancy rather than the safety IO module.

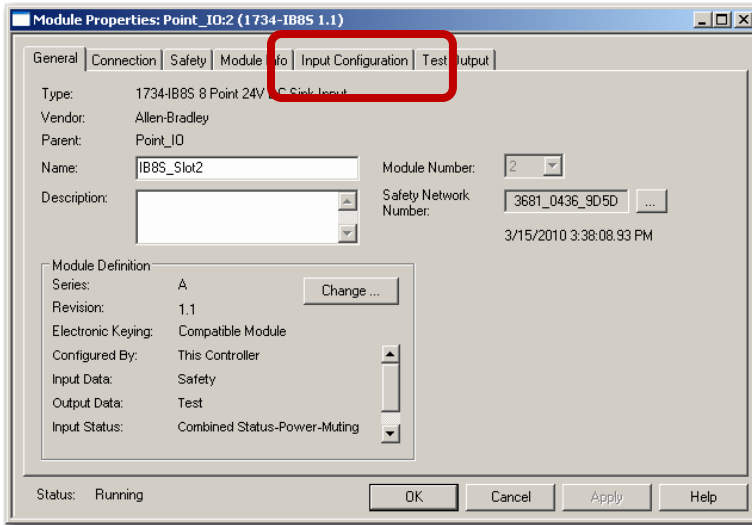
3. If not already online, In the Online toolbar, change the controller mode to Online by selecting Go Online.



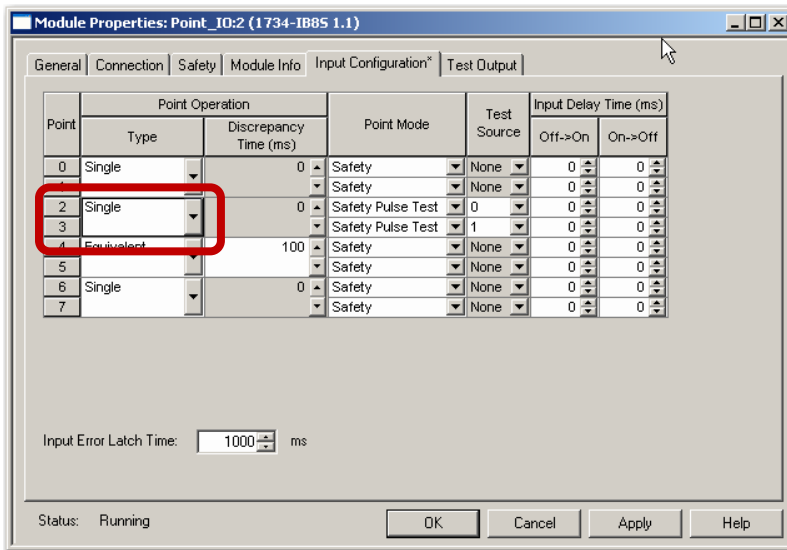
4. Select Yes to any prompts.
5. Open the 1734-IB85\_Slot2 Properties dialog box.



6. Select the **Input Configuration** tab

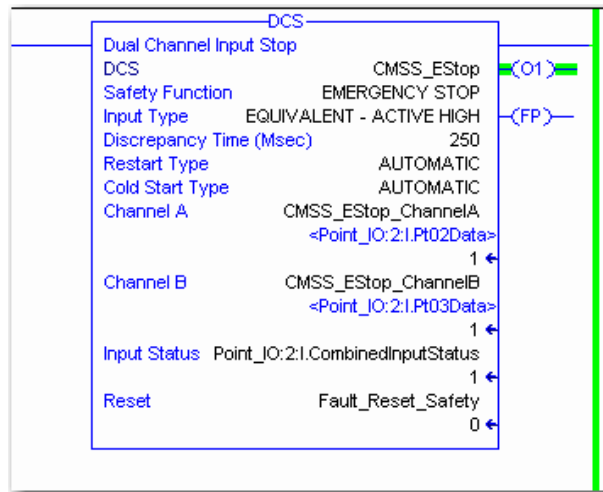


7. From the Input Configuration tab change the Point Operation for input channels 2 and 3 to single-channel mode.

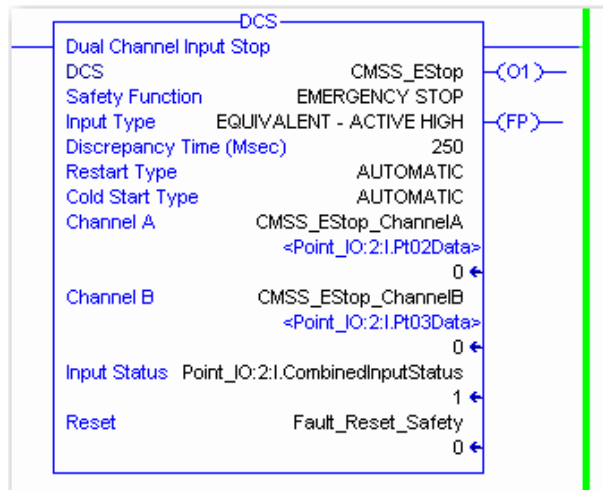


8. Click **Apply** and **Yes / Yes** at the prompts
9. Press **Cancel** to close the Module Properties dialog box
10. Cycle the flashing red selector switch on the demo case

11. Locate the DCS instruction on rung 0.



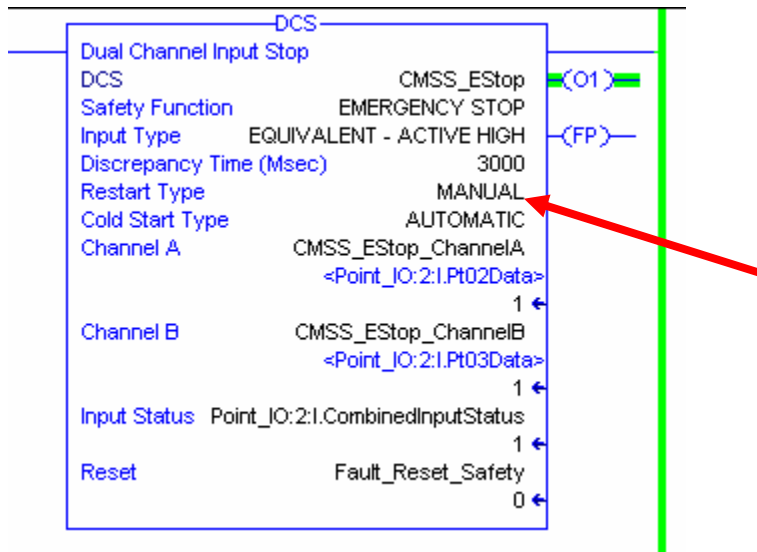
12. Press the EStop labeled **Emergency Stop** (lower) on the demo case.



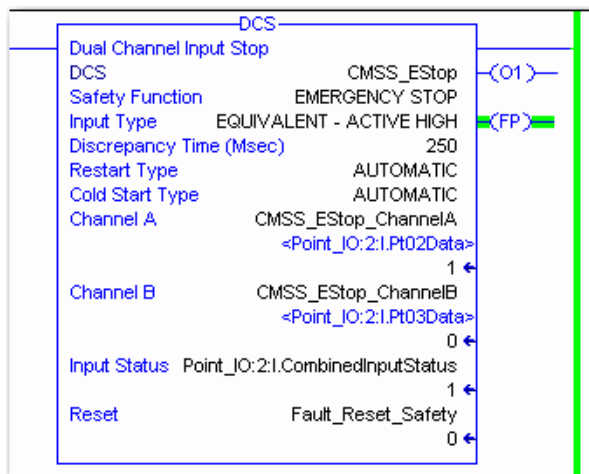
13. Release the lower Emergency stop button on the demo case.

When you cycle the Emergency Stop button on the demo case, notice that the output O1 simply follows the state of the button. This is due to the fact that the restart type is set to AUTOMATIC.

14. If confident making online edits, change the restart type to **MANUAL**



15. Press the Emergency Stop button (lower)  
Notice that the output goes LO
16. Pull the Emergency Stop button back out.  
This time the output does not energize.
17. Cycle the red selector switch (fault reset) to energize the output O1.  
Since the restart type is MANUAL, a manual reset is required to energize the output.
18. Change the DCS back to **AUTOMATIC** restart type
19. To simulate a discrepancy fault, press the **E-STOP WIRE OFF** button on the demo case.



When the EStop wire OFF button is pressed, channel B of the Emergency Stop button is broken. Watch input 3 on the slot 2 IB8S. The channels are now diverse, and if they remain diverse until the 3 second discrepancy timer expires, the DCS declares a fault.

20. Release the **E-STOP WIRE OFF** button on the demo case.

21. Cycle the flashing red selector switch to reset the fault on the DCS instruction.

The fault remains until the wire OFF is repaired, and the fault reset is cycled.

22. Cycle the Emergency Stop button (flashing)

Notice that the DCS output O1 does not go HI until the Emergency Stop button is cycled to prove that the fault has been fixed.

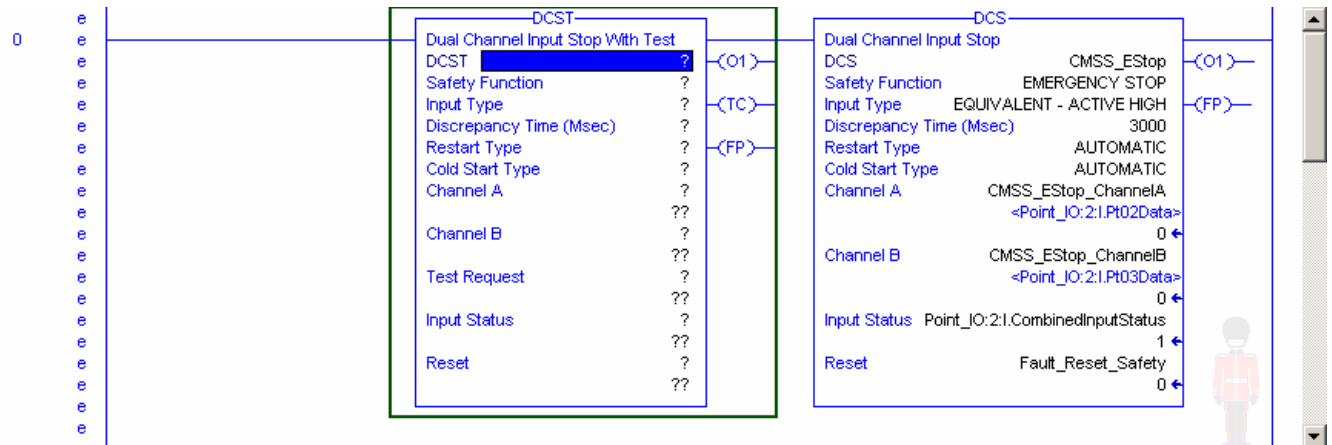
To summarize, the DCS instruction monitors dual channel devices and sets the output when both channels are in the active state (HI), and proper restart actions are completed. If the channels are not equivalent for longer than the discrepancy time, a fault is declared.

Many of the other safety input instructions simply build onto this base functionality.

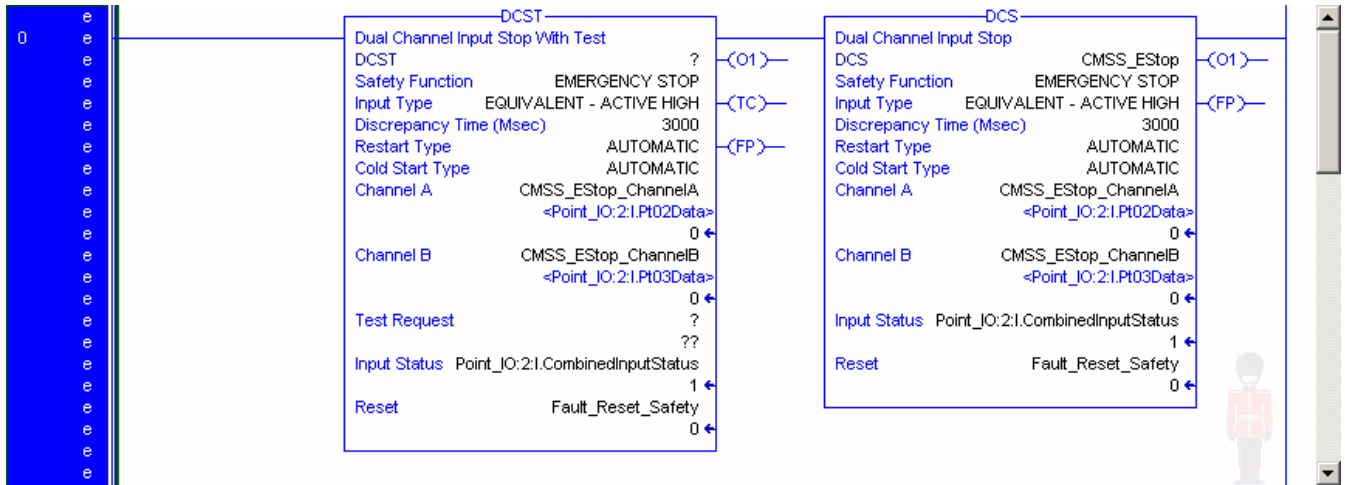
Let's add a DCST instruction and explore additional functionality.

23. Go offline

24. Add a DCST instruction to rung 0 as shown



25. Drag all the tags over from the DCS except for the DCST base tag and Test Request as shown

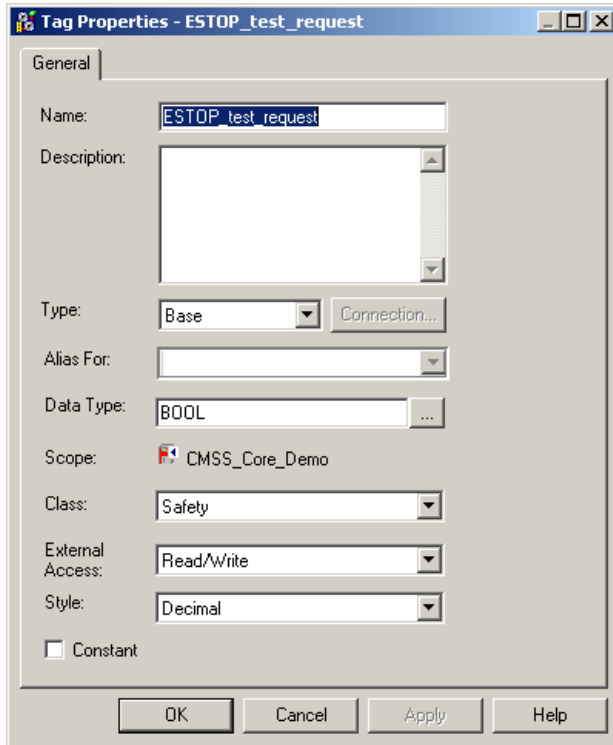


26. Enter a tag called Test for the DCST base tag. Create the tag. The default properties are OK.

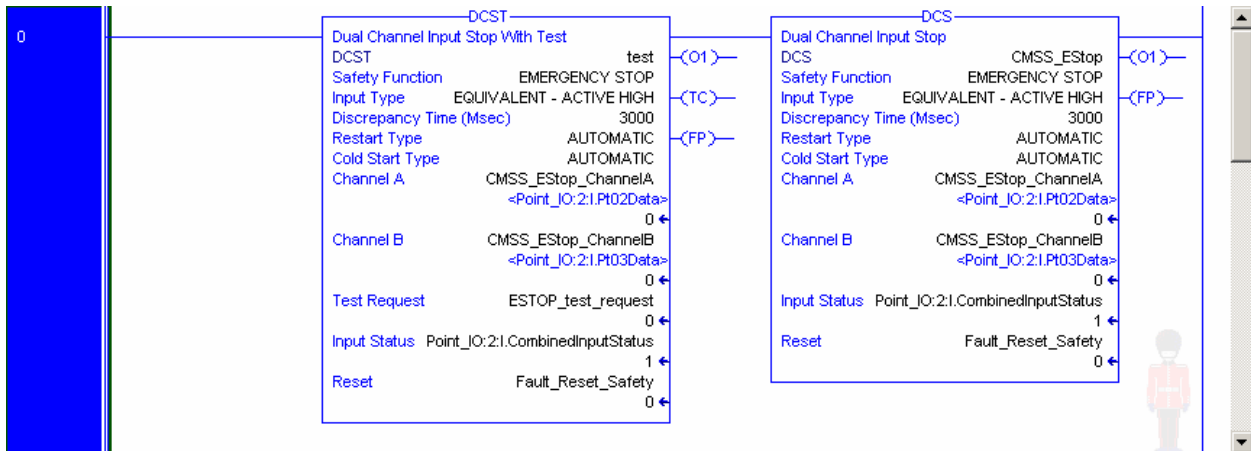
The 'New Tag' dialog box shows the following fields and values:

- Name: test
- Description: (empty)
- Usage: <normal>
- Type: Base
- Alias For: (empty)
- Data Type: DCI\_STOP\_TEST
- Scope: CMSS\_Core\_Demo
- Class: Safety
- External Access: Read/Write
- Style: (empty)
- Constant:
- Open Configuration:

- Enter **ESTOP\_test\_request** in the Test Request field and create the tag. Make it a BOOL and verify the class is SAFETY (those are the defaults).



Your code should appear as follows



- Save your code
- Download the code
- Return to **Remote Run** mode
- Press the Emergency Stop button (lower)
  - The outputs (O1) on both the DCST and DCS go LO

32. Pull out the EStop button

Both outputs go back HI

33. Set **EStop\_test\_request** HI and then LO (double click on the data value in the instruction)

On the LO transition, the output will go LO and the Test Command (TC) bit will go HI. TC informs the operator that the EStop is waiting to be functionally tested.

34. Press the EStop button (lower)

Notice that TC goes LO

35. Pull the EStop back out

The output will go HI.

You have just performed a functional test of the EStop button. This is common at the beginning of shifts, for example.

36. Close the **R01\_OB8S\_00\_O1** safety routine

37. Go back to the **IB8S\_Slot2** and configure inputs 2 and 3 as Equivalent. Verify that the discrepancy time is configured for 3000ms.

38. **[Apply] [Yes] [Yes]**

39. Close the IB8S\_Slot 2 module properties window using **[Cancel]**

40. Cycle the flashing red fault reset selector switch

41. Open the **R04\_Safety Instructions** safety routine

The safety instructions continue to add on to this base functionality.

The **DCST Dual Channel Input Stop with Test** adds one additional input and output parameter. The new input parameter, Test Request is used to generate a Test Command. Test Command is used to force a functional test of the device, typically a device with dual dry contacts. The functional test requires cycling the input channels from active, to safe, and back to the active state. During this test, wiring faults can be detected. Note that the output (O1) is de-energized during the test.

The **DCSTL Dual Channel Input Stop with Test and Lock** is typically used for devices that can be locked, such as a safety gate. It adds 3 additional input parameters and one additional output. Unlock Request and Hazard Stopped are used to generate an Unlock Command. And the Lock Feedback is used to ensure that the gate is operating properly.

The **DCSTM Dual Channel Input Stop with Test and Mute** adds Mute capability for presence sensing devices such as light curtains. When muted, the channels can go LO without affecting the instruction output O1. The Test Type (Manual or Active) of this instruction can be configured to work just like the DCST for devices that cannot test themselves. For intelligent devices, such as light curtains that have the ability to test themselves automatically, the output (O1) remains active (HI) during the test,

as long as the test completes within a configurable (Test Time) amount of time.

The **DCSRT Dual Channel Input Start** replaces the start types with an enable signal. This instruction is typically used for safety devices that start outputs, such as an enable pendant. For example, if the enable bit is HI, and the pendant inputs transition from LO to HI when the pendant is squeezed, then the instruction output goes HI.

The **DCM Dual Channel Input Monitor** is used for devices that toggle between HI and LO during normal run operation, and so you do not STOP the machine every time they go LO. CAM switches on a press are a good example of these types of devices. Because LO is normal, and not a condition to go to safe state, the IS (Instruction Status) output was added to know that LO is a faulted condition.

The **SMAT Safety Mat** instruction is used for dual channel devices that short the two channels when a demand is placed on it. Complications arise because the channel to channel short cannot cause faults that are difficult to recover from. By sourcing and sinking both channels, we can avoid these faults.

The **TSAM Two Sensor Asymmetrical Muting** instruction requires a specific order in which the two sensors and light curtain are blocked. This type of alignment is much harder to defeat than symmetrical muting. But it is also harder to setup and align the sensors. [S1 LO / S2 LO / LC LO / LC HI / S2 HI / S1 HI]

The **TSSM Two Sensor Symmetrical Muting** requires less of a specific order, is easier to defeat, yet easier to setup and align the sensors. [S1-S2 LO / LC LO / LC HI / S1-S2 HI]

The **FSBM Four Sensor Bidirectional Muting** instruction is used for larger objects, for example, a fork truck. Length of the object is the key item as the object must block all 4 sensors simultaneously as it passes through. The fork truck can break the beams simultaneously, but not a person. [S1 LO / S2 LO / LC LO / S3 LO / S4 LO / S1 HI / S2 HI / LC HI / S3 HI / S4 HI]

The **THRSe Two Hand Run Station enhanced** instruction is for devices such as two hand run stations, that use 1 NO and 1 NC contact on each device. They can be easily bypassed when not in use (such as third shift when fewer employees operate a machine) using the Disconnected input. Both inputs must be pressed within a certain time of each other when the operator 'palms up'.

Notice that all these instructions deal with dual channel devices

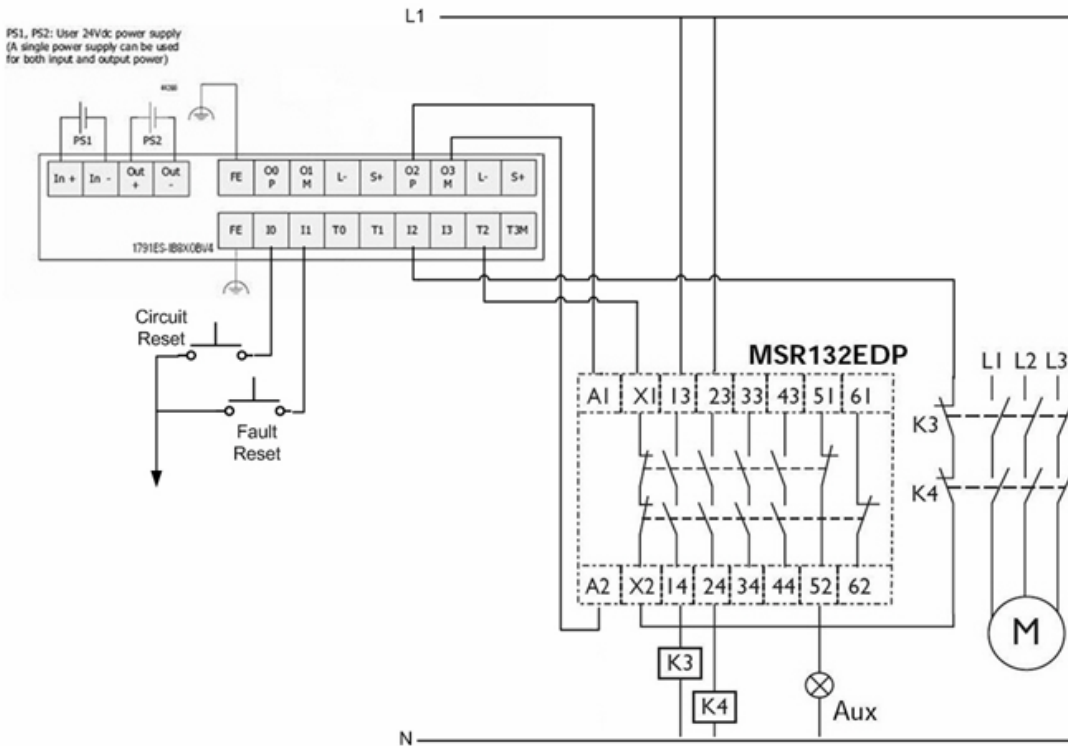
Question: what instruction would you use if your safety device was single channel?

The answer is an XIC (Examine ON). The DCS and the rest of the new safety instructions are used with Dual channel devices. They ensure that the dual channels are in sync. If your safety system is single channel, you just use the Boolean instructions.

42. Close the **R04\_Safety Instructions** safety routine

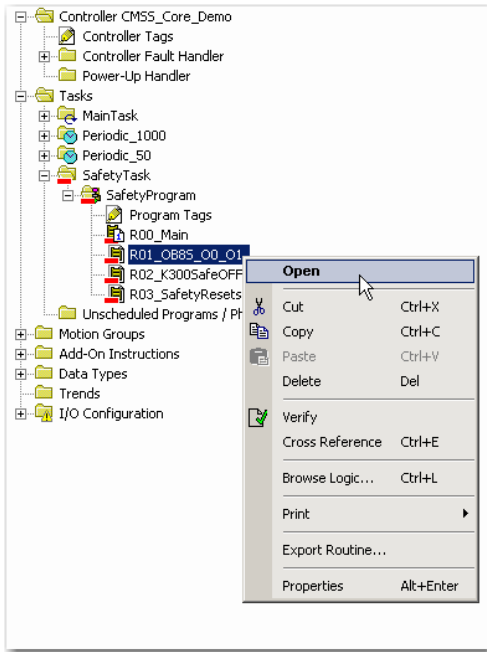
## Safety Output Instructions

Traditional electromechanical safety outputs typically appear similar to the diagram shown below. Two contactors are energized and output monitoring (feedback) is used to verify the contactors operated properly.

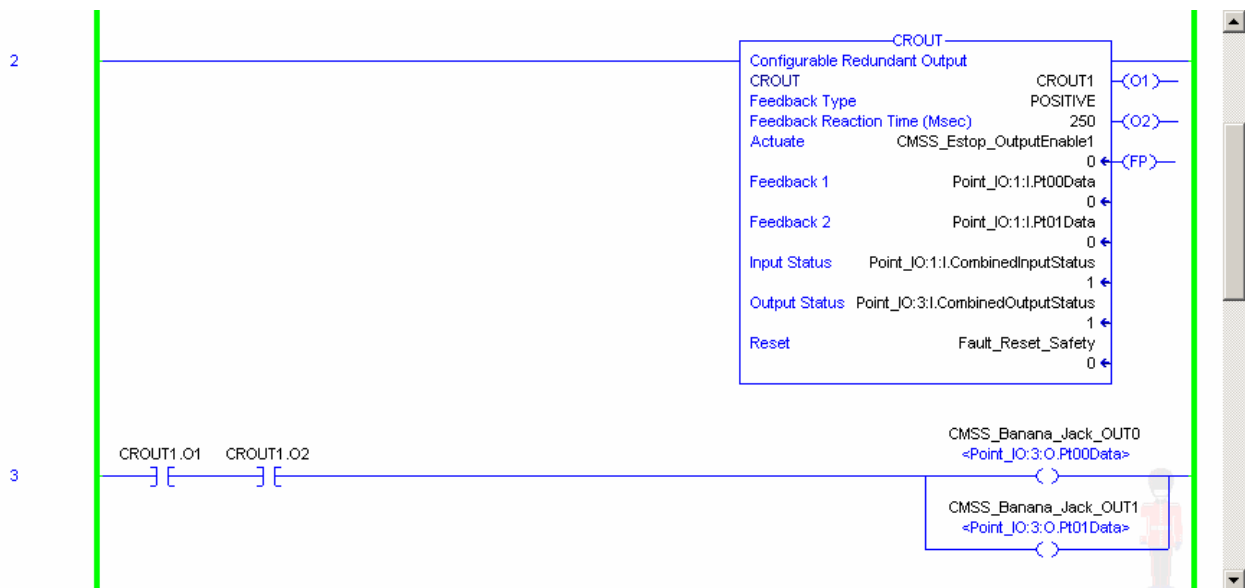


Since safety outputs typically are used in this fashion, there needs to be only one (1) safety output instruction, and that is the CROUT. The CROUT simply energizes two (2) outputs and monitors feedback. That happens to be exactly what safety relays do as well. Essentially, the CROUT has the same functionality as a safety relay. When the outputs are commanded HI the feedback is expected to follow within a configurable reaction time. If the feedback ever switches unexpectedly, the CROUT instruction faults.

1. Open the R01\_OB8S\_00\_01 safety routine.

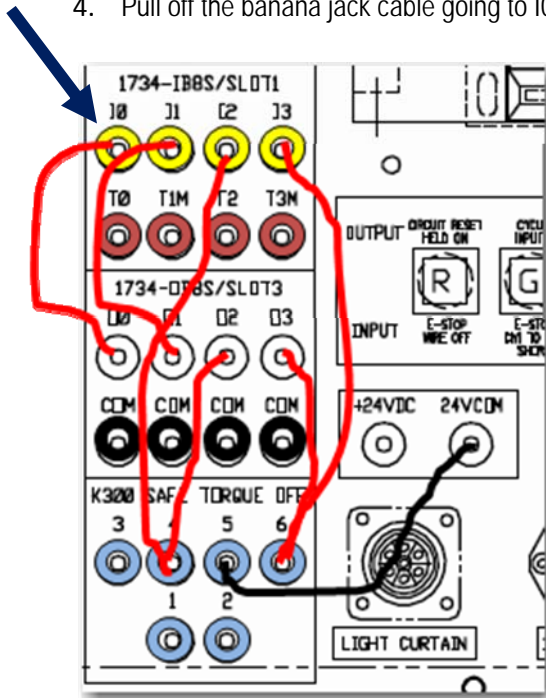


2. Scroll to rung 2 where the CROUT instruction is located.
3. Press the green fault reset button to energize the safety outputs



This CROUT instruction is being used to drive Safety Outputs O0 and O1 on the 1734-OB8S PointGuard module. Those outputs are wired to the white banana jacks on the demo case. We have already connected cables from those outputs to safety inputs I0 and I1 on the yellow banana jacks. These are the feedback signals for the CROUT. Since the instruction is configured for POSITIVE feedback, the feedback is LO when the outputs are LO and HI when the outputs are HI. If either of the feedback signals unexpectedly drops out, the CROUT will fault.

4. Pull off the banana jack cable going to I0 on the 1734-IB8S module to simulate a feedback fault.



Question: Why did Feedback 2 also go LO?

Because when the instruction faulted, the outputs were logically dropped out. This causes both feedback channels to drop out as well.

5. Re-attach the banana jack cable to I0
6. Cycle the red fault reset to clear the fault (it is not flashing)
7. Press the green circuit reset button (not flashing) to turn the CROUT outputs back on
8. Close the R01\_OB8S\_00\_01 routine

In summary, the CROUT instruction duplicates the safety functions of a safety relay, controlling dual outputs and monitoring up to two (2) feedback channels.

---

## Falling Edge Manual Reset

The following is taken from ISO 13849-1

### 5.2.2 Manual reset function

The following applies in addition to the requirements of Table 8.

After a stop command has been initiated by a safeguard, the stop condition shall be maintained until safe conditions for restarting exist.

The re-establishment of the safety function by resetting of the safeguard cancels the stop command. If indicated by the risk assessment, this cancellation of the stop command shall be confirmed by a manual, separate and deliberate action (manual reset).

The manual reset function shall

- be provided through a separate and manually operated device within the SRP/CS,
- only be achieved if all safety functions and safeguards are operative,
- not initiate motion or a hazardous situation by itself,
- be by deliberate action,
- enable the control system for accepting a separate start command,
- only be accepted by disengaging the actuator from its energized (on) position.

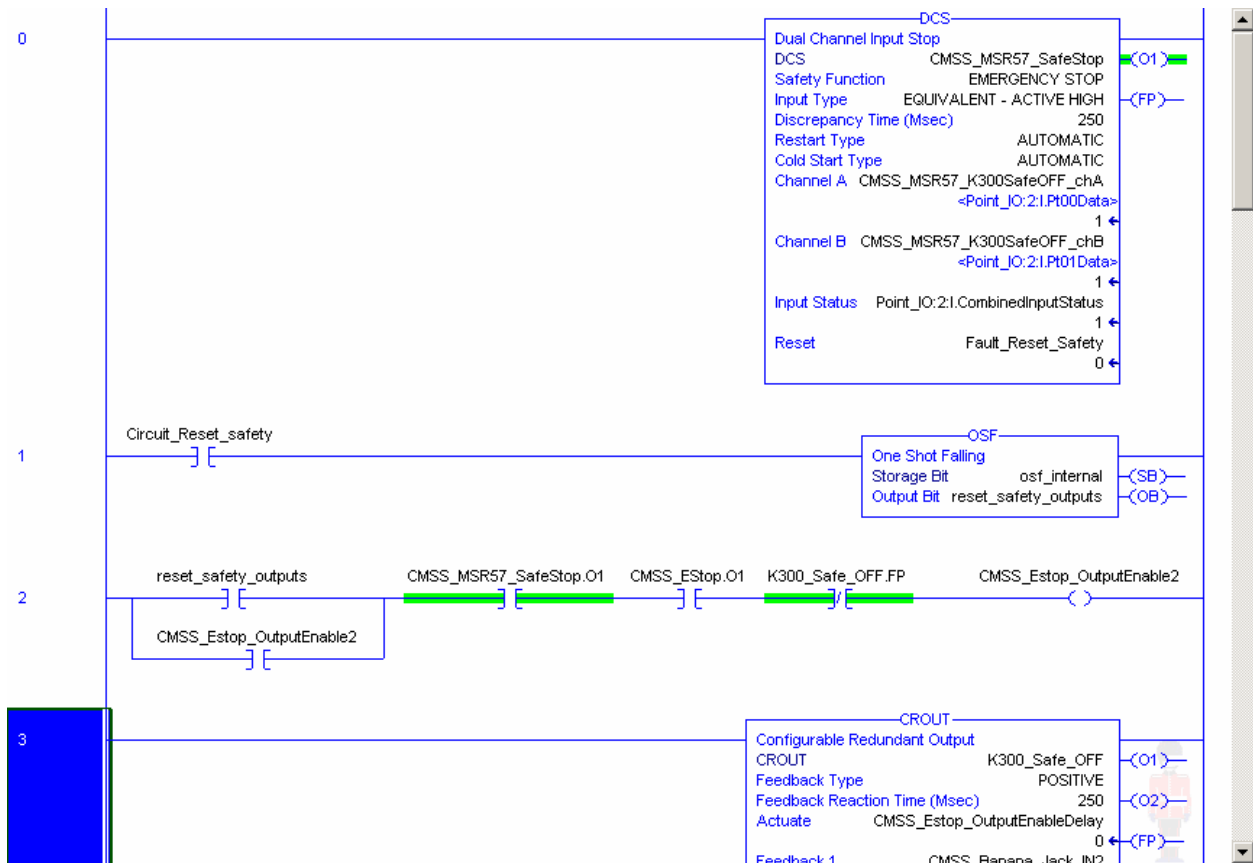
The performance level of safety-related parts providing the manual reset function shall be selected so that the inclusion of the manual reset function does not diminish the safety required of the relevant safety function.

It essentially says that a manual reset should operate on the trailing edge of the reset signal. The trailing edge is part of the reason a safety function such as reset can be accomplished with a single channel button. The reset button cannot diminish the safety rating of the safety function it is resetting. If the safety function is PLd, the reset button must not de-rate it. Using the trailing edge, a short to HI is tolerated and a wire OFF is tolerated.

1. With the motor spinning, press the Emergency Stop button (lower)
2. Release the Emergency Stop button
3. Press and HOLD the green safety circuit reset button

Notice that the K300 status (safety outputs) comes on when the button is pressed. This does not 'officially' meet the standard shown above.

- Go offline make the following code changes to **R02\_K300SafeOFF** safety routine. Rung 1 is new. Rung 2 has some edits. **OSF\_internal** and **reset\_safety\_outputs** are new tags that have to be created.



- Download the code and perform steps 1 thru 3 again

Notice the K300 status light energizes when the button is released

Some of the non-safety sequencing to step thru the reset sequence using flashing lights gets messed up by this new code. So you will have to hit the green button an additional time before the yellow motion start button flashes. Rather than fix it, let's just live with this the rest of the lab.

- Close the **R02\_K300SafeOFF** safety routine

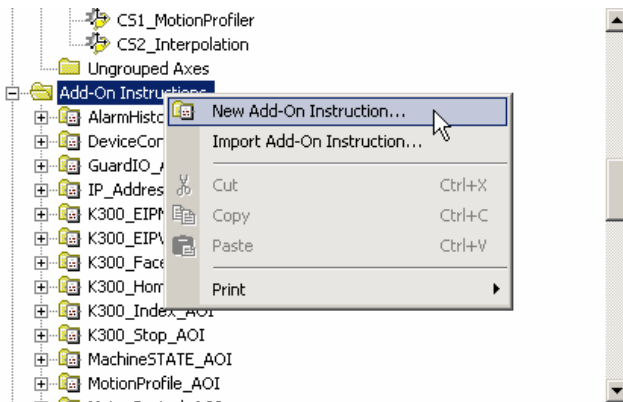
---

## Safety AOIs

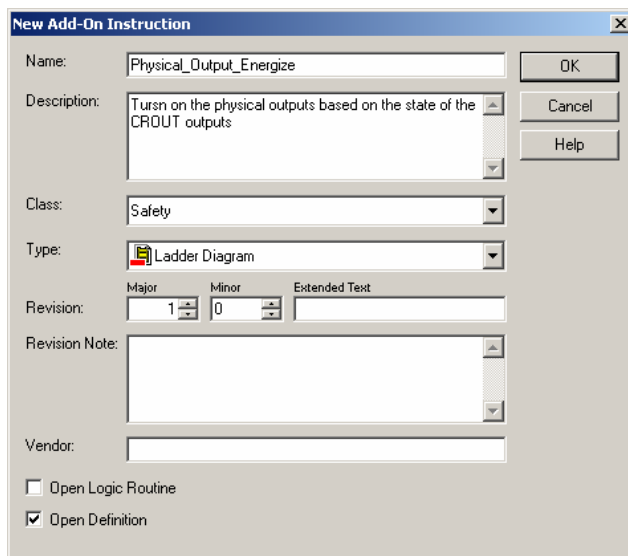
There is a misconception that users are not allowed to develop AOIs for a safety task, and that only certified safety instructions can be used. This is simply not true. OEMs can develop AOIs (intellectual property) for either standard or safety routines. And to verify that no unwanted changes have been made to their AOIs, a signature ID can be added. This is a very similar concept to the Safety Signature of safety memory, except that this signature ID is applied to only a single Add-On Instruction (AOI). If the Signature ID of the AOI has changed, then this is not the same AOI that you developed.

Let's see how easy it is to generate a safety AOI. You will create a very simple safety AOI that replaces rung 4 in the safety routine R02\_K300SafeOFF.

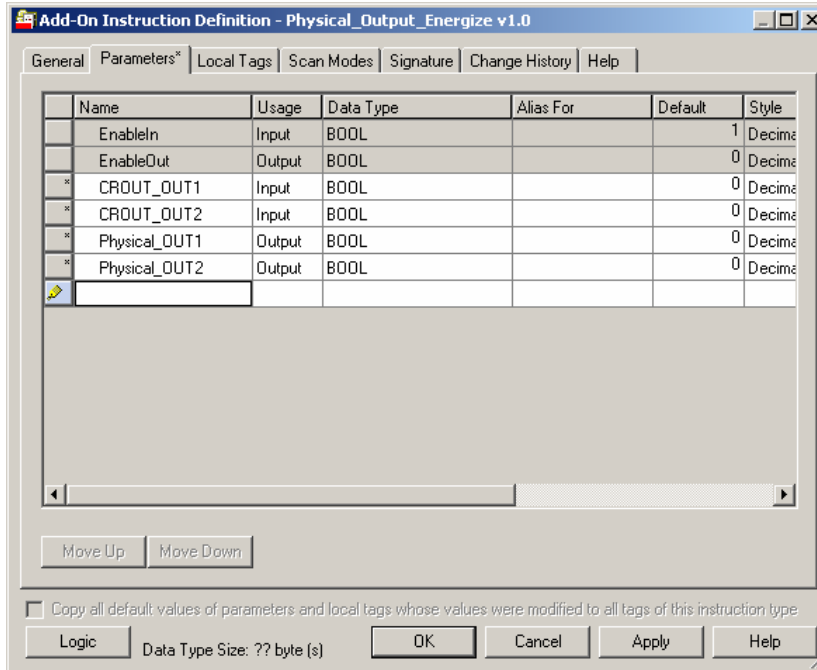
1. Go offline
2. Right click on Add On instructions and select **New Add-On Instruction**



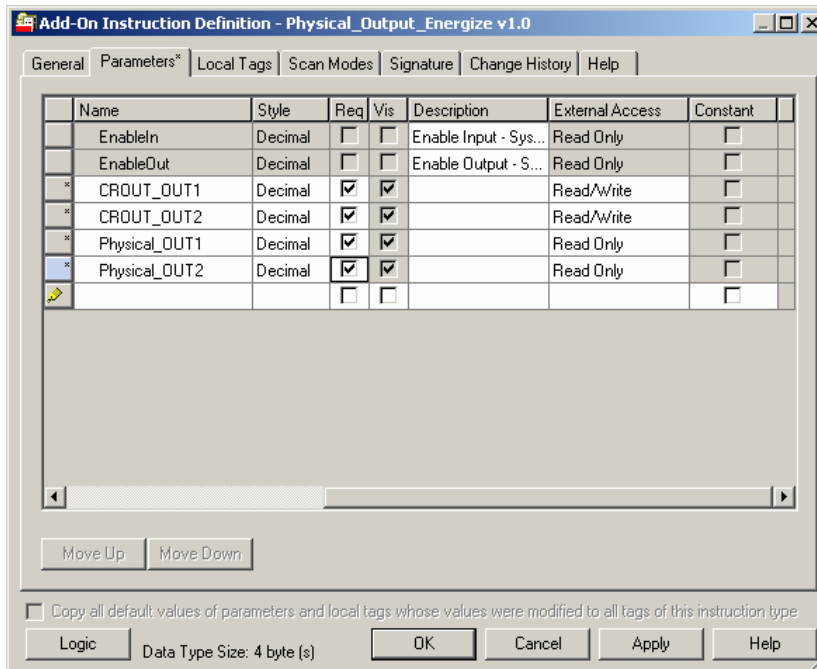
3. Fill in the parameters as show. Make sure you select 'Safety' Class.

A screenshot of the 'New Add-On Instruction' dialog box. The dialog has several fields and buttons. The 'Name' field contains 'Physical\_Output\_Energize'. The 'Description' field contains 'Turn on the physical outputs based on the state of the CRDOUT outputs'. The 'Class' dropdown menu is set to 'Safety'. The 'Type' dropdown menu is set to 'Ladder Diagram'. The 'Revision' field has 'Major' set to 1 and 'Minor' set to 0. The 'Revision Note' field is empty. The 'Vendor' field is empty. There are 'OK', 'Cancel', and 'Help' buttons on the right side. At the bottom, there are two checkboxes: 'Open Logic Routine' (unchecked) and 'Open Definition' (checked).

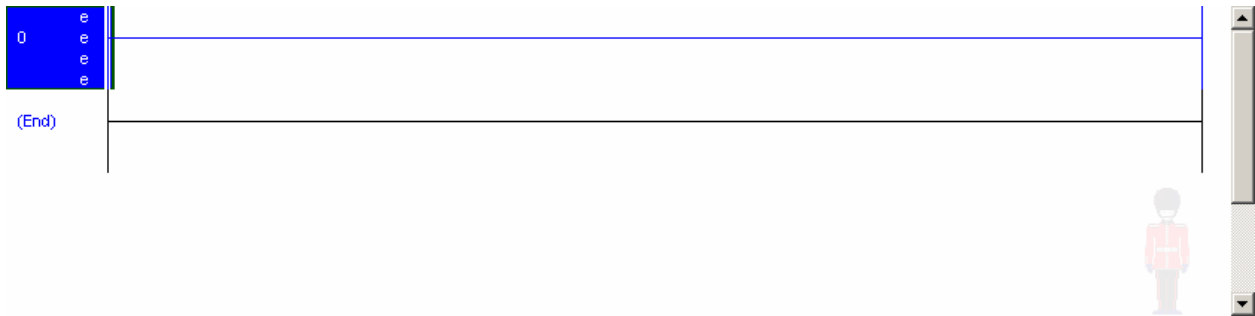
4. Select **OK**
5. Select the **Parameters** tab
6. Enter the following input and output parameters



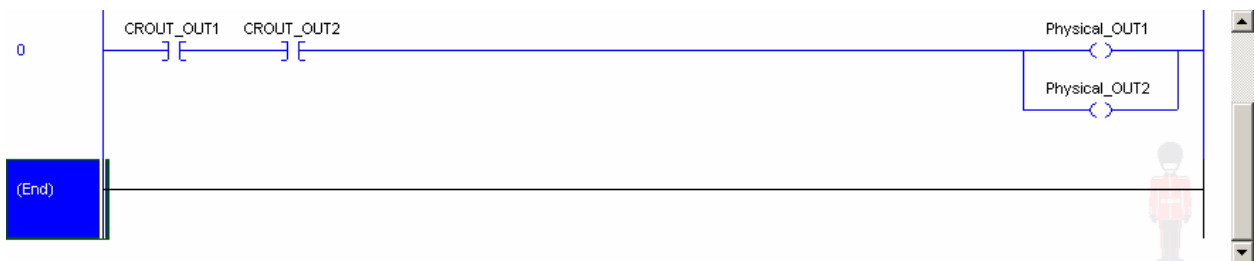
7. Make the parameters required and visible



8. Select **APPLY**
9. Click on the logic button (bottom left corner of AOI window)



10. Enter the following logic

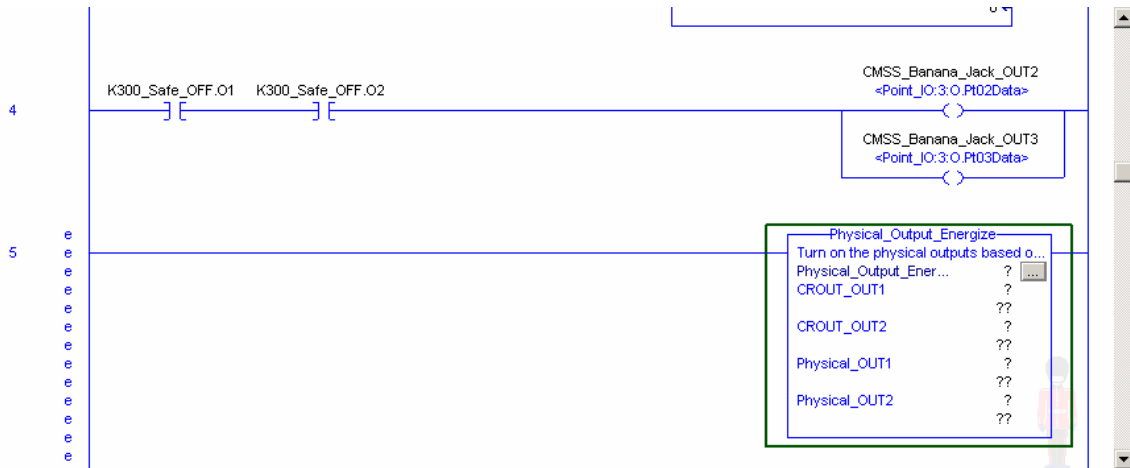


11. Close the ladder
12. Select **OK** to close the AOI properties
13. Add a new rung 5 (right click on rung 4 and select 'add rung') in the **R02\_K300SafeOFF** safety routine
14. Select the Add-On instruction tab and click on the only safety AOI in the list.

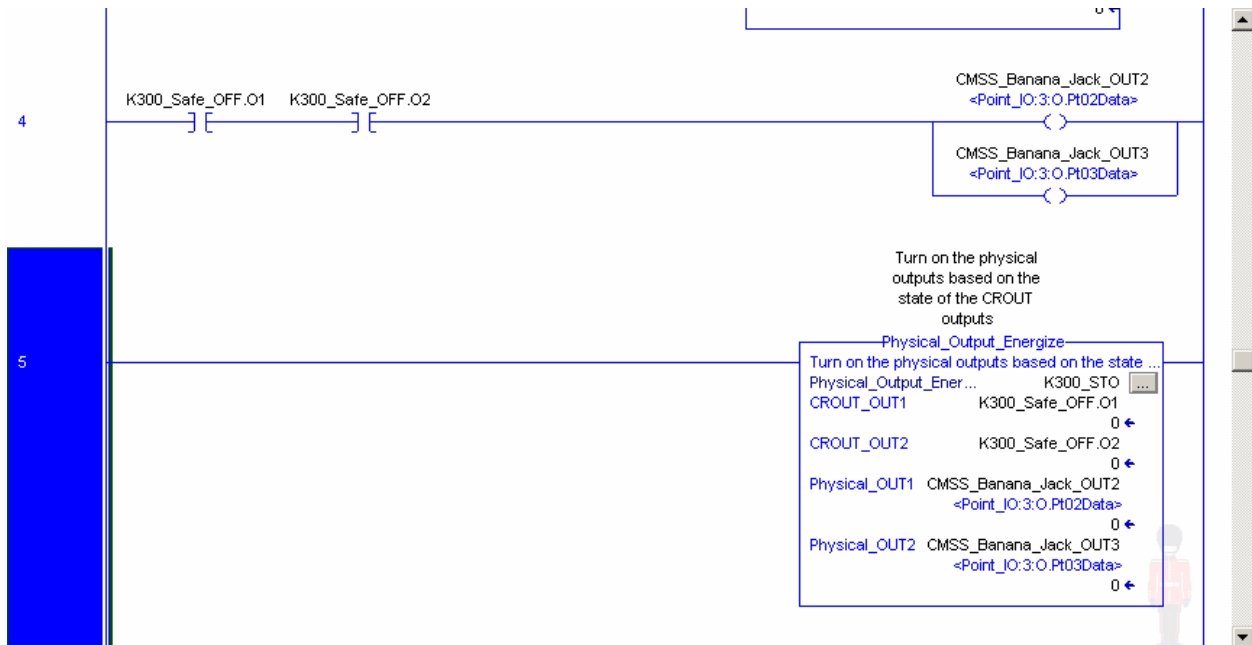
Note that the standard AOIs do not appear and cannot be used in the safety task.



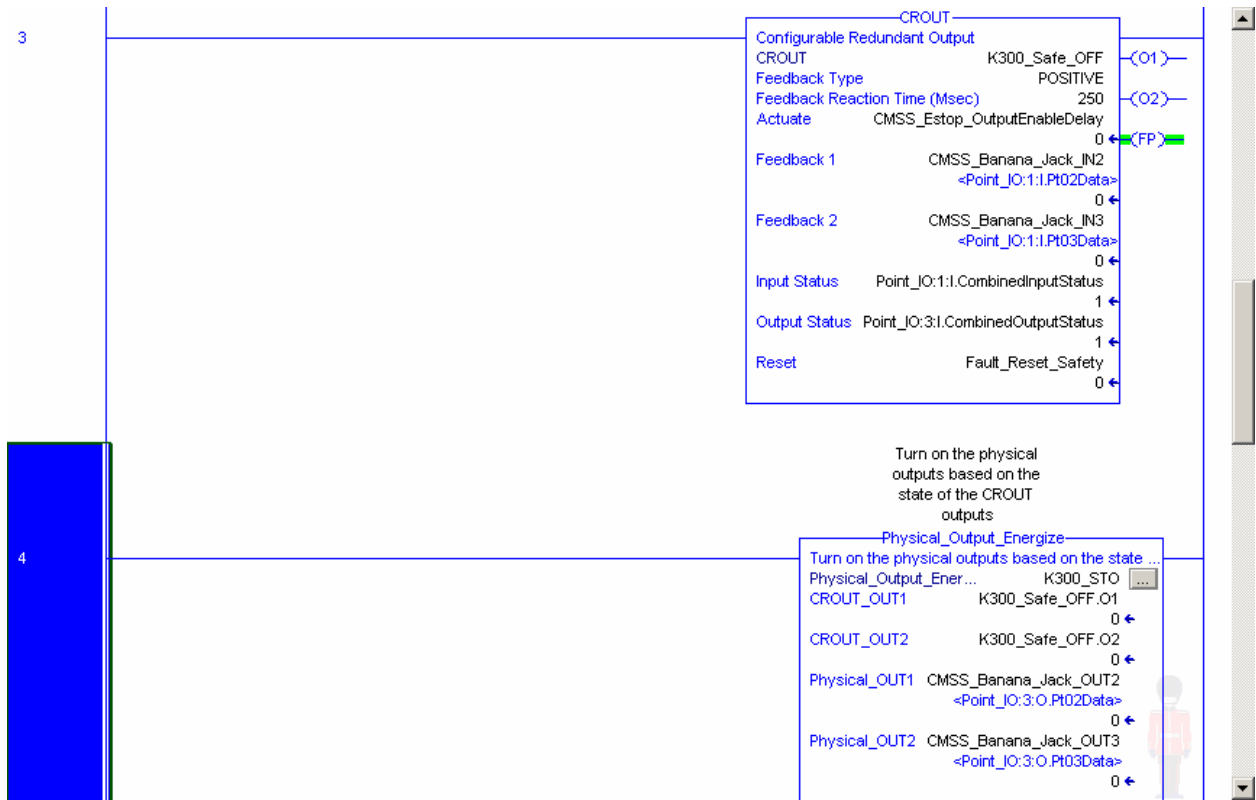
When Step 14 is complete, the logic should appear as follows



15. Enter **K300\_STO** for the AOI base tag and press [Enter]
16. Right click on **K300\_STO** and select **New K300\_STO**. Note the data type is **Physical\_Output\_Energize**
17. Select **OK**
18. Since the AOI will be replacing rung 4, simply drag the four (4) tags from rung 4 to the parameters of the AOI. When complete the AOI instruction should appear as follows:



19. Delete the original rung 4. The final code should appear as shown below.



There is nothing fancy about this AOI; just some simple code that we can use for demonstration purposes. The goal of this section is to highlight the High Integrity function of AOIs; not the AOIs themselves.

20. Close the **R02\_K300SafeOFF** routine

21. **Save and Download** your work at this point.

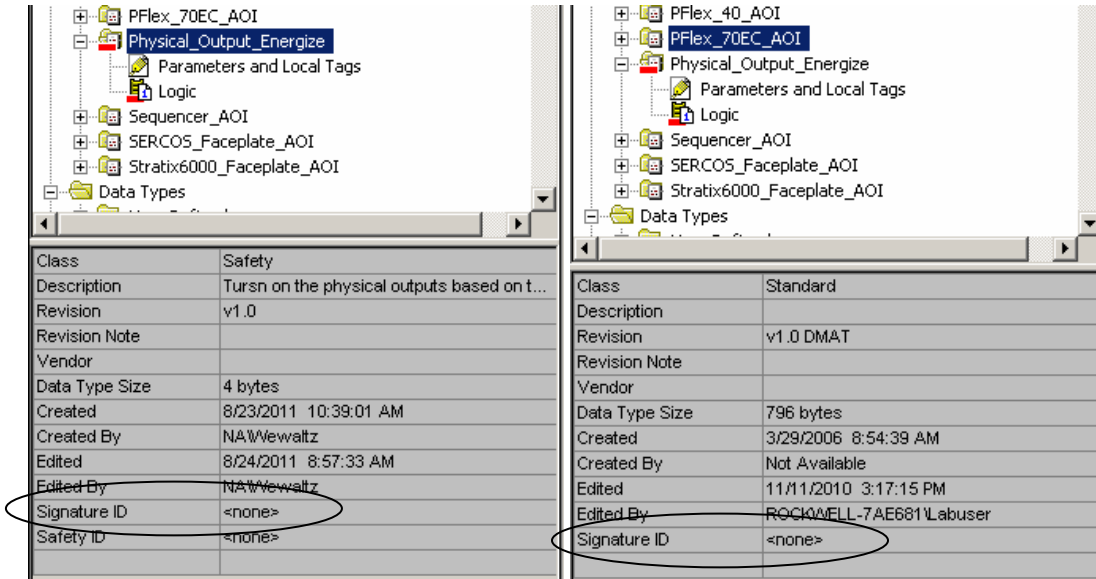
22. Place the controller in **Remote Run**.

23. Validate that the safety function works the same with the AOI

- If either the Emergency Stop or Safe OFF pilot lights are blinking, verify they are in the active state (pulled out).
- Cycle **Fault Reset** to clear any faults that may be present if the Red SS is blinking.
- Energize the K300 safety enables by pressing and releasing the **Circuit Reset green PB**
- The K300 status light should energize because the K300 STOs are in the active state.
- Press the green, then yellow flashing PBs and the Motor on the K300 demo case should begin to turn.
- Press the Emergency Stop button (lower) to stop the motor
- Pull out the Emergency Stop and press fault reset and motion start (green/yellow) to restart the motor

24. Left Click on the *Physical\_Output\_Energize* AOI in the controller organizer and observe the status window at the lower left of your screen. Then do the same for any of the standard AOIs.

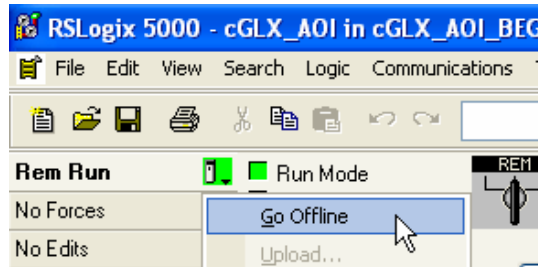
Note that both safe and standard AOIs have Signature IDs



What differentiates a high integrity AOI is the existence of a Signature ID on the instruction. This applies for both standard AOIs and safety AOIs. Once the programmer has validated an AOI, they can generate a Signature ID for the instruction that is unique in time.

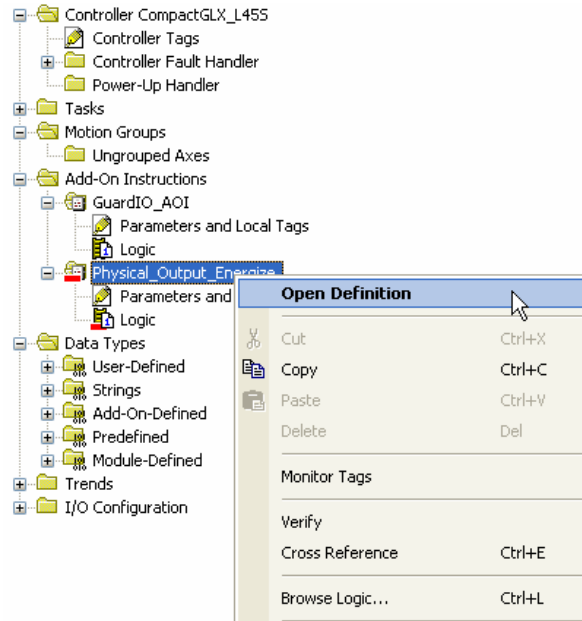
The **Signature ID** guarantees the integrity of the offline (non-compiled) code. If the signature ID has not changed, the code within the AOI has not changed.

25. *Go Offline* at this point.

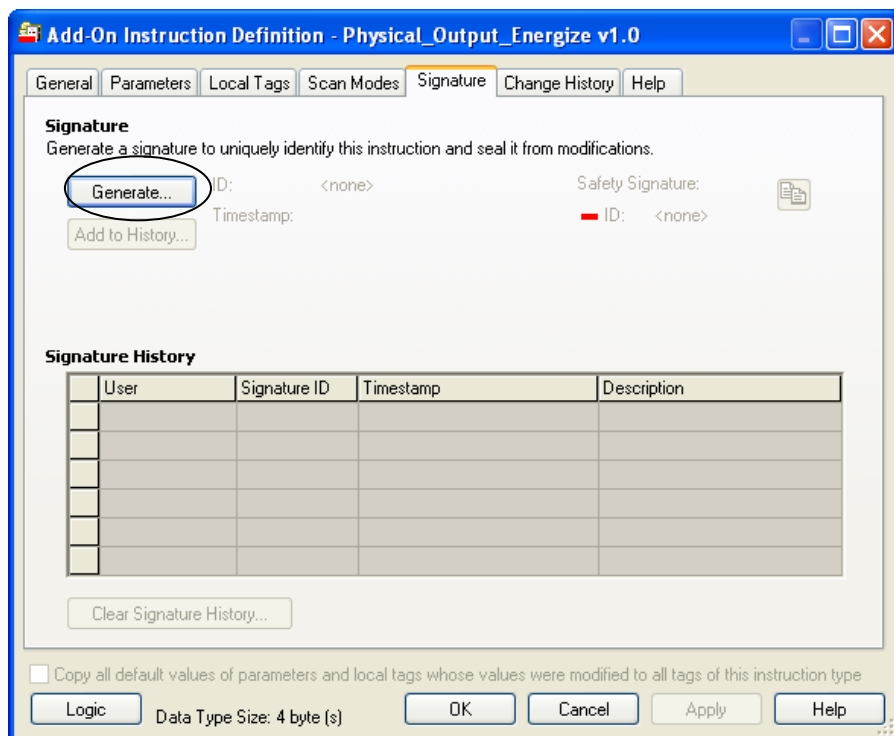


## Generate the Signature ID

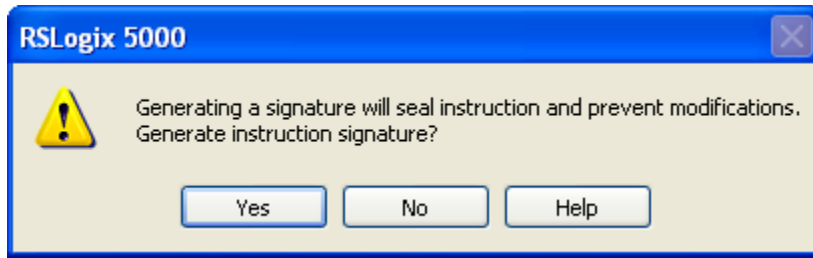
26. Right-click on the *Physical\_Output\_Energize* AOI in the controller organizer and select *Open Definition*



27. Select the *Signature* tab




28. Click *Generate* and the following prompt appears




29. Press YES to confirm that you do not want edits made to the AOI.

The Signature ID is generated. It consists of the ID and Timestamp, making it unique. No two (2) IDs will ever be the same.

**Signature**  
Generate a signature to uniquely identify this instruction and seal it from modifications.

ID: 3271B5BD Safety Signature:   
 Timestamp: 2011-08-26T16:51:16.340Z ■ ID: <none>

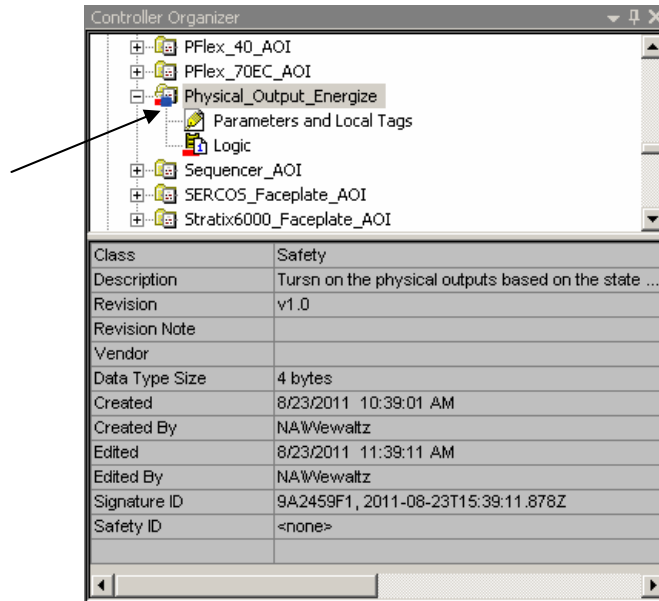
 The Safety Signature ID is automatically generated after download

**Signature History**

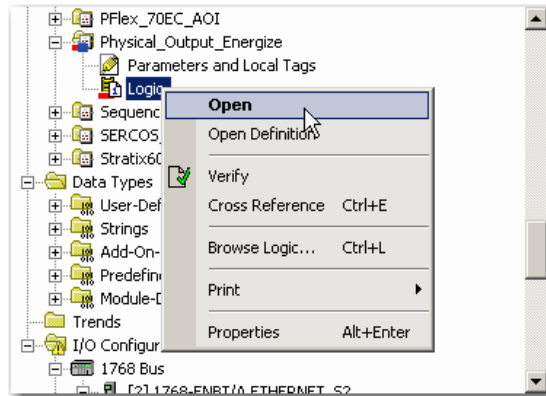
	User	Signature ID	Timestamp	Description
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

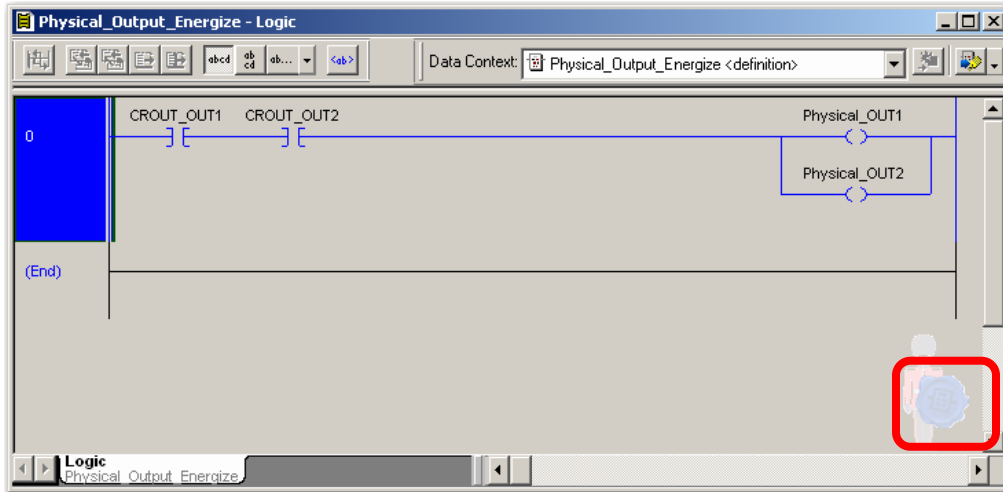
30. Press **OK** to close the definition for this AOI

Notice how the Signature ID status has been updated and the blue “seal” icon is shown on the AOI. This signifies that it is now a high integrity AOI.



31. Open the logic for the AOI by right clicking on Logic and selecting Open





The seal indicates the code cannot be edited. The Safety logo indicates this is a safety AOI.

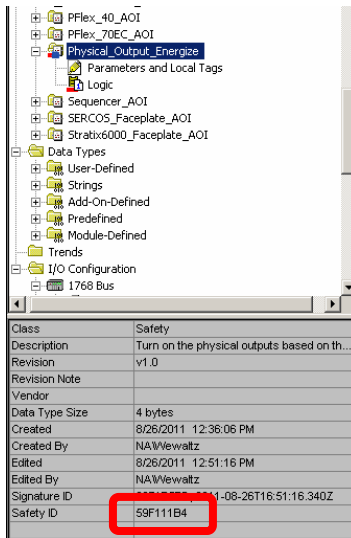
## Generate the Safety Signature ID

Editors are typically not safety certified, and this includes RSLogix 5000. This means that downloading safety routines is not classified as *safe*. Once downloaded, the safety programmer must verify that the code downloaded properly and is executing correctly. But there is an easy way to figure out if an AOI, essentially an instruction, has downloaded correctly. This is why high integrity AOIs used in safety applications will receive an additional signature, known as the Safety Signature ID.

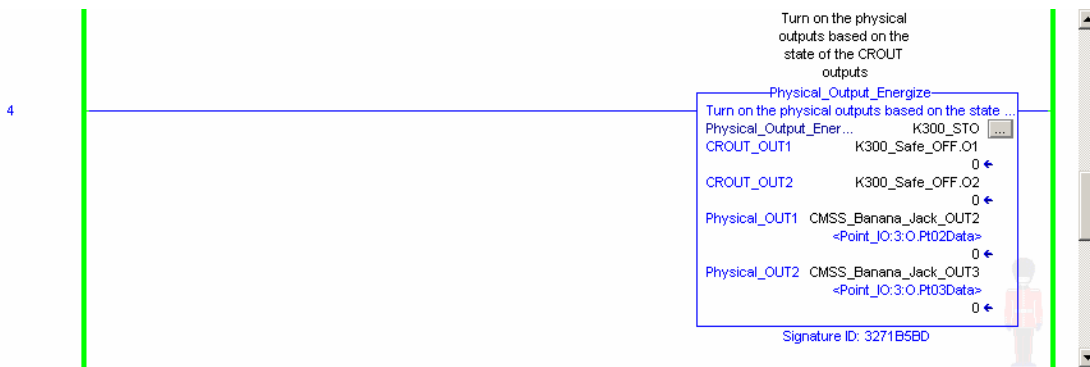
The **Safety Signature ID** deals with the online (compiled) code running in the safety controller. Each time the safety code is downloaded, the programmer can check this Safety Signature ID to verify that it was downloaded correctly.

32. **Save** and **Download** your project to the safety controller.
33. Place it back into **Remote Run** when done.
34. Click on the **Physical\_Output\_Energize** AOI again a in the Controller tree

The safety ID now exists, and any time this AOI is downloaded to a safety controller, the Safety ID should always be identical.



35. Return to the R02\_K300SafeOFF safety routine



The Signature ID is displayed on each instance of the high integrity AOI.

36. Close the safety routine R02\_K300SafeOFF

In summary:

For any given high integrity AOI in a particular project, all instances will have the same Signature ID because the code in each of them should be identical. If any edits are made, that Signature ID will change.

When the project is downloaded to a safety controller, the AOI instruction should always have the same Safety Signature ID. This shows that the instruction downloaded properly.

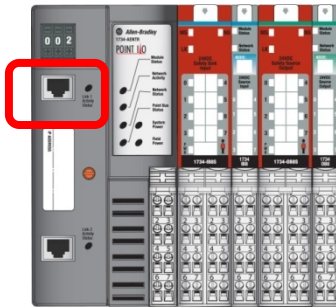
---

## CIP Safety

CIP safety connections are between the safety controller and safety IO modules. The 'path' between these end devices does not have to be safety rated, in fact, many of our customers are using wireless to connect safety IO modules. If the path faults, the safety rated end devices simply timeout and go to the safe state. That is why the motor stops when the Ethernet cable is pulled out. The integrity of the path will certainly affect the availability of your machine, but it has no effect on the safety of your machine.

With motor running and no faults

1. Pull the Ethernet cable out of the 1734-AENT



2. Re-insert the Ethernet cable (top port)

Wait for the CIP safety connection to re-established (be patient, the red fault reset selector switch will start flashing when the connection is re-established)

3. Cycle the flashing red fault reset button
4. Press the green flashing button to restart the motor
5. Press the flashing yellow button to start motion

## Connection Reaction Time Limit (CRTL)

Let's discuss the safe state for input and output connections. If an output connection times out, the 1734-OB8S simply turns off its outputs. If an input connection times out, the L43S will continue to scan the safety task with all the safety inputs at logic zero. In other words, it will act as if a demand had been placed on all the safety inputs, which likely will stop your machine. How long does the system wait before declaring a timeout and shutting down the system? The timeout for a safety connection is called the CRTL (Connection Reaction Time limit) and it is configured by the user.

Once every RPI, the safety module places its inputs on the wire. After every safety program scan, the GuardLogix controller places its outputs on the wire. The CRTL is essentially how long to wait for any of these messages to get thru to the other side. If you use the Input CRTL default of 4xRPI, then you actually are allowing any of 4 separate messages from the safety module to get thru to the controller before timing out. If you adjust the CRTL down so that it equals the RPI, then each message must get thru to avoid a timeout. Reducing the CRTL reduces the worse case reaction time (good) but may also lead to nuisance trips (bad). Increasing the CRTL increases the worse case reaction time (bad) but also reduces the possibility of nuisance trips (good). You need to find the best compromise.

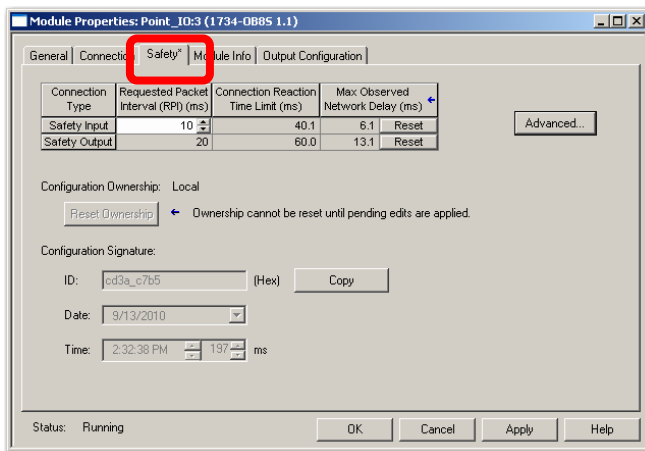
The CRTL is defined by three values:

- RPI (Requested Packet Interval)
- Timeout Multiplier
- Network Delay Multiplier

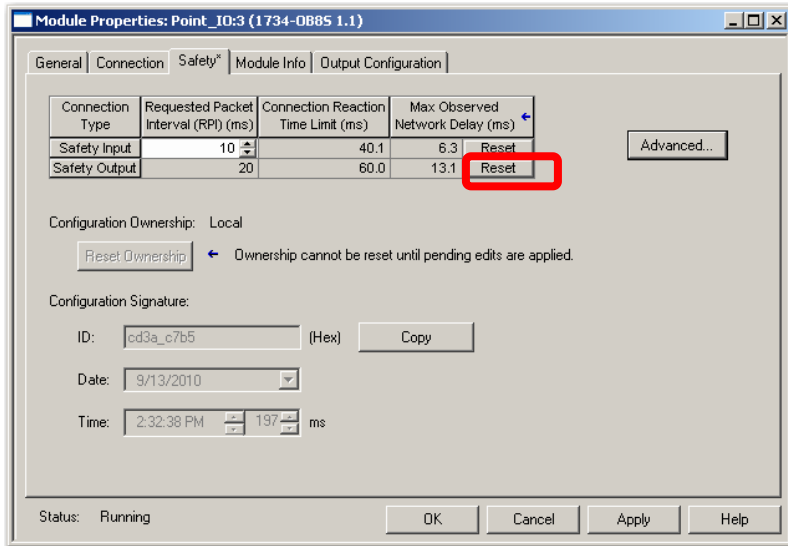
By adjusting these values, you can adjust the CRTL.

- Every additional Timeout Multiplier adds an additional RPI to the CRTL.
- Every additional 100% of Network delay adds an additional RPI to the CRTL.

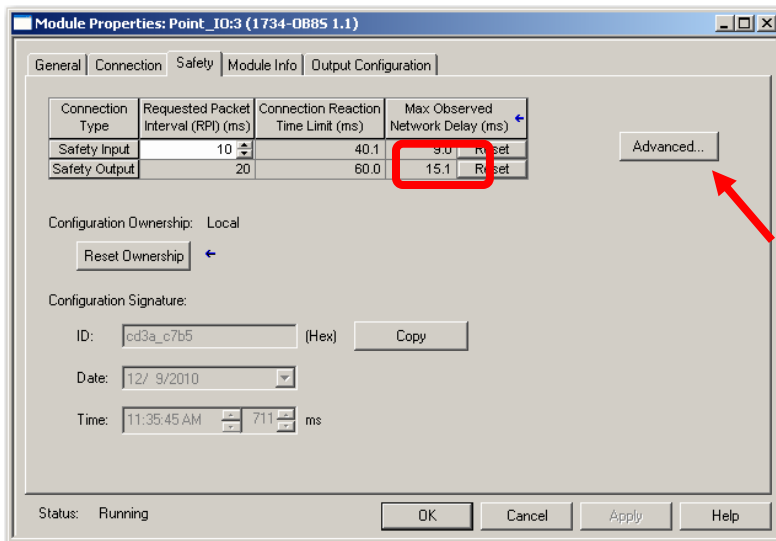
1. Call up the module properties for 1734-OB8S slot 3
2. Select the **Safety** Tab



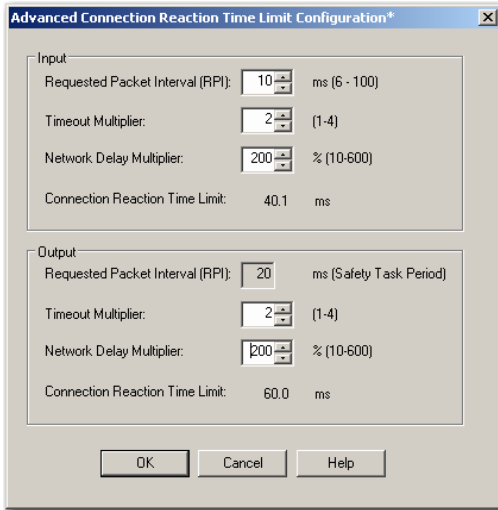
3. Click the Reset button in the Safety Output Row



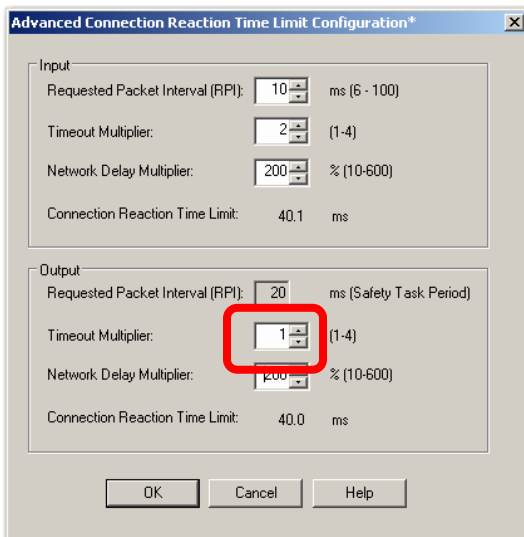
4. Wait a few seconds and note the Max Observed Network Delay  
Note the one shown below is 15.1ms



5. Click the **Advanced** button (arrow above)

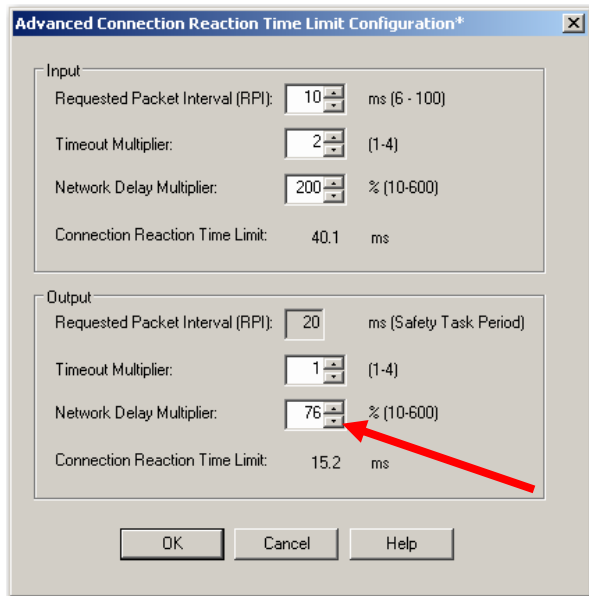


6. Change the Output Timeout Multiplier to 1



- Using the scroll down button for the Output Network Delay Multiplier (see arrow below), reduce the CRTL to the Max Observed Network Delay from Step 4 plus an additional 0.2ms.

Recall mine was 15.1ms. Adding 0.2 ms results is 15.3ms. Note that you may not be able to get the exact number. Just make it as close as possible.



### 8. OK / Apply / Yes

- Cycle the Flashing red circuit reset button
- Press the Flashing Green reset button.
- Press the flashing yellow motion start button. Wait for the motor to stop.

By reducing the CRTL to a value around the maximum observed network time, then we should be able to generate a CRTL timeout. This will cause the motor to stop because the 1734-OB8S drops out its outputs, which include the K300 Safe Torque OFF inputs.

- Change the Output Timeout Multiplier back to 2 and the Network Delay Multiplier back to 200
- OK / APPLY / YES
- Cycle the Flashing red circuit reset button
- Press the Flashing Green reset button to restart the motor
- Press the flashing yellow button to start motion
- Close the module properties window using [Cancel]

---

Notes <<Notes style>>

**[www.rockwellautomation.com](http://www.rockwellautomation.com)**

---

**Power, Control and Information Solutions Headquarters**

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication XXXX-XX###X-EN-P — Month Year  
Supersedes Publication XXXX-XX###X-EN-P — Month Year

Copyright© 2011 Rockwell Automation, Inc. All rights reserved.