Application Technique

Allen-Bradley

ControlLogix Peer I/O Control

Catalog Numbers 1756-IB16IF, 1756-LSC8XIB8I, 1756-OB16IEF

Important User Information

Solid-state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation® sales office or online at http://www.rockwellautomation.com/literature/) describes some important differences between solid-state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid-state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

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

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

Throughout this manual, when necessary, we use 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.
	**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, and recognize the consequence.
	**SHOCK HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
	**BURN HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

Notes:

This publication describes how to configure fast I/O modules for peer control applications. In a peer control operation, an output module establishes a connection to an input peer module for the purpose of energizing outputs without controller intervention.

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
ControlLogix Digital I/O Modules User Manual, publication 1756-UM058	Describes how to install and use ControlLogix® digital I/O modules.
ControlLogix Low-speed Counter Module User Manual, publication 1756-UM536	Provides details about how to install, configure, and troubleshoot the 1756-LSC8XIB8I module.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, http://www.ab.com	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at http:/www.rockwellautomation.com/literature/. To order paper copies of technical documentation, contact your local Allen-Bradley® distributor or Rockwell Automation sales representative.

Notes:

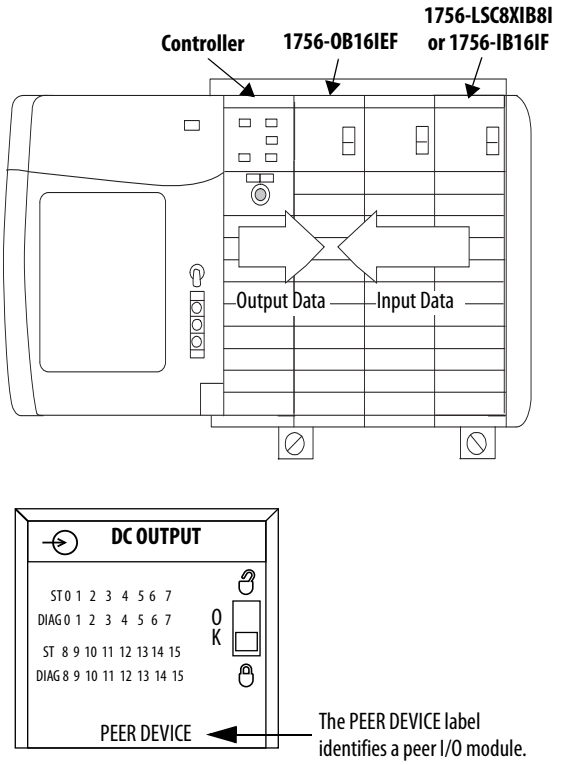
Peer Control

Peer Control Overview

In a peer control operation, a fast output module establishes a connection to an input peer module for the purpose of energizing outputs without controller intervention.

You can configure a fast output module to receive data from only a controller or both a peer input module and the controller. Figure 1 shows the output module receiving inputs from a peer input module and output data processed by the controller.

Figure 1 - Output Module Receives Data from Peer and Controller



When setting up a system for peer control, follow these guidelines:

- The input and output peer modules must be owned by the same controller.
- An output module and its input peer module must reside in the same physical chassis. However, the owner-controller is not required to be in the same chassis as the input and output peer modules.
- Each output module can have only one input peer module. However, output modules can share the same input peer module.

Peer Control Data

A 1756-OB16IEFmodule can monitor data from these modules:

- 1756-IB16IF—The output module can monitor input data from any point on the module. For fastest peer-to-peer operation, enable COS on monitored input points.
- 1756-LSC8XIB8I—The output module can monitor hardware transitions and counts from the counter module. The counter module produces data immediately upon any transition of its hardware inputs or any time an accumulated count enters or exits a user-defined window.
- Any owner-controller—An output module can use output data from a controller as a permissive along with data from an input peer.

IMPORTANT	Permissive data from a controller that does not reside in the same physical chassis as the peer modules may slow performance.

Once a peer I/O connection is established, the output module automatically listens for the data you specify in the output module's configuration. For more information, refer to Configure Modules for Peer Control on page 15.

Peer-to-Peer Response Time

Peer-to-peer response time varies depending on the number and type of modules in the same chassis as the peer modules. You can expect an input screw-to-output screw response time of 50 µs nominal...150 µs maximum if the modules used in the same chassis are restricted to the following:

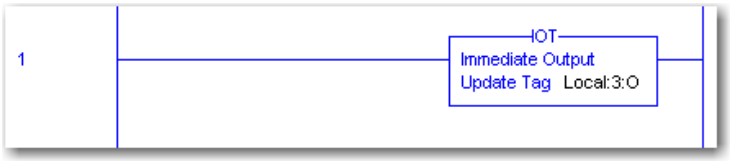
- A 1756-L7*x* controller
- One or more of these communication modules:
  - 1756-EN2T/C
  - 1756-EN2TR/B
  - 1756-EN2F/B
  - 1756-EN2TXT/C
  - 1756-EN3TR/A
  - Any of the above types in a later series
- Any 1756 I/O modules

IMPORTANT If any other type of modules are used in the same chassis as the peer modules, the response time may slow to 500 µs nominal. . . 1 ms maximum.

IMPORTANT In some versions of RSLogix™ 5000 software, output tag information is sent to peer modules only at the RPI rate defined during configuration. For optimal performance, use an Immediate Output (IOT) instruction:

- For the 1756-IB16IF and 1756-OB16IEF modules, use an IOT instruction in RSLogix 5000 software, versions 18.02.00 and 19.01.00.
- For the 1756-LSC8XIB8I module, use an IOT instruction in RSLogix 5000 software, versions 18.02.00 and later.

For example, the rung shown below contains an IOT instruction for an input module in slot 3. Add a similar rung to your last routine within the Main Task to mimic normal output tag processing.



Logic Restrictions

Possible output logic combinations depend on whether you configure logic via the Output Configuration tab of the Module Properties dialog box (Figure 2) or manually in the Output Map configuration tags (Figure 3):

- Use the Output Configuration tab to define up to three operands per output.
- Use the Output Map configuration tags to define four or more operands per output.

Figure 2 - Output Configuration Tab

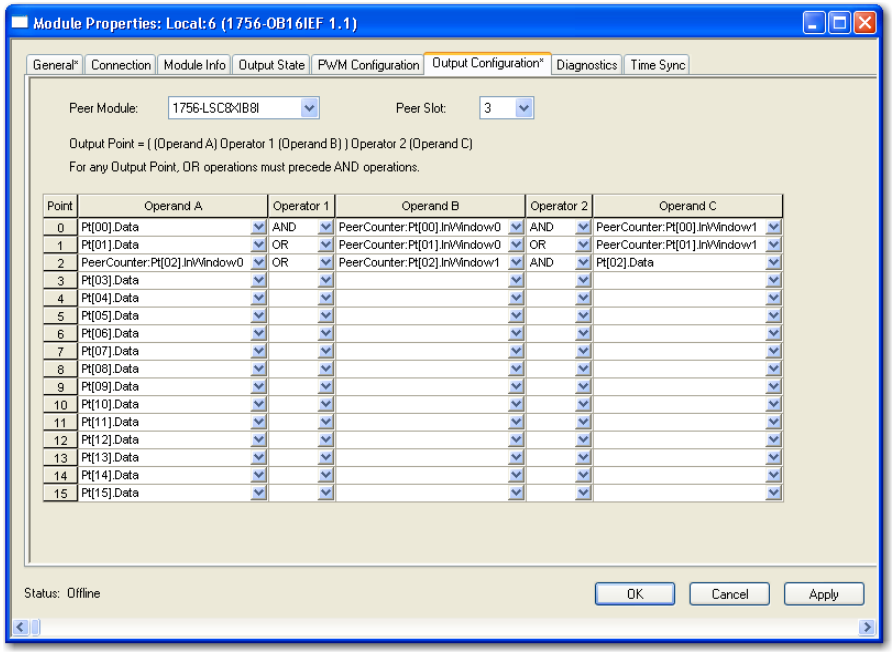
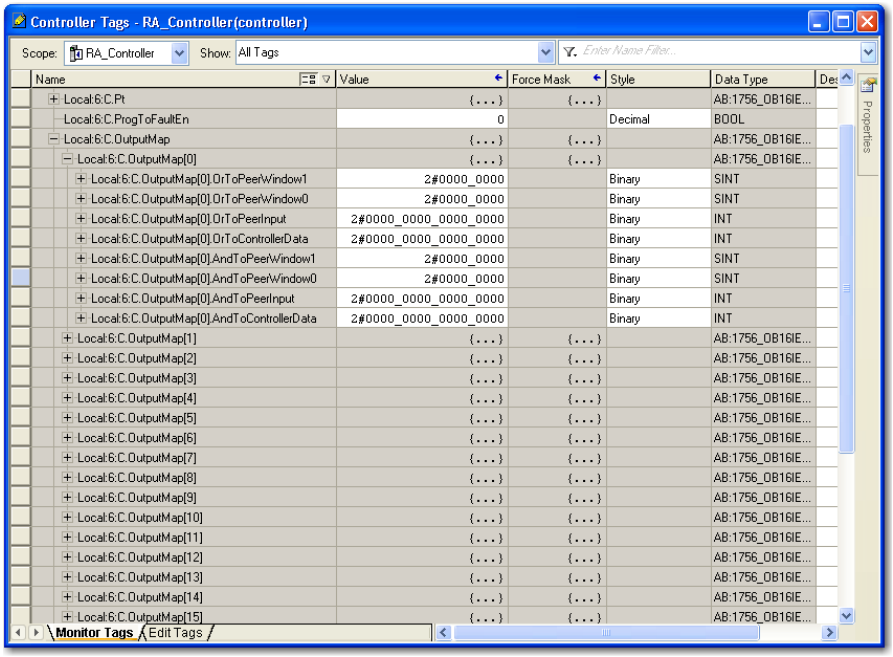


Figure 3 - Output Map Configuration Tags

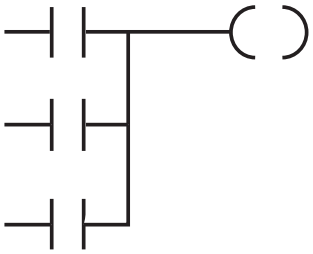


Output Logic via the Output Configuration Tab

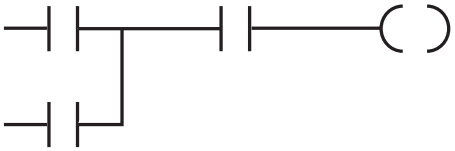
Output logic defined on the Output Configuration tab of the Module Properties dialog box is limited to the types of combinations below. Ladder logic representations are provided for illustration purposes only:

- AND logic applied to up to three peer input or controller operands

- OR logic applied to up to three peer input or controller operands



- OR logic applied to two peer input or controller operands followed by AND logic applied to one peer input or controller operand



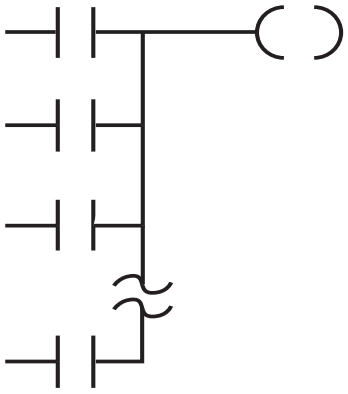
IMPORTANT	When using OR and AND operators in the same expression on the Output Configuration tab (Figure 2), Operator 1 must be OR and Operator 2 must be AND.

Output Logic via the Output Map Configuration Tags

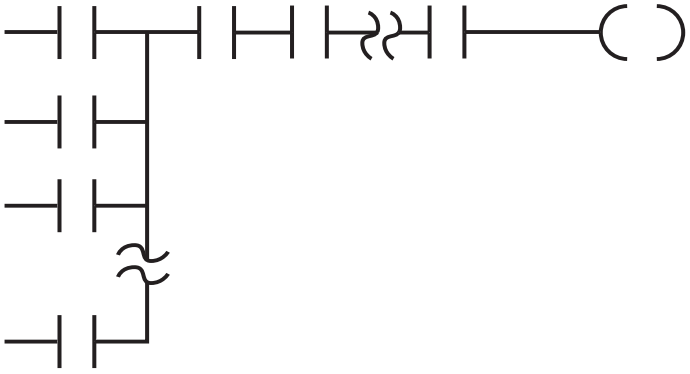
Output logic defined in the Output Map configuration tags can use the types of combinations below. Ladder logic representations are shown for illustration purposes only:

- AND logic applied to four or more peer input and controller operands

- OR logic applied to four or more peer input and controller operands



- OR logic applied to four or more peer input or controller operands combined with AND logic applied to four or more peer input or controller operands



Configure Peer Control

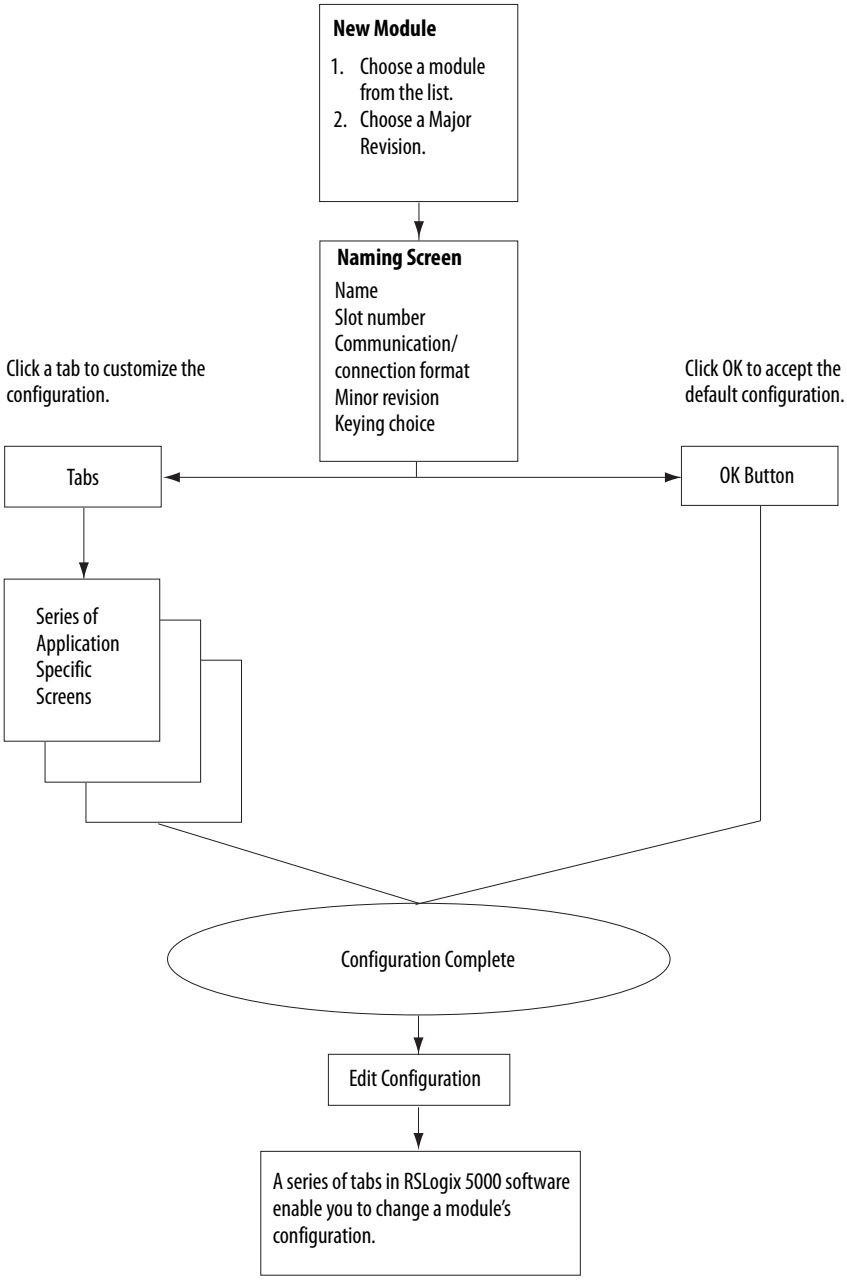
Configuration Process Overview

Follow these steps to configure a ControlLogix digital I/O module with RSLogix 5000 software.

1. Create a new module.
2. Customize the default configuration for the module for a peer ownership connection.
3. Edit the configuration as changes are needed.

Figure 4 on page 14 illustrates the configuration workflow. For detailed instructions, refer to the ControlLogix Digital I/O Modules User Manual, publication 1756-UM058.

Figure 4 - Full Configuration Profile Diagram



Configure Modules for Peer Control

By default, a fast output module behaves like any other 1756 output module receiving data for controlling its outputs from only the owner-controller. To use a fast output module in a peer control application, you must configure the module to consume input data from a peer module in RSLogix 5000 software.

IMPORTANT To configure the module, you must have the following:

- RSLogix 5000 software, version 18.02.00 or later
- The Add-on Profile (AOP) for the module available for download at http://support.rockwellautomation.com/controlflash/LogixProfiler.asp

You configure a fast output module for peer control by first defining a Peer Ownership connection and then specifying this information on the Output Configuration tab of the Module Properties dialog box:

- The type and location of the peer input module in the local chassis

  IMPORTANT An output module and its input peer module must reside in the same physical chassis. However, the owner-controller is not required to be in the same chassis as the input and output peer modules.

- The Boolean logic to apply to data consumed from the peer input module and the owner-controller

  IMPORTANT You are limited to defining up to three Boolean operands on the Output Configuration tab. To define more than three operands, use the output module's configuration tags as described on page 27.

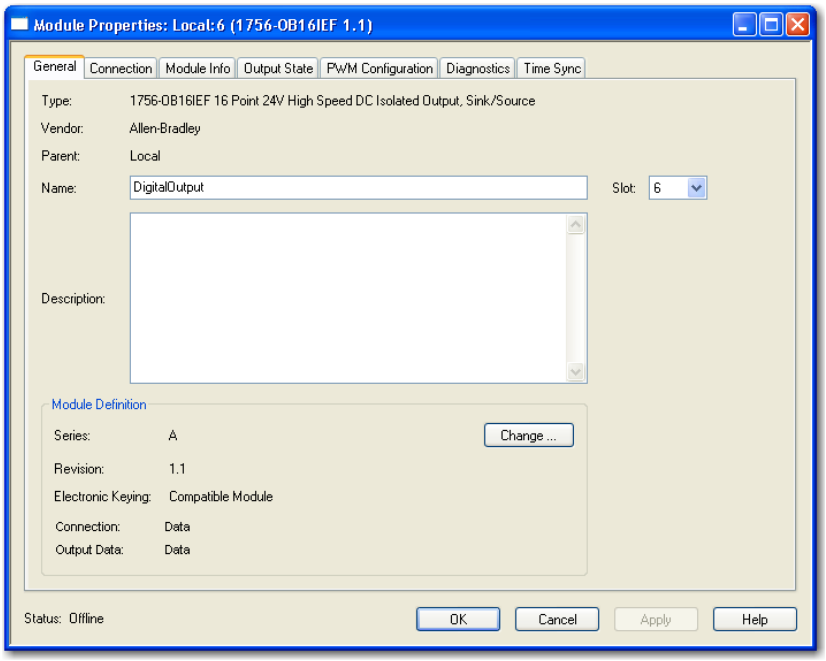
Follow these steps to configure fast I/O modules for peer control.

1. Add the modules to the I/O configuration tree in RSLogix 5000 software.
2. Configure each input peer module to define the preconditions for energizing outputs:
   - To configure the 1756-IB16IF module, refer to the ControlLogix Digital I/O User Manual, publication 1756-UM058. For the best peer control performance, configure inputs to be monitored by the output module for COS.
   - To configure the 1756-LSC8XIB8I module, refer to the ControlLogix 1756-LSC8XIB8I Counter Module User Manual, publication 1756-UM536. A counter module immediately produces data upon any transition of its hardware inputs and any time an accumulated count enters or exits a user-defined window.

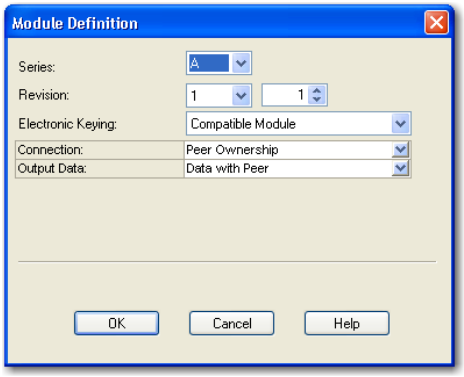
   IMPORTANT If the peer module pair is in a remote chassis using an EtherNet/IP network, make sure the Use Unicast Connection over EtherNet/IP checkbox on the input peer module's Connection tab is cleared.

3. Open the Module Properties dialog box for the output module.

4. In the Module Definition area of the General tab, click Change.

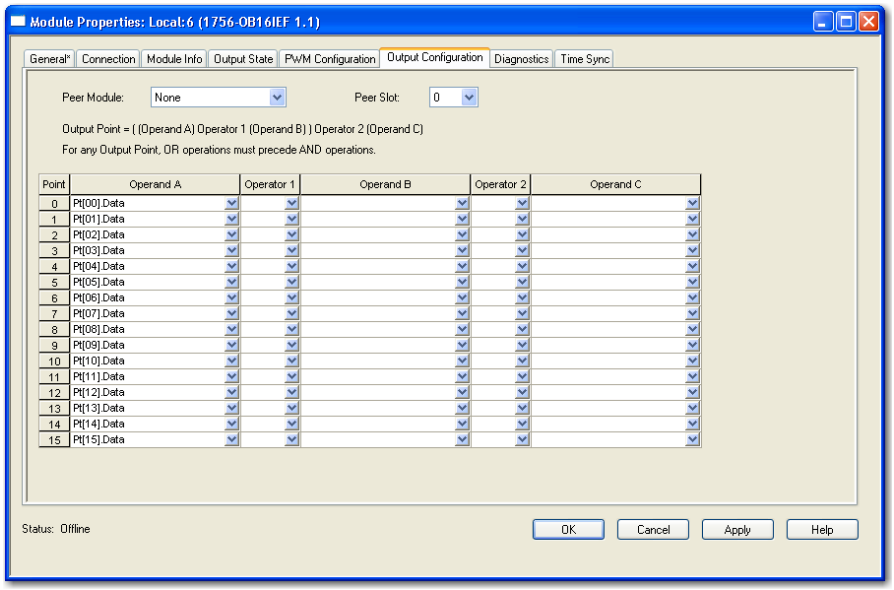


5. Define the module's connection as follows and click OK:
   - From the Connection pull-down menu, choose Peer Ownership.
   - From the Output Data pull-down menu, choose Data with Peer.



6. Click the Output Configuration tab.

   The Output Configuration tab only appears when the module is using a Peer Ownership connection type as defined in step 5.



7. From the Peer Module pull-down menu, choose the type of the peer module from which the output module will consume data.
8. From the Peer Slot pull-down menu, choose the slot number of the peer module.
9. In the grid, configure Boolean logic for each output point and click OK.

IMPORTANT When using OR and AND operators in the same expression on the Output Configuration tab, Operator 1 must be OR and Operator 2 must be AND.

Operand	Data Source
Pt[*x*].Data	1756-OB16IEF output data from the controller (O:Pt[*x*].Data) to be used as a permissive
Peer:Pt[*x*].Data	1756-IB16IF inputs (I:Pt[*x*].Data) **or** 1756-LSC8XIB8I hardware inputs (I:Pt[*x*].Data)
PeerCounter:Pt[*x*].InWindow0	1756-LSC8XIB8I window 0 (I:Counter[*x*].InputWindow0)
PeerCounter:Pt[*x*].InWindow1	1756-LSC8XIB8I window 1 (I:Counter[x].InputWindow1)

IMPORTANT Operands on the Output Configuration tab are evaluated for an On condition only.

Once you apply the configuration or navigate to another tab, the operands on the Output Configuration are reordered to match the order of execution.

Operands using all OR logic or all AND logic are reordered as follows.

Pt[00].Data...Pt[15].Data, Peer:Pt[00].Data...Peer:Pt[15]Data, PeerCounter:Pt[00].InWindow0...PeerCounter:Pt[07].InWindow0, PeerCounter:Pt[00].InWindow1...PeerCounter:Pt[07].InWindow1

EXAMPLE

You enter the following:

Pt[07].Data OR Pt[15].Data OR Pt[00].Data

When you click Apply or navigate to a different tab, the operands are reordered as follows:

Pt[00].Data OR Pt[07].Data OR Pt[15].Data

EXAMPLE

You enter the following:

Pt[07].Data AND Pt[15].Data AND Pt[00].Data

When you click Apply or navigate to a different tab, the operands are reordered as follows:

Pt[00].Data AND Pt[07].Data AND Pt[15].Data

EXAMPLE

You enter the following:

Peer:Pt[15].Data OR PeerCounter:Pt[07].InWindow1 OR Pt[07].Data

When you click Apply or navigate to a different tab, the operands are reordered as follows:

Pt[07].Data OR Peer:Pt[15].Data OR PeerCounter:Pt[07].InWindow1

When OR and AND operators are used in the same expression, Operator 1 must be OR and operator 2 must be AND. As a result, only the operands using OR logic are reordered as follows.

Pt[00].Data...Pt[15].Data, Peer:Pt[00].Data...Peer:Pt[15]Data, PeerCounter:Pt[00].InWindow0...PeerCounter:Pt[07].InWindow0, PeerCounter:Pt[00].InWindow1...PeerCounter:Pt[07].InWindow1

The operand using AND logic is not reordered, as shown in the example below.

EXAMPLE

You enter the following:

Pt[15].Data OR Pt[13].Data AND Pt[11].Data

When you click Apply or navigate to a different tab, the operands are reordered as follows:

Pt[13].Data OR Pt[15].Data AND Pt[11].Data

Peer Control Examples

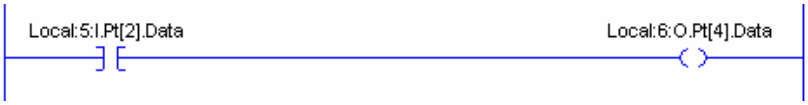
This section provides three examples on how to configure peer control:

- Use the Module Properties dialog box to configure a 1756-OB16IEF module to receive inputs from a 1756-IB16IF module.
- Use the Module Properties dialog box to configure a 1756-OB16IEF module to receive inputs from a 1756-LSC8XIB8I module and a permissive bit from the controller.
- Use configuration tags to configure a 1756-OB16IEF module to receive inputs from a 1756-LSC8XIB8I module and a permissive bit from the controller.

Output Module Receives Inputs from a 1756-IB16IF Module

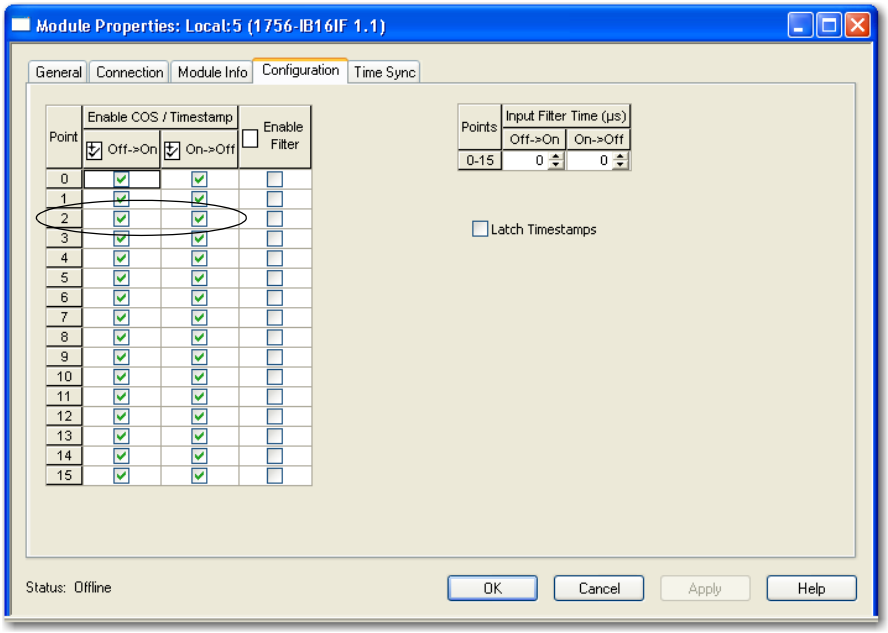
In this example, the output module energizes output point 4 when input point 2 of the peer input module is On.

The ladder logic representation of what will be accomplished via peer control is shown below. This rung is not required in program logic. It is for illustration purposes only.

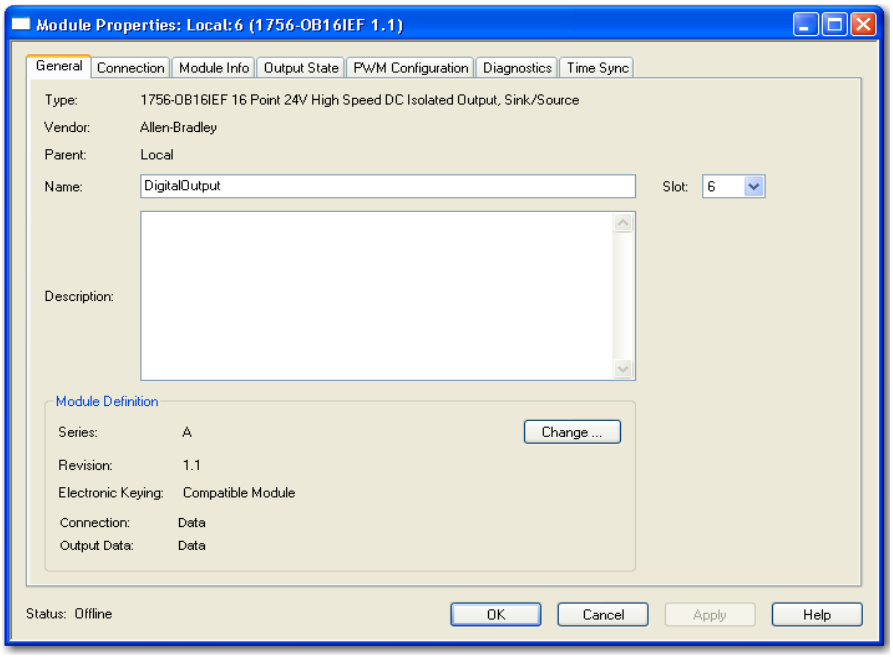


Use these steps to configure the above example in RSLogix 5000 software.

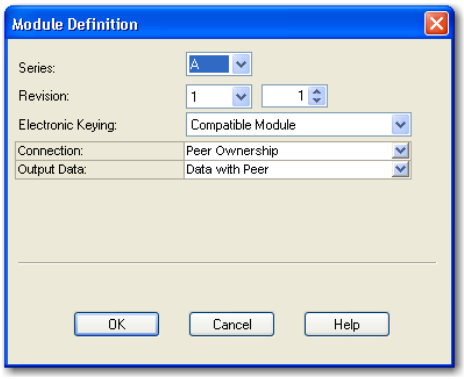
1. Add the peer modules to the I/O configuration tree.
2. Verify that point 2 on the 1756-IB16IF module is configured for COS.



3. Configure the output module.
   a. On the General tab, click Change.



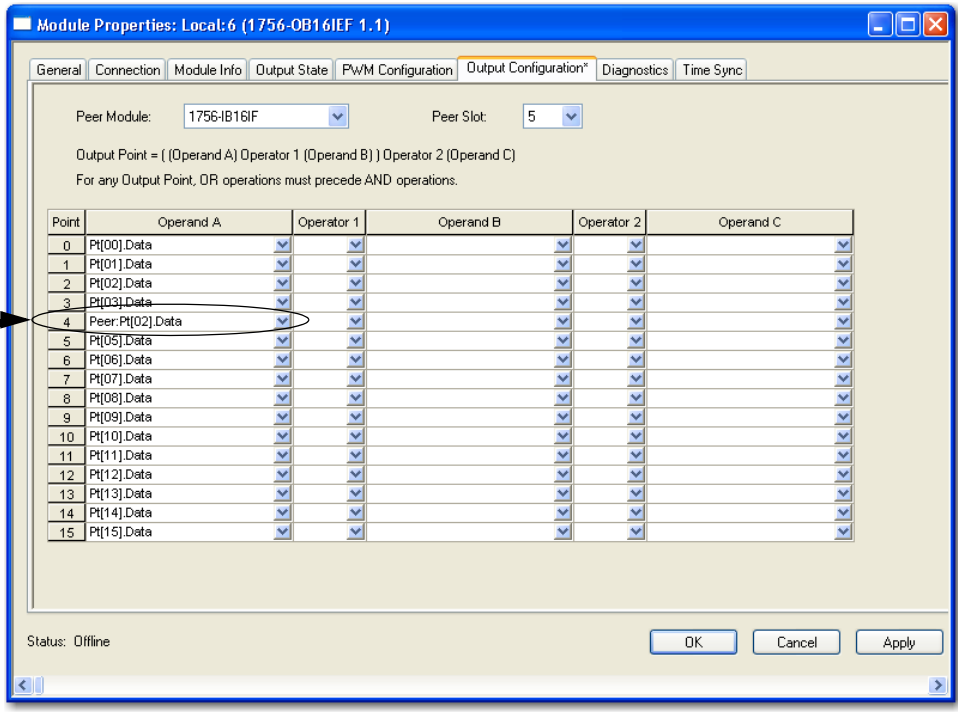
   b. From the Connection pull-down menu, choose Peer Ownership.
   c. From the Output Data pull-down menu, choose Data with Peer.
   d. Click OK.



e. Click the Output Configuration tab, choose the values in the table below, and click OK.

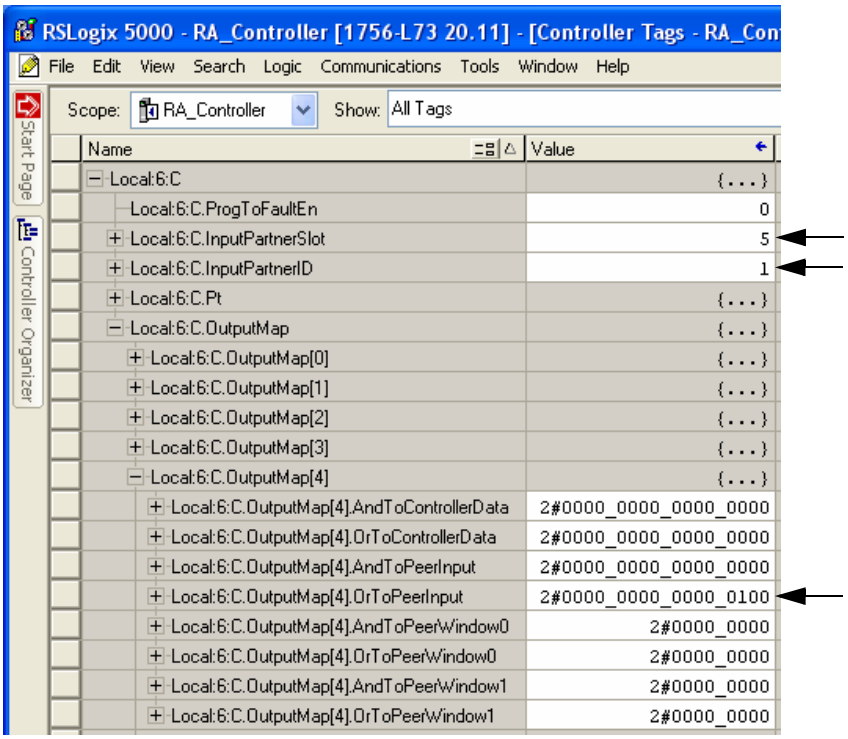
Field	Value
Peer Module	1756-IB16IF
Peer Slot	5
Point 4: Operand A	Peer:Pt[02].Data

Input point 2 on the peer module is mapped to point 4 on the output module.



The preceding configuration sets the following values in the configuration tags.

Configuration Tag	Value
InputPartnerSlot	5
InputPartnerID	1 1 = 1756-IB16IF 2 = 1756-LSC8XIB8I
OutputMap[4].OrToPeerInput	Bit 2 is set. Bits 15 … 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0



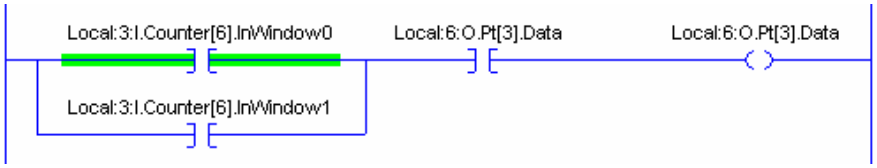
Output Module Receives Inputs from a 1756-LSC8XIB8I Module and a Permissive Bit from the Controller

In this example, the output module energizes output point 3 when these conditions are true:

- The incoming count is in window 0 or window 1 of counter 6.
- Permissive bit 3 from the controller is set.

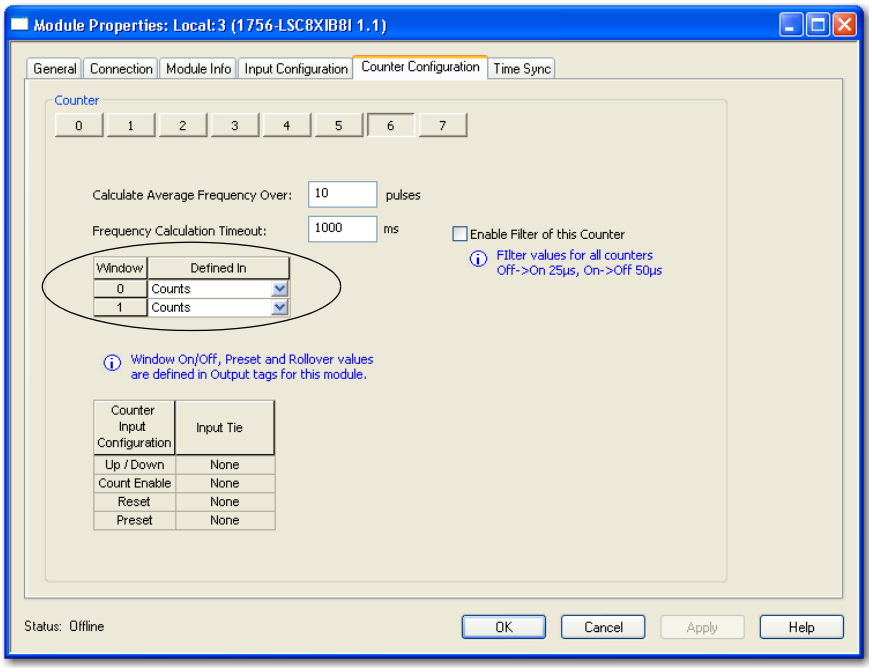
TIP We recommend that the permissive bit match the output point being configured via peer control. For example, if the output point is 3, make the permissive bit 3.

The ladder logic representation of what will be accomplished via peer control is shown below. This rung is not required in program logic. It is for illustration purposes only.



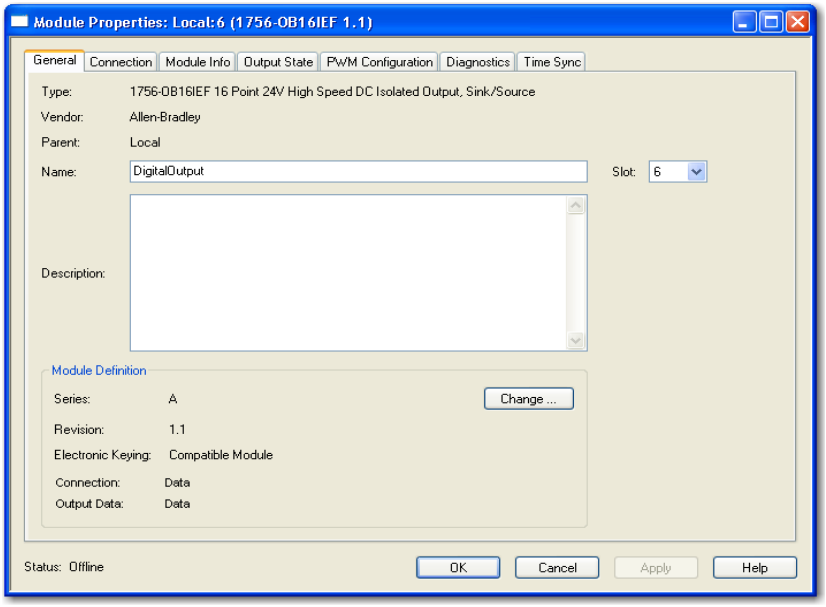
Use these steps to configure the above example in RSLogix 5000 software.

1. Add the peer modules to the I/O configuration tree.
2. Configure counter 6 of the counter module.
   a. Define both windows for counter 6 to use accumulated count.

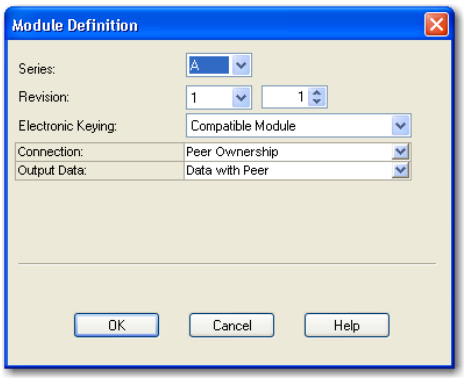


   b. In the output tags, define On/Off parameters for both windows in the Counter[6].Window0On, Counter[6].Window0Off, Counter[6].Window1On and Counter[6].Window1Off tags.

3. Configure the output module.
   a. On the General tab, click Change.



   b. From the Connection pull-down menu, choose Peer Ownership.
   c. From the Output Data pull-down menu, choose Data with Peer.
   d. Click OK.

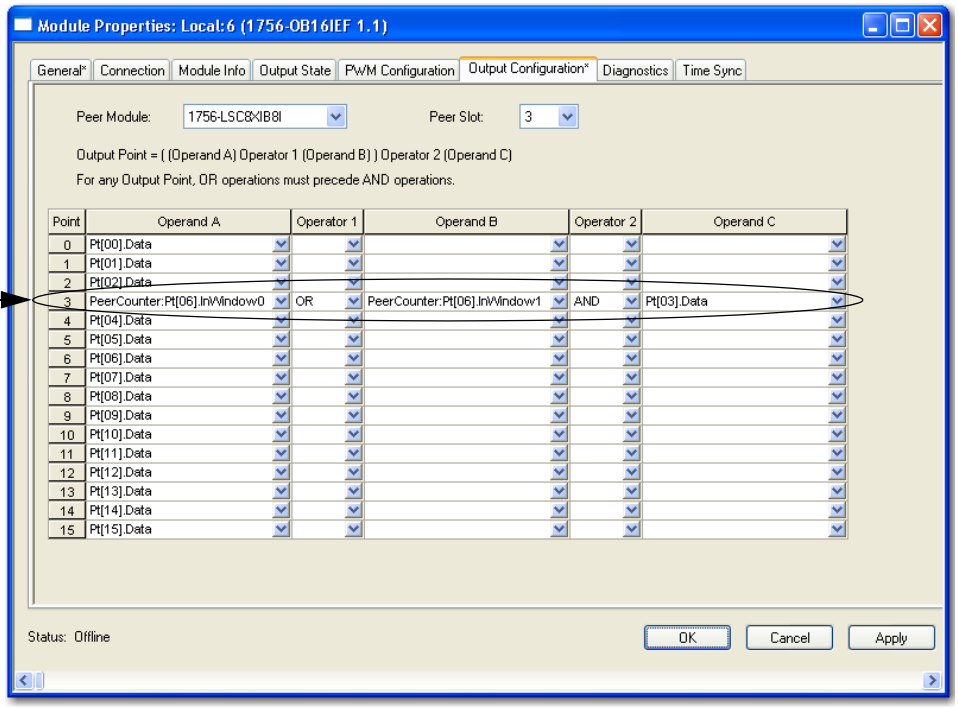


e. Click the Output Configuration tab, choose the values in the table below, and click OK.

Field	Value
Peer Module	1756-LSC8XIB8I
Peer Slot	3
Point 3: Operand A	PeerCounter.Pt[06].InWindow0
Point 3: Operator 1	OR
Point 3: Operand B	PeerCounter.Pt[06].InWindow1
Point 3: Operator 2	AND
Point 3:Operand C	Pt[3].Data

IMPORTANT When using OR and AND operators in the same expression on the Output Configuration tab, Operator 1 must be OR.

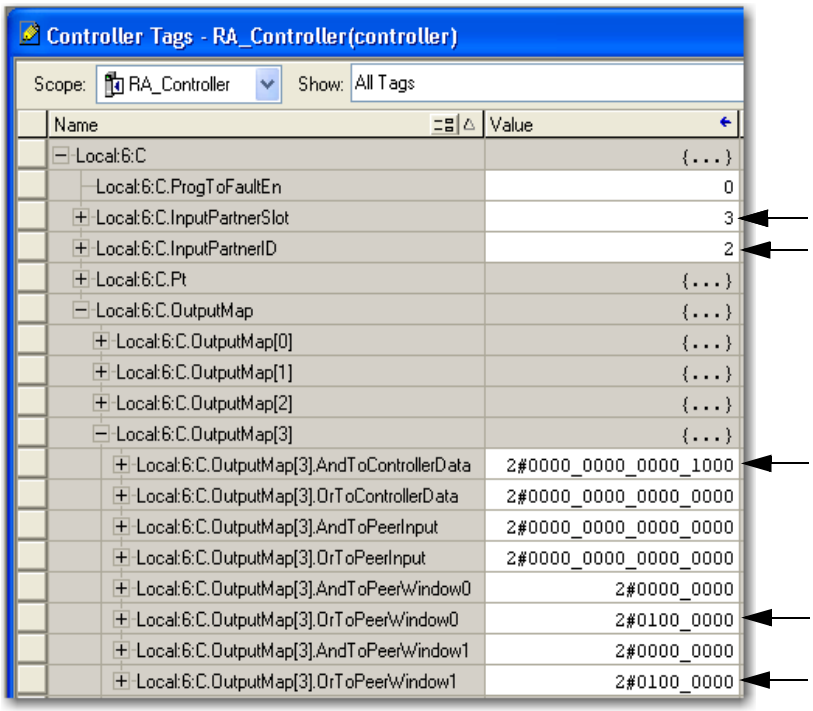
Both windows for counter 6 on the peer module and bit 3 from the controller are mapped to point 3 on the output module.



When a permissive bit from the controller is used in Boolean output logic, the permissive bit represents a set of preconditions defined in the controller program logic. The permissive bit ensures that the Boolean output logic on the 1756-OB16IEF module will not execute until the preconditions defined in the controller program logic are true. Both the input peer preconditions (the InWindow bits in the preceding example) and the controller program logic must be true before the output will energize. Make sure the permissive bit is the same as the output point being defined for peer control. In the example above, permissive bit 3 is used in the Boolean output logic for output point 3.

The preceding configuration sets these values in the configuration tags.

Configuration Tag	Value
InputPartnerSlot	3
InputPartnerID	2 1 = 1756-IB16IF 2 = 1756-LSC8XIB8I
OutputMap[3].AndToControllerData	Bit 3 is set. Bits 15 … 0 1
OutputMap[3].OrToPeerWindow0	Bit 6 is set. Bits 7 … 0 1
OutputMap[3].OrToPeerWindow1	Bit 6 is set. Bits 7 … 0 1



Define Output Logic via Tags

Applications that involve more than three Boolean operands require you to configure the output logic directly in the OutputMap configuration tags of the 1756-OB16IEF module.

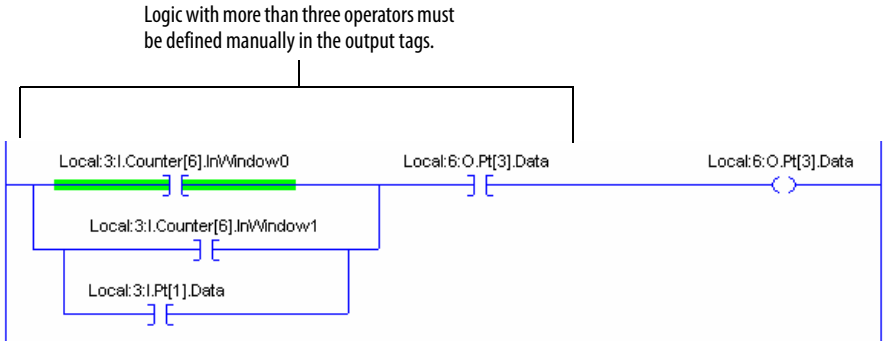
In this example, the output module receives inputs from the counter module and the controller and energizes output point 3 when these conditions are true:

- The incoming count is in window 0 or window 1 of counter 6.

  or

  Hardware input 1 is On.
- Permissive bit 3 from the controller is set.

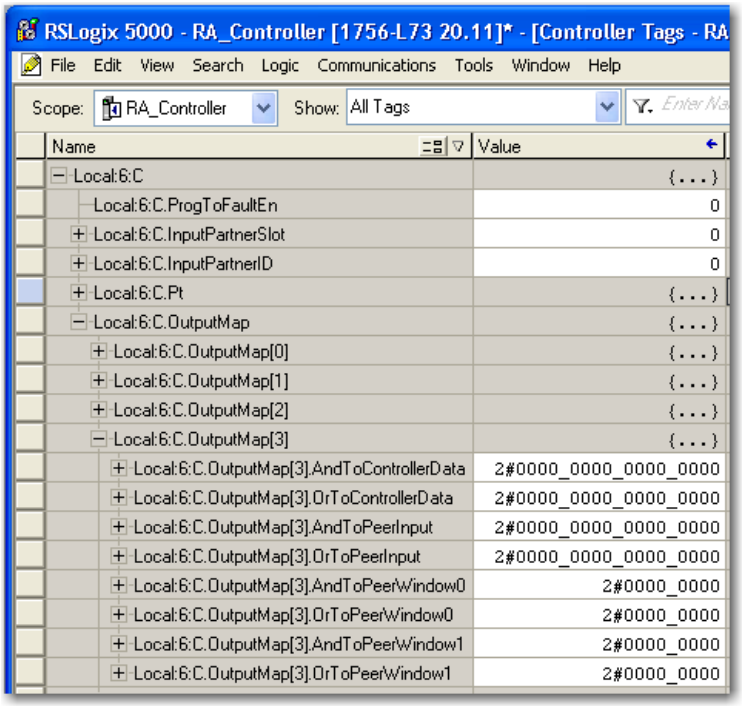
The ladder logic representation of what will be accomplished via peer control is shown below. This rung is not required in program logic. It is for illustration purposes only.



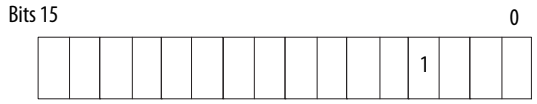
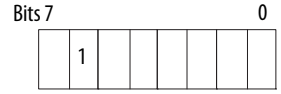
Use these steps to configure the above example in RSLogix 5000 software.

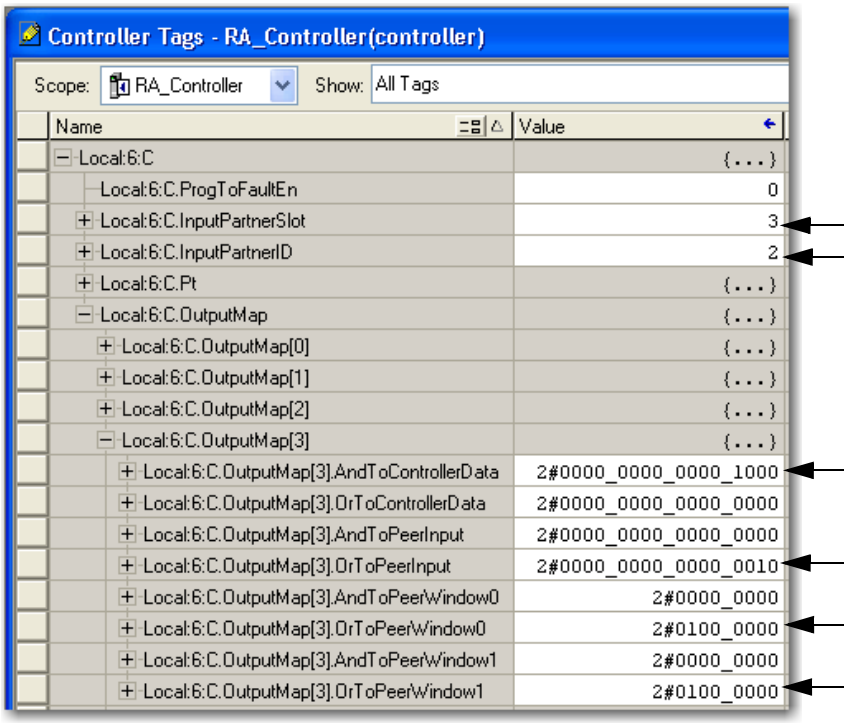
IMPORTANT Configure output logic via the configuration tags prior to downloading the configuration to the module. Configuration changes made directly via the configuration tags when online are **not** automatically downloaded to the module.

1. Open the tag editor and expand the configuration tags for the output module in slot 6 as shown below.



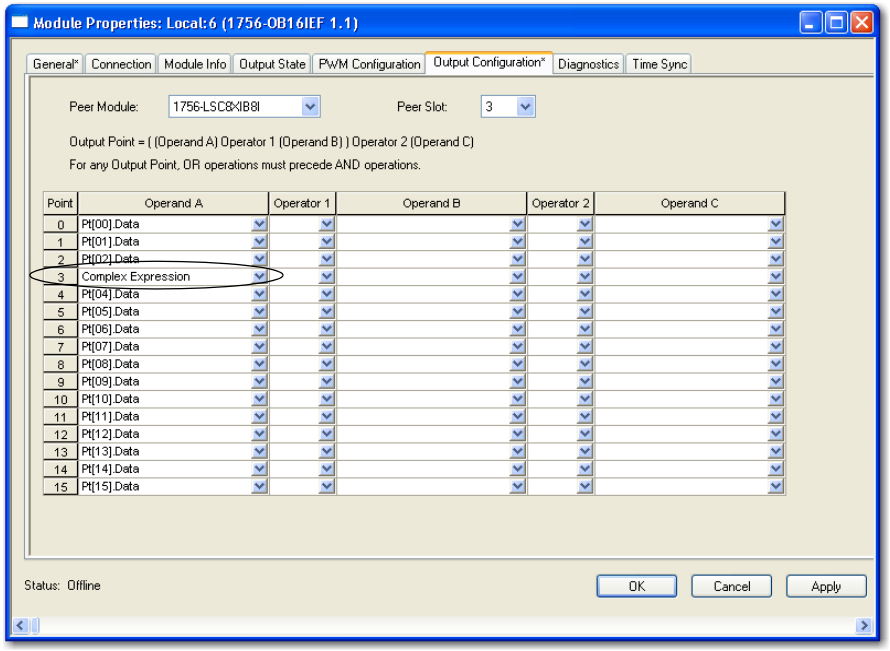
2. Define the following tag values and click Save.

Configuration Tag	Value
InputPartnerSlot	3
InputPartnerID	2 1 = 1756-IB16IF 2 = 1756-LSC8XIB8I
OutputMap[3].AndToControllerData	Set bit 3. This setting defines the AND logic to apply to the permissive data from the controller (O:Pt[3].Data). 
OutputMap[3].OrToPeerInput	Set bit 1. This setting defines the OR logic to apply to hardware input 1 of the counter module (I:Pt[1].Data). 
OutputMap[3].OrToPeerWindow0	Set bit 6. This setting defines the OR logic to apply to window 0 of counter 6 (I:Counter[6].InputWindow0). 
OutputMap[3].OrToPeerWindow1	Set bit 6. This setting defines the OR logic to apply to window 1 of counter 6 (I:Counter[6].InputWindow1). 



3. On the Output Configuration tab of the Module Properties dialog box, verify that the Complex Expression value is assigned to the output point.

IMPORTANT Even though your manual configuration is recognized on the Output Configuration tab, configuration changes made directly via the configuration tags when online are **not** automatically downloaded to the module.



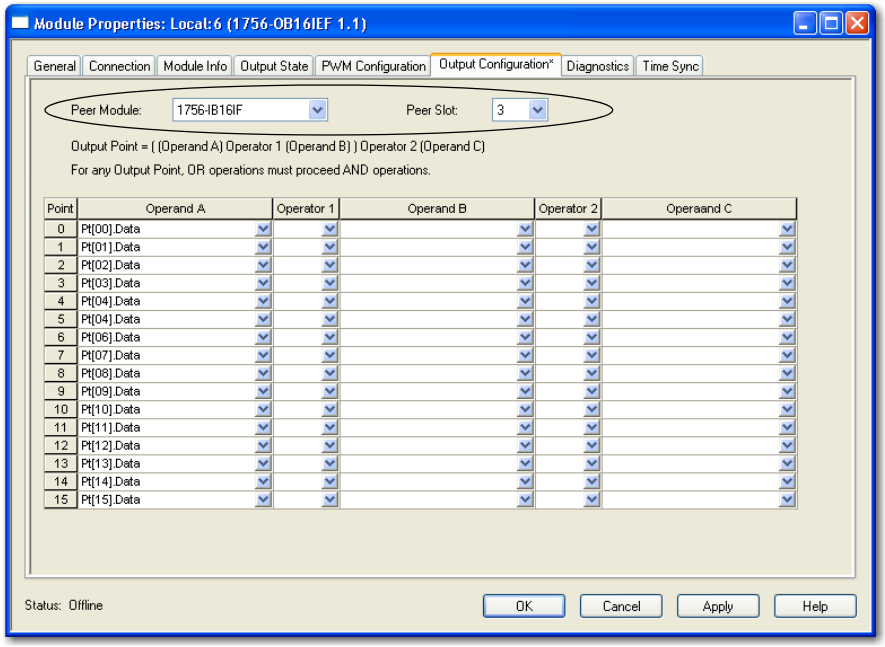
Troubleshoot Peer Modules

Peer I/O Connections

A connection is established between a1756-OB16IEF module and a peer input module when these conditions are met:

- A peer input module is identified on the Output Configuration tab for the 1756-OB16IEF module (Figure 5). No output logic is required to initiate the connection between peer I/O modules.
- The input and output peer modules are each owned by the same controller.

Figure 5 - Output Configuration Tab



The output module establishes a connection to its peer input partner prior to responding to its own connection request from the controller. If the output module cannot connect to the input peer, a peer connection error appears in the Module Fault area of the Connection tab. Table 1 describes possible errors.

Table 1 - Peer Connection Errors

Error Code	Description
0x323	The peer module is currently being updated.
0x324	The peer module has a unicast connection only. The owner-controller must use a multicast connection to the peer to allow peer operation.
0x325	Peer keying error. Invalid product code.
0x326	Peer keying error. Invalid product type.
0x326	Peer keying error. Incompatible revision.
0x328	The peer module is not owned by a controller.
0x329	The peer module connection request has timed out.
0x32A	Unable to connect to the peer module.

If any connection to the output module is lost, either from the input module or the controller, then all outputs go into their configured Fault mode states.

If the output module goes into Fault mode or is removed from the chassis, its input peer module continues to operate and broadcast inputs.

Status Indicators

The 1756-OB16IEF output module uses the status indicators shown below. For descriptions, refer to Table 2.

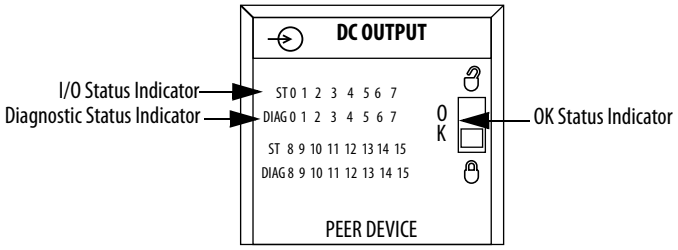


Table 2 - Status Indicators for 1756-OB16IEF Module

Indicator	Status	Description
OK	Steady green	The outputs are actively being controlled by a system processor.
	Flashing green	The module has passed internal diagnostics but is not actively controlled, is inhibited, or the controller is in Program mode. To enable communication to the module, establish a connection, uninhibit the connection, or transition the controller to Run mode.
	Flashing red	Previously established communication has timed out. Check the controller and chassis communication.
	Steady red	Replace the module.
I/O Status (ST)	Yellow	The output is On.
Diagnostic (DIAG)	Flashing red	The output is listening for peer inputs and actively using them to determine its output state. If the output is not working, refer to Troubleshoot an Output on page 35.
	Steady red	The output has encountered a fault. Check for a fuse blown condition. If the fuse is okay, replace the module.
	Off	The output bit is not configured for peer control. Troubleshoot the module as any other 1756 output module. Check program logic related to the output point, verify output tag states, and force I/O bits from the controller.

Data Override Tags

To check the behavior of an output point configured for peer control, use the data override tags provided with the 1756-OB16IEF module to override the output point itself and the operands used in Boolean output logic in a method similar to forcing I/O bits from the controller:

- Override input values to verify output logic. When you override input values, the module ignores any direct peer input data and applies only the override values.
- Override output values to verify output hardware only. When you override output values, the module ignores any transitions driven by local logic and applies only the output override value.

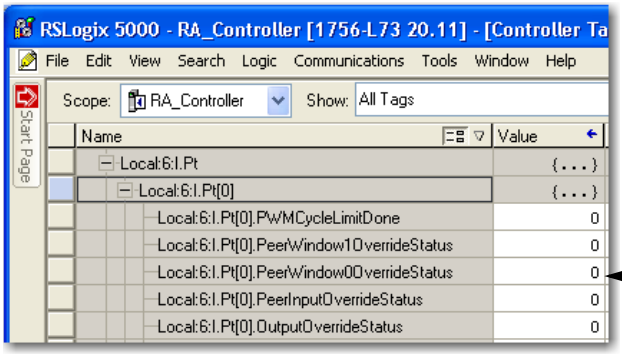
Because all override values are located in output tags, the module applies the values only when the controller is in Run mode. When the controller is not in Run mode, outputs transition to their configured Program mode or Fault mode states.

Table 3 lists the input and output tags that contain data override values, enable bits, and status bits. The status bits in the module's input tags (Figure 6) let you monitor whether data override values are in effect. For descriptions of each tag, refer to Appendix A.

Table 3 - 1756-OB16IEF Data Override Tags

Data to Override	Output Tags	Input Tags
Local output data	Pt[*x*].OverrideOutputEn Pt[*x*].OverrideOutputValue	Pt[*x*].OutputOverrideStatus
Peer input data	Pt[*x*].OverridePeerInputEn Pt[*x*].OverridePeerInputValue	Pt[*x*].PeerInputOverrideStatus
Peer window 0 inputs	Pt[*x*].OverridePeerWindow0En Pt[*x*].OverridePeerWindow0Value	Pt[*x*].PeerWindows0OverrideStatus
Peer window 1 inputs	Pt[*x*].OverridePeerWindow1En Pt[*x*].OverridePeerWindow1Value	Pt[*x*].PeerWindow1OverrideStatus

Figure 6 - Override Status Tags



Troubleshoot an Output

If an output on a peer output module is not working, check the red Diagnostic status indicator for the output. Refer to Table 2 on page 33. If the Diagnostic status indicator is blinking indicating that the module is listening for peer data, troubleshoot the module via the override tags as described below.

IMPORTANT When using override tags to troubleshoot a module, place the controller in Run mode. The controller cannot apply override values in Program mode.

1. Check the viability of the output hardware via the local output override tags.
   a. Set the Pt[*x*].OverrideOutputEn output bit.
   b. Toggle the Pt[*x*].OverrideOutputValue output bit On and Off.

      This is similar to performing a local force of the output on the module.
   c. Verify that the output turns On and Off and the I/O status indicator reflects its state.
   d. If the output turns On and Off at the screw, reset the Pt[*x*].OverrideOutputEn output bit to zero and proceed to step 2.

      or
   e. If the output does **not** turn On and Off at the screw, check whether the Pt[*x*].OutputOverrideStatus input bit is set.

      The status tag indicates whether the module is receiving the override values. If the status tag is set, and the output is still not working, replace the output module.
2. Check the peer control logic execution via the peer input override tags.
   a. Set the override enable bits that correspond with the Boolean preconditions you defined during the configuration of the output.

      For example, to override all peer inputs and peer window 1 bits for output point 2, set the enable bits for the Pt[2].OverridePeerInputEn and Pt[2].OverridePeerWindow1En output tags.
   b. Set the override value bits that correspond with the Boolean preconditions you defined during the configuration of the output.

      For example, to override all peer inputs and peer window 1 bits for output point 2, enter override values in the Pt[2].OverridePeerInputValue and Pt[2].OverridePeerWindow1Value tags. This is similar to performing a local force of the Boolean logic.
   c. If the Boolean logic includes a permissive bit from the controller, force the bit On from the controller.
   d. If the output turns On, reset the Pt[*x*].OverridePeerInputEn and Pt[*x*].OverridePeerWindow[*x*]En output tags.

      or

      If the output does **not** turn On, check whether the corresponding Pt[*x*].PeerInputOverrideStatus, Pt[*x*].PeerWindow0OverrideStatus, and Pt[*x*].PeerWindow1OverrideStatus input tags are set.

      The status tags indicate whether the module is receiving the override values. If the status tags are set, and the output is still not working, replace the output module.

Notes:

Peer Ownership Tag Definitions

Peer ownership tags are located within the ControlLogix 1756-OB16IEF output module's configuration, input, and output tags. Refer to the following tables for tag definitions.

IMPORTANT The Module Definition column in each table lists the connection type and input data type combinations that are required to create the corresponding tag.

For more information about features described in the following tags, refer to the ControlLogix Digital I/O Modules User Manual, publication 1756-UM058.

Configuration Tags

Configuration tags contain the structure of data sent from the controller to the I/O module upon powerup.

Table 4 - 1756-OB16IEF Module Configuration Tags

Name	Data Type	Tag Definition	Module Definition
ProgToFaultEn	BOOL	**Program to Fault Mode—**Enables the transition of outputs to Fault mode if a communication failure occurs in Program mode. Otherwise, outputs will remain in Program mode. See Pt[*x*].FaultMode, Pt[*x*]FaultValue Pt[*x*]ProgMode, and Pt[*x*]ProgValue. 0 = Outputs stay in Program mode if communication fails. 1 = Outputs go to Fault mode if communication fails.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
InputPartnerSlot	SINT	**Peer Partner Slot—**Identifies the slot number of the local chassis where the peer input module resides. Valid values: • 0…16 • -1 = No input module has been identified as a peer.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerID	SINT	**Peer Partner ID—**Identifies the peer input module that controls outputs on the 1756-OB16IEF module. The type of module determines the connection type of format of input data. Valid values: 0 = None (default) 1 = 1756-IB16IF 2 = 1756-LSC8XIB8I	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].FaultMode	BOOL	**Fault Mode—**Used in conjunction with the Pt[*x*].FaultValue tag to determine the state of outputs when a communication failure occurs. 0 = Uses the output value defined in the Pt[*x*].FaultValue configuration tag (default). 1 = Holds the last state of the output for the length of time defined in the Pt[*x*].FaultValueStateDuration tag. If PWM is enabled for the output point and the output is currently On, the output will continue PWM until the cycle limit is reached or a final fault state goes into effect via the Pt[*x*].FaultFinalState tag.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].FaultValue	BOOL	**Fault Value—**Defines the output value when a fault occurs. Holds the configured state of the output for the length of time defined in the Pt[*x*].FaultValueStateDuration tag. Requires the corresponding bit in the FaultMode tag to be cleared. 0 = Off 1 = On	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].FaultFinalState	BOOL	**Fault Final State—**Determines the final output state once the time in the Pt[*x*].FaultValueStateDuration tag elapses. 0 = Output turns Off once the time in the Pt[*x*].FaultValueStateDuration tag elapses, and module is still faulted. 1 = Output turns On once the time in the Pt[*x*].FaultValueStateDuration tag elapses, and module is still faulted.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].ProgMode	BOOL	**Program Mode—**Used in conjunction with the Pt[*x*].ProgValue tag to determine the state of outputs when the controller is in Program mode. 0 = Uses the output value defined in the Pt[*x*].ProgValue tag (default). 1 = Holds the last state of the output. If PWM is enabled for the output point and the output is currently On, the output will continue to use PWM until the cycle limit is reached.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].ProgValue	BOOL	**Program Value—**Defines the output state during Program mode. Requires the corresponding bit for the Pt[*x*].ProgMode tag to be cleared. 0 = The output state is Off during Program mode. 1 = The output state is On during Program mode.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer

Table 4 - 1756-OB16IEF Module Configuration Tags (continued)

Name	Data Type	Tag Definition	Module Definition
Pt[*x*].PWMEnable	BOOL	**Enable PWM—**When set, the pulse train for the output point is controlled by the current PWM configuration. 0 = PWM is disabled (default). 1 = PWM is enabled, and the output uses PWM when the output is On.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMExtendCycle	BOOL	**Extend PWM Cycle—**Determines the output behavior when the value in the Pt[*x*]PWMOnTime output tag is less than the value in the Pt[x].PWMMinimunOnTime configuration tag. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag. 0 = The duration of the pulse cycle is not extended (default). If the bit is cleared when the On time is less than the minimum On time, the output is never enabled. 1 = The duration of the pulse cycle is extended to maintain the On time to cycle time ratio while taking into account the minimum On time. **IMPORTANT:** An extension of the pulse cycle is limited to 10X the cycle time. If the requested On time is less than 1/10 of the minimum On time, the output will remain Off and the cycle will not extend.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMOnTimeInPercent	BOOL	**PWM On Time in Percent—**Determines whether PWM On time is defined as a percentage of the cycle time or is defined in seconds. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag. 0 = Defines PWM On time in seconds (default). 1 = Defines PWM On time as a percentage.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMStaggerOutput	BOOL	**Stagger PWM Outputs—**When set, minimizes the load on the power system by staggering On transitions for outputs. Otherwise, outputs turn On immediately at the start of a cycle. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag. 0 = Does not stagger output On transitions (default). Outputs will turn On immediately when the Pt[x].Data tag is set to 1 beginning the PWM cycle with a rising edge. 1 = Staggers output On transitions. All outputs configured for PWM staggering will turn On at a different intervals to minimize a possible power surge if many outputs became energized simultaneously.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMCycleLimitEnable	BOOL	**Enable PWM Cycle Limit—**Determines whether to allow only a fixed number of pulse cycles to occur. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag. 0 = Pulse cycles continue to occur until the output turns Off (default). 1 = Allows only the number of pulse cycles defined via the Pt[x].PWMCycleLimit tag to occur.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMExecuteAllCycles	BOOL	**Execute All PWM Cycles—**Determines whether to execute the number of cycles defined via the Pt[*x*].PWMCycleLimit tag regardless of the output logic. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag, and a cycle limit to be enabled via the Pt[*x*].PWMCycleLimitEnable tag. 0 = The output logic determines the number of cycles to produce (default). 1 = The Pt[*x*].PWMCycleLimit tag determines the number of cycles to produce regardless of output logic. For example, if you specify a cycle limit of 4, and the output turns Off after 3 cycles, all 4 cycles will still occur despite the output being instructed to turn Off.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].FaultValueStateDuration	SINT	**Fault State Duration—**Defines the length of time that the output state remains in the Fault mode state before transitioning to a final state of On or Off. The Fault mode state is defined in the Pt[x].FaultValue tag. Valid values: • 0 = Hold forever (default). Output remains in Fault mode for as long as the fault condition persists. • 1, 2, 5, or 10 seconds	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer

Table 4 - 1756-OB16IEF Module Configuration Tags (continued)

Name	Data Type	Tag Definition	Module Definition
Pt[*x*].PWMCycleLimit	SINT	**PWM Cycle Limit—**Defines the number of pulse cycles to occur when the output turns On: • If the corresponding bit in the Pt[*x*].PWMExecuteAllCycles tag is set, the configured number of cycles will occur even if the output turns Off. • If the corresponding bit in the Pt[*x*].PWMExecuteAllCycles tag is cleared, the configured number of cycles will occur only if the output remains On. For example, if the cycle limit is 4, and the output turns Off after 3 cycles, the 4th cycle will **not** occur. The default cycle limit is 10. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag, and cycle limits to be enabled via the Pt[x].PWMCycleLimitEnable tag.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMMinimumOnTime	REAL	**PWM Minimum On Time—**Defines the minimum length of time required for the output to turn On. Requires PWM to be enabled via the Pt[*x*].PWMEnable tag. Valid values: 0.0002…3600.0 seconds or 0…100.0 percent	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].AndToControllerData	INT	**Controller Data with AND Logic—**Determines the output state by applying AND logic to these sources: • Corresponding bits from the controller's output data (O:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].OrToControllerData	INT	**Controller Data with OR Logic—**Determines the output state by applying OR logic to these sources: • Corresponding bits from the controller's output data (O:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].AndToPeerInput	INT	**Peer Data with AND Logic—**Determines the output state by applying AND logic to these sources: • Corresponding bits from peer input data (I:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].OrToPeerInput	INT	**Peer Data with OR Logic—**Determines the output state by applying OR logic to these sources: • Corresponding bits from peer input data (I:Data) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].AndToPeerWindow0	SINT	**Peer Data with AND Logic—**Determines the output state by applying AND logic to these sources: • Corresponding bits from window 0 of the peer counter module (I:Counter[*x*].InputWindow0) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].OrToPeerWindow0	SINT	**Peer Data with OR Logic—**Determines the output state by applying OR logic to these sources: • Corresponding bits from window 0 of the peer counter module (I:Counter[*x*].InputWindow0) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].AndToPeerWindow1	SINT	**Peer Data with AND Logic—**Determines the output state by applying AND logic to these sources: • Corresponding bits from window 1 of the peer counter module (I:Counter[*x*].InputWindow1) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer
OutputMap[*x*].OrToPeerWindow1	SINT	**Peer Data with OR Logic—**Determines the output state by applying OR logic to these sources: • Corresponding bits from window 1 of the peer counter module (I:Counter[*x*].InputWindow1) • Other mapped bits specified in the output configuration	Connection = Peer Ownership Output Data = Data with Peer

Input Tags

Input tags contain the structure of data continually sent from the I/O module to the controller or peer input module. Input tags indicate the current operational status of the module.

Table 5 - 1756-OB16IEF Module Input Data Tags

Name	Data Type	Tag Definition	Module Definition
Fault	DINT	**Fault Status—**Indicates whether a point is faulted. If communication to the output module is lost, then all 32 bits of the Module Fault word will be set. 0 = No fault 1 = Fault	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
InputPartnerActive	BOOL	**Input Partner is Active—**Indicates whether the peer input module is actively producing input data to be consumed by a 1756-OB16IEF module. 0 = No input peer module is currently producing input data to be consumed by a 1756-OB16IEF module. 1 = The input peer module is actively producing input data to be consumed by a 1756-OB16IEF module for use in its peer logic.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerFault	BOOL	**Input Partner Fault—**Indicates whether the peer input module has faulted due to a connection loss. If the peer input module is faulted, the output module uses only controller data to determine the output state. 0 = The input peer module has not faulted. 1 = The input peer module has faulted and outputs will transition to the configured Fault mode state.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerSlot	SINT	**Input Partner Slot—**Indicates the slot number of the peer input module. Valid values: • 0…16 • -1 = No peer input module is defined.	Connection = Peer Ownership Output Data = Data with Peer
InputPartnerStatus	SINT	**Input Partner Status—**Indicates the status of the peer input module. Valid values: 2 = Communication Fault (Peer connection is lost) 6 = Run (Peer connection open and in Run mode)	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].Data	BOOL	**Data**—Indicates the current value to be sent to the corresponding output point. If PWM is enabled, this value will transition from 0 to 1 based on the PWM pulse train. 0 = Off 1 = On	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].Fault	BOOL	**Fault—**Indicates whether I/O data for the corresponding point may be incorrect due to a fault. 0 = No fault. 1 = A fault exists and I/O data may be incorrect. Check for one of the following: • Pt[x].FuseBlown = 1 • Pt[*x*].PWMCycleTime outside valid range of 0.001…3600.0 seconds • Pt[*x*].PWMOnTime outside valid range of 0.0002…3600.0 seconds or 0…100 percent • Pt[*x*].PWMCycleTime ≤ Pt[*x*].PWMOnTime	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer

Table 5 - 1756-OB16IEF Module Input Data Tags (continued)

Name	Data Type	Tag Definition	Module Definition
Pt[*x*].FuseBlown	BOOL	**Fuse is Blown—**Indicates whether a fuse has blown due to a short or overload condition for the corresponding point. All blown fuse conditions are latched and must be reset. 0 = Fuse is not blown. 1 = Fuse is blown and has not been reset.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMCycleLimitDone	BOOL	**PWM Cycle Limit Done—**Indicates whether the PWM pulse cycle limit defined in the Pt[*x*].PWMCycleLimit configuration tag has been reached. 0 = The PWM cycle limit has not yet been reached. The bit resets to 0 each time the output transitions to On to begin a new PWM cycle. 1 = The PWM cycle limit has been reached.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].CIPSyncValid	BOOL	**CIP Sync is Valid—**Indicates whether the module has synchronized to a valid CIPSync time master on the backplane. 0 = CIP Sync is not available. 1 = CIP Sync is available.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].CIPSyncTimeout	BOOL	**CIP Sync Timeout—**Indicates whether a valid time master on the backplane has timed out. 0 = A valid time master has not timed out. 1 = A valid time master was detected on the backplane, but the time master has timed out. The module is currently using its local clock.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OutputOverrideStatus	BOOL	**Output Override Status—**Indicates whether local output data or logic point is set up to be overridden by the value in the Pt[*x*].OverrideOutputValue output tag. Requires the Pt[*x*].OverrideOutputEn output tag to be enabled. 0 = The override feature for the corresponding output is not enabled. 1 = The override feature for the corresponding output is enabled.	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PeerInputOverrideStatus	BOOL	**Peer Input Override Status—**Indicates whether peer input data mapped to the corresponding output point is set up to be overridden by the value in the Pt[*x*].OverridePeerInputValue output tag. Requires the O:Pt[*x*].OverridePeerInputEn output tag to be enabled. 0 = The override feature for peer inputs is not enabled. 1 = The override feature for peer inputs is enabled.	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PeerWindows0OverrideStatus	BOOL	**Peer Window 0 Override Status—**Indicates whether peer window 0 data mapped to the corresponding output point is set up to be overridden by the value in the Pt[*x*].OverridePeerWindow0Value output tag. Requires the O:Pt[*x*].OverridePeerWindow0En output tag to be enabled. 0 = The override feature for peer window 0 is not enabled. 1 = The override feature for peer window 0 is enabled.	Connection = Peer Ownership Output Data = Data with Peer

Table 5 - 1756-OB16IEF Module Input Data Tags (continued)

Name	Data Type	Tag Definition	Module Definition
Pt[*x*].PeerWindow1OverrideStatus	BOOL	**Peer Window 1 Override Status—**Indicates whether peer window 1 data mapped to the corresponding output point is set up to be overridden by the value in the Pt[*x*].OverridePeerWindow1Value output tag. Requires the O:Pt[*x*].OverridePeerWindow1En output tag to be enabled. 0 = The override feature for peer window 1 is not enabled. 1 = The override feature for peer window 1 is enabled.	Connection = Peer Ownership Output Data = Data with Peer
LocalClockOffset	DINT	**Local Clock Timestamp—**Indicates the offset between the current CST and the CIP Sync value when a valid CIP Sync time is available.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
OffsetTimestamp	DINT	**Timestamp Offset—**Indicates when the CIP Sync LocalClockOffset and GrandMasterID were last updated in CIPSync format.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
GrandMasterClockID	DINT	**Grandmaster Clock ID—**Indicates the ID of the CIP Sync grandmaster to which the module is synced.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer
Timestamp	DINT	**Timestamp—**A 64-bit CIP Sync timestamp of the last new output data or FuseBlown event.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Listen Only Output Data = None or Connection = Peer Ownership Output Data = Data with Peer

Output Tags

Output tags contain the structure of data continually sent from the controller to the I/O module that can modify the module's behavior.

Table 6 - 1756-OB16IEF Module Output Data Tags

Name	Data Type	Tag Definition	Module Definition
Pt[*x*].Data	BOOL	**Data—**Indicates the On/Off state to apply to the output point. 0 = Off 1 = On	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].ResetFuseBlown	BOOL	**Reset Blown Fuse—**Attempts to clear a blown fuse status and apply output data when the bit transitions from Off to On.	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverrideOutputEn	BOOL	**Override Output—**Overrides local output data for peer logic with the value defined in the Pt[*x*].OverrideOutputValue tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverrideOutputValue	BOOL	**Override Output Value—**Indicates the On/Off status to apply to all outputs mapped to the output point when the corresponding bit in the Pt[*x*].OverrideOutputEn tag is set. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverridePeerInputEn	BOOL	**Override Peer Input—**Overrides peer input data mapped to the output point with the value defined in the Pt[*x*].OverridePeerInputValue output tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverridePeerInputValue	BOOL	**Override Peer Input Value—**Indicates the On/Off status to apply to all peer inputs mapped to the output point when the corresponding bit in the Pt[x].OverridePeerInputEn output tag is enabled. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverridePeerWindow0En	BOOL	**Override Peer Window 0—**Overrides peer window 0 inputs mapped to the output point with the value defined in the Pt[x].OverridePeerWindow0Value output tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverridePeerWindow0Value	BOOL	**Override Peer Window 0 Value**—Indicates the On/Off status to apply to peer window 0 inputs mapped to the output point when the corresponding bit in the Pt[x].OverridePeerWindow0En output tag is enabled. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverridePeerWindow1En	BOOL	**Override Peer Window 1—**Overrides peer window 1 inputs mapped to the output point with the value defined in the Pt[x].OverridePeerWindow1Value output tag. 0 = Disable 1 = Enable	Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].OverridePeerWindow1Value	BOOL	**Override Peer Window 1 Value**—Indicates the On/Off status to apply to peer window 1 inputs mapped to the output point when the corresponding bit in the Pt[x].OverridePeerWindow1En output tag is enabled. 0 = Off 1 = On	Connection = Peer Ownership Output Data = Data with Peer

Table 6 - 1756-OB16IEF Module Output Data Tags (continued)

Name	Data Type	Tag Definition	Module Definition
Pt[*x*].PWMCycleTime	REAL	**PWM Cycle Time—**Defines the duration of each pulse cycle. Requires PWM to be enabled via the Pt[x].PWMEnable configuration tag. Valid values: 0.001…3600.0 seconds	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
Pt[*x*].PWMOnTime	REAL	**PWM On Time—**Defines the length of time that a pulse is active. Requires PWM to be enabled via the Pt[x].PWMEnable configuration tag. Valid values: 0.0002…3600.0 seconds or 0…100 percent	Connection = Data Output Data = Data or Scheduled per Module or Connection = Peer Ownership Output Data = Data with Peer
TimestampOffset	DINT	**Timestamp Offset**—Indicates the difference between the system time and the module's local time. The timestamp is in CIP Sync time. This tag is **not** available with peer ownership connections.	Connection = Data Output Data = Scheduled per Module
Timestamp	DINT	**Timestamp—**CIP Sync time at which to apply scheduled output data. This tag is **not** available with peer ownership connections.	Connection = Data Output Data = Scheduled per Module

Notes:

Numerics

A

B

C

D

E

F

G

I

L

M

O

P

R

S

T

V

Notes:

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support, you can find technical manuals, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools. You can also visit our Knowledgebase at http://www.rockwellautomation.com/knowledgebase for FAQs, technical information, support chat and forums, software updates, and to sign up for product notification updates.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect[SM] support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

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 1756-AT016A-EN-P - April 2012