How to create produced and consumed tags in Studio 5000

This guide shows how to create produced and consumed tags in Studio 5000 so that two ControlLogix controllers can exchange data over EtherNet/IP. The screens shown are from RSLogix 5000. You set up a produced tag in the producer controller, add the producer to the consumer's I/O Configuration, create a matching consumed tag, and then confirm that the value transfers.

Prerequisites

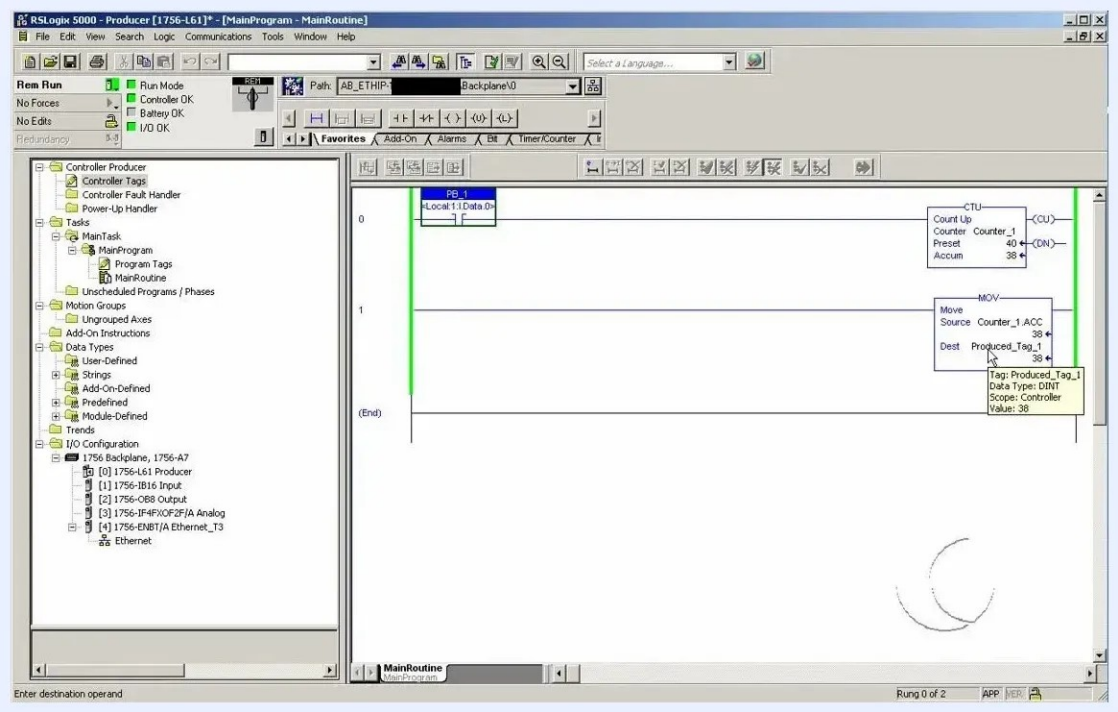
Before you start, make sure you have the following:

- An RSLogix 5000 project named "Producer" configured for a 1756-L61 processor, with the processor in Run mode and all modules (input, output, analog, and network cards) properly connected.
- A 1756-ENBT/A Ethernet/IP card in the producer's I/O Configuration for network communication.
- A second RSLogix 5000 project for the consumer controller. Its processor type can be the same (for example, 1756-L61), but its I/O tree differs from the producer's.
- The IP address, Ethernet card slot number, chassis size, and revisions of the producing PLC.

Part 1: Review the producer program

Step 1. Open the Producer project

Launch RSLogix 5000 and open the "Producer" project. The left pane lists the controller, tasks, data types, and I/O modules, including the 1756-ENBT/A Ethernet card. The right pane shows the ladder logic with a push button (PB_1), a counter (Counter_1), and a move instruction that transfers the counter's accumulated value to Produced_Tag_1.



RSLogix 5000 main window for the Producer controller, showing the I/O Configuration tree with the 1756-ENBT/A Ethernet card and ladder logic with PB_1, Counter_1, and a MOV instruction

Step 2. Verify the network configuration

In the I/O Configuration tree, confirm that these modules are present:

- 1756-L61 Producer (controller)
- 1756-IB16 Input
- 1756-OB8 Output
- 1756-IF4FXOF2F/A Analog
- 1756-ENBT/A Ethernet_T3 (Ethernet/IP card)

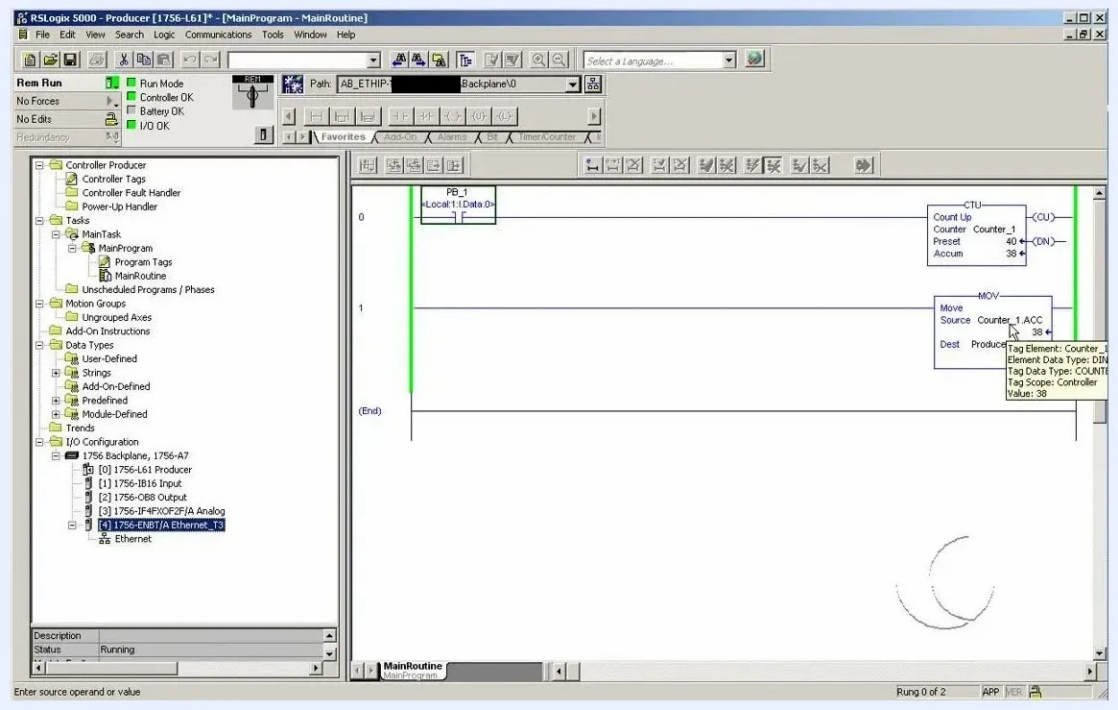
Also confirm in the status bar that the Ethernet module is configured and running.

Step 3. Inspect the ladder logic that feeds the produced tag

Examine the main routine:

- The first rung contains the push button PB_1 input (Local:1:I.Data.0). It increments Counter_1, which has a preset of 40 (the accumulated value in the example is 38).
- The second rung uses a MOV (Move) instruction. The source is Counter_1.ACC (value 38) and the destination is Produced_Tag_1.

Hover over or right-click the Produced_Tag_1 destination to view its details. In this example, the tag is a DINT with Controller scope and a value of 38.

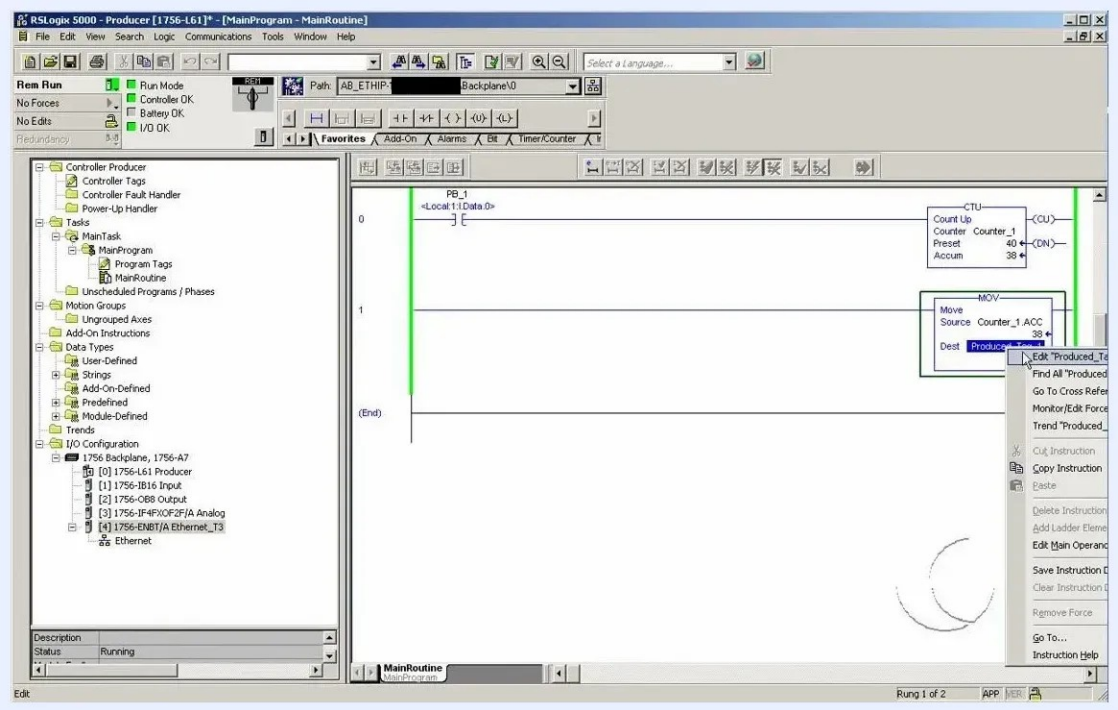


Ladder logic with the MOV instruction highlighted, showing Counter_1.ACC as the source and Produced_Tag_1 as the destination, with a tooltip showing Data Type DINT, Scope Controller, and Value 38

Part 2: Create the produced tag

Step 4. Open the produced tag properties

Right-click Produced_Tag_1 in the ladder logic or the tag browser. Then select **Edit "Produced_Tag_1"** or **Properties** from the context menu.



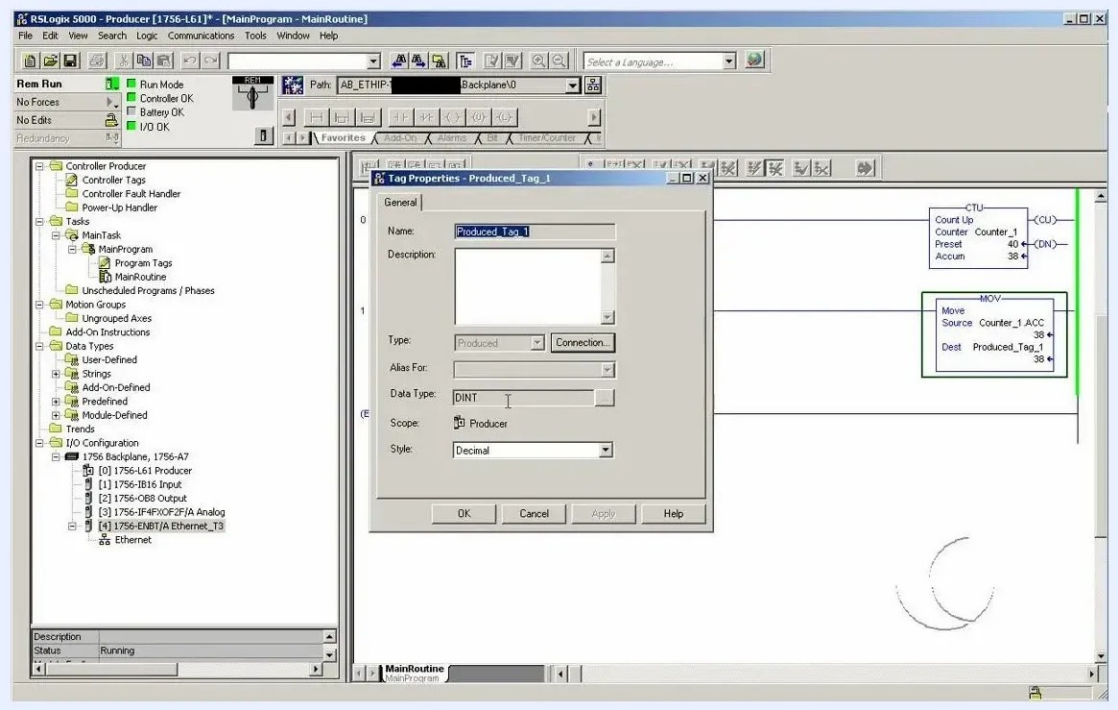
Context menu open after right-clicking Produced_Tag_1, with options such as Edit, Find All, Go To Cross Reference, Monitor/Edit Force, and Trend

Step 5. Configure the produced tag

First, choose a data type that works for produced and consumed tags:

- **DINT** (Double Integer) is the most common data type.
- **REAL** or **FLOAT** are optional if your application requires them.

- Boolean tags cannot be produced or consumed directly. To exchange Boolean data, create a **User-Defined Data Type (UDT)** and include the Boolean within it.

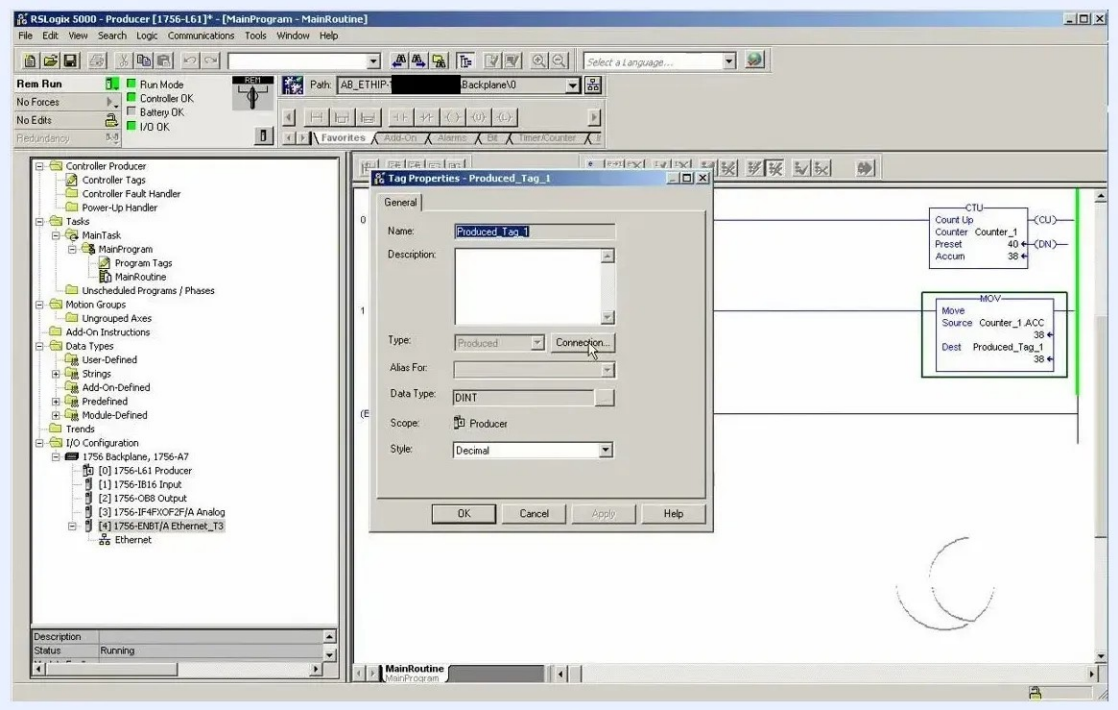


Tag Properties dialog for Produced_Tag_1 with DINT shown as the data type

In the Tag Properties dialog, enter the following:

- **Name:** Produced_Tag_1
- **Type:** Produced (select it from the drop-down menu)
- **Data Type:** DINT
- **Scope:** Producer (controller scope)
- **Style:** Decimal

Click **Connection...** to configure consumer settings. Under Connection, set the number of consumers to 1 if only one other processor will consume this tag.



Tag Properties dialog for Produced_Tag_1 showing Name, Type (Produced), Data Type (DINT), Scope (Producer), and Style (Decimal), with the Connection button highlighted

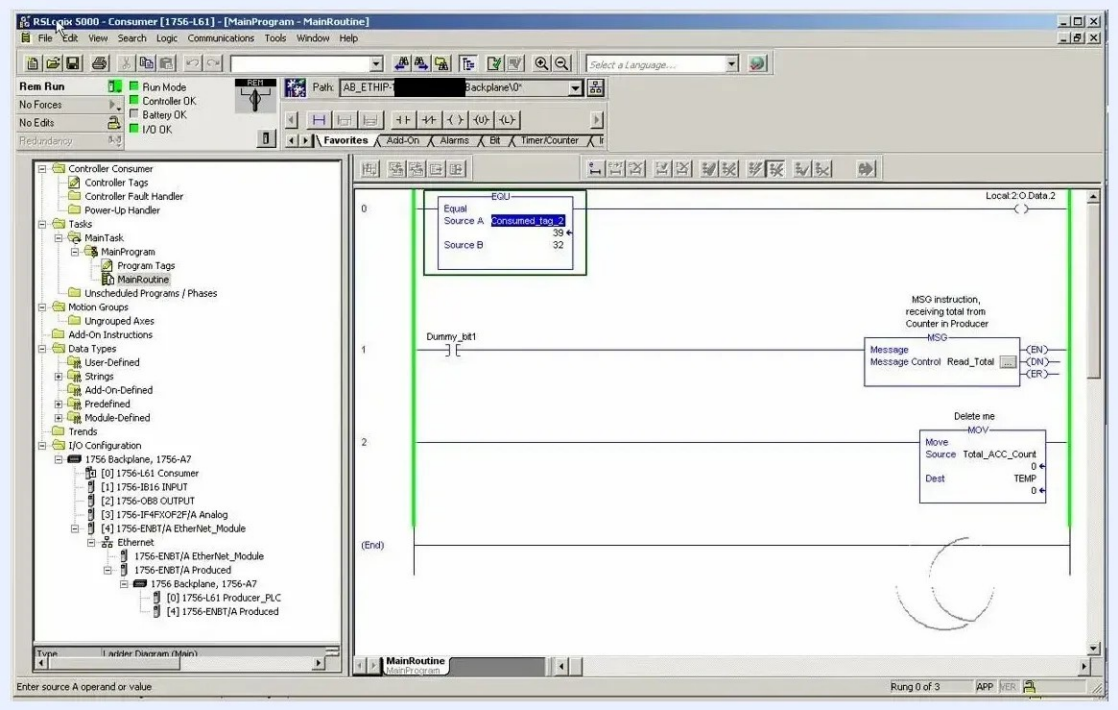
Click **OK** or **Apply** to save the settings. The tag is now a produced tag, ready for consumption by another controller.

Part 3: Prepare the consumer controller network

Step 6. Open the consumer controller project

Open the RSLogix 5000 project for the consumer controller (example name: **RSLogix 5000 Consumer**). In the consumer's main routine, locate the rung that uses the consumed tag (for example, Consumed_tag_2).

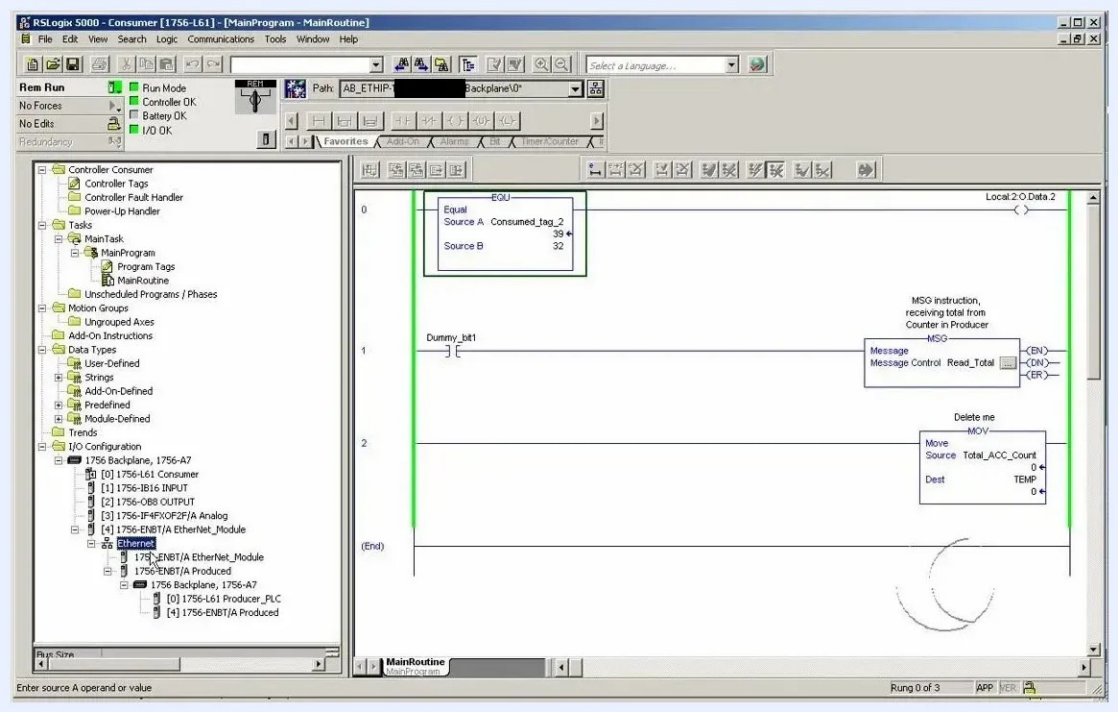
In this example, an EQU (Equal) instruction compares Consumed_tag_2 (value 39) to a constant (32). The logic can also include MSG (Message) and MOV (Move) instructions for other data handling.



RSLogix 5000 main window for the Consumer controller, showing the ladder logic and I/O Configuration tree, with Consumed_tag_2 used in the logic

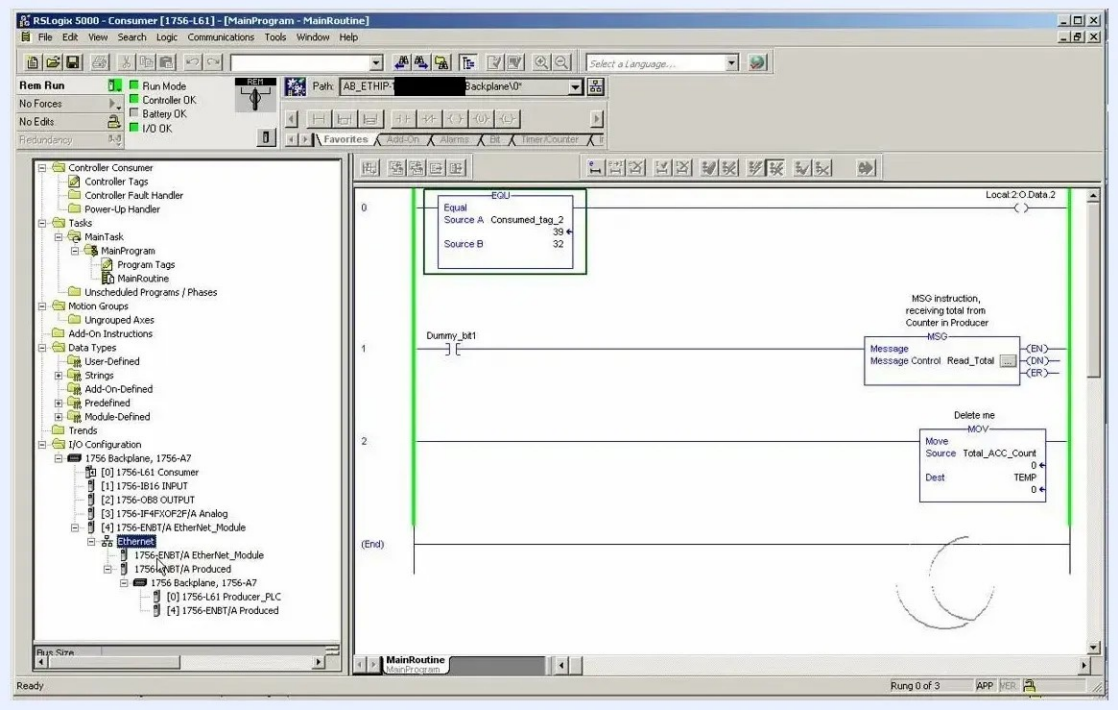
Step 7. Add a new module under the Ethernet card

In the consumer's I/O Configuration tree, find the **Ethernet** module (for example, 1756-ENBT/A EtherNet_Module). You use this module to establish the connection to the producer PLC.



Consumer controller's I/O Configuration tree with the Ethernet module highlighted

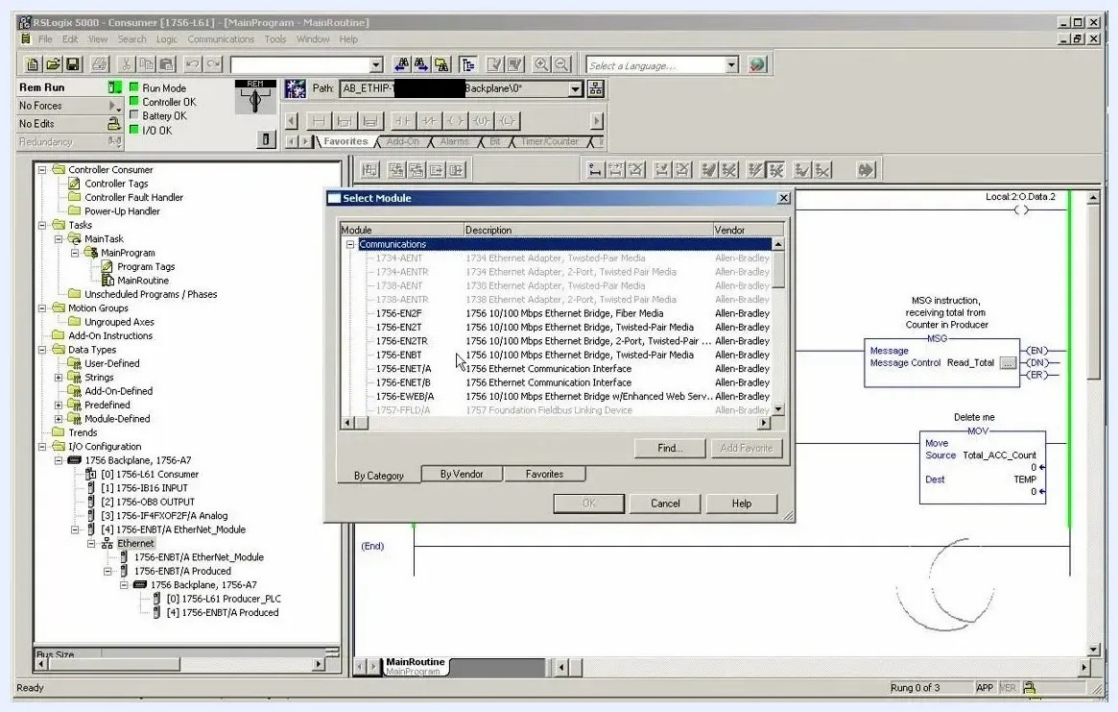
Right-click the **Ethernet** module and select **Add New Module**.



Consumer controller's I/O Configuration tree with the Ethernet module selected, ready for module addition

Step 8. Configure the network module for the producer PLC

In the Select Module dialog, choose the Ethernet module type that matches the producer PLC's Ethernet module (for example, **1756-ENBT/A**) from the list under Communications. Click **OK**.



Select Module dialog showing a list of Ethernet modules under Communications, with 1756-ENBT/A highlighted

In the New Module dialog, enter the following:

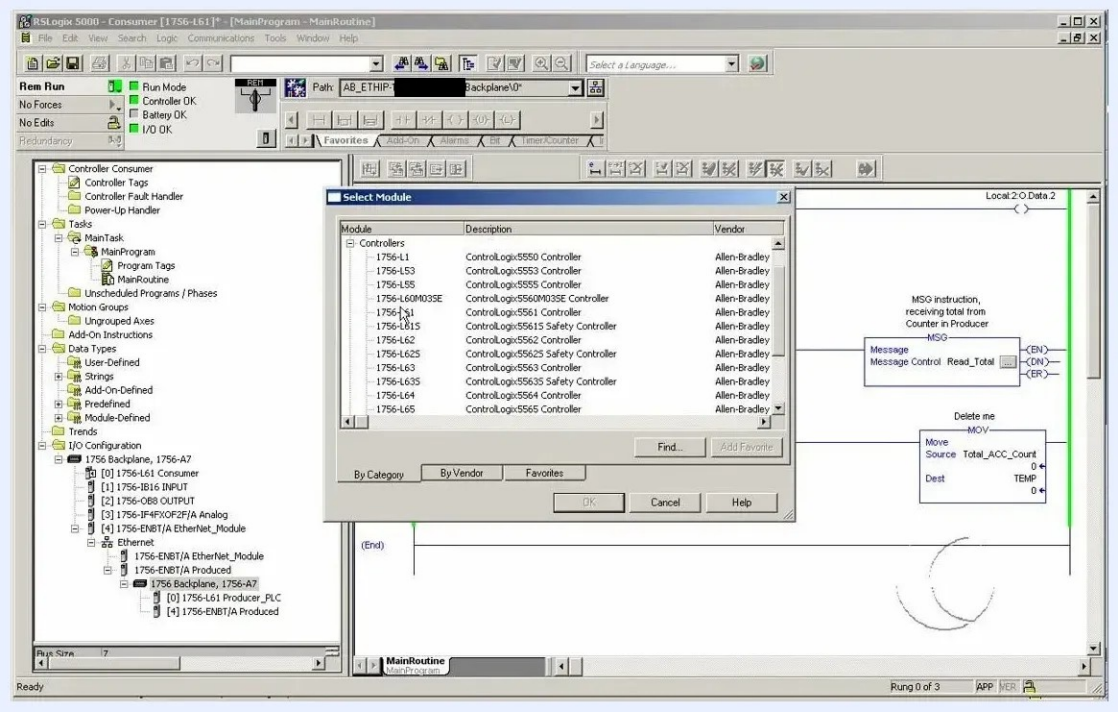
- **Name:** A descriptive name for the module (for example, EtherNet_Module).
- **Description:** Optional.
- **Slot:** The slot number of the network card in the producing PLC (for example, 0).
- **Chassis Size:** The chassis size of the producing PLC (for example, 17).
- **Revision:** The correct module revision (for example, 4.1).
- **IP Address:** The IP address of the producing PLC.
- **Electronic Keying:** The keying method (for example, Compatible Keying).

Do not enter the IP address of the consumer PLC. Use the address of the PLC you want to receive data from. Click **OK** to add the module, or cancel if you are only demonstrating the process.

After you add the module, RSLogix 5000 automatically adds the backplane for the producing PLC. You then see the network card and backplane under the Ethernet module in the I/O tree.

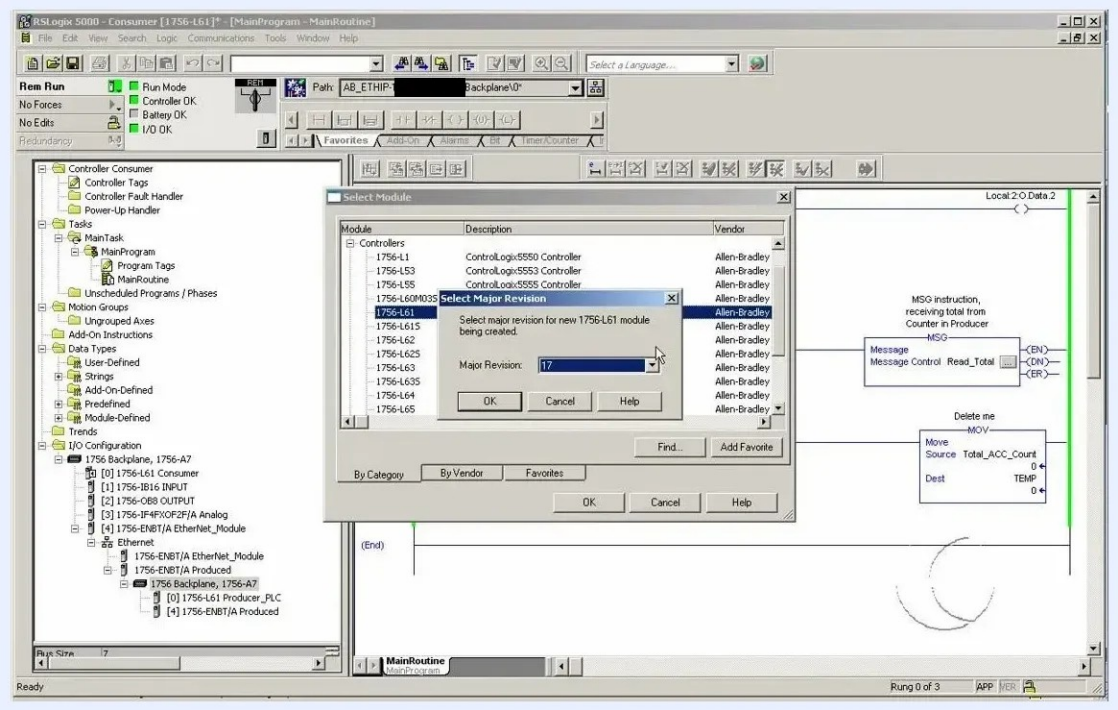
Step 9. Add the remote controller module

Right-click the new backplane under the producer PLC's Ethernet module and select **New Module**. In the Select Module dialog, choose the controller type that matches the producing PLC (for example, **1756-L61**).



Select Module dialog showing a list of ControlLogix controllers, with 1756-L61 highlighted

When prompted, select the major revision for the controller (for example, 17).

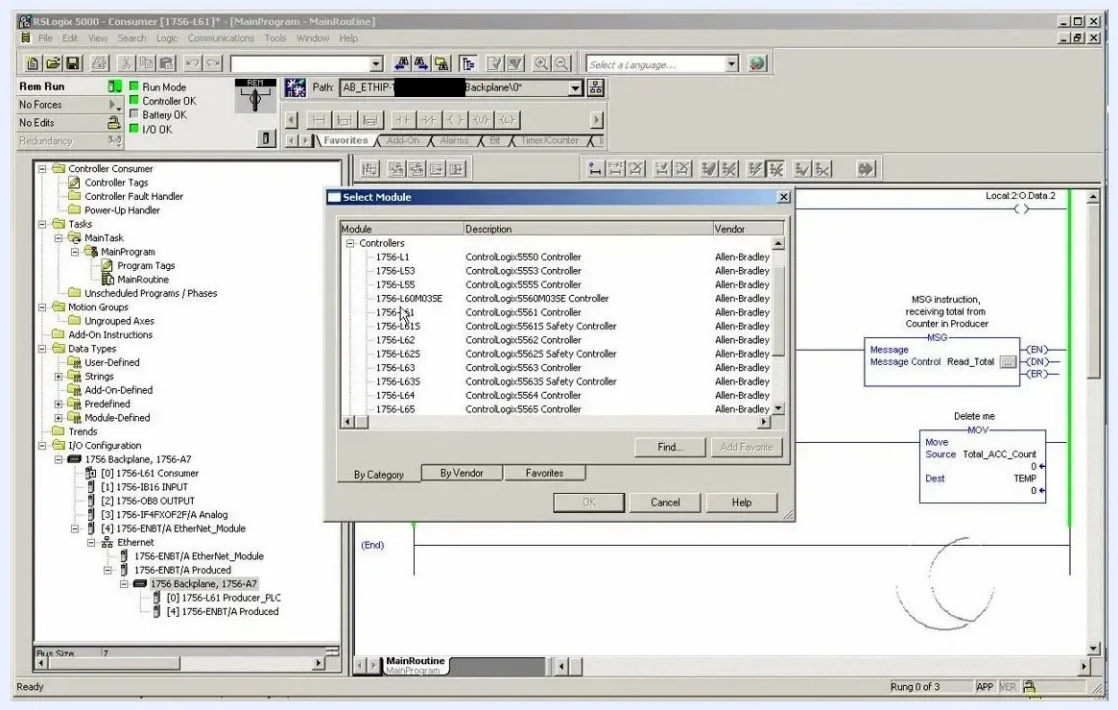


Select Major Revision dialog for 1756-L61 with major revision 17 selected

In the New Module dialog for the controller, enter the following:

- **Name:** A name that is meaningful for your project. It does not have to match the name on the producer PLC.
- **Slot:** The slot where the controller is installed on the remote system (for example, 1).
- **Description:** Optional.
- **Revision:** Confirm the revision (for example, 17.1).

Click **OK** to add the controller module, or **Cancel** if you are only demonstrating the process.



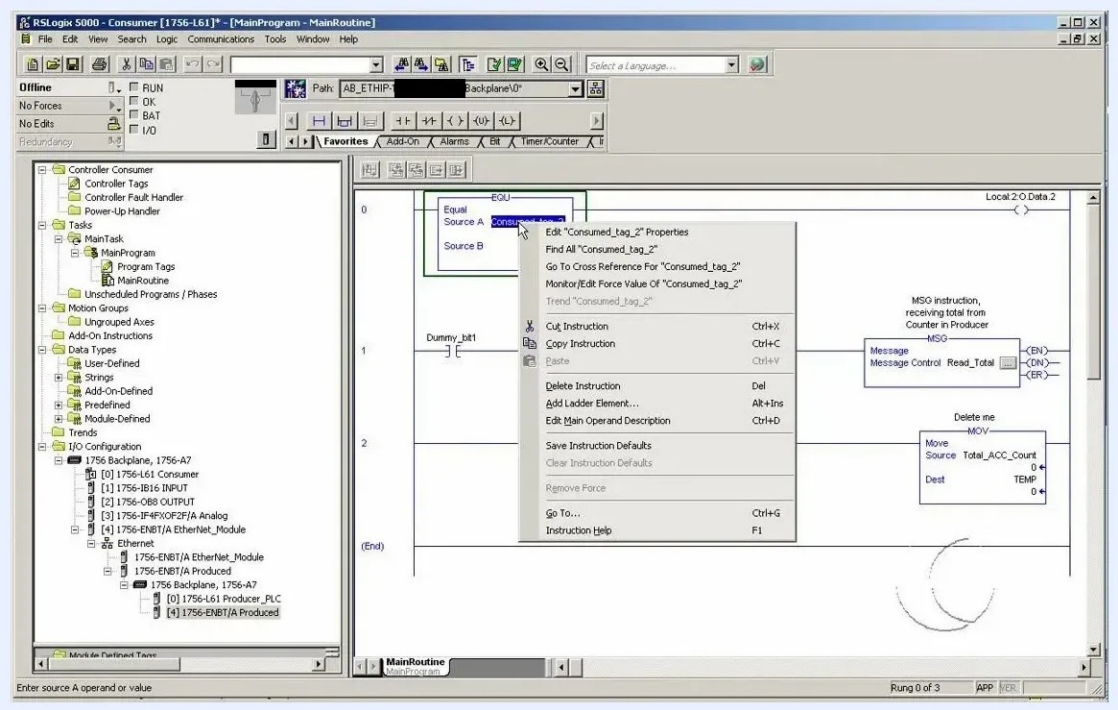
New Module dialog for the 1756-L61 controller, showing fields for Name, Slot, Description, and Revision

The backplane and controller now appear in the I/O Configuration tree, which confirms the setup.

Part 4: Create the consumed tag

Step 10. Edit the consumed tag properties

Locate the consumed tag (for example, Consumed_tag_2) in the ladder diagram. Right-click it and select **Edit "Consumed_tag_2" Properties**.



Ladder logic with the right-click context menu open on Consumed_tag_2, showing the Edit "Consumed_tag_2" Properties option

In the Tag Properties window:

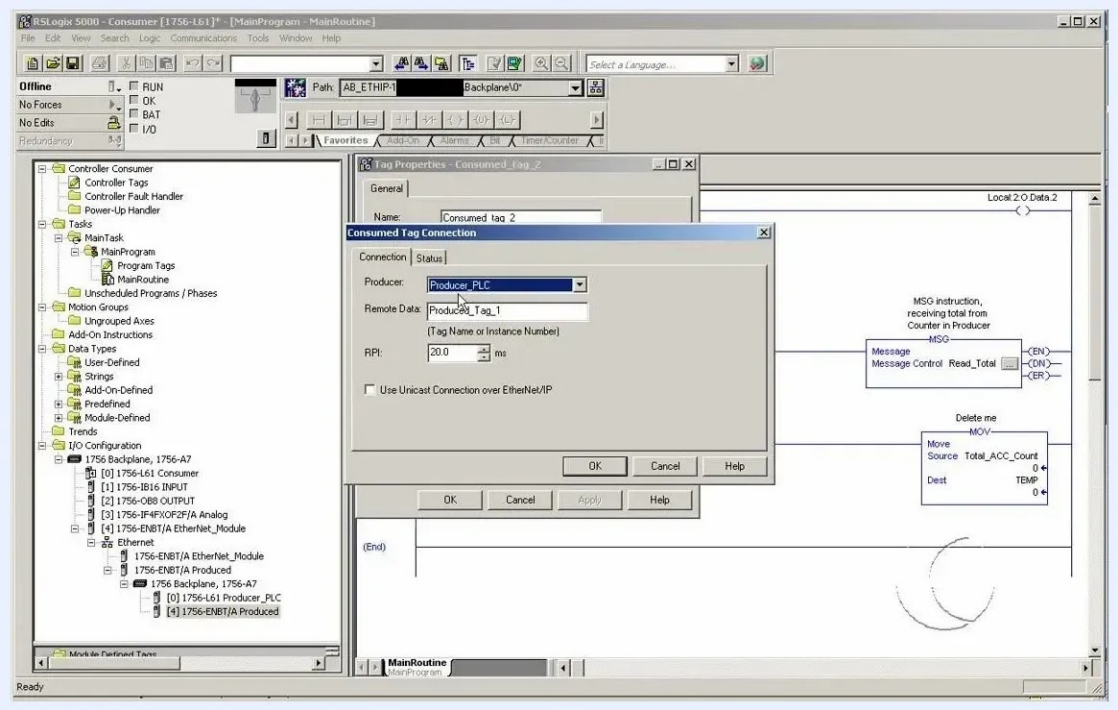
- Enter a name that makes sense for your application (for example, Consumed_tag_2).
- Optionally, add a description.
- Set **Type** to Consumed from the drop-down menu.
- Make sure **Data Type** matches the producer tag (for example, DINT).

Step 11. Configure the consumed tag connection

Click **Connection...** next to the Type field. In the Consumed Tag Connection window, set the following:

- **Producer:** Select the producer from the drop-down (for example, Producer_PLC).

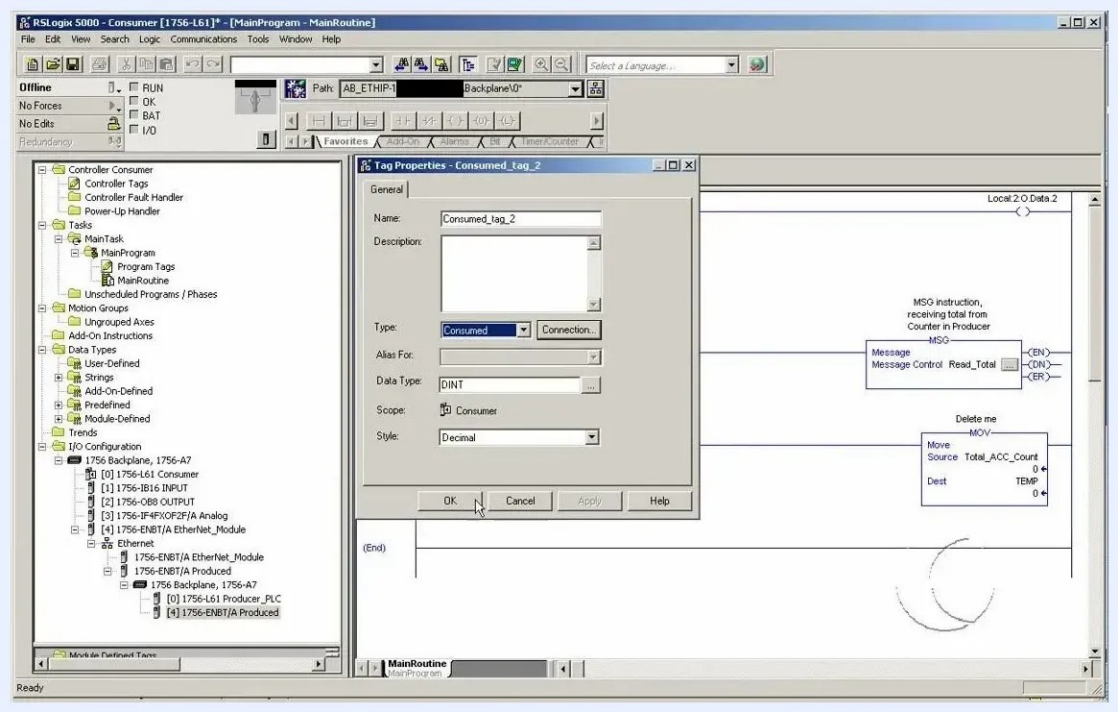
- **Remote Data:** Enter the exact name of the producing tag on the remote PLC (for example, Produced_Tag_1).
- **RPI (Requested Packet Interval):** Set as needed (for example, 20.0 ms).
- **Use Unicast Connection over EtherNet/IP:** Optionally select it if your network setup requires it.



Consumed Tag Connection dialog with Producer_PLC selected, Produced_Tag_1 entered as remote data, and RPI set to 20.0 ms

Step 12. Save the consumed tag

Click **OK** in the Consumed Tag Connection window. Then click **OK** in the Tag Properties window to save your changes.

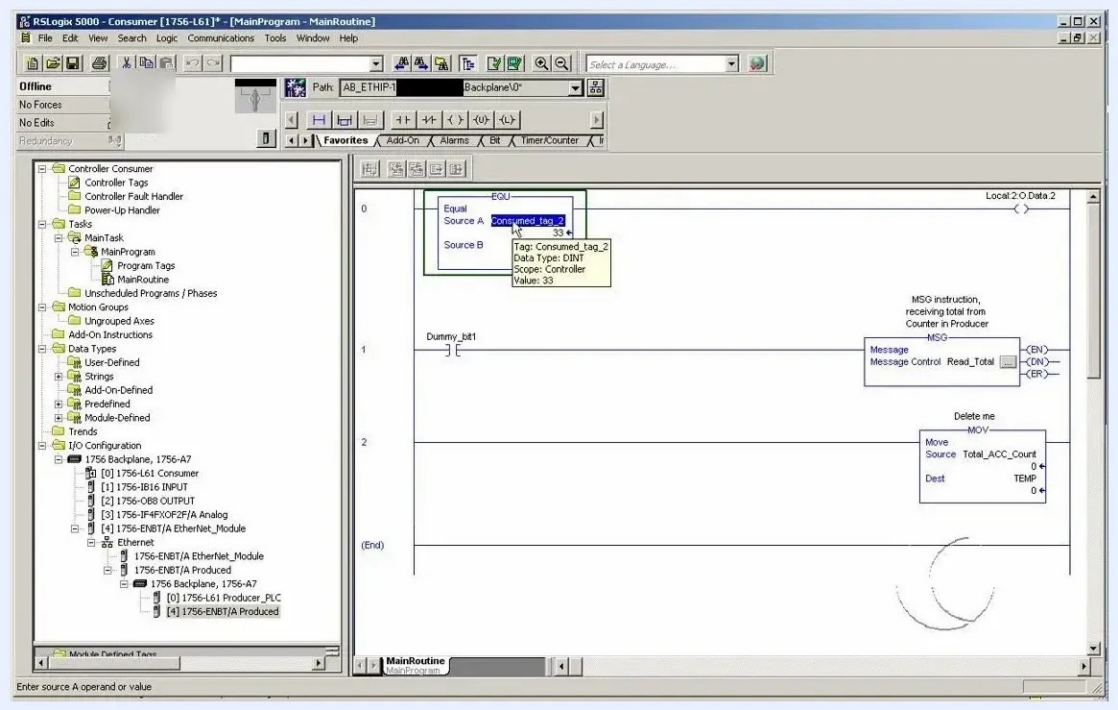


Tag Properties dialog for Consumed_tag_2 with the OK button highlighted

Step 13. Confirm the consumed tag in your logic

Confirm that the consumed tag is now used in your logic. In this example, Consumed_tag_2 is **Source A** in an **EQU (Equal)** instruction, compared to a fixed value (for example, 33).

Hover over the tag to see its data type (DINT), scope (Controller), and current value (for example, 33).

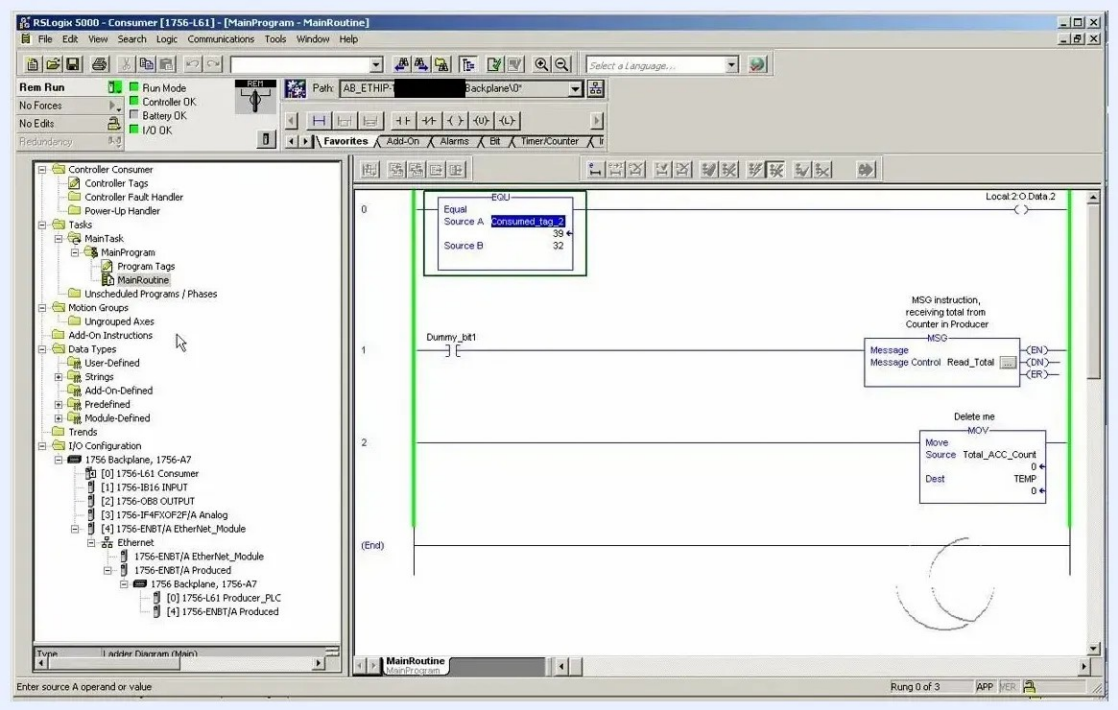


Ladder logic showing an EQU instruction with Consumed_tag_2 as Source A, value 33, and a tooltip with tag details

Part 5: Verify the data transfer

Step 14. Check the producer tag operation

Make sure you are online with the producer controller. When you toggle the PB_1 bit from its context menu, the counter's accumulated value increments. The MOV instruction then transfers that value to the produced tag (Produced_Tag_1).

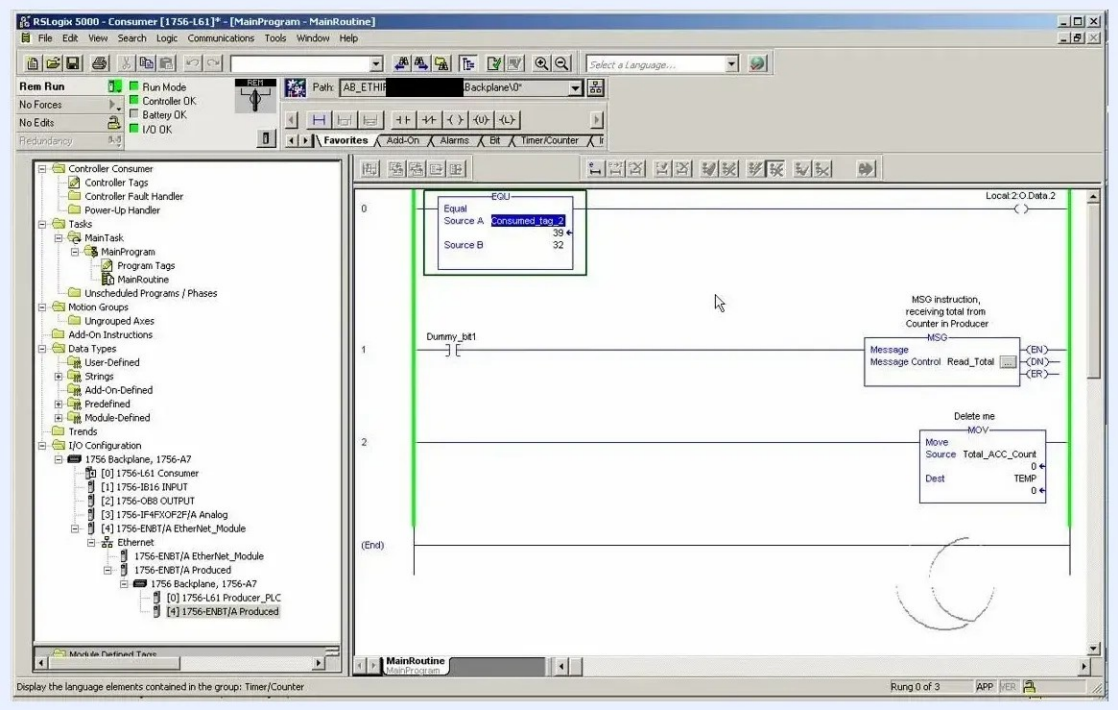


Ladder logic with the context menu open on PB_1, showing options such as Toggle Bit, Force On, and Force Off

Step 15. Go online with the consumer controller

Switch the consumer controller to **Run** mode if it is not already online. Observe the logic in real time. The consumed tag value updates as data arrives from the producer PLC.

In this example, Consumed_tag_2 is 39 and the EQU instruction compares it to 32.

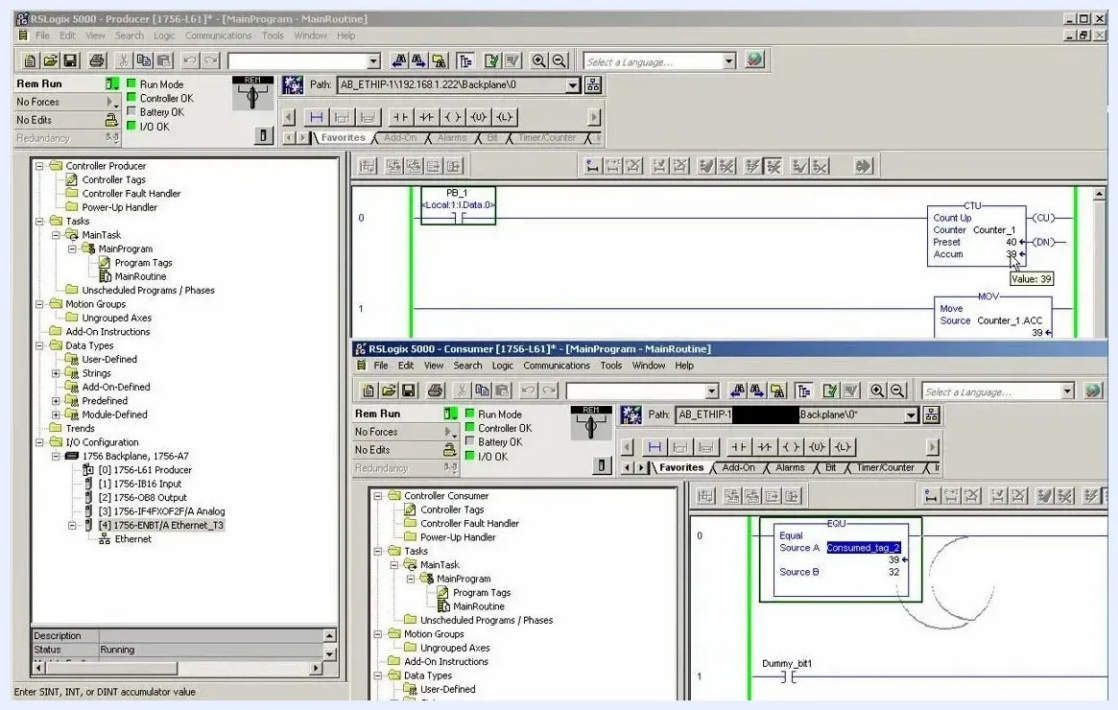


RSLogix 5000 in Run mode, with the ladder logic showing the Consumed_tag_2 value updating in real time

Step 16. Compare the producer and consumer values

In the Producer window, find the rung containing the **CTU (Count Up)** instruction. Confirm that the **Accum** value for Counter_1 is **39**, and that the **MOV** instruction moves Counter_1.ACC (39) to Produced_Tag_1.

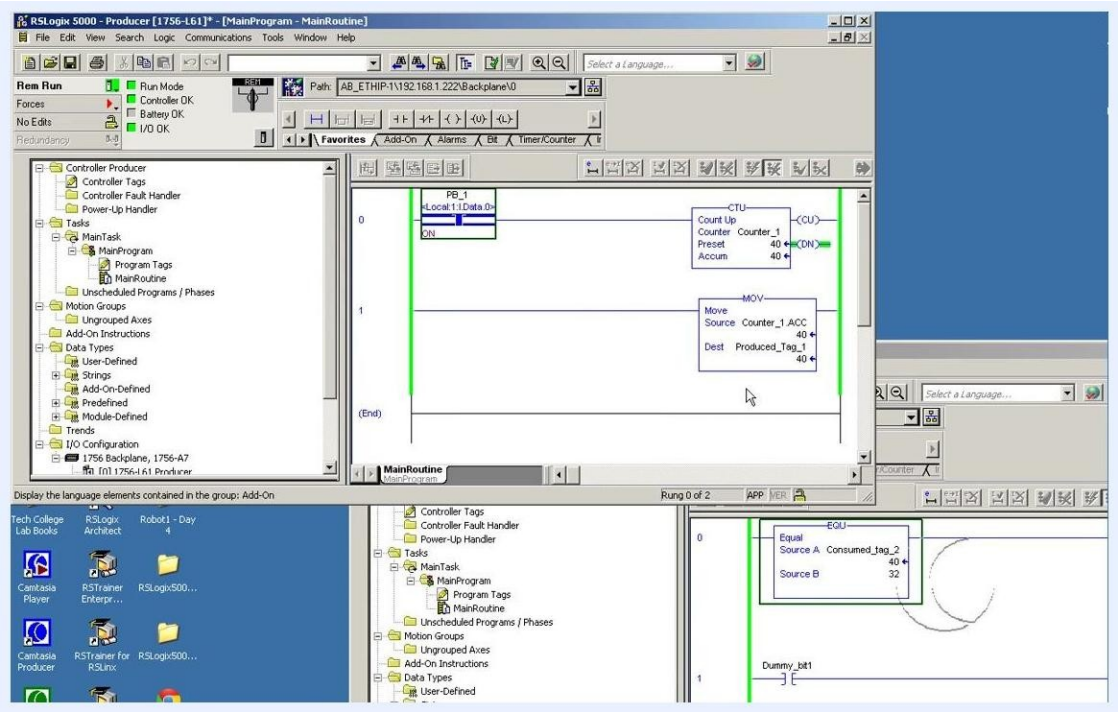
In the Consumer window, verify that the **EQU** instruction uses consumed_tag_2 as **Source A** with a value of **39**, and that **Source B** is **32**. The tooltip confirms the value of consumed_tag_2 is **39**. Adjust the window sizes as needed so both sets of logic are visible.



RSLogix 5000 Producer and Consumer windows showing Counter_1 Accum = 39, MOV to Produced_Tag_1 = 39, and consumed_tag_2 = 39 in the EQU instruction

Step 17. Confirm the remote update

As the value in the producer PLC changes, confirm that consumed_tag_2 in the consumer PLC updates accordingly. In this example, consumed_tag_2 becomes **40**, matching the updated Produced_Tag_1. The EQU instruction now compares 40 to 32.



RSLogix 5000 Producer and Consumer windows showing Counter_1.ACC = 40, Produced_Tag_1 = 40, and consumed_tag_2 = 40 in the consumer logic

Summary

To create produced and consumed tags in Studio 5000, you complete these tasks:

- Set the producer tag's **Type** to Produced, use a DINT data type (or a UDT for Boolean data), and set the number of consumers in **Connection...**.
- In the consumer project, add the producing PLC's Ethernet module, using the producer's IP address, slot, chassis size, and revisions.
- Add the remote controller under the new backplane. Its name is user-defined and does not need to match the producer's actual name.
- Set the consumed tag's **Type** to Consumed, match the producer's data type, and enter the exact remote tag name in **Remote Data**.

- Go online and confirm that the consumed tag value matches the produced tag value as the producer value changes.