

Logix5000 Controllers Function Block Diagram



Allen-Bradley

Catalog Numbers 1756 ControlLogix, 1769 CompactLogix, 1789 SoftLogix, 1794 FlexLogix, PowerFlex 700S with DriveLogix

Programming Manual



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.

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, 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.

Table of Contents**Preface**

Purpose of this Manual	5
How to Use this Manual	5

**Program a Function Block
Diagram****Chapter 1**

Introduction	7
Choose the Function Block Elements	8
Choose a Tag Name for an Element.	8
Define the Order of Execution	10
Data Latching.	10
Order of Execution	12
Resolve a Loop.	13
Resolve Data Flow Between Two Blocks.	15
Create a One Scan Delay.	15
Summary.	16
Identify any Connectors.	16
Define Program/Operator Control.	17
Add a Sheet.	20
Add a Function Block Element.	20
Create a Text Box.	22
Language Switching	23
Connect Elements	23
Show or Hide a Pin	23
Wire Elements Together	24
Mark a Wire with the Assume Data Available Indicator	24
Assign a Tag	25
Create and Assign a New Tag	25
Assign an Existing Tag	26
Assign an Immediate Value (Constant).	26
Use an IREF.	26
Enter a Value in the Tag of a Block	27
Connect Blocks with an OCON and ICON	27
Add an OCON	27
Add an ICON	28
Verify the Routine	28

Purpose of this Manual

This manual shows how to program Logix5000 controllers with the function block diagram (FBD) programming language. This manual is one of a set of related manuals that show common procedures for programming and operating Logix5000 controllers. For a complete list of common procedures manuals, see the Logix 5000 Controllers Common Procedures Programming Manual, publication 1756-PM001.

The term Logix5000 controller refers to any controller that is based on the Logix5000 operating system, such as:

- CompactLogix controllers
- ControlLogix controllers
- DriveLogix controllers
- FlexLogix controllers
- SoftLogix5800 controllers

How to Use this Manual

Some text is formatted differently from the rest of the text.

Text that is	Identifies	For example	Means
<i>Italic</i>	the actual name of an item that you see on your screen or in an example	Right-click <i>User-Defined ...</i>	Right-click the item that is named User-Defined.
<i>courier</i>	information that you must supply based on your application (a variable)	Right-click <i>name_of_program ...</i>	You must identify the specific program in your application. Typically, it is a name or variable that you have defined.
enclosed in brackets	a keyboard key	Press [Enter].	Press the Enter key.

Notes:

Program a Function Block Diagram

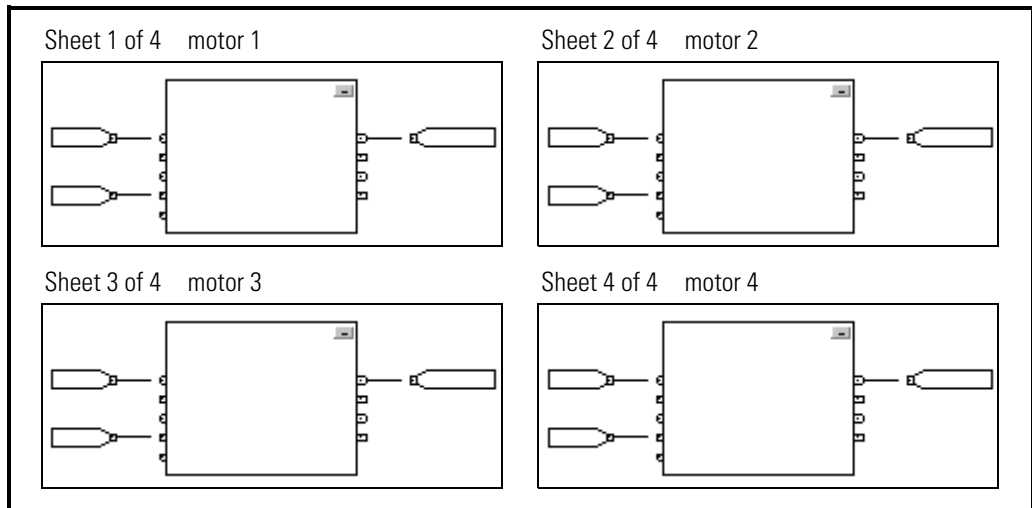
Introduction

To make it easier to navigate through a function block routine, divide the routine into a series of sheets.

- Sheets help you organize and find your function blocks. They do not affect the order in which the function blocks execute.
- When the routine executes, all the sheets execute.
- In general, use one sheet for each device (motor, valve, etc.)

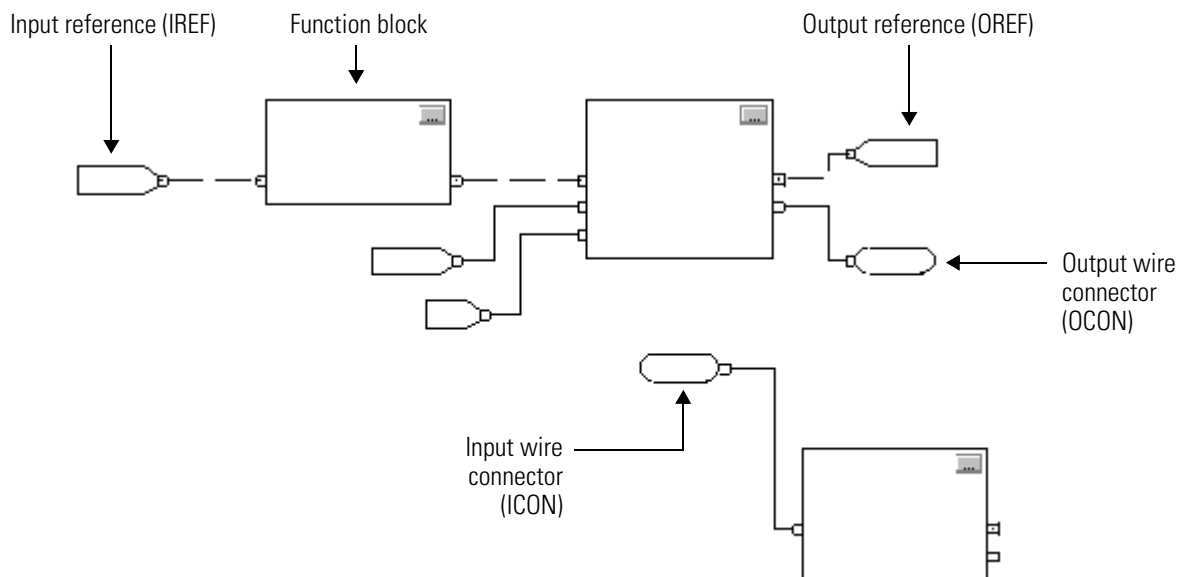
EXAMPLE

Motor Control Routine



Choose the Function Block Elements

To control a device, use these elements:



To choose function block elements:

If you want to	Then use a
Supply a value from an input device or tag	Input reference (IREF)
Send a value to an output device or tag	Output reference (OREF)
Perform an operation on an input value or values and produce an output value or values	Function block
Transfer data between function blocks when they are: <ul style="list-style-type: none"> • far apart on the same sheet. • on different sheets within the same routine. 	Output wire connector (OCON) and an input wire connector (ICON)
Disperse data to several points in the routine	Single output wire connector (OCON) and multiple input wire connectors (ICON)

Choose a Tag Name for an Element

Each function block uses a tag to store configuration and status information about the instruction.

- When you add function block instruction, RSLogix 5000 software automatically creates a tag for the block. You can use this tag as is, rename the tag, or assign a different tag.
- For IREFs and OREFs, you have to create a tag or assign an existing tag.

For a	Specify
Tag	<i>tag_name</i>
Bit number of a larger data type	<i>tag_name.bit_number</i>
Member of a structure	<i>tag_name.member_name</i>
Element of a one dimension array	<i>tag_name[x]</i>
Element of a two dimension array	<i>tag_name[x,y]</i>
Element of a three dimension array	<i>tag_name[x,y,z]</i>
Element of an array within a structure	<i>tag_name.member_name[x]</i>
Member of an element of an array	<i>tag_name[x,y,z].member_name</i>

where:

x is the location of the element in the first dimension.

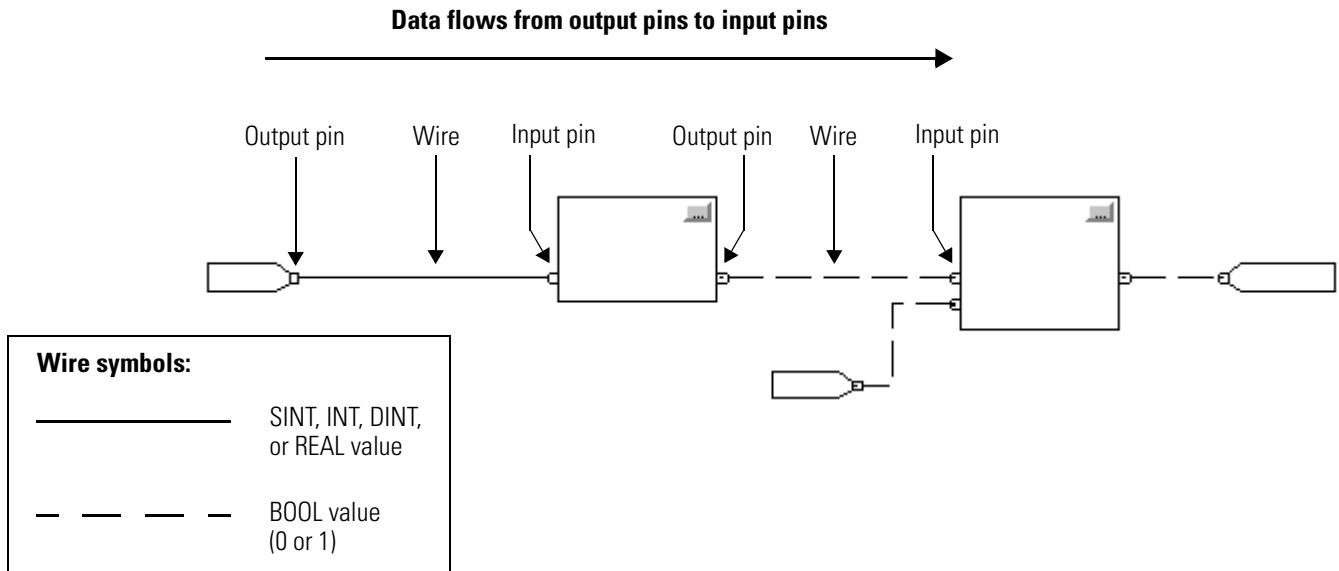
y is the location of the element in the second dimension.

z is the location of the element in the third dimension.

For a structure within a structure, add an additional *member_name*.

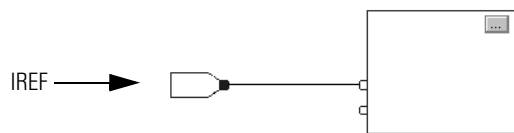
Define the Order of Execution

You define execution order (flow of data) by wiring elements together and indicating any input (feedback) wires, if necessary. The location of a block does not affect the order in which the blocks execute.

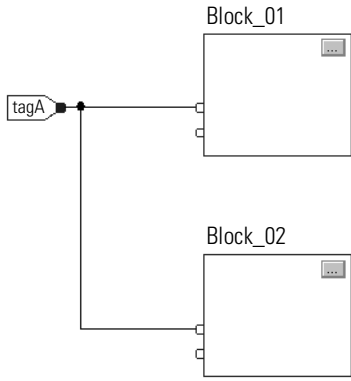


Data Latching

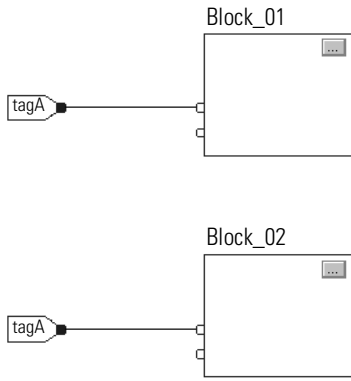
If you use an IREF to specify input data for a function block instruction, the data in that IREF is latched for the scan of the function block routine. The IREF latches data from program-scoped and controller-scoped tags. The controller updates all IREF data at the beginning of each scan.



In this example, the value of tagA is stored at the beginning of the routine's execution. The stored value is used when Block_01 executes. The same stored value is also used when Block_02 executes. If the value of tagA changes during execution of the routine, the stored value of tagA in the IREF does not change until the next execution of the routine.

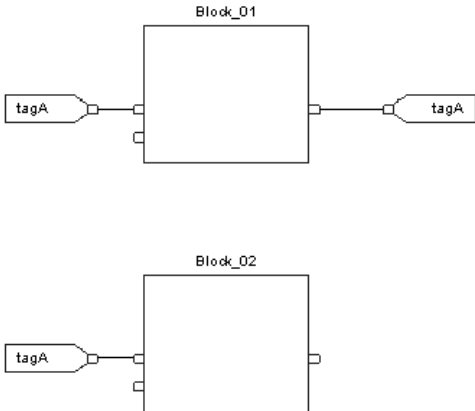


This example is the same as the one above. The value of tagA is stored only once at the beginning of the routine's execution. The routine uses this stored value throughout the routine.



Starting with RSLogix 5000 software, version 11, you can use the same tag in multiple IREFs and an OREF in the same routine. Because the values of tags in IREFs are latched every scan through the routine, all IREFs will use the same value, even if an OREF obtains a different tag value during execution of the routine. In this example, if tagA has a value of 25.4 when the routine starts executing this scan, and Block_01 changes the value of tagA to 50.9, the second IREF wired into Block_02 will still use a value of 25.4 when Block_02

executes this scan. The new tagA value of 50.9 will not be used by any IREFs in this routine until the start of the next scan.



Order of Execution

The RSLogix 5000 programming software automatically determines the order of execution for the function blocks in a routine when you:

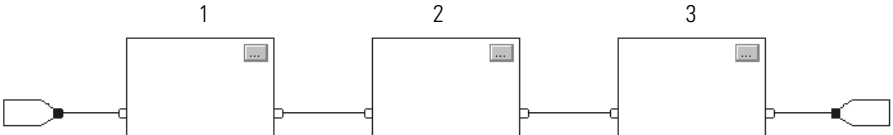
- verify a function block routine.
- verify a project that contains a function block routine.
- download a project that contains a function block routine.

You define execution order by wiring function blocks together and indicating the data flow of any feedback wires, if necessary.

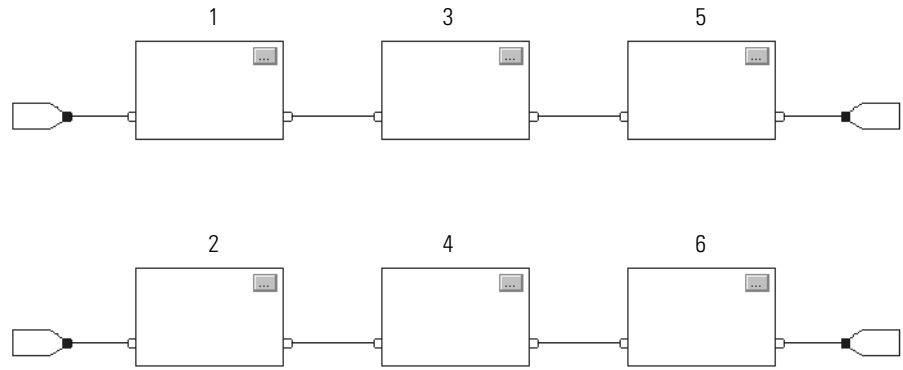
If function blocks are not wired together, it does not matter which block executes first. There is no data flow between the blocks.



If you wire the blocks sequentially, the execution order moves from input to output. The inputs of a block require data to be available before the controller can execute that block. For example, block 2 has to execute before block 3 because the outputs of block 2 feed the inputs of block 3.

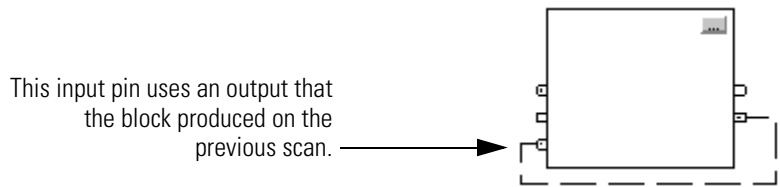


Execution order is only relative to the blocks that are wired together. This example is fine because the two groups of blocks are not wired together. The blocks within a specific group execute in the appropriate order in relation to the blocks in that group.

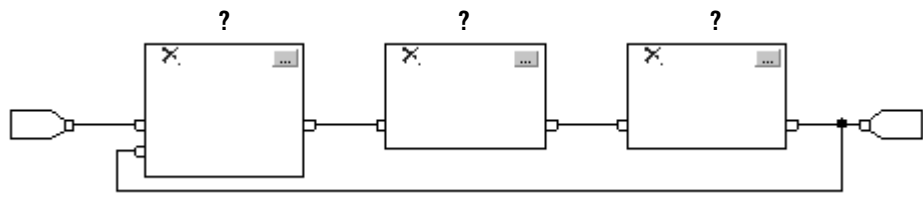


Resolve a Loop

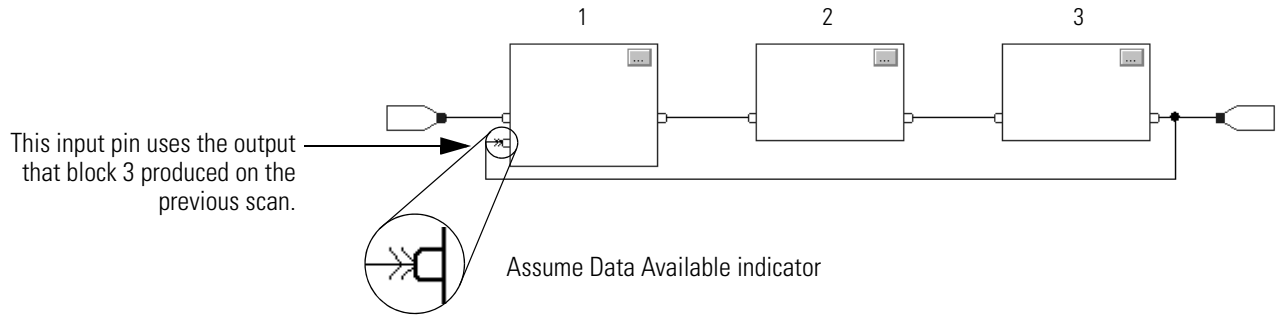
To create a feedback loop around a block, wire an output pin of the block to an input pin of the same block. This example is fine. The loop contains only a single block, so execution order does not matter.



If a group of blocks are in a loop, the controller cannot determine which block to execute first. In other words, it cannot resolve the loop.

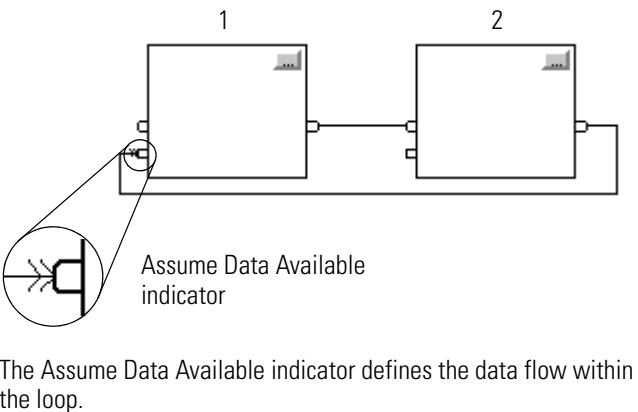
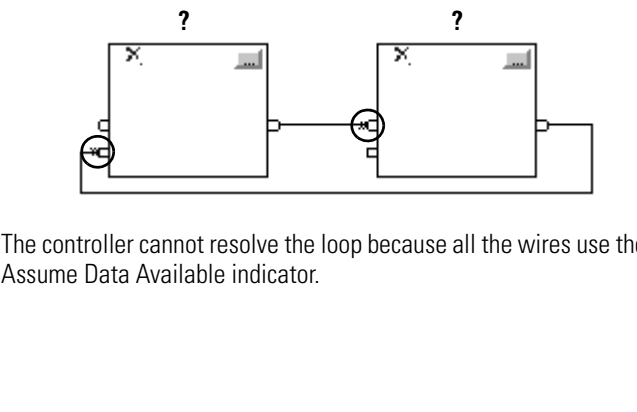


To identify which block to execute first, mark the input wire that creates the loop (the feedback wire) with the Assume Data Available indicator. In this example, block 1 uses the output from block 3 that was produced in the previous execution of the routine.



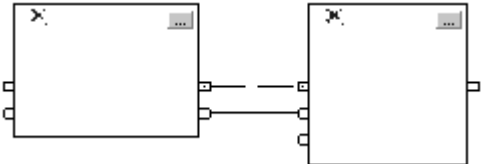
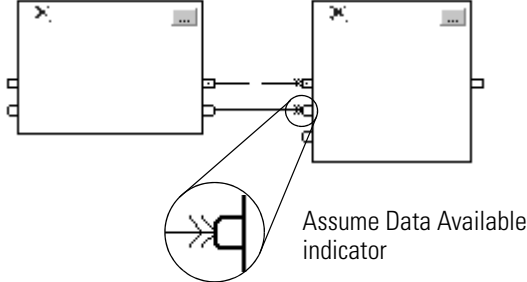
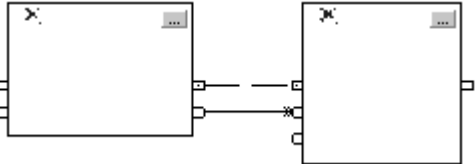
The Assume Data Available indicator defines the data flow within the loop. The arrow indicates that the data serves as input to the first block in the loop.

Do not mark all the wires of a loop with the Assume Data Available indicator.

This is OK	This is NOT OK
 <p>The Assume Data Available indicator defines the data flow within the loop.</p>	 <p>The controller cannot resolve the loop because all the wires use the Assume Data Available indicator.</p>

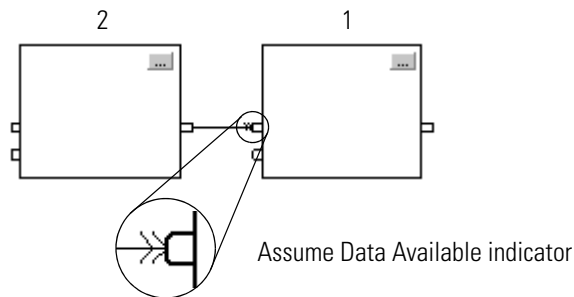
Resolve Data Flow Between Two Blocks

If you use two or more wires to connect two blocks, use the same data flow indicators for all of the wires between the two blocks.

This is OK	This is NOT OK
 <p data-bbox="154 716 685 743">Neither wire uses the Assume Data Available indicator.</p>  <p data-bbox="516 1020 743 1073">Assume Data Available indicator</p> <p data-bbox="154 1125 657 1152">Both wires use the Assume Data Available indicator.</p>	 <p data-bbox="820 716 1464 768">One wire uses the Assume Data Available indicator while the other wire does not.</p>

Create a One Scan Delay

To produce a one scan delay between blocks, use the Assume Data Available indicator. In this example, block 1 executes first. It uses the output from block 2 that was produced in the previous scan of the routine.



Summary

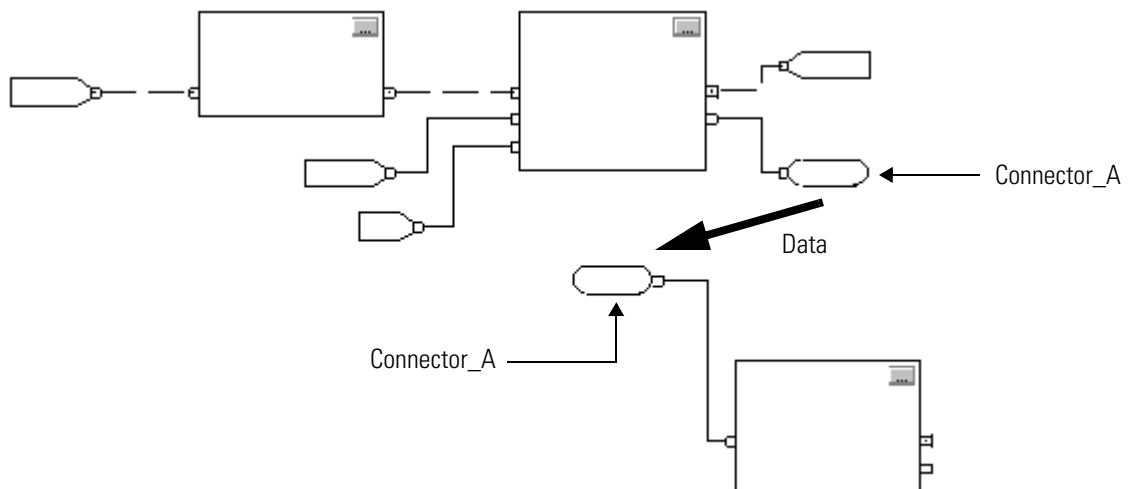
A function block routine executes in this order:

1. The controller latches all data values in IREFs.
2. The controller executes the other function blocks in the order determined by how they are wired.
3. The controller writes outputs in OREFs.

Identify any Connectors

Like wires, connectors transfer data from output pins to input pins. Use connectors when:

- the elements that you want to connect are on different sheets within the same routine.
- a wire is difficult to route around other wires or elements.
- you want to disperse data to several points in the routine.



To use connectors, follow these rules:

- Each OCON requires a unique name.
- For each OCON, you must have at least one corresponding ICON (for example, an ICON with the same name as the OCON).
- Multiple ICONs can reference the same OCON. This lets you disperse data to several points in your routine.

Define Program/Operator Control

Several instructions support the concept of Program/Operator control. These instructions include:

- Enhanced Select (ESEL)
- Totalizer (TOT)
- Enhanced PID (PIDE)
- Ramp/Soak (RMPS)
- Discrete 2-State Device (D2SD)
- Discrete 3-State Device (D3SD)

Program/Operator control lets you control these instructions simultaneously from both your user program and from an operator interface device. When in Program control, the instruction is controlled by the Program inputs to the instruction; when in Operator control, the instruction is controlled by the Operator inputs to the instruction.

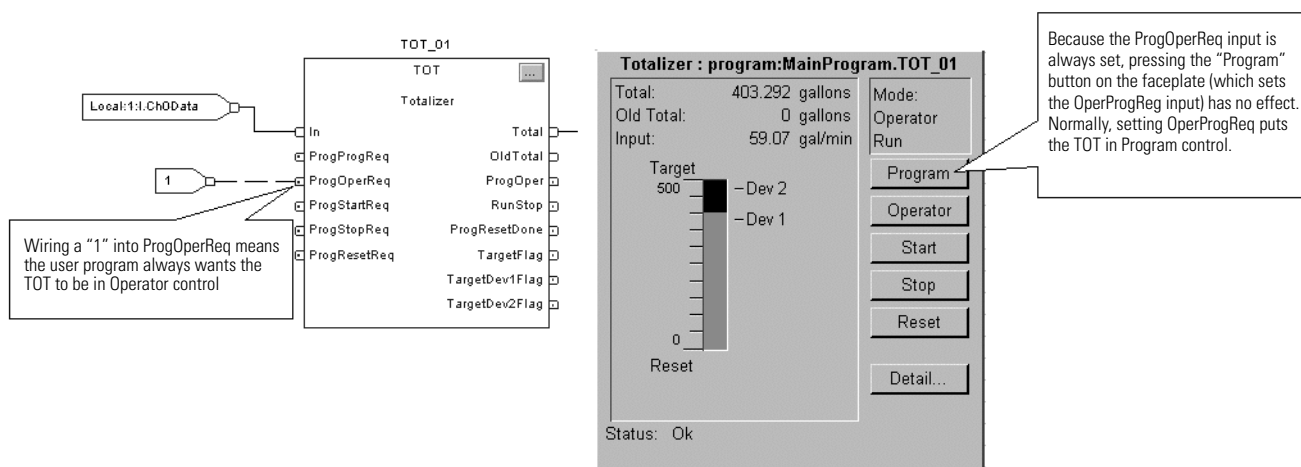
Program or Operator control is determined by using these inputs.

Input	Description
.ProgProgReq	A program request to go to Program control.
.ProgOperReq	A program request to go to Operator control.
.OperProgReq	An operator request to go to Program control.
.OperOperReq	An operator request to go to Operator control.

To determine whether an instruction is in Program or Control control, examine the ProgOper output. If ProgOper is set, the instruction is in Program control; if ProgOper is cleared, the instruction is in Operator control.

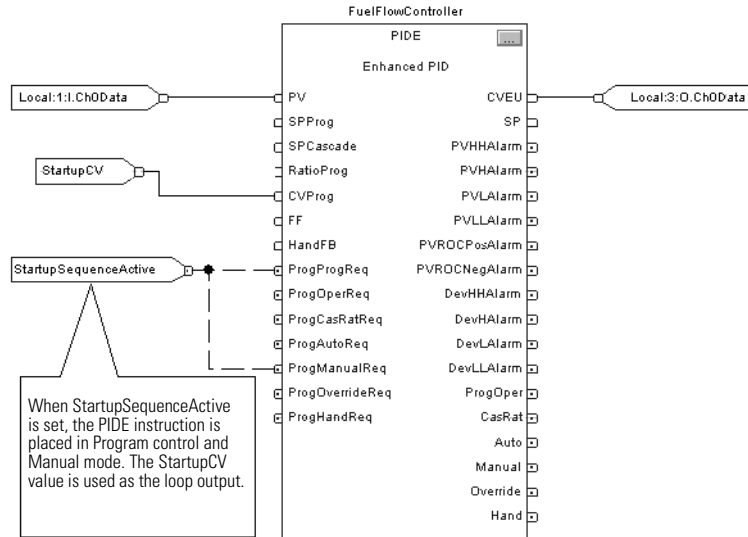
Operator control takes precedence over Program control if both input request bits are set. For example, if ProgProgReq and ProgOperReq are both set, the instruction goes to Operator control.

The Program request inputs take precedence over the Operator request inputs. This provides the capability to use the ProgProgReq and ProgOperReq inputs to “lock” an instruction in a desired control. For example, let’s assume that a Totalizer instruction will always be used in Operator control, and your user program will never control the running or stopping of the Totalizer. In this case, you could wire a literal value of 1 into the ProgOperReq. This would prevent the operator from ever putting the Totalizer into Program control by setting the OperProgReq from an operator interface device.



Likewise, constantly setting the ProgProgReq can “lock” the instruction into Program control. This is useful for automatic startup sequences when you want the program to control the action of the instruction without worrying about an operator inadvertently taking control of the instruction. In this example, you have the program set the ProgProgReq input during the startup, and then clear the ProgProgReq input once the startup was complete. Once the ProgProgReq input is cleared, the instruction remains in Program control until it receives a request to change. For example, the operator could set the

OperOperReq input from a faceplate to take over control of that instruction. This example shows how to lock an instruction into Program control.



Operator request inputs to an instruction are always cleared by the instruction when it executes. This lets operator interfaces work with these instructions by merely setting the desired mode request bit. You don't have to program the operator interface to reset the request bits. For example, if an operator interface sets the OperAutoReq input to a PIDE instruction, when the PIDE instruction executes, it determines what the appropriate response should be and clears the OperAutoReq.

Program request inputs are not normally cleared by the instruction because these are normally wired as inputs into the instruction. If the instruction clears these inputs, the input would just get set again by the wired input. There might be situations where you want to use other logic to set the Program requests in such a manner that you want the Program requests to be cleared by the instruction. In this case, you can set the ProgValueReset input and the instruction will always clear the Program mode request inputs when it executes.

In this example, a rung of ladder logic in another routine is used to one-shot latch a ProgAutoReq to a PIDE instruction when a pushbutton is pushed. Because the PIDE instruction automatically clears the Program mode requests, you don't have to write any ladder logic to clear the ProgAutoReq after the routine executes, and the PIDE instruction will receive only one request to go to Auto every time the pushbutton is pressed.

When the TIC101AutoReq Pushbutton is pressed, one-shot latch ProgAutoReq for the PIDE instruction TIC101. TIC101 has been configured with the ProgValueReset input set, so when the PIDE instruction executes, it automatically clears ProgAutoReq.



Add a Sheet

To add a sheet to a function block routine:

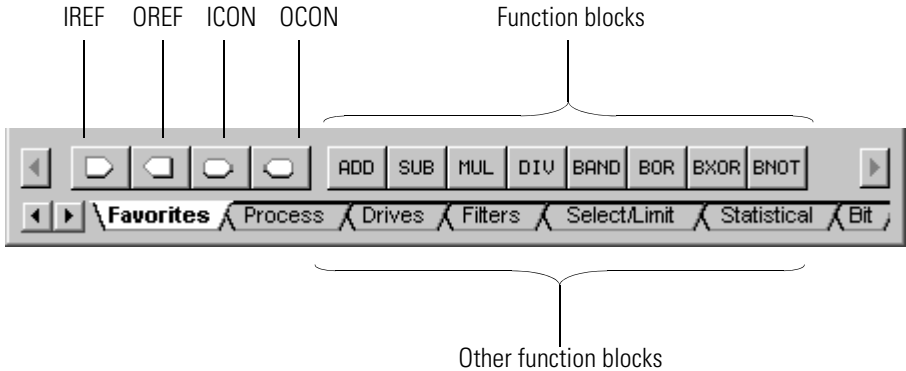
- 1. Click 



- 2. Type a description of the sheet (up to 50 characters).

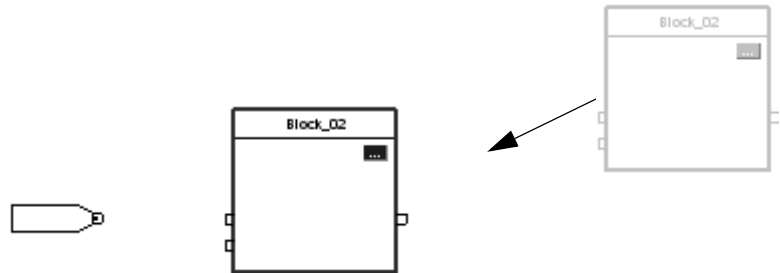
Add a Function Block Element

Use the Language Element toolbar to add a function block to a routine.

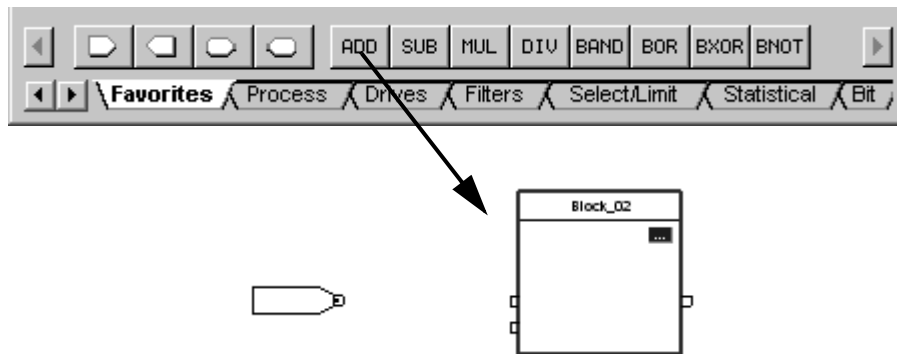


To add an element:

1. On the Language Element toolbar, click the button for the element that you want to add.
2. Drag the element to the desired location.



You can also drag the button for the element directly to the desired location.



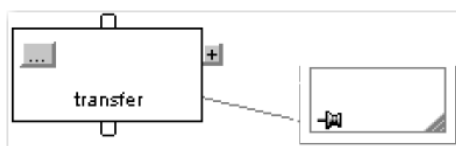
IMPORTANT

Use caution when copying and pasting components between different versions of RSLogix 5000 programming software. RSLogix 5000 software only supports pasting to the same version or newer version of RSLogix 5000. Pasting to a prior version of RSLogix 5000 software is not supported. When pasting to a prior version, the paste action may succeed but the results may not be as intended.

Create a Text Box

With a text box, you can add notes that clarify the function of an FBD element, such as blocks, references, and connectors. Text boxes are only stored in the offline, ACD project file. Text boxes are not downloaded into controller memory.

You can also use a text box to capture information that you will use later on, as shown in this example:



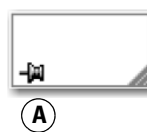
1. Click 


A text box appears.



2. Drag the text box to a location near the element to which it applies.
3. Double-click the text box and type the note.
4. Then press [Ctrl] + [Enter].
5. As you move the element on the FBD, what do you want the text box to do?

If you want the text box to	Then
Stay in the same spot	Stop. You are done.
Move with the element to which it applies	Go to step 6.



-  Green dot

6. Click the pin symbol in the text box and then click the FBD element to which you want to attach the text box.

A green dot shows a valid connection point.

Language Switching

With RSLogix 5000 software, version 17, you have the option to display project documentation, such as tag descriptions and rung comments for any supported localized language. You can store project documentation for multiple languages in a single project file rather than in language-specific project files. You define all the localized languages that the project will support and set the current, default, and optional custom localized language. The software uses the default language if the current language's content is blank for a particular component of the project. However, you can use a custom language to tailor documentation to a specific type of project file user.

Enter the localized descriptions in your RSLogix 5000 project, either when programming in that language or by using the import/export utility to translate the documentation off-line and then import it back into the project. Once you enable language switching in RSLogix 5000 software, you can dynamically switch between languages as you use the software.

Project documentation that supports multiple translations within a project includes:

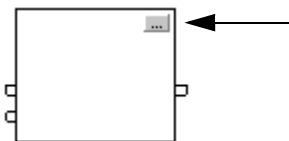
- Component descriptions in tags, routines, programs, user-defined data types, and Add-On Instructions.
- Equipment phases.
- Trends.
- Controllers.
- Alarm Messages (in ALARM_ANALOG and ALARM_DIGITAL configuration).
- Tasks.
- Property descriptions for modules in the Controller Organizer.
- Rung comments, SFC text boxes, and FBD text boxes.


For more information on enabling a project to support multiple translations of project documentation, see the online help.

Connect Elements

Show or Hide a Pin

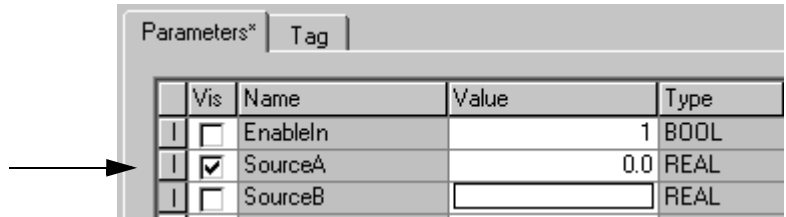
When you add a Function Block instruction, the block appears with a set of pins for the default parameters. The rest of the pins are hidden. To show or hide a pin:



1. Click the  button of the block.

2. Clear or check the Vis check box of the pin:

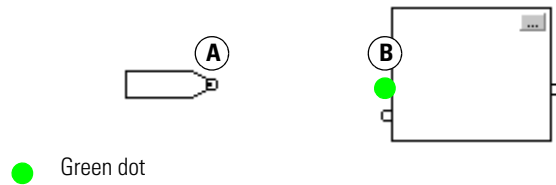
If you want to	Then
Hide a pin	Clear (uncheck) its Vis check box.
Show a pin	Check its Vis check box.



3. Click OK.

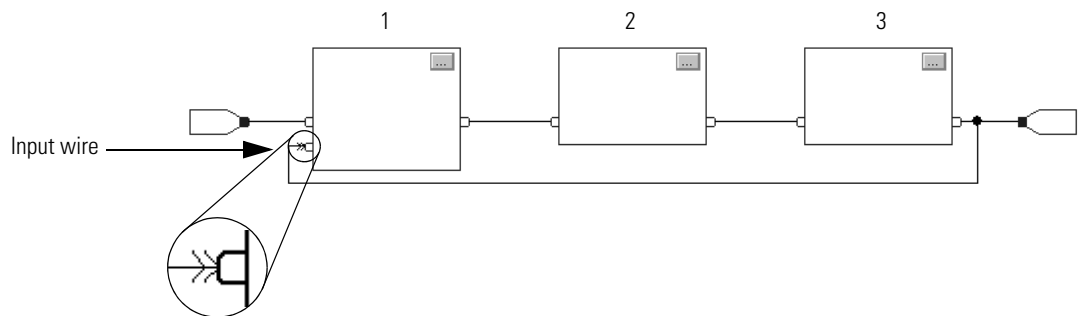
Wire Elements Together

To wire (connect) two elements together, click the output pin of the first element and then click the input pin of the other element. A green dot shows a valid connection point.

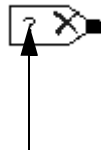


Mark a Wire with the Assume Data Available Indicator

To define a wire as an input, right-click the wire and choose Assume Data Available.



Assign a Tag



Create and Assign a New Tag

1. Double-click the operand area.
2. Type a name for the tag and press [Enter].
3. Right-click the tag name and choose New “tag_name”.

4. Click the [...] button.

5. Select the data type for the tag.
6. If the tag as an array, type the number of elements in each dimension.
7. Click OK.
8. Select the scope for the tag.

9. Click OK.

Assign an Existing Tag

Block_01 ←



1. Double-click the operand area.
2. Click the ▼.
3. Select the tag.

To select a:	Do this
Tag	Double-click the tag name.
Bit number	A. Click the tag name. B. To the right of the tag name, click ▼. C. Click the required bit.

4. Press [Enter] or click a different spot on the diagram.

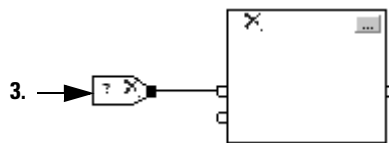
Assign an Immediate Value (Constant)

To assign a constant value instead of a tag value to an input parameter, you have these options:

If you want to	Then
Make the value visible on the diagram and reports	Use an IREF
Be able to change the value online without editing the routine	Enter a Value in the Tag of a Block routine

Use an IREF

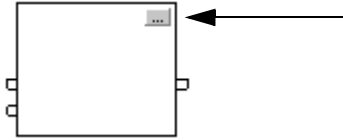
1. Add an IREF.
2. Wire the IREF to the input pin that gets the value.




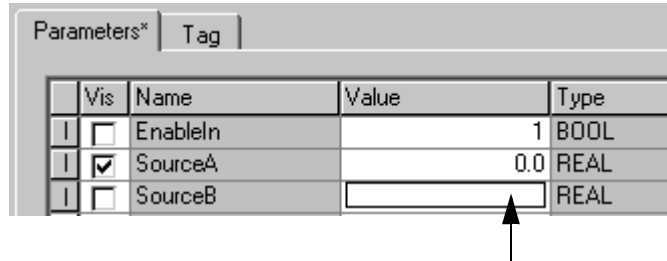
3. Double-click the operand area of the IREF.
4. Type the value and press [Enter].

Enter a Value in the Tag of a Block

To assign a value to a parameter when on wire connects to its pin:



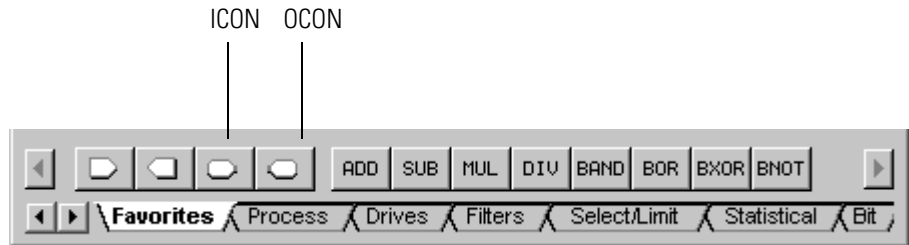
1. Click the  button of the block.



2. Type the value.
3. Click OK.

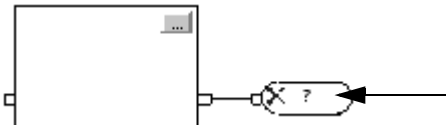
Connect Blocks with an OCON and ICON

Use an OCON or ICON to transfer data between sheets or in complex wiring situations:

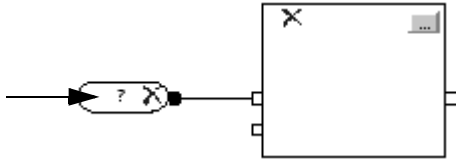


Add an OCON

1. Add an output wire connector (OCON) and place it near the output pin that supplies the value.
2. Wire the OCON to the output pin.
3. Double-click the operand area of the OCON.
4. Type a name that identifies the connector and press [Enter].




Add an ICON



1. Add an input wire connector (ICON) and place it near the input pin that gets the value from the corresponding OCON.
2. Wire the ICON to the input pin.
3. Double-click the operand area of the ICON.
4. Select the name of the OCON that supplies the value to this connector and then click a blank spot on the diagram

Verify the Routine

As you program your routine, periodically verify your work:

1. In the top-most toolbar of the RSLogix 5000 window, click 
2. If any errors are listed at the bottom of the window:
 - a. To go to the first error or warning, press [F4].
 - b. Correct the error according to the description in the Results window.
 - c. Go to step 1.
3. To close the Results window, press [Alt] + [1].

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, a knowledge base of FAQs, 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.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect 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 an anomaly 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/>.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

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

Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, 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-PM009C-EN-P - October 2009

Supersedes Publication 1756-PM009B-EN-P - July 2008

Copyright © 2009 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.