Reference Manual

Original Instructions

Allen-Bradley

MicroLogix Controllers to Micro800 Controllers Migration Guide

Catalog Numbers Bulletin 1761, Bulletin 1762, Bulletin 1763, and Bulletin 2080

Allen-Bradley • Rockwell Software

Rockwell Automation

Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

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

Labels may also be on or inside the equipment to provide specific precautions.

	**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.
	**ARC FLASH HAZARD:** Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

About This Publication

This document serves as a guide for replacing your existing MicroLogix™ 1000, MicroLogix 1100, or MicroLogix 1200 controller with a Micro800™ family of controllers.

The Micro800 family of controllers includes the Micro810®, Micro820®, Micro830®, Micro850®, and Micro870® controllers.

Descriptions, wiring diagrams, dimensions, features, and specifications of the controllers are provided to help you select the appropriate Micro800 controller to replace your MicroLogix controller.

This document shows you how to use the software tools to select a suitable Micro800 controller, and also how to convert your MicroLogix programs to work with the Micro800 controller.

Audience

The intended audience of this document is owners of MicroLogix 1000, MicroLogix 1100, and MicroLogix 1200 controllers who are migrating to the Micro800 family of controllers, and who are familiar with the RSLogix 500®/ RSLogix™ Micro programming software. Knowledge of programming in ladder language is expected to be able to program Micro800 systems effectively.

Required Software

To complete the steps in this document, Connected Components Workbench™ software version 12 or later is required. As the main programming software for Micro800 systems. Connected Components Workbench software provides a choice of IEC 61131-3 programming languages (ladder diagram, function block diagram, structured text) with user-defined function block support that optimizes machine control.

You need Connected Components Workbench software to write your ladder diagram, function block diagram, or structured text programs, to execute the programs, and to see the results.

This document uses two features that are available in Connected Components Workbench software version 12 or later.

- MicroLogix to Micro800 Converter tool

  The MicroLogix to Micro800 Converter tool converts an RSLogix 500/ RSLogix Micro project into a Connected Components Workbench project. It provides conversion for ladder diagram (LD) programming languages in the MicroLogix processor.

The onverter tool can convert most RSLogix 500/RSLogix Micro instruction blocks. However, you may need to modify the converted function blocks to confirm that they work properly. All information that requires additional modifications are logged in a conversion report, and this document shows you how to make the changes.

- Micro800 Simulator

  The Micro800 Simulator can be used to perform testing and troubleshooting of a Connected Components Workbench project, without a physical Micro800 controller.

Summary of Changes

Topic	Page
Updated Preface.	5
Combined dimensions for various Micro830 controllers. Added dimensions for MicroLogix 1100, MicroLogix 1200, Micro 850, and Micro870 controllers.	11, 12,14, 15, 16
Updated feature and specification comparison table for MicroLogix 1000 controllers. Added tables for MicroLogix 1100 and MicroLogix 1200 controllers.	17, 18, 19
Added chapter "Plan Hardware Migration with Integrated Architecture Builder".	21
Renamed chapter "Select a Suitable Micro800 Controller" to "Migration Considerations". Added information for migrating from a MicroLogix 1100 or MicroLogix 1200 controller.	25, 27, 29
Added wiring diagrams for MicroLogix 1100, MicroLogix 1200, Micro850, and Micro870 controllers.	39, 42, 54, 56
Renamed chapter "Convert an RSLogix 500 Project to a Connected Components Project" to "Convert a MicroLogix Project to a Micro800 Project". Revised chapter with new information on the conversion process and the use of the MicroLogix to Micro800 Converter tool and Micro800 Simulator.	59, 65, 79, 84, 91
Updated High-Speed Counter instruction description with information of new HSC instruction set in Connected Components Workbench software.	173
Moved some examples from chapter "Convert a MicroLogix Project to a Micro800 Project" into a new appendix.	179

Additional Resources

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

Resource	Description
Micro800 Programmable Controllers General Instructions, publication 2080-RM001	Provides reference information about the instruction set available for developing programs for use in Micro800 control systems.
Micro820 Programmable Controllers User Manual, publication 2080-UM005	A more detailed description of how to install and use your Micro820 programmable controllers.
Micro830, Micro850, and Micro870 Programmable Controllers User Manual, publication 2080-UM002	A more detailed description of how to install and use your Micro830, Micro850, and Micro870 programmable controllers.
Micro800 Expansion Modules User Manual, publication 2080-UM003	Description of features, installation, wiring, and specifications for the Micro800 expansion modules.
Micro800 Plug-in Modules User Manual, publication 2080-UM004	Description of features, installation, wiring, and specifications for the Micro800 plug-in modules.

Resource	Description
Getting Started with Motion Control Using a Simulated Axis Quick Start, publication 2080-QS001	Provides instructions to implement a motion control project using Connected Components Workbench software.
Micro800 Controllers: Getting Started with CIP Client Messaging Quick Start, publication 2080-QS002	Provides instructions on how to use CIP Generic and CIP Symbolic messaging with Micro800 controllers.
Micro800 Programmable Controllers: Getting Started with PanelView Plus Quick Start, publication 2080-QS003	Provides instructions on how to use global variables with Micro800 controllers together with PanelView™ Plus HMI terminals.
Setup Micro800 Controllers on FactoryTalk Gateway Quick Start, publication 2080-QS005	Provides instructions on how to configure a Micro800 controller on FactoryTalk® Gateway.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, https://www.rockwellautomation.com/global/certification/overview.page	Provides declarations of conformity, certificates, and other certification details.

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

Notes:

Micro800 Controller Overview

Micro800 controllers are designed for low-cost, standalone machines. These economical small-size programmable logic controllers (PLCs) are available in different form factors based on the number of I/O points that are embedded in the base, with a range of features that are intended to address different requirements. The Micro800 family shares programming environment, accessories, and plug-ins that allow machine builders to personalize the controller for specific capabilities.

Micro810 controllers function as a smart relay with high current relay outputs, but with the programming capabilities of a micro PLC. The Micro810 controllers come in a 12-point form factor.

Micro820 controllers are designed for smaller standalone machines and remote automation projects. They have embedded Ethernet and serial ports and a microSD™ slot for data logging and recipe management. These controllers come as 20-point form factors that can accommodate up to two plug-in modules. They also support the Micro800 Remote LCD (2080-REMLCD) module for easier configuration of settings such as IP address. The Remote LCD module can also function as a simple IP65 text display.

Micro830 controllers are designed for standalone machine control applications. They have flexible communications and I/O capabilities with up to five plug-ins. They come in 10-, 16-, 24-, or 48-point form factors.

Micro850 expandable controllers are designed for applications that require more digital and analog I/O or higher performance analog I/O. They can support up to four expansion I/O. Micro850 controllers include additional communication connection options through an embedded 10/100 Base-T Ethernet port.

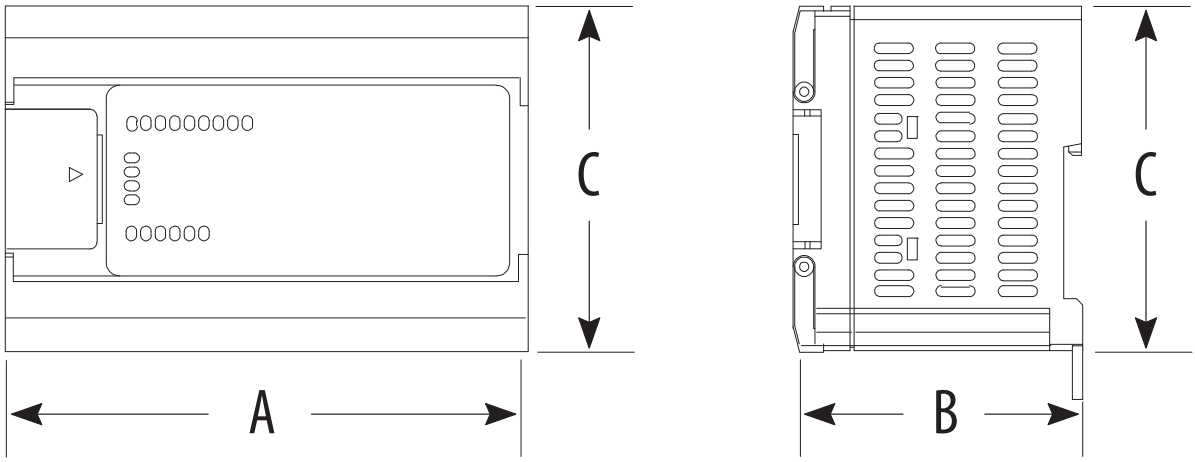
Micro870 controllers offer machine builders and end users a higher level of scalability, flexibility, and customization. Designed for large standalone machine applications, the Micro870 controller comes with great memory capacity to enable more modular program and use of user-defined function blocks. They can support up to eight expansion I/O.

Several Micro830, Micro850, and Micro870 controllers support basic positioning through embedded pulse train outputs (PTO). These controllers also allow you to configure up to six high-speed counters (HSC), and choose from nine HSC operation modes. HSC is supported on all Micro830, Micro850, and Micro870 controller catalogs, except on 2080-LCxx-xxAWB. PTO is only supported on Micro830, Micro850, and Micro870 controller catalog numbers that end in BB or VB.

Controller Dimensions

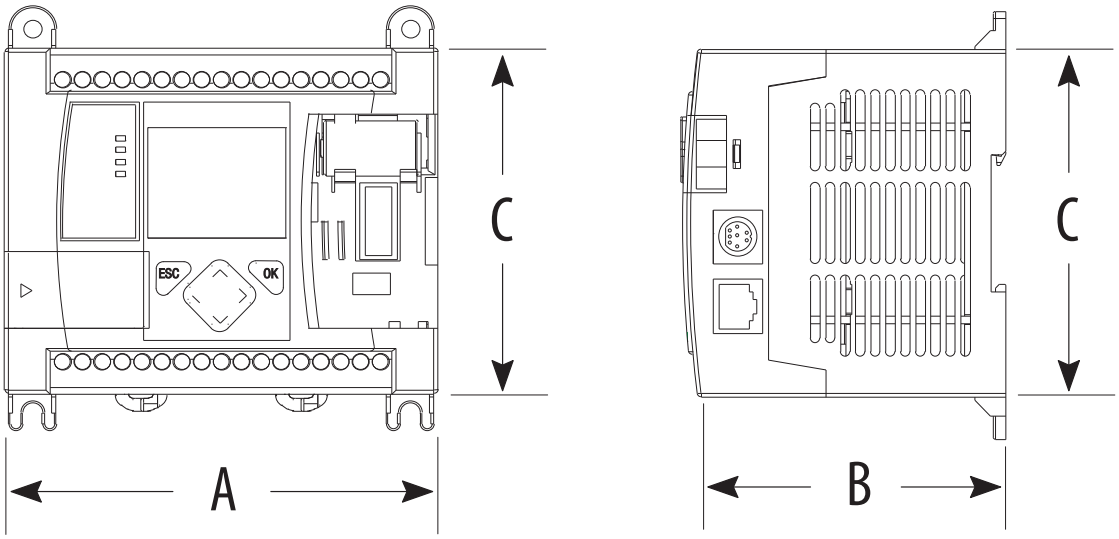
The following tables describe the dimensions for the MicroLogix controllers and the Micro800 controllers.

MicroLogix 1000 Controller Dimensions



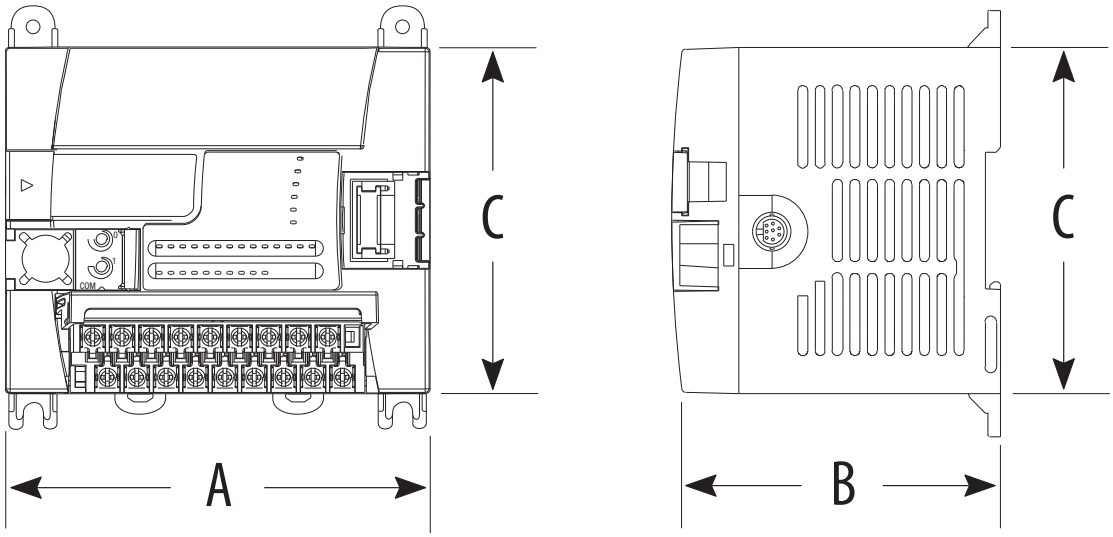
Catalog Number	A	B	C
1761-L10BWA	120 mm (4.72 in.)	73 mm (2.87 in.)	80 mm (3.15 in.)
1761-L16AWA	133 mm (5.24 in.)		
1761-L16BWA	120 mm (4.72 in.)		
1761-L16NWA			
1761-L20AWA-5A	200 mm (7.87 in.)		
1761-L20BWA-5A			
1761-L32AWA			
1761-L32BWA			
1761-L32AAA			
1761-L10BWB	120 mm (4.72 in.)	40 mm (1.57 in.)	
1761-L10BXB			
1761-L16BBB			
1761-L16BWB			
1761-L16NWB			
1761-L20BWB-5A	200 mm (7.87 in.)		
1761-L32BBB			
1761-L32BWB			

MicroLogix 1100 Controller Dimensions



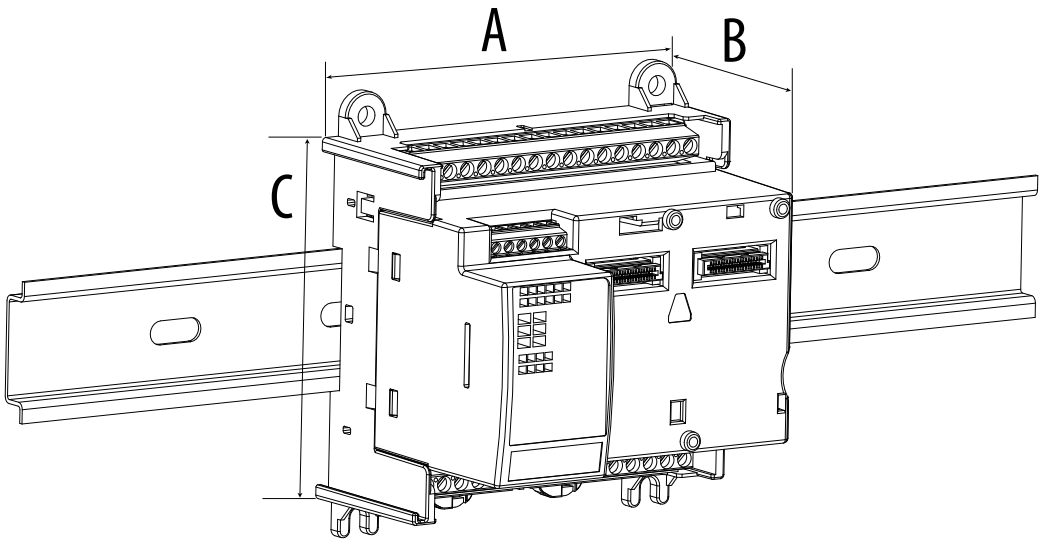
Catalog Number	A	B	C
1763-L16AWA	110 mm (4.33 in.)	87 mm (3.43 in.)	90 mm (3.54 in.)
1763-L16BWA			
1763-L16BBB			
1763-L16DWD			

MicroLogix 1200 Controller Dimensions



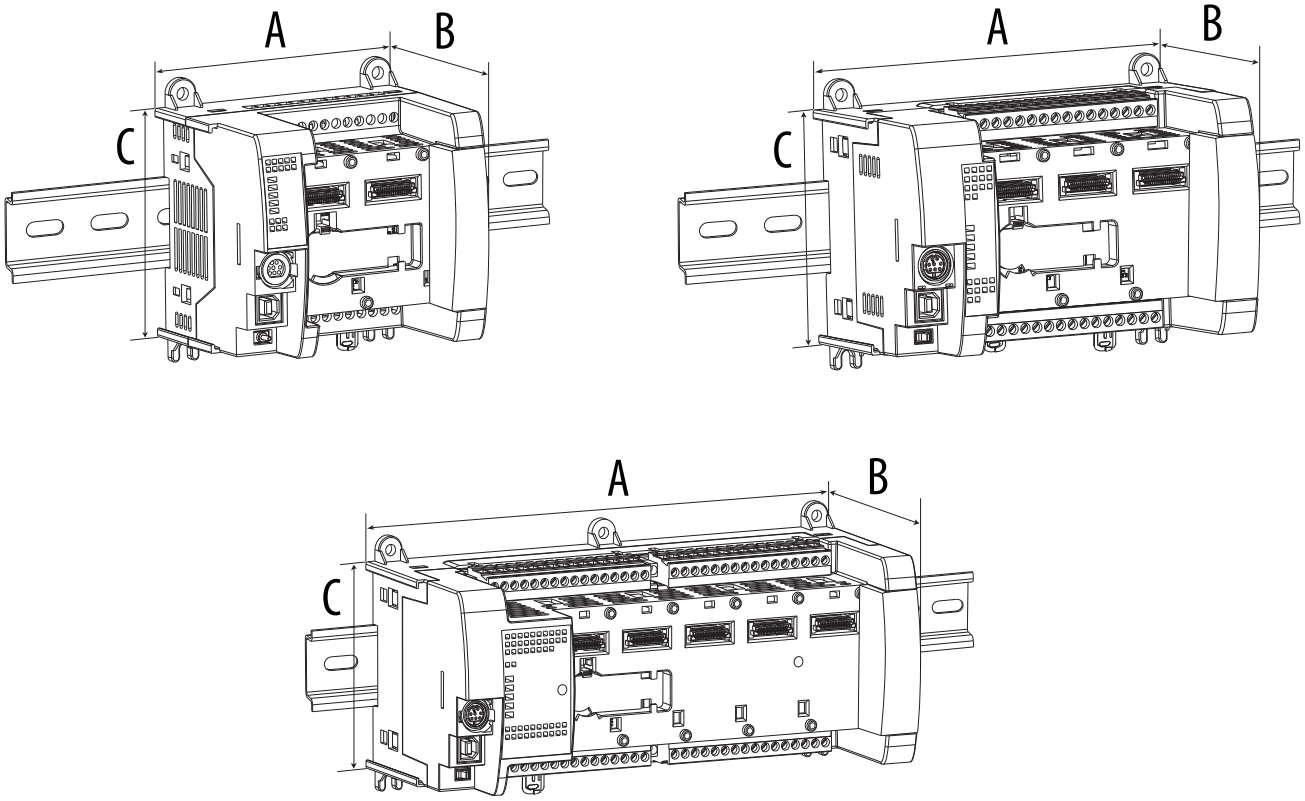
Catalog Number	A	B	C
1762-L24AWA	110 mm (4.33 in.)	87 mm (3.43 in.)	90 mm (3.54 in.)
1762-L24AWAR			
1762-L24BWA			
1762-L24BWAR			
1762-L24BXB			
1762-L24BXBR			
1762-L40AWA	160 mm (6.30 in.)		
1762-L40AWAR			
1762-L40BWA			
1762-L40BWAR			
1762-L40BXB			
1762-L40BXBR			

Micro820 Controller Dimensions



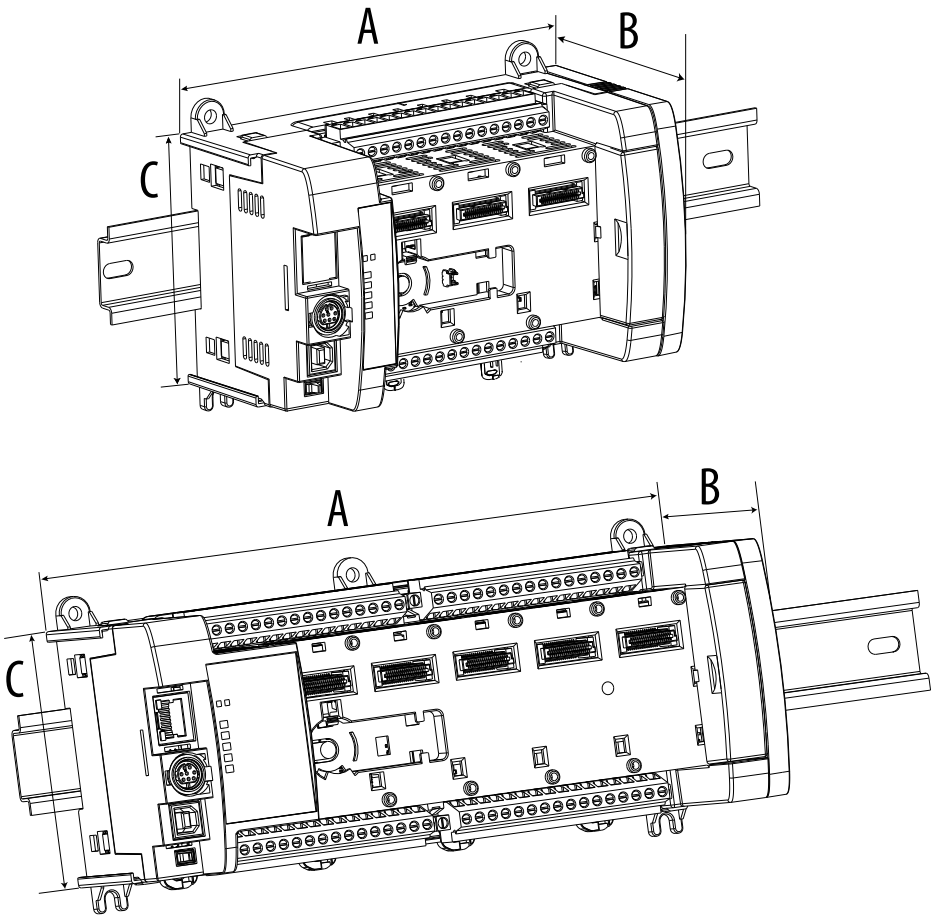
Catalog Number	A	B	C
2080-LC20-20AWB	104 mm (4.09 in.)	75 mm (2.95 in.)	90 mm (3.54 in.)
2080-LC20-20AWBR			
2080-LC20-20QWB			
2080-LC20-20QWBR			
2080-LC20-20QBB			
2080-LC20-20QBBR			

Micro830 Controller Dimensions



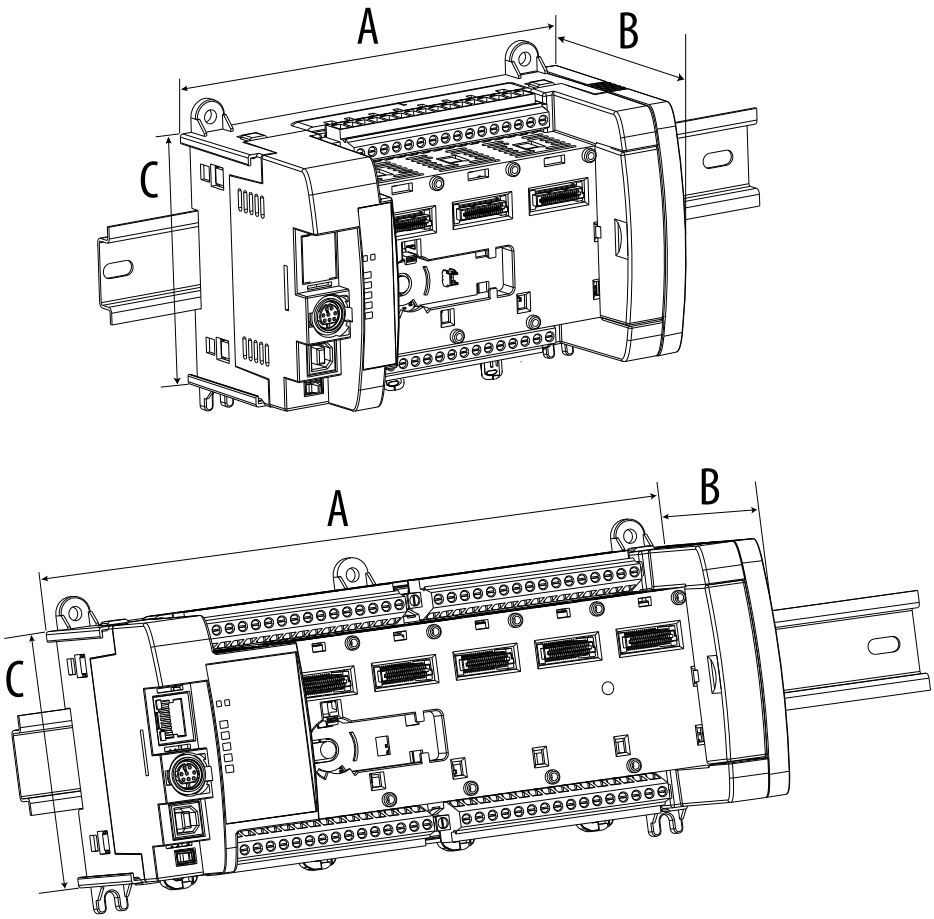
Catalog Number	A	B	C
2080-LC30-10QWB	100 mm (3.94 in.)	80 mm (3.15 in.)	90 mm (3.54 in.)
2080-LC30-10QVB			
2080-LC30-16AWB			
2080-LC30-16QWB			
2080-LC30-16QVB			
2080-LC30-24QWB	150 mm (5.91 in.)		
2080-LC30-24QVB			
2080-LC30-24QBB			
2080-LC30-48AWB	210 mm (8.27 in.)		
2080-LC30-48QWB			
2080-LC30-48QVB			
2080-LC30-48QBB			

Micro850 Controller Dimensions



Catalog Number	A	B	C
2080-LC50-24AWB	158 mm (6.22 in.)	80 mm (3.15 in.)	90 mm (3.54 in.)
2080-LC50-24QWB			
2080-LC50-24QVB			
2080-LC50-24QBB			
2080-LC50-48AWB	283 mm (9.37 in.)		
2080-LC50-48QWB			
2080-LC50-48QVB			
2080-LC50-48QBB			

Micro870 Controller Dimensions



Catalog Number	A	B	C
2080-LC70-24AWB	157 mm (6.22 in.)	80 mm (3.15 in.)	90 mm (3.54 in.)
2080-LC70-24QWB			
2080-LC70-24QWBK			
2080-LC70-24QBB			
2080-LC70-24QBBK			

Feature and Specification Comparison

The following tables describe the differences in features and specifications between MicroLogix controllers and Micro800 controllers. For more details on the specifications, see the respective controller user manual.

MicroLogix 1000 Controllers and Micro800 Controllers Comparison

Features	MicroLogix 1000 Controller	Micro820 Controller	Micro830 Controller
Memory			
Memory (in user words) User program/User data	1 KB combined (preconfigured)	10/20 KB	4/8 KB – 10/16-point controllers 10/20 KB – 24/48-point controllers
Memory module (for program backup and transport)	Handheld programmer	MicroSD card(1)	Plug-in module – 2080-MEMBAK-RTC or 2080-MEMBAK-RTC2
Online editing/Run Mode Change	None	Yes(2)	
Inputs / Outputs			
Embedded digital I/O, max	21	19	48
Embedded analog I/O	Two current and two voltage inputs with one current or voltage output on 20 point controllers	One 0…10V analog output, four 24V DC digital inputs that can be configured as 0…10V analog inputs (DC input controllers only), and plug-in module – 2080-IF2, 2080-IF4	Plug-in module – 2080-IF2, 2080-IF4
Expansion I/O supported	None		
Thermocouple/RTD	None	Plug-in module – 2080-RTD2, 2080-TC2	
Network expansion I/O	None	Plug-in module – 2080-DNET20 (up to 20 nodes for I/O operation)	
Added Functionality			
Trim potentiometer	None	Plug-in module – 2080-TRIMPOT6	
PID	None	Yes (limited only by memory and I/O)	
High-speed counters (embedded)	1 @ 6.6 kHz (not supported on AC input controllers)	Plug-in module – 2080-MOT-HSC	2 @100 kHz – 10/16-point controllers 4 @100 kHz – 24-point controllers 6 @100 kHz – 48-point controllers (not supported on AC input controllers)
Motion: PTO/PWM support	None	PWM only 1 @ 5.5 kHz (not supported on relay output controllers)	1 @ 100 kHz – 10/16-point controllers 2 @ 100 kHz – 24-point controllers 3 @ 100 kHz – 48-point controllers (not supported on relay output controllers)
Real-time clock	None	Embedded	Plug-in module – 2080-MEMBAK-RTC, 2080-MEMBAK-RTC2
Recipe storage	None	MicroSD card(1)	Plug-in module – 2080-SDMEMRTC-SC and microSD card(1)
Data logging	None	MicroSD card(1)	Plug-in module – 2080-SDMEMRTC-SC and microSD card(1)
Floating point math	None	32-bit and 64-bit	
Operating Power			
120/240V AC	Yes	Power supply module – 2080-PSAC-12W	Power supply module – 2080-PS120-240VAC
24V DC	Yes		
Communication			
RS-232 port	8-pin mini DIN	Embedded RS-232/RS-485 serial port combo	8-pin min DIN RS-232/RS-485 serial port combo

Features	MicroLogix 1000 Controller	Micro820 Controller	Micro830 Controller
DeviceNet Peer-to-Peer Messaging, Slave I/O	None	Plug-in module – 2080-DNET20 (up to 20 nodes for I/O operation)	
EtherNet/IP	With 1761-NET-ENI or 1761-NET-ENIW	Yes	None
DH-485	With 1761-NET-AIC	None	
SCADA RTU – DF1 Half-duplex Slave	Yes	None	
SCADA RTU – DF1 Radio Modem	None		
SCADA RTU – Modbus RTU Slave	None	Yes	
SCADA RTU – Modbus RTU Master	None	Yes	
Modbus TCP	None	Yes	None
ASCII – Read/Write	None	Yes	
CIP Serial	None	Yes	

(1) We recommend using the Allen-Bradley 2080-SD-2GB microSD card. The 2080-SDMEMRTC-SC plug-in module is an Encompass™ partner product.

(2) Requires Connected Components Workbench Developer Edition software version 12 or later, and Micro800 controller firmware revision 12 or later.

MicroLogix 1100 Controllers and Micro800 Controllers Comparison

Features	MicroLogix 1100 Controller	Micro820 Controller	Micro850 Controller
Memory			
Memory (in user words) User program/User data	4 KB user program with 4 KB user data	120 KB user program with 20 KB user data[1]	
Memory module (for program backup and transport)	1763-MM1 memory module	MicroSD card[2]	Plug-in module – 2080-MEMBAK-RTC, 2080-MEMBAK-RTC2
Online editing/Run Mode Change	Yes	Yes[3]	
Inputs / Outputs			
Embedded digital I/O, max	16	19	48
Embedded analog I/O	Two 0…10V analog inputs	One 0…10V analog output, four 24V DC digital inputs that can be configured as 0…10V analog inputs (DC input controllers only), and plug-in module – 2080-IF2, 2080-IF4	Plug-in module – 2080-IF2, 2080-IF4
Expansion modules supported	Up to four expansion modules	None	Up to four expansion modules
Thermocouple/RTD	Expansion module – 1762-IT4, 1762-IR4	None	Expansion module – 2085-IRT4
Network expansion I/O	None	Plug-in module – 2080-DNET20 (up to 20 nodes for I/O operation)	
Added Functionality			
Trim potentiometer	LCD and keypad	Plug-in module – 2080-TRIMPOT6	
PID	Yes (limited only by memory and I/O)		
High-speed counters (embedded)	1 @ 40 kHz	Plug-in module – 2080-MOT-HSC	4 @ 100 kHz – 24-point controllers 6 @ 100 kHz – 48-point controllers (not supported on AC input controllers)
Motion: PTO/PWM support	1763-L16BBB only 2 @ 40 kHz	PWM only 1 @ 5.5 kHz (not supported on relay output controllers)	PTO only 2 @ 100 kHz – 24-point controllers 3 @ 100 kHz – 48-point controllers (not supported on relay output controllers)
Real-time clock	Embedded	Embedded	Plug-in module – 2080-MEMBAK-RTC, 2080-MEMBAK-RTC2

Features	MicroLogix 1100 Controller	Micro820 Controller	Micro850 Controller
Recipe storage	Yes	MicroSD card(2)	Plug-in module – 2080-SDMEMRTC-SC and microSD card(2)
Data logging	Yes	MicroSD card(2)	Plug-in module – 2080-SDMEMRTC-SC and microSD card(2)
Floating point math	32-bit	32-bit and 64-bit	
Operating Power			
120/240V AC	Yes	Power supply module – 2080-PSAC-12W	Power supply module – 2080-PS120-240VAC
24V DC	Yes		
12V DC	Yes	None	
Communication			
RS-232/485 port	8-pin mini DIN (isolated)	Plug-in module – 2080-SERIALISOL (isolated) or 6-pin terminal block (non-isolated)	Plug-in module – 2080-SERIALISOL (isolated) or 8-pin min DIN (non-isolated)
DeviceNet Peer-to-Peer Messaging, Slave I/O	None	Plug-in module – 2080-DNET20 (up to 20 nodes for I/O operation)	
EtherNet/IP	Yes		
DH-485	Yes	None	
SCADA RTU – DF1 Half-duplex Slave	Yes	None	
SCADA RTU – DF1 Radio Modem	Yes	None	
SCADA RTU – Modbus RTU Slave	Yes		
SCADA RTU – Modbus RTU Master	Yes		
Modbus TCP	None	Yes	
ASCII – Read/Write	Yes		
CIP Serial	None	Yes	

(1) For a similar program, a Micro800 program appears to be about five times larger than a MicroLogix program. However Micro820, Micro850, and Micro870 controllers have over 160 KB of memory. Based on an allocation of 120 KB for user programs, their effective memory is about four times larger than a MicroLogix controller.

(2) We recommend using the Allen-Bradley 2080-SD-2GB microSD card. The 2080-SDMEMRTC-SC plug-in module is an Encompass partner product.

(3) Requires Connected Components Workbench Developer Edition software version 12 or later, and Micro800 controller firmware revision 12 or later.

MicroLogix 1200 Controllers and Micro800 Controllers Comparison

Features	MicroLogix 1200 Controller	Micro850 Controller	Micro870 Controller
Memory			
Memory (in user words) User program/User data	6 KB (3 KB user program with 3 KB user data	120 KB user program with 20 KB user data(1)	240 KB user program with 40 KB user data(1)
Memory module (for program backup and transport)	Yes, 1762-MM1 or 1762-MM1RTC	Plug-in module – 2080-MEMBAK-RTC, 2080-MEMBAK-RTC2	
Run Mode Change	None	Yes(2)	
Inputs / Outputs			
Embedded digital I/O, max	40	48	24
Embedded analog I/O	None		
Expansion modules supported	Up to six expansion modules	Up to four expansion modules	Up to eight expansion modules
Thermocouple/RTD	Expansion module – 1762-IT4, 1762-IR4	Expansion module – 2085-IRT4	

Features	MicroLogix 1200 Controller	Micro850 Controller	Micro870 Controller
Network expansion I/O	None	Plug-in module – 2080-DNET20 (up to 20 nodes for I/O operation)	
Added Functionality			
Trim potentiometer	Two built-in digital trim potentiometers	Plug-in module – 2080-TRIMPOT6	
PID	Yes (limited only by memory and I/O)		
High-speed counters (embedded)	Up to four high-speed DC inputs	4 @ 100 kHz – 24-point controllers 6 @ 100 kHz – 48-point controllers (not supported on AC input controllers)	4 @ 100 kHz (not supported on AC input controllers)
Motion: PTO/PWM support	1 @ 20 kHz (supported by 1762-LxxBXB and 1762-LxxBXBR controllers only)	PTO only 2 @ 100 kHz – 24-point controllers 3 @ 100 kHz – 48-point controllers (not supported on relay output controllers)	PTO only 2 @ 100 kHz (not supported on relay output controllers)
Real-time clock	Yes, 1762-RTC or 1762-MM1RTC	Plug-in module – 2080-MEMBAK-RTC, 2080-MEMBAK-RTC2	Plug-in module – 2080-MEMBAK-RTC2
Recipe storage	None	Plug-in module – 2080-SDMEMRTC-SC and microSD card(3)	
Data logging	None	Plug-in module – 2080-SDMEMRTC-SC and microSD card(3)	
Floating point math	32-bit	32-bit and 64-bit	
Operating Power			
120/240V AC	Yes	Power supply module – 2080-PS120-240VAC	
24V DC	Yes		
Communication			
RS-232/485 port	8-pin mini DIN (isolated)	Plug-in module –2080-SERIALISOL (isolated) or 8-pin mini DIN (non-isolated)	
DeviceNet Peer-to-Peer Messaging, Slave I/O	None	Plug-in module – 2080-DNET20 (up to 20 nodes for I/O operation)	
EtherNet/IP	None	Yes	
DH-485	Yes	None	
SCADA RTU – DF1 Half-duplex Slave	Yes	None	
SCADA RTU – DF1 Radio Modem	Yes	None	
SCADA RTU – Modbus RTU Slave	Yes		
SCADA RTU – Modbus RTU Master	Yes		
Modbus TCP	None	Yes	
ASCII – Read/Write	Yes		
CIP Serial	None	Yes	

(1) For a similar program, a Micro800 program appears to be about five times larger than a MicroLogix program. However Micro820, Micro850, and Micro870 controllers have over 160 KB of memory. Based on an allocation of 120 KB for user programs, their effective memory is about four times larger than a MicroLogix controller.

(2) Requires Connected Components Workbench Developer Edition software version 12 or later, and Micro800 controller firmware revision 12 or later.

(3) We recommend using the Allen-Bradley 2080-SD-2GB microSD card. The 2080-SDMEMRTC-SC plug-in module is an Encompass partner product.

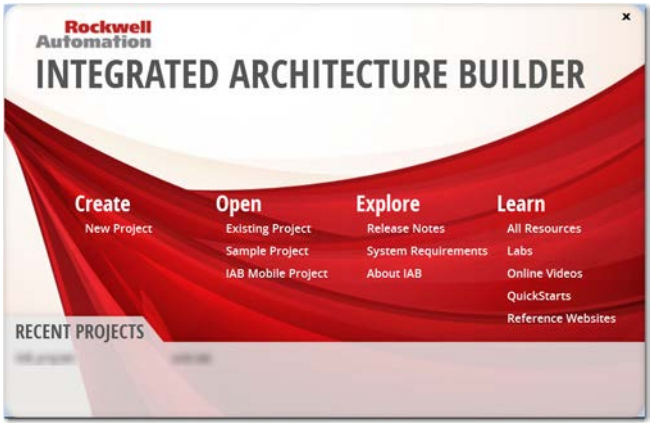
Plan Hardware Migration with Integrated Architecture Builder

This chapter describes how to use the MicroLogix Migration Wizard within Integrated Architecture® Builder (IAB) software to assist with converting your MicroLogix controller to a compatible controller. At the base level, MicroLogix 1000, MicroLogix 1100, and MicroLogix 1200 controllers migrate to Micro800 controllers. MicroLogix 1500 controllers migrate to MicroLogix 1400 controllers or CompactLogix™ (L1/L2) controllers.

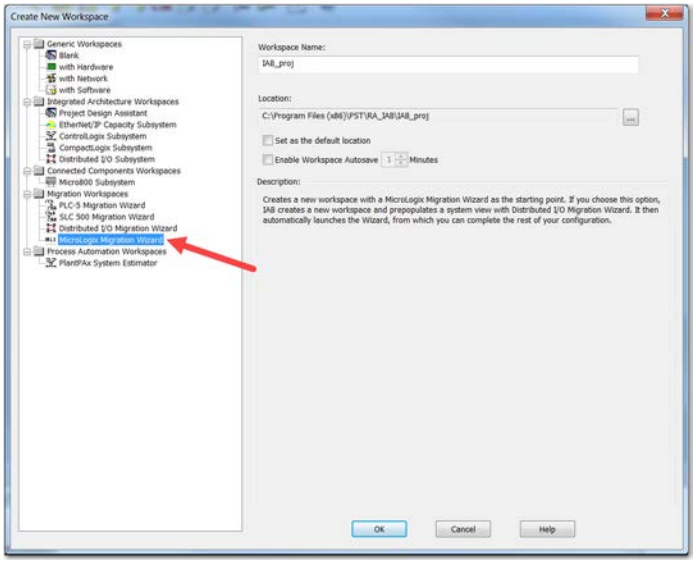
Generate Hardware Configuration

To convert your MicroLogix system to a compatible controller system, do the following:

1. Launch Integrated Architecture Builder software from Start -> Programs -> Rockwell Automation -> Integrated Architecture Builder -> Integrated Architecture Builder.
   Alternatively, you can double-click the IAB icon on your computer.
2. Under Create, click New Project.

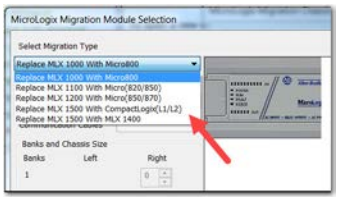


3. Enter a name into the Workspace Name field, such as 'MicroLogix Migration Wizard', and click OK.

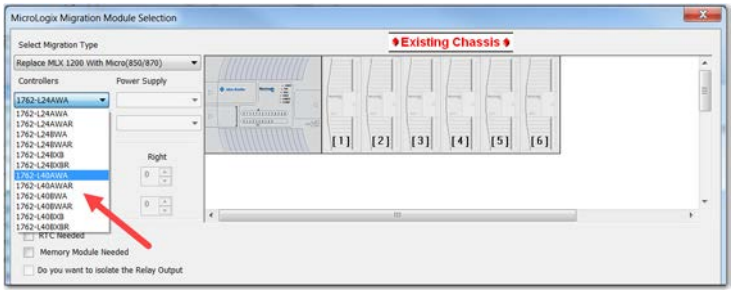


4. Click Add Chassis to add your MicroLogix configuration.

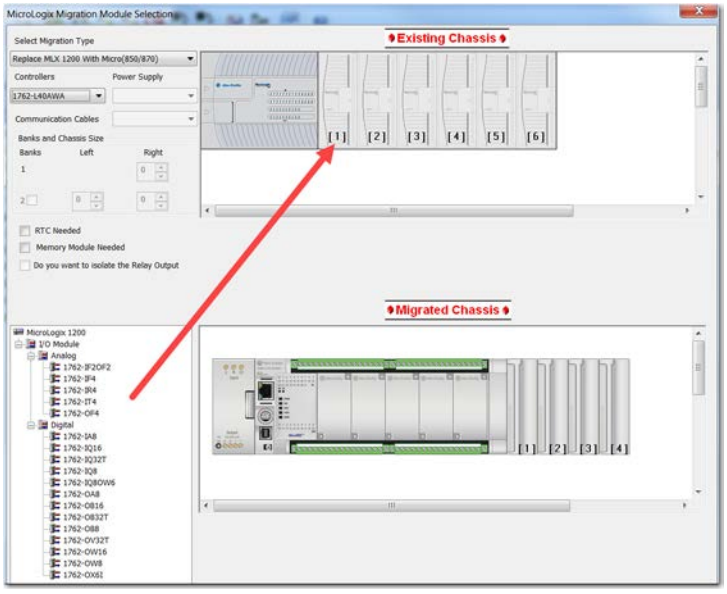
5. Select the Migration Type based on your MicroLogix controller series.



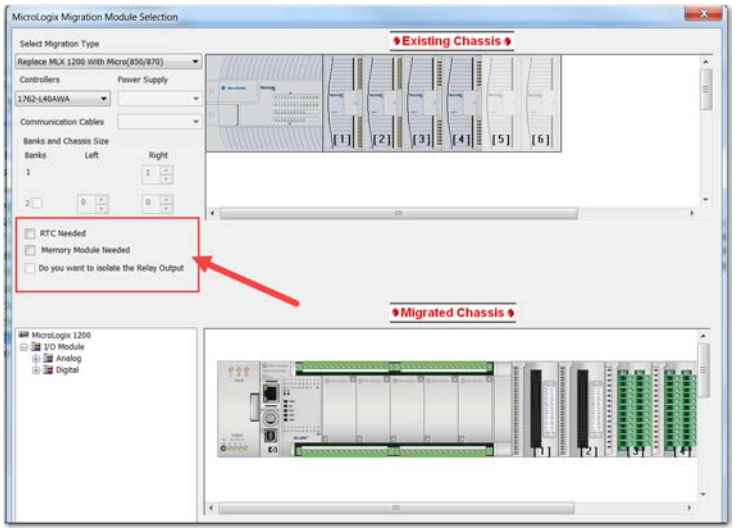
6. Select your MicroLogix controller catalog number.



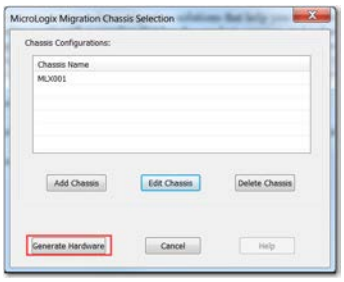
7. Add expansion modules, if any, to match your configuration.
   If the configuration exceeds the limit of the target controller, the wizard prompts you to change the target controller.



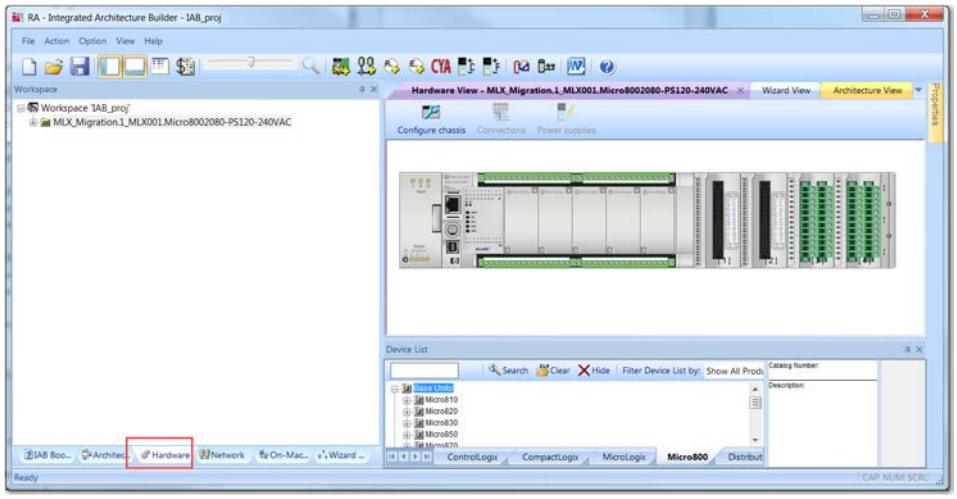
8. Select the options that apply to your MicroLogix application.



9. When the configuration is complete, click OK.
10. Click Generate Hardware.



11. Once the hardware is generated, you can view your configuration in the Hardware tab.



12. Click the Save icon to save your project.

Migration Considerations

This chapter describes how to optimize the configuration that you generated in the previous chapter. Use the wiring diagrams provided for reference.

Migrate From a MicroLogix 1000 Controller

This section helps you determine how to select and wire a suitable Micro800 controller (either a Micro820 or a Micro830 controller) for your existing MicroLogix 1000 controller wiring configuration.

Any MicroLogix 1000 DC input can be configured as sinking or sourcing depending on how the DC COM terminal is wired.

IMPORTANT For applications that require High Speed Counter (HSC) function, migrate to a Micro830 controller. See Convert to a Micro830 Controller on page 26 for more information.

Convert to a Micro820 Controller

Check the following table to see which Micro820 controller is suitable to replace your MicroLogix 1000 controller. Click the catalog number link to see the applicable wiring configuration.

MicroLogix 1000 Controller	Micro820 Controller	Plug-in Modules / Accessories
1761-L10BWB 1761-L16BWB 1761-L16NWB	2080-LC20-20QWB, 2080-LC20-20QWBR	—
1761-L32BWB	—	—
1761-L10BWA 1761-L16BWA 1761-L16NWA	2080-LC20-20QWB, 2080-LC20-20QWBR	2080-PSAC-12W x 1
1761-L16AWA 1761-L32AAA 1761-L32AWA 1761-L32BWA	—	—
1761-L10BXB 1761-L16BBB	2080-LC20-20QWB, 2080-LC20-20QWBR	2080-OB4 x 1
1761-L20BWA-5A 1761-L20AWA-5A 1761-L20BWB-5A 1761-L32BBB	—	—

IMPORTANT An external power supply is required when migrating from any **1761-L*A** controller to a Micro820 controller.

- For more information on the 2080-PSAC-12W power supply, see publication 2080-IN011.
- For more information on the 2080-OB4 digital output, see publication 2080-WD011.

Convert to a Micro830 Controller

Check the following table to see which Micro830 controller is suitable to replace your MicroLogix 1000 controller. Click the catalog number link to see the applicable wiring configuration.

MicroLogix 1000 Controller	Micro830 Controller	Plug-in Modules / Accessories
1761-L10BWB	2080-LC30-10QWB	—
1761-L16BWB	2080-LC30-16QWB	—
1761-L16NWB	2080-LC30-16QWB	—
1761-L32BWB	2080-LC30-48QWB	—
1761-L10BWA	2080-LC30-10QWB	2080-PS120-240VAC x 1
1761-L16BWA	2080-LC30-16QWB	
1761-L16NWA	2080-LC30-16QWB	
1761-L16AWA	2080-LC30-16AWB	
1761-L32AAA	2080-LC30-48AWB	
1761-L32AWA	2080-LC30-48AWB	
1761-L32BWA	2080-LC30-48QWB	
1761-L10BXB	2080-LC30-10QWB	2080-OB4 x 1
1761-L16BBB	2080-LC30-16QWB	
1761-L20BWA-5A	2080-LC30-24QWB	2080-PS120-240VAC x 1 2080-IF4 x 1 2080-OF2 x 1
1761-L20AWA-5A	2080-LC30-48AWB	
1761-L20BWB-5A	2080-LC30-24QWB	2080-IF4 x 1 2080-OF2 x 1
1761-L32BBB	2080-LC30-48QBB	2080-OW4I x 1

IMPORTANT An external power supply is required when migrating from any **1761-L*A** controller to a Micro830 controller.

- For more information on the 2080-PS120-240VAC power supply, see publication 2080-IN001.
- For more information on the 2080-IF4 analog input, see publication 2080-WD003.
- For more information on the 2080-OF2 analog output, see publication 2080-WD004.
- For more information on the 2080-OW4I relay output, see publication 2080-WD010.
- For more information on the 2080-OB4 digital output, see publication 2080-WD011.

Migrate From a MicroLogix 1100 Controller

This section helps you determine how to select and wire a suitable Micro800 controller (either a Micro820 or a Micro850 controller) for your existing MicroLogix 1100 controller wiring configuration.

Any MicroLogix 1100 DC input can be configured as sinking or sourcing depending on how the DC COM terminal is wired.

IMPORTANT For applications that require High Speed Counter (HSC) function, migrate to a Micro850 controller. See Convert to a Micro850 Controller (if there are expansion modules) on page 28 for more information.

Convert to a Micro820 Controller (if there is no expansion module)

You can consider migrating to a Micro820 controller if there is no expansion module in your MicroLogix configuration. Check the following table to see which Micro820 controller is suitable to replace your MicroLogix 1100 controller. Click the catalog number link to see the applicable wiring configuration.

MicroLogix 1100 Controller	Micro820 Controller	Plug-in Modules / Accessories
1763-L16AWA	2080-LC20-20AWB 2080-LC20-20AWBR	• Use 2080-PSAC-12W • Input 00…03 can be used as analog or digital input simultaneously • Use 2080-IF2 if you require more than eight digital inputs

MicroLogix 1100 Controller	Micro820 Controller	Plug-in Modules / Accessories
1763-L16BWA	2080-LC20-20QWB 2080-LC20-20QWBR	• Use 2080-PSAC-12W • Input 00…03 can be used as analog or digital input simultaneously • Use 2080-IF2 if you require more than eight digital inputs • Outputs are not individually isolated, use 2080-OW4I if you require isolation • Embedded high-speed input is not available, use 2080-MOT-HSC if you require high-speed input • If you require more than two plug-ins, upgrade to a Micro850 controller
1763-L16BBB	2080-LC20-20QBB 2080-LC20-20QBBR	• Use2080-IF2 if you require more than eight digital inputs • Outputs are not individually isolated, use 2080-OW4I if you require isolation • Embedded high-speed input is not available, use 2080-MOT-HSC if you require high-speed input • If you require more than two plug-ins, upgrade to a Micro850 controller
1763-L16DWD	Does not support 12V DC	—

Convert to a Micro850 Controller (if there are expansion modules)

You have to migrate to a Micro850 controller if there are expansion modules in your MicroLogix configuration. Check the following table to see which Micro850 controller is suitable to replace your MicroLogix 1100 controller. Click the catalog number link to see the applicable wiring configuration.

MicroLogix 1100 Controller	Micro850 Controller	Plug-in Modules / Accessories
1763-L16AWA	2080-LC50-24AWB	• Use 2080-PS120-240VAC • No embedded analog input, use 2080-IF2 or 2085-IF4 if you require analog input • No RTC, use 2080-MEMBAK-RTC if you require RTC • No datalog, use 2080-SDMEM-RTC-SC if you require data logging (third-party plug-in module from Encompass Partner) • Requires 2085-ECR end cap at the end of expansion modules
1763-L16BWA	2080-LC50-24QWB	• Use 2080-PS120-240VAC • No embedded analog input, use 2080-IF2 or 2085-IF4 if you require analog input • Embedded outputs are not individually isolated, use 2080-OW4I if you require isolation • No RTC, use 2080-MEMBAK-RTC if you require RTC • No datalog, use 2080-SDMEM-RTC-SC if you require data logging (third-party plug-in module from Encompass Partner) • Requires 2085-ECR end cap at the end of expansion modules

MicroLogix 1100 Controller	Micro850 Controller	Plug-in Modules / Accessories
1763-L16BBB	2080-LC50-24QBB	• No embedded analog input, use 2080-IF2 or 2085-IF4 if you require analog input • No embedded relay outputs, use 2080-OW4I if you require relay output • No RTC, use 2080-MEMBAK-RTC if you require RTC • No datalog, use 2080-SDMEM-RTC-SC if you require data logging (third-party plug-in module from Encompass Partner) • Requires 2085-ECR end cap at the end of expansion modules
1763-L16DWD	Does not support 12V DC	—

Migrate From a MicroLogix 1200 Controller

This section helps you determine how to select and wire a suitable Micro800 controller (either a Micro850 or a Micro870 controller) for your existing MicroLogix 1200 controller wiring configuration.

Any MicroLogix 1200 DC input can be configured as sinking or sourcing depending on how the DC COM terminal is wired.

Convert to a Micro850 Controller (up to four expansion modules)

You can consider migrating to a Micro850 controller if there are up to four expansion modules in your MicroLogix configuration. Check the following table to see which Micro850 controller is suitable to replace your MicroLogix 1200 controller. Click the catalog number link to see the applicable wiring configuration.

MicroLogix 1200 Controller	Micro850 Controller	Plug-in Modules / Accessories
1762-L24AWA 1762-L24AWAR	2080-LC50-24AWB	• Use 2080-PS120-240VAC for AC option • No RTC, use 2080-MEMBAK-RTC if you require RTC • Requires 2085-ECR end cap at the end of expansion modules
1762-L40AWA 1762-L40AWAR	2080-LC50-48AWB	
1762-L24BWA 1762-L24BWAR	2080-LC50-24QWB	
1762-L40BWA 1762-L40BWAR	2080-LC50-48QWB	
1762-L24BXB 1762-L24BXBR	2080-LC50-24QBB	• There is no combination output, use 2080-OW4I or 2085-OW8 if your require relay output • No RTC, use 2080-MEMBAK-RTC if you require RTC • Requires 2085-ECR end cap at the end of expansion modules
1762-L40BXB 1762-L40BXBR	2080-LC50-48QBB	

Convert to a Micro870 Controller (more than four expansion modules)

You have to migrate to a Micro870 controller if there are more than four expansion modules in your MicroLogix configuration. Check the following table to see which Micro870 controller is suitable to replace your MicroLogix 1200 controller. Click the catalog number link to see the applicable wiring configuration.

MicroLogix 1200 Controller	Micro870 Controller	Plug-in Modules / Accessories
1762-L24AWA 1762-L24AWAR	2080-LC70-24AWB	• Requires 2085-EP24VDC power supply module when expanding beyond four expansion modules • Use 2080-PS120-240VAC for AC option • No RTC, use 2080-MEMBAK-RTC2 if you require RTC • Requires 2085-ECR end cap at the end of expansion modules • If migrating from 1762-L40AWA or 1762-L40AWAR, add 2085-IA8 and 2085-OW8 to fill I/O gap
1762-L40AWA 1762-L40AWAR		
1762-L24BWA 1762-L24BWAR	2080-LC70-24QWB 2080-LC70-24QWBK	• Requires 2085-EP24VDC power supply module when expanding beyond four expansion modules • Use 2080-PS120-240VAC for AC option • No RTC, use 2080-MEMBAK-RTC2 if you require RTC • Requires 2085-ECR end cap at the end of expansion modules • If migrating from 1762-L40BWA or 1762-L40BWAR, add 2085-IQ16 and 2085-OW8 to fill I/O gap
1762-L40BWA 1762-L40BWAR		
1762-L24BXB 1762-L24BXBR	2080-LC70-24QBB 2080-LC70-24QBBK	• Requires 2085-EP24VDC power supply module when expanding beyond four expansion modules • Use 2080-PS120-240VAC for AC option • There is no combination output, use 2080-OW4I or 2085-OW8 if you require relay output • No RTC, use 2080-MEMBAK-RTC2 if you require RTC • Requires 2085-ECR end cap at the end of expansion modules • If migrating from 1762-L40BXB or 1762-L40BXBR, add 2085-IQ16 and 2085-OW8 to fill I/O gap
1762-L40BXB 1762-L40BXBR		

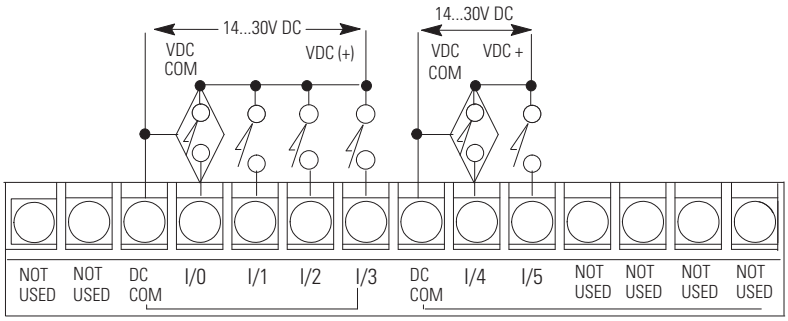
Wiring Configuration

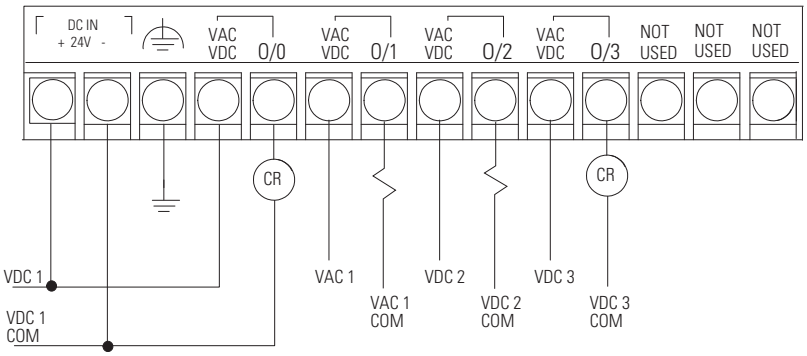
This section contains the following wiring diagrams:

- MicroLogix 1000 Controller Wiring
- MicroLogix 1100 Controller Wiring
- MicroLogix 1200 Controller Wiring
- Micro820 Controller Wiring
- Micro830 Controller Wiring
- Micro850 Controller Wiring
- Micro870 Controller Wiring

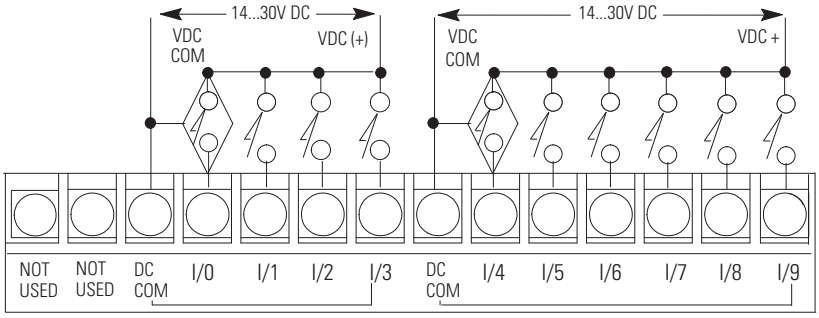
MicroLogix 1000 Controller Wiring

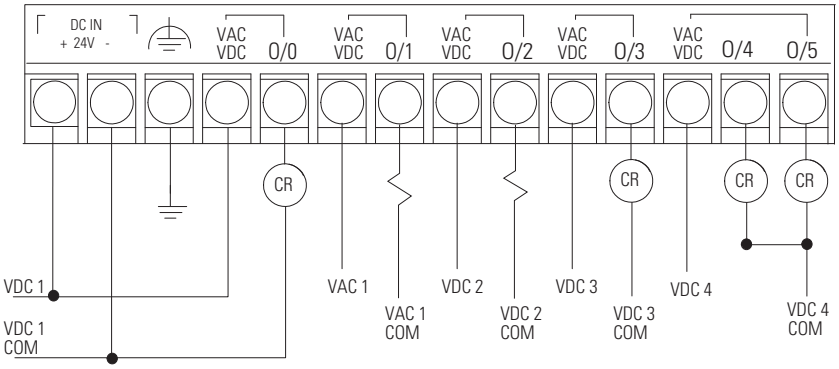
1761-L10BWB





1761-L16BWB





1761-L16NWA and 1761-L16NWB

The 1761-L16NWA and 1761-L16NWB controllers are equipped with input circuits capable of 24V AC or 24V DC operation. Specifications for operation of the input circuits are given in the following table. Except for the input circuits, the 1761-L16NWA and1761-L16NWB controllers are identical in operation to the 1761-L16BWA and 1761-L16BWB, respectively.

For more information, see the MicroLogix 1000 Programmable Controllers Document Update, publication 1761-DU001.

24V AC / 24V DC Input Specifications for 1761-L16NWA and 1761-L16NWB

Specification[1]		AC Excitation[3]	DC Excitation
On State Voltage	Minimum	18V AC	14V DC
	Nominal	24V AC	24V DC
	Maximum	26.4V AC @ 55 °C (131 °F) 30V AC @ 30 °C (86 °F)	26.4V DC @ 55 °C (131 °F) 30V DC @ 30 °C (86 °F)
On State Current	Minimum	3.0 mA @ 18V AC	2.5 mA @ 14V DC
	Nominal	8.0 mA @ 24V AC	8.0 mA @ 24V DC
	Maximum	12 mA @ 30V AC	12 mA @ 30V DC
Off State Voltage	Minimum	0.0V AC	0.0V DC
	Maximum	3.0V AC	5.0V DC
Off State Current	Minimum	1.0 mA	1.5 mA
Frequency	Nominal	50/60 Hz	See Turn On Time/Turn Off Time
	Range	47…63 Hz	
Turn On Time[2]	Minimum	2 ms	2 ms
	Maximum	20 ms	20 ms
Turn Off Time[2]	Minimum	10 ms	10 ms
	Maximum	20 ms	20 ms

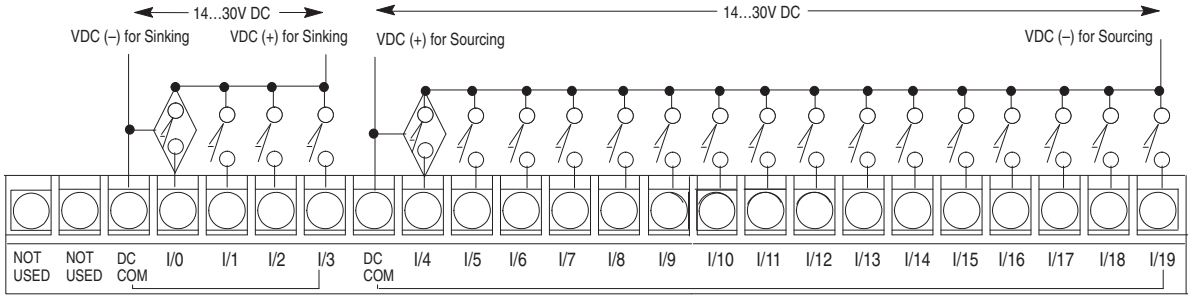
(1) Input circuits may be operated AC or DC on a group basis only.

(2) Turn On and Turn Off Times are *not* adjustable.

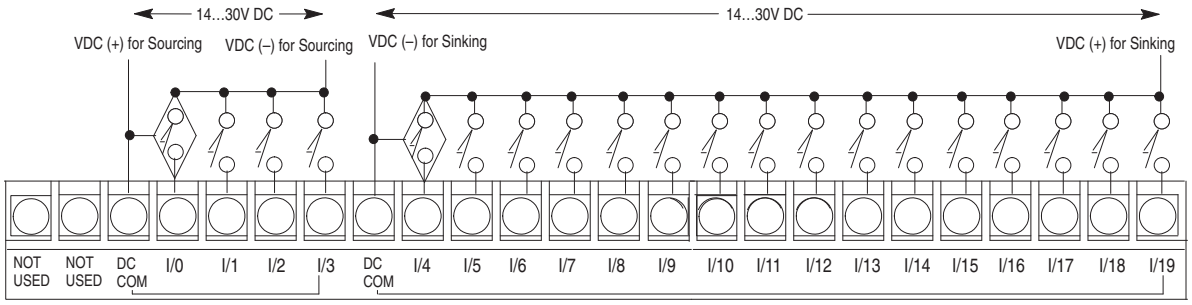
(3) All AC specifications are sinusoidal RMS values.

1761-L32BWB

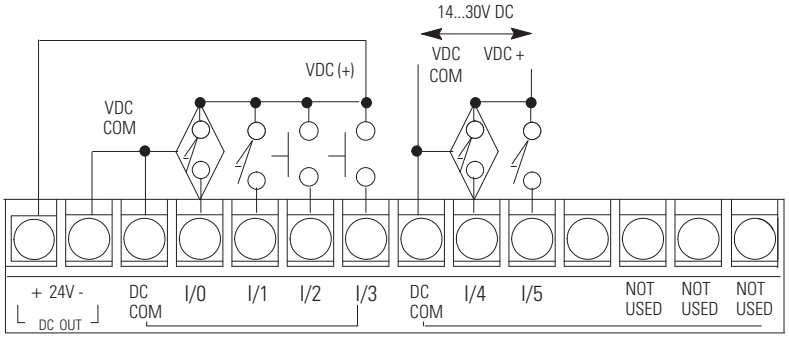
Sinking Inputs **Sourcing Inputs**

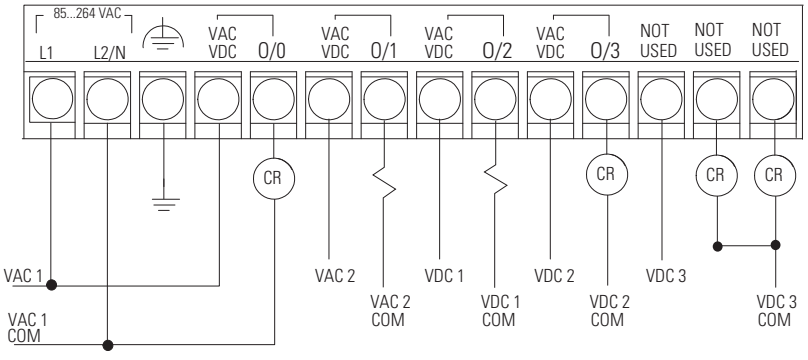


Sourcing Inputs **Sinking Inputs**

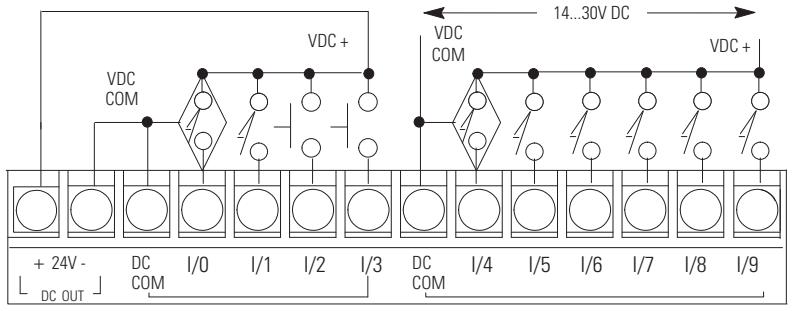


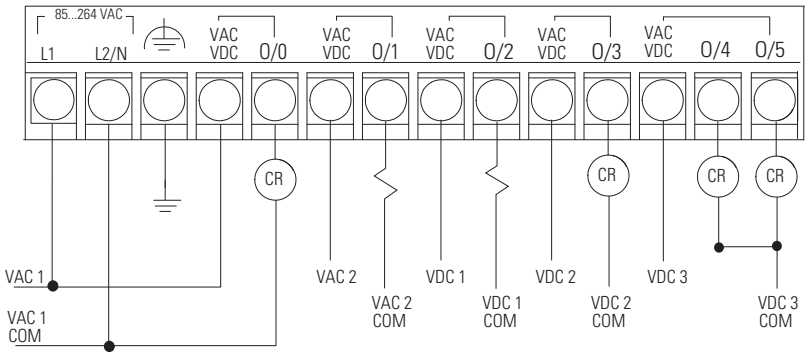
1761-L10BWA



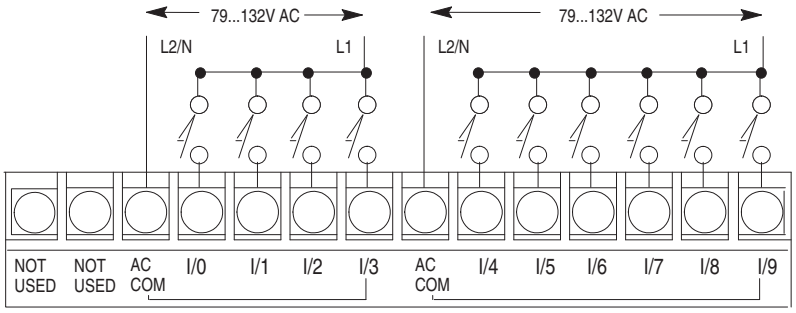


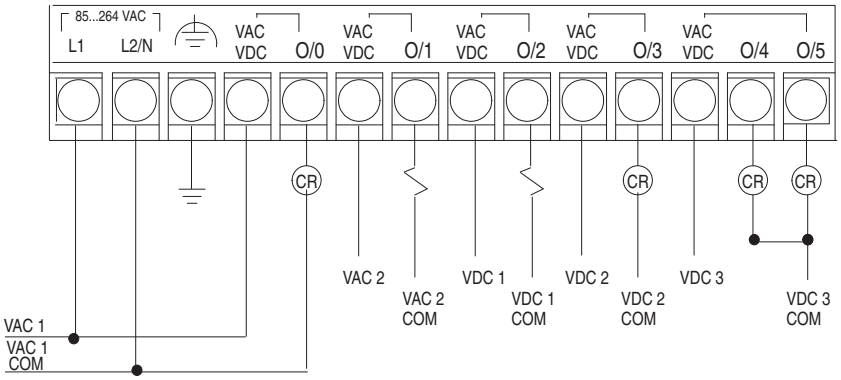
1761-L16BWA



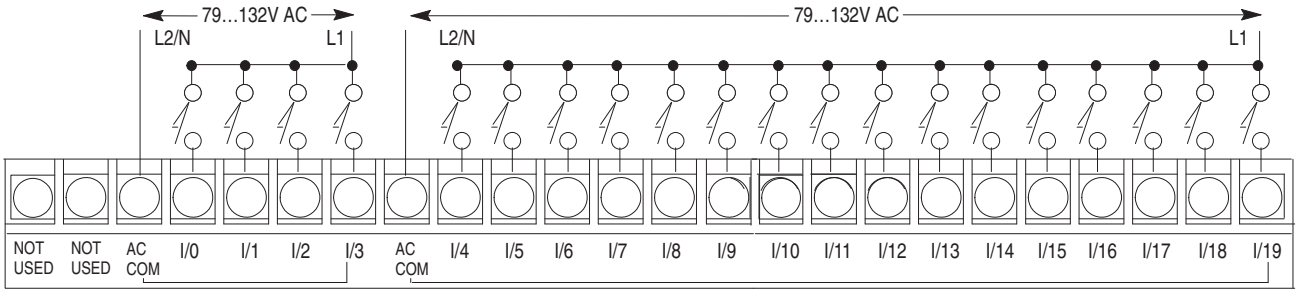


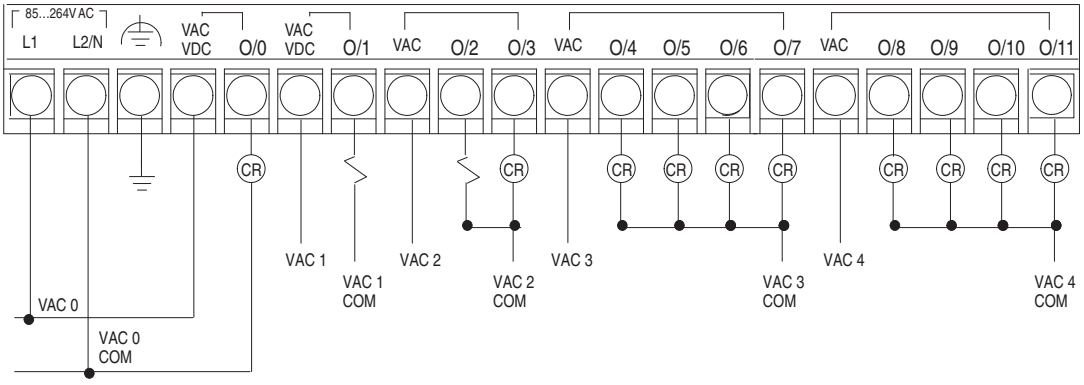
1761-L16AWA



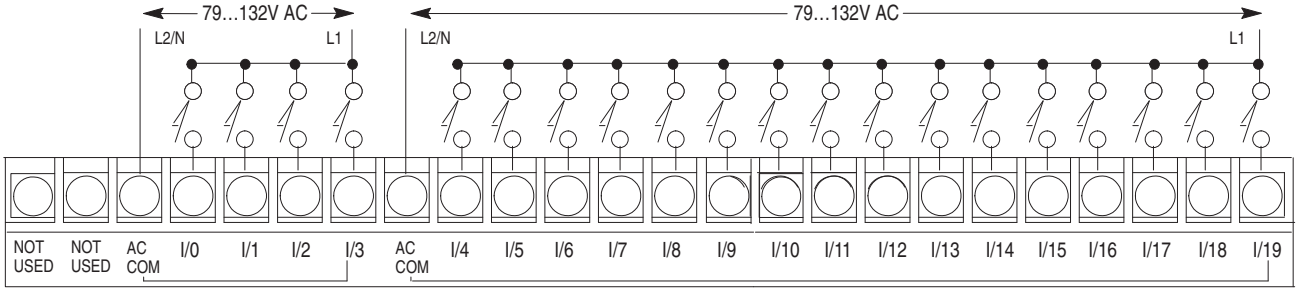


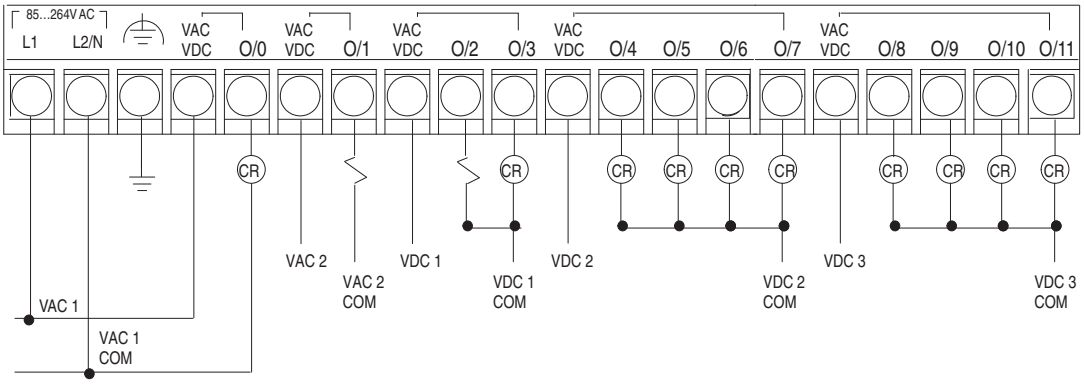
1761-L32AAA



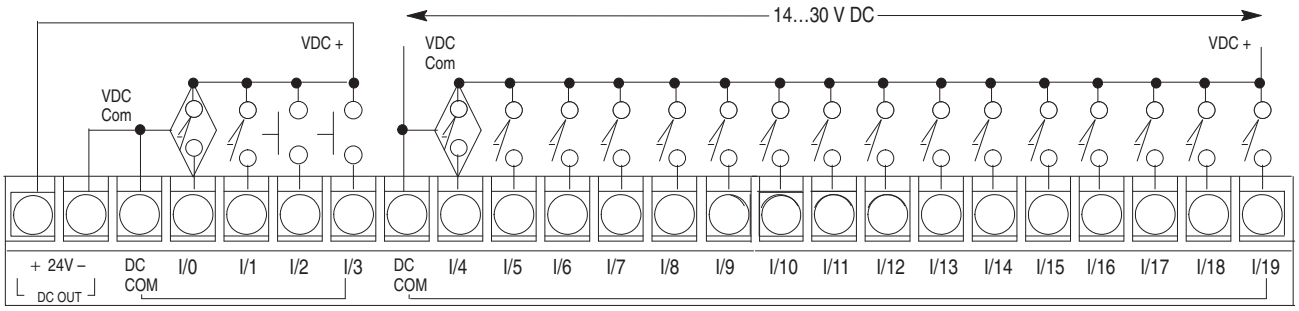


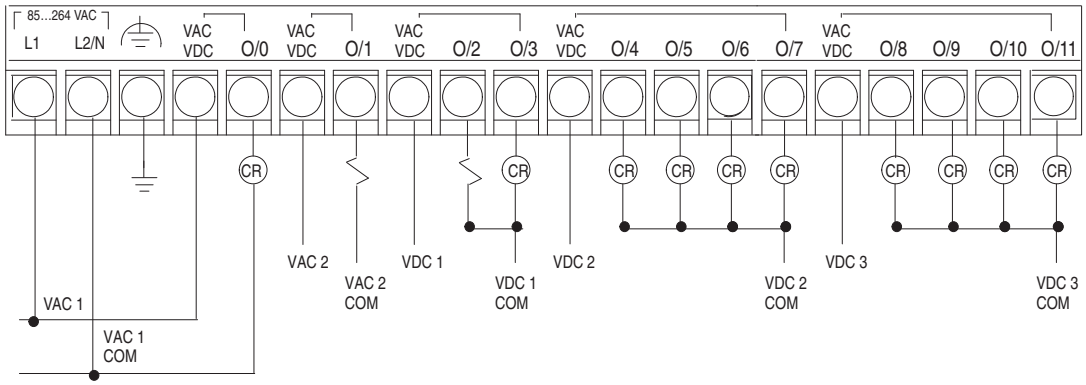
1761-L32AWA



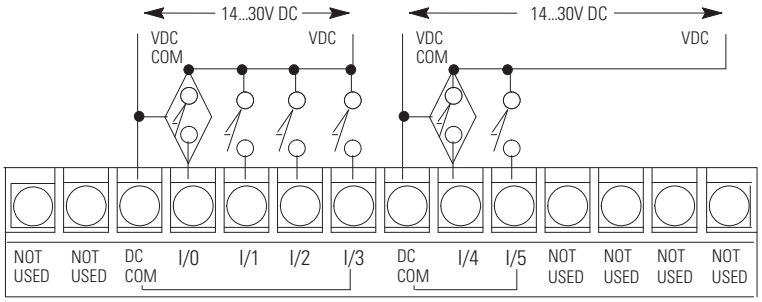


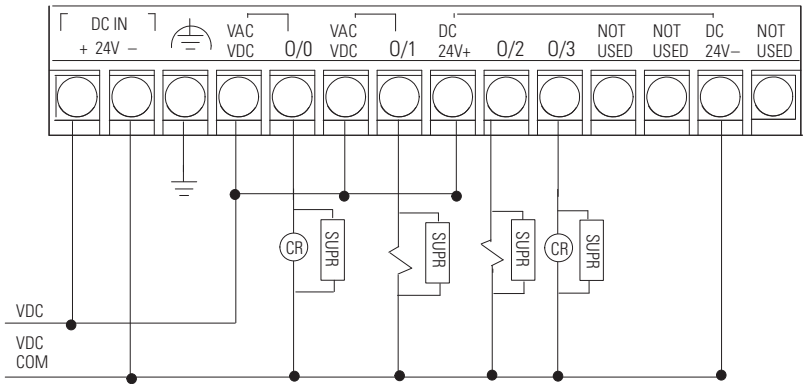
1761-L32BWA



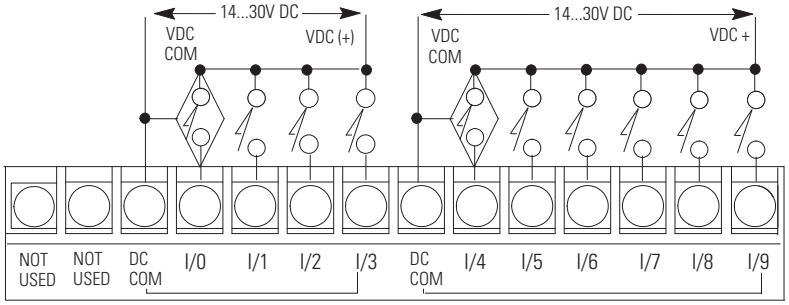


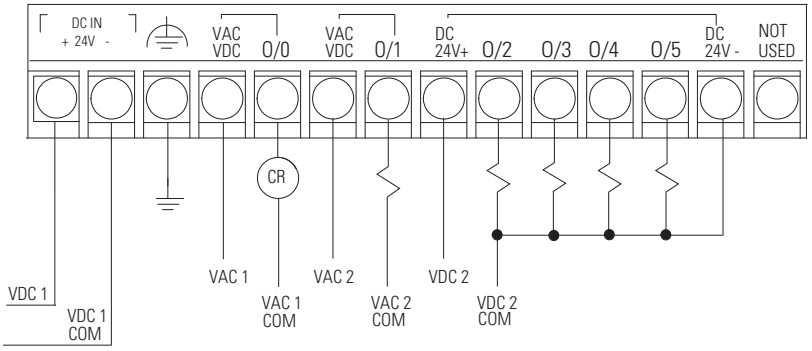
1761-L10BXB



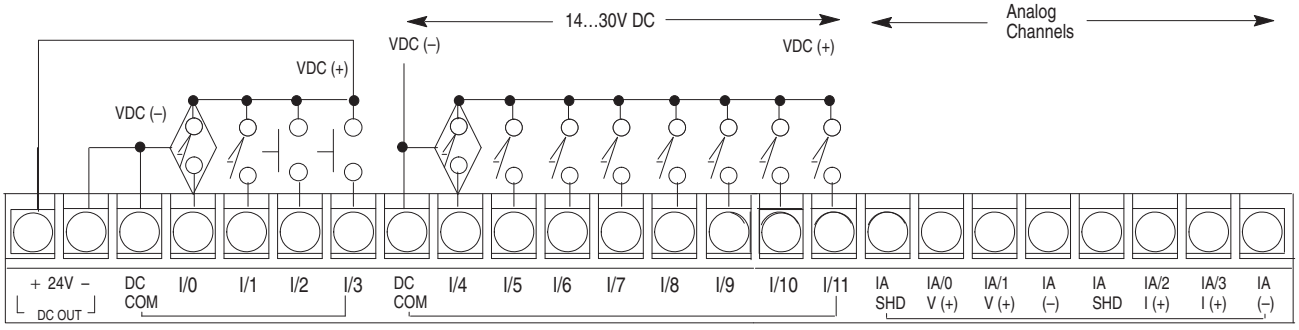


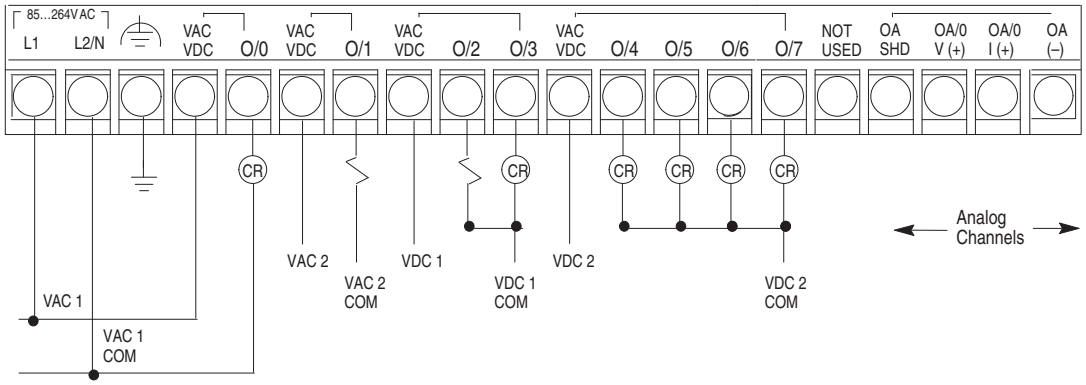
1761-L16BBB



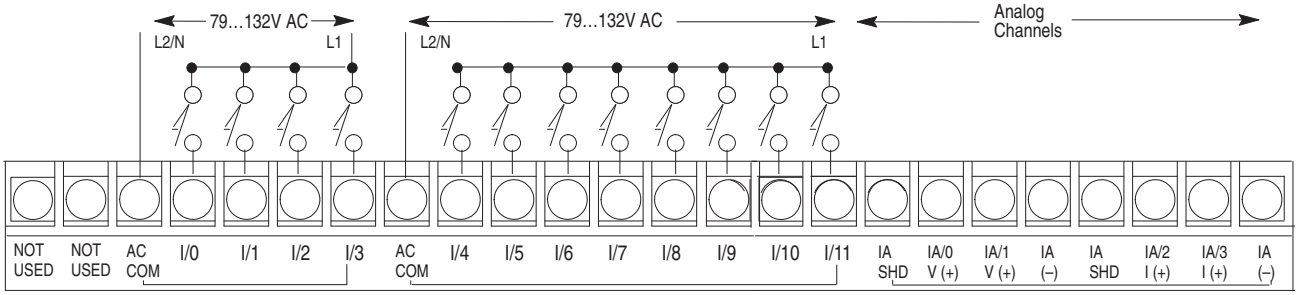


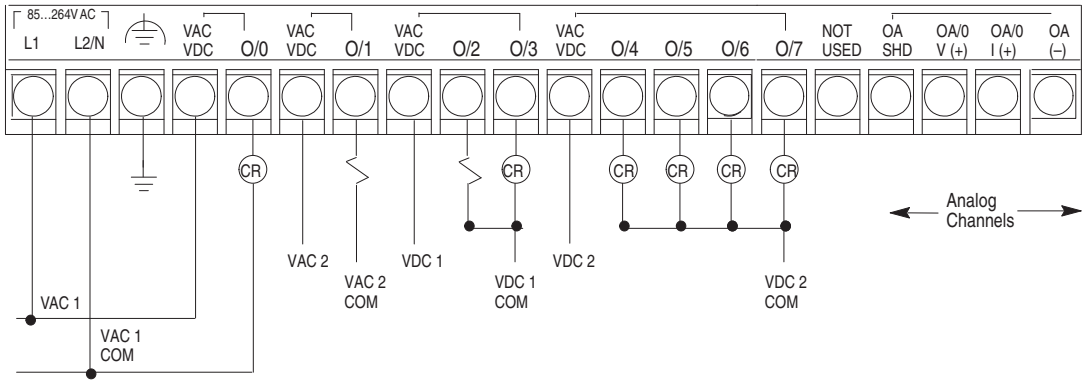
1761-L20BWA-5A



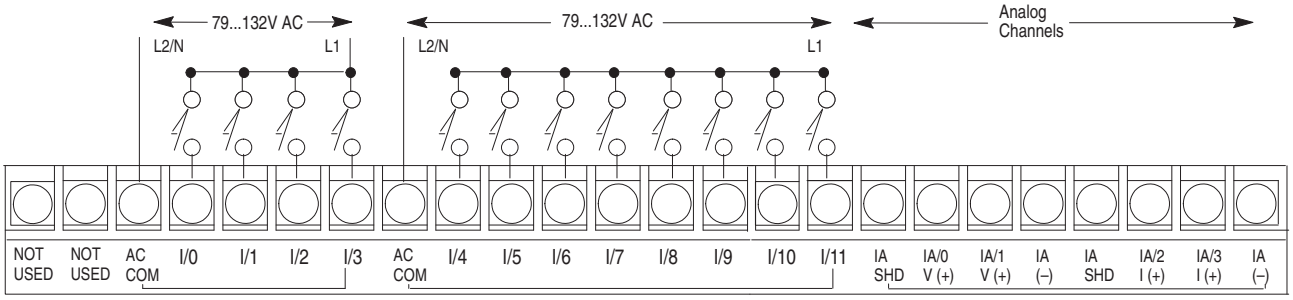


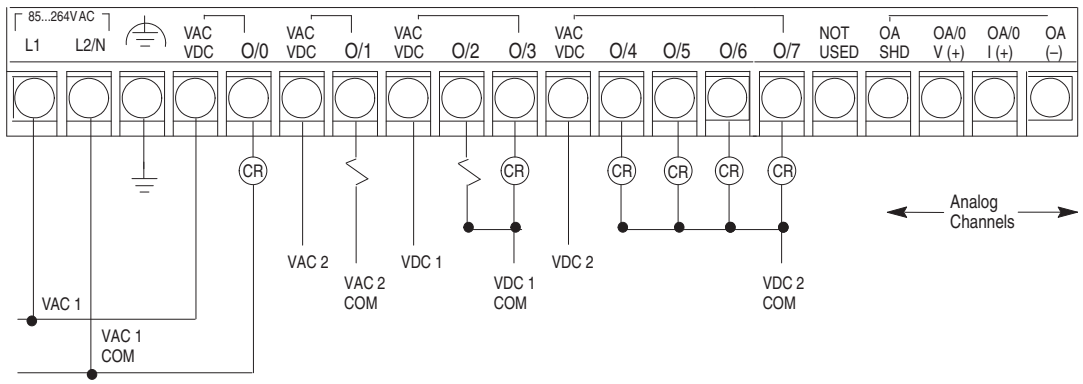
1761-L20AWA-5A





1761-L20BWB-5A

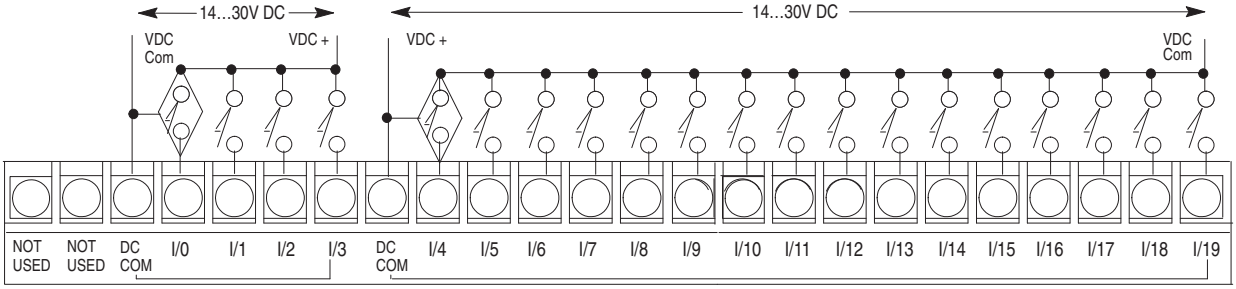




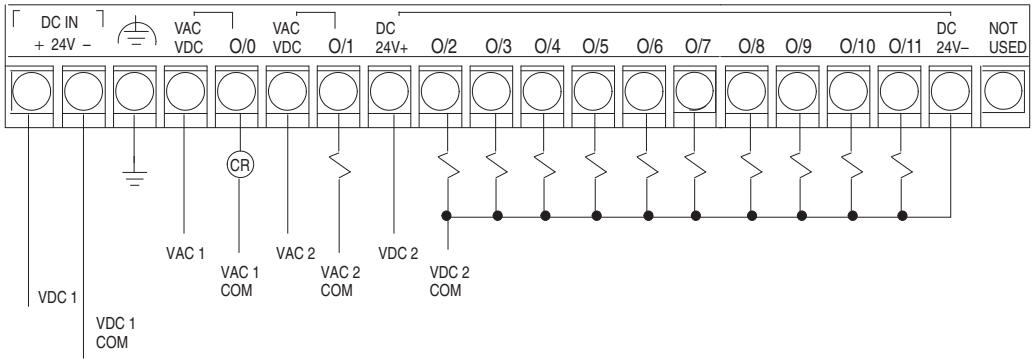
1761-L32BBB

Sinking Configuration

Sourcing Configuration



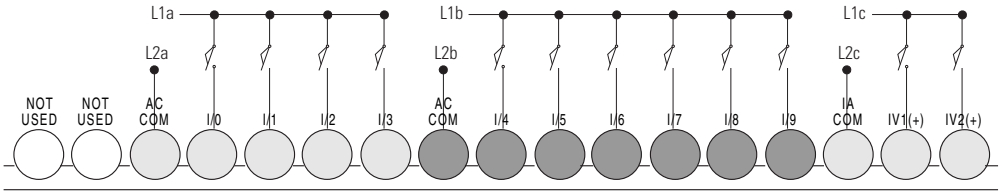
Sourcing Outputs



MicroLogix 1100 Controller Wiring

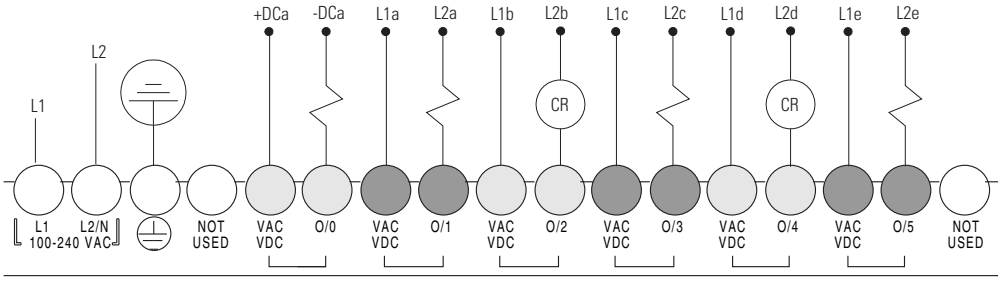
1763-L16AWA

Inputs



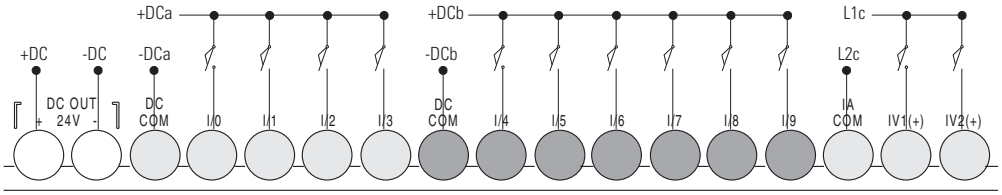
"NOT USED" terminals are not intended for use as connection points.

Outputs

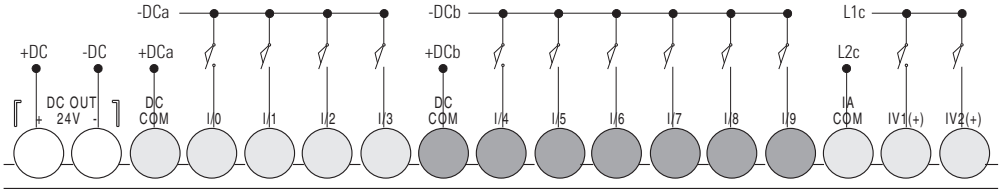


1763-L16BWA

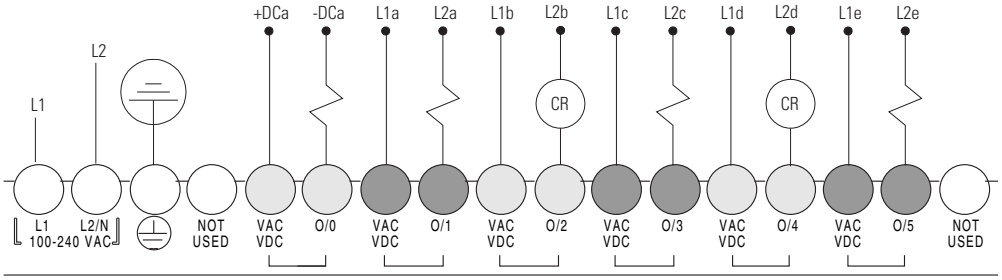
Sinking Inputs



Sourcing Inputs

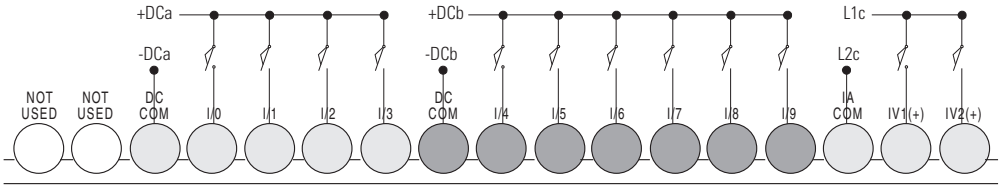


Outputs

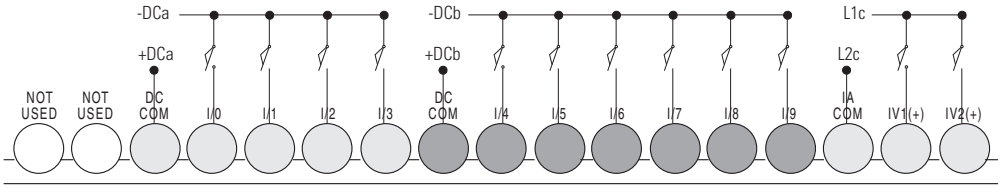


1763-L16BBB

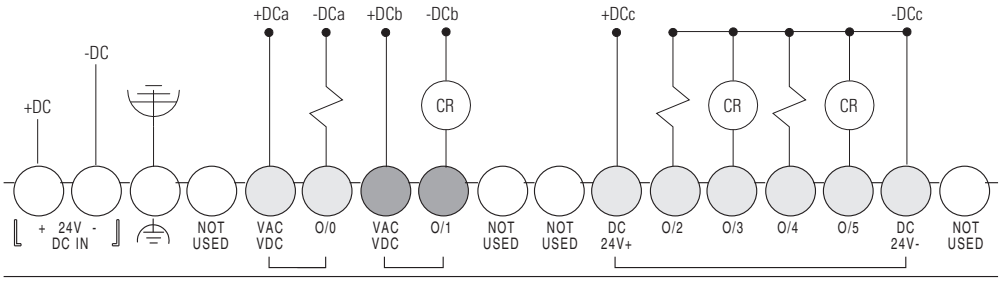
Sinking Inputs



Sourcing Inputs

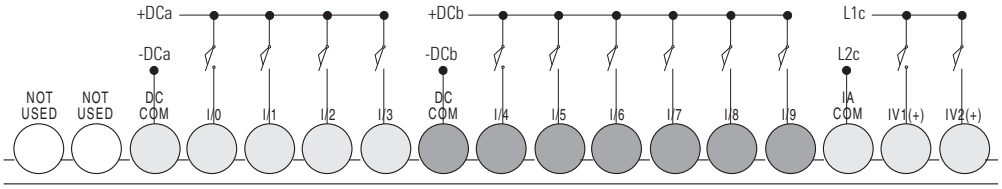


Outputs

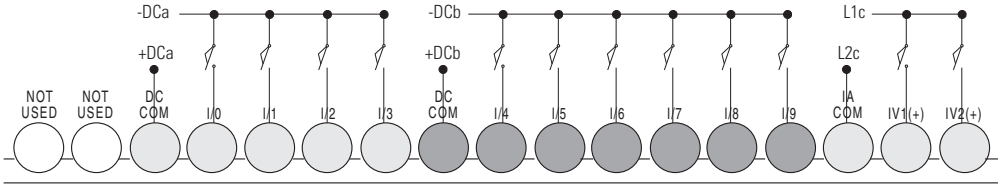


1763-L16DWD

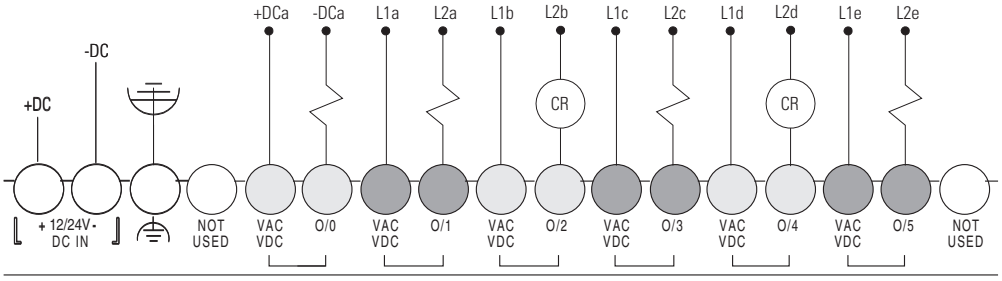
Sinking Inputs



Sourcing Inputs

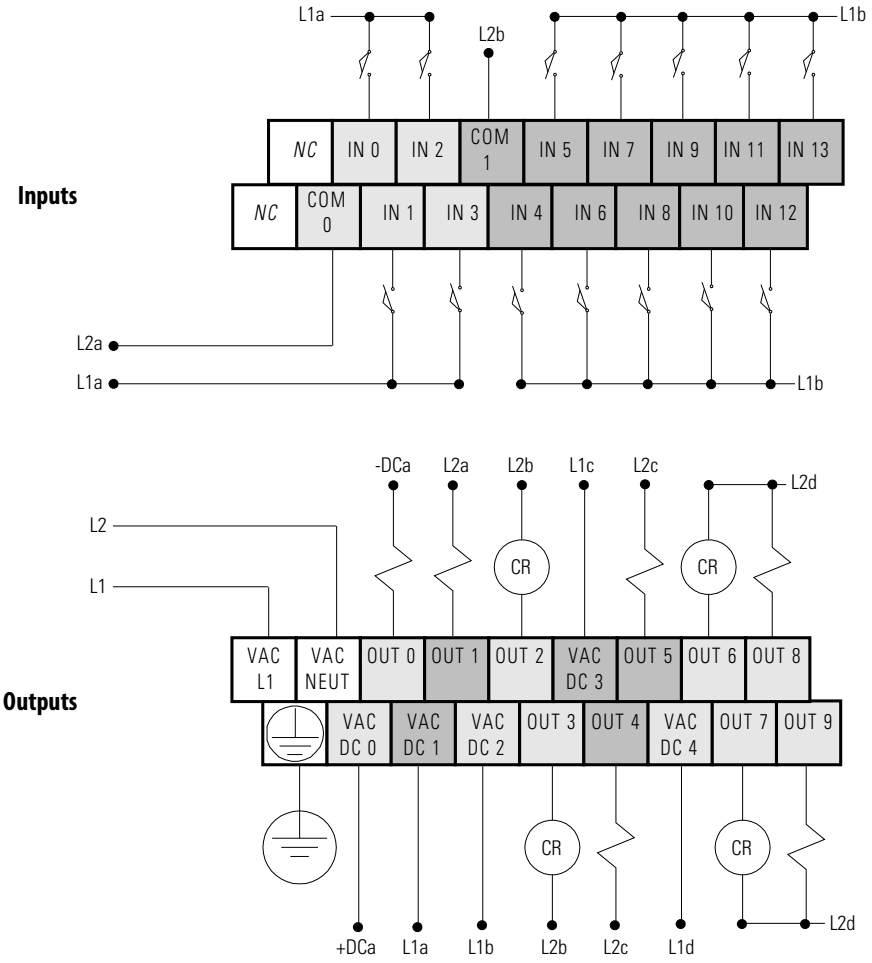


Outputs

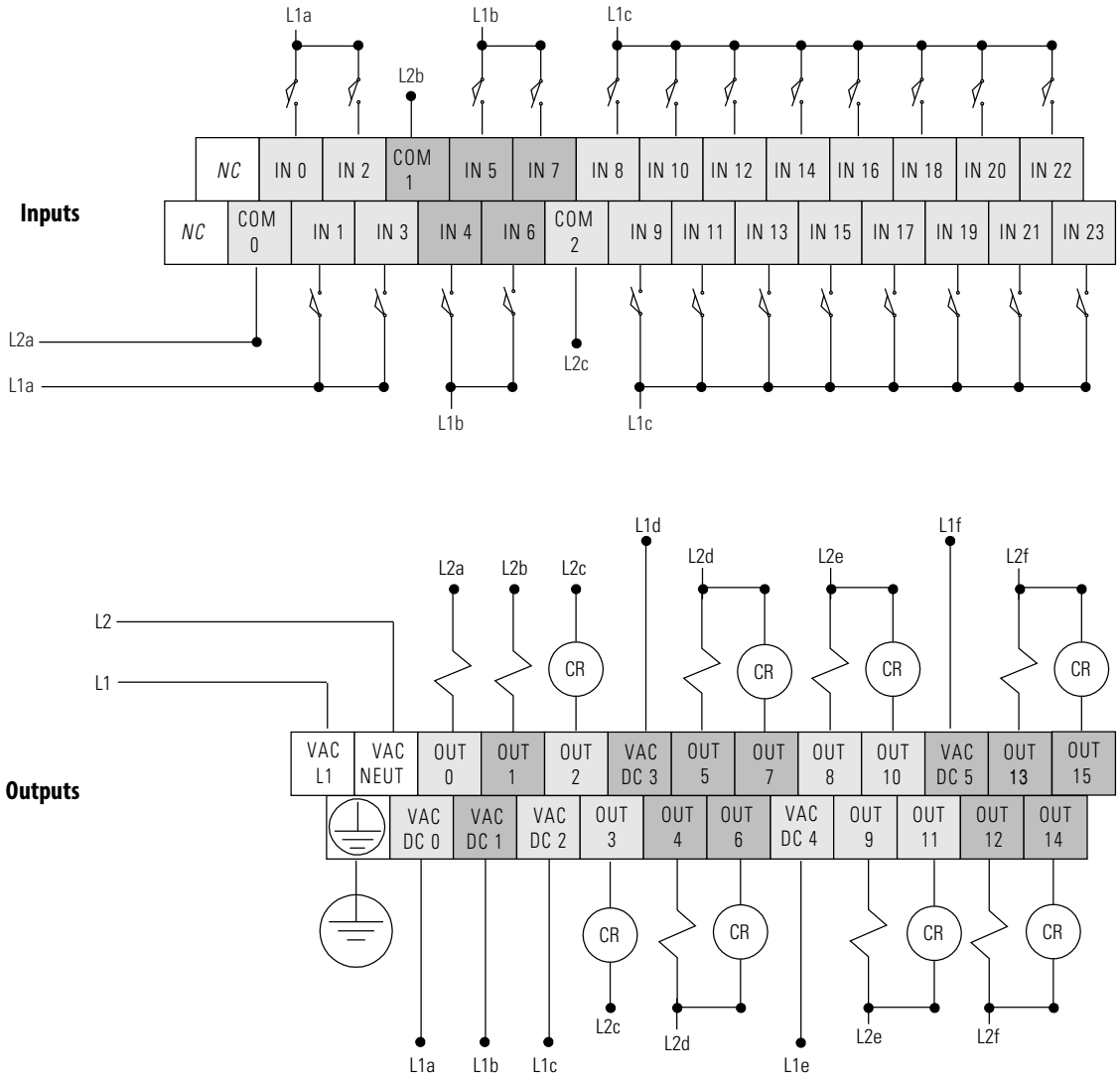


MicroLogix 1200 Controller Wiring

1762-L24AWA, 1762-L24AWAR

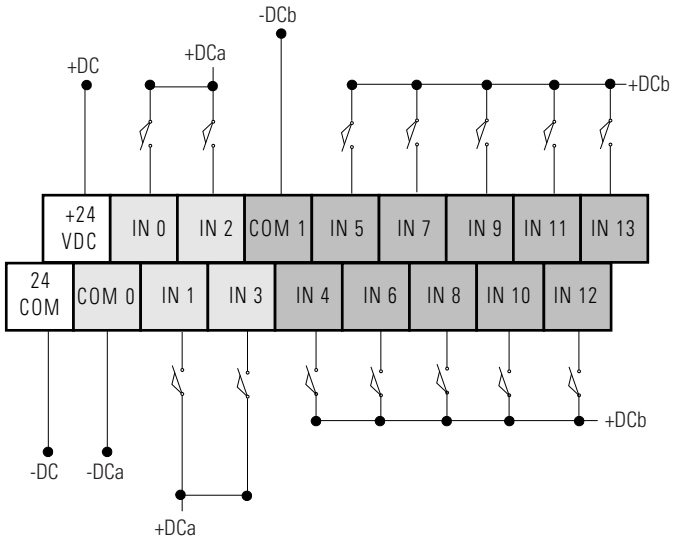


1762-L40AWA, 1762-L40AWAR

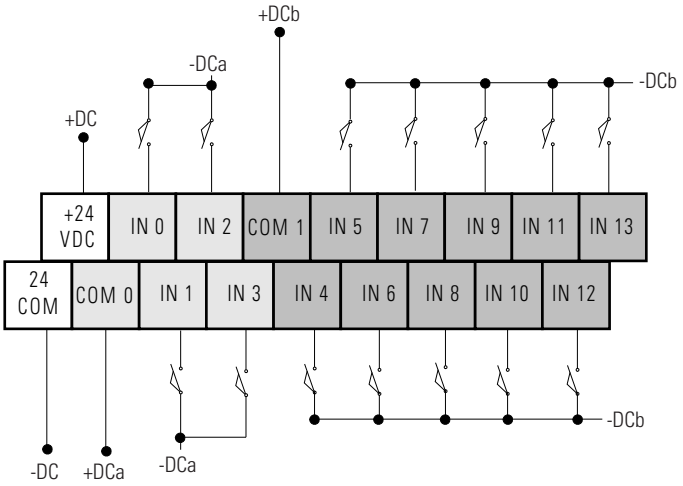


1762-L24BWA, 1762-L24BWAR

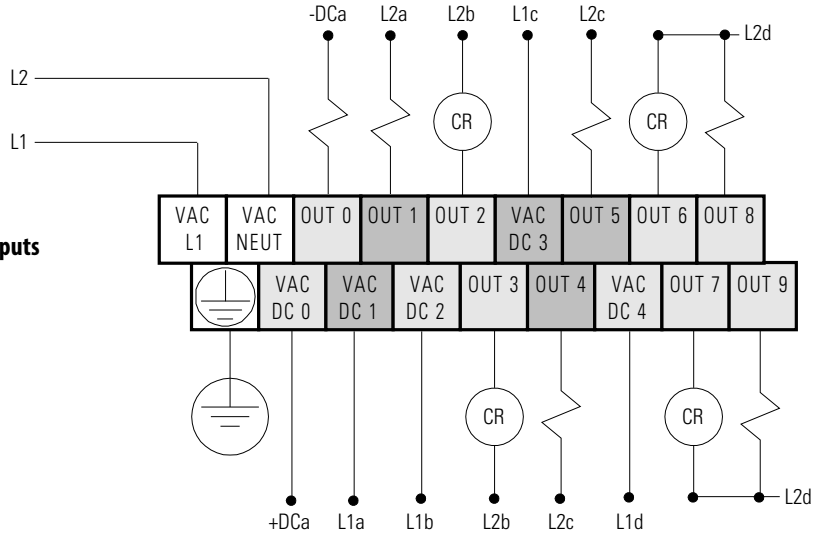
Sinking Inputs



Sourcing Inputs

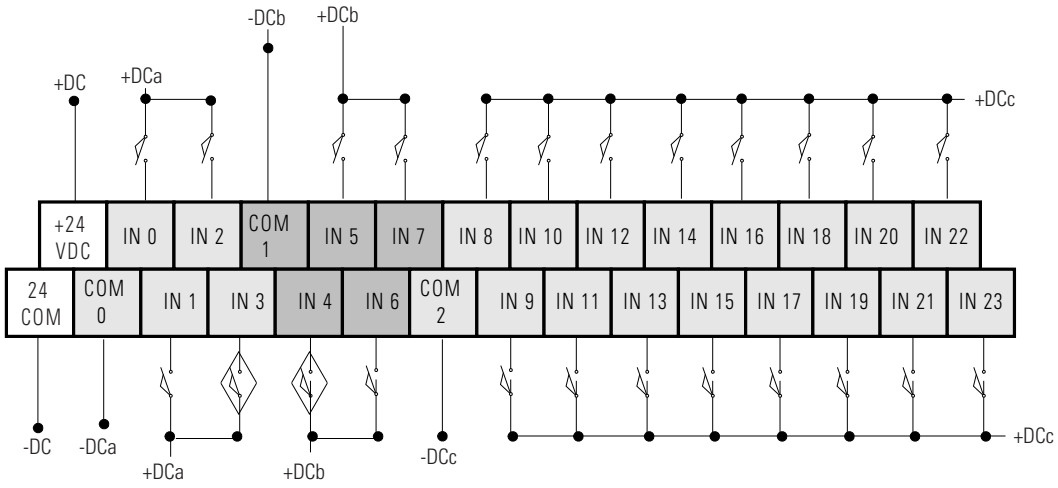


Outputs

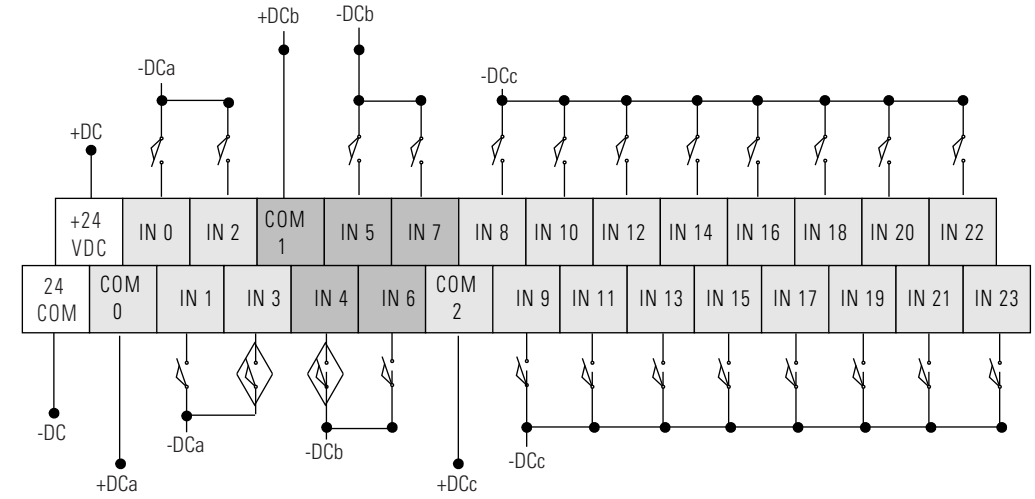


1762-L40BWA, 1762-L40BWAR

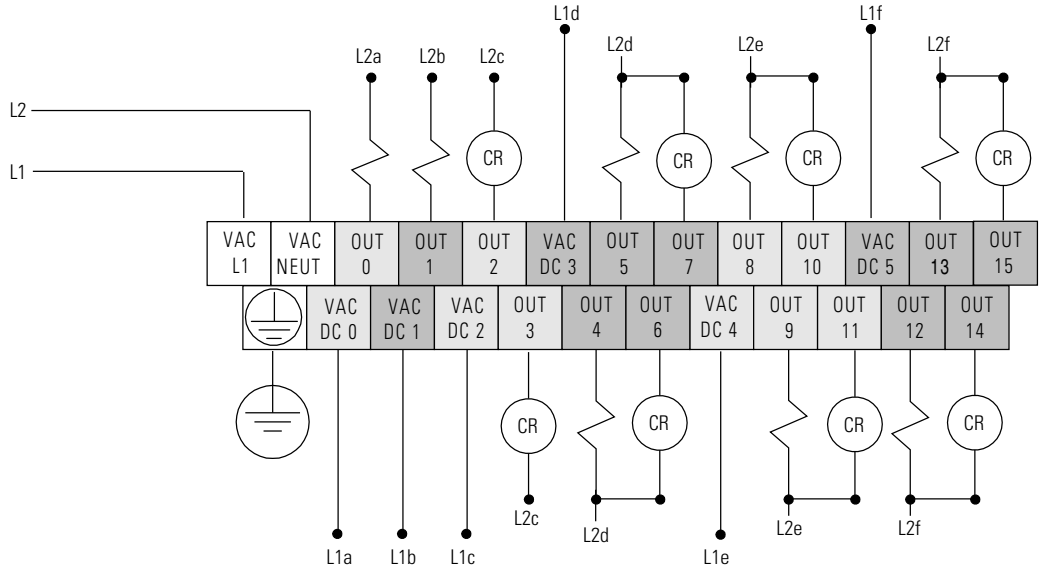
Sinking Inputs



Sourcing Inputs

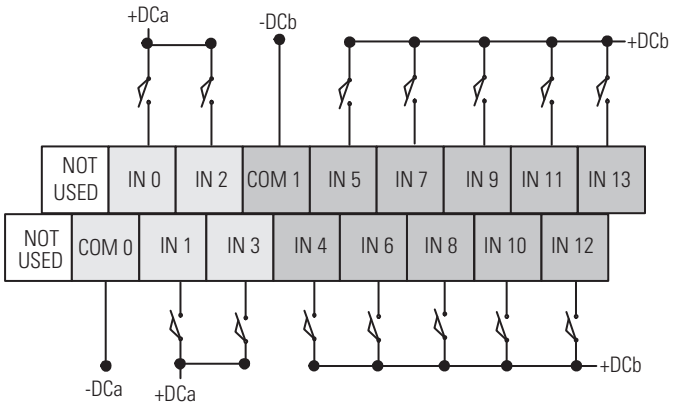


Outputs

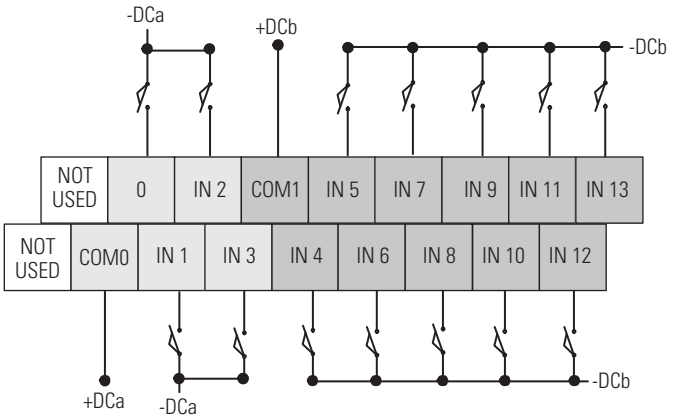


1762-L24BXB, 1762-L24BXBR

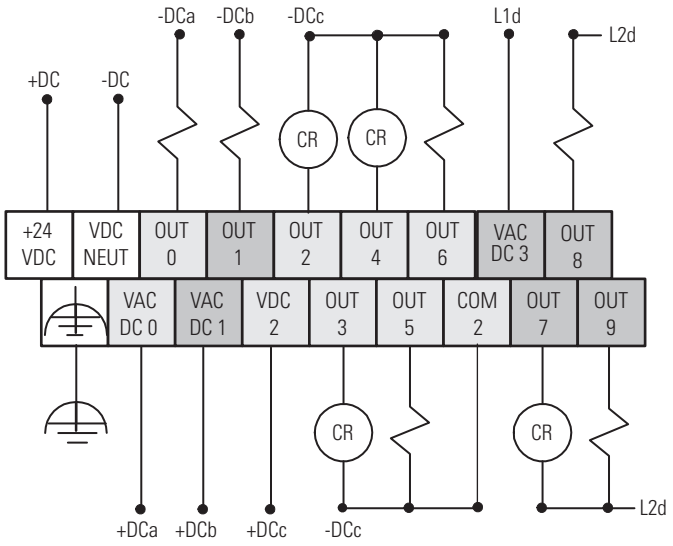
Sinking Inputs



Sourcing Inputs

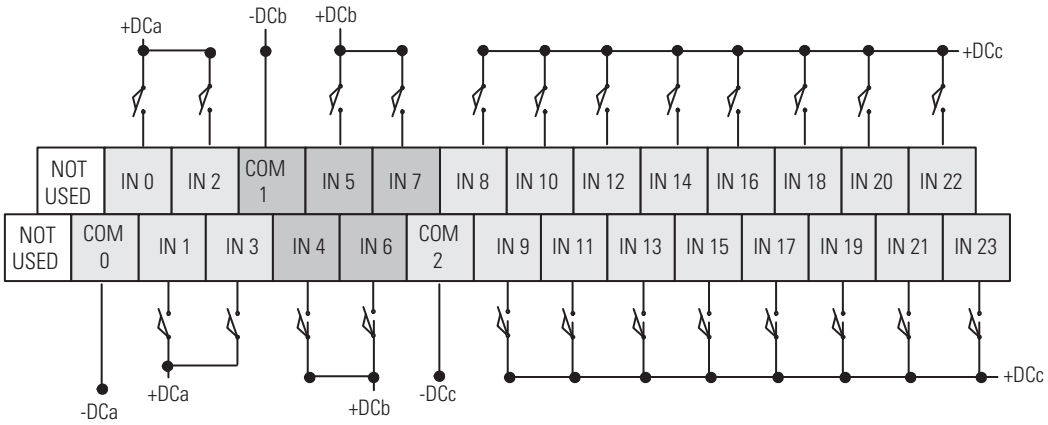


Outputs

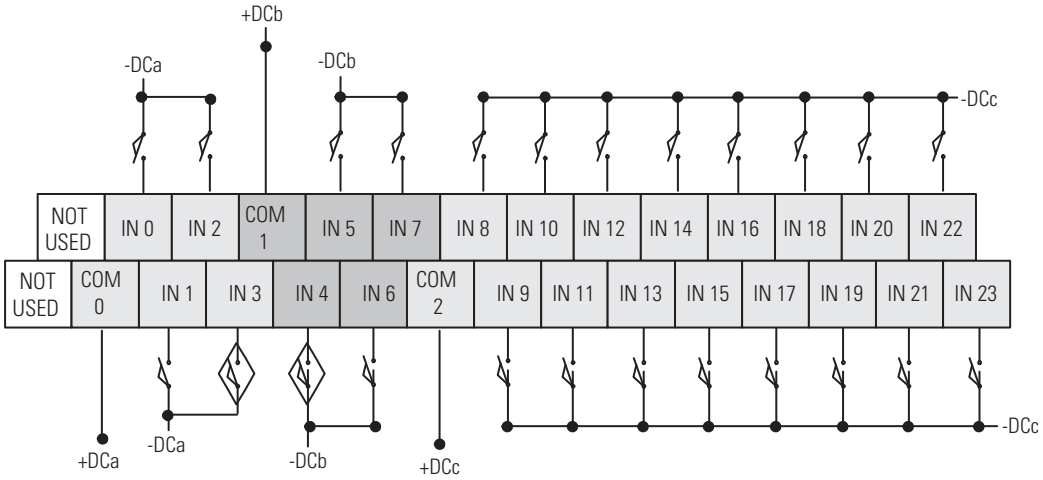


1762-L40BXB, 1762-L40BXBR

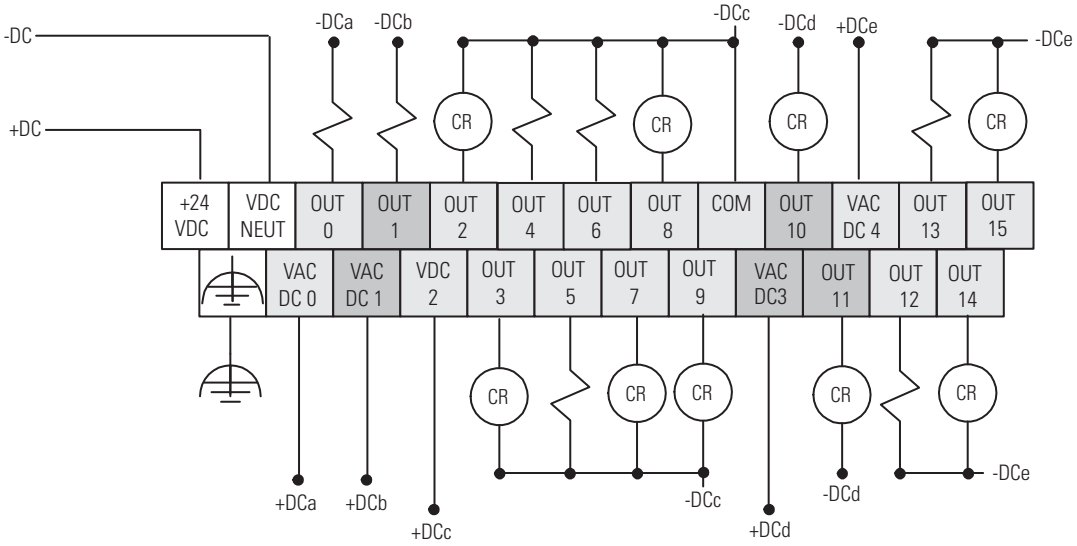
Sinking Inputs



Sourcing Inputs



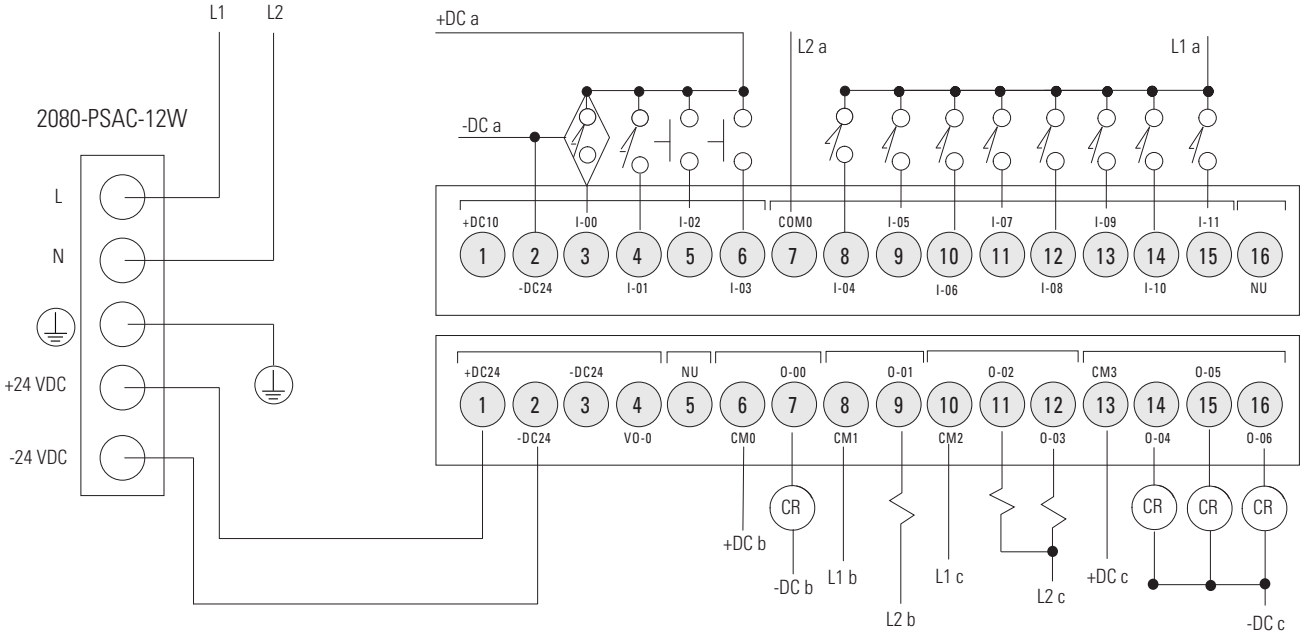
Outputs



Micro820 Controller Wiring

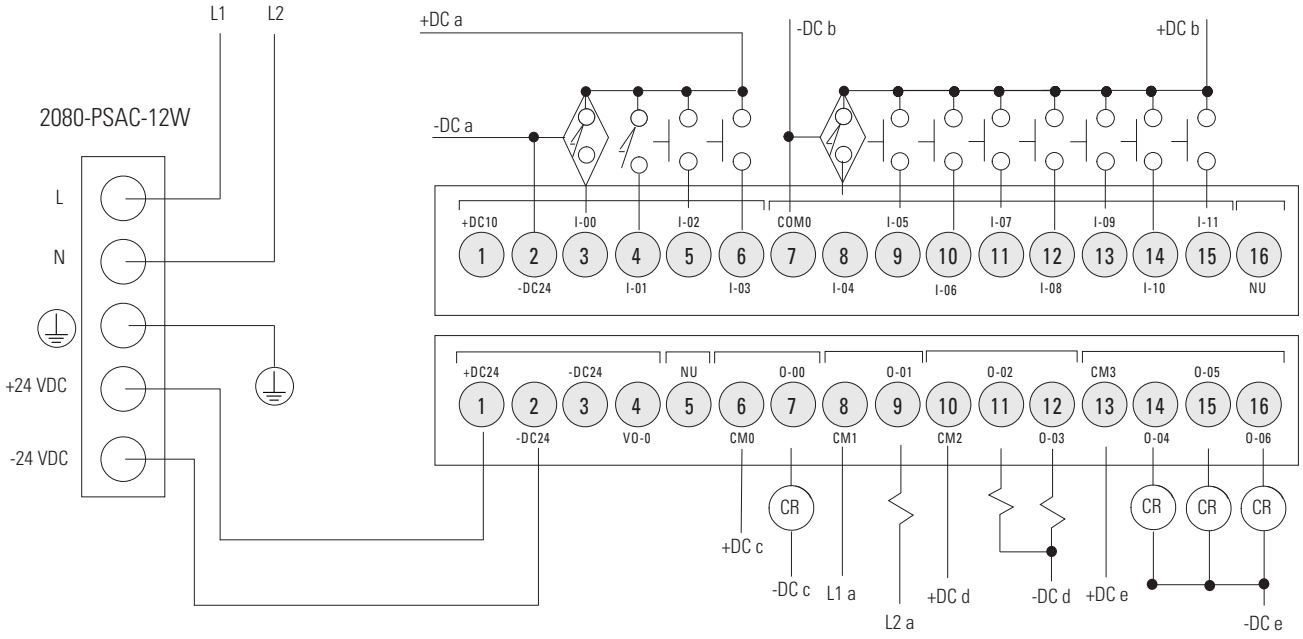
2080-LC20-20AWB, 2080-LC20-20AWBR

Sinking Inputs

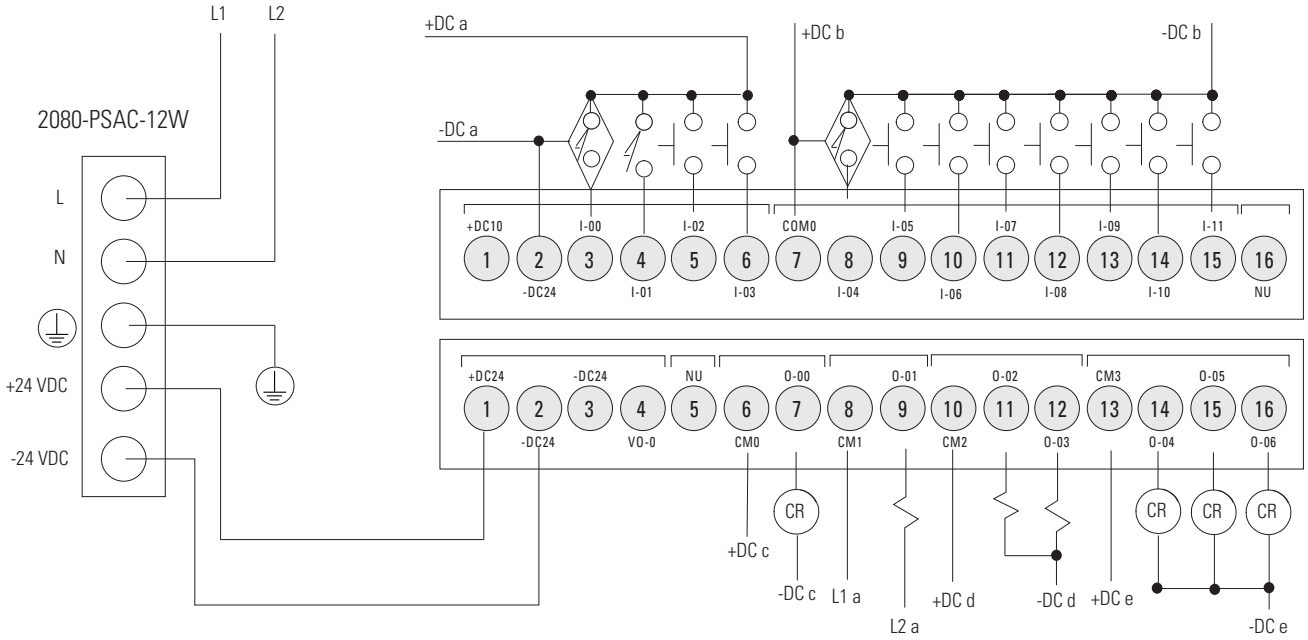


2080-LC20-20QWB, 2080-LC20-20QWBR

DC Sinking Input Configuration Inputs – Inputs 00…11

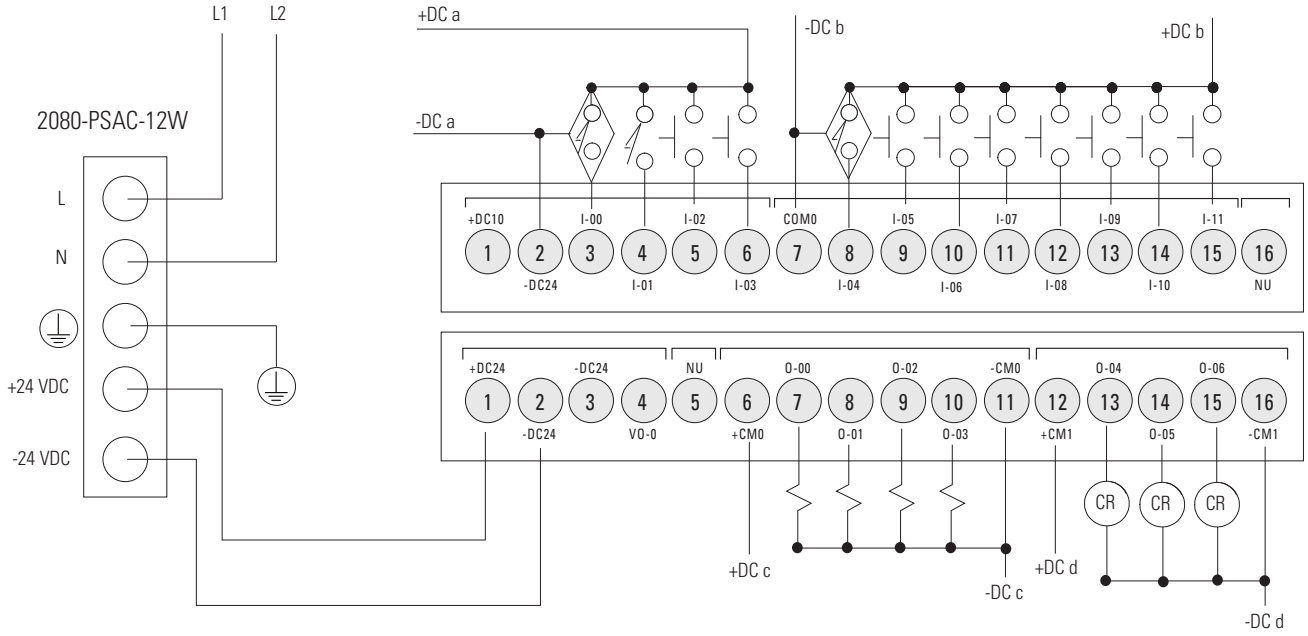


DC Sourcing Input Configuration Inputs – Inputs 4…11 only

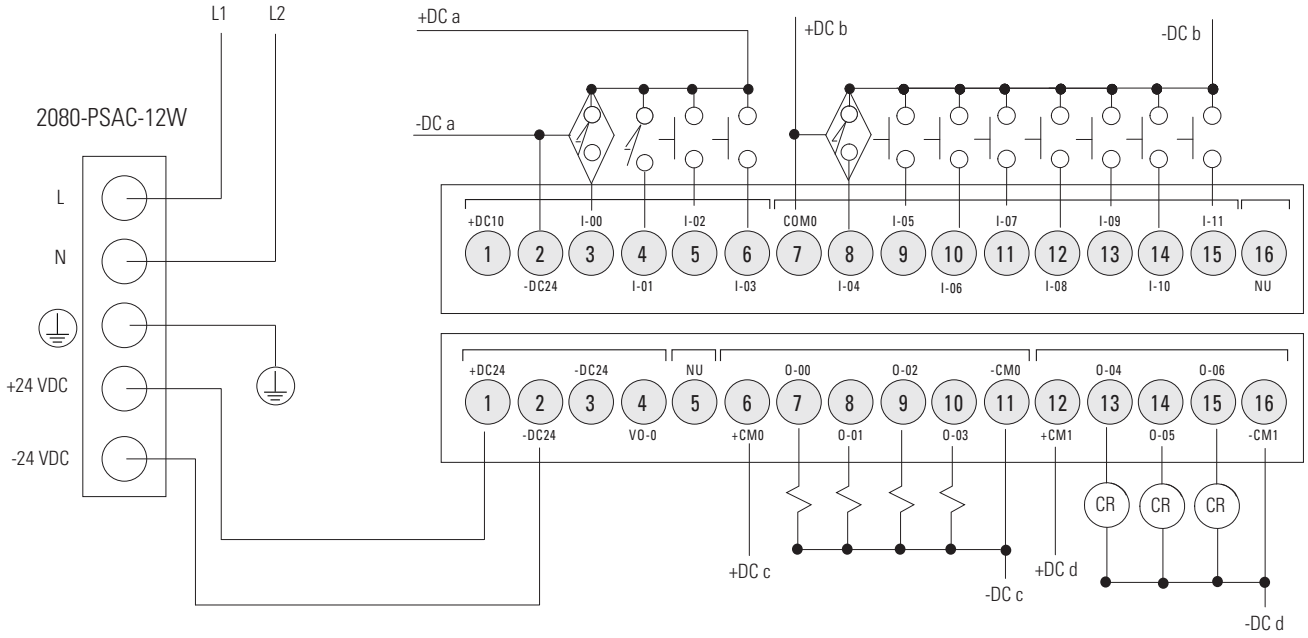


2080-LC20-20QBB, 2080-LC20-20QBBR

DC Sinking Input Configuration Inputs – Inputs 00…11



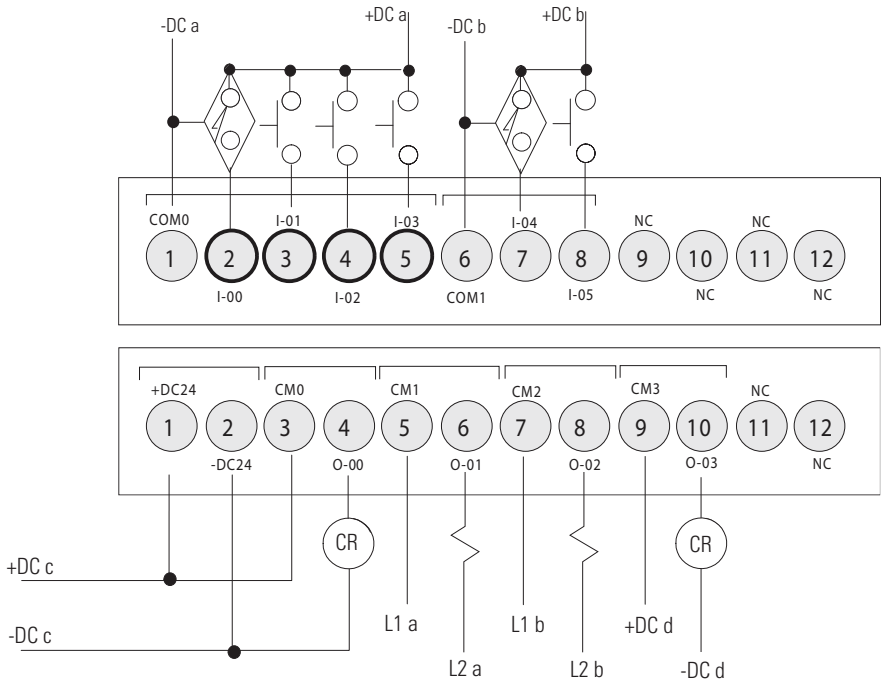
DC Sourcing Input Configuration Inputs – Inputs 4…11 only



Micro830 Controller Wiring

2080-LC30-10QWB

DC Sinking Input Configuration Inputs – Inputs 00…05

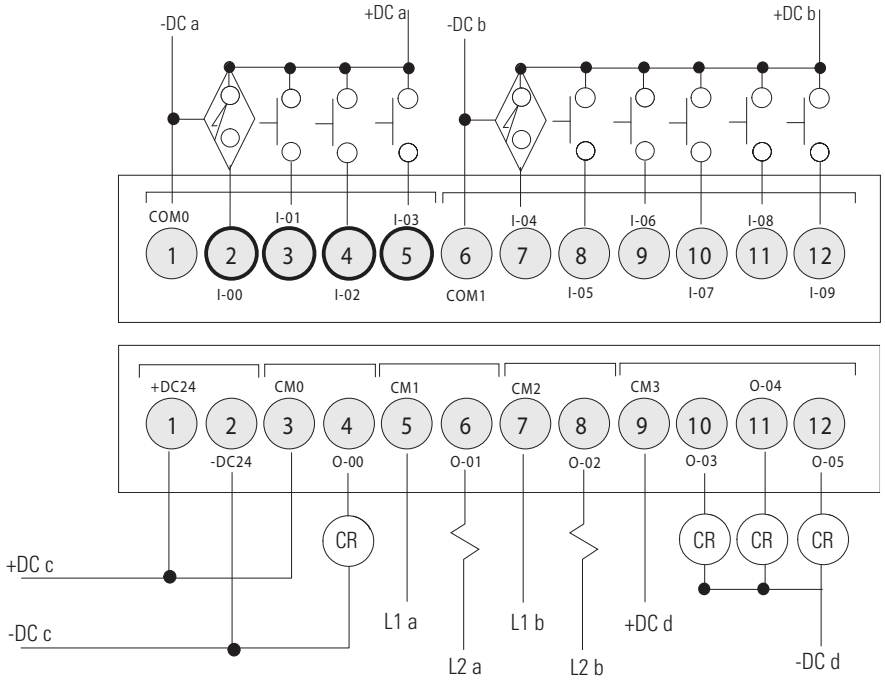


○ indicates high-speed inputs and outputs

IMPORTANT Do not connect –DC24 m(Output terminal 2) to Earth/Chassis Ground.

2080-LC30-16QWB

DC Sinking Input Configuration Inputs – Inputs 00…09

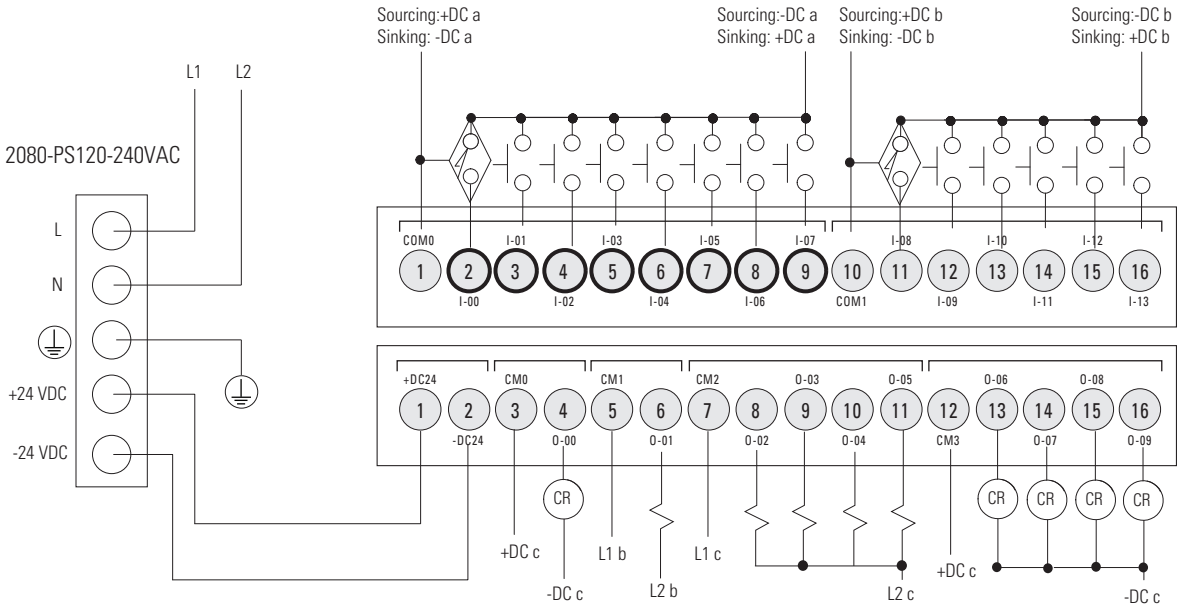


○ indicates high-speed inputs and outputs

IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

2080-LC30-24QWB

DC Sinking Inputs – Inputs 00…13

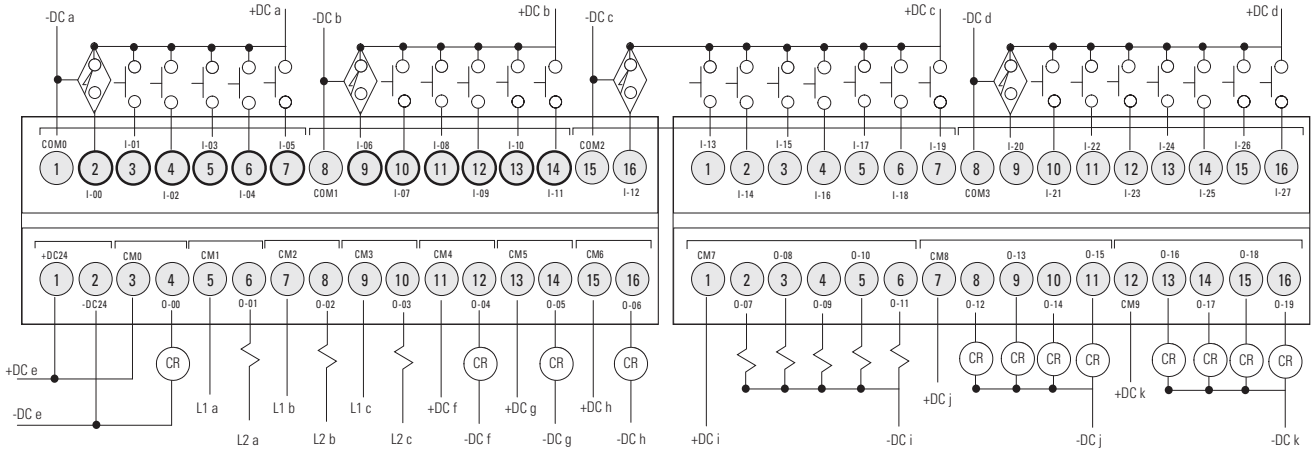


○ indicates high-speed inputs and outputs

IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

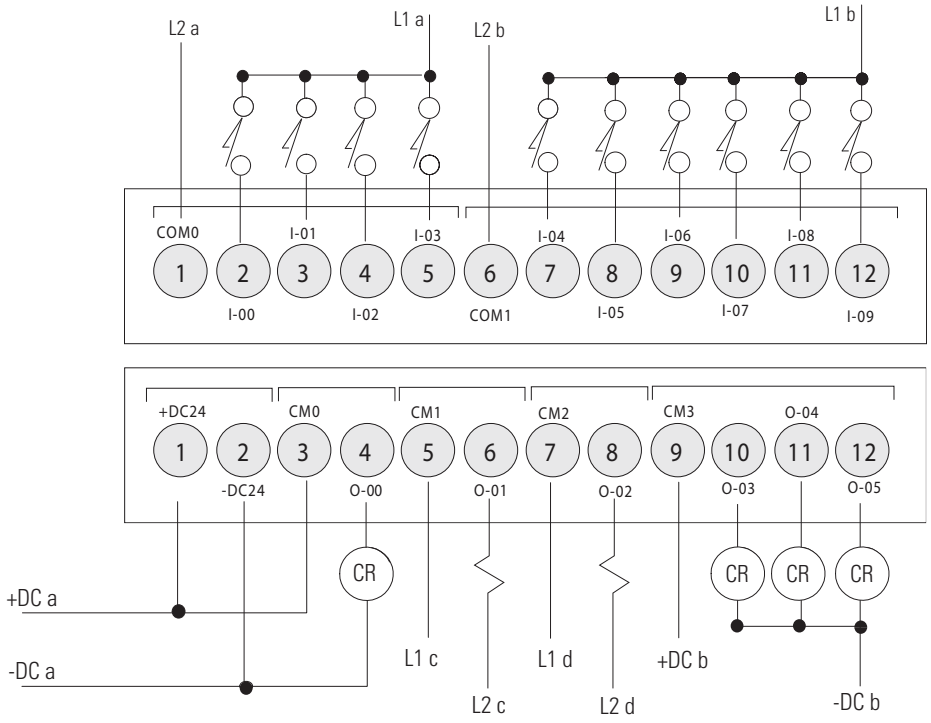
2080-LC30-48QWB

DC Sinking Inputs – Inputs 00…27



IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

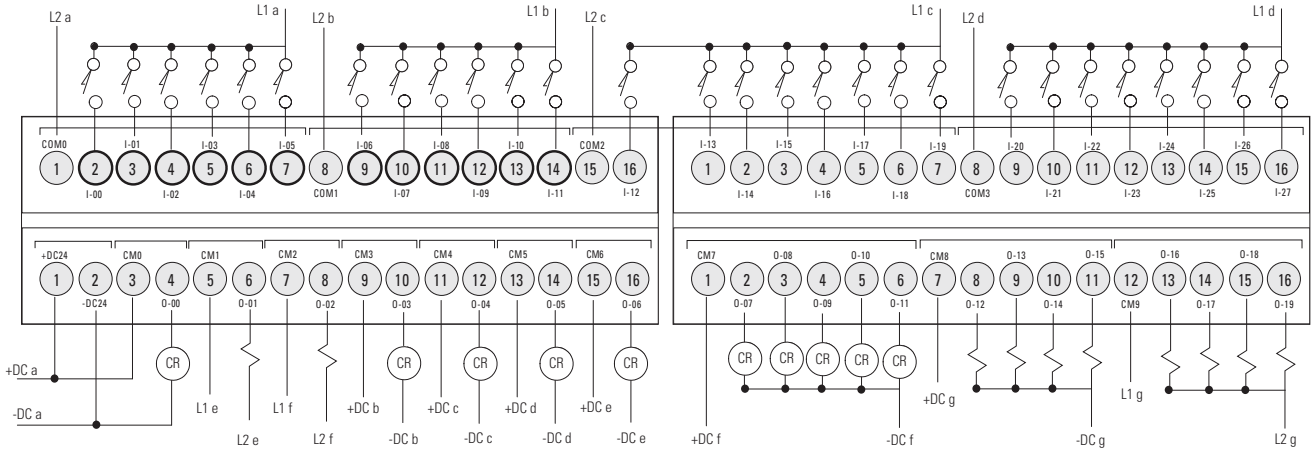
2080-LC30-16AWB



2080-LC30-16AWB has no high-speed inputs.

IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

2080-LC30-48AWB

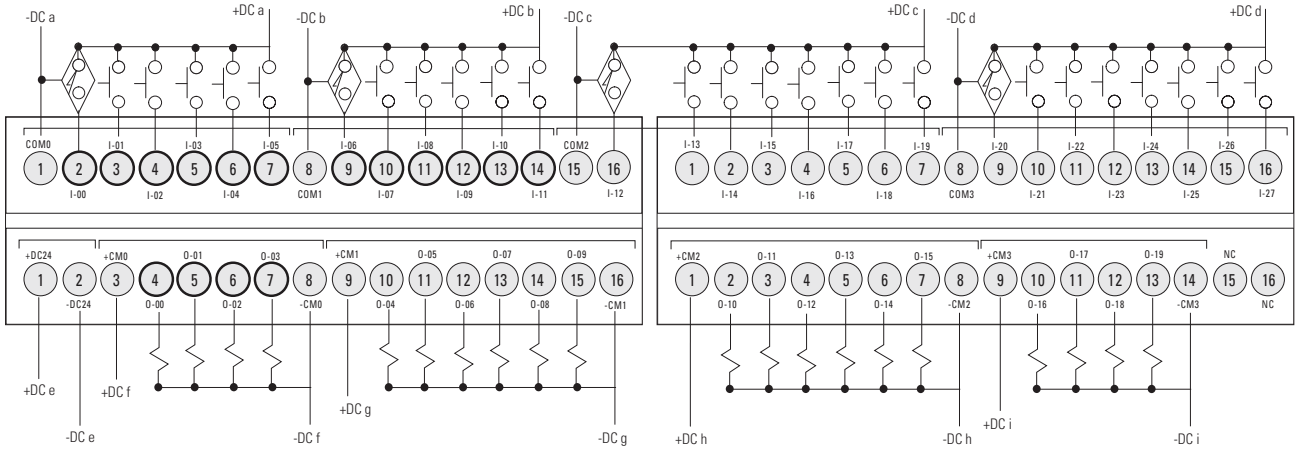


2080-LC30-48AWB has no high-speed inputs.

IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

2080-LC30-48QBB

DC Sinking Inputs – Inputs 00…27



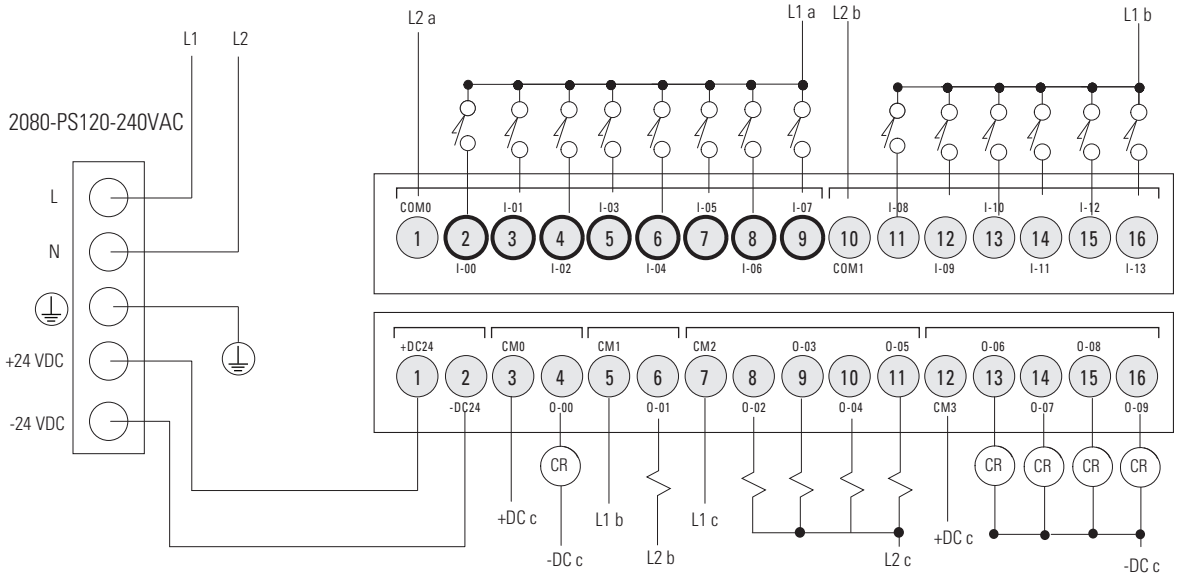
○ indicates high-speed inputs and outputs

IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

Micro850 Controller Wiring

2080-LC50-24AWB

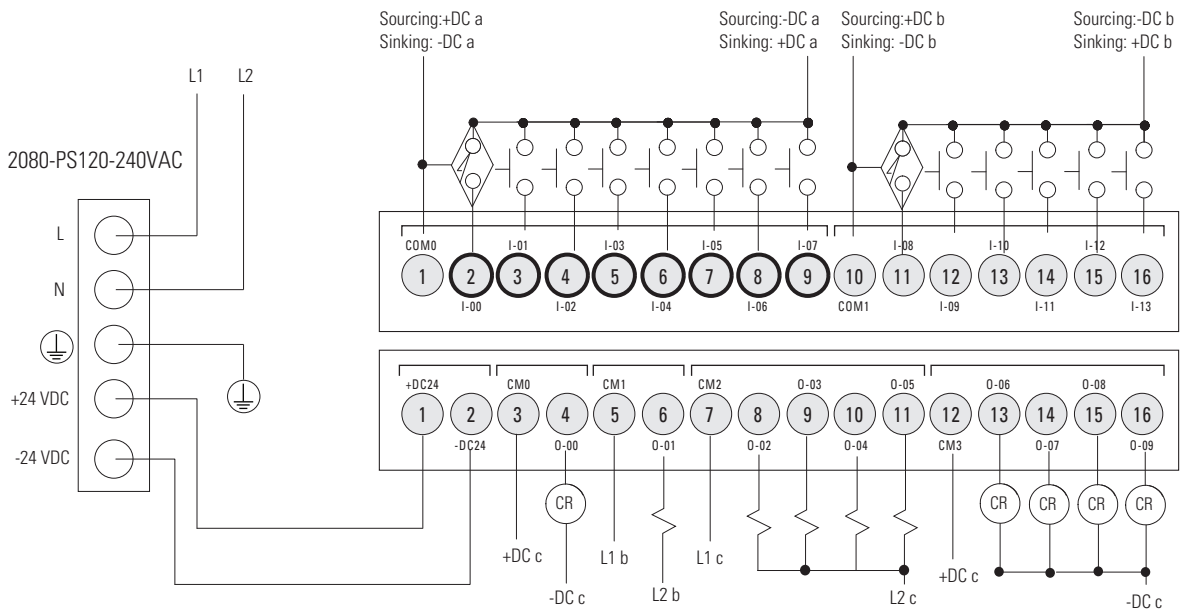
Inputs



IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

2080-LC50-24QWB

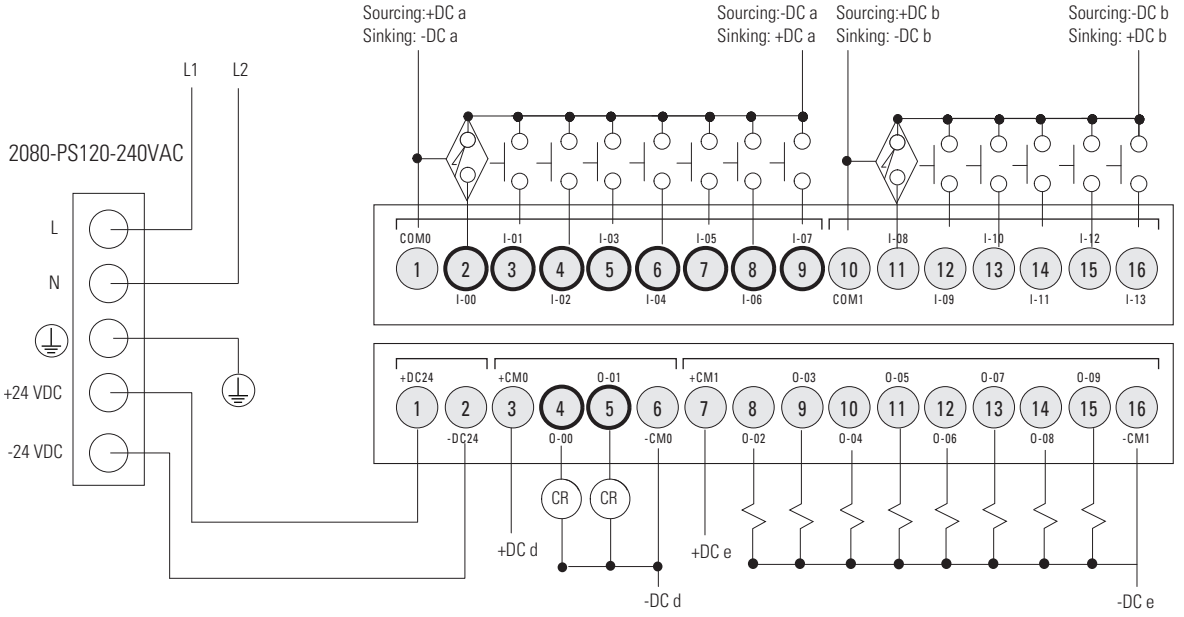
Inputs



IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

2080-LC50-24QBB

Inputs

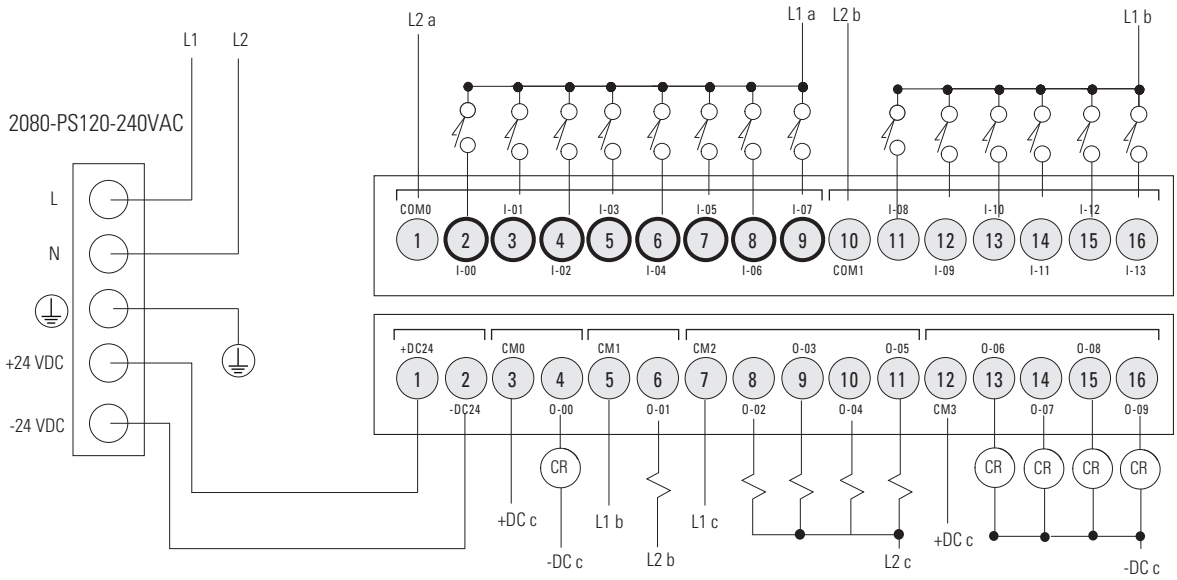


IMPORTANT Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.

Micro870 Controller Wiring

2080-LC70-24AWB

Inputs

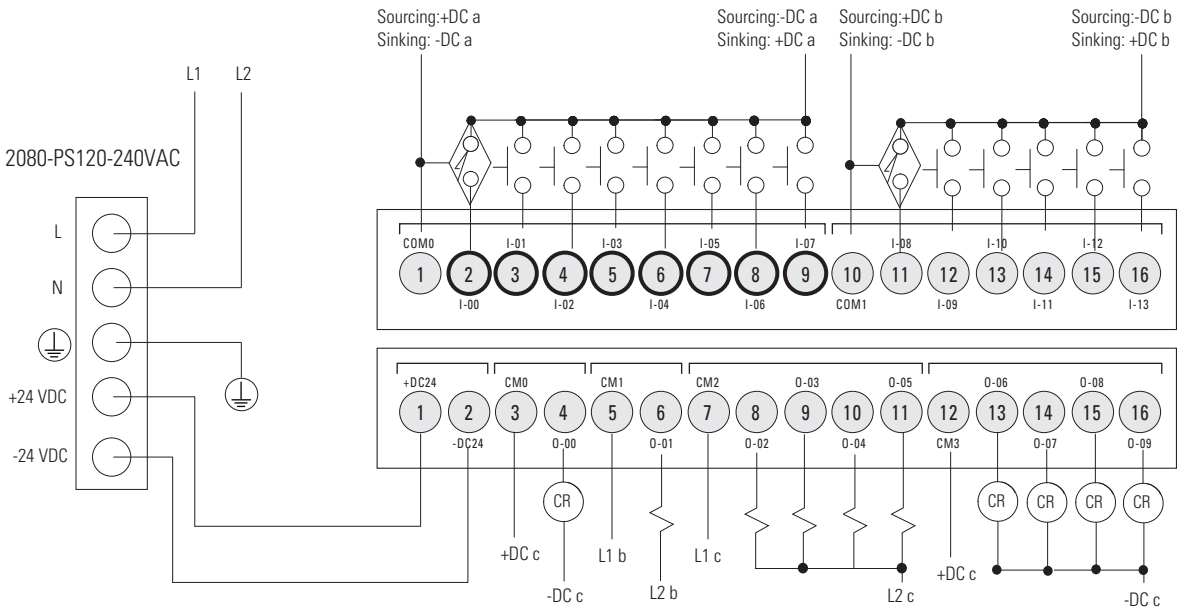


IMPORTANT

- Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.
- In Micro870 systems that use more than four Micro800 Expansion I/O modules, we recommend using a 1601-XLP60EQ power supply instead of a 2080-PS120-240VAC power supply. Make sure to wire both the Micro870 controller and 2085-EP24VDC expansion power supply to the same 1601-XLP60EQ power supply.

2080-LC70-24QWB, 2080-LC7024QWBK

Inputs

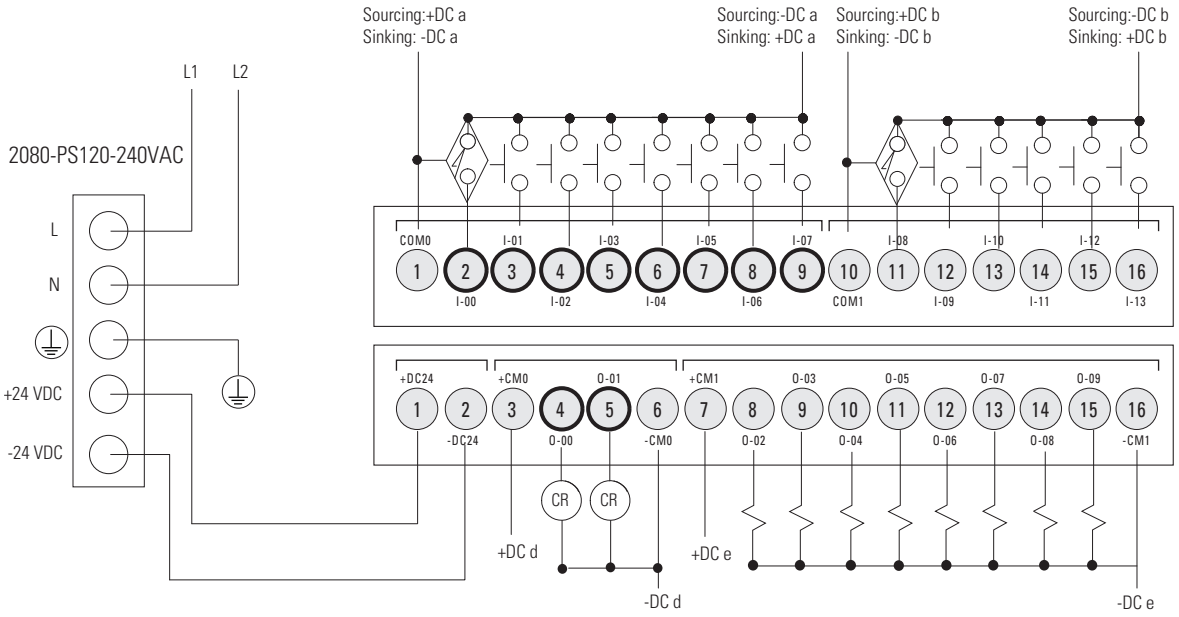


IMPORTANT

- Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.
- In Micro870 systems that use more than four Micro800 Expansion I/O modules, we recommend using a 1601-XLP60EQ power supply instead of a 2080-PS120-240VAC power supply. Make sure to wire both the Micro870 controller and 2085-EP24VDC expansion power supply to the same 1601-XLP60EQ power supply.

2080-LC70-24QBB, 2080-LC70-QBBK

Inputs



IMPORTANT

- Do not connect –DC24 (Output terminal 2) to Earth/Chassis Ground.
- In Micro870 systems that use more than four Micro800 Expansion I/O modules, we recommend using a 1601-XLP60EQ power supply instead of a 2080-PS120-240VAC power supply. Make sure to wire both the Micro870 controller and 2085-EP24VDC expansion power supply to the same 1601-XLP60EQ power supply.

Convert a MicroLogix Project to a Micro800 Project

Overview

This chapter describes two ways that you can convert your existing MicroLogix project to a Micro800 project – convert the project manually, or use the MicroLogix to Micro800 Converter tool.

Before You Begin

There is a new feature in Connected Components Workbench software version 12 or later that changes the instructions to be more similar to RSLogix 500/ RSLogix Micro. This feature is called the Logix theme and the purpose is to switch Connected Components Workbench software instructions into RSLogix 500/RSLogix Micro instructions. If you use Connected Components Workbench software version 11, you can get this feature by downloading the feature pack software update.

Overview of Program Execution

A Micro800 cycle or scan consists of reading inputs, executing programs in sequential order, updating outputs, and performing housekeeping (datalog, recipe, communications).

Program names must begin with a letter or underscore, followed by up to 127 letters, digits, or single underscores. Use programming languages such as ladder logic, function block diagrams, and structured text.

Up to 256 programs can be included in a project, depending on available controller memory. By default, the programs are cyclic (executed once per cycle or scan). As each new program is added to a project, it is assigned the next consecutive order number. When you start up the Project Organizer in Connected Components Workbench software, it displays the program icons based on this order. You can view and modify an order number for a program from the program properties. However, the Project Organizer does not show the new order until the next time the project is opened.

The Micro800 controller supports jumps within a program. Call a subroutine of code within a program by encapsulating that code as a User Defined Function (UDF) or User Defined Function Block (UDFB). A UDF is similar to a traditional subroutine and uses less memory than a UDFB, while a UDFB can have multiple instances. Although a UDFB can be executed within another

UDFB, a maximum nesting depth of five is supported. A compilation error occurs if this limit is exceeded. This limit also applies to UDFs.

Alternatively, you can assign a program to an available interrupt and have it executed only when the interrupt is triggered. A program assigned to the User Fault Routine runs once before the controller goes into Fault mode.

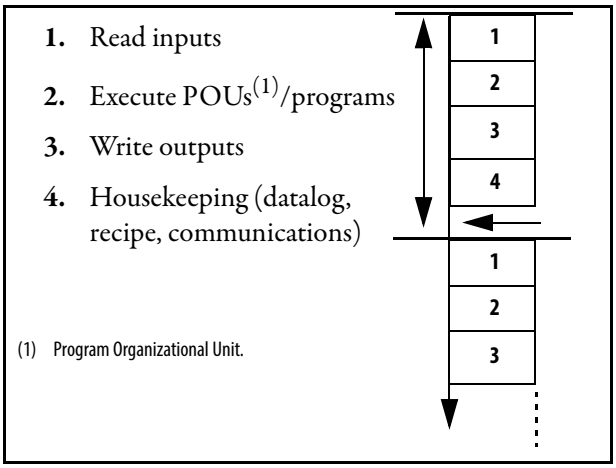
Besides the User Fault Routine, Micro800 controllers also support two Selectable Timed Interrupts (STI). STIs execute assigned programs once every setpoint interval (1...65535 ms).

The Global System Variables that are associated with cycles/scans are:

- __SYSVA_CYCLECNT – Cycle counter
- __SYSVA_TCYCURRENT – Current cycle time
- __SYSVA_TCYMAXIMUM – Maximum cycle time since last start.

Execution Rules

This section illustrates the execution of a program. The execution follows four main steps within a loop. The loop duration is a cycle time for a program.



When a cycle time is specified, a resource waits until this time has elapsed before starting the execution of a new cycle. The POUs execution time varies depending on the number of active instructions. When a cycle exceeds the specified time, the loop continues to execute the cycle but sets an overrun flag. In such a case, the application no longer runs in real time.

When a cycle time is not specified, a resource performs all steps in the loop then restarts a new cycle without waiting.

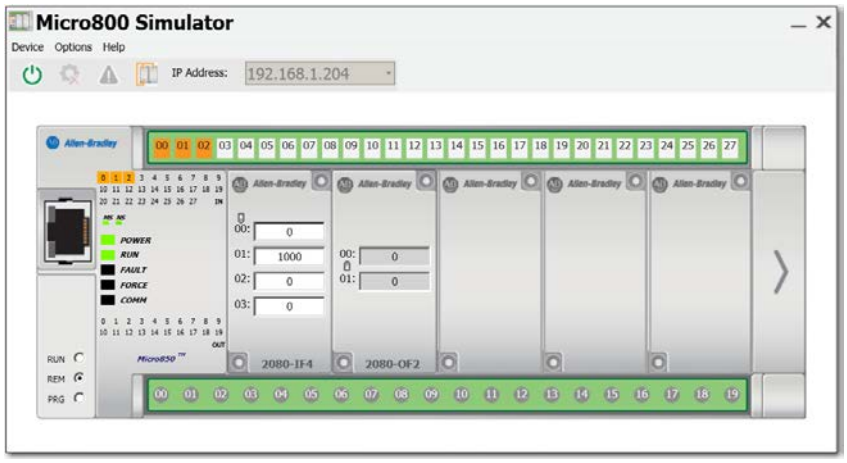
For more information, see the chapter, "Program Execution in Micro800 Controllers" in the Micro830, Micro850, and Micro870 Programmable Controllers User Manual, publication 2080-UM002.

High-Speed Counter

High-Speed Counter (HSC) is available in Micro800 controllers, however configuration and operation is done through Connected Components Workbench software instructions. The more common HSC configuration can be done from the software interface, however doing it through instructions gives you access to all configuration options. See the HSC example under the Logix examples section later in this chapter.

RSLogix Emulate 500 to Micro800 Simulator

While you can emulate your MicroLogix program in RSLogix Emulate 500, you can also do the same now for a Micro800 program with the new Micro800 Simulator feature in Connected Components Workbench software version 12 or later. The new Micro800 Simulator feature allows you to do more than emulate running your program – you can create your own virtual wiring, manipulate input signals in the simulator interface, and interact with the simulator with a customized simulator program.

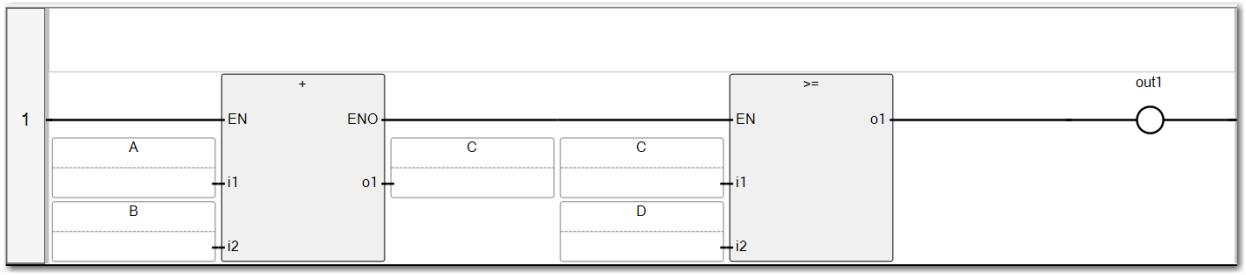


Logix Theme

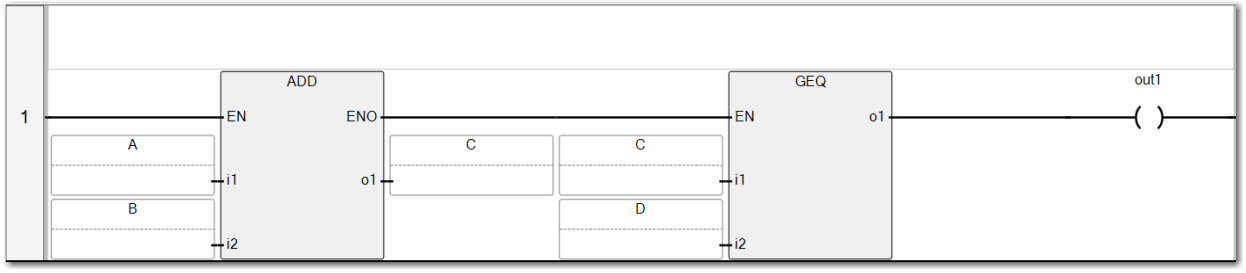
The Logix theme can be selected from the navigation bar. When you change to the Logix theme, all your ladder diagram instructions names are updated to how they were named in the RSLogix 500/RSLogix Micro software.

The following example shows the difference in instruction names between the two themes.

Instruction in Default Theme

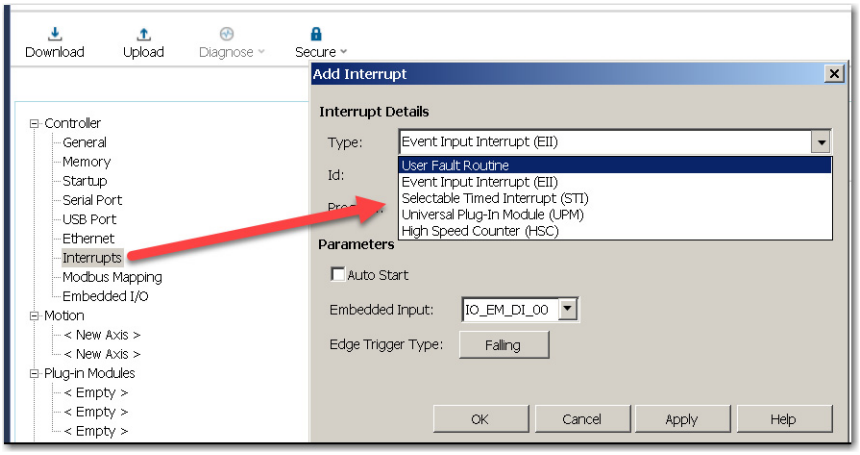


Instruction in Logix Theme



Interrupt

There are a few types of interrupts in the Micro800 controller –The User Fault Routine, Event Input Interrupt (EII), Selectable Timed Interrupt (STI), and High-Speed Counter (HSC) (applicable to embedded HSC only). These interrupts are configurable under the controller branch. For more information on interrupts, see the Connected Components Workbench software help.



It is important to understand how to configure your own interrupt because the MicroLogix to Micro800 Converter tool does not handle it. You have to identify and configure interrupts manually. For an example of how to configure an interrupt, see Configure Interrupts on a Micro800 Controller on page 179.

Copy and Paste Code Between Software

In Connected Components Workbench software version 12 or later, the copy and paste function is greatly enhanced. You can copy code (ladder rungs) from RSLogix 500/RSLogix Micro software and paste it directly into Connected Components Workbench software. For example, when you are migrating from MicroLogix controllers to Micro800 controllers, besides using the MicroLogix to Micro800 Converter tool, you can also copy the entire ladder diagram from your RSLogix 500/RSLogix Micro project and paste it into your Connected Components Workbench project. This enhancement makes reusing code easy.

After you paste the code into your Connected Components Workbench project, there are cases where you are required to make some changes manually. These cases are categorized as supported instruction, semi-supported instruction, and unknown instructions. The following sections briefly explain what to expect when you copy and paste a program.

Supported Instruction

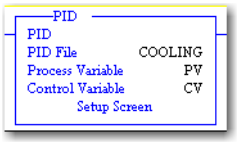
This group of instructions have identical functions to their respective instruction in RSLogix 500/RSLogix Micro software. Generally, you do not have to make any changes when you copy and paste such code. Most of the operator type instructions work like this, such as ADD, SUB, EQU.

Semi-supported Instructions

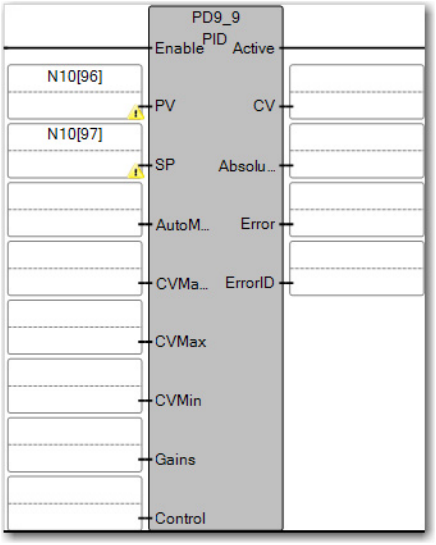
This category applies to slightly more advanced instructions. Timer and counter are two such instructions because when you copy and paste them into Connected Components Workbench software, some feature or parameter does not match. You must verify the logic and make necessary changes. For example, when you paste a PID instruction from RSLogix 500/RSLogix Micro software to Connected Components Workbench software, the instruction is not identical.

You have to convert the parameters and its associated output in the program manually.

PID instruction in RSLogix 500/ RSLogix Micro software



PID instruction in Connected Components Workbench software

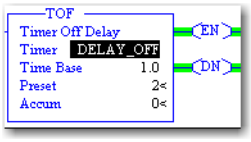


Unsupported Instruction

When you see 'UNK:xxx' in the Ladder Text Input box, it means that the 'xxx' instruction is not supported in Connected Components Workbench software. For example, program control instructions such as JSR, SBR, and MCR are treated as unknown instructions (UNK) as they are not required in Connected Components Workbench software.

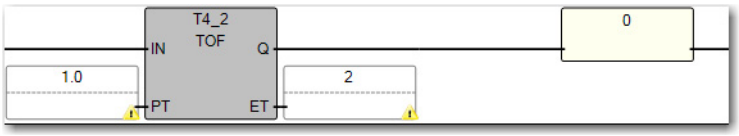
Sometimes an unknown instruction can result from converting a semi-supported instruction. When an RSLogix 500/RSLogix Micro instruction is converted to a Connected Components Workbench instruction that has fewer parameters, the excess parameters are dropped and considered as UNK. To determine how to configure the instruction, see the Connected Components Workbench software help.

TOF instruction in RSLogix 500/ RSLogix Micro software



The instruction is configured with a two second delay.

TOF instruction in Connected Components Workbench software



The unknown instruction '0' is dropped out of the TOP instruction.

What You Need

You need the following software to perform the conversion.

- Connected Components Workbench software version 12 or later.
- RSLogix 500/RSLogix Micro software.

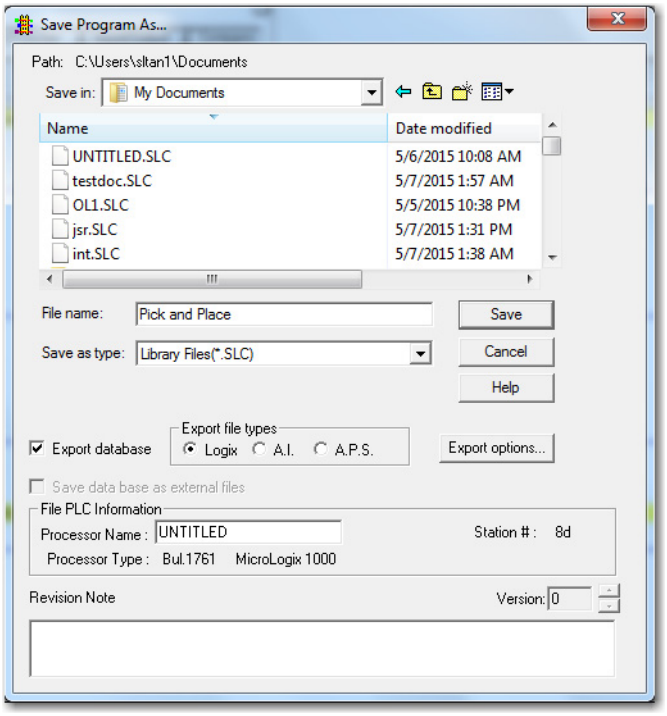
Convert Your Project with the Converter Tool

The general steps to convert your project with the MicroLogix to Micro800 Converter tool can be summarized as follows:

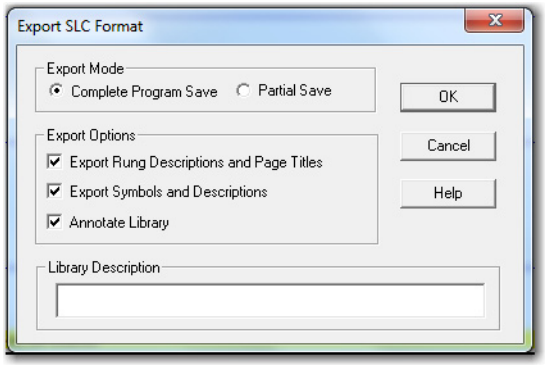
1. Save the RSLogix 500/RSLogix Micro Project as an SLC File
2. Run the MicroLogix to Micro800 Converter Tool
3. Convert the SLC File to a Connected Components Workbench Project
4. Understand the Conversion Process
5. Resolve Compilation Errors

Save the RSLogix 500/RSLogix Micro Project as an SLC File

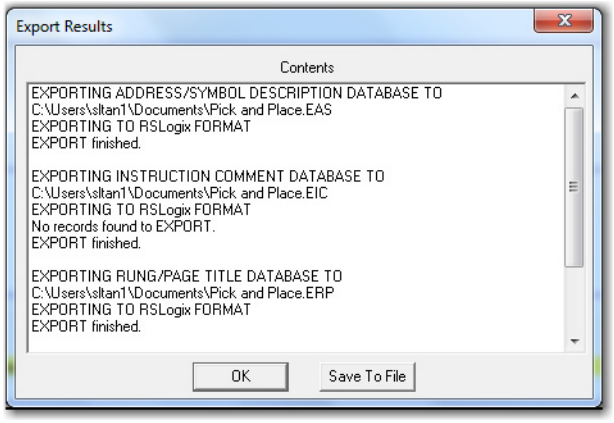
1. Open the RSLogix 500/RSLogix Micro project file (.RSS) that you want to convert.
2. Save the opened project file as a .SLC file with the following settings:
   - Save as type = Library Files (*.SLC)
   - Export database = Selected
   - Export File types = Logix



3. Click Save.
   The Export SLC Format dialog box displays.



4. Click OK.
   The Export Results dialog box displays.



5. Click OK to close the dialog box.

MicroLogix to Micro800 Converter Tool

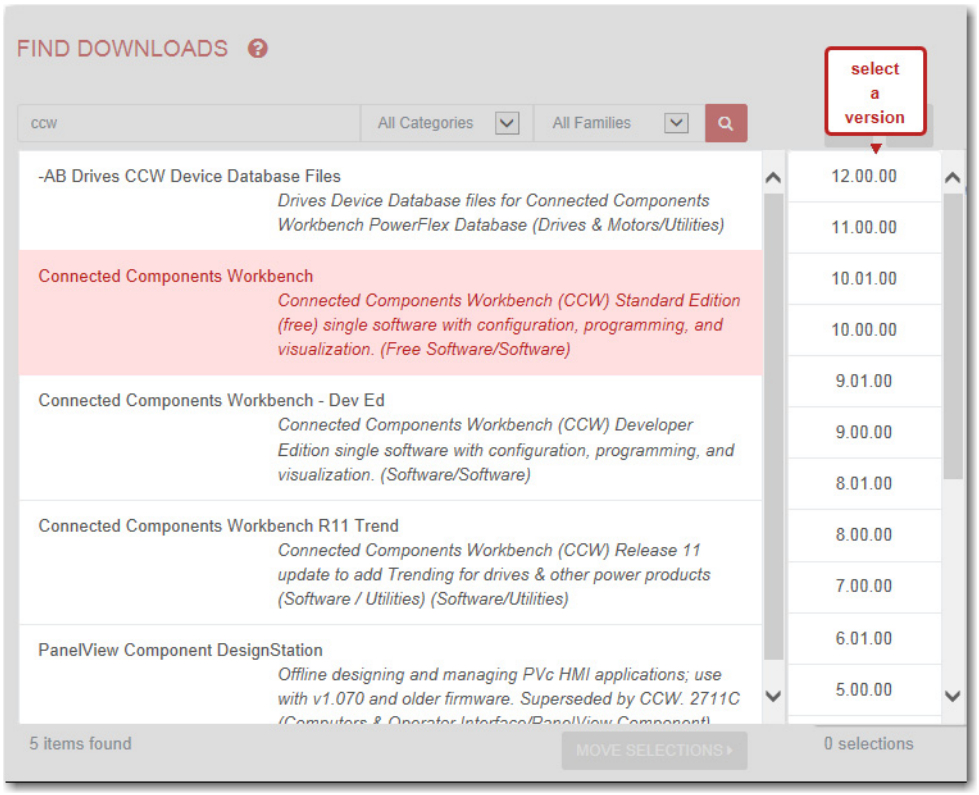
The benefits of using the MicroLogix to Micro800 Converter tool would be faster conversion time and that the instructions have similar input/output parameters to MicroLogix instructions. However, there is larger memory consumption by the User-Defined Function Block (UDFB) programs.

IMPORTANT The converter tool only supports MicroLogix controllers. The tool does not work with other controllers even if you convert their project files to .SLC files.

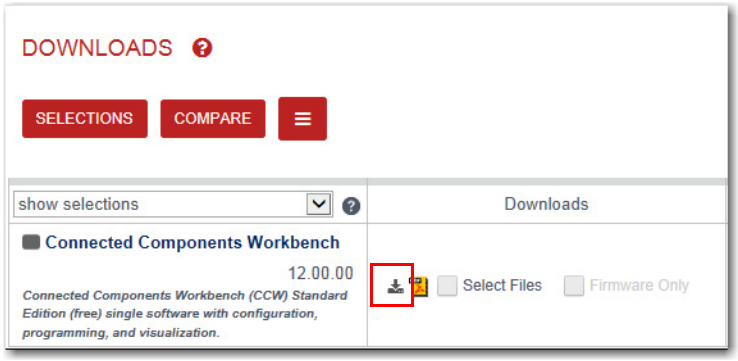
For general information about the converter tool, see the MicroLogix to Micro800 Converter tool help.

Download and Install Connected Components Workbench Software

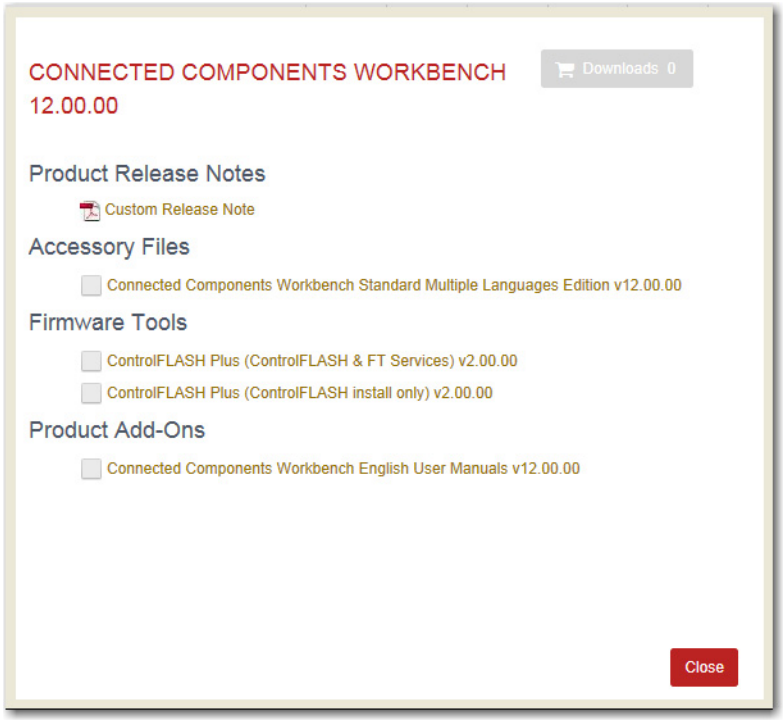
1. Open the following link on your browser:
   https://www.rockwellautomation.com/rockwellautomation/support/pcdc.page
2. Click Find Downloads.
   The Find Downloads page displays.
3. Enter the keyword 'CCW' in the search bar.
4. Select Connected Components Workbench Standard Edition (select Developer Edition if you have a valid activation key), then select version 12.00.00.



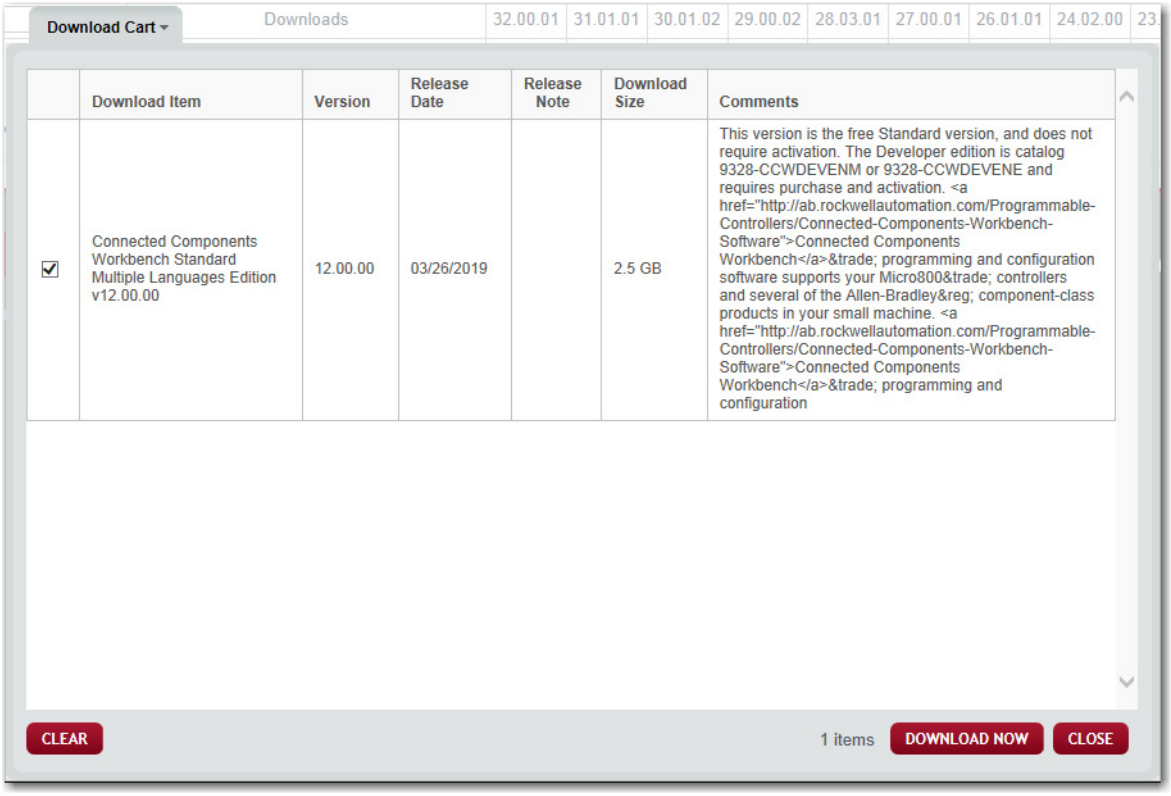
5. Click Downloads.
   The Downloads page displays.
6. Click the Show Downloads icon.



The Available Downloads dialog displays.



7. Select the Connected Components Workbench software, then click Downloads.
   The Download Cart dialog displays.

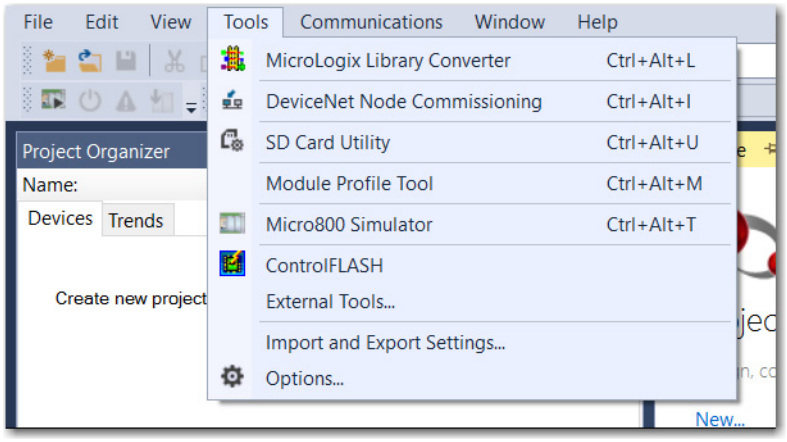


8. Click Download Now.
   If you are not signed-in to the website, you are prompted to do so.
9. Download and install the application.

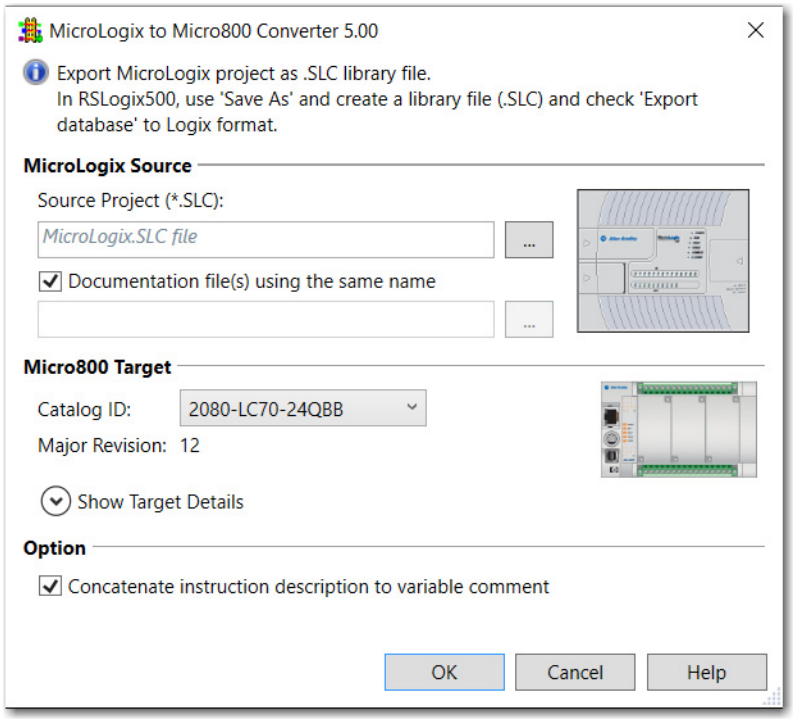
Run the MicroLogix to Micro800 Converter Tool

There are two ways to run the MicroLogix to Micro800 Converter tool:

- From the Connected Components Workbench menu,
  Select Tools -> MicroLogix Library Converter.



The MicroLogix to Micro800 Converter tool dialog displays.



- By command-line execution
  The command must be executed in the Connected Components Workbench installation directory with the following syntax:

```
CCW.Shell.exe/MicroLogixConv SourceSlcFilePath TargetCatalogID [optionConcatenateComment]
```

Argument	Description
SourceSlcFilePath	Provides the path to the .slc file to be converted.
TargetCatalogID	Specifies the Catalog ID for the target controller.
optionConcatenateComment	Has a value of either True or False. Determines whether the instruction description is shortened to just the comment.

In the following example, the SLC file that is named 'Pick and Place' is converted for use with a Micro830 controller (catalog number 2080-LC30-16QWB) and to not concatenate the instruction descriptions.

```
C:\Program Files (x86)\Rockwell Automation\CCW>ccw.shell.exe/MicroLogixConv Pick
 and Place.slc 2080-LC30-16QWB false
```

Convert the SLC File to a Connected Components Workbench Project

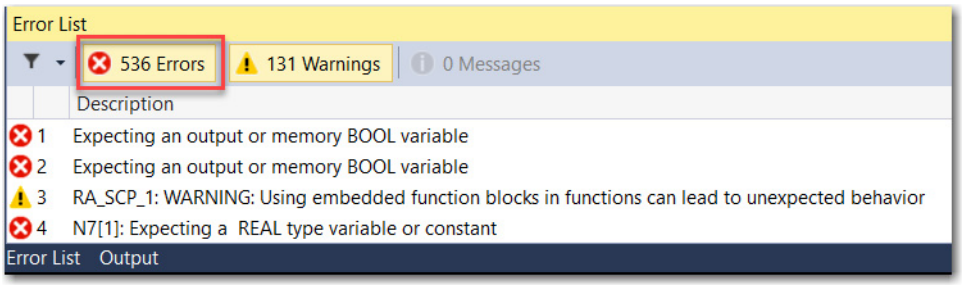
1. Run the MicroLogix to Micro800 Converter tool.
   From the Connected Components Workbench menu, select Tools –> MicroLogix Library Converter.
2. Under MicroLogix Source, do the following:
   a. In Source Project (*.SLC), locate your saved SLC file.
      Verify that the documentation files are in the same directory.
   b. Select one or more documentation files with the same name checkbox.
3. Under Micro800 Target, a compatible Micro800 catalog number automatically populates the catalog ID field. If you want, you can select another catalog number.

   TIP The MicroLogix to Micro800 Converter tool help lists the recommendations for which MicroLogix controller is converted to which Micro800 controller.

4. Under Option, select the checkbox if you want to concatenate the instruction description to variable comment.
5. Click OK to generate the Connected Components Workbench project.

Once the project is converted, you must make additional changes before it can be used. If you build the project immediately after conversion, there are many errors

and warning messages. Follow the rest of this guide to understand what the errors mean and how to resolve them.



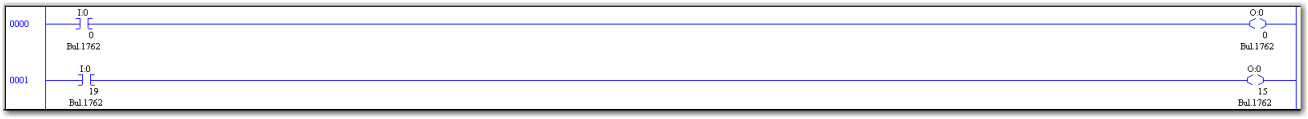
Understand the Conversion Process

The MicroLogix to Micro800 Converter tool helps to convert your program, I/O table, and user documentation at a basic level. As a result, if you compile your project immediately after conversion, there are many errors and warnings. This section describes what the converter tool does and explains why those errors and warnings appear.

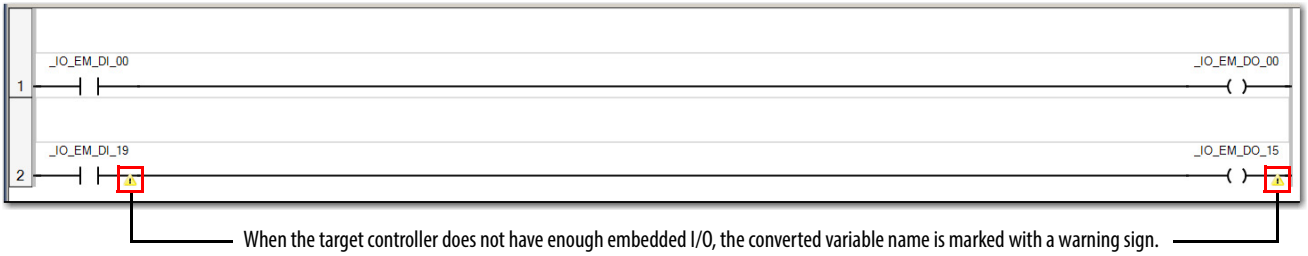
Automatically Rename Embedded I/O

When you convert a project, all embedded I/O are renamed automatically. For example, I:0/0 is renamed to _IO_EM_DI_00. If all embedded points are converted properly, you do not need to reassign any I/O.

MicroLogix embedded I/O terminals



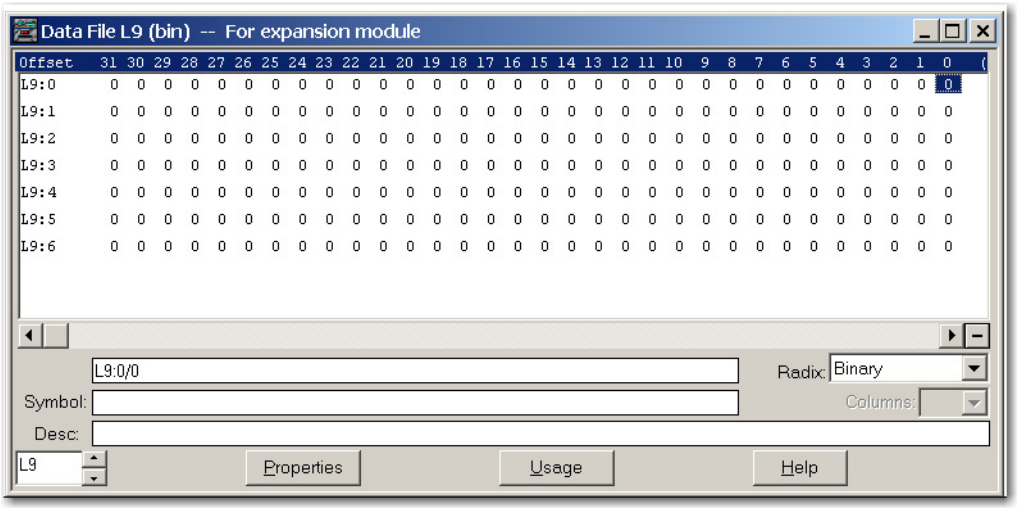
Micro800 embedded I/O terminals



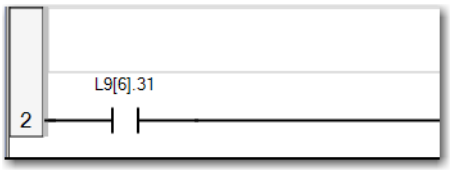
However, there are some situations where you must reassign I/O. For example, when you migrate from a 40-point MicroLogix controller with five or six expansion I/O modules, to a 24-point Micro870 controller. Since only Micro870 controllers support more than four expansion I/O modules, you must assign the remaining MicroLogix embedded I/O points to the Micro870 expansion I/O modules.

The converter tool version 5.0 does not reassign terminals on the expansion I/O modules automatically. This behavior can cause confusion as the converter tool renames the terminals on the modules to the Micro800 embedded I/O terminals, and results in duplicate outputs.

To resolve this error, you have to rename all expansion I/O addresses. A recommended method is to create a Long data file (For example, L9) and rename all expansion I/O addresses to the bit address of the data file. For example, rename the address I:6/31 to L9:6/31, retaining the slot and channel number for reference.



With this method, when you migrate into Connected Components Workbench software, L9:6/31 is renamed to L9[6].31. Since the slot and channel number information is converted into the new array data format, it is easy to reassign them manually to the respective I/O terminal on the Micro800 expansion module.



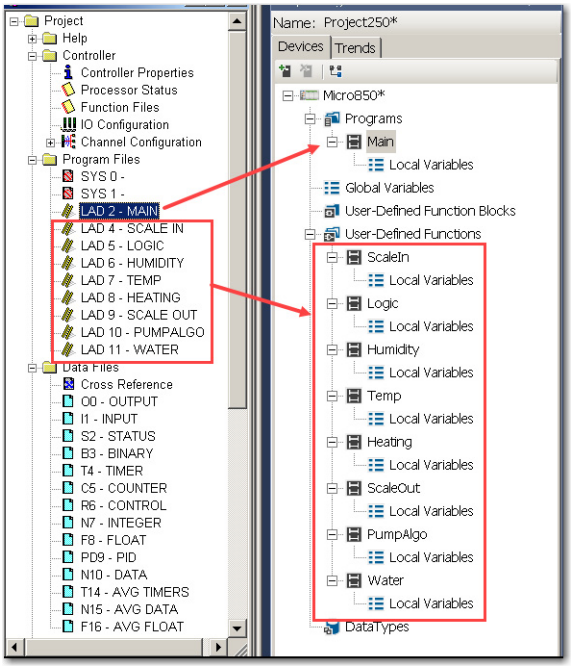
This address, L9[6].31 can be denoted as channel 31 of the sixth expansion module on the MicroLogix controller.
This method makes it easy to map the I/O onto a Micro800 expansion module.

Program and Subroutine

If the project is converted successfully, there is only one ladder diagram in your program file and the rest of the ladder programs are located under the User-Defined Functions (UDFs) branch. The converter tool converts subroutines into UDFs and your program calls these functions in the main routine. However, if any of the subroutines is initiated by an interrupt, you must convert the interrupt routine manually. See the Connected Components Workbench software help on

how to create an interrupt program. You can also see Configure Interrupts on a Micro800 Controller on page 179 for more information.

RSLogix 500/RSLogix Micro **Connected Components Workbench**



MicroLogix Instructions

Most of the compilation errors are generated due to instructions. Not all MicroLogix instructions have their equivalent in Micro800 instructions. Some instructions are similar but have differences that require minor changes to make the Micro800 instructions behave the same as the MicroLogix instruction.

The help file for the converter tool contains a list of the supported and unsupported MicroLogix instructions. Unsupported instructions are probably not needed anymore due to the difference in design between MicroLogix controllers and Micro800 controllers. For supported instructions, there are a few categories as described here.

- **Direct Replacement**

  MicroLogix instructions that fall under this category are basic instructions such as coil, contact, and basic math instructions such as ADD, SUB, MUL.

- **Similar Replacement**

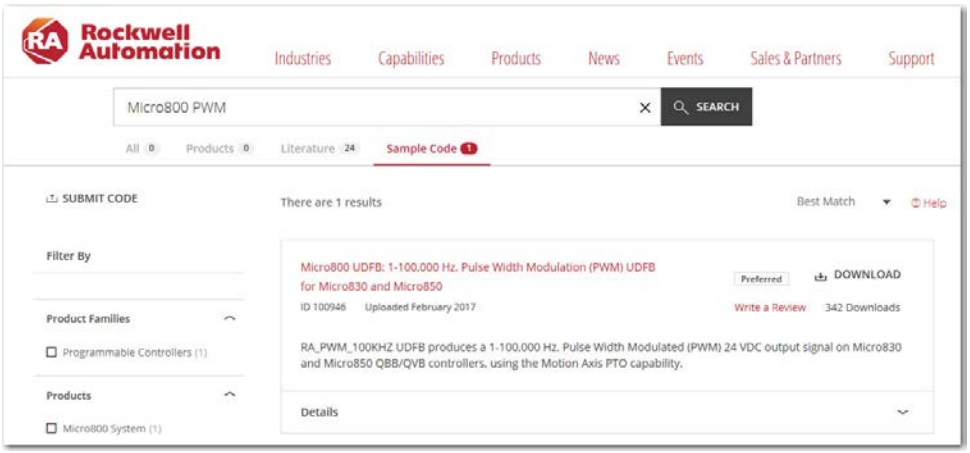
  MicroLogix instructions that fall under this category means that the same instruction is available in Micro800 controllers, but the instruction has less or more parameters. The Micro800 instructions function slightly differently, but they can easily be configured to achieve the same result as the MicroLogix instructions. Examples of similar instructions are ONS, OSF, OSR, counter, and timer.

- **UDFB Placeholder**

  MicroLogix instructions that fall under this category require more effort to resolve. For example, the MicroLogix PWM instruction. A UDFB placeholder is a function block without code, or empty function block. Since a Micro800 controller has its own timer, you are expected to use this timer to develop your own PWM function within the block.

  Alternatively, an example of the Micro800 PWM UDFB code is available on the Rockwell Automation Sample Code Library at https://www.rockwellautomation.com/global/sample-code/overview.page.

  This code is not identical to the MicroLogix PWM instruction, but you can use it as a guide to configure the UDFB to suit your application.



- **UDFB Replacement**

  MicroLogix instructions that fall under this category are replaced automatically by the converter tool. UDFBs that replace supported MicroLogix instructions are developed to behave as close as possible to the original instruction. Examples of UDFBs are RA_TON_MICROLOGIX, RA_CTD_MICROLOGIX, and RA_CTU_MICROLOGIX.

Data Table Addresses

The MicroLogix processors store all data in global data tables. You can access this data by specifying the address of the data you want. A Micro800 controller supports data that is local to a program, and data that is global to all tasks within the controller. A Micro800 controller can also share data with other controllers, and instead of addresses, you use tags to access the data you want.

Each MicroLogix data table file can store several words of related data. A Micro800 controller uses arrays to store related data. The converter tool converts the MicroLogix data table files to Micro800 arrays.

With a Micro800 controller, you use a tag (alphanumeric name) to address data (variables). The controller uses the tag name internally and does not need to cross-reference a physical address. The following are some examples of how a MicroLogix data table is mapped to Micro800 variables.

MicroLogix Address	Map to Micro800 Address
N7:500	N7[500]
N17:25	N17[25]
R6:100	R6[100]
C5:0	C5[0]
T4:6	T4[6]
I:0/5	_IO_EM_DI_05
O:0/4	_IO_EM_DO_04

For an extended list of examples, see the MicroLogix to Micro800 Converter tool help.

User Documentation

If the Logix database was exported when your RSLogix project as a .SLC file, the symbols, address comments, instruction comments, and rung comments in the program are preserved. There are some limitations on the number of characters and descriptions for the comments. For example, certain strings become concatenated. To understand more about this limitation, see the help file for the converter tool for details.

Unconverted Project Components

The following is a list of components that are not converted.

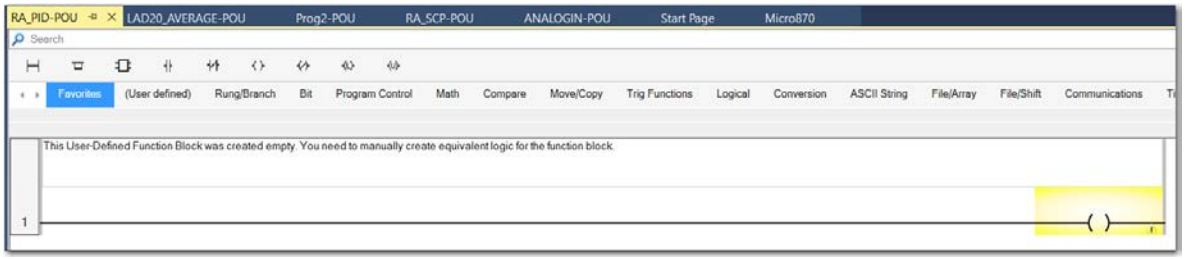
Project Component	Corrective Action
Status data file features	Although the status file is not supported, certain status file functions are available in Micro800 controllers. For example, 'First Scan' is available as a system variable in Micro800 controllers.
Data file properties	Addressing in Micro800 controllers is based on tags. These tags can be manipulated manually.
Index and indirect addressing for Status, Input, and Output data files	Handle this component programmatically.
Controller configuration	Configure the controller manuall.y
I/O data tables	Add the I/O manually.
I/O configuration	Configure the I/O manually.
Interrupts	Configure the interrupts manually.
Passwords	Configure the controller password manually.
Indirect addressing	Implement indirect addressing manually.
Function files	The design of Micro800 controllers eliminates the use of function files, but configuration of functions is done through instructions or I/O configuration. Instructional behavior differences are described in RLL Instruction Mapping on page 99.

Resolve Compilation Errors

This section describes how to handle the various errors that were described in the previous section. Application-related issues due to behavior differences between MicroLogix instructions and Micro800 instructions are not covered. The focus is on to solve error messages that are generated from the conversion.

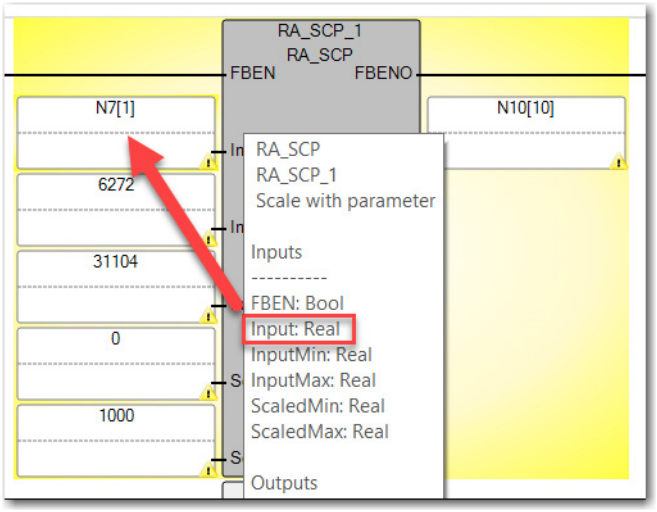
Expecting an Output or Memory BOOL Variable – UDFBs

Most of the time this error message appears due to the UDFB placeholders. Since most instructions are converted into UDFBs, there may be many empty blocks that require your attention. Developing the application code for these UDFBs helps to resolve a significant number of errors because a particular UDFB function may be used multiple times within a project.



Expecting a REAL Type Variable or Constant – Data Type

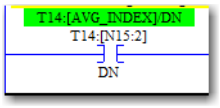
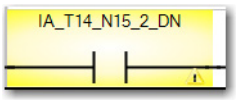
Instruction parameter data type is difference between MicroLogix controllers and Micro800 controllers. Many of the INT arrays that are created by the converter tool are used in Micro800 instructions that require a REAL data type. Resolving data type errors typically reduces the number of errors significantly.



Undeclared Identifier – Variable not Declared

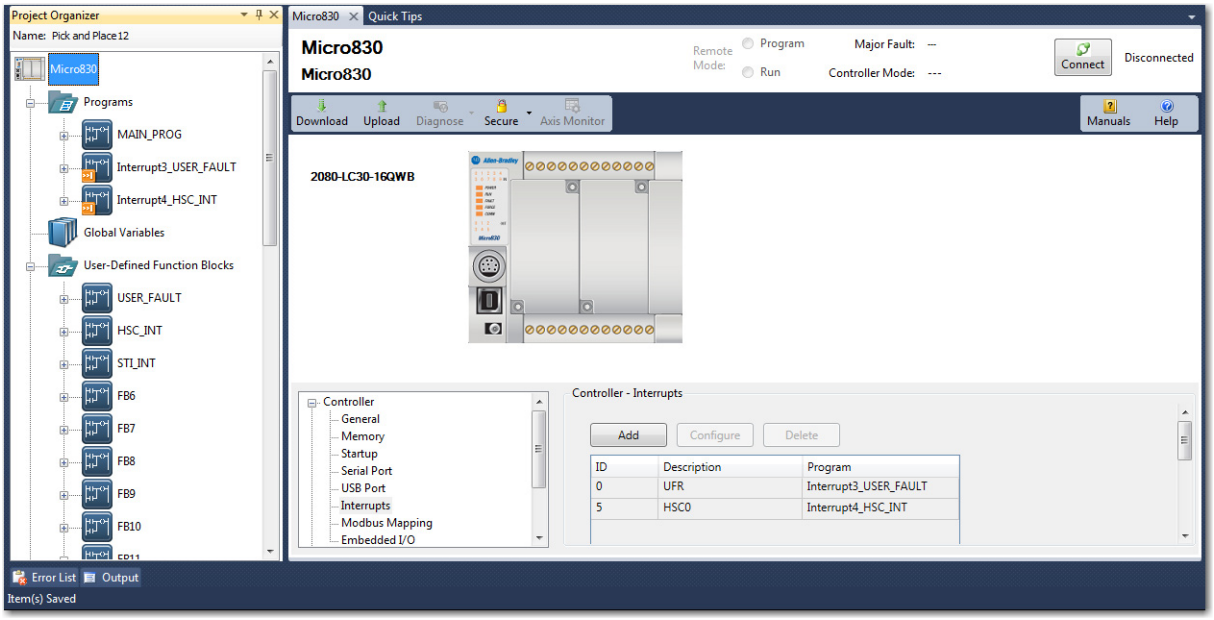
This variable arises from a converted project and is an indication that the MicroLogix address format is not supported. The converter tool automatically renames the address to something that is supported. You must resolve this error programmatically.

The following is an example of the indirect addressing error.

MicroLogix Address Format	Converted Micro800 Variable
	

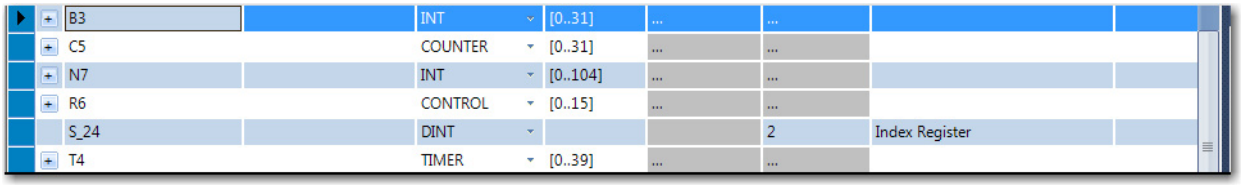
Notes on a Converted Pick and Place Application

The converted SLC file appears similar to the following:

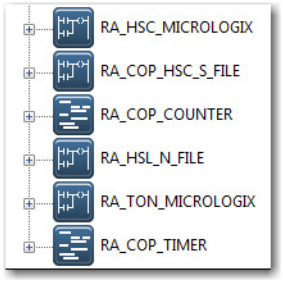


- By default, the Interrupt3_USER_FAULT POU program is created. From the RSLogix 500/RSLogix Micro project, the MAIN_PROG (Lad 2) program is converted to “MAIN_PROG” in Connected Components Workbench, while USER_FAULT (Lad 3), HSC_INT (Lad 4), STI_INT (Lad 5), and Lad 6 to Lad 16 from the RSLogix 500/RSLogix Micro project are converted to a User Defined Function Block (UDFB).
- The converter tool adds other Interrupts (for example, Interrupt4_HSC_INT) if any HSC-related instructions are present in the project.
  You can configure it to an interrupt from the controller configuration section or delete the interrupt program if the interrupt is not needed in the program.
- For this project, the Interrupt3_USER_FAULT can be deleted when not in use.

- For Jump to Subroutine (JSR), the converter tool creates a call to the UDFB (subroutine that is called in JSR) in the program.
- The following data types are converted into an array (under Global Variables).

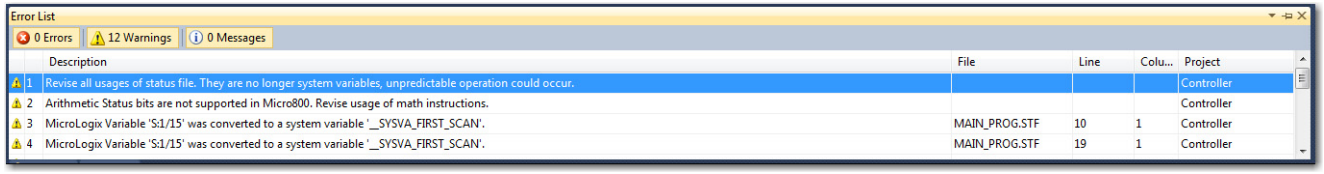


- The partially supported MicroLogix 1000 instructions are converted into a user-defined function block (UDFB) that contains logic to perform the function of the MicroLogix instruction.

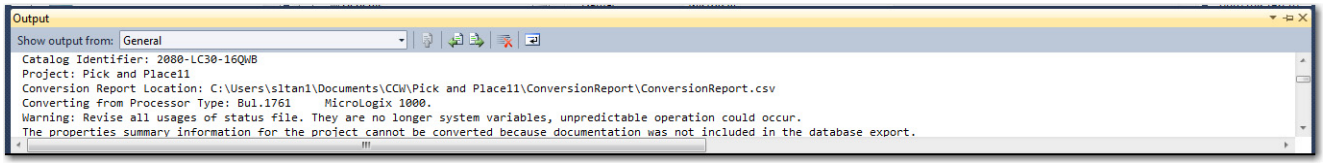


Refer to the converter tool help for more information about the fully supported, partially supported, and unsupported MicroLogix 1000 instructions.

- Results are displayed in the Output and Error List windows at the end of the conversion process, and stored in the conversion log file.
  The Error List window provides information about the items in that did not fully convert from your source project to your new project.



The Output window displays the location of the Conversion Report in CSV format, where you can find all information from the Error list.



Once no errors are encountered and all warnings have been addressed, save the project and test to see if it has the desired behavior.

See Appendix B to view the Pick and Place project in RSLogix 500/RSLogix Micro and in Connected Components Workbench after you have used the converter tool.

Convert Your Project Manually

To migrate all your existing RSLogix 500/RSLogix Micro project code, you have to create a Connected Component Workbench project with a similar structure. This section describes how to create your original program structure in the Connected Components Workbench software environment and how to handle I/O addressing differences.

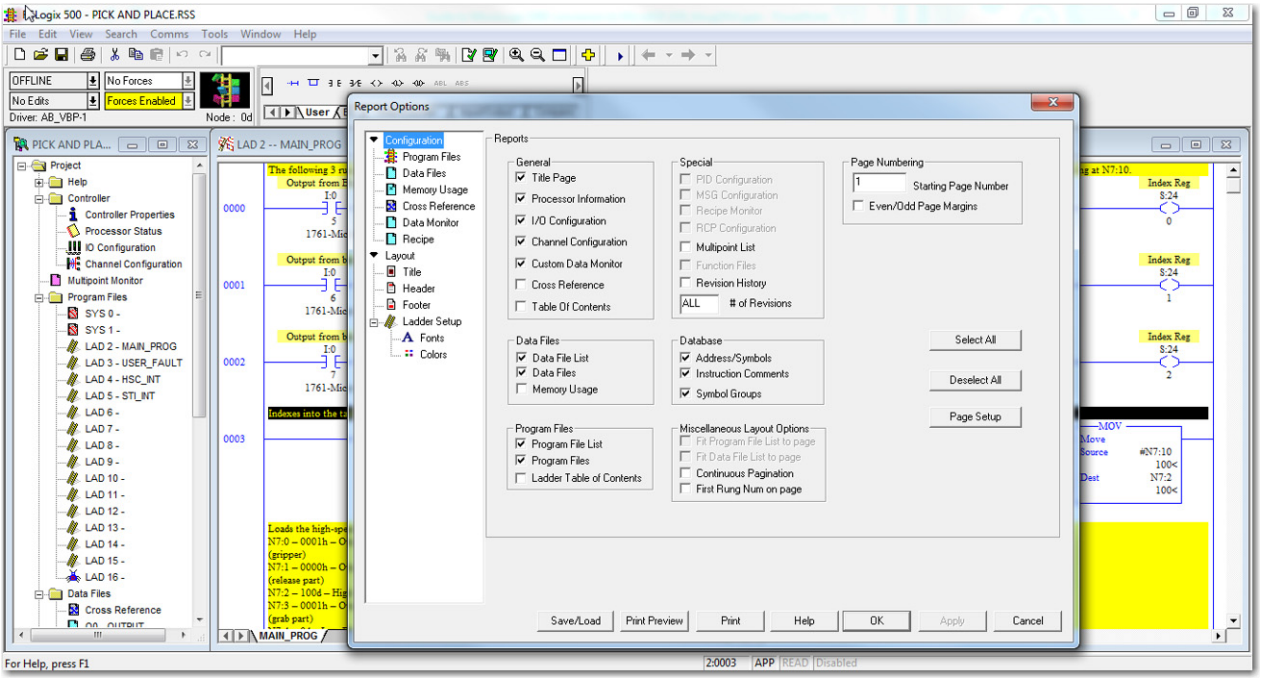
The general steps for manual conversion can be summarized as follows:

- Generate an Existing RSLogix 500/RSLogix Micro Project Report
- Create Equivalent Program Files
- Create Representative Data Files
- Create Equivalent Logic in Program File

Generate an Existing RSLogix 500/RSLogix Micro Project Report

The RSLogix 500/RSLogix Micro report for your application contains information such as a program file list, a data file list, and ladder diagrams. This information is used as reference when creating a program in Connected Components Workbench software.

1. Open the existing RSLogix 500/RSLogix Micro project.
2. Go to File –> Report Options.



3. Modify the report options as necessary.
4. Click Print.

Create Equivalent Program Files

The program execution between MicroLogix controllers and Micro800 controllers is different. To understand more about Micro800 controller execution order, review the earlier section Overview of Program Execution on page 59.

To create the equivalent program files for your RSLogix 500/RSLogix Micro project in Connected Components Workbench software, do the following.

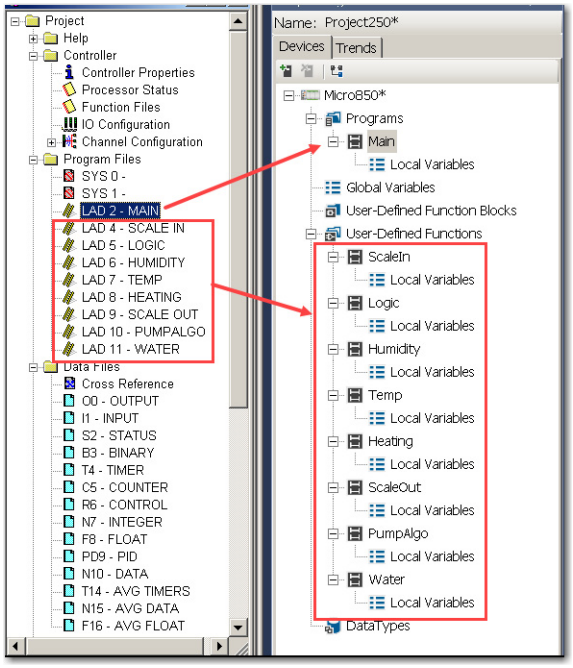
1. Launch the Connected Components Workbench software.
2. Click New.
3. Enter a project name and click Create.
   Verify that add device on create checkbox is selected, otherwise the add device menu does not appear.
4. From the device catalog, expand 'Controllers' and select your target controller.
   You can see the earlier section to determine a suitable target controller.
5. Select the latest firmware revision and click Add to Project.
6. In the Project Organizer, right-click Program to add a ladder diagram as your main routine.

Take note that subsequent programs are not subroutine. All programs under project organizer are executed in sequential order, unless they are configured for STI or interrupt. You can incorporate all subroutine logic into one ladder diagram or create a UDF or UDFB to replace the subroutine.

Create UDF or UDFB to Replace Subroutine

1. Right-click UDFB or UDF under project organizer and add a ladder diagram.
2. Rename the ladder diagram according to your subroutine.

3. Repeat these steps until a UDF or UDFB has been created for each subroutine.



4. Save your project.

Create Representative Data Files

Use the report that is generated for your project as a reference to create representative data files in Connected Components Workbench software by configuring the variables for the project. If your configuration requires expansion I/O modules, do it here. This section provides some guidance for configuring embedded I/O, binary file, and integer file. If your project requires you to configure more data files for expansion I/O or other data types, use this section as a guide to continue configuring the other data files.

The following data files are used in the examples in this section:

- Output (O0)
- Input (I1)
- Binary (B3)
- Integer (N7)

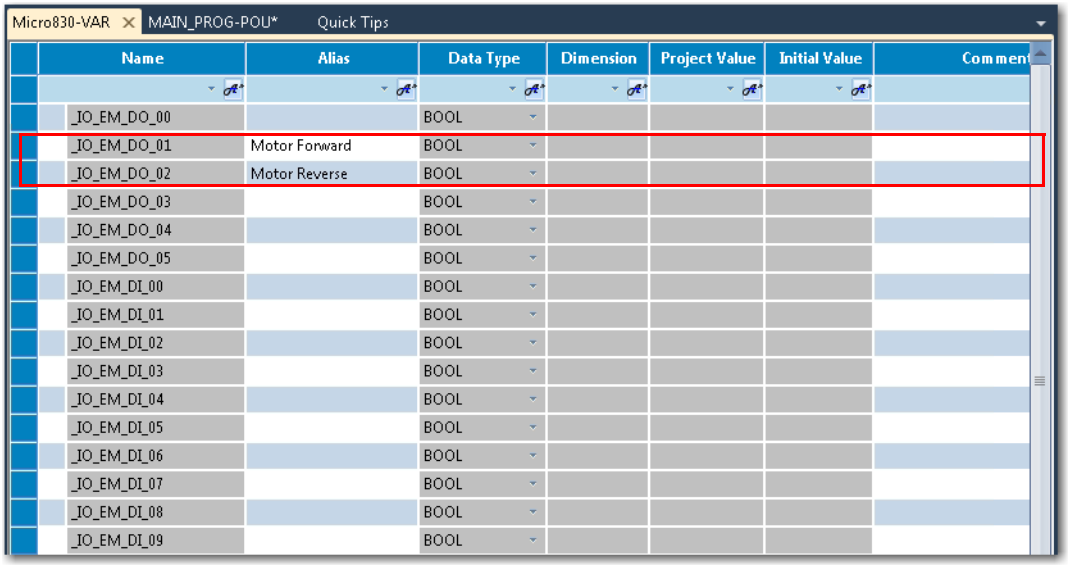
Set Up Embedded I/O Variables

1. On the Project Organizer panel, double-click Global Variables.
2. Create an alias for the embedded I/O following the comments in RSLogix 500 software.

Follow this Connected Components Workbench I/O addressing:

- OUTPUT (O0) = _IO_EM_DO_XX
  (For example, O:0/0 = _IO_EM_DO_00)
- INPUT (I1) = _IO_EM_DI_XX
  (For example, I:1/0 = _IO_EM_DI_00)

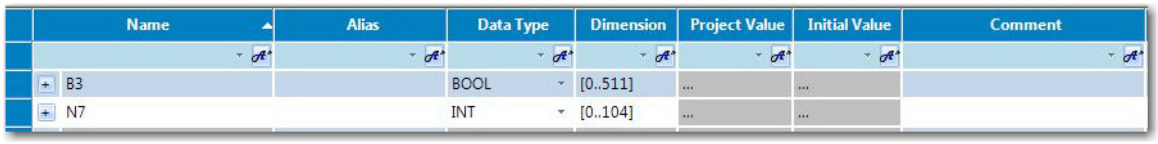
Embedded I/O Address in RSLogix 500/RSLogix Micro Software	Connected Components Workbench Global Variable Name	Alias in Connected Components Workbench Software
0:0/1	_IO_EM_DO_01	Motor Forward
0:0/2	_IO_EM_DO_02	Motor Reverse



Substitute Binary (B3) and Integer (N7) Data Files

Create arrays to substitute the data files BINARY (B3) and INTEGER (N7) in Connected Components Workbench software. You need Connected Components Workbench software version 8 or later to support array of integer bit addressing.

1. On the Project Organizer panel, double-click Global Variables.
2. Enter the N7 and B3 variables as shown in the following example:



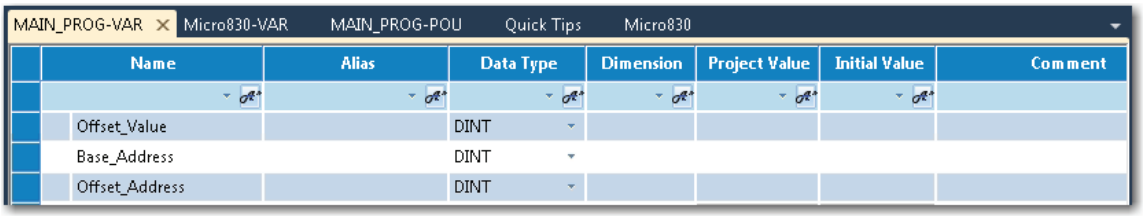
Set Up Variables for Index Addressing

In this example, Index Addressing is used to store the encoder counts of the bin locations (total eight bins).

The section shows the steps for configuring the variables that are needed for programming Index Addressing in Connected Components Workbench software. Programming steps are shown in Program Index Addressing on page 84.

1. On the Project Organizer panel, under MAIN_PROG, double-click Local Variables.
2. Add the following variables:

Name	Data Type
Offset_Value	DINT
Base_Address	DINT
Offset_Address	DINT



- Offset_Value is the equivalent of the Index Register (S:24 in RSLogix 500)
- The sum of the Offset_Value and Base_Address is the Offset_Address.
- The data stored at N7[Offset_Address] is then passed to N7[2].

Create Equivalent Logic in Program File

This section describes how to transfer the logic from an RSLogix 500/RSLogix Micro program. In Connected Components Workbench software version 12 or later, copy and paste between the RSLogix 500/RSLogix Micro and Connected Components Workbench software environment is supported. You can select all logic in an RSLogix 500/RSLogix Micro program, copy, and paste it into a Connected Components Workbench program.

However not all instructions are supported. Therefore, some rungs do not appear as expected. You must fix the unsupported instructions and syntax-related errors manually.

1. To paste code from an RSLogix 500/RSLogix Micro program, you must change Connected Components Workbench software to the Logix theme. This theme allows the ladder editor to understand the code that is being pasted.

2. Copy your main program (LAD 2) from the RSLogix project and paste it in your main ladder diagram of your Connected Components Workbench project.
3. Since Connected Components Workbench software does not support the JSR instruction, you must replace all JSRx instructions with their respective UDFB or UDF that were created previously.
   a. Double-click the JSRx instruction to bring up the Instruction Block Selector.
   b. Search for the respective UDF or UDFB.
   c. Select it and click OK to insert.
   d. Repeat step 3 until all JSRx instructions have been replaced with their respective UDF or UDFB.
4. Copy the logic from the subroutine and paste them into their respective UDFB or UDF that were created for them.

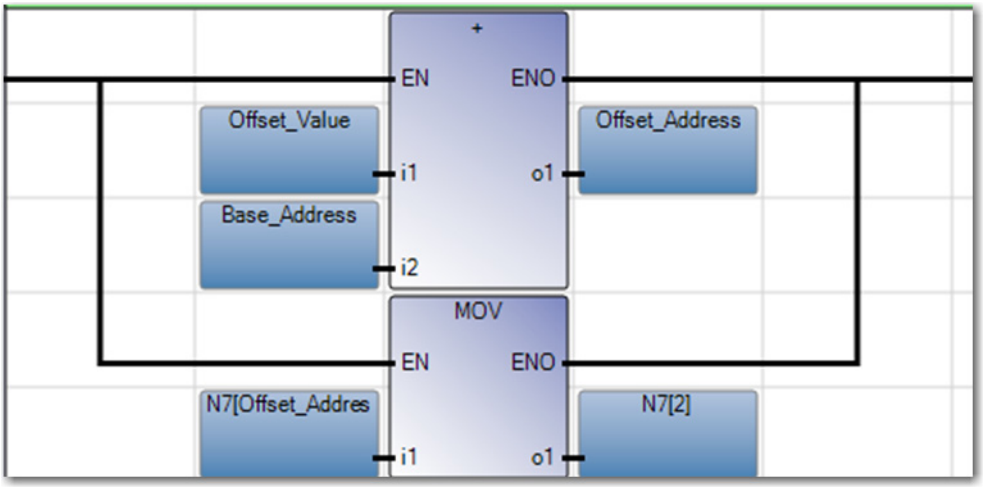
Logix Examples

Here are some program examples to show how things are done in the Connected Components Workbench software environment. For more information, see the Connected Components Workbench software help.

Program Index Addressing

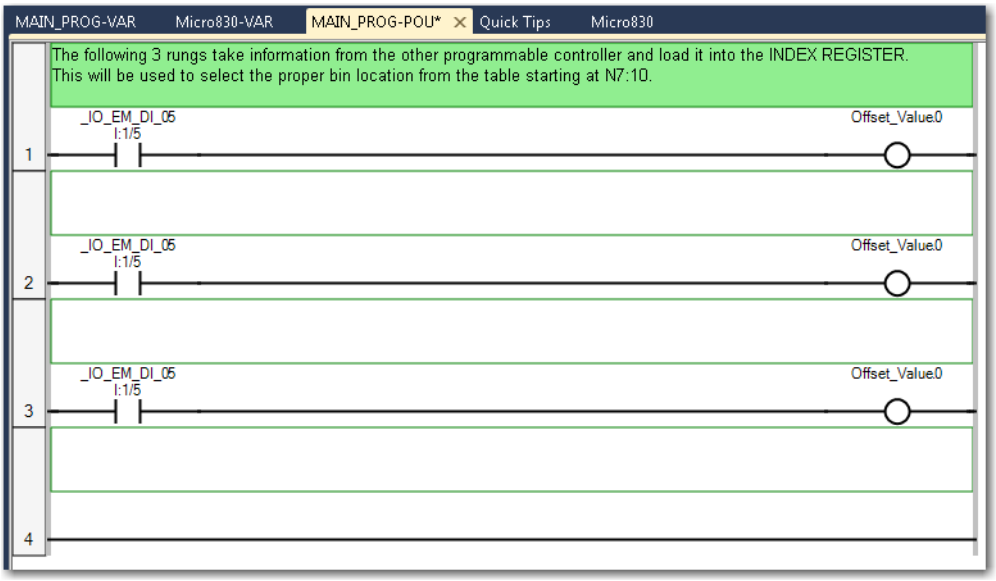
In the RSLogix 500/RSLogix Micro project, Index Register is used to select the proper bin location from the table starting at N7:10. In Connected Components Workbench software, we replace it with N7[10], which was created earlier.

The following ladder program enables indexed addressing to be used in Connected Components Workbench software.



1. Add a rung.

   On the Toolbox panel, double-click Rung. A new rung displays.

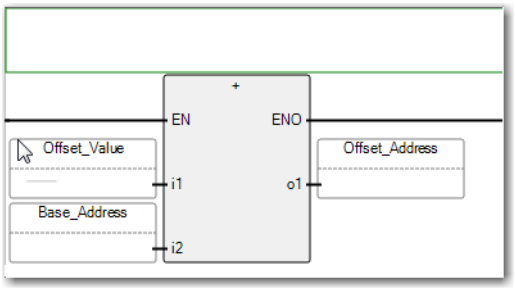


2. Add a + (Addition) instruction block to rung 4.

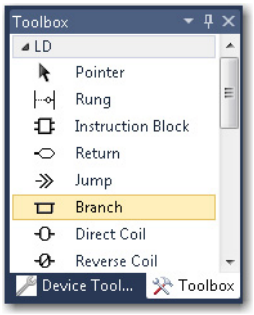
   On the Toolbox panel, drag Instruction Block into the blank rung and select MOV from the Instruction Block Selector window.

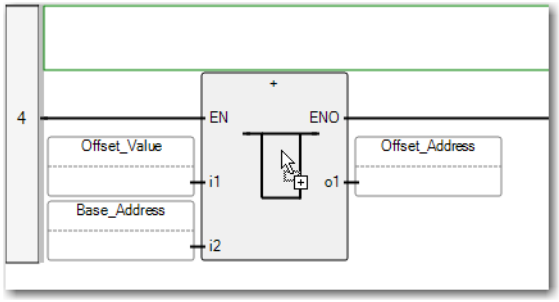
   Enter the following parameters:

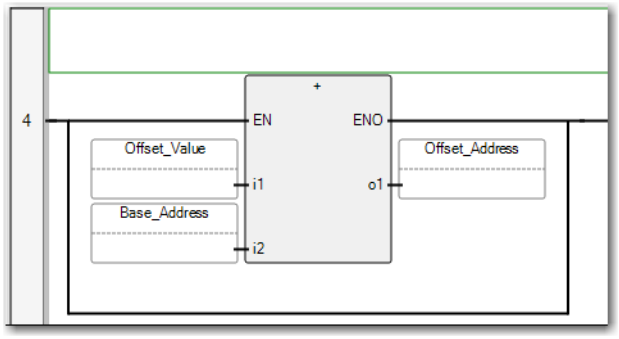
   - i1 = Output_Value
   - i2 = Base_Address
   - o1= Offset_Address



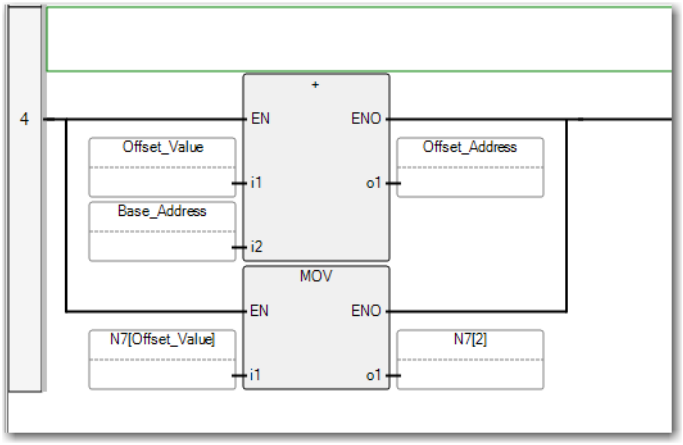
Add a branch to the + instruction block.
On the Toolbox panel, drag Branch and place it above the + (Addition) instruction block.







3. On the Branch, place a MOV instruction block with the following parameters:
   - i1 = N7[Offset_Value]
   - o1 = N7[2]



This instruction block allows the program to move the data selected based on the Offset_Address to N7[2].

Program Timer On Delay (TON) Instruction

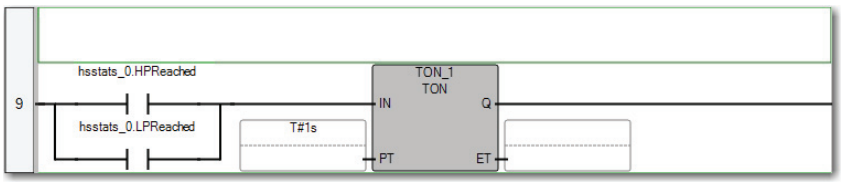
The RSLogix 500/RSLogix Micro project also uses an on-delay timer. This section shows you how to configure the TON instruction in Connected Components Workbench software.

1. Add a rung. This rung is used to add the TON instruction block.

   On the blank rung, add the TON instruction block with the following parameters:

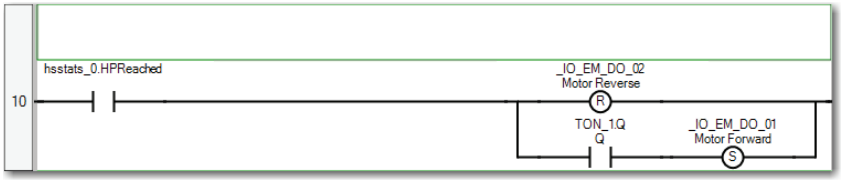
Parameter	Value (Example)	Description
PT	T#1s	Preset time. The time expression must begin with **T#** or **TIME#** prefix. The letter **'s'** must follow to indicate number of seconds. In this case, it is a 1 second on delay timer.
ET	(blank)	Elapsed time. This parameter shows the current elapsed time. Assign a variable if you want to monitor the elapsed time.

The rungs displays as follows:

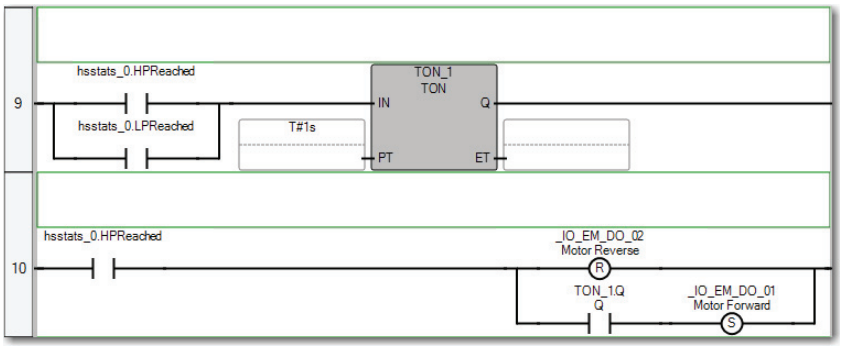


2. Add a rung. This rung is used to add the Timer Done bit.

3. On the blank rung, add the following elements and their respective variables:
   - Reset Coil = _IO_EM_DO_02
   - Set Coil = _IO_EM_DO_01
   - Direct Contact = TON_1.Q



The following graphic shows the converted rung that uses the TON instruction in Connected Components Workbench software:



Program Pulse Train Output (PTO) Motion

The PTO design for Micro800 controllers is the same as MicroLogix controllers, however the configuration, and programming is different. You can see article number 602158 on the Rockwell Automation Knowledgebase for guidance on how to setup and program PTO for your application.

Program High Speed Counter (HSC) Instruction

Programming the HSC instruction consists of two parts:

- Loading the HSC parameters
- Starting the HSC

Loading the HSC parameters

In RSLogix 500/RSLogix Micro software, the HSL instruction is required to configure the low and high presets, the output patterns, and mask bit patterns.

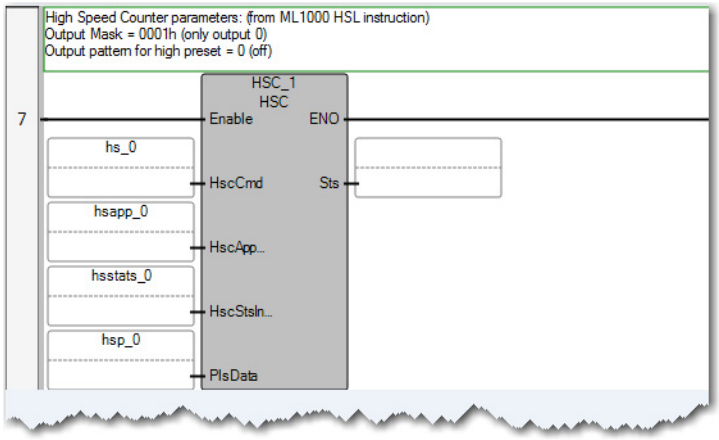
In Connected Components Workbench software, only the HSC instruction is required. It can be used to start/stop HSC counter, to refresh HSC status, to reload HSC setting, and to reset HSC accumulator.

For more information, See the chapter "Use the High-Speed Counter and Programmable Limit Switch" in the Micro830, Micro850, and Micro870 Programmable Controllers User Manual, publication 2080-UM002.

1. Add a rung. This rung is used to add the HSC instruction block.
2. On the blank rung, add the HSC instruction block with the following parameters:

Parameter	Value (Example)	Description
HscCmd	hs_0	HSC Commands
HscApp...	hsapp_0	HSC Configuration data
HscStsIn...	hsstats_0	HSC Status information
PlsData	hsp_0	Programmable Limit Switch Data
Sts	(blank)	

The following rung shows the HSC instruction that is created in Connected Components Workbench software:



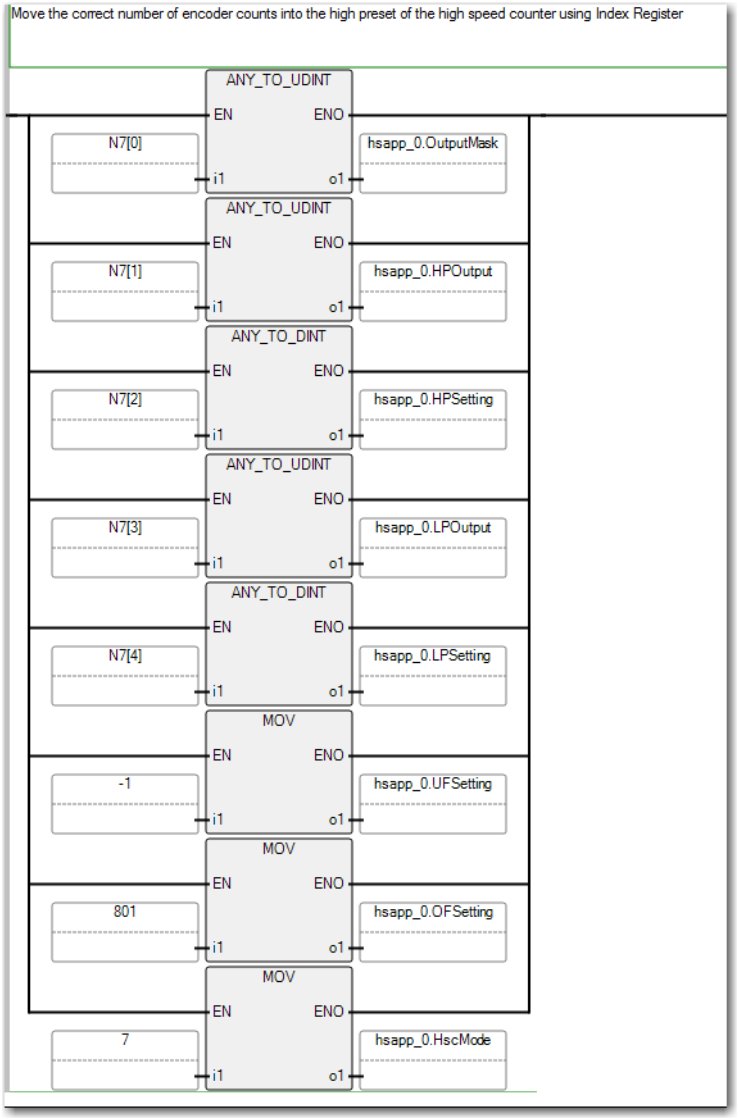
3. Add a rung. This rung is used to add the ANY_TO_UDINT and MOV instruction blocks.
4. On the blank rung, add the HSC and MOV instruction blocks with the following parameters:

Instruction	Parameter	Values	Details
ANY_TO_UDINT	i1	N7[0]	Output Mask
	o1	hsapp_0.OutputMask	
ANY_TO_UDINT	i1	N7[1]	High Preset Output
	o1	hsapp_0.HPOutput	
ANY_TO_DINT	i1	N7[2]	High Preset Setting
	o1	hsapp_0.HPSetting	
ANY_TO_UDINT	i1	N7[3]	Low Preset Output
	o1	hsapp_0.LPOutput	
ANY_TO_DINT	i1	N7[4]	Low Preset Setting
	o1	hsapp_0.LPSetting	

Instruction	Parameter	Values	Details
MOV	i1	-1	Underflow setting
	o1	hsapp_0.UFSetting	
MOV	i1	801	Overflow setting
	o1	hsapp_0.OFSetting	
MOV	i1	7	Quadrature Counter with external Reset and Hold
	o1	hsapp_0.HscMode	

Verify that proper values for the variables OFSetting, HPSetting, and UFSetting are set before triggering Start/Run for the HSC. Otherwise, the controller is faulted. It is optional to set a value for LPSetting for certain counting modes.

The following rung shows the data assignment to the required HSC parameters.



Starting the HSC

Before you enable the HSC function block, verify that HscCmd has been set to a valid value from 1...4. Otherwise, the controller is faulted.

Build and Test Your Project

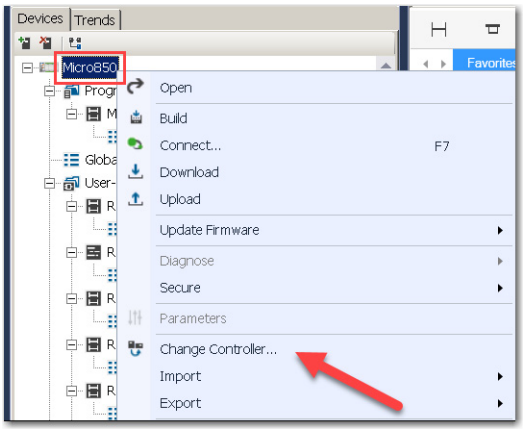
After you have created the equivalent RSLogix 500/RSLogix Micro program in Connected Components Workbench software, you must build and verify your project. Use an actual controller, or the Micro800 Simulator feature that is available in Connected Components Workbench software version 12 or later to perform the test. Depending on the method that you select, see the corresponding section for instructions on how to perform the test.

- Test Project With the Micro800 Simulator
- Test Project With a Physical Controller

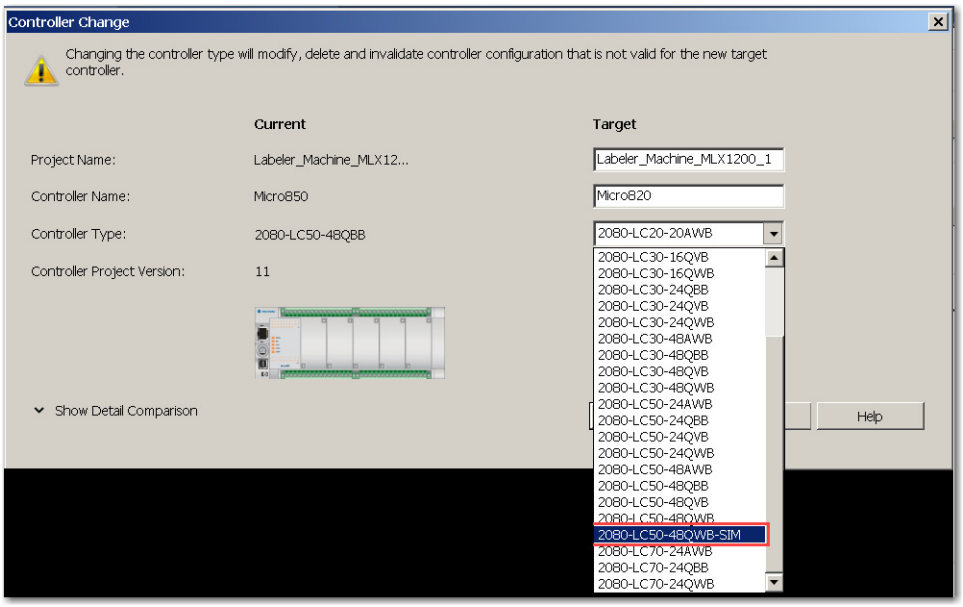
Test Project With the Micro800 Simulator

The Micro800 Simulator is a unique type of controller in the Connected Components Workbench environment. You must change the existing controller in your project to use the simulation function. Connected Components Workbench software allows you to switch to the simulator controller seamlessly to test your code in the simulation environment.

1. Change controller to the simulator controller.
   a. Right-click the controller and select 'Change Controller...'



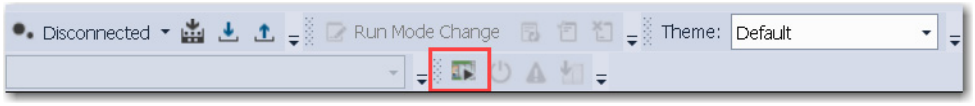
   b. Select the Micro850 simulator controller 2080-LC50-48QWB-SIM and click OK.



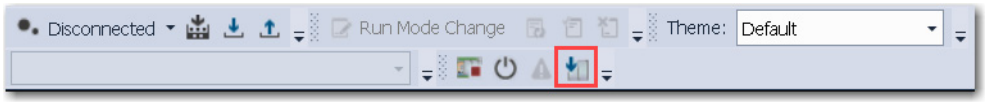
Another project is created for the simulator controller.

Only selected I/O modules (plug-in and expansion) are supported in the simulator. Unsupported modules are removed from the simulator controller. If your original project uses any unsupported I/O module (for example, 2085-OW16), you must convert them to a supported I/O module before changing your controller.

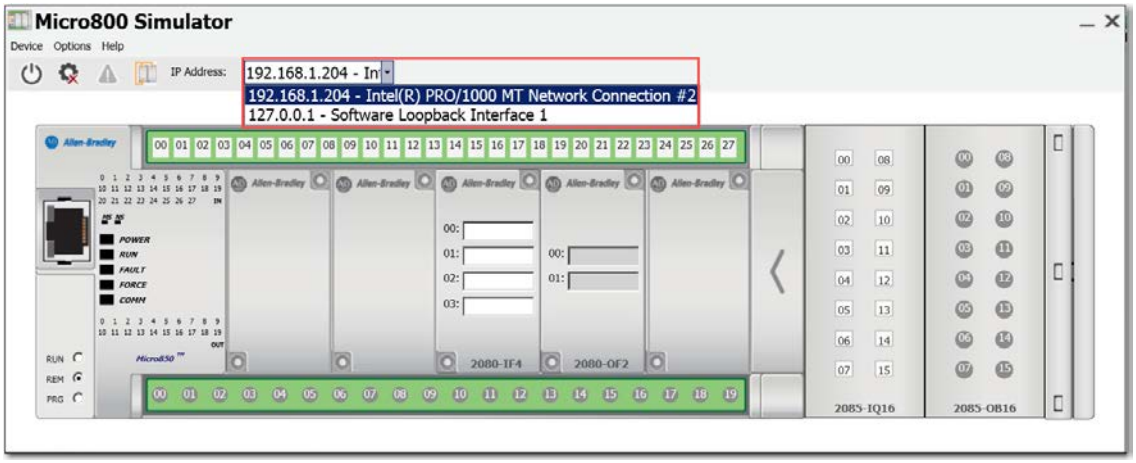
2. Start and power on the simulator.
   a. Click the 'Start Micro800 Simulator' icon from the toolbar to launch the Micro800 Simulator.



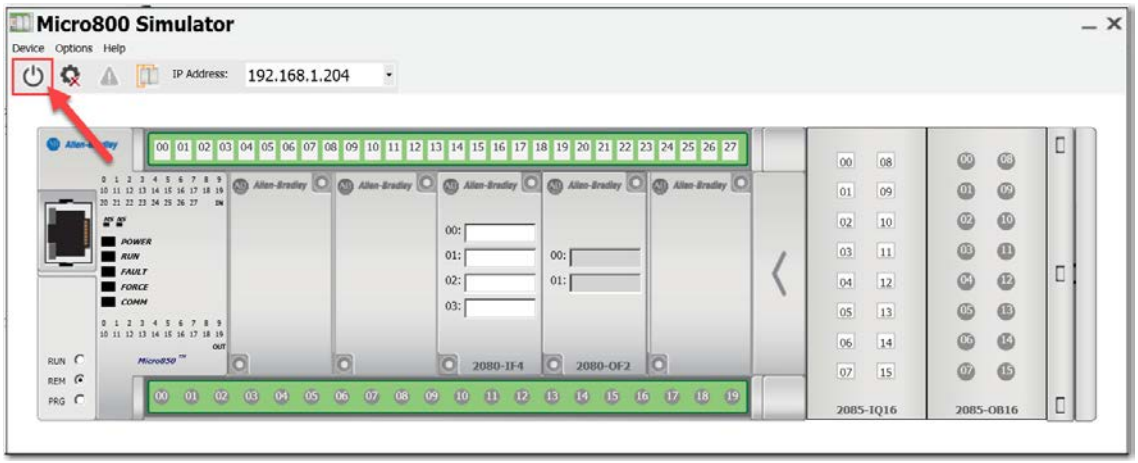
   b. Click the ‘Synchronize’ icon from the toolbar to sync your I/O configuration with the simulator.



c. Select the IP address of the simulator from the pull-down menu.

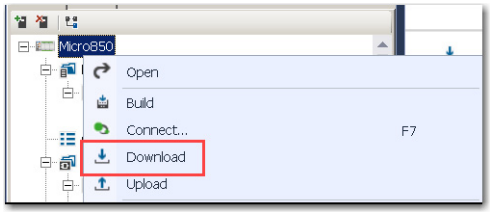


d. Click the 'Power' icon to power on your simulator.

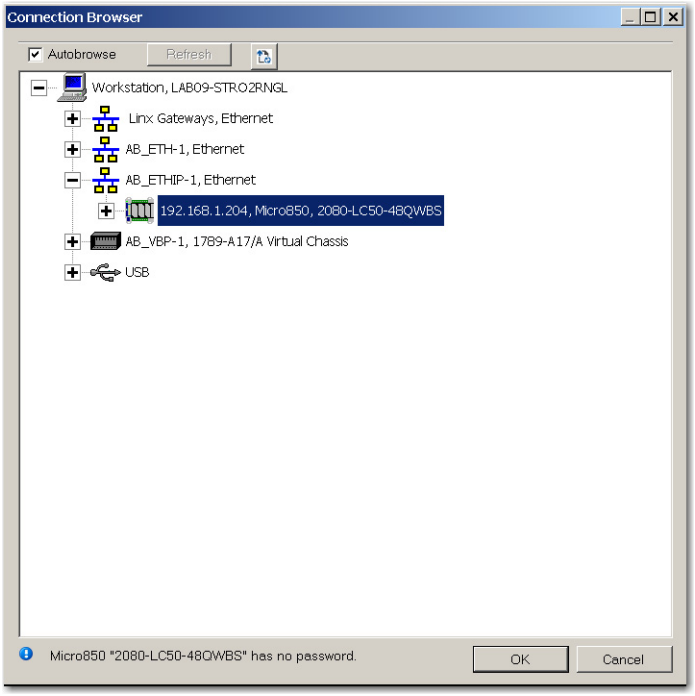


3. When the Micro800 Simulator is powered on, it appears in the FTLinx and RSLinx® software. You can download a project to the simulator as though it is a physical Micro800 controller.

   a. Right-click the controller and select Download.

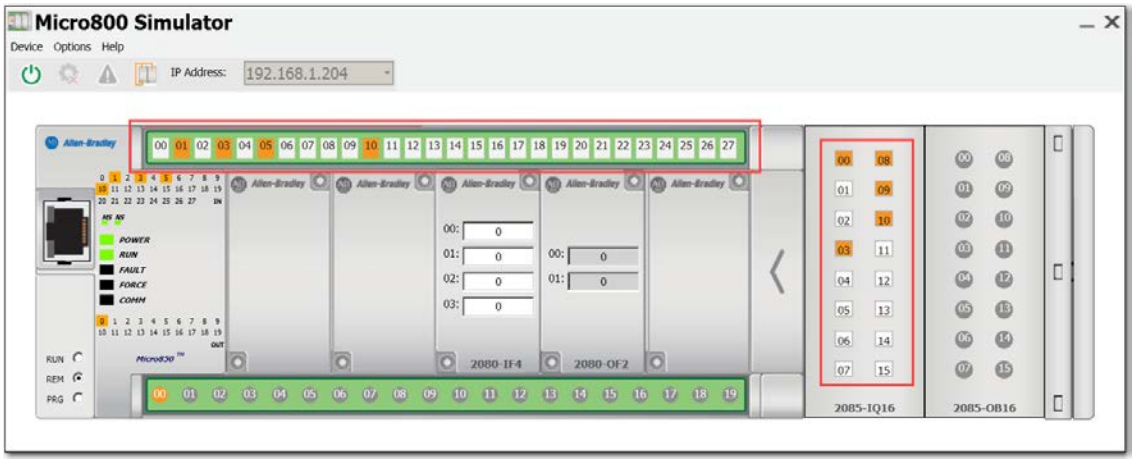


b. Select the Micro800 simulator in the RSLinx Connection Browser and click OK.

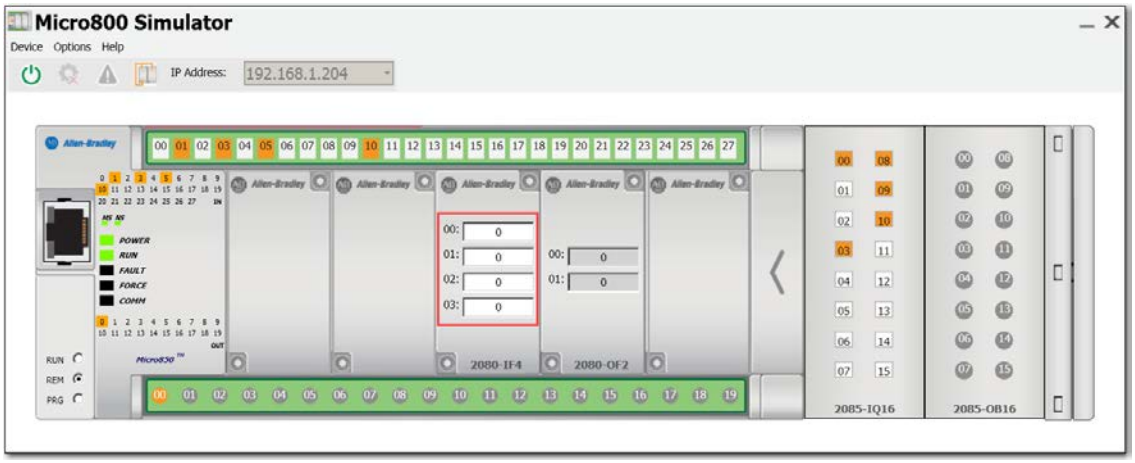


4. Test your application with Micro800 Simulator.

a. Click an input terminal to turn on the corresponding digital input.

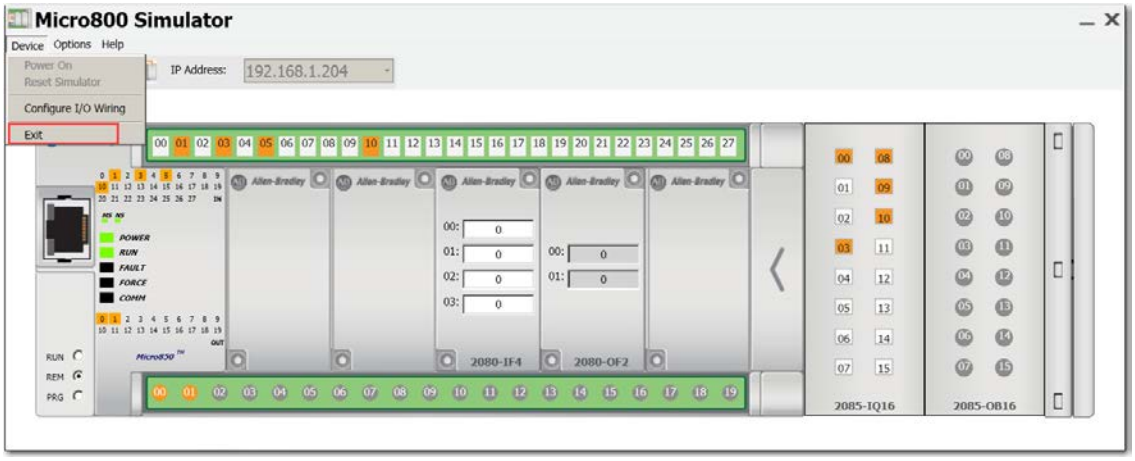


b. Enter a value in the input fields on the analog plug-in module.



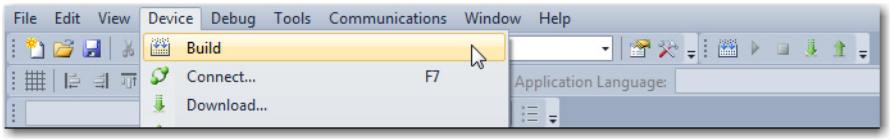
You cannot manipulate digital and analog outputs in the simulation environment because they are controlled by the program.

5. Click Device -> Exit to close the simulator.

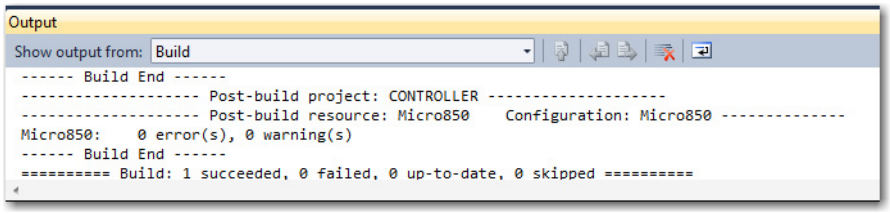


Test Project With a Physical Controller

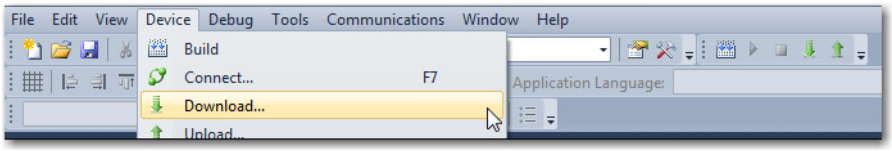
1. Build the project by selecting Device -> Build from the menu.



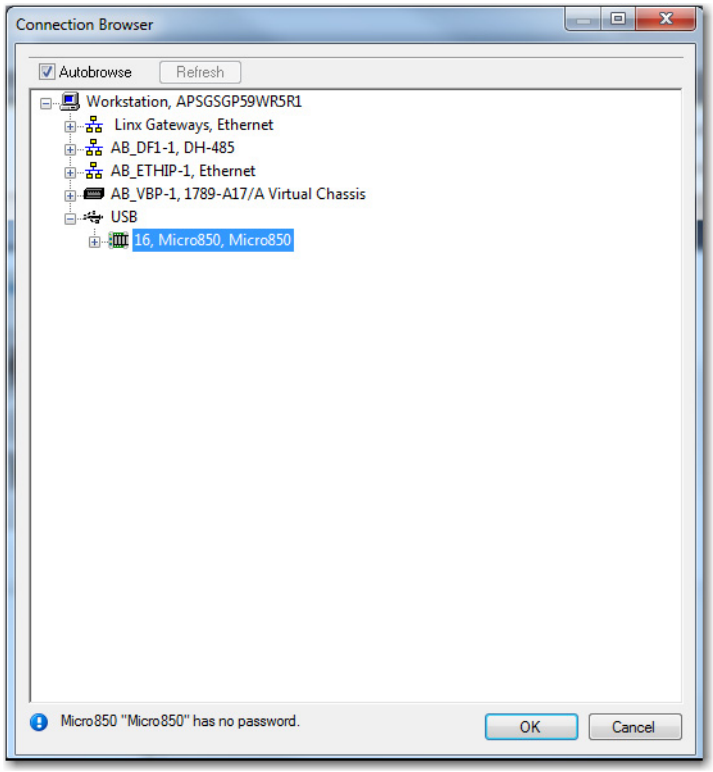
The Output window displays. The build result should show zero errors.



2. Download the project to the controller by selecting Device -> Download from the menu bar.

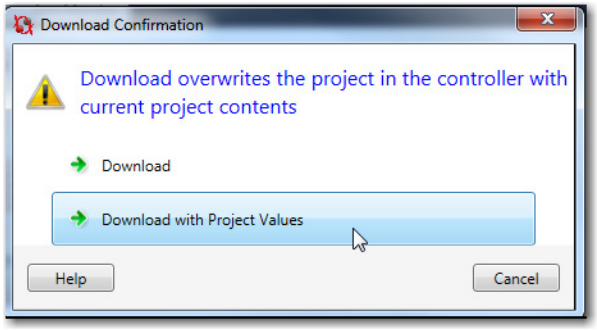


The Connection Browser window displays.



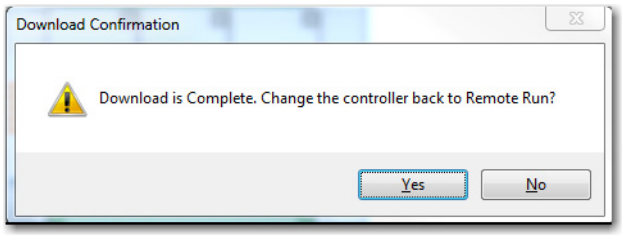
3. Select the controller and click OK.

The Download Confirmation dialog displays.



4. Select Download to download the project to the controller.

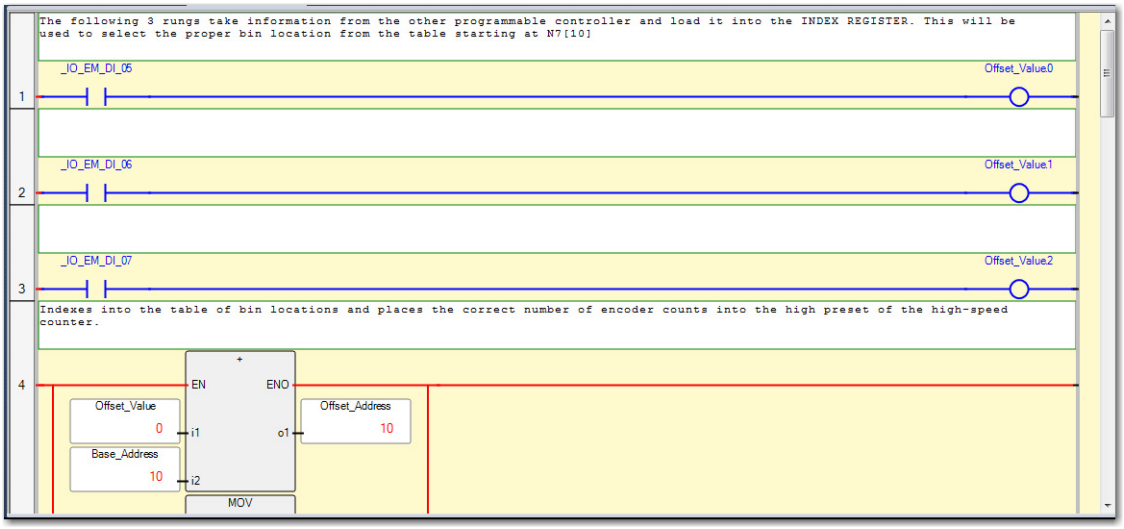
   Once the download is completed, the Download Confirmation dialog displays.



5. Click Yes to put the controller into RUN mode to test the program.
6. On the Project Organizer panel, double-click Main_PROG to show the ladder program.

   Active rungs are displayed in red and inactive rungs in blue. You can monitor the live values in the program.

   The program goes into Debug mode for Connected Components Workbench software version 8 or later.



Notes:

RLL Instruction Mapping

Overview

The purpose of this chapter is to identify the functional difference between the Relay Ladder Logic (RLL) instructions of the RSLogix 500/RSLogix Micro and Connected Components Workbench software. The RSLogix 500/RSLogix Micro software are ladder logic programming packages for the MicroLogix processors. Connected Components Workbench software is used to develop programs for Micro800 controllers.

The comparison only identifies the logic and behavioral differences and does not explain each instruction in detail. See the reference manual for more details.

Definitions, Acronyms, and Abbreviations

See the following table for definitions of terms used in this chapter.

Term	Definition
RLL	Relay Ladder Logic
CCW	Connected Components Workbench software
MSB	Most Significant Bit

Bit Shift

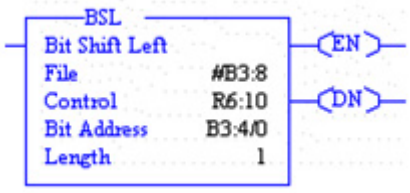
BSL

Description

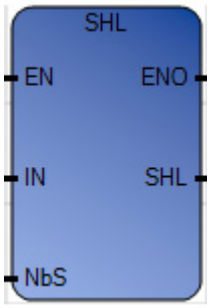
Shift bits to the left.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	EN	Level Triggered Input
F > T		When rung goes from false to true, the data provided shifts to the left by one bit provided there are no errors		When EN is TRUE, for 32-bit integers only, it performs a shift by NbS bits and places 0 in the LSB location.
T > F		On a false rung, the EN, DN, and ER bits are reset.		Do Nothing
F > F		On a false rung, the EN, DN, and ER bits are reset.		Do Nothing
T > T		Do Nothing		When EN is TRUE, for 32-bit integers only, it performs a shift by NbS bits and places 0 in the LSB location.
Input	File	Immediate value or files that contain value	IN	32-bit variable to be shifted.
Input			NbS	Number of bits to be shifted.
Input	Bit Address	Holds the bit value to be copied to the lowest bit position when the shift is made		Not supported
Input	Length	Provides the length of the bits file in the File that must be shifted. Minimum value is 16 bits. If less than that is provided, then also 16-bit shift is made.		Not supported
Output	Control	Contains the destination address of the control flags		DN (done), UL (unloaded), and ER (error) flags are NOT supported
Output	File	The output is in the same file as input	SHL	Provides the shifted output
Output	DN	Indicates that the operation is over		Not Supported
Output	EN	Rung enable output	ENO	Rung enable output

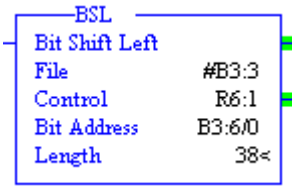
Connected Components Workbench Software Limitations

1. The SHL instruction only provides a 32-bit data right shift.
2. There is NO indication of the last unloaded bit (UL) when a left shift is made.
3. It does not indicate that the operation is overusing a DN bit.
4. Does NOT have the provision to insert a bit at the lowest bit position of the input. In order words, Bit Address feature is not present.

Behavioral Differences

1. The Connected Components Workbench SHL instruction shifts one 32-bit data field. But the RSLogix BSL instruction can shift any size of bits; (for example, 34-bit or 48-bit field sizes) by 1.

   Example: With the following ladder example, the program shifts a 34-bit data field from the B3 data file:



2. Bits continue to shift until the MSB reaches the 15th bit position in the Binary Data File
3. The shift operation clears the index register S:24 to zero.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for BSL is as follows:

Table 1 - BSL Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
File	•	•		•		•			•		•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Source	•	•		•	•	•			•																•	•	•			

(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IN								•										
NbS								•										
ENO																		
SHL	•							•										

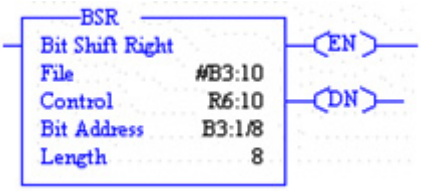
BSR

Description

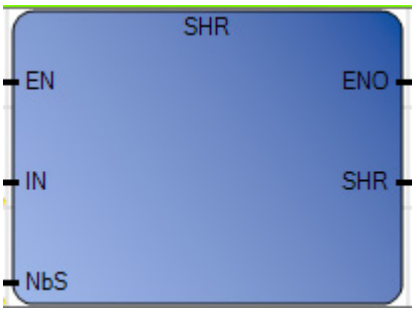
Shift bits to the right.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	EN	Level Triggered Input
F > T		When rung goes from false to true, the data provided shifts to the right by one bit provided there is no error		When EN is TRUE, for 32-bit integers only, it performs a shift by NbS bits and places 0 in the MSB location.
T > F		On a false rung, the EN, DN, and ER bits are reset.		Do Nothing
F > F		On a false rung, the EN, DN, and ER bits are reset.		Do Nothing
T > T		Do Nothing		When EN is TRUE, for 32-bit integers only, it performs a shift by NbS bits and places 0 in the MSB location.
Input	File	Immediate value or files that contain value	IN	32-bit variable to be shifted.

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input			NbS	Number of bits to be shifted.
Input	Bit Address	Holds the bit value to be copied to the highest bit position when the shift is made		Not supported
Input	Length	Provides the length of the bits file in the File that must be shifted. Minimum value is 16 bits. If less than that is provided, then also 16-bit shift is made.		Not supported
Output	Control	Contains the destination address of the control flags		DN (done), UL (unloaded), and ER (error) flags are NOT supported
Output	File	The output is in the same file as input	SHL	Provides the shifted output
Output	DN	Indicates that the operation is over		Not Supported
Output	EN	Rung enable output	ENO	Rung enable output

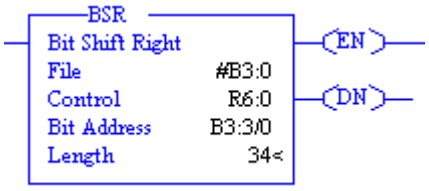
Connected Components Workbench Software Limitations

1. The SHR instruction only provides a 32-bit data right shift.
2. There is NO indication of the last unloaded bit (UL) when a right shift is made.
3. It does not indicate that the operation is overusing a DN bit.
4. Does NOT have the provision to insert a bit at the highest bit position of the input. In order words, Bit Address feature is not present.

Behavioral Differences

1. The Connected Components Workbench SHR instruction shifts one 32-bit data field. But the RSLogix BSR instruction can shift any size of bits (for example, 34-bit or 48-bit field sizes) by 1.

   Example: With the following ladder example, the program shifts a 34-bit data field from the B3 data file:



2. The number of bits in the bit array, up to 1680 bits. A length value of 0 causes the input bit to be transferred to the UL bit.
3. The shift operation clears the index register S:24 to zero.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for BSR are as follows:

Table 2 - BSR Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files											Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	L	ST	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
File	•	•		•		•		•			•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Source	•	•		•	•	•		•																	•	•	•			

(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																		
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IN								•										
NbS								•										
ENO																		
SHL	•							•										

Communication

MSG

The MSG instruction is an output instruction that allows the controller to initiate an exchange of data with other devices. The relationship with the other devices can be either peer-to-peer communication or master-to-slave communication. The type of communication that a particular application requires determine the programming configuration requirements of the MSG instruction.

Communication Considerations

See the following chart for serial port and EtherNet/IP™ communications from MicroLogix 1000 controllers to Micro820/Micro830 controllers:

RS-232 Communication (MicroLogix 1000)	**Description**	**Micro820**	**Micro830**
DF1 full-duplex (all 1761)	Point-to-point	CIP Serial (embedded serial port/ 2080-SERIALISOL plug-in)	CIP Serial (embedded serial port/ 2080-SERIALISOL plug-in)
DF1 half-duplex (1761 series D and later, 1761 analog controllers)	Single master, multiple slaves up to 255 devices Baud (300 to 38.4 kbps) slave to slave messaging	Modbus RTU (embedded serial port/ 2080-SERIALISOL plug-in)	Modbus RTU (embedded serial port/ 2080-SERIALISOL plug-in)
DH-485 half-duplex (series C and later, 1761 analog controllers), requires 1761-NET-AIC	RS-485, up to 32 devices Baud (9600/19200)	Modbus RTU (embedded serial port/ 2080-SERIALISOL plug-in)	Modbus RTU (embedded serial port/ 2080-SERIALISOL plug-in)
Ethernet Communication (MicroLogix 1000)	**Description**	**Micro820**	**Micro830**
Requires 1761-NET-ENI	—	Embedded Ethernet	No embedded Ethernet, propose Micro850

Note: In some applications where DH485/ DF1 is required and Modbus RTU does not fit, you have the choice to migrate MicroLogix 1000 to MicroLogix 1100 or MicroLogix 1200 controller.

For MicroLogix 1000 controllers, the MSG instruction is an output instruction that transfers data from one node to another on the DH-485 communication network. The instruction can be programmed as a write or read message. The target device can be another SLC 500 processor on the network, or a non-SLC 500 device, using the common interface file (485CIF file 9 in SLC 500 processors). The 485CIF protocol is also used for PLC2 type messages.

For Micro800 controllers, refer to the following table for supported communication instructions.

Function block	**Description**
MSG_CIPGENERIC	Send a CIP generic explicit message.
MSG_CIPSYMBOLIC	Send a CIP symbolic explicit message.
MSG_MODBUS	Send a Modbus message.
MSG_MODBUS2	Send a MODBUS/TCP message over an Ethernet Channel.

The following table lists the communication protocols that the Modbus and CIP™ message function blocks support.

Messaging Protocol	Communication media	Use this function block
Modbus/RTU client and server	Through a Serial port that is configured as Modbus RTU	MSG_MODBUS
Modbus/TCP client and server	Over the Ethernet instead of through a serial port	MSG_MODBUS2
Ethernet IP client and server	Through an embedded Ethernet channel	MSG_CIPSYMBOLIC MSG_CIPGENERIC
CIP Serial client and server	Ethernet cable or Serial cable	MSG_CIPSYMBOLIC
ASCII	Through an RS-232 port configured with an ASCII driver	See ASCII serial port instructions

Comparison

The MicroLogix supported data types for EQU, NEQ, GRT, LES, GEQ, and LEQ are as follows:

Table 3 - EQU, NEQ, GRT, LES, GEQ, and LEQ Instructions – Valid Addressing Modes and File Types

Parameter	Data Files												Function Files(1)										Address Mode(3)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC(2)	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	
Source B	•	•	•	•	•	•	•		•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) Only use the High-Speed Counter Accumulator (HSC.ACC) for Source A in GRT, LES, GEQ, and LEQ instructions.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

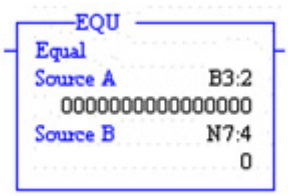
EQU

Description

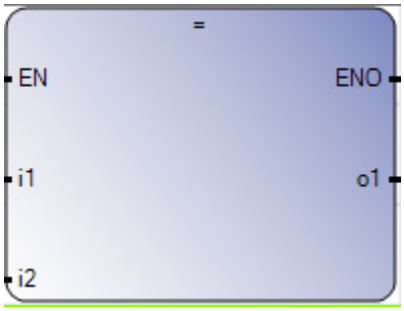
Compare two data types.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs equality comparison operation.		When EN is TRUE, then comparison is computed.
F		No operation	s	No operation
Input	Source A	Files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output		When the values are equal, the rung goes true and the output is energized (provided no other forces affect the status of the rung).	o1	TRUE if Source A and Source B are equal
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the inputs must be of the same data type.

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

See Comparison on page 106.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•(1)	•	•
i2	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•(1)	•	•
ENO	•																	
o1	•																	

(1) The Time input applies to the ST, LD, and FBD languages.

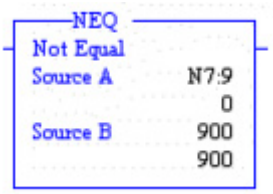
NEQ

Description

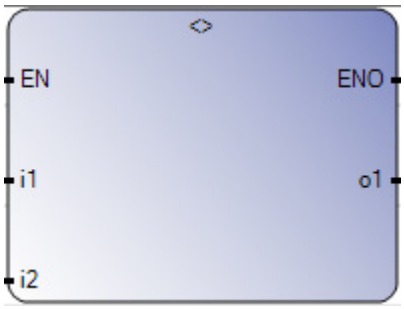
Compare two data types for inequality.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs inequality comparison operation.		When EN is TRUE, then comparison is computed. If EN is FALSE, there is no computation.
F		No operation		No operation
Input	Source A	Files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output		When the values are not equal, the rung goes true and the output is energized.	o1	TRUE if Source A and Source B are unequal
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the inputs must be of the same data type.

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

See Comparison on page 106.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•
i2	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•
ENO	•																	
o1	•																	

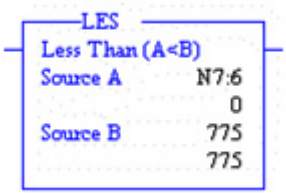
LES

Description

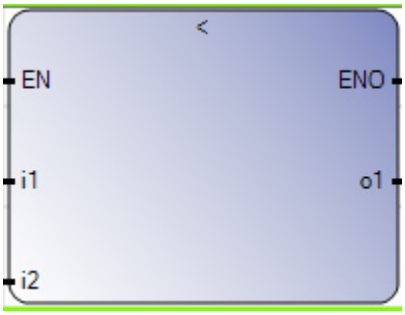
This instruction checks whether one value is less than another value.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs comparison operation.		When EN is TRUE, then comparison is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output		When the value of Source A is less than the value in Source B, the rung goes true and the output is energized (provided no other forces affect the status of the rung).	o1	TRUE if Source A is less than Source B
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the inputs must be of the same data type.

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

See Comparison on page 106.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•
i2		•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•
ENO	•																	
o1	•																	

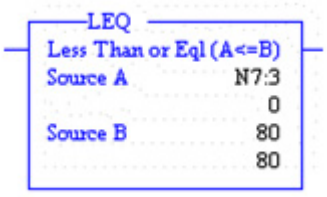
LEQ

Description

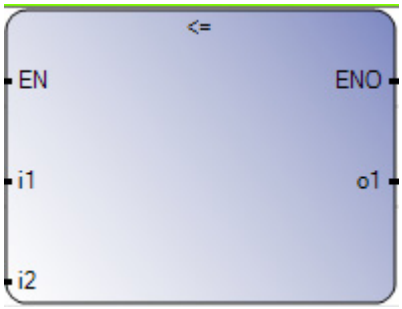
This instruction checks whether one value is less than or equal to another value.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs comparison operation.		When EN is TRUE, then comparison is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output		When the value of Source A is less than or equal to the value in Source B, the rung goes true and the output is energized (provided no other forces affect the status of the rung).	o1	TRUE if Source A is less than or equal to Source B
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the inputs must be of the same data type.

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

See Comparison on page 106.

Supported Data Types in Micro800 Controllers

Parameter	Data Types																			
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING	Immediate	Direct
EN	•																			
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•(1)	•	•		
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•(1)	•	•		
ENO	•																			
o1	•																			

(1) The Time input applies to the ST, LD, and FBD languages.

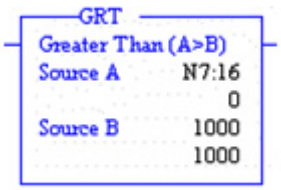
GRT

Description

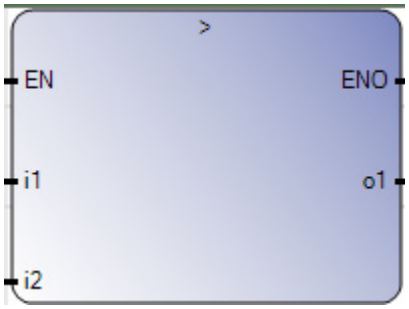
This instruction checks whether one value is greater than another value.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs comparison operation.		When EN is TRUE, then equality comparison is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output		When the value of Source A is greater than the value in Source B, the rung goes true and the output is energized (provided no other forces affect the status of the rung).	o1	TRUE if Source A is greater than Source B
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the inputs must be of the same data type.

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

See Comparison on page 106.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ENO	•																	
o1	•																	

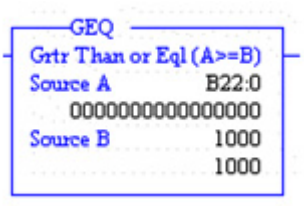
GEQ

Description

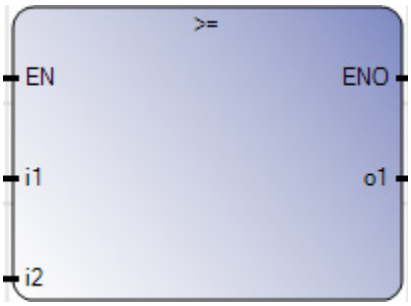
This instruction checks whether one value is greater than or equal to another value.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs comparison operation.		When EN is TRUE, then comparison is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
		When the value of Source A is greater than or equal to the value in Source B, the rung goes true and the output is energized (provided no other forces affect the status of the rung).	o1	TRUE if Source A is greater than Source B
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the inputs must be of the same data type.

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

See Comparison on page 106.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ENO	•																	
o1	•																	

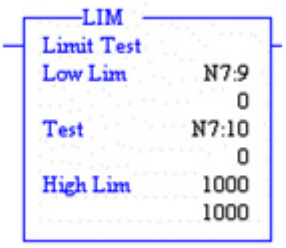
LIM

Description

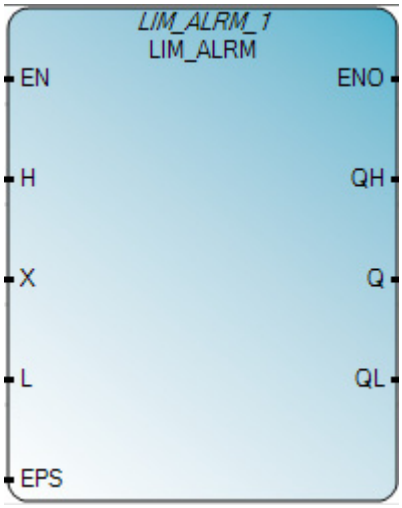
This instruction is used to compare a value with defined lower and upper limits.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this LIM instruction performs comparison operation.		When EN is TRUE, then LIM comparison is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Test	Files that contain value	X	Source variable
Input	High Limit	Immediate value or files that contain value	H	Source variable
Input	Low Limit	Immediate value or files that contain value	L	Source variable
Input			EPS	Hysteresis Value (cannot be 0)
Output		TRUE when the Test value is within or equal to limits		
Output			QH	TRUE if X is above High limit
Output			Q	TRUE if X is out of limits
Output			QL	TRUE if X is below Low limit
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Only allows Floating Point values.

Behavioral Differences

The behavioral differences between MicroLogix controllers and Micro800 controllers are as follows:

Case 1: Low Limit <= High Limit

LIM Results

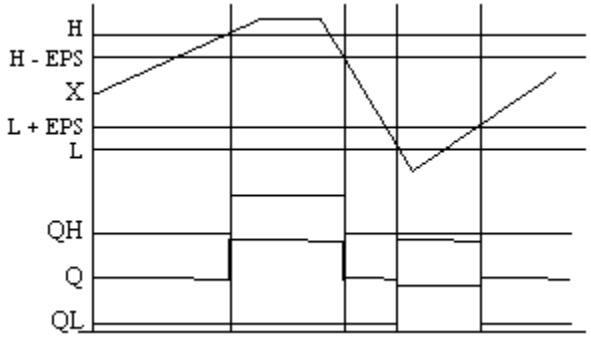
The output is TRUE when the Test value is within or equal to the Lower and Upper limit, that is

Lower Limit <= Test Value <= Upper Limit.

Otherwise the output is FALSE.

LIM_ALRM Results

The behavior of the output Q is shown according to the following graph:



Case 2: Low Limit > High Limit

LIM Results

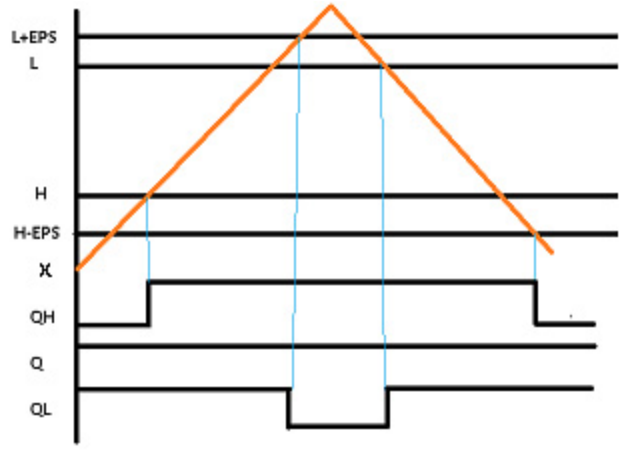
The output is FALSE when the Test value is within the Upper and Lower limit, that is

Upper Limit < Test value < Lower Limit.

Otherwise the output is TRUE.

LIM_ALRM Results

The behavior of the output Q is shown according to the following graph:



Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for LIM is as follows:

Table 4 - LIM Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]											Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Low Limit	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Test	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
High Limit	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																		
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
H														•				
X														•				
L														•				
EPS														•				
QH	•													•				
Q	•																	
QL	•																	

Control

JMP

Description

Jump to label.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters	Connected Components Workbench Parameters
Input	Level Triggered Input	Level Triggered Input
T	When the rung condition for this output instruction is true, the processor jumps forward or backward to the corresponding label instruction (LBL) and resumes program execution at the label. Multiple JMP instructions can jump to the same label. Jumping forward to a label saves program scan time by omitting a program segment until needed. Jumping backward lets the controller execute program segments repeatedly.	When the connection on the left of the Jump is TRUE, the diagram at the label is executed.
F	When the rung state is false, execution proceeds to the instruction immediately following the JMP instruction.	When the connection on the left of the Jump is FALSE, the next instruction after the jump is executed.

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

Not applicable

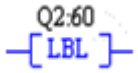
LBL

Description

Labels are used as a target for jump instructions or to control the execution of the diagram.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters	Connected Components Workbench Parameters
Input	This input instruction is the target of the JMP instruction having the same label number. It is always evaluated as true or logic 1.	Provide the label on the rung for the jump instruction to move the execution to this rung.

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

Not applicable

RET

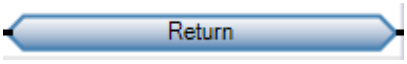
Description

This output instruction marks the end of subroutine execution or the end of the subroutine file.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters	Connected Components Workbench Parameters
Input	Level Triggered Input	Level Triggered Input
T	When the rung state is true, execution reverts to the next instruction in the program, either following the calling JSR instruction (subroutine call) or to the point where a user or user fault routine started executing.	When the left connection of the rung is in TRUE state, the diagram ends without executing the instructions that follow it.
F	When the rung state is false, execution proceeds to the instruction immediately following the RET instruction.	When the left side of the rung is FALSE, the instructions below it continue to get executed.

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

Not applicable

SUS

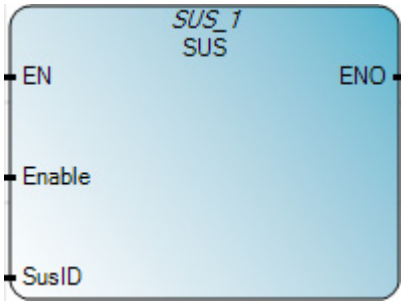
Description

This instruction places the controller in the Suspend Idle mode.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN/Enable	Level Triggered Input
T		On a true rung, this instruction causes the processor to enter the Suspend Idle mode and causes all outputs to be de-energized.		When the rung is TRUE, the controller is suspended and it remains in RUN mode.
F		No operation		No operation
Input	Suspend ID	Suspend ID value	Sus ID	Suspend ID value
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Can only support positive Suspend IDs.

Behavioral Differences

For MicroLogix controllers:

1. The range of suspend IDs are from -32768 to 32767.
2. The suspend ID is placed in word 7 (S:7) of the status file.
3. The suspend file (program or subroutine number that identifies where the executed SUS instruction resides) is placed in word 8 (S:8) of the status file.

Supported Data Types in MicroLogix Controllers

Not applicable

Supported Data Types for Micro800

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN/ENABLE	•																	
SusID						•												
ENO	•																	

TND

Description

This instruction is used to debug a program progressively, or conditionally omit the balance of your current program file or subroutines.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Inputs	EN/Enable	Level Triggered Input
T		When the rung that contains this instruction is true, it stops the controller from scanning the rest of the program file, updates the I/O, and resumes scanning at rung 0 of the main program.		Same
F		No operation		No operation
Output		If TRUE, the function is performed	TND/ENO	Same as TND

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

When using a MicroLogix controller, do not execute this instruction from the user error fault routine (file 3), high-speed counter routine (file 4), or selectable timed routine (file 5), otherwise a fault (major fault 0035) occurs.

Supported Data Types in MicroLogix Controllers

Not applicable

Supported Data Types for Micro800

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
TND	•																	
ENO	•																	

I/O Related Interrupt

IIM

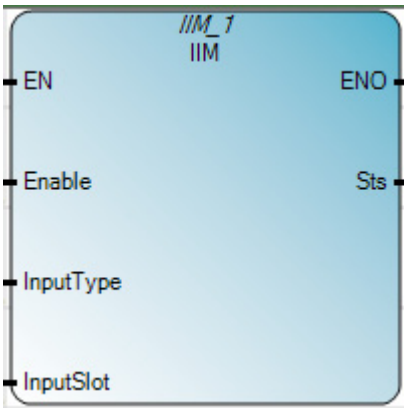
Description

This instruction is used to update the input data without waiting for the next input scan to begin.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN/Enable	Level Triggered Input
T		When the rung that contains this instruction is true, the program scan cycle is interrupted. Each word of data that starts at the specified Input has Mask applied to it. The masked data is then transferred to the input file, making the masked data available to instructions following this instruction in the ladder program. The program scan then resumes.		When the rung condition is TRUE, it updates the input data without waiting for the next input scan
F		No operation		No operation
Input	Slot	Indicates the slot or a particular word in a slot.	InputSlot	Slot number
Input	Mask	Immediate value or files that contain value		Not supported
Input			InputType	0: Embedded, 1: Plug-in input
Output			Sts	
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

1. Does not allow masking of the data bits that are read from the input slot.
2. Cannot specify individual words in a particular slot.

Behavioral Differences

For MicroLogix controllers:

1. IIM instruction cannot be used with expansion I/O slots.
2. Mask Value – Can specify with a constant or register address. The constant can be binary, decimal, or hexadecimal. For the mask, a (1) in the bit position of an input passes data from the source to the destination. A (0) inhibits data from passing from the source to the destination.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for IIM is as follows:

Table 5 - IIM Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Slot		•									•														•			•		
Mask	•	•		•	•	•																		•	•	•		•		
Length																								•						

Supported Data Types for Micro800

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN/ENABLE	•																	
InputType			•															
InputSlot			•															
Sts			•															
ENO	•																	

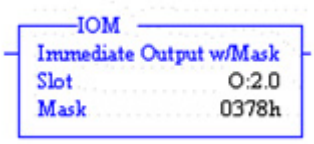
IOM

Description

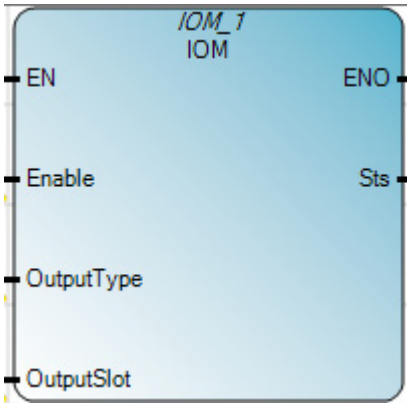
This instruction is used to update the outputs without waiting for the next output scan.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN/Enable	Level Triggered Input
T		When the rung that contains this instruction is true, the program scan cycle is ed. Each word of data from the specified output file has Mask applied to it. The masked data is then transferred to the specified Output. The program scan then resumes.		When the rung condition is TRUE, it updates the output without waiting for the next input scan
F		No operation		No operation
Input	Slot	Indicates the slot or a particular word in a slot.	Output Slot	Slot number
Input	Mask	Immediate value or files that contain value	i2	Not supported

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input			OutputType	0: Embedded, 1: Plug-in input
Output			Sts	
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

1. Does not allow masking of the data bits that are written to the output slot.
2. Cannot specify individual words in a particular slot.

Behavioral Differences

For MicroLogix controllers:

1. IOM instruction cannot be used with expansion I/O slots.
2. Mask Value - Can specify with a constant or register address. The constant can be binary, decimal, or hexadecimal. For the mask, a (1) in the bit position of an input passes data from the source to the destination. A (0) inhibits data from passing from the source to the destination.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for IOM is as follows:

Table 6 - IOM Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Slot	•										•														•			•		
Mask	•	•		•	•	•																		•	•	•		•		
Length																								•						

Supported Data Types for Micro800

Parameter	Data Type																		
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN/ENABLE	•																	
OutputType			•															
OutputSlot			•															
Sts			•															
ENO	•																	

Selectable Timed Interrupts

STD

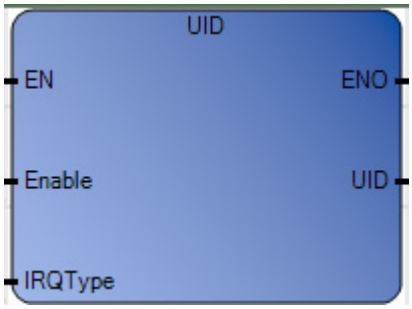
Description

Disable timer.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN/Enable	Level Triggered Input
T		When true, this instruction resets the STI enable bit and prevents the STI subroutine from executing.		Disables the
F		When the rung goes false, the STI enable bit remains reset until a true STS or STE instruction is executed. The STI timer continues to operate while the enable bit is reset.		No operation
			IRQ Type	Select the type
Output		Rung enable output	UID/ENO	Rung enable output
			IRQ Type	Select the type

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

Not applicable

Supported Data Types for Micro800

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IRQType									•									
UID	•																	

STE

Description

Enable timer.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN/Enable	Level Triggered Input
T		When true, this instruction sets the STI enable bit and allows execution of the STI subroutine.		Enable the
F		When the rung goes false, the STI enable bit remains set until a true STD instruction is executed. This instruction has no effect on the operation of the STI timer or setpoint.		No operation
Output		Rung enable output	UIE/ENO	Rung enable output
			IRQ Type	Select the type

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

Not applicable

Supported Data Types for Micro800

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IRQType									•									
UID	•																	

STS

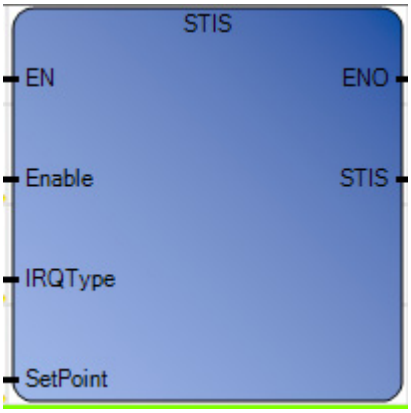
Description

Configure timer

Functional Block Diagram

RSLogix 500/RSLogix Micro

Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN/Enable	Level Triggered Input
T		Upon a true execution of the rung, this instruction enters the file number and setpoint in the status file (S:31 and S:30), and overwrites the existing data. Simultaneously, the STI timer is reset and begins timing: at timeout, the STI subroutine executes.		When the enable is TRUE, then it updates the new setpoint with the value that is provided for a particular IRQ Type.
F		No operation		No operation
			IRQ Type	Supports up to 4 timer s
Input	File	STI File Number		
Input	Time	Specify time period in "ms". A value of zero disables this function.	Setpoint	Specify time period in "ms". A value of zero disables this function.
Output		Rung enable output	STIS/ENO	Rung enable output

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

The file number and setpoint is entered into the status file (S:31 and S:30).

Supported Data Types in MicroLogix Controllers

Not applicable

Supported Data Types for Micro800

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN/ENABLE	•																	
IRQType									•									
SetPoint						•												
STIS	•																	

File Manipulation

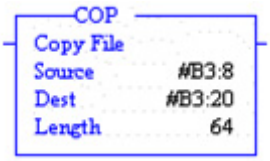
COP

Description

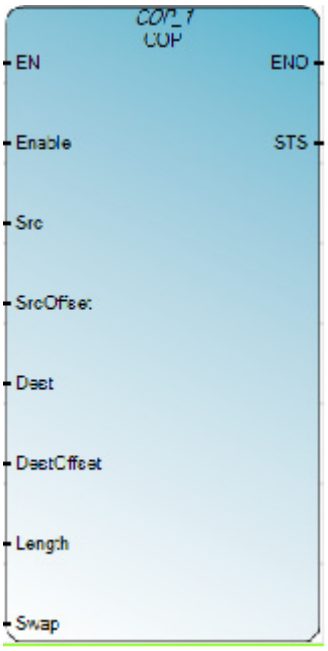
Copy a block of data from source to destination.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		When the rung condition is true, the instruction gets executed.	EN/Enable	When the rung condition is TRUE, then the instruction is executed.
Input	Source	File address from where the data has to be copied	Src	Source array data
Input		Not supported	SrcOffset	Source offset value
Input	Destination	File address to which the data has to be copied	Dest	Destination array data
Input		Not supported	DestOffset	Destination offset value
Input	Length	Number of words to be copied from source to destination	Length	Number of elements to copy.
Input			Swap	Swap the bytes
Output			STS	Status Values
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

If the source or destination is a String data type, the other party must also be a String data type, or a USINT (UCHAR and BYTE) data type. If it is not, a data type mismatch is reported.

Behavioral Differences

For MicroLogix controllers:

The maximum length that can be copied depending on data type is as follows:

Table 7 - Maximum Lengths for the COP Instruction

Source/Destination Data Type	Range of Length Operand
1 word elements (for example, word)	1…128
2 word elements (for example, long word)	1…64
3 word elements (for example, counter)	1…42
42 word elements (for example, string)	1…3

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for COP is as follows:

Table 8 - COP Instruction Valid Addressing Modes and File Types

Parameter	Data Files													Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•	•	•	•	•		•														•	•				•
Destination	•	•		•	•	•	•	•	•	•		•														•	•				•
Length																									•						

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN/ENABLE	•																	
Src	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SrcOffst						•												
Dest	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
DestOffset						•												
Length						•												
Swap	•																	
STS						•												
ENO	•																	

Math

The MicroLogix supported data types for ADD, SUB, MUL, DIV, NEG, and CLR are as follows:

Table 9 - Math Instructions (ADD, SUB, MUL, DIV, NEG, CLR) Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]										Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Source B	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•			•					•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

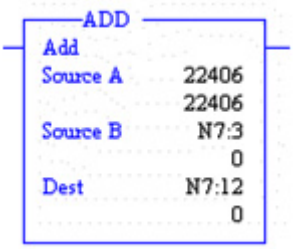
ADD

Description

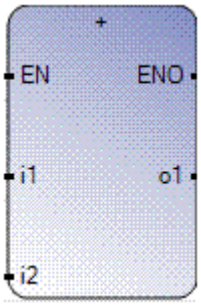
Add two values.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction adds Source A to Source B and stores the result at the destination address. If the rung is false, then there is no computation.		When EN is TRUE, then addition is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output	Dest	Contains the destination address	o1	Destination variable
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Only supports same data types for both input and output.

Behavioral Differences

For MicroLogix controllers:

1. If the destination bit receives a value less than -32,768 or greater than +32,767 (a number that requires more than 16 bits to represent), the processor sets S:0/1 (overflow bit) and S:5/0 (overflow trap bit, major error 0020). Bit S:5/0 in the program can be monitored.

If a MicroLogix processor (capable of 32-bit addition) is used, the math overflow bit (S:2/14) in the status file should be set. This causes the unsigned, truncated, least significant 16 bits to remain in the destination.

If this bit is not set and an underflow or overflow conditions occurs, the destination address contains a 32767 (if the result is positive) or -32768 (if the result is negative).

2. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Sets if carry is generated; otherwise resets
S:0/1	Overflow (V)	Sets if overflow is detected at destination; otherwise resets. On overflow, the minor error flag is also set. The value -32,768 or 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	Sets if result is zero, otherwise resets
S:0/3	Sign (S)	Sets if result is negative, otherwise resets

Supported Data Types in MicroLogix Controllers

See Math on page 133.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•
o1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•
ENO	•																	

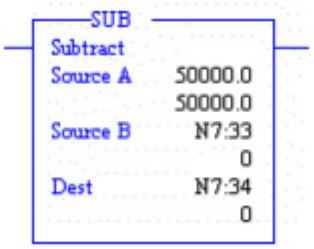
SUB

Description

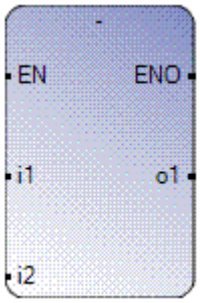
Subtract two values.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction subtracts Source A to Source B and stores the result at the destination address. If the rung is false, then there is no computation.		When EN is TRUE, then subtraction is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output	Dest	Contains the destination address	o1	Destination variable
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Only supports same data types for both input and output.

Behavioral Differences

For MicroLogix controllers:

1. If the destination bit receives a value less than -32,768 or greater than +32,767 (a number that requires more than 16 bits to represent), the processor sets S:0/1 (overflow bit) and S:5/0 (overflow trap bit, major error 0020). Bit S:5/0 in the program can be monitored.

If a MicroLogix processor (capable of 32-bit addition) is used, the math overflow bit (S:2/14) in the status file should be set. This causes the unsigned, truncated, least significant 16 bits to remain in the destination.

If this bit is not set and an underflow or overflow conditions occurs, the destination address contains a 32767 (if the result is positive) or -32768 (if the result is negative).

2. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Sets if borrow is generated; otherwise resets
S:0/1	Overflow (V)	Sets if underflow; otherwise reset. On underflow, the minor error flag is also set. The value -32,768 or 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	Sets if result is zero, otherwise resets
S:0/3	Sign (S)	Sets if result is negative, otherwise resets

Supported Data Types in MicroLogix Controllers

See Math on page 133.

For this instruction, the source has to be a word (16 bits), so all relevant data types with word sizes are supported.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		
o1		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		
ENO	•																	

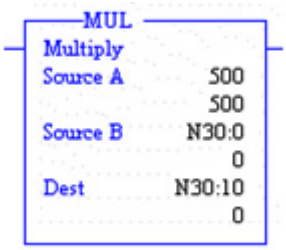
MUL

Description

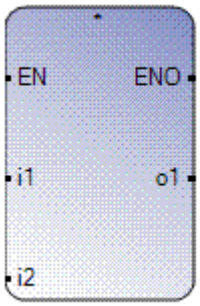
Perform multiplication on two numbers.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this instruction multiplies Source A by Source B and stores the result in the destination.		When EN is TRUE, then multiplication between the two inputs is computed. If EN is FALSE, there is no computation.
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output	Dest	Contains the destination address	o1	Destination variable

Connected Components Workbench Software Limitations

Only supports same data types for both input and output.

Behavioral Differences

For MicroLogix controllers:

1. If a value greater than +32,767 is returned, a minor error flag is set, and the value 32,767 is placed in the destination. For MicroLogix processor if the S:2/14 (math overflow selection bit) set, then the unsigned, truncated, least significant 16 bits of the overflow value remains in the destination. The math register contains the 32-bit signed integer result of the multiply operation. This result is valid at overflow.

2. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets
S:0/1	Overflow (V)	Sets if overflow is detected at destination; otherwise resets. On overflow, the minor error flag is also set. The value -32,768 or 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	Sets if result is zero, otherwise resets
S:0/3	Sign (S)	Sets if result is negative, otherwise resets

Supported Data Types in MicroLogix Controllers

See Math on page 133.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
o1		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
ENO	•																	

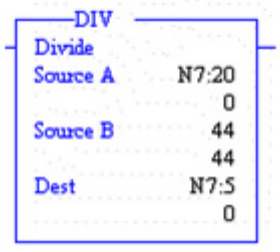
DIV

Description

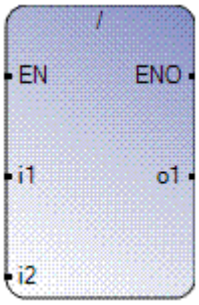
Perform division on two numbers.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this instruction divides Source A by Source B and stores the result in the destination and the math register		When EN is TRUE, then division between the two inputs is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable
Input	Source B	Immediate value or files that contain value	i2	Source variable
Output	Dest	Contains the destination address	o1	Destination variable

Connected Components Workbench Software Limitations

1. Only supports same data types for both input and output.
2. The value that is stored in the destination is NOT rounded.

Behavioral Differences

For MicroLogix controllers:

1. If a value greater than +32,767 is returned, a minor error flag is set, and the value 32,767 is placed in the destination. However, if you are using a Series C or later MicroLogix processor and have S:2/14 (math overflow selection bit) set, then the unsigned, truncated least significant 16 bits of the overflow remains in the destination.
2. If the remainder is 0.5 or greater, the destination is rounded up. The unrounded quotient is placed in the most significant word of the math register; the remainder is placed in the least significant word.
3. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets
S:0/1	Overflow (V)	Sets if division by zero or overflow is detected; otherwise resets. On overflow, the minor error flag is also set. The value 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	Sets if result is zero, otherwise resets; undefined if overflow is set.
S:0/3	Sign (S)	Sets if result is negative, otherwise resets; undefined if overflow is set.

Supported Data Types in MicroLogix Controllers

See Math on page 133.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
o1		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
ENO	•																	

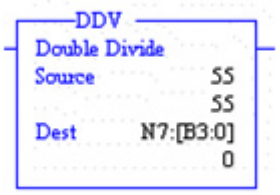
DDV

Description

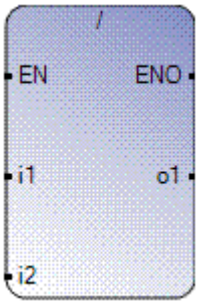
Perform division on two numbers.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this instruction divides the 32-bit math register by Source B and stores the result in the destination and the math register		When EN is TRUE, then division between the two inputs is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	32-bit math register	i1	Source variable
Input	Source (or divisor)	Immediate value or files that contain value (16 bits)	i2	Source variable
Output	Dest (or quotient)	Contains the destination address	o1	Destination variable

Connected Components Workbench Software Limitations

1. Only supports same data types for both input and output.
2. The value that is stored in the destination is NOT rounded.

Behavioral Differences

For MicroLogix controllers:

1. If the value of the quotient is greater than 32767, then the overflow flag is set and the value 32767 is placed in the destination. In normal cases, the quotient is rounded.
2. The math register initially contains the dividend of the DDV operation. Upon execution, the unrounded quotient is placed in the most significant word of the math register. The remainder is placed in the least significant word of the math register.
3. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets
S:0/1	Overflow (V)	Sets if division by zero or if result is greater than 32,767 or less than -32,768; otherwise resets. On overflow, the minor error flag is also set. The value 32,767 is placed in the destination.
S:0/2	Zero (Z)	Sets if result is zero, otherwise resets.
S:0/3	Sign (S)	Sets if result is negative, otherwise resets; undefined if overflow is set.

Supported Data Types in MicroLogix Controllers

See Math on page 133.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
i2		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
o1		•	•	•	•	•	•	•	•	•	•	•	•	•	•			
ENO	•																	

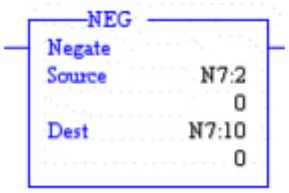
NEG

Description

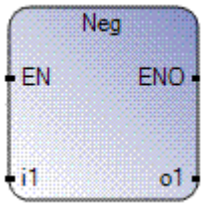
Changes the sign on the value.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, the NEG instruction changes the sign of the Source and places the result in the Destination.		When EN is TRUE, it converts the sign on the input
F		No operation		No operation
Input	Source	Files that contain value	i1	Source variable
Output	Dest	Contains the destination address	o1	Destination variable
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Both the input and the output must be of the same data type.

Behavioral Differences

For MicroLogix controllers:

This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Clears if 0 or overflow; otherwise sets.
S:0/1	Overflow (V)	Sets if overflow; otherwise resets. Overflow occurs only if -32,768 is the source. The value 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	Sets if result is zero, otherwise resets.
S:0/3	Sign (S)	Sets if result is negative, otherwise resets.

Supported Data Types in MicroLogix Controllers

See Math on page 133.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1		•			•			•			•			•	•			
o1		•			•			•			•			•	•			
ENO	•																	

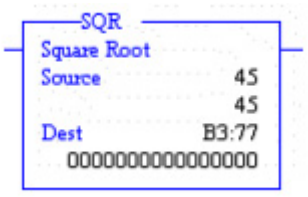
SQR

Description

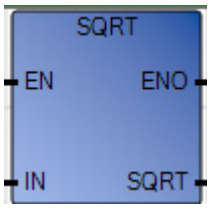
Find the square root of a number

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, the SQR instruction calculates the square root and places the rounded result in the Destination.		When EN is TRUE, it calculates the square root of the input
F		No operation		No operation
Input	Source	Files that contain value or immediate value	i1	Source variable. Must be greater than equal to zero
Output	Dest	Contains the destination address	SQRT	Destination variable. The result is zero for negative input
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

1. Does not allow negative numbers as input.
2. Both the input and the output must be of the same data type.

Behavioral Differences

For MicroLogix controllers:

1. For MicroLogix processors, constants are invalid for the source parameter.
2. This instruction calculates the square root of a negative number without overflow or faults. In applications where the source value may be negative, use a comparison instruction to evaluate the source value to determine if the destination is invalid.
3. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Sets if the source is negative; otherwise cleared.
S:0/1	Overflow (V)	Always resets.
S:0/2	Zero (Z)	Sets if destination value is zero.
S:0/3	Sign (S)	Always resets.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types SQR is as follows:

Table 10 - SQR Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	ST	F	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•		•	•		•													•	•	•		•	•	
Destination	•	•		•	•	•		•	•		•														•	•		•	•	

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IN														•				
SQRT														•				
ENO	•																	

Move and Logical

The MicroLogix supported data types for AND, OR, XOR and NOT are as follows:

Table 11 - Logical Instructions Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[2]										Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS- Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Source B[1]	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•			•	•	•		•		•	•	•			•					•	•		•	•	

(1) Source B does not apply to the NOT instruction. The NOT instruction only has one source value.

(2) PTOX and PWMX files are valid for MicroLogix 1400 BXB or BXBA unit.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
o1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ENO	•																	

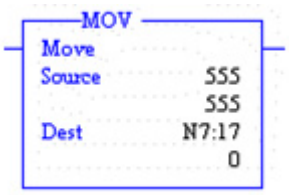
MOV

Description

Move data from source to destination.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions that precede this instruction are true, the MOV instruction moves a copy of the source to the destination each scan. The original value remains intact and unchanged in its source location.		When EN is TRUE, copy the value of i1 to o1.
F		No operation		No operation
Input	Source	Immediate value or files that contain value	i1	Source variable
Output	Dest	Contains the destination address	o1	Destination variable
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

Only supports same data types for both input and output.

Behavioral Differences

For MicroLogix controllers:

1. The bit field variable can also be moved to an integer file. For example, B3:0 can be added to N7:0 where all 16 bits in the B3:0 fields are used for moving.

2. This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets.
S:0/1	Overflow (V)	Always resets.
S:0/2	Zero (Z)	Sets if result is zero; otherwise resets.
S:0/3	Sign (S)	Sets if result is negative (most significant bit is set); otherwise resets.

Supported Data Types in MicroLogix Controllers

Table 12 - Logical Instructions Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]											Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS- Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•	•	•	•	•	•			(2)	(3)	(3)	(3)								•	•		•	•	

(1) PTOX and PWMX files are valid for MicroLogix 1400 BXB or BXBA unit.

(2) Some elements can be written to. Consult the function file for details.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
o1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ENO	•																	

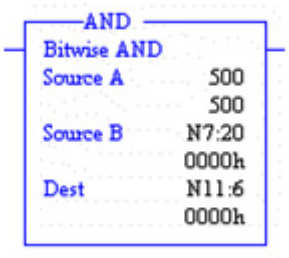
AND

Description

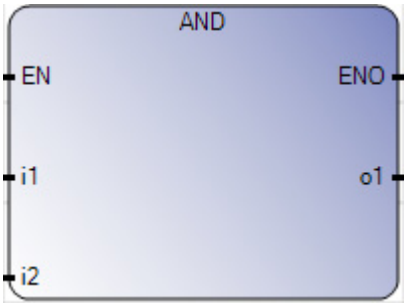
Perform logical AND operation on two values bit by bit.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs logical AND operation, bit by bit on Source A and Source B and stores the result at the destination address. If the rung is false, then there is no computation.		When rung conditions are true, this instruction is executed.
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable (BOOL only)
Input	Source B	Immediate value or files that contain value	i2	Source variable (BOOL only)
Output	Dest	Contains the destination address	o1	Destination variable (BOOL only)
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

The Connected Components Workbench AND operator only supports BOOL values.

Behavioral Differences

For MicroLogix controllers:

This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets.
S:0/1	Overflow (V)	Always resets.
S:0/2	Zero (Z)	Sets if result is zero; otherwise resets.
S:0/3	Sign (S)	Sets if most significant bit is set; otherwise resets.

Supported Data Types in MicroLogix Controllers

See Move and Logical on page 146.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•																	
o1	•																	
ENO	•																	

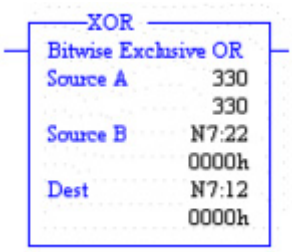
XOR

Description

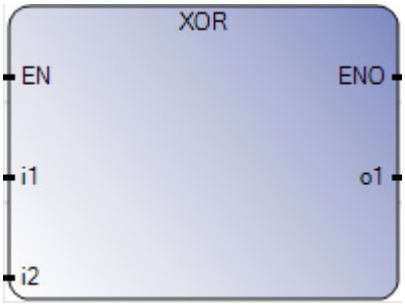
Perform logical XOR operation on two values bit by bit.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs logical XOR operation, bit by bit on Source A and Source B and stores the result at the destination address. If the rung is false, then there is no computation.		When EN is TRUE, then logical XOR between the two inputs is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable (BOOL only)
Input	Source B	Immediate value or files that contain value	i2	Source variable (BOOL only)
Output	Dest	Contains the destination address	o1	Destination variable (BOOL only)
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

The Connected Components Workbench XOR operator only supports BOOL values.

Behavioral Differences

For MicroLogix controllers:

This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets.
S:0/1	Overflow (V)	Always resets.
S:0/2	Zero (Z)	Sets if result is zero; otherwise resets.
S:0/3	Sign (S)	Sets if result is negative (most significant bit is set); otherwise resets.

Supported Data Types in MicroLogix Controllers

See Move and Logical on page 146.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•																	
i2	•																	
o1	•																	
ENO	•																	

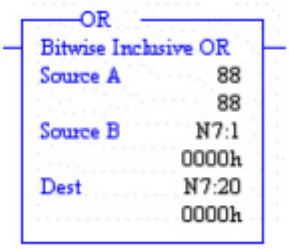
OR

Description

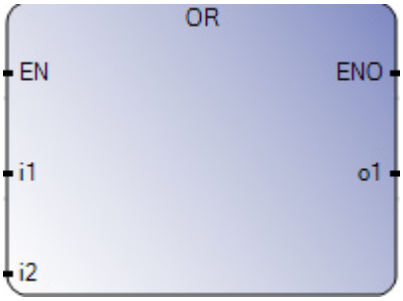
Perform logical OR operation on two values bit by bit.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs logical OR operation, bit by bit on Source A and Source B and stores the result at the destination address. If the rung is false, then there is no computation.		When EN is TRUE, then logical OR between the two inputs is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable (BOOL only)
Input	Source B	Immediate value or files that contain value	i2	Source variable (BOOL only)
Output	Dest	Contains the destination address	o1	Destination variable (BOOL only)
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

The Connected Components Workbench OR operator only supports BOOL values.

Behavioral Differences

For MicroLogix controllers:

This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets.
S:0/1	Overflow (V)	Always resets.
S:0/2	Zero (Z)	Sets if result is zero; otherwise resets.
S:0/3	Sign (S)	Sets if result is negative (most significant bit is set); otherwise resets.

Supported Data Types in MicroLogix Controllers

See Move and Logical on page 146.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•																	
i2	•																	
o1	•																	
ENO	•																	

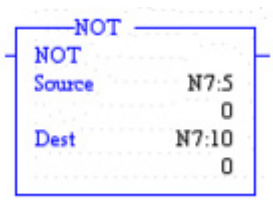
NOT

Description

Perform logical NOT operation on two values bit by bit.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	EN	Level Triggered Input
T		When rung conditions are true, this output instruction performs logical NOT operation, bit by bit on Source A and stores the result at the destination address. If the rung is false, then there is no computation.		When EN is TRUE, then logical NOT of the input is computed. If EN is FALSE, there is no computation
F		No operation		No operation
Input	Source A	Immediate value or files that contain value	i1	Source variable (BOOL only)
Output	Dest	Contains the destination address	o1	Destination variable (BOOL only)
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

The Connected Components Workbench NOT operator only supports BOOL values.

Behavioral Differences

For MicroLogix controllers:

This instruction impacts the status flags in the following way:

With this Bit:		The Controller:
S:0/0	Carry (C)	Always resets.
S:0/1	Overflow (V)	Always resets.
S:0/2	Zero (Z)	Sets if result is zero; otherwise resets.
S:0/3	Sign (S)	Sets if result is negative (most significant bit is set); otherwise resets.

Supported Data Types in MicroLogix Controllers

See Move and Logical on page 146.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
i1	•																	
o1	•																	
ENO	•																	

Relay Type

XIC

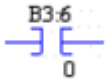
Descriptions

Examine whether the bit field is ON based on the rung state.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Examine if Closed



Connected Components Workbench

Direct Contact

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Rung state is TRUE		Rung state is TRUE
Output		If the address bit is TRUE, then outputs on the rung are energized.		If the address bit is TRUE, then outputs on the rung are energized.

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for XIC is as follows:

Table 13 - XIC and XIO Instructions Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Operand Bit	•	•	•	•	•	•			•	•			•	•	•	•	•	•	•	•	•	•	•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

XIO

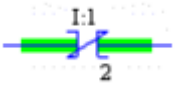
Description

Examine whether the bit field is OFF based on the rung state.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Examine if Open



Connected Components Workbench

Reverse Contact

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Rung state is TRUE		Rung state is TRUE
Output		If the address bit is TRUE, then outputs on the rung are de-energized.		If the address bit is TRUE, then outputs on the rung are de-energized.

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for XIO is as follows:

Table 14 - XIC and XIO Instructions Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Operand Bit	•	•	•	•	•	•			•	•			•	•	•	•	•	•	•	•	•	•	•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

OTE

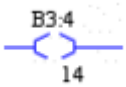
Description

Energize the bit field based on the rung state.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Output Energize



Connected Components Workbench

Direct Coil

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Rung state		Rung state
Output		This instruction is used to turn on a bit location when rung conditions are evaluated as true and off when the rung is evaluated as false.		Same

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

The OTE is reset when programmed within an inactive or false Master Control Reset (MCR) zone.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for OTE is as follows:

Table 15 - OTE Instruction Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Destination Bit	•	•	•	•	•	•			•	•			•	•	•	•	•						•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

OTL

Description

Latch (Set) the bit field.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Output Latch

Connected Components Workbench

Set Coil

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Rung state		Rung state
Output		When the rung conditions are TRUE, this bit is set or "latched" on. Once a bit has been set "on" (1 in the memory), it remains "on" even if the rung condition goes false. The bit must be reset with an OTU instruction.		When the rung conditions are TRUE, this bit is set or "latched" on. Once a bit has been set "on" (1 in the memory), it remains "on" even if the rung condition goes false. The bit must be reset with a Reset Coil instruction.

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

1. If there is a power loss, any OTL-controlled output device energizes with the return of power if the OTL bit was set when power was lost.
2. If an error condition occurs that halts processing, the physical output is turned off. But once the error condition is cleared the controller resumes operation with the OTL in the state that is determined by its data table value.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for OTL is as follows:

Table 16 - OTL and OTU Instructions Valid Addressing Modes and File Types

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Operand Bit	•	•	•	•	•	•			•	•			•	•	•	•	•						•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

OTU

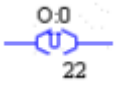
Description

Unlatch (Reset) the bit field.

Functional Block Diagram

RSLogix 500/RSLogix Micro

Output Unlatch



Connected Components Workbench

Reset Coil

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Rung state		Rung state
Output		When the rung conditions are TRUE, this bit is reset or "unlatched" on. Once a bit has been reset "off" (0 in the memory), it remains "off" even if the rung condition goes false. The bit must be set with an OTL instruction.		When the rung conditions are TRUE, this bit is reset or "unlatched" on. Once a bit has been reset "off" (0 in the memory), it remains "off" even if the rung condition goes false. The bit must be set with a Set Coil instruction.

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

1. If there is a power loss, any OTL-controlled output device energizes with the return of power if the OTL bit was set when power was lost.
2. If an error condition that halts processing occurs, the physical output is turned off. But once the error condition is cleared the controller resumes operation with the OTU in the state that is determined by its data table value.

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for OTU is as follows:

Table 17 - OTL and OTU Instructions Valid Addressing Modes and File Types.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Operand Bit	•	•	•	•	•	•			•	•			•	•	•	•	•						•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

OSR

Description

Trigger an event one at a time.

Functional Block Diagram

RSLogix 500/RSLogix Micro

One Shot Rising

Connected Components Workbench

R_TRIG

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input		Edge Triggered Input
F > T		On a false to true rung state transition, this instruction sets the Output bit and the Storage bit. Note: When the rung conditions that precede the OSR instruction go from false-to-true, the OSR instruction is true for one scan. After one scan is complete, the OSR instruction becomes false, even if the rung conditions that precede it remain true. The OSR instruction becomes true again if the rung conditions that precede it transition from false-to-true.		The output sQ is set when the input transitions from false to true. In the next scan cycle, the output is false if the clock remains true.
T > F		The Storage bit and the Output bit are reset when the rung state is false.		Output Q is false
F > F		The Storage bit and the Output bit are reset when the rung state is false.		Output Q is false
T > T		While the rung remains true, the Output bit is reset and the Storage bit remains set		Output Q is false

Connected Components Workbench Software Limitations

None

Behavioral Differences

None

Supported Data Types in MicroLogix Controllers

The MicroLogix supported data types for OSR is as follows:

Table 18 - OSR and OSF Instructions Valid Addressing Modes and File Types

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Storage Bit				•		•																			•		•			
Output Bit	•	•		•	•	•			•																•		•			

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
CLK	•																	
Q	•																	
ENO	•																	

Timer and Counter

The MicroLogix supported data types for TON, TOF, and RTO are as follows:

Table 19 - Timer Instructions Valid Addressing Modes and File Types

Parameter	Data Files[1]												Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Timer					•																			•					•
Time Base																							•						•
Preset																							•				•		
Accumulator																							•				•		

(1) Valid for Timer Files only.

The MicroLogix supported data types for CTD and CTU are as follows:

Table 20 - CTD and CTU Instructions Valid Addressing Modes and File Types

Parameter	Data Files[1]												Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Counter					•																			•					•
Preset																							•				•		
Accumulator																							•				•		

(1) Valid for Counter Files only.

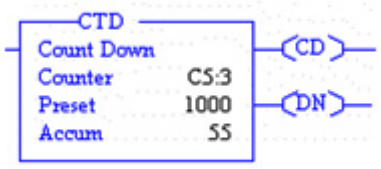
CTD

Description

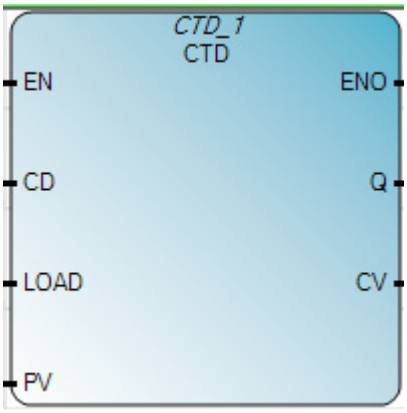
Count down from a certain value one by one.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	CD	Edge Triggered Input
F > T		This instruction counts down on false to true rung transition		Down counts when CD is a rising edges
T > F		Whenever the rung state is false, the counter shall not count and CD is reset. If either of the OV or UN bit is set, the DN bit remains in its last state. If the OV and UN bits are clear, the DN bit is reset if Accumulator < Preset, otherwise it is set.		Updates the underflow flag (Q)
F > F		Same as T > F state		Updates the underflow flag (Q)
T > T		Whenever the rung state is true, the counter shall not count and CU remains set.		Updates the underflow flag (Q)
Input	Preset	Programmed initial Value (16-bit signed integer)	PV	Programmed initial value (32 bit signed integer)
Input			LOAD	CV = PV when LOAD is TRUE.
Output	Counter CD (Enable)	This bit is set to TRUE when the rung is TRUE. It is cleared when rung is false or RES instruction is used.		Not supported
Output	Counter UV (Underflow)	When an underflow occurs from -32768 to 32767, this bit is set to TRUE. The counter continues to count down from 32767.	Q	It is set to TRUE when the count value reaches zero. The down counter freezes at zero.
Output	Counter DN (Done)	This bit is set when the Accum value is greater than or equal to the Preset value.		Not supported
Output	Accum	Count down value	CV	Count down value
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

1. Does NOT allow counting down below the value of zero.
2. Does NOT support the counter enable (EN) and counter done (DN) bit.

Behavioral Differences

1. On the low to high transition of the rung, the counter counts down by 1. The done bit (DN) is set as along as the Accum value is greater than or equal to the Preset value. It is cleared when its value is less than the Preset value.
2. The counter in MicroLogix continues to count down even after reaching zero. This behavior is NOT the same with Connected Components Workbench instruction. The underflow in MicroLogix is SET only when the Accum value crosses over from -32768 to 32767 whereas in Connected Components Workbench, the underflow bit is SET when the CV value reaches zero.

Supported Data Types in MicroLogix Controllers

See Timer and Counter on page 162.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
LOAD	•																	
PV								•										
Q	•																	
CV								•										
ENO	•																	

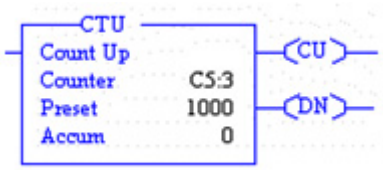
CTU

Description

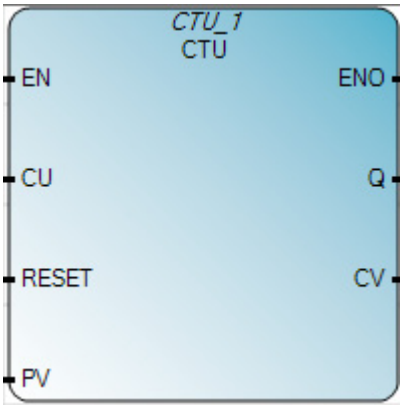
This instruction counts up from a certain value one by one.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	CU	Edge Triggered Input
F > T		This instruction counts up on false to true rung transition		It counts up when CU is a rising edge
T > F		Whenever the rung state is false, the counter shall not count and the CU bit is reset. If either of the OV or UN bits is set, the DN bit remains in its last state. If the OV and UN bits are clear, the DN bit is reset if Accumulator < Preset, otherwise it is set.		Updates the overflow flag (Q)
F > F		Same as T > F state		Updates the overflow flag (Q)
T > T		Whenever the rung state is true, the counter shall not count and CU remains set.		Updates the overflow flag (Q)
Input	Preset	Programmed initial value (16-bit signed integer)	PV	Programmed initial value (32 bit signed integer)
Input			RESET	When RESET is TRUE, it initializes the CV value to zero.
Output	Counter CU (Enable)	This bit is set to TRUE when the rung is TRUE. It is cleared when rung is false or RES instruction is used.		Not supported
Output	Counter DN (Done)	It is set to TRUE when the Accum value is greater than or equal to Preset value.	Q	It is set to TRUE when the count value is greater than or equal to the value of PV. The counter stops counting up when it reaches the value of PV.
Output	Counter OV (Overflow)	When an overflow occurs from 32767 to -32768, this bit is set to TRUE. The counter continues to up from -32768.		Not supported
Output	Accum	16-bit accumulated count. It can be reset to zero by using the RES instruction.	CV	32-bit accumulated value. It is reset by setting the RESET to TRUE.
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

None

Behavioral Differences

In Connected Components Workbench, the counter counts upwards until it reaches the limit of CV whereas in RSLogix, the counter continues to count upwards.

Supported Data Types in MicroLogix Controllers

See Timer and Counter on page 162.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
CU	•																	
RESET	•																	
PV								•										
Q	•																	
CV								•										
ENO	•																	

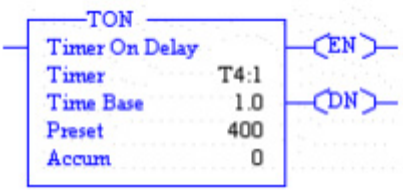
TON

Description

This instruction is used to turn an output on or off after the timer has been on for a preset time interval.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench

Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	IN	Edge Triggered Input
F > T		When the rung transitions from false to true, the error checks specified under Faults Generated are performed. If there are no errors, the timer is updated as follows. - If the DN bit is set, the EN bit is set and the TT bit is reset. Nothing else is updated. - If the DN bit is reset, the timer is initialized to begin counting Timebase intervals starting from the time the rung state transition is detected. The EN and TT bits are set.		Starts increasing the internal timer on the rising edge under the following conditions: - If the IN is TRUE and the Q flag is FALSE
T > F		Whenever the rung state is false, the Accumulator is set to zero and the EN, TT, and DN bits are reset.		Stops and resets the internal timer on the falling edge.
F > F		Same as T > F		Same as T > F
T > T		While the rung state remains true, the error checks specified under Faults Generated are performed. If there are no errors, the timer is updated as follows. - If the DN bit is set, the TT bit is reset. Nothing else is updated. - If the DN bit is reset, the Accumulator is updated according to the number of Timebase intervals that have passed since the last time the timer was updated. - If Accumulator < Preset, the TT bit is set. - If Accumulator > Preset, the DN bit is set and the TT bit is reset.		Same as F > T case.
Input	Preset	Programmed initial value (16-bit unsigned integer)	PT	Programmed initial value (Time data type)
Input	Time Base	1 ms, 10 ms, or 1 s		Supported
Output	Timer EN (Enable)	This bit is set to TRUE when the rung is TRUE. It is cleared when rung is false.		Not supported
Output	Timer DN (Done)	It is set to TRUE when the Accum value is greater than or equal to Preset value.	Q	It is set to TRUE when the count value is equal to the value of PT. The counter stops counting up when it reaches the value of PT.
Output	Timer TT (Timing)	It is set when rung conditions are TRUE and the accumulated value is less than the preset value. It is cleared when the rung state is FALSE or when done bit is set.		Not supported
Output	Accum	16-bit accumulated count. It can be reset to zero when the rung conditions are FALSE.	ET	In Time data type format
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

1. If power is lost while a TON is timing but has not reached its preset value, the EN and TT bits remain set, and the accumulated value (ACCUM) remains the same. This is also true if the processor changes from the REM Run or REM Test mode to the REM Program mode.
2. If either the Accumulator or the Preset is negative when the TON instruction is executed on a true rung, a Major Fault (0034) is generated.

Status Bits

This Bit	Is Set When	And Remains Set Until One of the Following
Timer Done Bit DN (bit 13)	Accumulated value is equal to or greater than the preset value	Rung conditions go false
Timer Enable Bit EN (bit 14)	Rung conditions are true	Rung conditions go false
Timer Timing Bit TT (bit 15)	Rung conditions are true and the accumulated value is less than the preset value	Rung conditions go false or when the done bit is set

On returning to the REM Run or REM Test mode, the following can happen:

Condition	Result
If the rung is true:	EN bit remains set. TT bit remains set. ACC value is reset.
If the rung is false:	EN bit is reset. TT bit is reset. ACC value is reset.

Supported Data Types in MicroLogix Controllers

See Timer and Counter on page 162.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IN	•																	
PT																•		
Q	•																	
ET																•		
ENO	•																	

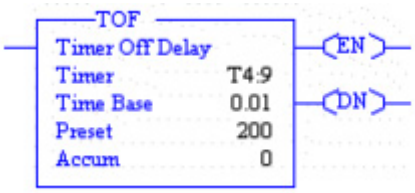
TOF

Description

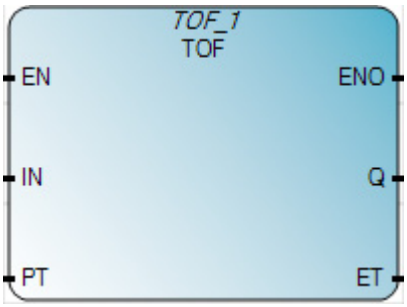
This instruction is used to turn an output on or off after its rung has been off for a preset time interval.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	IN	Edge Triggered Input
F > T		Whenever the rung state is true, the Accumulator is set to zero, the EN and DN bits are set, the TT bit is reset.		Stops and resets the internal timer on the rising edge.
T > F		When the rung transitions from true to false, the error checks specified under Faults Generated are performed. If there are no errors, the timer is updated as follows. -If the DN bit is reset, the EN and TT bits are reset. Nothing else is updated. -If the DN bit is set, the timer is initialized to begin counting Timebase intervals starting from the time the rung state transition is detected. The EN bit is reset and TT bit is set.		Starts increasing the internal timer on the falling edge under the following conditions: -If the IN is FALSE and the Q flag is TRUE
F > F		While the rung state remains false, the error checks specified under Faults Generated are performed. If there are no errors, the timer is updated as follows. -If the DN bit is reset, the TT bit is reset. Nothing else is updated. -If the DN bit is set, the Accumulator is updated according to the number of Timebase intervals that have passed since the last time the timer was updated. -If Accumulator < Preset, the TT bit is set. -If Accumulator > Preset, the DN and TT bits are reset.		Same as T > F. case.
T > T		Whenever the rung state is true, the Accumulator is set to zero, the EN and DN bits are set, the TT bit is reset.		Sets the overflow flag (Q).
Input	Preset	Programmed initial value (16-bit unsigned integer)	PT	Programmed initial value (Time data type)

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input	Time Base	1 ms, 10 ms, or 1 s		Supported
Output	Timer EN (Enable)	This bit is set to TRUE when the rung is TRUE. It is cleared when rung is false.		Not supported
Output	Timer DN (Done)	It is set to FALSE when the Accum value is greater than or equal to Preset value.	Q	It is set to FALSE when the ET value is equal to the value of PT. The internal timer stops when it reaches the value of PT.
Output	Timer TT (Timing)	It is set when rung conditions are FALSE or the accumulated value is less than the Preset value. It is cleared once the rung is TRUE or done bit is reset.		Not supported
Output	Accum	16-bit accumulated count. It can be reset to zero when the rung conditions are TRUE.	ET	In Time data type format
Output		Rung enable output	ENO	Rung enable output

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

When controller operation changes from REM Run or REM Test mode to REM Program mode or if user power is lost while a TOF is timing but has not reached its preset value, the EN, TT, and DN bits remain set, and the accumulated value (ACCUM) remains the same.

On returning to the REM Run or REM Test mode, the following can happen:

Condition	Result
If the rung is true:	TT bit is reset DN bit remains set EN bit is set ACC value is reset.
If the rung is false:	TT bit is reset DN bit is reset EN bit is reset ACC value is set equal to the preset value

Supported Data Types in MicroLogix Controllers

See Timer and Counter on page 162.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IN	•																	
PT																•		
Q	•																	
ET																•		
ENO	•																	

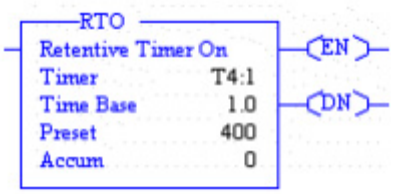
RTO

Description

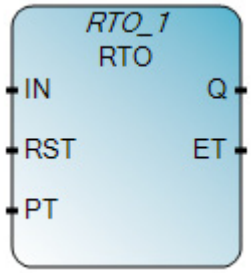
This retentive instruction lets the timer stop and start without resetting the accumulated value.

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Edge Triggered Input	IN	Edge Triggered Input
F > T		Whenever the rung state is true, the Accumulator remains the same and resumes incrementing, the EN and the TT bits are set.		If rising edge, starts increasing internal timer.
T > F		When the rung transitions from true to false, the TT and EN bits are reset. The Accumulator value and DN bit remain in its last state.		If falling edge, stops and does not reset the internal timer.
F > F		The TT and EN bits are reset. The Accumulator value and the DN bit remain in its last state.		No operation
T > T		Whenever the rung state is true, the Accumulator value remains the same and resumes incrementing. The EN and the TT bits are set.		

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input	Preset	Programmed initial value (16-bit unsigned integer)	PT	Programmed initial value (Time data type)
Input	Time Base	1 ms, 10 ms, or 1 s		Supported
			RST	If rising edge, resets the internal timer.
Output	Timer EN (Enable)	This bit is set to TRUE when the rung is TRUE. It is cleared when rung is false.		
Output	Timer DN (Done)	It is set to TRUE when the Accum value is equal to or greater than the Preset value.	Q	If TRUE, programmed time is elapsed.
Output	Timer TT (Timing)	It is set when rung conditions are TRUE or the accumulated value is less than the Preset value. It is cleared once the rung is FALSE or done bit is set.		Not supported
Output	Accum	16-bit accumulated count. It can be reset to zero when the rung conditions are TRUE.	ET	Elapsed time. Possible values range from 0 ms to 1193h2m47s294ms.

Connected Components Workbench Software Limitations

If using a Micro810 or Micro820 controller, the RTO internal timer does not persist through a power cycle by default. To persist the internal timer, set the Retained configuration parameter to true.

If using a Micro830, Micro850, or Micro870 controller, the RTO internal timer persists through a power cycle.

Behavioral Differences

For MicroLogix controllers:

When the processor changes from the REM Run or REM Test mode to the REM Program or REM Fault mode, or user power is lost while the timer is timing but not yet at the preset value, the following occurs:

- Timer Enable (EN) bit remains set.
- Timer Timing (TT) bit remains set.
- Accumulated value (ACC) remains the same.

On returning to the REM Run or REM Test mode, the following can happen:

Condition	Result
If the rung is true:	TT bit remains set EN bit remains set ACC value remains the same and resumes incrementing.
If the rung is false:	TT bit is reset DN bit remains in its last state. EN bit is reset ACC value remains in its last state.

Supported Data Types in MicroLogix Controllers

See Timer and Counter on page 162.

Supported Data Types in Micro800 Controllers

Parameter	Data Type																	
	BOOL	SINT	USINT	BYTE	INT	UINT	WORD	DINT	UDINT	DWORD	LINT	ULINT	LWORD	REAL	LREAL	TIME	DATE	STRING
EN	•																	
IN	•																	
RST	•																	
PT																•		
Q	•																	
ET																•		
ENO	•																	

High-Speed Counter

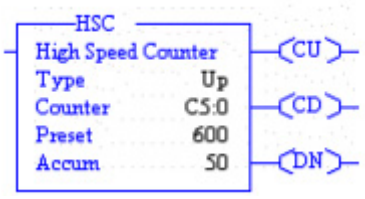
Description

The HSC instruction counts high-speed pulses from a high-speed input with a specified maximum pulse rate. An alternative set of high-speed counter instructions have been added in Connected Components Workbench software version 11 or later. This set of instructions makes them more intuitive than the standard HSC instruction. They also provide additional status information, such as pulse rate and touch probe position. To learn more about these instructions, see the Connected Components Workbench software help for the following.

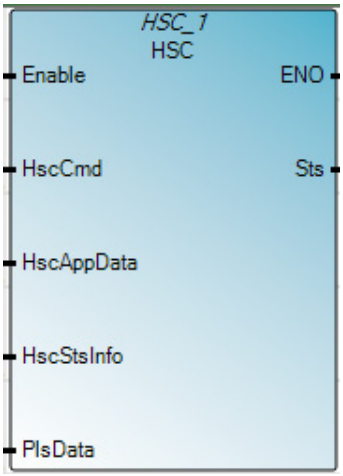
- HSCE
- HSCE_CFG
- HSCE_READ_STS
- HSCE_SET_STS

Functional Block Diagram

RSLogix 500/RSLogix Micro



Connected Components Workbench



Instruction Parameters

IO Type	RSLogix 500/RSLogix Micro Parameters		Connected Components Workbench Parameters	
Input		Level Triggered Input	Enable	Level Triggered Input
T		Whenever the rung state is true, the Accumulator is set to zero, the EN and DN bits are set, the TT bit is reset.		Function block enable. When Enable = TRUE, execute the HSC operation that is specified in the HSC command parameter.
F		The high-speed counter is disabled from counting		When Enable = FALSE, no HSC commands are issued.
Input	Type	Select from Up counters or Bidirectional counters. **Up Counters** clear the accumulator values and reload the preset values when the previous preset is reached. In **Bidirectional Counters** the accumulator and preset values are not changed by the high-speed counter when the presets are reached.	HscCmd	Issues commands to the HSC.
Input	Counter	Always C5:0	HscAppData	HSC application configuration, which is usually needed once.
Input	Preset	The accumulated value that triggers an action such as updating outputs or generating a high-speed counter.	HscStsInfo	HSC application configuration, which is usually needed once.
Input	Accum	The number of accumulated counts.	PlsData	Programmable Limit Switch (PLS) data structure.
Output	CU	Count Up Enabled	ENO	Rung enable output
Output	CD	Count Down Enabled	Sts	HSC execution status.
Output	DN	High Preset Reached		

Connected Components Workbench Software Limitations

None

Behavioral Differences

For MicroLogix controllers:

When the high-speed counter is enabled, data table counter C5:0 is used by the ladder program for monitoring the high-speed counter accumulator and status.

Counter Data File Elements (C5:0)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
Word 0	CU	CD	DN	OV	UN	UA	HP	LP	IV	IN	IH	IL	PE	LS	IE		<- Status Word
Word 1	Preset Value																
Word 2	Accumulator Value																

Counter preset and accumulated values are stored as signed integers.

For Micro800 controllers:

The high-speed counter accumulator and status are found under the **HscSTSInfo** parameters.

Parameter	Data type	HSC mode	User program access	Description
CountEnable	BOOL	0...9	read only	Counting enabled.
ErrorDetected	BOOL	0...9	read/write	Nonzero means error detected.
CountUpFlag	BOOL	0...9	read only	Count up flag.
CountDwnFlag	BOOL	2...9	read only	Count down flag.
Mode1Done	BOOL	0 or 1	read/write	HSC is Mode 1A or Mode 1B; accumulator counts up to the HP value.
OVF	BOOL	0...9	read/write	Overflow is detected.
UNF	BOOL	0...9	read/write	Underflow is detected.
CountDir	BOOL	0...9	read only	1: count up; 0: count down
HPReached	BOOL	2...9	read/write	High preset reached.
LPReached	BOOL	2...9	read only	Low preset reached.
OFCauseInter	BOOL	0...9	read/write	Overflow caused an HSC Interrupt.
UFCauseInter	BOOL	2...9	read/write	Underflow caused an HSC Interrupt.
HPCauseInter	BOOL	0...9	read/write	High preset reached, causing an HSC Interrupt.
LPCauseInter	BOOL	2...9	read/write	Low preset reached, causing an HSC Interrupt.
PlsPosition	UINT	0...9	read only	Position of the Programmable Limit Switch (PLS).
ErrorCode	UINT	0...9	read/write	Displays the error codes that are detected by the HSC subsystem.
Accumulator	DINT		read/write	Actual accumulator reading.
HP	DINT		read only	Last high preset setting.
LP	DINT		read only	Last low preset setting.
HPOutput	UDINT		read/write	Last high preset output setting.

CountEnable	BOOL	0...9	read only	Counting enabled.
LPOutput	UDINT		read/write	Last low preset output setting.

Supported Data Types in MicroLogix Controllers

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
HSC	High-Speed Counter	type	immediate		0…7, where: 0 = up 1 = up and reset/hold 2 = pulse/direction 3 = pulse/direction & reset/hold 4 = up/down 5 = up/down & reset/hold 6 = encoder 7 = encoder and reset/hold
		counter	direct		Not applicable
		preset	(contained in the counter register)	C5:0. C5:1 (element level)	-32,768-32,767
		accum	(contained in the counter register)		-32,768-32,767

Miscellaneous

Notes on Unsupported RSLogix 500/RSLogix Micro Instruction Set

See the following table for notes on unsupported RSLogix 500/RSLogix Micro instruction set.

Unsupported RSLogix 500/ RSLogix Micro Instruction Set	Description	Notes
LFL	LIFO Load	Replace with User-defined Function Block (UDFB)
LFU	LIFO Unload	
FFL	FIFO Load	Replace with User-defined Function Block (UDFB)
FFU	FIFO Unload	
MEQ	Masked Comparison for Equal	Replace with User-defined Function Block (UDFB)
JSR	Jump to subroutine	Replace with User-defined Function Block (UDFB)
SBR	Subroutine	Part of Ladder mechanism, not required to implement as an instruction
MCR	Master control reset	Currently not replicated
INT	Interrupt subroutine	See Configure Interrupts on a Micro800 Controller on page 179
FLL	Fill File	Replace with User-defined Function Block (UDFB)
CLR	Clear	Can be performed by MOV Instruction Block
TOD	Convert to BCD	Replace with User-defined Function Block (UDFB)
FRD	Convert from BCD to Integer	Replace with User-defined Function Block (UDFB)
DCD	Decode 4 to 1 of 16	Replace with User-defined Function Block (UDFB)
SCL	Scale Data	Replace with User-defined Function Block (UDFB)
MVM	Masked Move	Replace with User-defined Function Block (UDFB)

Unsupported RSLogix 500/ RSLogix Micro Instruction Set	Description	Notes
RES	Reset	Counter Instruction Block in Connected Components Workbench software has "Reset" as input parameter
HSD	HSC Interrupt Disable	The HSC can be configured from the configuration workspace of the controller.
HSE	HSC Interrupt Enable	
RAC	HSC Reset Accumulator	Available as an HSC command in the HSC Instruction Block
RES for HSC	Reset HSC Interrupt	Replace with User-defined Function Block (UDFB)
ENC	Encode 1 of 16 to 4-bit data	Replace with User-defined Function Block (UDFB)
SQC	Sequencer Compare	Currently not replicated
SQL	Sequencer Load	Currently not replicated
SQO	Sequencer Output	Currently not replicated
HSL	Configures the low and high presets, the output patterns, and mask bit patterns.	Part of the HSC Instruction Block

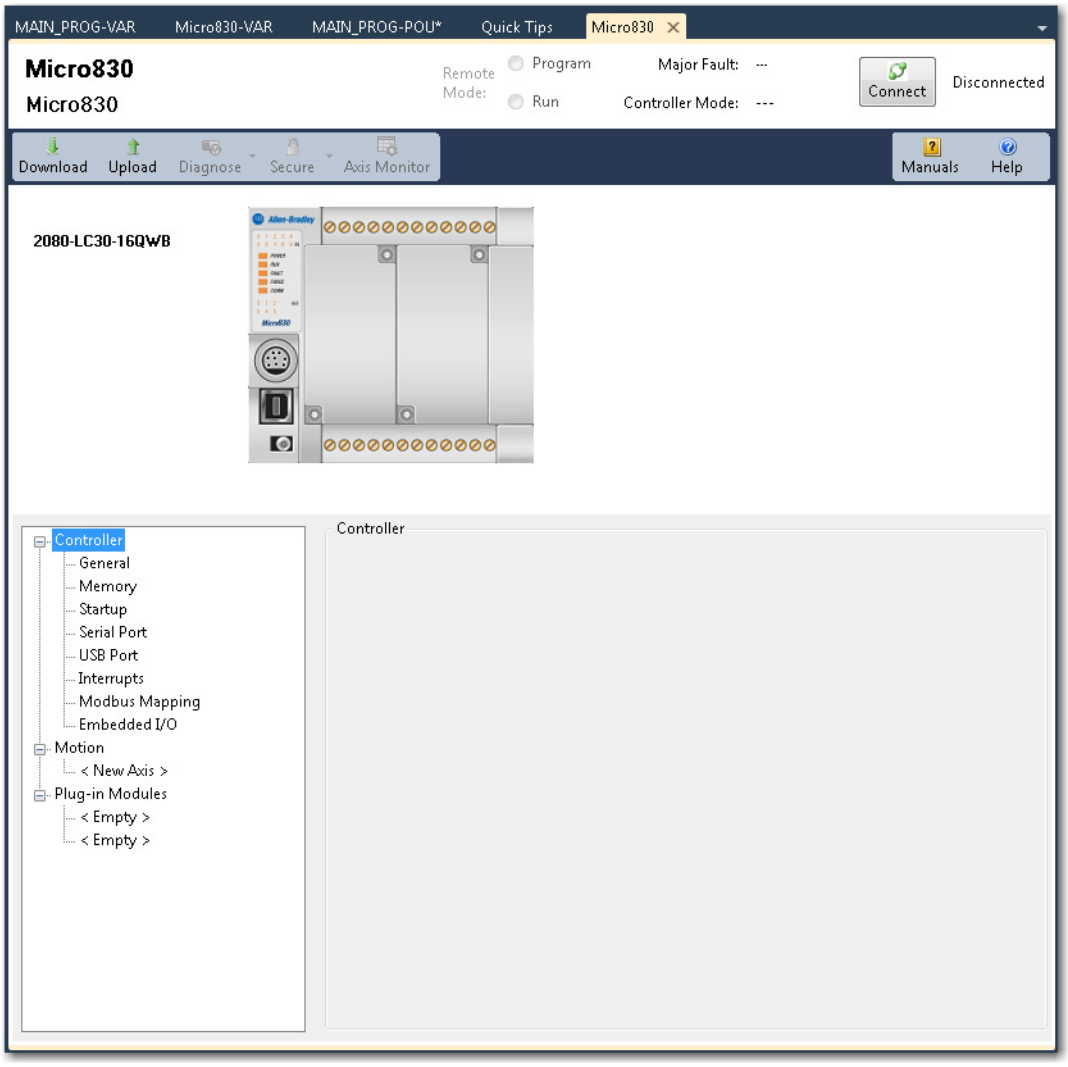
Notes:

Additional Examples

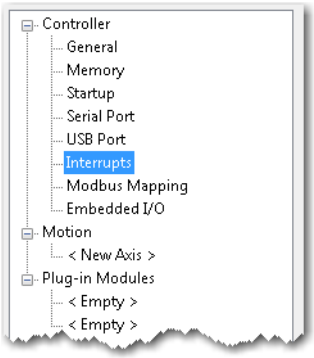
Configure Interrupts on a Micro800 Controller

For this example, use a Selectable Timed Interrupt (STI).

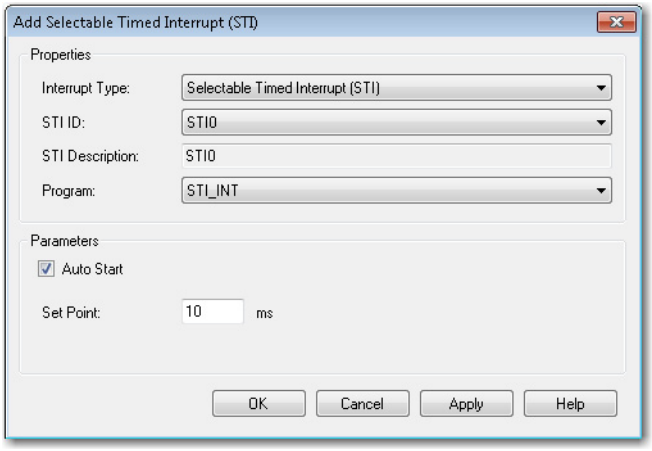
1. Create a program to execute when the interrupt occurs.
   a. On the Project Organizer panel, right-click Programs and select Add –> New LD: Ladder Diagram.
   b. Rename the program as STI_INT.
2. On the Project Organizer panel, double-click Micro830.
   The Micro830 controller tab displays.



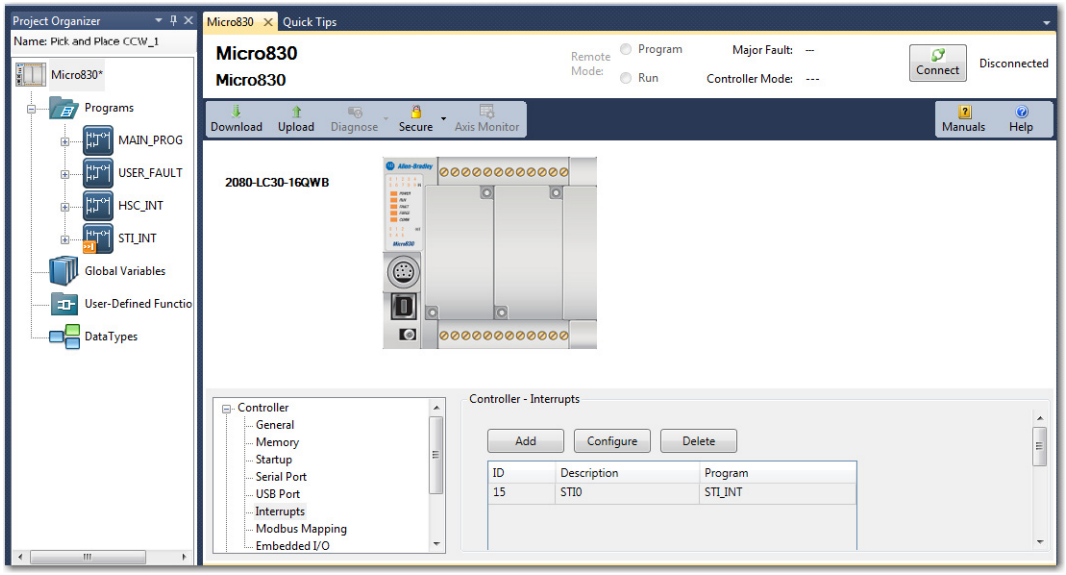
3. On the lower left of the tab, expand Controller, then click Interrupts.



4. On the Controller - Interrupts section (right), click Add.
   The Add Selectable Time Interrupt (STI) window displays.
5. Set the STI properties and parameters as follows:
   a. Interrupt Type [Selectable Timed Interrupt (STI)]
   b. STI ID (STI0)
   c. Program (the program created earlier)
   d. Auto Start (selected)
   e. Set Point (10 ms)



6. Click Apply, then click OK.
   The Micro830 workspace displays.



TIP The configured Interrupts can be configured or deleted from the Controller - Interrupts workspace.

Set Up High-Speed Counter (HSC) Instruction Variables

The controller uses Indexed Addressing to locate the correct encoder count from the data table N7[10] to N7[17] and load the information into the high preset of the high-speed counter.

The HSC instruction is required to allow the HSC parameters (N7[0] to N7[4]) to be loaded for the same instruction:

Name	Data Value	Details
N7[0]	0001h	Output Mask – Control gripper
N7[1]	0000h	Output pattern for High Preset – Turn off gripper
N7[2]	100d	High Preset – loaded from table N7[10] to N7[17].
N7[3]	0001h	Output pattern for Low Preset – Turn on gripper
N7[4]	0d	Low Preset – home position when encoder triggers Z-reset.

The number of pulses the head must travel to reach each bin location is stored in a data table that starts at address N7 [10] and ends at N7 [17]. This value is entered under the Initial Value field so that it is used as the value of a variable when a controller starts execution for the first time, such as after a program download.

Name	Data Value	Details
N7[10]	100d	Number of pulses to reach Bin Location A
N7[11]	200d	Number of pulses to reach Bin Location B
N7[12]	300d	Number of pulses to reach Bin Location C
N7[13]	400d	Number of pulses to reach Bin Location D
N7[14]	500d	Number of pulses to reach Bin Location E
N7[15]	600d	Number of pulses to reach Bin Location F
N7[16]	700d	Number of pulses to reach Bin Location G
N7[17]	800d	Number of pulses to reach Bin Location H

1. On the Project Organizer panel, double-click Global Variables.
2. Click the + symbol for the variable N7 to expand the row.
3. Double-click the Initial Value field for N7 [0], then enter "01".
4. Repeat step 3 for N7 [1] to N7[4] and N7[10] to N7[17] for the rest of the data values as shown in the earlier tables.

   The following image shows the completed entries:

Name	Alias	Data Type	Dimension	Project Value	Initial Value	Comment
N7		INT	[0..104]	...	...	
N7[0]		INT			1	
N7[1]		INT			0	
N7[2]		INT				
N7[3]		INT			1	
N7[4]		INT			0	
N7[5]		INT				
N7[6]		INT				
N7[7]		INT				
N7[8]		INT				
N7[9]		INT				
N7[10]		INT			100	
N7[11]		INT			200	
N7[12]		INT			300	
N7[13]		INT			400	
N7[14]		INT			500	
N7[15]		INT			600	
N7[16]		INT			700	
N7[17]		INT			800	

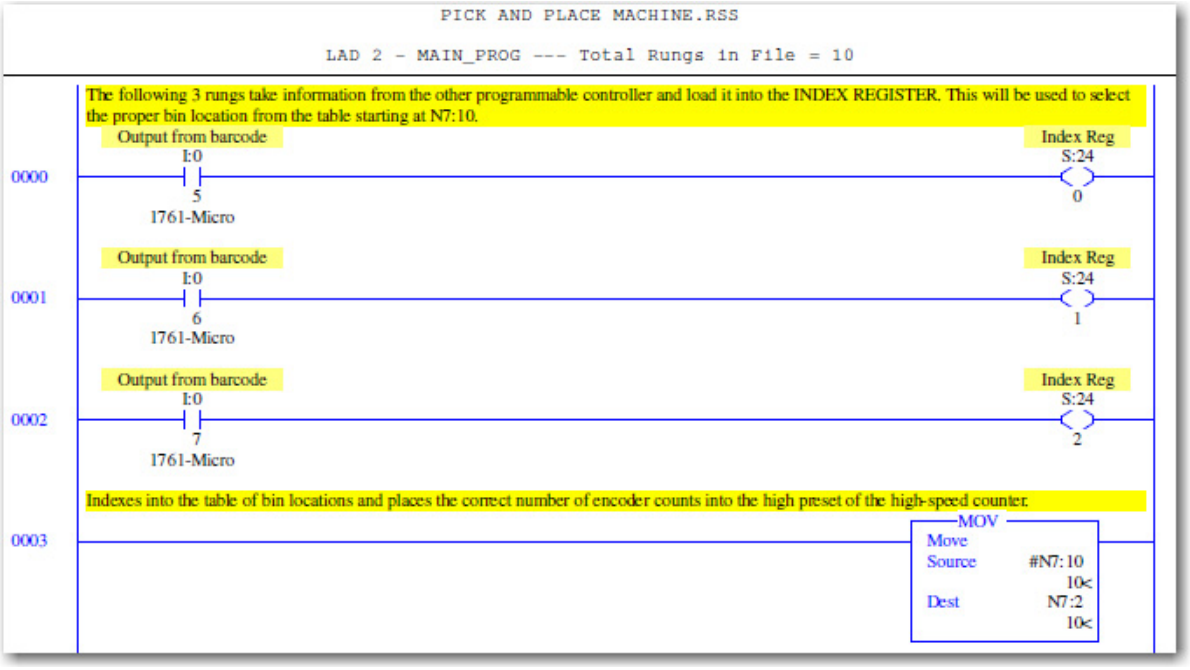
Original and Converted Pick-and-Place Ladder Diagrams

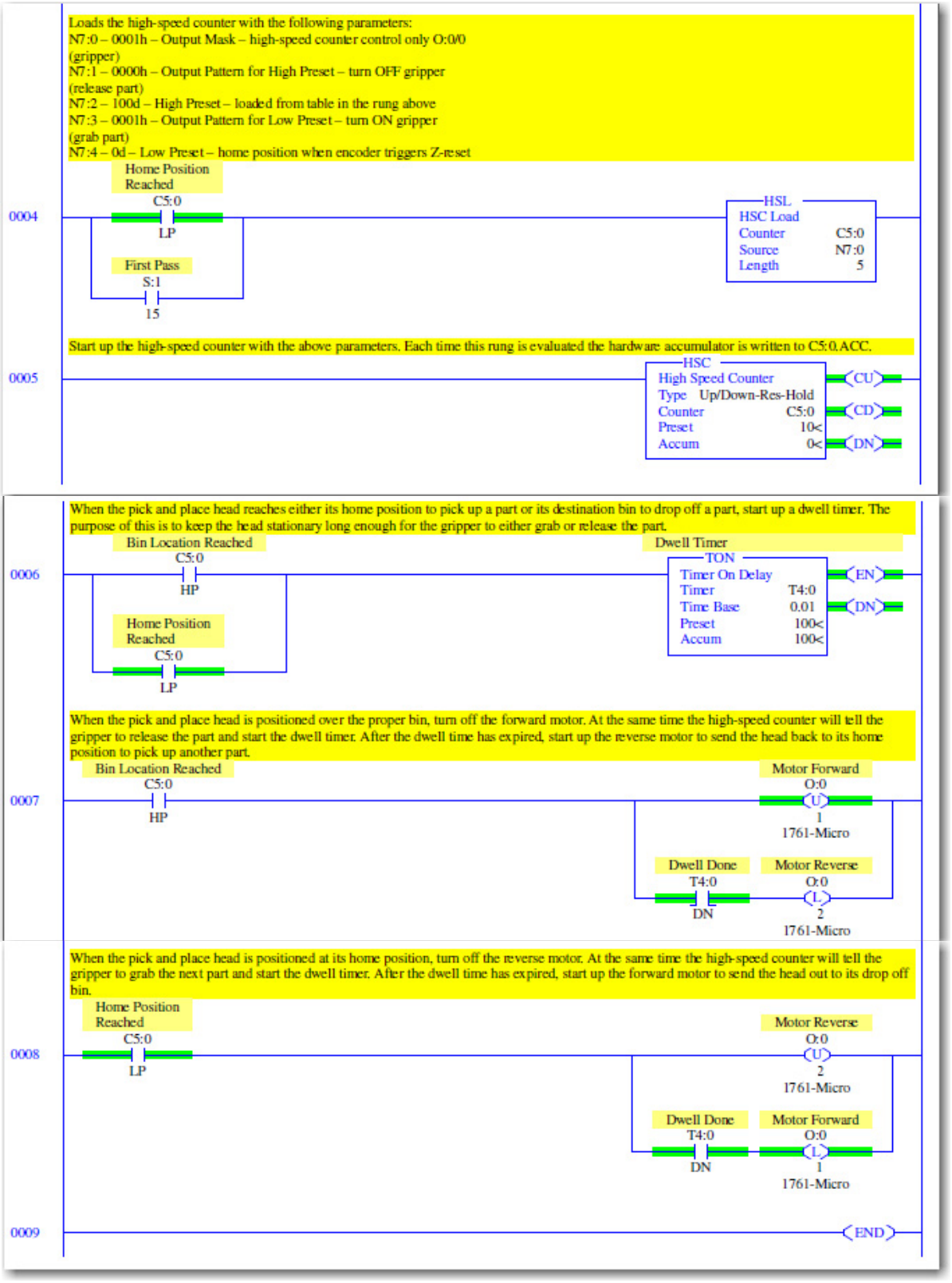
In this appendix, you can view and compare the three different ladder diagrams of the Pick-and-Place application.

- Original RSLogix 500/RSLogix Micro Ladder Diagram
- Connected Components Workbench Ladder Diagram (Converter Tool)
- Connected Components Workbench Ladder Diagram (Manual Conversion)

Original RSLogix 500/ RSLogix Micro Ladder Diagram

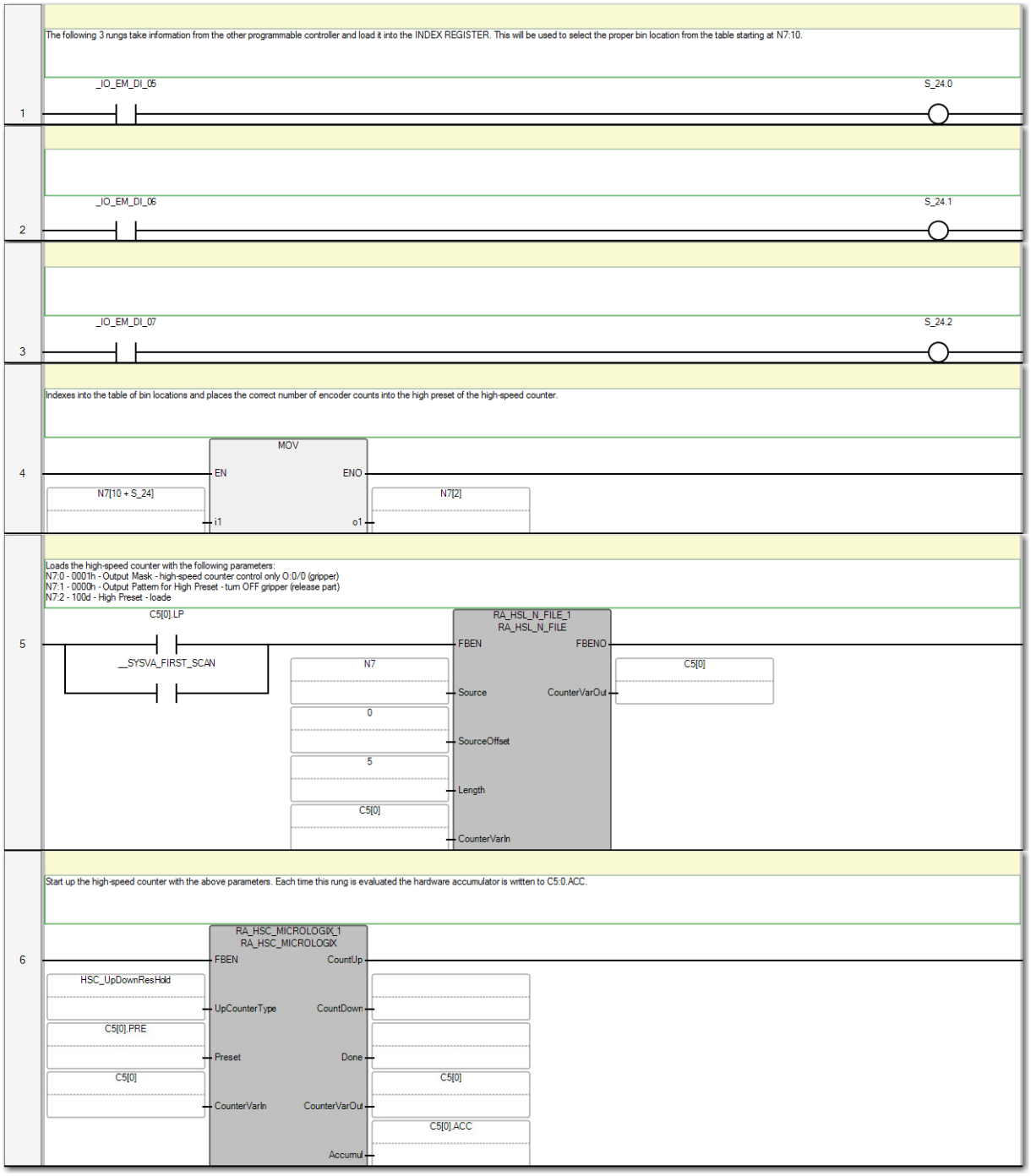
The following shows the original Pick-and-Place application ladder diagram in the RSLogix 500/RSLogix Micro report.

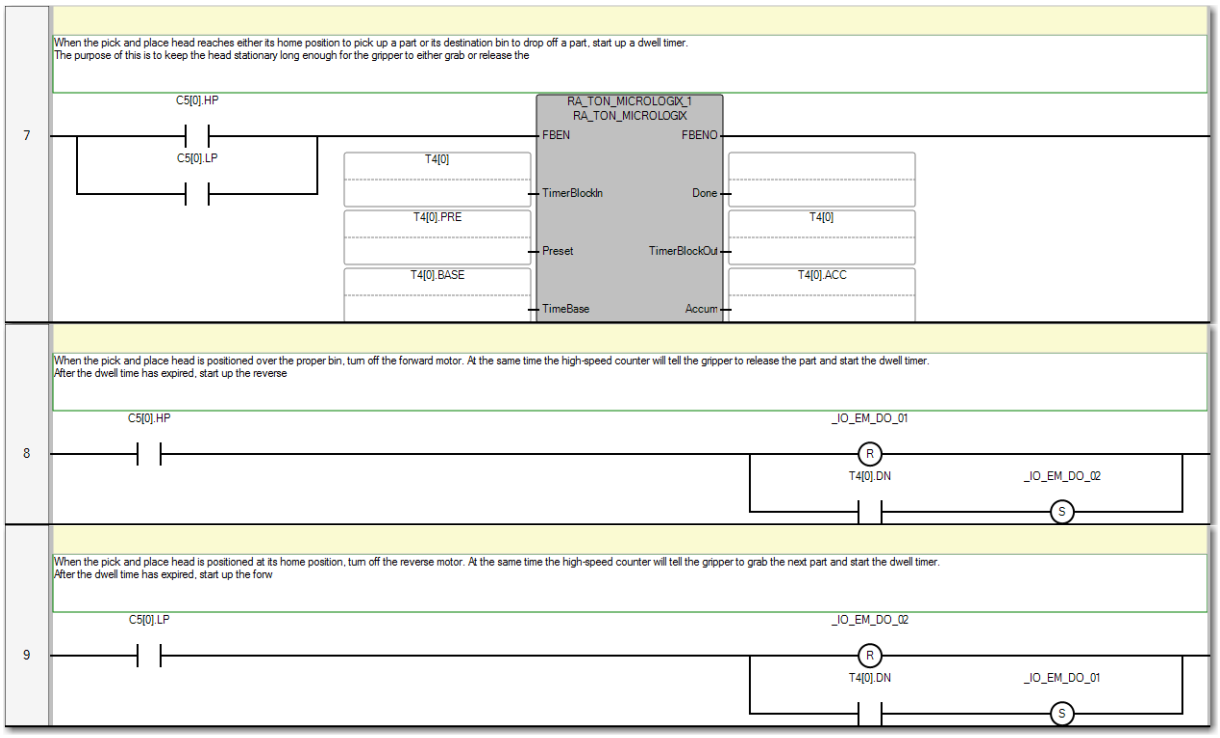



Loads the high-speed counter with the following parameters:
N7:0 – 0001h – Output Mask – high-speed counter control only O:0/0 (gripper)
N7:1 – 0000h – Output Pattern for High Preset – turn OFF gripper (release part)
N7:2 – 100d – High Preset – loaded from table in the rung above
N7:3 – 0001h – Output Pattern for Low Preset – turn ON gripper (grab part)
N7:4 – 0d – Low Preset – home position when encoder triggers Z-reset
Home Position Reached
C5:0
LP
First Pass
S:1
15
0004
HSL
HSC Load
Counter C5:0
Source N7:0
Length 5
Start up the high-speed counter with the above parameters. Each time this rung is evaluated the hardware accumulator is written to C5:0.ACC.
0005
HSC
High Speed Counter
Type Up/Down-Res-Hold
Counter C5:0
Preset 10<
Accum 0<
CU
CD
DN
When the pick and place head reaches either its home position to pick up a part or its destination bin to drop off a part, start up a dwell timer. The purpose of this is to keep the head stationary long enough for the gripper to either grab or release the part.
Bin Location Reached
C5:0
HP
Home Position Reached
C5:0
LP
Dwell Timer
0006
TON
Timer On Delay
Timer T4:0
Time Base 0.01
Preset 100<
Accum 100<
EN
DN
When the pick and place head is positioned over the proper bin, turn off the forward motor. At the same time the high-speed counter will tell the gripper to release the part and start the dwell timer. After the dwell time has expired, start up the reverse motor to send the head back to its home position to pick up another part.
Bin Location Reached
C5:0
HP
0007
Motor Forward
O:0
U
1
1761-Micro
Dwell Done
T4:0
DN
Motor Reverse
O:0
L
2
1761-Micro
When the pick and place head is positioned at its home position, turn off the reverse motor. At the same time the high-speed counter will tell the gripper to grab the next part and start the dwell timer. After the dwell time has expired, start up the forward motor to send the head out to its drop off bin.
Home Position Reached
C5:0
LP
0008
Motor Reverse
O:0
U
2
1761-Micro
Dwell Done
T4:0
DN
Motor Forward
O:0
L
1
1761-Micro
0009
END

Connected Components Workbench Ladder Diagram (Converter Tool)

The following shows the Pick-and-Place application ladder diagram that was converted with the MicroLogix to Micro800 Converter tool in Connected Components Workbench software.





Tool Conversion Results

See the following log for information on the conversion results.

------ Conversion Started ------

Source:

C:\Pick and Place Machine\PICK AND PLACE MACHINE.SLC

C:\Pick and Place Machine\PICK AND PLACE MACHINE.EAS

C:\Pick and Place Machine\PICK AND PLACE MACHINE.EIC

C:\Pick and Place Machine\PICK AND PLACE MACHINE.ERP

C:\Pick and Place Machine\PICK AND PLACE MACHINE.ESG

Destination:

Catalog Identifier: 2080-LC30-16QWB

Project: PICK AND PLACE MACHINE2

Conversion Report Location: C:\Users\user1\Documents\CCW\PICK AND PLACE MACHINE2\ConversionReport\ConversionReport.csv

Converting from Processor Type: Bul.1761 MicroLogix 1000 DH-485/ HDSlave.

Warning: Revise all usages of status file. They are no longer system variables, unpredictable operation could occur.

The properties summary information for the project cannot be converted because documentation was not included in the database export.

Warning: Arithmetic Status bits are not supported in Micro800. Revise usage of math instructions.

MicroLogix Program File 'LAD 2 - MAIN_PROG' was converted to Program 'MAIN_PROG'.

MicroLogix Program File 'LAD 3 - USER_FAULT' was converted to User-Defined Function Block 'USER_FAULT'.

MicroLogix Program File 'LAD 4 - HSC_INT' was converted to User-Defined Function Block 'HSC_INT'.

MicroLogix Program File 'LAD 5 - STI_INT' was converted to User-Defined Function Block 'STI_INT'.

MicroLogix Program File 'LAD 6' was converted to User-Defined Function Block 'FB6'.

MicroLogix Program File 'LAD 7' was converted to User-Defined Function Block 'FB7'.

MicroLogix Program File 'LAD 8' was converted to User-Defined Function Block 'FB8'.

MicroLogix Program File 'LAD 9' was converted to User-Defined Function Block 'FB9'.

MicroLogix Program File 'LAD 10' was converted to User-Defined Function Block 'FB10'.

MicroLogix Program File 'LAD 11' was converted to User-Defined Function Block 'FB11'.

MicroLogix Program File 'LAD 12' was converted to User-Defined Function Block 'FB12'.

MicroLogix Program File 'LAD 13' was converted to User-Defined Function Block 'FB13'.

MicroLogix Program File 'LAD 14' was converted to User-Defined Function Block 'FB14'.

MicroLogix Program File 'LAD 15' was converted to User-Defined Function Block 'FB15'.

MicroLogix Program File 'LAD 16' was converted to User-Defined Function Block 'FB16'.

C:\Pick and Place Machine\PICK AND PLACE MACHINE.EAS(6,1): For entry 'N7:0', no conversion occurred of empty symbol and empty description.

C:\Pick and Place Machine\PICK AND PLACE MACHINE.EAS(7,1): For entry 'N7:2', no conversion occurred of empty symbol and empty description.

C:\Pick and Place Machine\PICK AND PLACE MACHINE.EAS(8,1): For entry 'N7:10', no conversion occurred of empty symbol and empty description.

C:\Pick and Place Machine\PICK AND PLACE MACHINE.ERP(3,1): Entry 'O0000:000.000/01' was not used in the project. No conversion occurred for title: '' and description: 'When the pick and place head is positioned at its home position, turn off the reverse motor. At the same time the high-speed counter will tell the gripper to grab the next part and start the dwell timer.

After the dwell time has expired, start up the forward motor to send the head out to its drop off bin..'

C:\Pick and Place Machine\PICK AND PLACE MACHINE.ERP(5,1): Entry 'O0000:000.000/02' was not used in the project. No conversion occurred for title: '' and description: 'When the pick and place head is positioned at its home position, turn off the reverse motor. At the same time the high-speed counter will tell the gripper to grab the next part and start the dwell timer.

After the dwell time has expired, start up the forward motor to send the head out to its drop off bin..'

MicroLogix Literal Parameter 'Up/Down-Res-Hold' was converted to Defined Word 'HSC_UpDownResHold'.

Variable 'I:0.0/5' was converted to '_IO_EM_DI_05'.

Variable 'I:0.0/6' was converted to '_IO_EM_DI_06'.

Variable 'I:0.0/7' was converted to '_IO_EM_DI_07'.

Variable 'O:0.0/1' was converted to '_IO_EM_DO_01'.

Variable 'O:0.0/2' was converted to '_IO_EM_DO_02'.

Description of MicroLogix variable 'S:1/15' cannot be applied to system variable '__SYSVA_FIRST_SCAN'. (Dropping description: 'S:1/15:First Pass')

Variable 'S:1/15' was converted to '__SYSVA_FIRST_SCAN'.

MAIN_PROG(10,1): Rung Comment truncated to 255 characters. Dropping: 'd from table in the rung above

N7:3 - 0001h - Output Pattern for Low Preset - turn ON gripper

(grab part)

N7:4 - 0d - Low Preset - home position when encoder triggers Z-reset'.

MAIN_PROG(11,1): Warning: MicroLogix Variable 'S:1/15' was converted to a system variable '__SYSVA_FIRST_SCAN'.

MAIN_PROG(10,2): Warning: RA_HSL_B_FILE and RA_HSL_N_FILE User-Defined Function Blocks are rising edge triggered. The behavior is not the same as MicroLogix HSL which is level triggered.

MAIN_PROG(22,1): Rung Comment truncated to 255 characters. Dropping: ' part.'.

MAIN_PROG(22,2): Warning: Revise usage of Timer's Accumulator Value. The RA_TON_MICROLOGIX User-Defined Function Block does not start with the associated timer's accumulator value.

MAIN_PROG(27,1): Rung Comment truncated to 255 characters. Dropping: ' motor to send the head back to its home position to pick up another part.'.

MAIN_PROG(30,1): Rung Comment truncated to 255 characters. Dropping: 'ard motor to send the head out to its drop off bin.'.

Interrupt3_USER_FAULT(1,1): Warning: Interrupt 'Interrupt3_USER_FAULT' is calling subroutine 'USER_FAULT'. Revise interrupt logic and configuration.

Interrupt3_USER_FAULT(1,1): Warning: Revise all usages of 'USER_FAULT'. Each Function Block call may need to be surrounded by calls to UID and UIE to prevent unpredictable operation. This situation is indicated by build warning “Multi-thread access to global variable may need to be surrounded by calls to UID and UIE.”

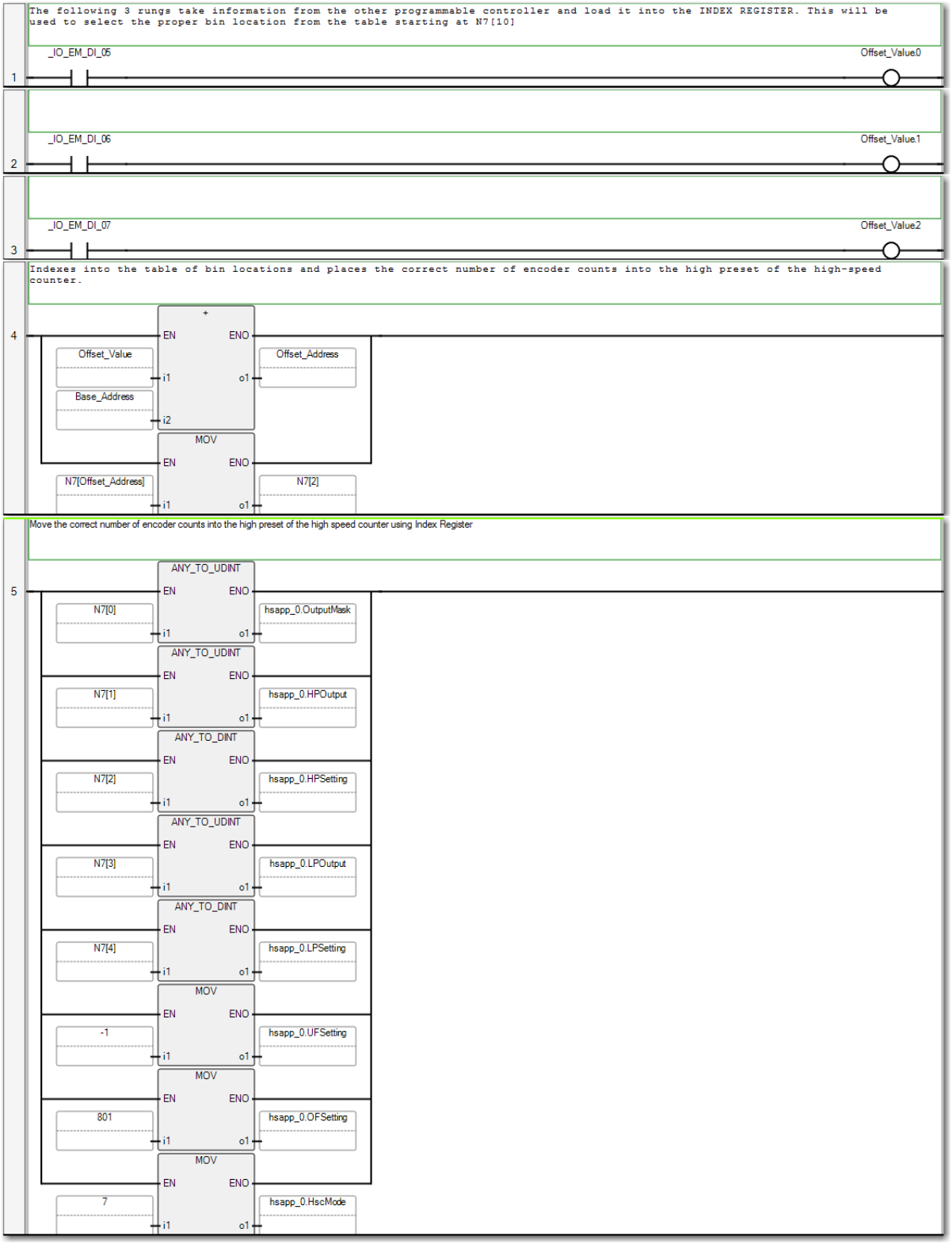
Interrupt4_HSC_INT(1,1): Warning: Interrupt 'Interrupt4_HSC_INT' is calling subroutine 'HSC_INT'. Revise interrupt logic and configuration.

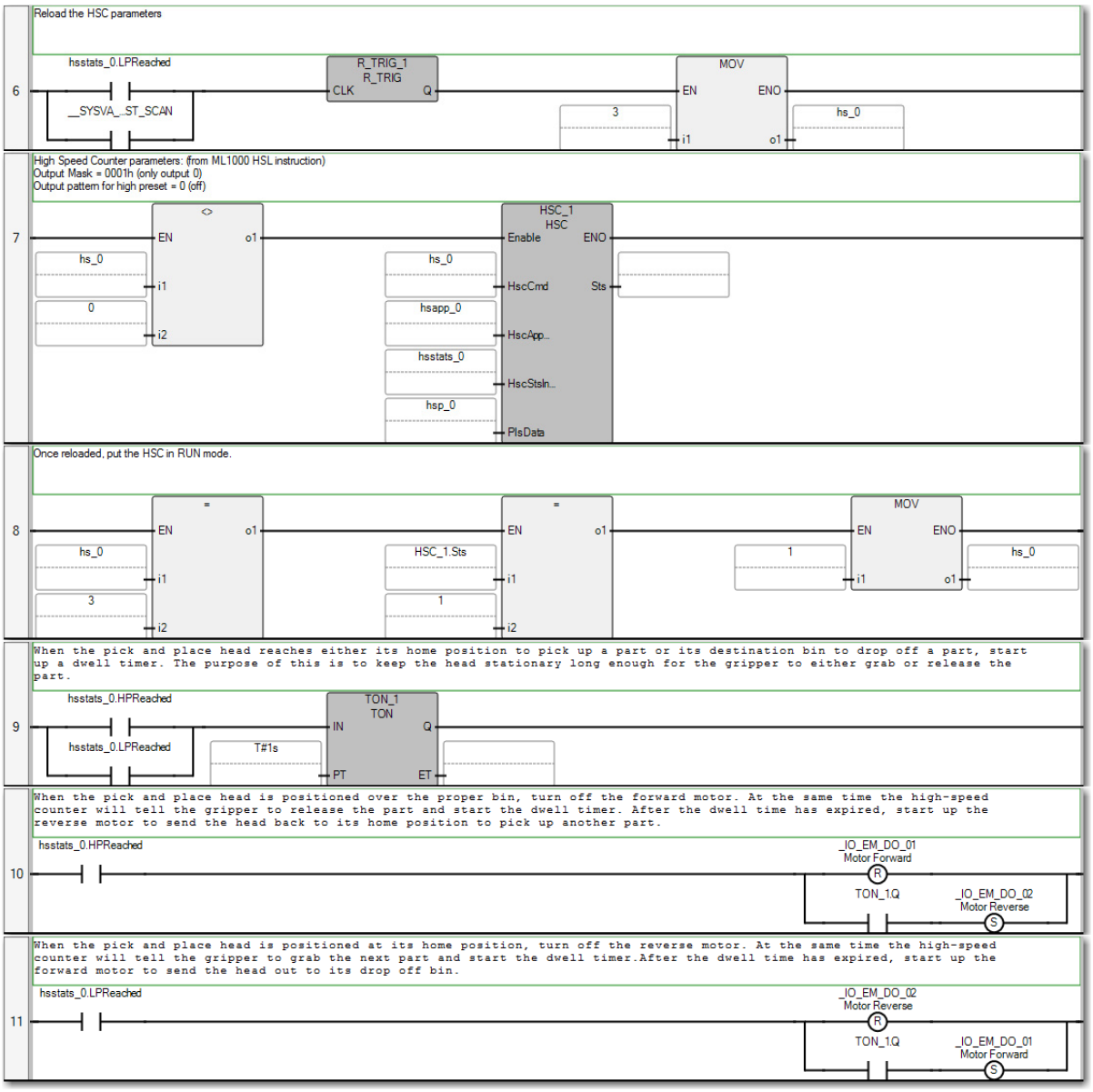
Interrupt4_HSC_INT(1,1): Warning: Revise all usages of 'HSC_INT'. Each Function Block call may need to be surrounded by calls to UID and UIE to prevent unpredictable operation. This situation is indicated by build warning “Multi-thread access to global variable may need to be surrounded by calls to UID and UIE.”

========= Conversion ends with 0 error(s) and 9 warning(s). =========

Connected Components Workbench Ladder Diagram (Manual Conversion)

The following shows the Pick-and-Place application ladder diagram that was converted manually in Connected Components Workbench software.




Reload the HSC parameters
6
hsstats_0.LPReached
__SYSVA_..ST_SCAN
R_TRIG_1
R_TRIG
CLK
Q
MOV
EN
ENO
3
i1
hs_0
o1
High Speed Counter parameters: (from ML1000 HSL instruction)
Output Mask = 0001h (only output 0)
Output pattern for high preset = 0 (off)
7
<>
EN
o1
hs_0
i1
0
i2
HSC_1
HSC
Enable
ENO
hs_0
HscCmd
Sts
hsapp_0
HscApp...
hsstats_0
HscStsIn...
hsp_0
PlsData
Once reloaded, put the HSC in RUN mode.
8
=
EN
o1
hs_0
i1
3
i2
HSC_1.Sts
1
MOV
ENO
When the pick and place head reaches either its home position to pick up a part or its destination bin to drop off a part, start up a dwell timer. The purpose of this is to keep the head stationary long enough for the gripper to either grab or release the part.
9
hsstats_0.HPReached
hsstats_0.LPReached
TON_1
TON
IN
Q
T#1s
PT
ET
When the pick and place head is positioned over the proper bin, turn off the forward motor. At the same time the high-speed counter will tell the gripper to release the part and start the dwell timer. After the dwell time has expired, start up the reverse motor to send the head back to its home position to pick up another part.
10
_IO_EM_DO_01
Motor Forward
R
TON_1.Q
_IO_EM_DO_02
Motor Reverse
S
When the pick and place head is positioned at its home position, turn off the reverse motor. At the same time the high-speed counter will tell the gripper to grab the next part and start the dwell timer.After the dwell time has expired, start up the forward motor to send the head out to its drop off bin.
11
hsstats_0.LPReached
_IO_EM_DO_02
Motor Reverse
R
TON_1.Q
_IO_EM_DO_01
Motor Forward
S

Notes:

Rockwell Automation Support

Use the following resources to access support information.

Technical Support Center	Knowledgebase Articles, How-to Videos, FAQs, Chat, User Forums, and Product Notification Updates.	https://rockwellautomation.custhelp.com/
Local Technical Support Phone Numbers	Locate the phone number for your country.	http://www.rockwellautomation.com/global/support/get-support-now.page
Direct Dial Codes	Find the Direct Dial Code for your product. Use the code to route your call directly to a technical support engineer.	http://www.rockwellautomation.com/global/support/direct-dial.page
Literature Library	Installation Instructions, Manuals, Brochures, and Technical Data.	http://www.rockwellautomation.com/global/literature-library/overview.page
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	http://www.rockwellautomation.com/global/support/pcdc.page

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the How Are We Doing? form at http://literature.rockwellautomation.com/idc/groups/literature/documents/du/ra-du002_-en-e.pdf.

Rockwell Automation maintains current product environmental information on its website at http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page.

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 2080-RM002B-EN-E - June 2019
Supersedes Publication 2080-RM002A-EN-E - July 2015