



Micro820 Programmable Controllers

Catalog Numbers 2080-LC20-20QWB, 2080-LC20-20QBB, 2080-LC20-20AWB, 2080-LC20-20QWBR, 2080-LC20-20QBRR, 2080-LC20-20AWBR



Important User Information

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

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

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

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

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

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.



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



ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence



SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.



BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.



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).

IMPORTANT

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

Read this preface to familiarize yourself with the rest of the manual. It provides information concerning:

- who should use this manual
- the purpose of this manual
- related documentation
- supporting information for Micro800™

Who Should Use this Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use Micro800 controllers.

You should have a basic understanding of electrical circuitry and familiarity with relay logic. If you do not, obtain the proper training before using this product.

Purpose of this Manual

This manual is a reference guide for Micro820 controllers. It describes the procedures you use to install, wire, and troubleshoot your controller. This manual:

- explains how to install and wire your controllers
- gives you an overview of the Micro800 controller system

Refer to the Online Help provided with Connected Components Workbench™ software for more information on programming your Micro800 controller.

Additional Resources

These documents contain additional information concerning related Rockwell Automation products.

Resource	Description
Micro800 Plug-in Modules 2080-UM004	Information on features, configuration, installation, wiring, and specifications for the Micro800 plug-in modules.
Micro800 Programmable Controllers General Instructions 2080-RM001	Information on instruction sets for developing programs for use in Micro800 control systems.
Micro800 Programmable Controllers: Getting Started with CIP Client Messaging 2080-QS002	Provides quickstart instructions for using CIP GENERIC and CIP Symbolic Messaging.
Micro800 Programmable Controllers: Getting Started with PanelView Plus 2080-QS003	Provides quickstart instructions for using global variables for Micro800 controllers together with PanelView Plus HMI terminals.
Configuring Micro800 Controllers on FactoryTalk Gateway 2080-QS005	Provides quickstart instructions for configuring a Micro800 controller on FactoryTalk Gateway.
Micro800 Programmable Controller External AC Power Supply Installation Instructions 2080-IN001	Information on mounting and wiring the optional external power supply.
Micro820 Programmable Controllers Installation Instructions, 2080-IN009	Information on installing, mounting, and wiring the Micro820 controller.
Micro800 Remote LCD Installation Instructions, 2080-IN010	Information on installing, mounting, and wiring the Micro800 Remote LCD module.

Resource	Description
Micro800 RS232/485 Isolated Serial Port Plug-in Module Wiring Diagrams 2080-WD002	Information on mounting and wiring the Micro800 RS232/485 Isolated Serial Port Plug-in Module.
Micro800 Non-isolated Unipolar Analog Input Plug-in Module Wiring Diagrams 2080-WD003	Information on mounting and wiring the Micro800 Non-isolated Unipolar Analog Input Plug-in Module.
Micro800 Non-isolated Unipolar Analog Output Plug-in Module Wiring Diagrams 2080-WD004	Information on mounting and wiring the Micro800 Non-isolated Unipolar Analog Output Plug-in Module.
Micro800 Non-isolated RTD Plug-in Module Wiring Diagrams 2080-WD005	Information on mounting and wiring the Micro800 Non-isolated RTD Plug-in Module.
Micro800 Non-isolated Thermocouple Plug-in Module Wiring Diagrams 2080-WD006	Information on mounting and wiring the Micro800 Non-isolated Thermocouple Plug-in Module.
Micro800 Memory Backup and High Accuracy RTC Plug-In Module Wiring Diagrams 2080-WD007	Information on mounting and wiring the Micro800 Memory Backup and High Accuracy RTC Plug-In Module.
Micro800 6-Channel Trimpot Analog Input Plug-In Module Wiring Diagrams 2080-WD008	Information on mounting and wiring the Micro800 6-Channel Trimpot Analog Input Plug-In Module.
Micro800 Digital Relay Output Plug-in Module Wiring Diagrams 2080-WD010	Information on mounting and wiring the Micro800 Digital Relay Output Plug-in Module.
Micro800 Digital Input, Output, and Combination Plug-in Modules Wiring Diagrams 2080-WD011	Information on mounting and wiring the Micro800 Digital Input, Output, and Combination Plug-in Modules.
Micro800 High Speed Counter Plug-in Module, 2080-WD012	Information on mounting and wiring the High Speed Counter Plug-in module.
Micro800 DeviceNet Plug-in Module, 2080-WD013	Information on mounting and wiring the Micro800 DeviceNet plug-in module.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, http://www.rockwellautomation.com/products/certification/	Provides declarations of conformity, certificates, and other certification details.
Application Considerations for Solid-State Controls SG1-1.1	A description of important differences between solid-state programmable controller products and hard-wired electromechanical devices.
National Electrical Code - Published by the National Fire Protection Association of Boston, MA.	An article on wire sizes and types for grounding electrical equipment.
Allen-Bradley Industrial Automation Glossary AG-7.1	A glossary of industrial automation terms and abbreviations.

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

You can download the latest version of Connected Components Workbench for your Micro800 at the URL below.

<http://www.rockwellautomation.com/global/products-technologies/connected-components/>.

Preface	Who Should Use this Manual	iii
	Purpose of this Manual	iii
	Additional Resources	iii
	Chapter 1	
Hardware Overview	Hardware Features	1
	Chapter 2	
About Your Controller	Programming Software for Micro800 Controllers	7
	Controller Changes in Run Mode	7
	Using Run Mode Change (RMC)	8
	Using Run Mode Configuration Change (RMCC)	13
	Agency Certifications	18
	Compliance to European Union Directives	19
	Installation Considerations	20
	Safety Considerations	22
	Power Considerations	24
	Preventing Excessive Heat	25
	Master Control Relay	25
	Chapter 3	
Install Your Controller	Controller Mounting Dimensions	31
	Connect the Controller to an EtherNet/IP Network	33
	Install the microSD Card	34
	Install the 2080-REMLCD Module	35
	Chapter 4	
Wire Your Controller	Wiring Requirements and Recommendation	37
	Use Surge Suppressors	38
	Grounding the Controller	41
	Wiring Diagrams	41
	Controller I/O Wiring	48
	Wiring Analog Channels	51
	Chapter 5	
Communication Connections	Overview	55
	Supported Communication Protocols	55
	CIP Communications Pass-thru	60
	Use Modems with Micro800 Controllers	61
	Configure Serial Port	62
	Configure Ethernet Settings	68
	OPC Support Using RSLinx Enterprise	70

	Chapter 6	
Program Execution in Micro800	Overview of Program Execution	71
	Optional Module.....	72
	Controller Load and Performance Considerations	73
	Power Up and First Scan.....	74
	Memory Allocation.....	75
	Guidelines and Limitations for Advanced Users.....	75
	Chapter 7	
Controller Security	Exclusive Access.....	77
	Password Protection.....	77
	Compatibility.....	78
	Work with a Locked Controller	79
	Configure Controller Password.....	81
	Recover from a Lost Password	82
	Chapter 8	
Using the Micro800 Remote LCD	Overview	83
	USB Mode.....	84
	Text Display Mode	85
	Backup and Restore.....	91
	Hardware Features, Installation, and Specifications.....	91
	Chapter 9	
Using microSD Cards	Overview	93
	Project Backup and Restore	94
	Data Log.....	99
	Recipe	104
	Quickstart Projects for Data Log and Recipe Function Blocks	108
	Appendix A	
Specifications	General Specifications	123
	Environmental Specifications.....	131
	Certifications	132
	Appendix B	
Troubleshooting	Status Indicators on the Controller	133
	Error Codes.....	135
	Controller Error Recovery Model.....	144
	Calling Rockwell Automation for Assistance.....	145

Quickstarts	Appendix C
	Flash Upgrade Your Micro800 Firmware..... 147
	Establish Communications between RSLinx and a Micro820 Controller through USB Port on 2080-REMLCD 154
	Configure Controller Password 158
	Forcing I/Os..... 162
	Using Run Mode Change..... 164
PID Function Blocks	Appendix D
 171
	PID Function Block..... 172
	IPIDCONTROLLER Function Block..... 175
	How to Autotune 178
	Troubleshooting an Autotune Process 179
	PID Application Example..... 180
Modbus Mapping for Micro800	Appendix E
	Modbus Mapping 183
Index189

Notes:

Hardware Overview



This chapter provides an overview of the Micro820 hardware features. It has the following topics:

Topic	Page
Hardware Features	1
Embedded microSD (Micro Secure Digital) Card Slot	3
Embedded RS232/RS485 Serial Port Combo	4
Embedded Ethernet Support	5

Hardware Features

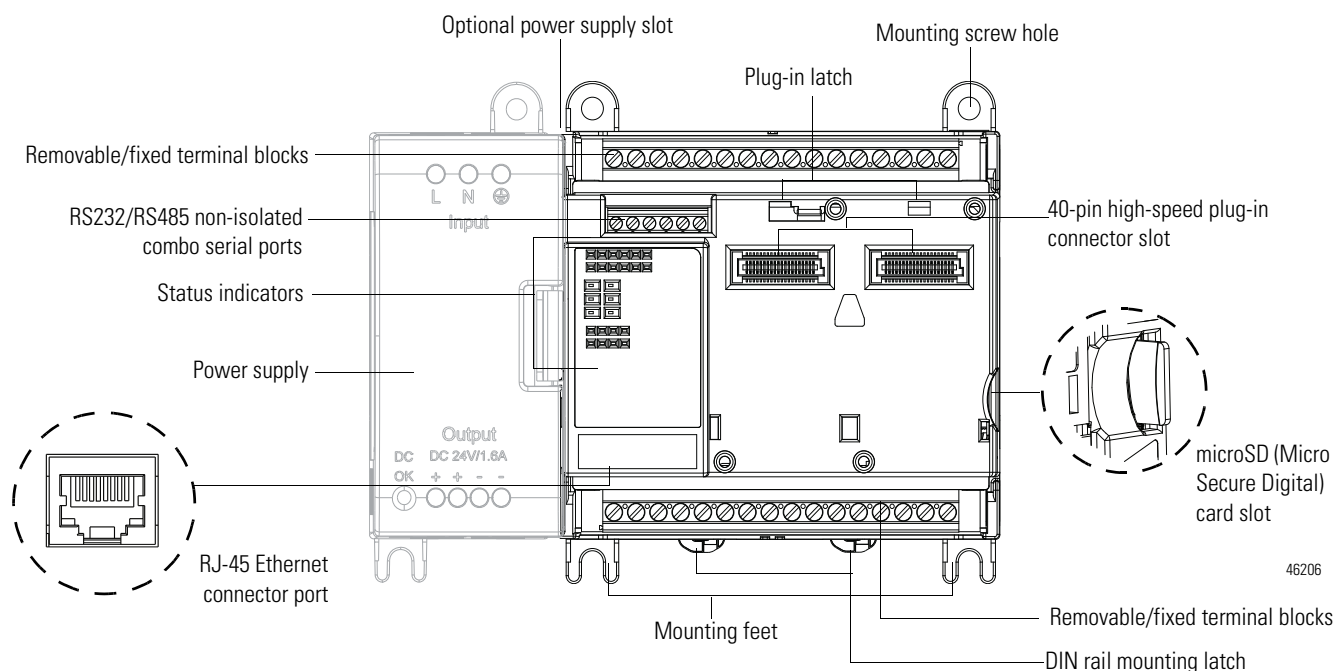
Micro820 controllers are 20-point economical brick style controllers with embedded inputs and outputs. These controllers can accommodate up to two plug-in modules and can connect to a remote LCD (2080-REMLCD) for configuring. The Micro820 controller also has a microSD™ card slot for project backup and restore, and data log and recipe.

IMPORTANT The Micro820 controller supports all Micro800 plug-in modules, except for the 2080-MEMBAK-RTC.
For more information, see Micro800 Plug-in Modules User, publication [2080-UM004](#).

For information on the REMLCD module, see [Using the Micro800 Remote LCD on page 83](#).

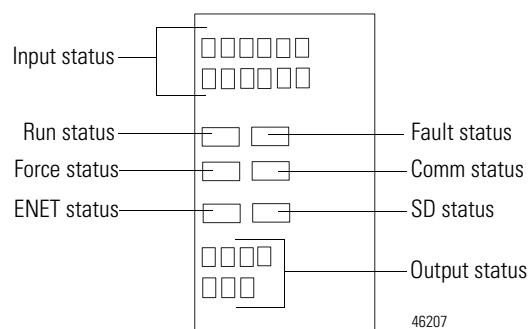
The controller also accommodates any class 2 rated 24V DC output power supply that meets minimum specifications such as the optional Micro800 power supply.

Micro820 Controllers



ATTENTION: Removable terminal blocks are available on catalog numbers that end in R (for example, 2080-LC20-20QBBR). Fixed terminal blocks are available on catalog numbers that do not end in R (for example, 2080-LC20-20QBB).

Status Indicators



See [Troubleshooting on page 133](#) for descriptions of status indicator operation.

Inputs and Outputs

Number and Types of Inputs/Outputs for Micro820 Controllers

Controller Family	Catalogs	Inputs			Outputs			Analog Out 0...10V DC	Analog In 0...10V (shared with DC In)	PWM Support
		120V AC	120 / 240V AC	24V DC	Relay	24V DC Source	24V DC Sink			
Micro820	2080-LC20-20QBB	—	—	12	—	7	—	1	4	1
	2080-LC20-20QWB	—	—	12	7	—	—	1	4	—
	2080-LC20-20AWB	8	—	4	7	—	—	1	4	—
	2080-LC20-20QBBR	—	—	12	—	7	—	1	4	1
	2080-LC20-20QWBR	—	—	12	7	—	—	1	4	—
	2080-LC20-20AWBR	8	—	4	7	—	—	1	4	—

Embedded microSD (Micro Secure Digital) Card Slot

Micro820 controllers support microSD cards through an embedded microSD card slot. The design supports Class 6 and 10 SDSC and SDHC microSD cards, with FAT32/16 formats. The microSD file system supports only one file partition. Class 4 cards are **not** supported.

The microSD card is primarily used for project backup and restore, as well as data log and recipe functions. It can also be used to configure powerup settings (such as controller mode, IP address, and so on) through an optional ConfigMeFirst.txt file.

For more information, see [Using microSD Cards on page 93](#).

To help you troubleshoot microSD card-related errors, see [Troubleshooting on page 133](#).

For a list of microSD cards that have been tested for industrial use, go to the Rockwell Automation Support Center at <https://rockwellautomation.custhelp.com/app/home/> and search the Knowledgebase with Answer ID 732175.

IMPORTANT You must sign in to the Rockwell Automation website before searching the Knowledgebase.

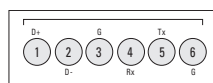
Embedded RS232/RS485 Serial Port Combo

The Micro820 controller supports an embedded non-isolated RS232/RS485 combo communications port. Only one port (RS232 or RS485) can work at any given time. The baud rate of this port supports up to 38.4 K.

The communication port uses a 6-pin 3.5 mm terminal block with pin definition shown in the following table.

IMPORTANT Serial port cables should not exceed 3 m length.

RS232/RS485 Serial Port Pin Definition



Pin	Definition	RS485 Example	RS232 Example
1	RS485+	RS485+	(not used)
2	RS485-	RS485-	(not used)
3	GND	GND	GND
4	RS232 input (receiver)	(not used)	RxD
5	RS232 output (driver)	(not used)	TxD
6	GND	GND	GND

The communication port (both RS232 and RS485) are non-isolated. The signal ground of the port is not isolated to the logic ground of the controller.

The RS232 port supports connection to the Micro800 Remote LCD module (2080-REMLCD).

REMLCD to Micro820 Serial Port Terminal Block Wiring

REMLCD Serial Port Terminal Block		Micro820 Serial Port Terminal Block	
Signal	Pin number	Pin number	Signal
RS232 TX	1	4	RX RS232
RS232 RX	2	5	TX RS232
RS232 G	3	6	G RS232

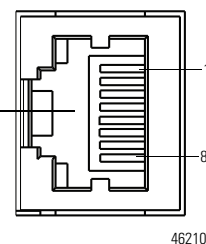
Embedded Ethernet Support

A 10/100 Base-T Port is available for connection to an Ethernet network through any standard RJ-45 Ethernet cable.

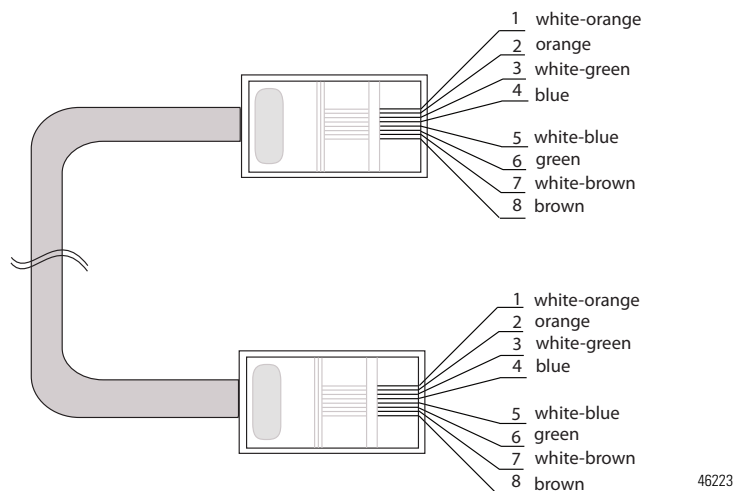
RJ-45 Ethernet Port Pin Mapping

Contact Number	Signal	Direction	Primary Function
1	TX+	OUT	Transmit data +
2	TX-	OUT	Transmit data -
3	RX+	IN	Receive data +
4	—	—	—
5	—	—	—
6	RX-	IN	Receive data -
7	—	—	—
8	—	—	—

RJ-45 connector



Ethernet port pin-to-pin connection



See [Troubleshooting on page 133](#) for descriptions of ENET status indicator.

Notes:

About Your Controller

Programming Software for Micro800 Controllers

Connected Components Workbench is a set of collaborative tools supporting Micro800 controllers. It is based on Rockwell Automation and Microsoft Visual Studio technology and offers controller programming, device configuration and integration with HMI editor. Use this software to program your controllers, configure your devices and design your operator interface applications.

Connected Components Workbench 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.

Obtain Connected Components Workbench

A free download is available at:

<http://www.rockwellautomation.com/global/products-technologies/connected-components/>.

Use Connected Components Workbench

To help you program your controller through the Connected Components Workbench software, you can refer to the Connected Components Workbench Online Help (it comes with the software).

Controller Changes in Run Mode

Micro820/Micro830/Micro850 controllers allow you to make certain changes while in run mode by using the following features:

- Run Mode Change (RMC)
This feature allows logic modifications to a running project without going to remote program mode.
For more information, see [Using Run Mode Change \(RMC\) on page 8](#).
- Run Mode Configuration Change (RMCC)
This feature allows changing the address configuration of the controller to be made within a program during run mode.
For more information, see [Using Run Mode Configuration Change \(RMCC\) on page 13](#).

Using Run Mode Change (RMC)

Run Mode Change (RMC) is a productivity enhancement feature introduced in Release 8 for Micro820/Mico830/Micro850 controllers. It saves the user time by allowing logic modifications to a running project without going to remote program mode and without disconnecting from the controller. It requires the Connected Components Workbench Developer Edition release 8 software to use this feature.

IMPORTANT Micro820/Micro830/Micro850 controller firmware revision 8.xxx or higher is also required to use Run Mode Change.

RMC is useful when the user is developing a project by incrementally adding small changes to the logic and immediately wants to see the effects of the changes on the machine. With RMC, since the controller stays in remote run mode, the controller logic and machine actuators will not have to constantly reinitialize, which can occur if the controller is switched to remote program mode (for example, first scan bit is checked in program logic to clear outputs).

When user is editing, building, and downloading a project without using RMC, a full build of the entire controller project is performed and also a full download of the project is performed. During RMC an incremental build is performed and only incremental changes are downloaded to the controller.

IMPORTANT Do not disconnect from the controller after performing Run Mode Change, do a full build, and try to reconnect. Connected Components Workbench will treat the project in the controller as different from the project in Connected Components Workbench, and ask to either upload or download even though the logic is identical.

RMC is performed incrementally at the end of every program scan in order to prevent a large delay in the program scan. This adds up to an additional 12 ms to the scan time. For example, if the program scan is normally 10 ms, it may increase to 22 ms during RMC until the update is finished. Similarly user interrupts may be delayed.

Example of the Benefits of Using RMC – 20% Reduction in Download Time

Number of Changes	Time to Perform Conventional Download (seconds)	Time to Test Logic and Accept Changes (seconds)
1	36	29
5	180	130
10	360	255

Memory size of project used for comparison:

Data = 14784 bytes; Program = 2352 bytes

Note: The duration starts when the RMC button is clicked while connected to the controller and ends when the accept is finished. For example:

1. When connected to the controller, click RMC
 2. Modify program
 3. Click Test Logic
 4. Click Accept to finish, or click Test Logic to make another change
-



ATTENTION: Use extreme caution when you use Run Mode Change. Mistakes can injure personnel and damage equipment. Before using Run Mode Change:

- assess how machinery will respond to the changes.
- notify all personnel about the changes.

A new global variable `__SYSVA_PROJ_INCOMPLETE` has been added to indicate when Run Mode Changes are being made. This can be used to notify personnel on the HMI that there are uncommitted changes in the controller.

Bit Definitions of Global Variable – `__SYSVA_PROJ_INCOMPLETE`

Bit	Definition
0	Set when the Run Mode Change process starts. Cleared once the Run Mode Change is written permanently to the controller (completion of Accept or Undo). This bit can be used to warn operators that a run mode change is in progress and that there are uncommitted changes in the controller.
1	Set if an error occurred while saving the changes to flash or an integrity check failed during Run Mode Change. Cleared on the next successful Run Mode change.

When you perform a Test Logic Change, the value of the variable is changed from zero to one. After you choose to accept or undo the changes, the value of the variable is reset to zero.

IMPORTANT When a Test Logic is performed, or undoing changes after the Test Logic is completed, any active communication instructions will be aborted while the changes are downloaded to the controller.

Uncommitted Changes

Uncommitted changes are changes made in RMC that have not been accepted or undone after a Test Logic Change has been performed.

If the controller power loses power while there are uncommitted changes, you will not be able to re-enter RMC upon reconnection. You can choose to re-download the project to keep the changes, or upload if the uncommitted changes are not wanted.

If you choose to upload a project with uncommitted changes from the controller, you cannot enter RMC until you have done a full download.

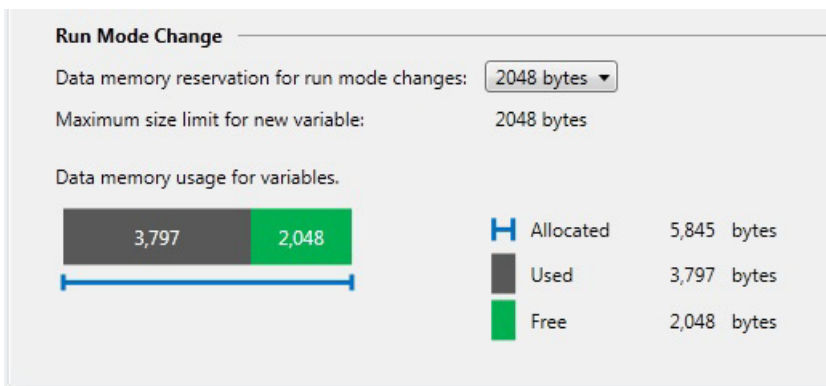
RMC Memory

Run Mode Change (RMC) memory is used to store both the logic and user variable changes made during RMC. The default amount of memory allocated is 2KB and can be increased up to 8KB. However there is still a limit of 2KB for logic and user variables changes per Test Logic. To adjust the amount of RMC memory, the controller must be offline. After you have adjusted the amount, you must build the project and download it to the controller.

IMPORTANT In a Connected Components Workbench Release 8 project, the available user data space was reduced by 6 KB to support optimal project settings for the new RMC feature.

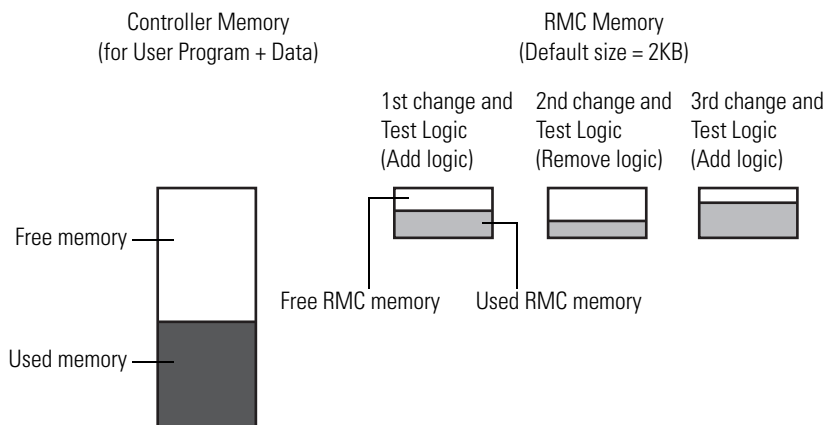
If you have a project that was developed before Release 8, you may need to reduce the default "Allocated" 8 KB Temporary Variables section from the Memory page in order to compile the project successfully.

Controller Memory Diagnostics Page in Connected Components Workbench



During RMC an incremental build is performed and only incremental changes are downloaded to the controller until the RMC memory has been filled.

RMC Memory Usage Example

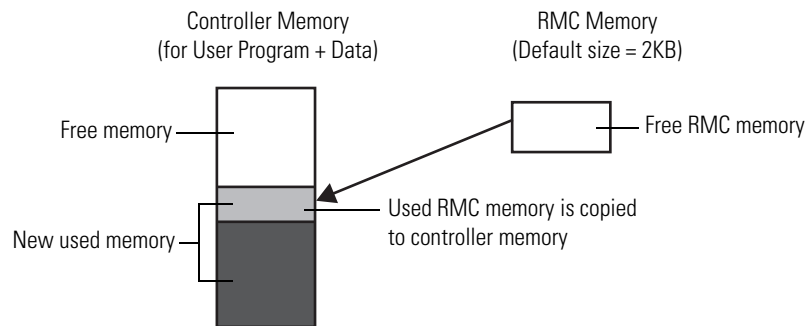


If not enough RMC memory is available to make more changes (for example, a “not enough memory” error message appears during RMC build or Test Logic), then a full download must be performed to transfer the incremental changes from the RMC memory to standard user program and data memory.

Transferring Contents in RMC Memory to Controller Memory

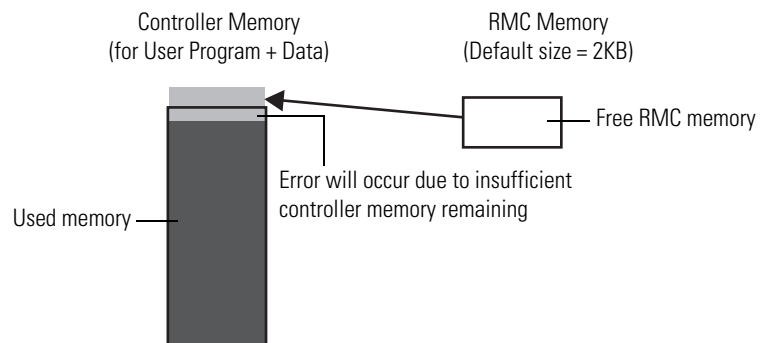
The changes that you have made during RMC are stored in RMC memory and will remain there until you perform a full build and download (while the controller is disconnected).

RMC Memory Usage When Performing Full Build and Download Example



However if the controller memory does not have enough space remaining to copy the contents of the RMC memory as shown below, the operation will fail and a “not enough memory” error message will appear. Do not use RMC if you are near the limits of your controller memory.

Insufficient Controller Memory Example



Limitations of RMC

Take note of the following limitations when using the Run Mode Change (RMC) feature:

- Configuration changes cannot be made (for example, change filter times).
- Up to 2KB of logic (approximately 150 boolean instructions) and user variables and can be added for each Test Logic.
- Total memory allocated for RMC (cumulative of all Test Logic Changes) can be increased from 2KB to 8KB, but the 2KB limit for logic and user variables per Test Logic remains.
- A total of 20 POU (Program Organizational Units) can be added for each RMC (for example, if you currently have 5 POU, you can add 20 more for a total of 25 POU).
- If a User Defined Function Block is modified that changes the local variables, the local variables will be reinitialized or reset to zero and a warning message will be shown during the build. If you want to reapply the initial value, right-click on the UDFB and select Refactor -> Reset Initial Values of Instances.
- RMC is not possible after doing a Discover Project operation if a new module is detected because the configuration has changed.
- Exchange files cannot be imported when in RMC because it is considered a configuration change.
- Making changes to the display configuration (for example, hiding comments) are treated as logic changes and require you to build the project.
- Global variables cannot be deleted or modified in RMC, but can be added. To delete or modify a global variable, the workbench must be disconnected from the controller.
- If you create a global variable in RMC, it will not be shown in the LCD display.
- When using CIP messaging in RMC, setting the CIPTARGETCFG data type parameter ConnClose to TRUE has no effect. The Ethernet session does not close immediately upon successful messaging and you have to wait for the connection to timeout after 60 seconds. This applies to Release 9 or earlier projects. For Release 10 or later projects, the CIP connection timeout is configurable.



WARNING: If you delete the output rung when in Run Mode Change and accept the changes, the output on the controller will remain ON.

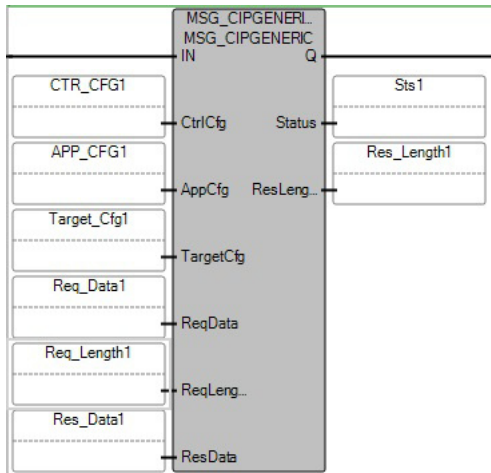
See [Using Run Mode Change on page 164](#) for an example on how to use this feature.

Using Run Mode Configuration Change (RMCC)

Run Mode Configuration Change (RMCC) is a productivity enhancement feature introduced in Release 9 for Micro820/Micro830/Micro850 controllers. It allows users to reuse an identical program with multiple controllers simply by changing the address configuration of a controller within the program during run mode. Micro820/Micro830/Micro850 controller firmware revision 9.xxx or higher is required to use this feature.

RMCC can be used to change the address configuration of the controller during run mode when the communication protocol is set to Modbus RTU for serial ports or EtherNet/IP for the Ethernet port. RMCC uses a CIP Generic message which can only be sent from within a controller program and not from an external device to the controller.

CIP Generic Message Instruction for Run Mode Configuration Change



Run Mode Configuration Change (RMCC) can only be performed by the controller that is sending the message. To do that, you need to configure the CIP Generic message as a loop-back message by setting the path to “0,0”.

Configure CIP Generic Message as a Loop-back Message

Name	Data Type	Dimension	String Size	Initial
Target_Cfg1	CIPTARGETCFG			...
Target_Cfg1.Path	STRING		80	'0,0'
Target_Cfg1.CipConnMode	USINT			0
Target_Cfg1.UcommTimeout	UDINT			0
Target_Cfg1.ConnMsgTimeout	UDINT			0
Target_Cfg1.ConnClose	BOOL			

For Micro820 controllers, the address configuration change is not permanent and will be lost when the controller is power cycled. If you want the address to be changed immediately after a power cycle, you must execute this CIP Generic message after every power up. This can be done by doing the following:

- Use the system variable __SYSVA_POWER_UP_BIT.
- Set the Retained flag for the variable containing the new address.

Set Retained Flag – Modbus Address

Prog1-VAR ▸ × Prog1-POU* Micro820 Start Page						
	Name	Data Type	Dimension	String Size	Project Value	Retained
	Req_Data1 ▾ ▴	▾ ▴	▾ ▴	▾ ▴	▾ ▴	▾ ▴
-	Req_Data1	USINT ▾	[1..22]		...	<input checked="" type="checkbox"/>
	Req_Data1[1]	USINT			3	<input checked="" type="checkbox"/>
	Req_Data1[2]	USINT			1	<input checked="" type="checkbox"/>
	Req_Data1[3]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[4]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[5]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[6]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[7]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[8]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[9]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[10]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[11]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[12]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[13]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[14]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[15]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[16]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[17]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[18]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[19]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[20]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[21]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[22]	USINT			0	<input checked="" type="checkbox"/>

Set Retained Flag – EtherNet/IP Address

Prog1-VAR ▸ × Prog1-POU* Micro820 Start Page						
	Name	Data Type	Dimension	String Size	Project Value	Retained
	Req_Data1 ▾ ▴	▾ ▴	▾ ▴	▾ ▴	▾ ▴	▾ ▴
-	Req_Data1	USINT ▾	[1..22]		...	<input checked="" type="checkbox"/>
	Req_Data1[1]	USINT			10	<input checked="" type="checkbox"/>
	Req_Data1[2]	USINT			1	<input checked="" type="checkbox"/>
	Req_Data1[3]	USINT			168	<input checked="" type="checkbox"/>
	Req_Data1[4]	USINT			192	<input checked="" type="checkbox"/>
	Req_Data1[5]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[6]	USINT			255	<input checked="" type="checkbox"/>
	Req_Data1[7]	USINT			255	<input checked="" type="checkbox"/>
	Req_Data1[8]	USINT			255	<input checked="" type="checkbox"/>
	Req_Data1[9]	USINT			1	<input checked="" type="checkbox"/>
	Req_Data1[10]	USINT			1	<input checked="" type="checkbox"/>
	Req_Data1[11]	USINT			168	<input checked="" type="checkbox"/>
	Req_Data1[12]	USINT			192	<input checked="" type="checkbox"/>
	Req_Data1[13]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[14]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[15]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[16]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[17]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[18]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[19]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[20]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[21]	USINT			0	<input checked="" type="checkbox"/>
	Req_Data1[22]	USINT			0	<input checked="" type="checkbox"/>

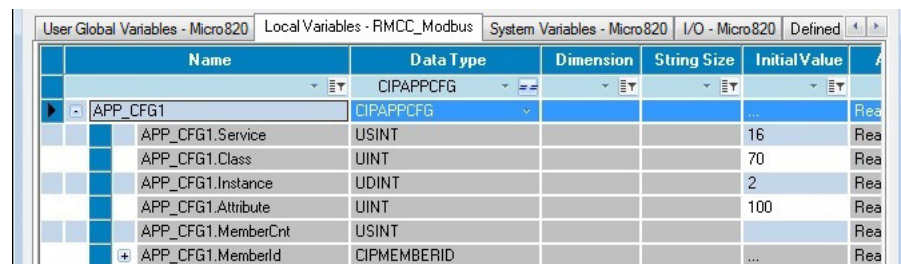
Using Modbus RTU Communication

To use RMCC with the Modbus RTU communication protocol, the serial port must be set to the Modbus slave role. A CIP Generic message is sent from within a program with the following parameters.

CIP Generic Message Parameters for RMCC using Modbus RTU

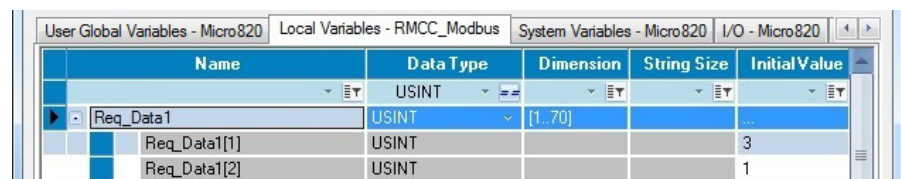
Parameter	Value
Service	16
Class	70
Instance	2 – Embedded serial port 5, 6, 7, 8, or 9 – Plug-in modules
Attribute	100
ReqData	New node address, 1
ReqLen	2

RMCC Modbus Example – Set the Parameters



Name	Data Type	Dimension	String Size	Initial Value	Role
APP_CFG1	CIPAPPCFG			...	Real
APP_CFG1.Service	USINT			16	Real
APP_CFG1.Class	UINT			70	Real
APP_CFG1.Instance	UDINT			2	Real
APP_CFG1.Attribute	UINT			100	Real
APP_CFG1.MemberCnt	USINT				Real
APP_CFG1.MemberId	CIPMEMBERID			...	Real

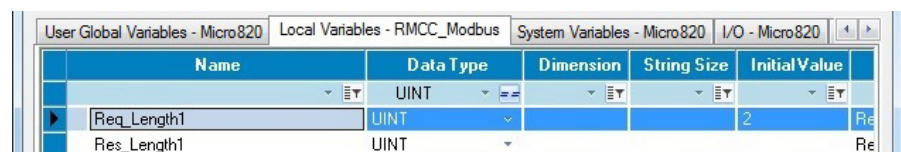
RMCC Modbus Example – Set the New Node Address



Name	Data Type	Dimension	String Size	Initial Value	Role
Req_Data1	USINT	[1..70]		...	Real
Req_Data1[1]	USINT			3	Real
Req_Data1[2]	USINT			1	Real

The first byte indicates the new node address for the controller. For this example, the new node address is “3”. The second byte must always be “1”, this indicates that the Modbus role is configured as Slave.

RMCC Modbus Example – Set the Message Length



Name	Data Type	Dimension	String Size	Initial Value	Role
Req_Length1	UINT			2	Real
Res_Length1	UINT				Real

When the new node address is configured and applied, the port is not restarted.

IMPORTANT You must ensure that the new node address being configured is unique as it will not be checked against existing node addresses of other devices.

You can verify that the node address has changed after performing RMCC by looking at the Communication Diagnostics tab for the controller.

RMCC Modbus Example – Verify Address Change

The screenshot shows the 'Micro820 - Communication Diagnostics' window. At the top, there are tabs for 'Micro820' and 'Micro820 - Communication Diagnostics'. Below the tabs, there are three dropdown menus: 'Communication:' set to 'Serial Port', 'Channel:' set to 'Slot 1 | 2080-SERIALISOL at port 5', and 'Drivers:' set to 'Modbus RTU'. To the right of these dropdowns is a 'Reset Counters' button. Below the dropdowns is a section titled 'Link Counters' which contains a table of communication statistics. At the bottom is a section titled 'Common Settings' which includes a 'Unit Address' field set to '3'.

Link Counters			
Characters Received:	3,088	Characters Sent:	2,464
Frame Received:	386	Frames Sent:	352
Good Transactions:	352	Broadcasts:	0
Good Exceptions:	0	Mismatch Errors:	0
Bad CRC:	0	No Response:	0
		Other Errors:	0

Common Settings	
Unit Address:	3

Using EtherNet/IP Communication

To use RMCC with the EtherNet/IP communication protocol, the controller must be configured to use a static IP address. If the controller is configured to use BOOTP or DHCP, the change will be rejected. A CIP Generic message is sent from within a program with the following parameters.

CIP Generic Message Parameters for RMCC using EtherNet/IP

Parameter	Value
Service	16
Class	245
Instance	1
Attribute	5
ReqData	IP address, Subnet mask, Gateway address
ReqLen	22 bytes

RMCC EtherNet/IP Example – Set the Parameters

The screenshot shows the 'Local Variables' tab for the project 'RMCC_EIP_LengthTest_SimplePro'. The variable 'APP_CFG1' is declared with the data type 'CIPAPPCFG'. Its members are listed in the table below:

Name	Data Type	Dimension	String Size	Initial Value	Attribute
APP_CFG1	CIPAPPCFG				Read/Wr
APP_CFG1.Service	USINT			16	Read/Wr
APP_CFG1.Class	UINT			245	Read/Wr
APP_CFG1.Instance	UDINT			1	Read/Wr
APP_CFG1.Attribute	UINT			5	Read/Wr
APP_CFG1.MemberCnt	USINT				Read/Wr
APP_CFG1.MemberId	CIPMEMBERID				Read/Wr

RMCC EtherNet/IP Example – Set the New IP Address

The screenshot shows the 'Local Variables' tab for the project 'RMCC_EIP_LengthTest_SimplePro'. The variable 'Req_Data1' is declared with the data type 'USINT' and a dimension of '[1..70]'. Its members are listed in the table below:

Name	Data Type	Dimension	String Size	Initial Value
Req_Data1	USINT	[1..70]		
Req_Data1[1]	USINT			10
Req_Data1[2]	USINT			1
Req_Data1[3]	USINT			168
Req_Data1[4]	USINT			192
Req_Data1[5]	USINT			0
Req_Data1[6]	USINT			255
Req_Data1[7]	USINT			255
Req_Data1[8]	USINT			255
Req_Data1[9]	USINT			1
Req_Data1[10]	USINT			1
Req_Data1[11]	USINT			168
Req_Data1[12]	USINT			192
Req_Data1[13]	USINT			0
Req_Data1[14]	USINT			0
Req_Data1[15]	USINT			0
Req_Data1[16]	USINT			0
Req_Data1[17]	USINT			0
Req_Data1[18]	USINT			0
Req_Data1[19]	USINT			0
Req_Data1[20]	USINT			0
Req_Data1[21]	USINT			0

For this example, the new IP Address is set to the following:

- IP address = 192.168.1.10
- Subnet mask = 255.255.255.0
- Gateway address = 192.168.1.1

RMCC EtherNet/IP Example – Set the Message Length

User Global Variables - Micro820					Local Variables - RMCC_EIP_LengthTest_SimplePro					System Variables - Micro820				
Name	Data Type	Dimension	String Size	Initial Value										
Req_Length1	UINT			22										
Res_Length1	UINT													

After the new IP address is configured and applied, the controller will disconnect from Connected Components Workbench if communication is through Ethernet.

IMPORTANT You should not perform IP address changes continuously. Allow an interval of at least six seconds before performing the next IP address change in order for duplicate address detection to work properly.

You can verify that the IP address has changed after performing RMCC by looking at the Ethernet settings for the controller.

RMCC EtherNet/IP Example – Verify Address Change

Controller - Ethernet

Internet Protocol (IP) Settings

☐ Obtain IP address automatically using DHCP

☒ Configure IP address and settings

IP Address: 192 . 168 . 1 . 10

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 192 . 168 . 1 . 1

☒ Detect duplicate IP address

Port Settings

Port State: ☒ Enabled ☐ Disabled

☒ Auto-Negotiate Speed and Duplex Mode

Agency Certifications

- UL Listed Industrial Control Equipment, certified for US and Canada. UL Listed for Class I, Division 2 Group A,B,C,D Hazardous Locations, certified for U.S. and Canada.
- CE marked for all applicable directives
- C-Tick marked for all applicable acts
- KC - Korean Registration of Broadcasting and Communications Equipment, compliant with: Article 58-2 of Radio Waves Act, Clause 3.

Compliance to European Union Directives

This product has the CE mark and is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

EMC Directive

This product is tested to meet Council Directive 2004/108/EC Electromagnetic Compatibility (EMC) and the following standards, in whole or in part, documented in a technical construction file:

- EN 61131-2; Programmable Controllers (Clause 8, Zone A & B)
- EN 61131-2; Programmable Controllers (Clause 11)
- EN 61000-6-4
EMC - Part 6-4: Generic Standards - Emission Standard for Industrial Environments
- EN 61000-6-2
EMC - Part 6-2: Generic Standards - Immunity for Industrial Environments

This product is intended for use in an industrial environment.

Low Voltage Directive

This product is tested to meet Council Directive 2006/95/EC Low Voltage, by applying the safety requirements of EN 61131-2 Programmable Controllers, Part 2 - Equipment Requirements and Tests.

For specific information required by EN 61131-2, see the appropriate sections in this publication, as well as the following Allen-Bradley publications:

- *Industrial Automation Wiring and Grounding Guidelines for Noise Immunity*, publication [1770-4.1](#).
- *Guidelines for Handling Lithium Batteries*, publication AG-5.4
- *Automation Systems Catalog*, publication B115

Installation Considerations

Most applications require installation in an industrial enclosure (Pollution Degree 2⁽¹⁾) to reduce the effects of electrical interference (Over Voltage Category II⁽²⁾) and environmental exposure.

Locate your controller as far as possible from power lines, load lines, and other sources of electrical noise such as hard-contact switches, relays, and AC motor drives. For more information on proper grounding guidelines, see the *Industrial Automation Wiring and Grounding Guidelines* publication [1770-4.1](#).



WARNING: When used in a Class I, Division 2, hazardous location, this equipment must be mounted in a suitable enclosure with proper wiring method that complies with the governing electrical codes.

WARNING: If you connect or disconnect the serial cable with power applied to this module or the serial device on the other end of the cable, an electrical arc can occur. This could cause an explosion in hazardous location installations. Be sure that power is removed or the area is nonhazardous before proceeding.

WARNING: The local programming terminal port is intended for temporary use only and must not be connected or disconnected unless the area is assured to be nonhazardous.

WARNING: Exposure to some chemicals may degrade the sealing properties of materials used in the Relays. It is recommended that the User periodically inspect these devices for any degradation of properties and replace the module if degradation is found.

WARNING: If you insert or remove the plug-in module while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations. Be sure that power is removed or the area is nonhazardous before proceeding.

WARNING: When you connect or disconnect the Removable Terminal Block (RTB) with field side power applied, an electrical arc can occur. This could cause an explosion in hazardous location installations.

WARNING: Be sure that power is removed or the area is nonhazardous before proceeding.



ATTENTION: To comply with the CE Low Voltage Directive (LVD), this equipment must be powered from a source compliant with the following: Safety Extra Low Voltage (SELV) or Protected Extra Low Voltage (PELV).

ATTENTION: To comply with UL restrictions, this equipment must be powered from a Class 2 source.

ATTENTION: Be careful when stripping wires. Wire fragments that fall into the controller could cause damage. Once wiring is complete, make sure the controller is free of all metal fragments.

ATTENTION: Electrostatic discharge can damage semiconductor devices inside the module. Do not touch the connector pins or other sensitive areas.

ATTENTION: The serial cables are not to exceed 3.0 m (9.84 ft).

ATTENTION: Do not wire more than 2 conductors on any single terminal.

ATTENTION: Do not remove the Removable Terminal Block (RTB) until power is removed.

(1) Pollution Degree 2 is an environment where normally only non-conductive pollution occurs except that occasionally temporary conductivity caused by condensation shall be expected.

(2) Overvoltage Category II is the load level section of the electrical distribution system. At this level, transient voltages are controlled and do not exceed the impulse voltage capability of the products insulation.

Environment and Enclosure



This equipment is intended for use in a Pollution Degree 2 industrial environment, in overvoltage Category II applications (as defined in IEC 60664-1), at altitudes up to 2000 m (6562 ft) without derating.

This equipment is considered Group 1, Class A industrial equipment according to IEC/CISPR 11. Without appropriate precautions, there may be difficulties with electromagnetic compatibility in residential and other environments due to conducted and radiated disturbances.

This equipment is supplied as open-type equipment. It must be mounted within an enclosure that is suitably designed for those specific environmental conditions that will be present and appropriately designed to prevent personal injury resulting from accessibility to live parts. The enclosure must have suitable flame-retardant properties to prevent or minimize the spread of flame, complying with a flame spread rating of 5VA, V2, V1, V0 (or equivalent) if non-metallic. The interior of the enclosure must be accessible only by the use of a tool. Subsequent sections of this publication may contain additional information regarding specific enclosure type ratings that are required to comply with certain product safety certifications.

In addition to this publication, see:

- Industrial Automation Wiring and Grounding Guidelines, Rockwell Automation publication [1770-4.1](#), for additional installation requirements.
- NEMA Standard 250 and IEC 60529, as applicable, for explanations of the degrees of protection provided by different types of enclosure.

Preventing Electrostatic Discharge



This equipment is sensitive to electrostatic discharge, which can cause internal damage and affect normal operation. Follow these guidelines when you handle this equipment:

- Touch a grounded object to discharge potential static.
 - Wear an approved grounding wriststrap.
 - Do not touch connectors or pins on component boards.
 - Do not touch circuit components inside the equipment.
 - Use a static-safe workstation, if available.
 - Store the equipment in appropriate static-safe packaging when not in use.
-

Safety Considerations

Safety considerations are an important element of proper system installation. Actively thinking about the safety of yourself and others, as well as the condition of your equipment, is of primary importance. We recommend reviewing the following safety considerations.

North American Hazardous Location Approval

The following information applies when operating this equipment in hazardous locations:	Informations sur l'utilisation de cet équipement en environnements dangereux:
<p>Products marked "CL I, DIV 2, GP A, B, C, D" are suitable for use in Class I Division 2 Groups A, B, C, D, Hazardous Locations and nonhazardous locations only. Each product is supplied with markings on the rating nameplate indicating the hazardous location temperature code. When combining products within a system, the most adverse temperature code (lowest "T" number) may be used to help determine the overall temperature code of the system. Combinations of equipment in your system are subject to investigation by the local Authority Having Jurisdiction at the time of installation.</p>	<p>Les produits marqués "CL I, DIV 2, GP A, B, C, D" ne conviennent qu'à une utilisation en environnements de Classe I Division 2 Groupes A, B, C, D dangereux et non dangereux. Chaque produit est livré avec des marquages sur sa plaque d'identification qui indiquent le code de température pour les environnements dangereux. Lorsque plusieurs produits sont combinés dans un système, le code de température le plus défavorable (code de température le plus faible) peut être utilisé pour déterminer le code de température global du système. Les combinaisons d'équipements dans le système sont sujettes à inspection par les autorités locales qualifiées au moment de l'installation.</p>
<div data-bbox="164 930 259 1016"></div> <p>EXPLOSION HAZARD</p> <ul style="list-style-type: none"> Do not disconnect equipment unless power has been removed or the area is known to be nonhazardous. Do not disconnect connections to this equipment unless power has been removed or the area is known to be nonhazardous. Secure any external connections that mate to this equipment by using screws, sliding latches, threaded connectors, or other means provided with this product. Substitution of any component may impair suitability for Class I, Division 2. If this product contains batteries, they must only be changed in an area known to be nonhazardous. 	<div data-bbox="821 930 917 1016"></div> <p>RISQUE D'EXPLOSION</p> <ul style="list-style-type: none"> Couper le courant ou s'assurer que l'environnement est classé non dangereux avant de débrancher l'équipement. Couper le courant ou s'assurer que l'environnement est classé non dangereux avant de débrancher les connecteurs. Fixer tous les connecteurs externes reliés à cet équipement à l'aide de vis, loquets coulissants, connecteurs filetés ou autres moyens fournis avec ce produit. La substitution de tout composant peut rendre cet équipement inadapté à une utilisation en environnement de Classe I, Division 2. S'assurer que l'environnement est classé non dangereux avant de changer les piles.

Disconnecting Main Power



WARNING: Explosion Hazard

Do not replace components, connect equipment, or disconnect equipment unless power has been switched off.

The main power disconnect switch should be located where operators and maintenance personnel have quick and easy access to it. In addition to disconnecting electrical power, all other sources of power (pneumatic and hydraulic) should be de-energized before working on a machine or process controlled by a controller.

Safety Circuits

**WARNING:** Explosion Hazard

Do not connect or disconnect connectors while circuit is live.

Circuits installed on the machine for safety reasons, like overtravel limit switches, stop push buttons, and interlocks, should always be hard-wired directly to the master control relay. These devices must be wired in series so that when any one device opens, the master control relay is de-energized, thereby removing power to the machine. Never alter these circuits to defeat their function. Serious injury or machine damage could result.

Power Distribution

There are some points about power distribution that you should know:

- The master control relay must be able to inhibit all machine motion by removing power to the machine I/O devices when the relay is de-energized. It is recommended that the controller remain powered even when the master control relay is de-energized.
- If you are using a DC power supply, interrupt the load side rather than the AC line power. This avoids the additional delay of power supply turn-off. The DC power supply should be powered directly from the fused secondary of the transformer. Power to the DC input and output circuits should be connected through a set of master control relay contacts.

Periodic Tests of Master Control Relay Circuit

Any part can fail, including the switches in a master control relay circuit. The failure of one of these switches would most likely cause an open circuit, which would be a safe power-off failure. However, if one of these switches shorts out, it no longer provides any safety protection. These switches should be tested periodically to assure they will stop machine motion when needed.

Power Considerations

The following explains power considerations for the micro controllers.

Isolation Transformers

You may want to use an isolation transformer in the AC line to the controller. This type of transformer provides isolation from your power distribution system to reduce the electrical noise that enters the controller and is often used as a step-down transformer to reduce line voltage. Any transformer used with the controller must have a sufficient power rating for its load. The power rating is expressed in volt-amperes (VA).

Power Supply Inrush

During power-up, the Micro800 power supply allows a brief inrush current to charge internal capacitors. Many power lines and control transformers can supply inrush current for a brief time. If the power source cannot supply this inrush current, the source voltage may sag momentarily.

The only effect of limited inrush current and voltage sag on the Micro800 is that the power supply capacitors charge more slowly. However, the effect of a voltage sag on other equipment should be considered. For example, a deep voltage sag may reset a computer connected to the same power source. The following considerations determine whether the power source must be required to supply high inrush current:

- The power-up sequence of devices in a system.
- The amount of the power source voltage sag if the inrush current cannot be supplied.
- The effect of voltage sag on other equipment in the system.

If the entire system is powered-up at the same time, a brief sag in the power source voltage typically will not affect any equipment.

Loss of Power Source

The optional Micro800 AC power supply is designed to withstand brief power losses without affecting the operation of the system. The time the system is operational during power loss is called program scan hold-up time after loss of power. The duration of the power supply hold-up time depends on power consumption of controller system, but is typically between 10 milliseconds and 3 seconds.

Input States on Power Down

The power supply hold-up time as described above is generally longer than the turn-on and turn-off times of the inputs. Because of this, the input state change from “On” to “Off” that occurs when power is removed may be recorded by the processor before the power supply shuts down the system. Understanding this concept is important. The user program should be written to take this effect into account.

Other Types of Line Conditions

Occasionally the power source to the system can be temporarily interrupted. It is also possible that the voltage level may drop substantially below the normal line voltage range for a period of time. Both of these conditions are considered to be a loss of power for the system.

Preventing Excessive Heat

For most applications, normal convective cooling keeps the controller within the specified operating range. Ensure that the specified temperature range is maintained. Proper spacing of components within an enclosure is usually sufficient for heat dissipation.

In some applications, a substantial amount of heat is produced by other equipment inside or outside the enclosure. In this case, place blower fans inside the enclosure to assist in air circulation and to reduce “hot spots” near the controller.

Additional cooling provisions might be necessary when high ambient temperatures are encountered.

TIP

Do not bring in unfiltered outside air. Place the controller in an enclosure to protect it from a corrosive atmosphere. Harmful contaminants or dirt could cause improper operation or damage to components. In extreme cases, you may need to use air conditioning to protect against heat build-up within the enclosure.

Master Control Relay

A hard-wired master control relay (MCR) provides a reliable means for emergency machine shutdown. Since the master control relay allows the placement of several emergency-stop switches in different locations, its installation is important from a safety standpoint. Overtravel limit switches or mushroom-head push buttons are wired in series so that when any of them opens, the master control relay is de-energized. This removes power to input and output device circuits. Refer to the figures on pages 28 and 29.



WARNING: Never alter these circuits to defeat their function since serious injury and/or machine damage could result.

TIP

If you are using an external DC power supply, interrupt the DC output side rather than the AC line side of the supply to avoid the additional delay of power supply turn-off.

The AC line of the DC output power supply should be fused.

Connect a set of master control relays in series with the DC power supplying the input and output circuits.

Place the main power disconnect switch where operators and maintenance personnel have quick and easy access to it. If you mount a disconnect switch inside the controller enclosure, place the switch operating handle on the outside of the enclosure, so that you can disconnect power without opening the enclosure.

Whenever any of the emergency-stop switches are opened, power to input and output devices should be removed.

When you use the master control relay to remove power from the external I/O circuits, power continues to be provided to the controller's power supply so that diagnostic indicators on the processor can still be observed.

The master control relay is not a substitute for a disconnect to the controller. It is intended for any situation where the operator must quickly de-energize I/O devices only. When inspecting or installing terminal connections, replacing output fuses, or working on equipment within the enclosure, use the disconnect to shut off power to the rest of the system.

TIP

Do not control the master control relay with the controller. Provide the operator with the safety of a direct connection between an emergency-stop switch and the master control relay.

Using Emergency-Stop Switches

When using emergency-stop switches, adhere to the following points:

- Do not program emergency-stop switches in the controller program. Any emergency-stop switch should turn off all machine power by turning off the master control relay.
- Observe all applicable local codes concerning the placement and labeling of emergency-stop switches.

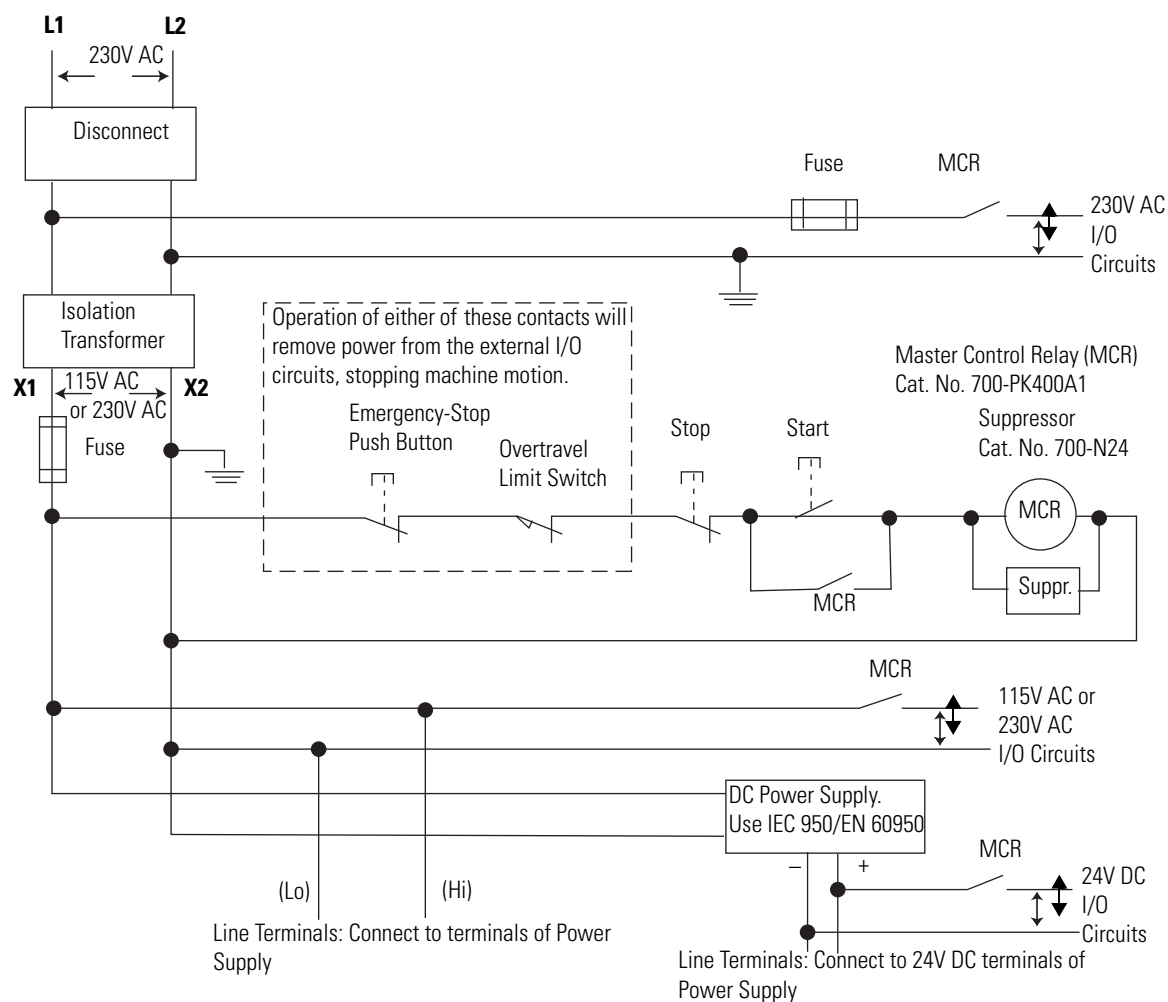
- Install emergency-stop switches and the master control relay in your system. Make certain that relay contacts have a sufficient rating for your application. Emergency-stop switches must be easy to reach.
- In the following illustration, input and output circuits are shown with MCR protection. However, in most applications, only output circuits require MCR protection.

The following illustrations show the Master Control Relay wired in a grounded system.

TIP

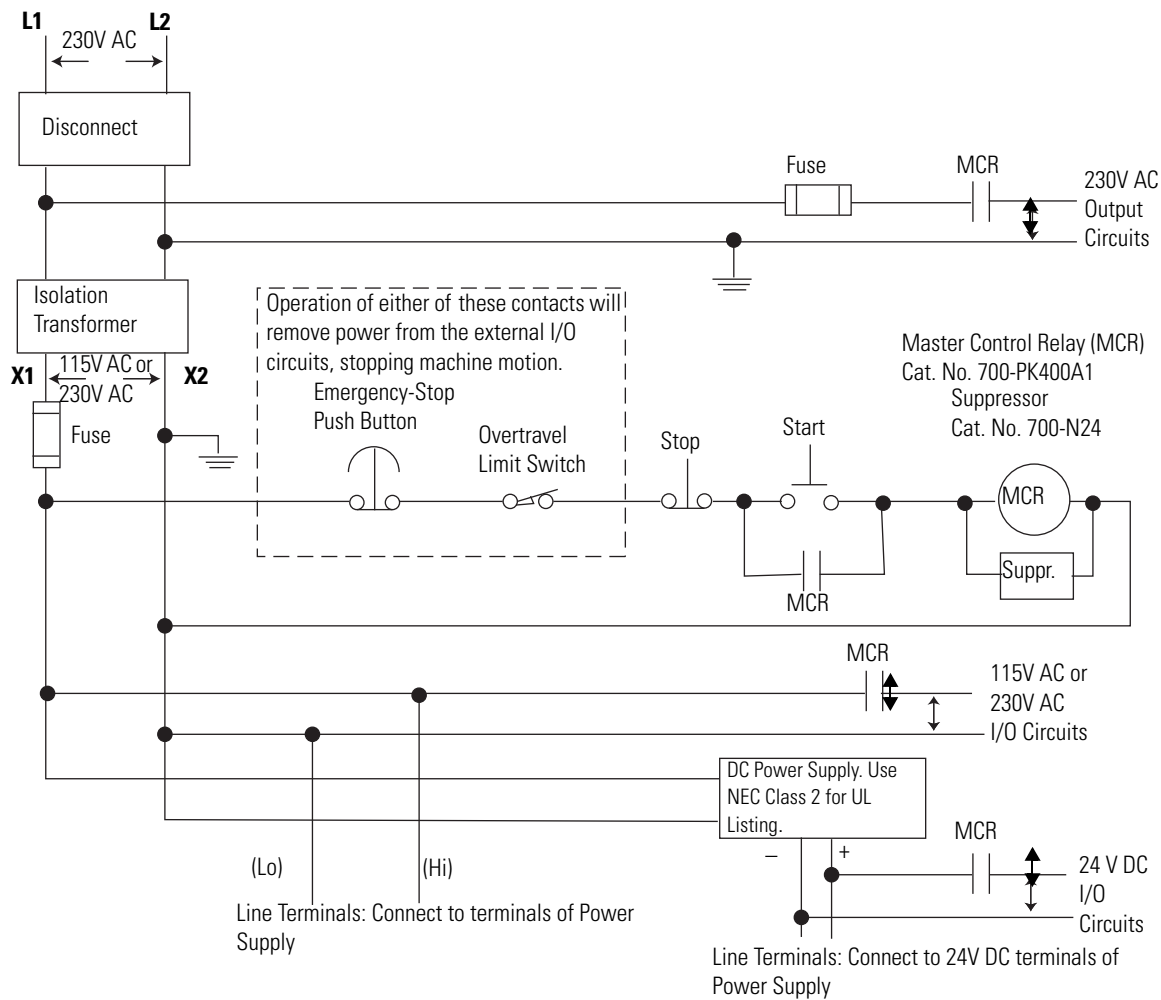
In most applications input circuits do not require MCR protection; however, if you need to remove power from all field devices, you must include MCR contacts in series with input power wiring.

Schematic (Using IEC Symbols)



44564

Schematic (Using ANSI/CSA Symbols)



44565

Notes:

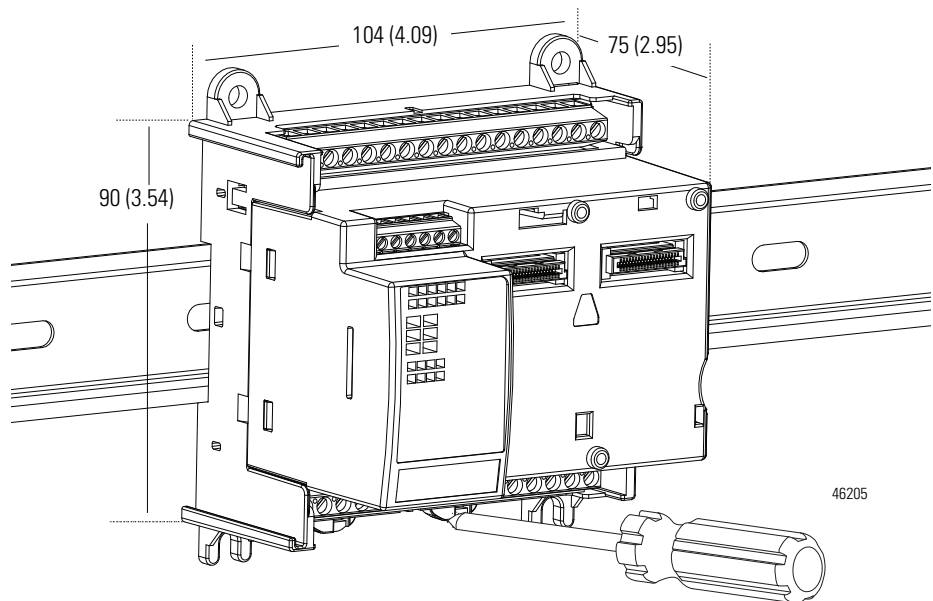
Install Your Controller

This chapter serves to guide the user on installing the controller. It includes the following topics.

Topic	Page
Controller Mounting Dimensions	31
Connect the Controller to an EtherNet/IP Network	33
Module Spacing	32
DIN Rail Mounting	32
Panel Mounting	32
Panel Mounting Dimensions	33
Install the microSD Card	34
Install the 2080-REMLCD Module	35

Controller Mounting Dimensions

Mounting dimensions do not include mounting feet or DIN rail latches.



Measurements in millimeters (inches)

Module Spacing

Maintain spacing from enclosure walls, wireways, and adjacent equipment. Allow 50.8 mm (2 in.) of space on all sides. This provides ventilation and electrical isolation. If optional accessories/modules are attached to the controller, such as the power supply 2080-PS120-240VAC or expansion I/O modules, make sure that there is 50.8 mm (2 in.) of space on all sides after attaching the optional parts.

DIN Rail Mounting

The module can be mounted using the following DIN rails: 35 x 7.5 x 1 mm and 35 x 15 mm (EN 50 022 - 35 x 7.5 and EN 50 022 - 35 x 15).

TIP For environments with greater vibration and shock concerns, use the panel mounting method, instead of DIN rail mounting.

Before mounting the module on a DIN rail, use a flat-blade screwdriver in the DIN rail latch and pry it downwards until it is in the unlatched position.

1. Hook the top of the DIN rail mounting area of the controller onto the DIN rail, and then press the bottom until the controller snaps onto the DIN rail.
2. Push the DIN rail latch back into the latched position.
Use DIN rail end anchors (Allen-Bradley part number 1492-EAJ35 or 1492-EAHJ35) for vibration or shock environments.

To remove your controller from the DIN rail, pry the DIN rail latch downwards until it is in the unlatched position.

Panel Mounting

The preferred mounting method is to use four M4 (#8) screws per module. Hole spacing tolerance: ± 0.4 mm (0.016 in.).

Follow these steps to install your controller using mounting screws.

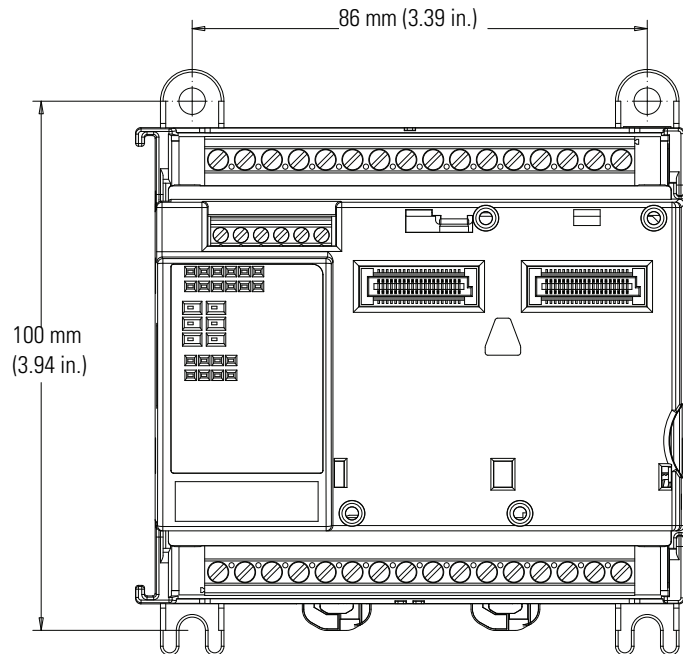
1. Place the controller against the panel where you are mounting it. Make sure the controller is spaced properly.
2. Mark drilling holes through the mounting screw holes and mounting feet then remove the controller.
3. Drill the holes at the markings, then replace the controller and mount it. Leave the protective debris strip in place until you are finished wiring the controller and any other devices.

Panel Mounting Dimensions

Micro820 20-point controllers

2080-LC20-20AWB, 2080-LC20-20QWB, 2080-LC20-20QBB

2080-LC20-20AWBR, 2080-LC20-20QWBR, 2080-LC20-20QBRR



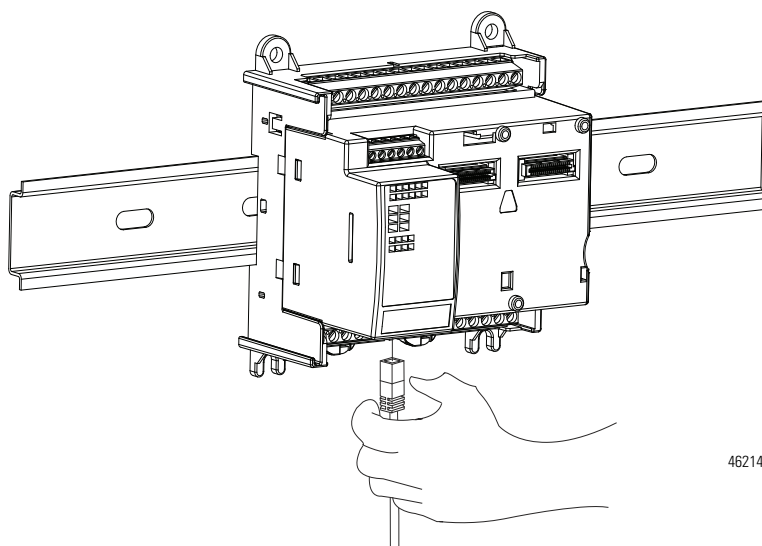
Connect the Controller to an EtherNet/IP Network



WARNING: If you connect or disconnect the communications cable with power applied to this module or any device on the network, an electrical arc can occur. This could cause an explosion in hazardous location installations.

Be sure that power is removed or the area is nonhazardous before proceeding.

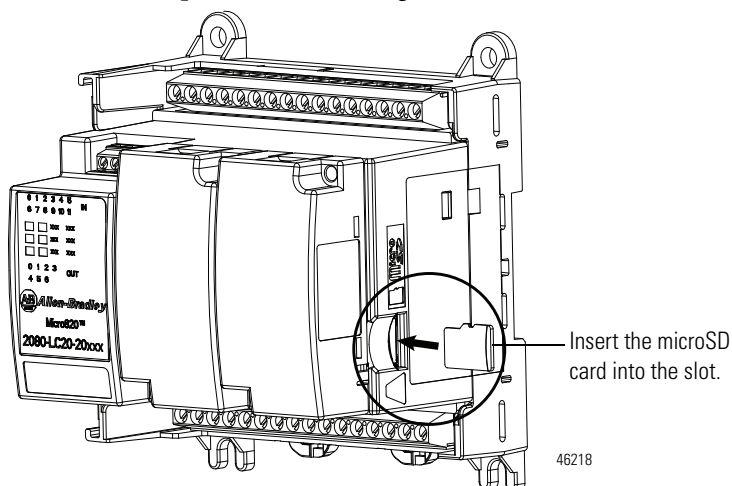
Connect the RJ-45 connector of the Ethernet cable to the Ethernet port on the controller. The port is on the bottom of the controller.



46214

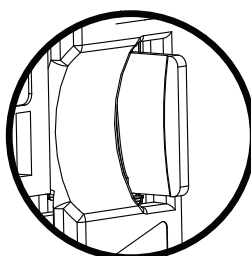
Install the microSD Card

1. Insert the microSD card into the card slot.
You can install the microSD card in one orientation only. The beveled corner should be at the bottom. If you feel resistance when inserting the microSD card, pull it out and change the orientation.



46218

2. Gently press the card until it clicks into place.



46219

3. To remove the microSD card from the slot, gently press the card until it clicks back and releases itself from the slot.

Install the 2080-REMLCD Module

The Micro820 controller supports the 2080-REMLCD module, a simple text display interface for configuring settings such as IP address. It can be mounted through a front panel or on the same DIN rail as the controller.

For information on how the Remote LCD interfaces with the Micro820 controller, see [Using the Micro800 Remote LCD on page 83](#).

To learn about installation, hardware features, and specifications of the 2080-REMLCD module, refer to the Installation Instructions, publication [2080-IN010](#) in the Literature Library.

Notes:

Wire Your Controller

This chapter provides information on the Micro820 controller wiring requirements. It includes the following sections:

Topic	Page
Wiring Requirements and Recommendation	37
Use Surge Suppressors	38
Recommended Surge Suppressors	40
Grounding the Controller	41
Wiring Diagrams	41
Controller I/O Wiring	48
Minimize Electrical Noise	48
Analog Channel Wiring Guidelines	48
Minimize Electrical Noise on Analog Channels	49
Grounding Your Analog Cable	49
Wiring Examples	50

Wiring Requirements and Recommendation



WARNING: Before you install and wire any device, disconnect power to the controller system.



WARNING: Calculate the maximum possible current in each power and common wire. Observe all electrical codes dictating the maximum current allowable for each wire size. Current above the maximum ratings may cause wiring to overheat, which can cause damage.
United States Only: If the controller is installed within a potentially hazardous environment, all wiring must comply with the requirements stated in the National Electrical Code 501-10 (b).

- Allow for at least 50 mm (2 in.) between I/O wiring ducts or terminal strips and the controller.
- Route incoming power to the controller by a path separate from the device wiring. Where paths must cross, their intersection should be perpendicular.

TIP

Do not run signal or communications wiring and power wiring in the same conduit. Wires with different signal characteristics should be routed by separate paths.

- Separate wiring by signal type. Bundle wiring with similar electrical characteristics together.
- Separate input wiring from output wiring.
- Label wiring to all devices in the system. Use tape, shrink-tubing, or other dependable means for labeling purposes. In addition to labeling, use colored insulation to identify wiring based on signal characteristics. For example, you may use blue for DC wiring and red for AC wiring.

Wire Requirements

Wire Requirements for fixed terminal blocks

	Min	Max	
Solid	0.14 mm ² (26 AWG)	2.5 mm ² (14 AWG)	rated @ 90 °C (194 °F) insulation max
Stranded	0.14 mm ² (26 AWG)	1.5 mm ² (16 AWG)	

Wire requirements for removable terminal blocks

	Min	Max	
Solid and Stranded	0.2 mm ² (24 AWG)	2.5 mm ² (14 AWG)	rated @ 90 °C (194 °F) insulation max

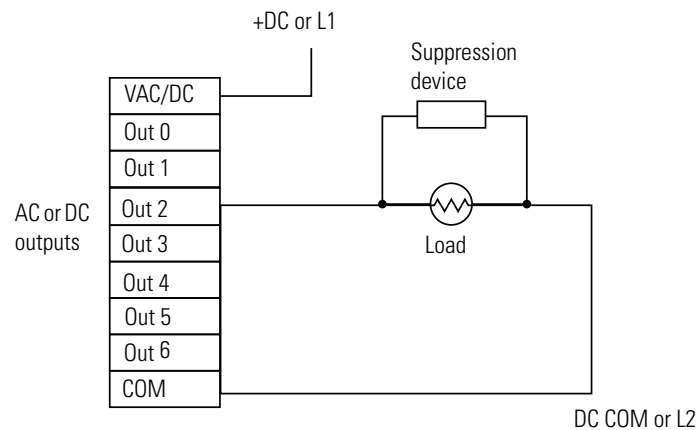
Wire requirements for RS232/RS485 serial port terminal block

	Min	Max	
Solid	0.14 mm ² (26 AWG)	1.5 mm ² (16 AWG)	rated @ 90 °C (194 °F) insulation max
Stranded	0.14 mm ² (26 AWG)	1.0 mm ² (18 AWG)	

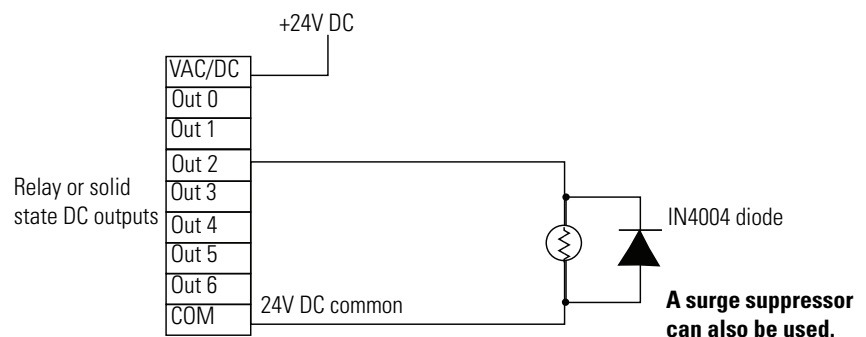
Use Surge Suppressors

Because of the potentially high current surges that occur when switching inductive load devices, such as motor starters and solenoids, the use of some type of surge suppression to protect and extend the operating life of the controllers output contacts is required. Switching inductive loads without surge suppression can *significantly* reduce the life expectancy of relay contacts. By adding a suppression device directly across the coil of an inductive device, you prolong the life of the output or relay contacts. You also reduce the effects of voltage transients and electrical noise from radiating into adjacent systems.

The following diagram shows an output with a suppression device. We recommend that you locate the suppression device as close as possible to the load device.



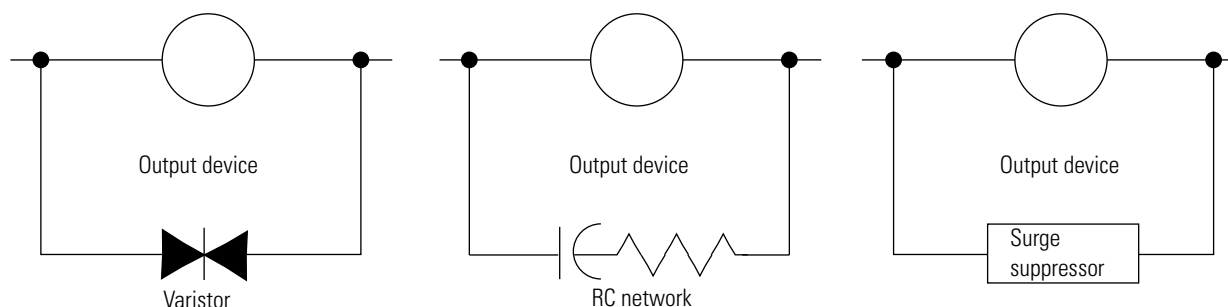
If the outputs are DC, we recommend that you use an 1N4004 diode for surge suppression, as shown below. For inductive DC load devices, a diode is suitable. A 1N4004 diode is acceptable for most applications. A surge suppressor can also be used. See [Recommended Surge Suppressors on page 40](#). As shown below, these surge suppression circuits connect directly across the load device.



Suitable surge suppression methods for inductive AC load devices include a varistor, an RC network, or an Allen-Bradley surge suppressor, all shown below. These components must be appropriately rated to suppress the switching

transient characteristic of the particular inductive device. See [Recommended Surge Suppressors on page 40](#) for recommended suppressors.

Surge Suppression for Inductive AC Load Devices



Recommended Surge Suppressors

Use the Allen-Bradley surge suppressors in the following table for use with relays, contactors, and starters.

Recommended Surge Suppressors

Device	Coil Voltage	Suppressor Catalog Number	Type ⁽⁴⁾
Bulletin 100/104K 700K	24...48V AC	100-KFSC50	RC
	110...280V AC	100-KFSC280	
	380...480V AC	100-KFSC480	
	12...55 V AC, 12...77V DC	100-KFSV55	MOV
	56...136 VAC, 78...180V DC	100-KFSV136	
	137...277V AC, 181...250 V DC	100-KFSV277	
	12...250V DC	100-KFSD250	Diode
Bulletin 100C, (C09 - C97)	24...48V AC	100-FSC48 ⁽¹⁾	RC
	110...280V AC	100-FSC280 ⁽¹⁾	
	380...480V AC	100-FSC480 ⁽¹⁾	
	12...55V AC, 12...77V DC	100-FSV55 ⁽¹⁾	MOV
	56...136V AC, 78...180V DC	100-FSV136 ⁽¹⁾	
	137...277V AC, 181...250V DC	100-FSV277 ⁽¹⁾	
	278...575V AC	100-FSV575 ⁽¹⁾	
	12...250V DC	100-FSD250 ⁽¹⁾	Diode
Bulletin 509 Motor Starter Size 0 - 5	12...120V AC	599-K04	MOV
	240...264V AC	599-KA04	

Recommended Surge Suppressors

Device	Coil Voltage	Suppressor Catalog Number	Type ⁽⁴⁾
Bulletin 509 Motor Starter Size 6	12...120V AC	199-FSMA1 ⁽²⁾	RC
	12...120V AC	199-GSMA1 ⁽³⁾	MOV
Bulletin 700 R/RM Relay	AC coil	Not Required	
	24...48V DC	199-FSMA9	MOV
	50...120V DC	199-FSMA10	
	130...250V DC	199-FSMA11	
Bulletin 700 Type N, P, PK or PH Relay	6...150V AC/DC	700-N24	RC
	24...48V AC/DC	199-FSMA9	MOV
	50...120V AC/DC	199-FSMA10	
	130...250V AC/DC	199-FSMA11	
	6...300V DC	199-FSMZ-1	Diode
Miscellaneous electromagnetic devices limited to 35 sealed VA	6...150V AC/DC	700-N24	RC

(1) Catalog numbers for screwless terminals include the string 'CR' after '100-'. For example: Cat. No. 100-FSC48 becomes Cat. No. 100-**CR**FSC48; Cat. No. 100-FSV55 becomes 100-**CR**FSV55; and so on.

(2) For use on the interposing relay.

(3) For use on the contactor or starter.

(4) RC Type not to be used with Triac outputs. Varistor is not recommended for use on the relay outputs.


Grounding the Controller

WARNING: All devices connected to the RS232/RS485 communication port must be referenced to controller ground, or be floating (not referenced to a potential other than ground). Failure to follow this procedure may result in property damage or personal injury.

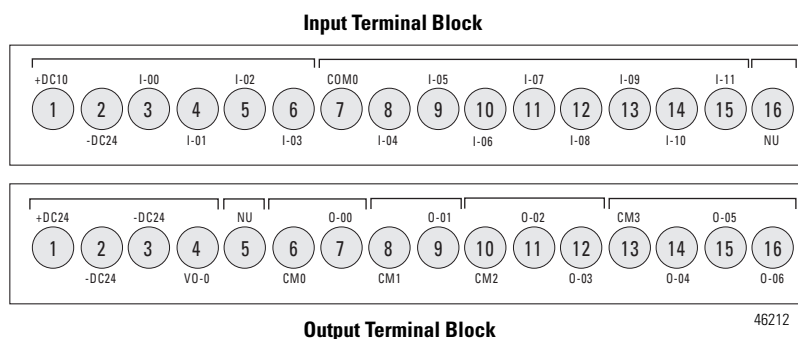
This product is intended to be mounted to a well grounded mounting surface such as a metal panel. Refer to the Industrial Automation Wiring and Grounding Guidelines, publication [1770-4.1](#), for additional information.

Wiring Diagrams

The following illustrations show the wiring diagrams for the Micro800 controllers. Controllers with DC inputs can be wired as either sinking or sourcing inputs. Sinking and sourcing does not apply to AC inputs.

High-speed inputs and outputs are indicated by .

*2080-LC20-20AWB, 2080-LC20-20QWB,
2080-LC20-20AWBR, 2080-LC20-20QWBR*

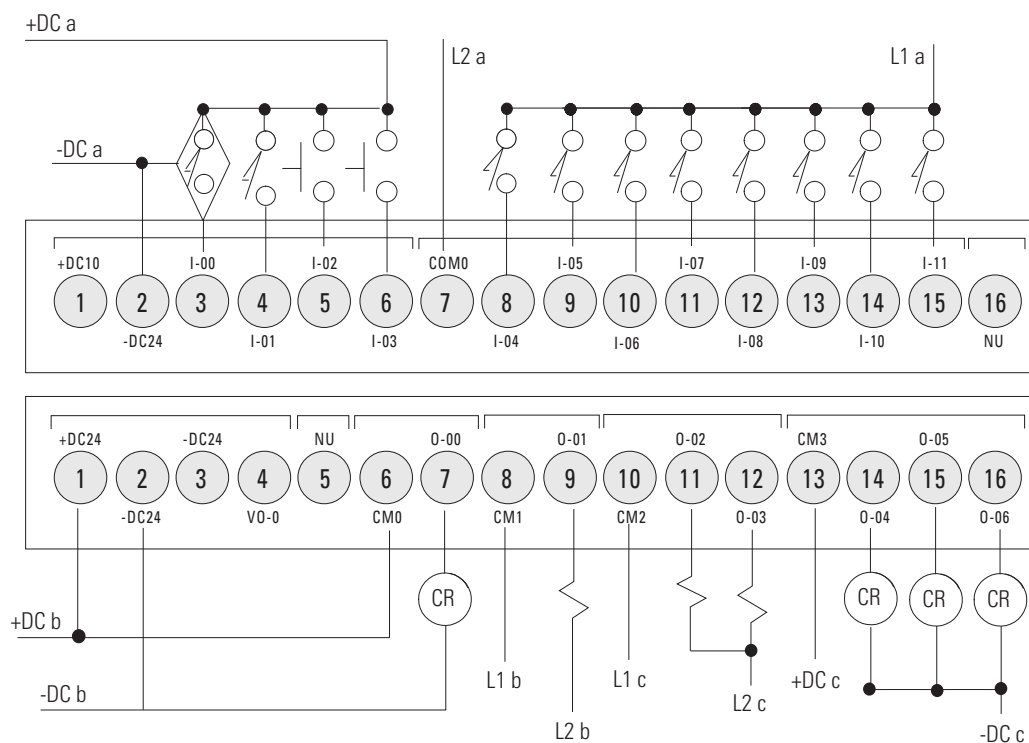


IMPORTANT Note the following:

- The “-DC24” terminals on the Input (terminal 2) and Output (terminals 2 and 3) Terminal Blocks are internally shorted.
- “NU” means that the terminal is not used / no connection.
- Inputs I-00, I-01, I-02, and I-03 are shared between digital and analog inputs.
- Inputs I-00, I-01, I-02, and I-03 can only be used in sinking input configuration.

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

Sinking Input Configuration





ATTENTION: For 2080-LC20-20AWB/R catalogs, inputs 00...03 are limited to 24V DC. All other inputs (04...11) are limited to 120V AC.

Digital Input

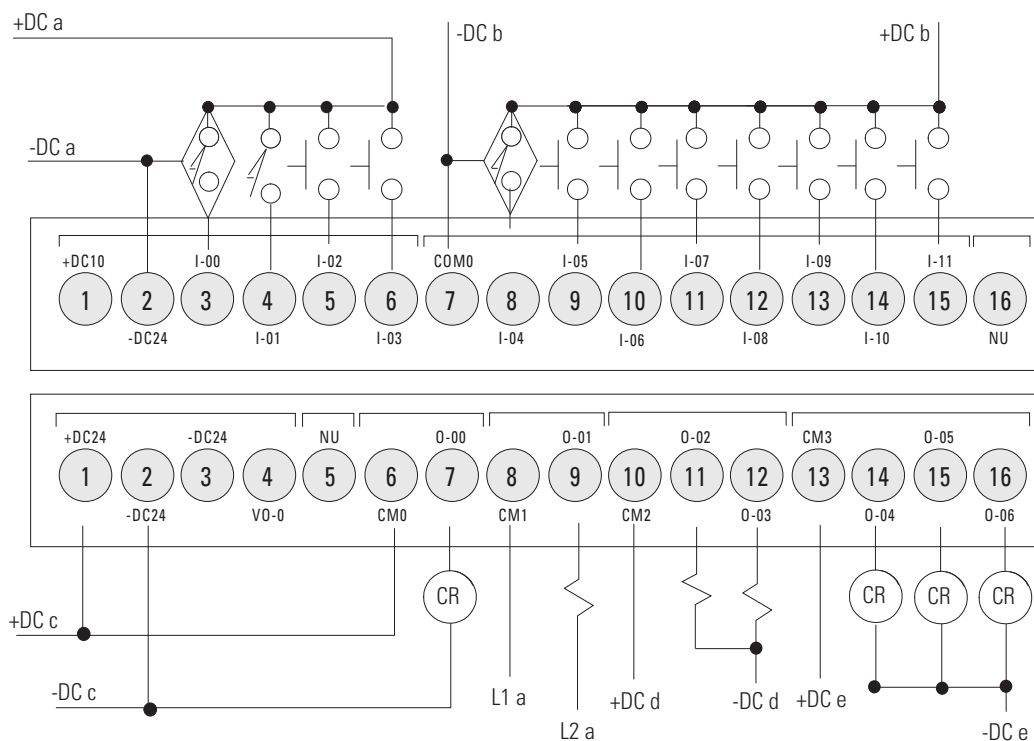
Controller	Terminal Number	Input Common Terminal Label	Terminal Number	Input Terminal Label
2080-LC20-20AWB, 2080-LC20-20AWBR	2	“-DC24” (24V DC sink only)	3	I-00
			4	I-01
			5	I-02
			6	I-03
	7	CM0 (120V AC)	8	I-04
			9	I-05
			10	I-06
			11	I-07
			12	I-08
			13	I-09
			14	I-10
			15	I-11

Digital Output

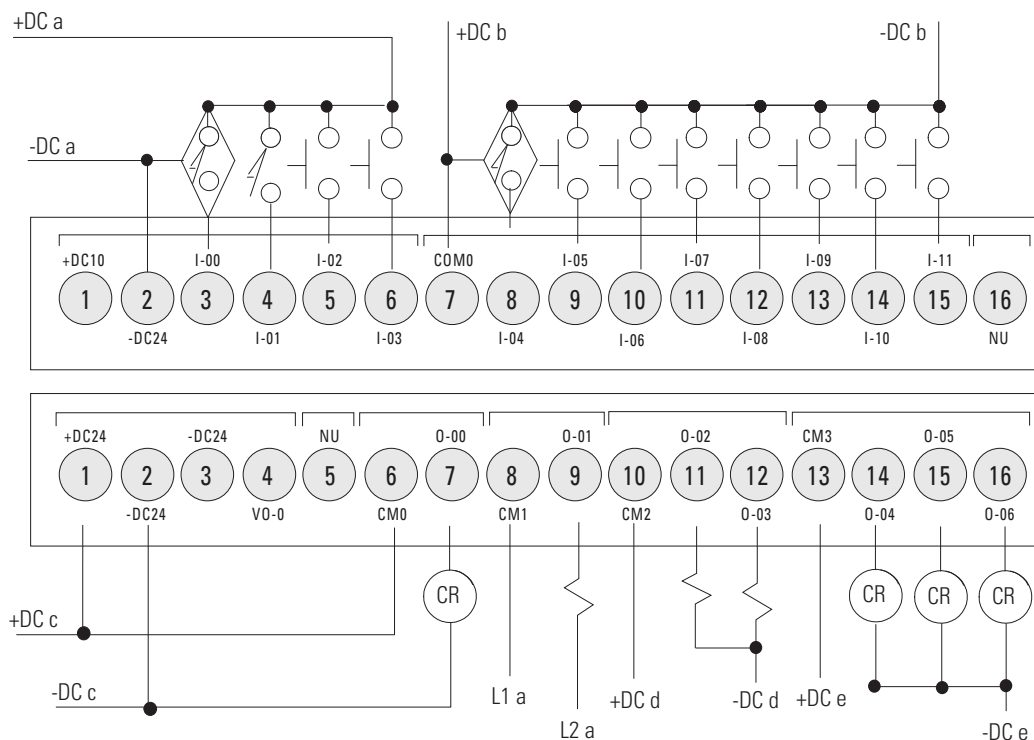
Controller	Terminal Number	Input Common Terminal Label	Terminal Number	Input Terminal Label
2080-LC20-20AWB, 2080-LC20-20AWBR	6	CM0 (VAC/DC)	7	O-00
	8	CM1 (VAC/DC)	9	O-01
	10	CM2 (VAC/DC)	11, 12	O-02, O-03
	13	CM3 (VAC/DC)	14, 15, 16	O-04, O-05, O-06

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

DC Sinking Input Configuration – Inputs 00...11



DC Sourcing Input Configuration – Inputs 4...11



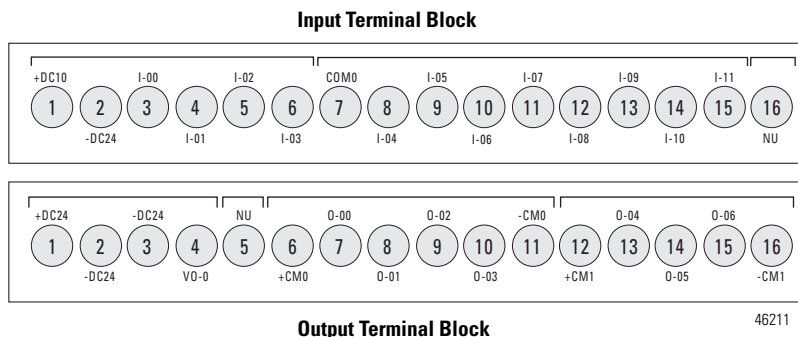
Digital Input

Controller	Terminal Number	Input Common Terminal Label	Terminal Number	Input Terminal Label
2080-LC20-20QWB, 2080-LC20-20QWBR	2	“-DC24” (24V DC sink only)	3	I-00
			4	I-01
			5	I-02
			6	I-03
	7	CM0 (24V DC sink/source)	8	I-04
			9	I-05
			10	I-06
			11	I-07
			12	I-08
			13	I-09
			14	I-10
			15	I-11

Digital Output

Controller	Terminal Number	Input Common Terminal Label	Terminal Number	Input Terminal Label
2080-LC20-20QWB, 2080-LC20-20QWBR	6	CM0 (VAC/DC)	7	O-00
	8	CM1 (VAC/DC)	9	O-01
	10	CM2 (VAC/DC)	11, 12	O-02, O-03
	13	CM3 (VAC/DC)	14, 15, 16	O-04, O-05, O-06

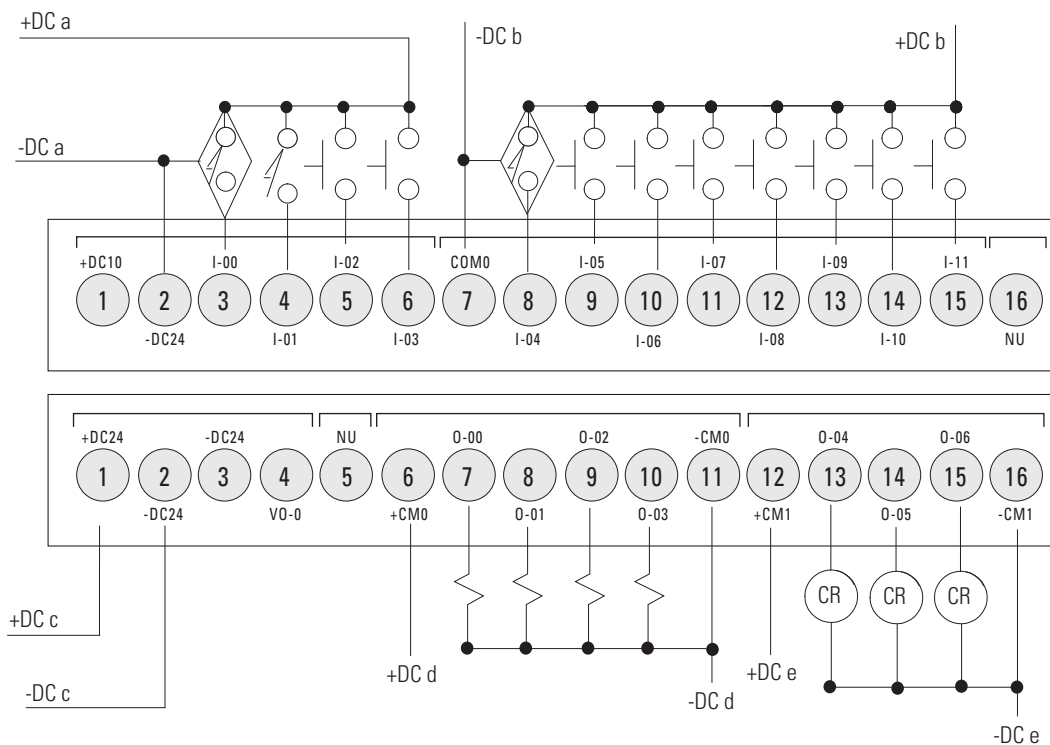
2080-LC20-20QBB / 2080-LC20-20QBBR

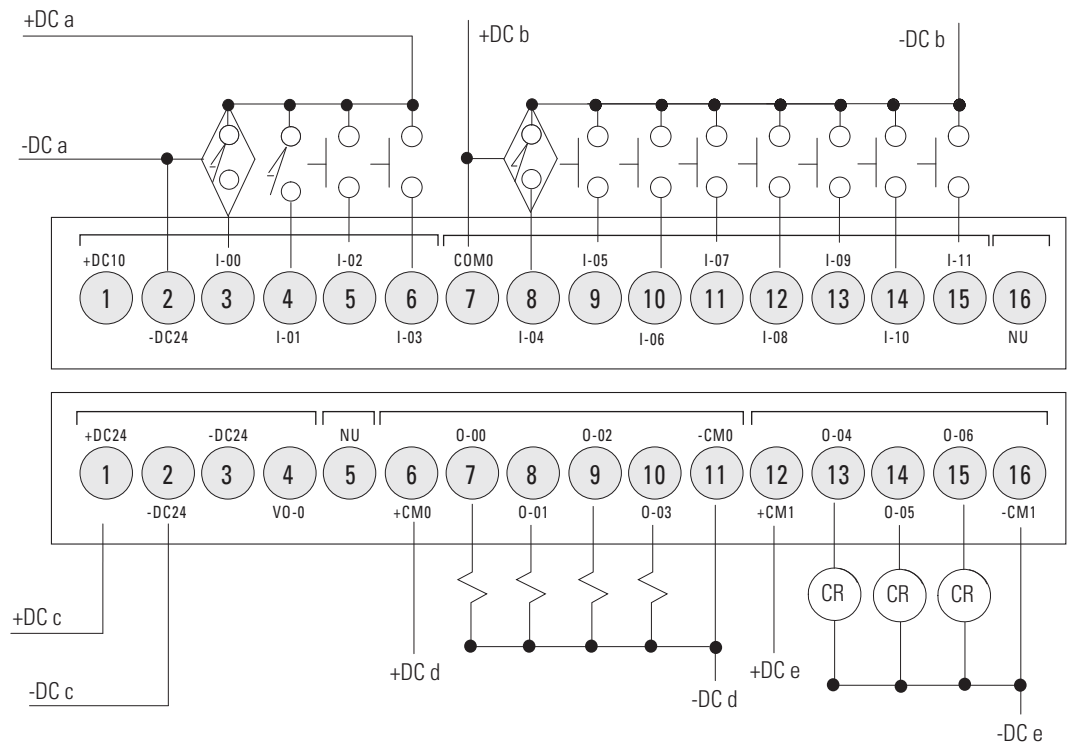


IMPORTANT Note the following:

- The “-DC24” terminals on the Input (terminal 2) and Output (terminals 2 and 3) Terminal Blocks are internally shorted.
- “NU” means that the terminal is not used / no connection.
- Inputs I-00, I-01, I-02, and I-03 are shared between digital and analog inputs.
- Inputs I-00, I-01, I-02, and I-03 can only be used in sinking input configuration.

DC Sinking Input Configuration – Inputs 00...11



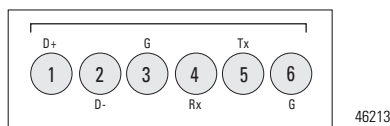
DC Sourcing Input Configuration – Inputs 4...11**Digital Input**

Controller	Terminal Number	Input Common Terminal Label	Terminal Number	Input Terminal Label
2080-LC20-20QBB, 2080-LC20-20QBRR	2	"-DC24" (24V DC sink only)	3	I-00
			4	I-01
			5	I-02
			6	I-03
	7	CM0 (24V DC sink/source)	8	I-04
			9	I-05
			10	I-06
			11	I-07
			12	I-08
			13	I-09
			14	I-10
			15	I-11

Digital Output

Controller	Terminal Number	Input Common Terminal Label	Terminal Number	Input Terminal Label
2080-LC20-20QBB, 2080-LC20-20QBRR	6	+CM0 (VDC source)	7, 8, 9, 10	O-00, O-01, O-02, O-03
	11	-CM0		
	12	+CM1 (VDC source)	13, 14, 15	O-04, O-05, O-06
	16	-CM1		

Serial Port Terminal Block



(View into terminal block)

Pin 1 RS485 Data +
 Pin 2 RS485 Data -
 Pin 3 RS485 Ground⁽¹⁾
 Pin 4 RS232 Receive
 Pin 5 RS232 Transmit
 Pin 6 RS232 Ground⁽¹⁾

(1) Non-isolated.

Controller I/O Wiring

This section contains some relevant information about minimizing electrical noise and also includes some wiring examples.

Minimize Electrical Noise

Because of the variety of applications and environments where controllers are installed and operating, it is impossible to ensure that all environmental noise will be removed by input filters. To help reduce the effects of environmental noise, install the Micro800 system in a properly rated (for example, NEMA) enclosure. Make sure that the Micro800 system is properly grounded.

A system may malfunction due to a change in the operating environment after a period of time. We recommend periodically checking system operation, particularly when new machinery or other noise sources are installed near the Micro800 system.

Analog Channel Wiring Guidelines

Consider the following when wiring your analog channels:

- The analog common (-DC24) is not electrically isolated from the system, and is connected to the power supply common.
- Analog channels are not isolated from each other.
- Use Belden cable #8761, or equivalent, shielded wire.
- Under normal conditions, the drain wire (shield) should be connected to the metal mounting panel (earth ground). Keep the shield connection to earth ground as short as possible.
- To ensure optimum accuracy for voltage type inputs, limit overall cable impedance by keeping all analog cables as short as possible. Locate the I/O system as close to your voltage type sensors or actuators as possible.

Minimize Electrical Noise on Analog Channels

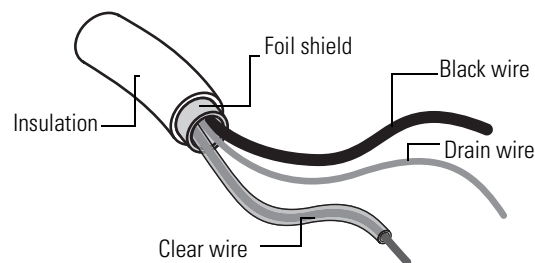
Inputs on analog channels employ digital high-frequency filters that significantly reduce the effects of electrical noise on input signals. However, because of the variety of applications and environments where analog controllers are installed and operated, it is impossible to ensure that all environmental noise will be removed by the input filters.

Several specific steps can be taken to help reduce the effects of environmental noise on analog signals:

- install the Micro800 system in a properly rated enclosure, for example, NEMA/IP. Make sure that the shield is properly grounded.
- use Belden cable #8761 for wiring the analog channels, making sure that the drain wire and foil shield are properly earth grounded.
- route the Belden cable separately from any AC wiring. Additional noise immunity can be obtained by routing the cables in grounded conduit.

Grounding Your Analog Cable

Use shielded communication cable (Belden #8761). The Belden cable has two signal wires (black and clear), one drain wire, and a foil shield. The drain wire and foil shield must be grounded at one end of the cable.



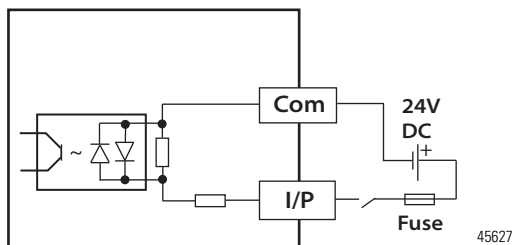
44531

IMPORTANT Do not ground the drain wire and foil shield at both ends of the cable.

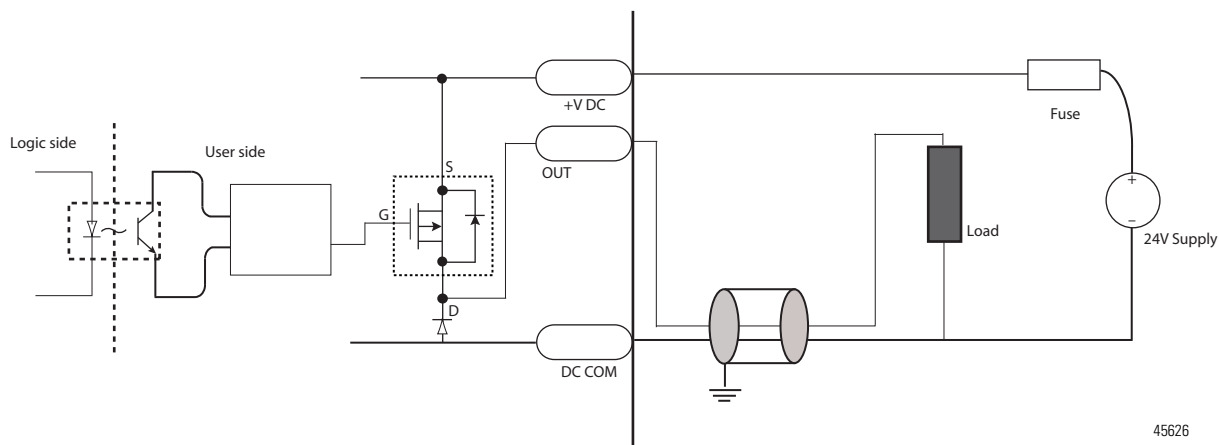
Wiring Examples

Examples of sink/source, input/output wiring are shown below.

Sink Input Wiring Example

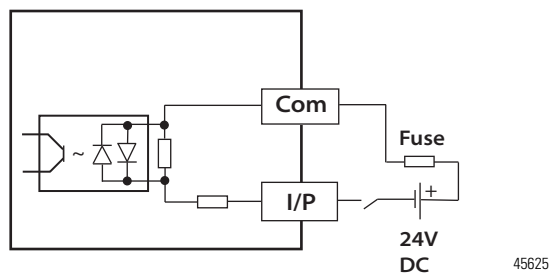


Source Output Wiring Example



IMPORTANT For 2080-LC20-20QBB(R) discrete output 06, shielded cable is required if the output is used as PWM. Otherwise, unshielded cable can be used.

Source Input Wiring Example



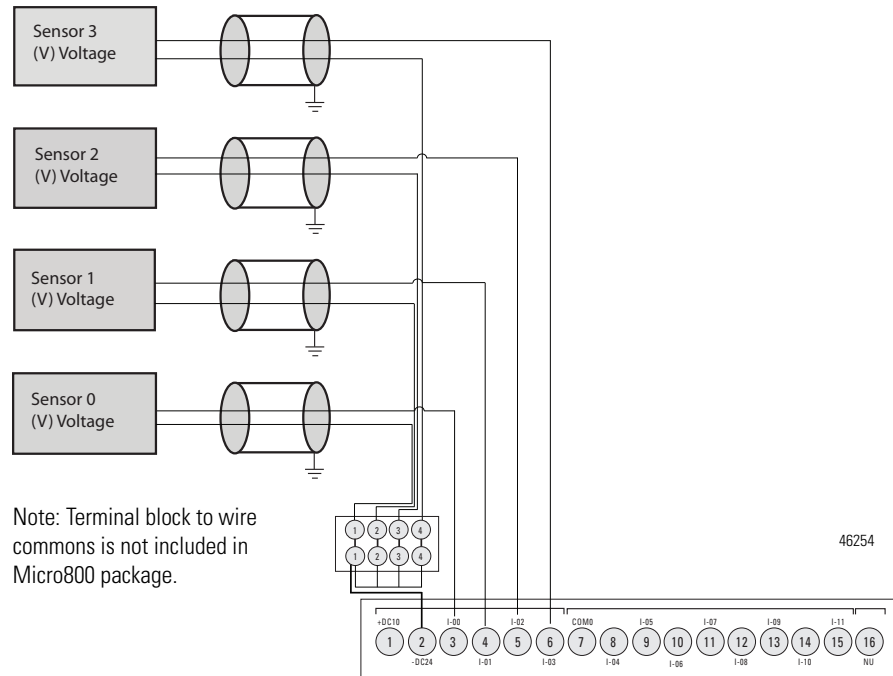
Wiring Analog Channels

Analog input circuits can monitor voltage signals and convert them to serial digital data as shown in the following illustration.



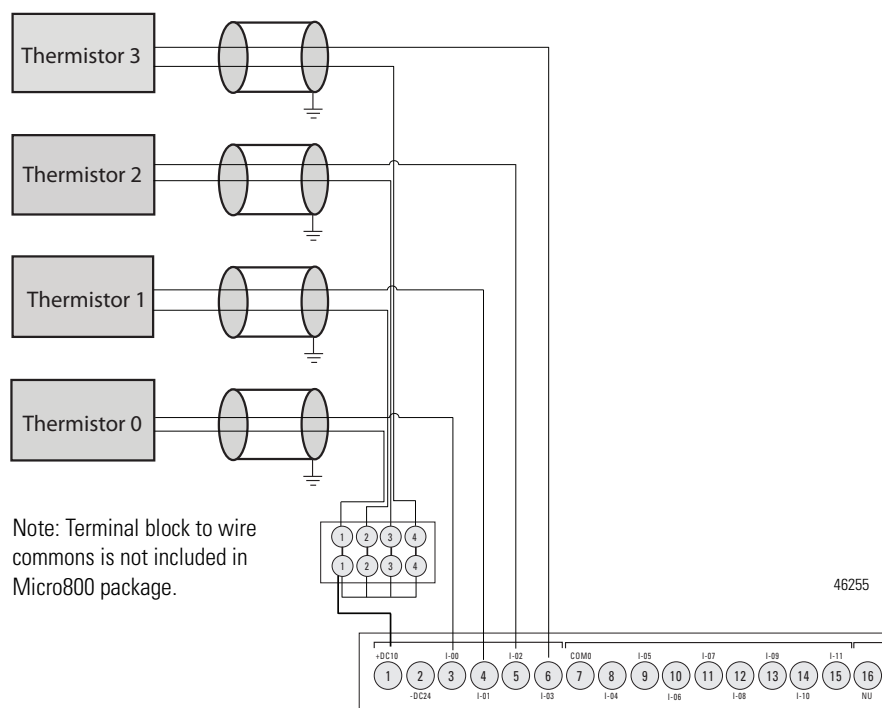
ATTENTION: Analog inputs and outputs are not isolated.

Analog input to sensors



The “-DC24” terminal is the analog ground connection for analog inputs (I-00 to I-03).

Analog input to thermistors



The “+DC10” terminal supplies 10V DC power source to the Thermistor inputs (I-00 to I-03).

Calculate for Thermistor Resistance

While connecting Analog input to thermistor as shown in previous diagram, calculate input voltage using the following equation:

$$V_i = \frac{R_i}{R_i + R_t} * V_{ref}$$

To calculate for thermistor resistance, use the following equation:

$$R_t = \frac{R_i V_{ref} - V_i R_i}{V_i}$$

Where:

V_i = Voltage input ($\pm 5\%$ without calibration; $\pm 2\%$ with calibration)

R_i = Resistance input ($14.14 \text{ K}\Omega \pm 2\%$)

R_t = Thermistor resistance ($10 \text{ K}\Omega$ Thermistor is recommended)

$V_{ref} = 10\text{V} \pm 0.5\text{V}$

IMPORTANT Micro820 controllers support $10 \text{ K}\Omega$ type thermistors.
In order to get the best results, the system must be calibrated.

To convert this resistance into a temperature, use the following Steinhart-Hart equation:

$$\frac{1}{T} = a + b \ln(R) + c [\ln(R)]^3$$

Where:

Coefficients a, b, and c are provided by the thermistor manufacturer.

Calibrate Thermistor

1. Connect a resistor (10 K Ω is recommended) across Vref and Analog Input 00 of your Micro820 controller following the diagram, [Analog input to thermistors on page 52](#). The resistor is measured as Ri using a precision multimeter.
2. Calculate the ideal counts (C1) for resistor (Ri) following this equation:

$$C1 = 14.14 \text{ K}\Omega / (14.14 \text{ K}\Omega + Ri) * 4095$$
3. Read the actual counts (C2) of Analog Input 00 from Connected Components Workbench.
4. Calculate for calibration Gain.

$$\text{Gain} = C1/C2$$

For example:

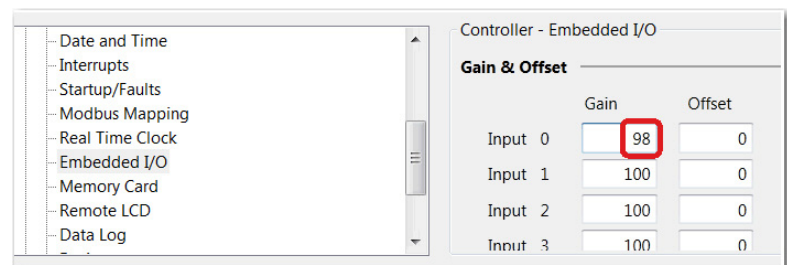
If Ri is measured as 10.00 K Ω , then

$C1 = 14.14 / (14.14 + 10.00) * 4095 = 2399$ counts;

C2 is read from Connected Components Workbench as 2440; so

$\text{Gain} = 2399/2440 = 98\%$.

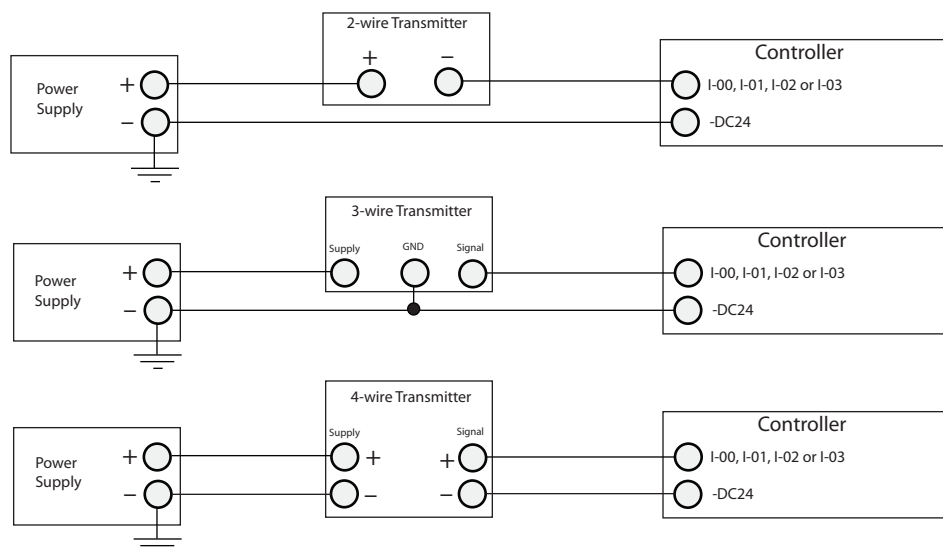
5. In Connected Components Workbench, go to Embedded I/O configuration page. Change the Gain parameter value for Input 00 to 98.



No changes are required to the Offset parameter value.

6. Repeat the same steps to calibrate all the other analog input channels.

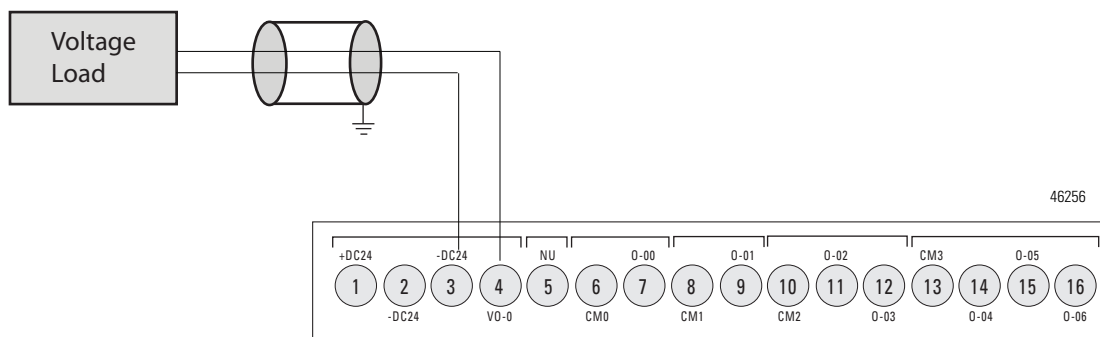
Transmitters to Analog Input



46257

Analog Output

The analog output can support voltage function as shown in the following illustration.



46256

The "-DC24" terminal is the analog ground connection for analog output (VO-0).

Communication Connections

Overview

This chapter describes how to communicate with your control system and configure communication settings. The method you use and cabling required to connect your controller depends on what type of system you are employing. This chapter also describes how the controller establishes communication with the appropriate network. Topics include:

Topic	Page
Supported Communication Protocols	55
Use Modems with Micro800 Controllers	61
Configure Serial Port	62
Configure Ethernet Settings	68
OPC Support Using RSLinx Enterprise	70

The Micro820 controllers have the following embedded communication channels:

- a non-isolated RS232/RS485 combo port
- RJ45 Ethernet port

Supported Communication Protocols

Micro820 controllers support communication through the embedded RS232/RS485 serial port as well as any installed serial port plug-in modules. In addition, Micro820 controllers also support communication through the embedded Ethernet port, and can be connected to a local area network for various devices providing 10 Mbps/100 Mbps transfer rate.

These are the communication protocols supported by Micro820 controllers:

- Modbus RTU Master and Slave
- CIP Serial Client/Server (RS232 only)
- ASCII
- EtherNet/IP Client/Server
- Modbus TCP Client/Server
- CIP Symbolic Client/Server
- DHCP Client
- Sockets Client/Server TCP/UDP

Connection Limits for Micro820 Controllers

Description		Connections
CIP Connections		
Total number of client plus server connections		24
Maximum number of client connections for all ports		16
Maximum number of server connections for all ports		24
Maximum number of EtherNet/IP connections	Client	16
	Server	24
Maximum number of USB connections	Client	–
	Server	
Maximum number of serial connections	Client	16
	Server	24
TCP Connections		
Total number of client plus server connections		64
Maximum number for EtherNet/IP	Client	16
	Server	16
Maximum number for Modbus TCP	Client	16
	Server	16
Maximum number for User Programmable Sockets		8
User Programmable Sockets		
Total number of User Programmable Sockets (any combination of UDP plus TCP Client/Server)		8

TIP The 2080-REMLCD USB uses the CIP serial server connections for embedded serial port.

IMPORTANT If all client/server connections are fully loaded, performance may be affected, such as data loss and intermittent delays during communication.

Here are some configuration examples based on the limits described in the table above:

1. The maximum number of drives that can be controlled over EtherNet/IP is 16. This is due to the maximum limit of TCP Client connections is 16, and the maximum limit of EtherNet/IP Client connections is also 16.
2. If you have 10 devices controlled over EtherNet/IP, the maximum number of devices that can be controlled over Serial is six. This is due to the maximum limit of Client connections is 16.
3. The total number of UDP sockets plus TCP Client/Server sockets has a maximum limit of eight.

Modbus RTU

Modbus is a half-duplex, master-slave communications protocol. The Modbus network master reads and writes bits and registers. Modbus protocol allows a single master to communicate with a maximum of 247 slave devices. Micro800 controllers support Modbus RTU Master and Modbus RTU Slave protocol. For more information on configuring your Micro800 controller for Modbus protocol, refer to the Connected Components Workbench Online Help. For more information about the Modbus protocol, refer to the Modbus Protocol Specifications (available from <http://www.modbus.org>).

See [Modbus Mapping for Micro800 on page 183](#) for information on Modbus mapping. To configure the Serial port as Modbus RTU, see [Configure Modbus RTU on page 65](#).

TIP Use MSG_MODBUS instruction to send Modbus messages over serial port.

CIP Serial Client/Server (RS232 only)

CIP Serial Client/Server allows CIP protocol to be used over an RS232 serial port. It is typically used with modems. The advantage over non-CIP serial protocols is that since the protocol is CIP, program downloads are supported including CIP pass-through from the serial port to Ethernet.

ASCII

ASCII provides connection to other ASCII devices, such as bar code readers, weigh scales, serial printers, and other intelligent devices. You can use ASCII by configuring the embedded or any plug-in serial RS232 or RS485 port for the ASCII driver. Refer to the Connected Components Workbench Online Help for more information.

To configure the serial port for ASCII, see [Configure ASCII on page 67](#).

Modbus TCP Client/Server

The Modbus TCP Client/Server communication protocol uses the same Modbus mapping features as Modbus RTU, but instead of the Serial port, it is supported over Ethernet. Modbus TCP Server takes on Modbus Slave features on Ethernet.

The Micro820 controller supports up to 16 simultaneous Modbus TCP Client connections and 24 simultaneous Modbus TCP Server connections.

No protocol configuration is required other than configuring the Modbus mapping table. For information on Modbus mapping, see [Modbus Mapping for Micro800 on page 183](#).

TIP Use MSG_MODBUS2 instruction to send Modbus TCP message over Ethernet port.

CIP Symbolic Client/Server

CIP Symbolic is supported by any CIP compliant interface including Ethernet (EtherNet/IP) and Serial Port (CIP Serial). This protocol allows HMIs to easily connect to the Micro820 controller.

CIP Serial, supported on the Micro820 controller, makes use of DF1 Full Duplex protocol, which provides point-to-point connection between two devices.

The Micro800 controllers support the protocol through RS232 connection to external devices, such as computers running RSLinx Classic software, PanelView Component terminals (firmware revisions 1.70 and above), PanelView 800 terminals or other controllers that support CIP Serial over DF1 Full-Duplex, such as ControlLogix and CompactLogix controllers that have embedded serial ports.

EtherNet/IP, supported on the Micro820 controller, makes use of the standard Ethernet TCP/IP protocol.

The Micro820 controller supports up to 16 simultaneous EtherNet/IP Client connections and 24 simultaneous EtherNet/IP Server connections.

To configure CIP Serial, see [Configure CIP Serial Driver on page 63](#).

To configure for EtherNet/IP, see [Configure Ethernet Settings on page 68](#).

CIP Symbolic Addressing

Users may access any global variable through CIP Symbolic addressing except for system and reserved variables.

One- or two-dimension arrays for simple data types are supported (for example, ARRAY OF INT[1..10, 1..10]) are supported but arrays of arrays (for example, ARRAY OF ARRAY) are not supported. Array of strings are also supported.

Supported Data Types in CIP Symbolic

Data Type⁽¹⁾	Description
BOOL	Logical Boolean with values TRUE(1) and FALSE(0) (Uses up 8 bits of memory)
SINT	Signed 8-bit integer value
INT	Signed 16-bit integer value
DINT	Signed 32-bit integer value
LINT ⁽²⁾	Signed 64-bit integer value
USINT	Unsigned 8-bit integer value
UINT	Unsigned 16-bit integer value
UDINT	Unsigned 32-bit integer value
ULINT ⁽²⁾	Unsigned 64-bit integer value
REAL	32-bit floating point value
LREAL ⁽²⁾	64-bit floating point value
STRING	Character string (1 byte per character)
DATE ⁽³⁾	Unsigned 32-bit integer value
TIME ⁽³⁾	Unsigned 32-bit integer value

⁽¹⁾ Logix MSG instruction can read/write SINT, INT, DINT, LINT and REAL data types using “CIP Data Table Read” and “CIP Data Table Write” message types.
BOOL, USINT, UINT, UDINT, ULINT, LREAL, STRING, SHORT_STRING, DATE, and TIME data types are not accessible with the Logix MSG instruction.

⁽²⁾ Not supported in PanelView Component or PanelView 800.

⁽³⁾ Can be used by sending data to UDINT, mainly for use with PanelView Plus and PanelView 800 HMI terminals.

CIP Client Messaging

CIP Generic and CIP Symbolic messages are supported on Micro800 controllers through the Ethernet and serial ports. These client messaging features are enabled by the MSG_CIPSYMBOLIC and MSG_CIPGENERIC function blocks.

See Micro800 Programmable Controllers: Getting Started with CIP Client Messaging, publication [2080-QS002](#), for more information and sample quickstart projects to help you use the CIP Client Messaging feature.

Sockets Client/Server TCP/UDP

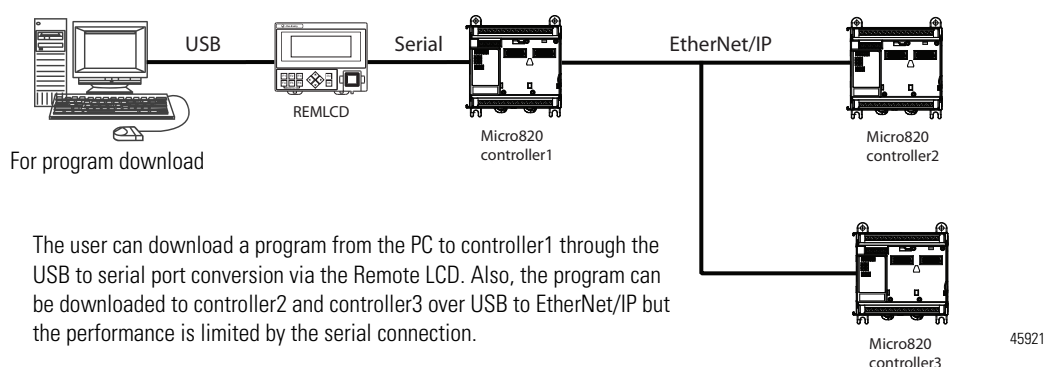
Sockets protocol is used for Ethernet communications to devices which do not support Modbus TCP and Ethernet/IP. Sockets support client and server, and TCP and UDP. Typical applications include communicating to printers, barcode readers, and PCs.

CIP Communications Pass-thru

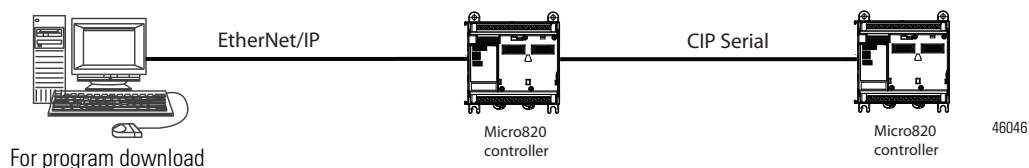
The Micro820 controllers support pass-thru on any communications port that supports Common Industrial Protocol (CIP) for applications such as program download. It does not support applications that require dedicated connections such as HMI. The maximum number of supported hops is **one**. A hop is defined to be an intermediate connection or communications link between two devices – in Micro800, this is through EtherNet/IP or CIP Serial.

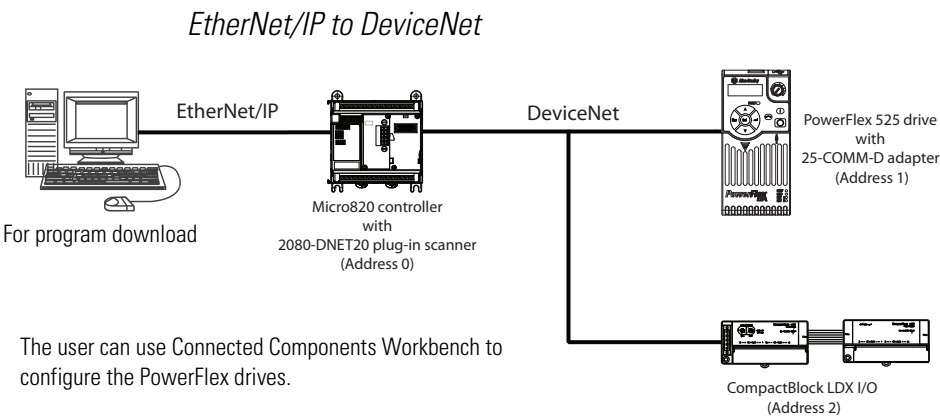
Examples of Supported Architectures

CIP Serial to EtherNet/IP



EtherNet/IP to CIP Serial





IMPORTANT Micro800 controllers do not support more than one hop (for example, from Ethernet/IP → CIP Serial → Ethernet/IP).

Use Modems with Micro800 Controllers

Serial modems can be used with the Micro820 controllers.

Making a DF1 Point-to-Point Connection

You can connect the Micro820 programmable controller to your serial modem. The recommended protocol for this is Modbus RTU.

Construct Your Own Modem Cable

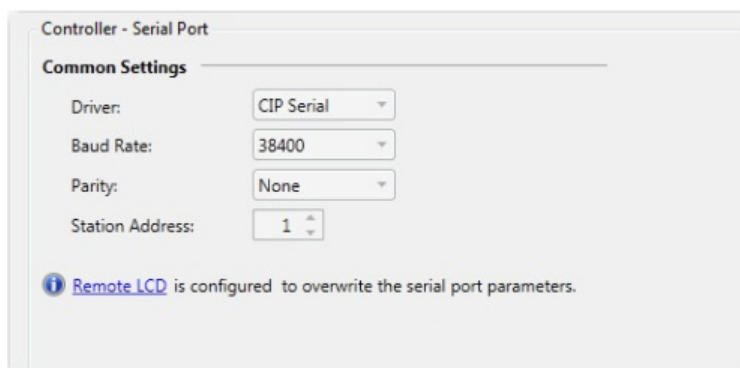
If you construct your own modem cable, the maximum cable length is 3 m (10 ft) with a 25-pin or 9-pin connector. Refer to the following typical pinout for constructing a straight-through cable:

DTE Device (Micro820 Channel 0)		DCE Device (Modem, etc.)		
6-Pin		25-Pin	9-Pin	
5	TXD	TXD	2	3
4	RXD	RXD	3	2
6	GND	GND	7	5
1	B(+)	DCD	8	1
2	A(-)	DTR	20	4
3	GND	DSR	6	6
		CTS	5	8
		RTS	4	7

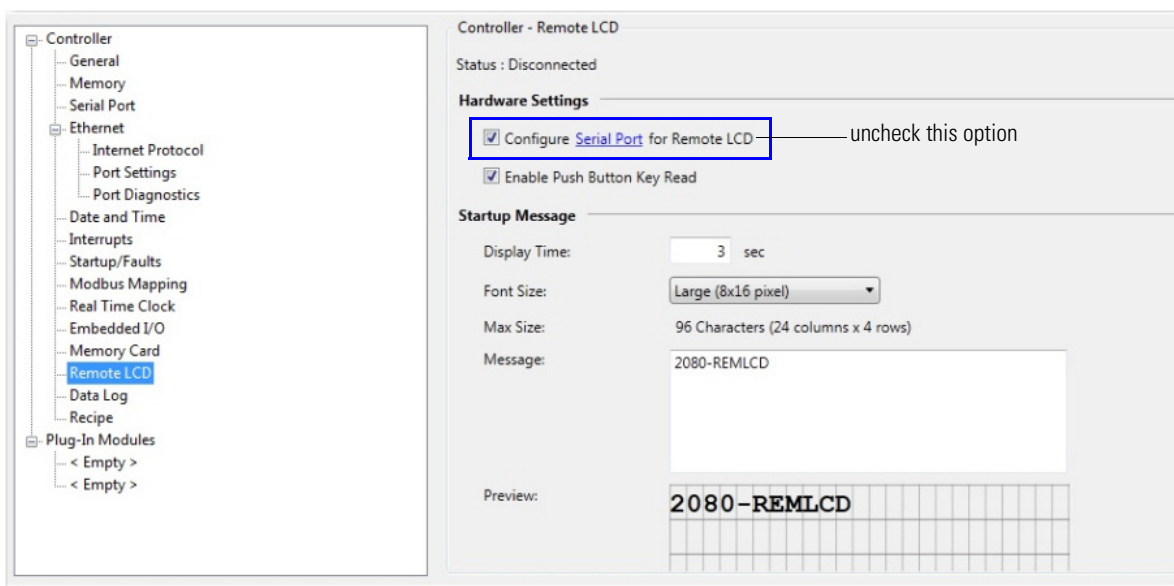
Configure Serial Port

You can configure the Serial Port driver as CIP Serial, Modbus RTU, ASCII or choose Shutdown through the Controller Configuration tree in Connected Components Workbench software.

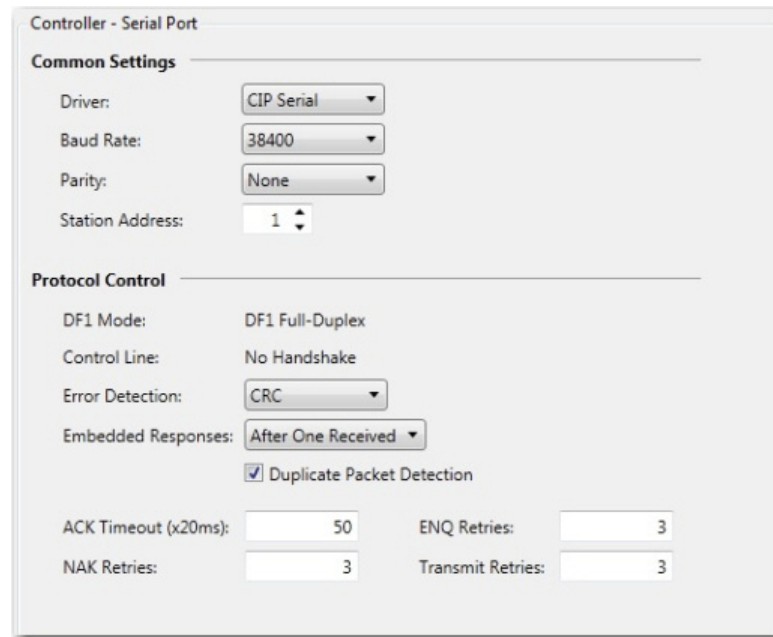
By default, when a Micro820 controller is added to the Project Organizer in Connected Components Workbench, Remote LCD parameters are configured to overwrite serial port settings.



To edit serial port settings, go to the Remote LCD configuration page and uncheck the Configure Serial Port for Remote LCD option button.



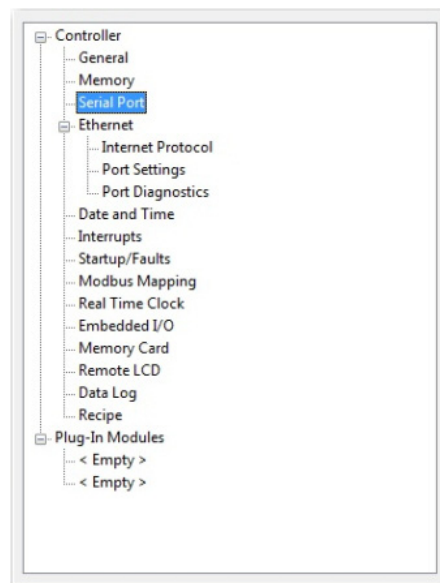
When the Remote LCD configuration is unchecked, the serial port values are visible and can be edited.



IMPORTANT After changing the serial port settings on the controller, power cycle the Remote LCD.

Configure CIP Serial Driver

1. Open your Connected Components Workbench project. On the device configuration tree, go to the Controller properties. Click Serial Port.



2. Select CIP Serial from the Driver field.

3. Specify a baud rate. Select a communication rate that all devices in your system support. Configure all devices in the system for the same communication rate. Default baud rate is set @ 38400 bps.
4. In most cases, parity and station address should be left at default settings.
5. Click Advanced Settings and set Advanced parameters. Refer to the table [CIP Serial Driver Parameters on page 64](#) for a description of the CIP Serial parameters.

CIP Serial Driver Parameters

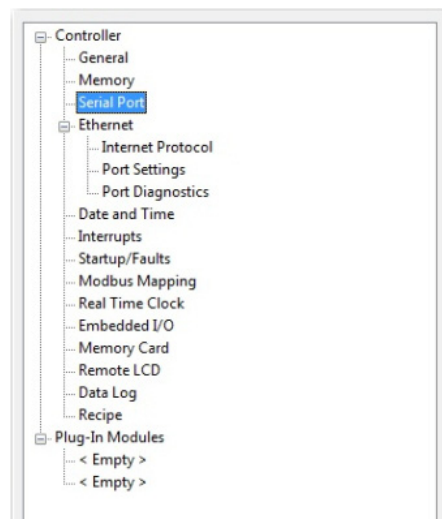
Parameter	Options	Default
Baud rate	Toggles between the communication rate of 1200, 2400, 4800, 9600, 19200, and 38400.	38400
Parity	Specifies the parity setting for the serial port. Parity provides additional message-packet error detection. Select Even, Odd, or None.	None
Station Address	The station address for the serial port on the DF1 master. The only valid address is 0...254.	1
DF1 Mode	DF1 Full Duplex (read only)	Configured as full-duplex by default.
Control Line	No Handshake (read only)	Configured as no handshake by default.
Duplicate Packet Detection	Detects and eliminates duplicate responses to a message. Duplicate packets may be sent under noisy communication conditions when the sender's retries are not set to 0. Toggles between Enabled and Disabled.	Enabled
Error Detection	Toggles between CRC and BCC.	CRC

CIP Serial Driver Parameters

Embedded Responses	To use embedded responses, choose Enabled Unconditionally. If you want the controller to use embedded responses only when it detects embedded responses from another device, choose After One Received. If you are communicating with another Allen-Bradley device, choose Enabled Unconditionally. Embedded responses increase network traffic efficiency.	After One Received
NAK Retries	The number of times the controller will resend a message packet because the processor received a NAK response to the previous message packet transmission.	3
ENQ Retries	The number of enquiries (ENQs) that you want the controller to send after an ACK timeout occurs.	3
Transmit Retries	Specifies the number of times a message is retried after the first attempt before being declared undeliverable. Enter a value from 0...127.	3
ACK Timeout (x20 ms)	Specifies the amount of time after a packet is transmitted that an ACK is expected.	50

Configure Modbus RTU

1. Open your Connected Components Workbench project. On the device configuration tree, go to the Controller properties. Click Serial Port.



2. Select Modbus RTU on the Driver field.

Controller - Serial Port

Common Settings

Driver: Modbus RTU

Baud Rate: 19200

Parity: None

Modbus Role: Master

Protocol Control

Media: RS232 no handshake

Data Bits: 8

Stop Bits: 1

Response Timer: 200 ms

Broadcast Pause: 200 ms

Inter-Char Timeout: 0 μs

RTS Pre-Delay: 0 μs

RTS Post-Delay: 0 μs

3. Specify the following parameters:

- Baud rate
- Parity
- Unit address
- Modbus Role (Master, Slave, Auto)

Modbus RTU Parameters

Parameter	Options	Default
Baud Rate	1200, 2400, 4800, 9600, 19200, 38400	19200
Parity	None, Odd, Even	None
Modbus Role	Master, Slave, Auto	Master

4. Click Advanced Settings to set advanced parameters.

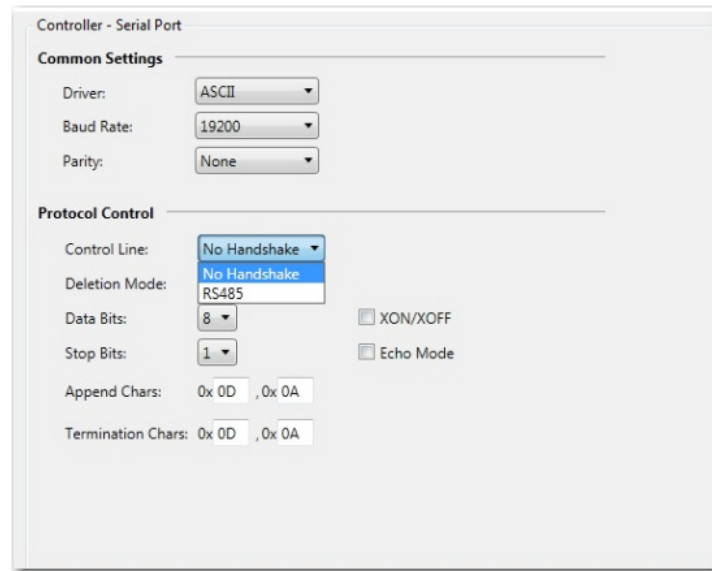
Refer to the table for available options and default configuration for advanced parameters.

Modbus RTU Advanced Parameters

Parameter	Options	Default
Media	RS-232, RS-232 RTS/CTS, RS-485	RS-232
Data bits	Always 8	8
Stop bits	1, 2	1
Response timer	0...999,999,999 milliseconds	200
Broadcast Pause	0...999,999,999 milliseconds	200
Inter-char timeout	0...999,999,999 microseconds	0
RTS Pre-delay	0...999,999,999 microseconds	0
RTS Post-delay	0...999,999,999 microseconds	0

Configure ASCII

1. Open your Connected Components Workbench project. On the device configuration tree, go to Controller properties. Click Serial Port.
2. Select ASCII on the Driver field.



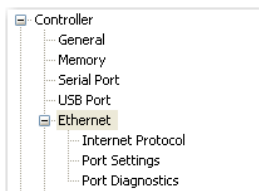
3. Specify baud rate and parity.

ASCII Parameters

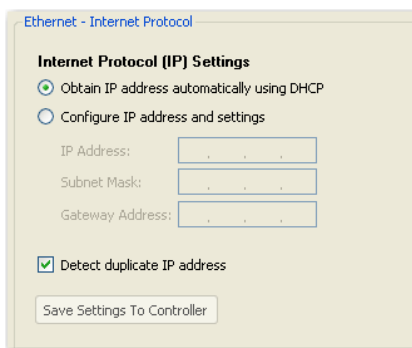
Parameter	Options	Default
Baud Rate	1200, 2400, 4800, 9600, 19200, 38400	19200
Parity	None, Odd, Even	None
Control Line	RS485 No Handshake	No Handshake
Deletion Mode	CRT Ignore Printer	Ignore
Data Bits	7, 8	8
XON/XOFF	Enabled or Disabled	Disabled
Stop Bits	1, 2	1
Echo Mode	Enabled or Disabled	Disabled
Append Chars	0x0D,0x0A or user-specified value	0x0D,0x0A
Termination Chars	0x0D,0x0A or user-specified value	0x0D,0x0A

Configure Ethernet Settings

1. Open your Connected Components Workbench project (for example, Micro820). On the device configuration tree, go to Controller properties. Click Ethernet.



2. Under Ethernet, click Internet Protocol.
Configure Internet Protocol (IP) settings. Specify whether to obtain the IP address automatically using DHCP or manually configure IP address, subnet mask, and gateway address.



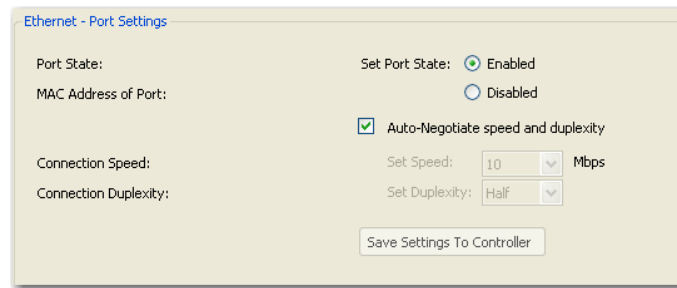
TIP The Ethernet port defaults to the following out-of-the box settings:

- DHCP (dynamic IP address)
- Address Duplicate Detection: On

IMPORTANT When a DHCP server fails, the Micro800 controller allocates IP addresses in the private range 169.254.0.1 to 169.254.255.254. The Micro800 controller verifies its address is unique on the network using ARP. When the DHCP server is again able to service requests, the Micro800 controller updates its address automatically.

3. Click the checkbox Detect duplicate IP address to enable detection of duplicate address.

4. Under Ethernet, click Port Settings.



5. Set Port State as Enabled or Disabled.
6. To manually set connection speed and duplexity, uncheck the option box Auto-Negotiate speed and duplexity. Then, set Speed (10 or 100 Mbps) and Duplexity (Half or Full) values.
7. Click Save Settings to Controller if you would like to save the settings to your controller.
8. On the device configuration tree, under Ethernet, click Port Diagnostics to monitor Interface and Media counters. The counters are available and updated when the controller is in Debug mode.

Validate IP Address

Modules must validate the incoming IP address configuration, whether it is obtained through explicit configuration or through DHCP.

The following rules must be obeyed when configuring the IP address:

- The IP address for the module cannot be set to zero, a multicast address, a broadcast address, or an address on the Class A loopback network (127.x.x.x).
- The IP address should not start with zero, and the IP address network ID should be not zero.
- The Network mask cannot be set to 255.255.255.255.
- The Gateway address must be on the same subnet as the IP address that is being configured.
- The Name Server address cannot be set to zero, a multicast address, a broadcast address, or an address on the Class A loopback network (127.x.x.x).

The valid range of static IPv4 IP address exclude:

- Broadcast or zero IP (255.255.255.255 or 0.0.0.0)
- IP address starting with 0 or 127 (0.xxx.xxx.xxx or 127.xxx.xxx.xxx)
- IP address ending with 0 or 255 (xxx.xxx.xxx.0 or xxx.xxx.xxx.255)
- IP addresses in range 169.254.xxx.xxx (169.254.0.0 to 169.254.255.255)
- IP addresses in range 224.0.0.0 to 255.255.255.255

Ethernet Host Name

Micro800 controllers implement unique host names for each controller, to be used to identify the controller on the network. The default host name is comprised of two parts: product type and MAC address, separated by a hyphen. For example: 2080LC20-xxxxxxxxxxxx, where xxxxxxxxxxxx is the MAC address.

The user can change the host name using the CIP Service Set Attribute Single when the controller is in Program/Remote Program mode.

OPC Support Using RSLinx Enterprise

Support for Open Platform Communications (OPC) using CIP symbolic has been added from firmware release 7.0 onwards. This can be used in place of Modbus addressing.

RSLinx Enterprise version 5.70 (CPR9 SR7) or later and FactoryTalk Gateway version 3.70 (CPR9 SPR7) or later are required.

Program Execution in Micro800

This section provides a brief overview of running or executing programs with a Micro800 controller.

IMPORTANT This section generally describes program execution in Micro800 controllers. Certain elements may not be applicable or true in certain models (for example, Micro820 does not support PTO motion control).

For detailed information regarding ladder diagrams, instructions, function blocks and so on, see the Micro800 Programmable Controllers Instruction Manual, publication [2080-RM001](#).

Overview of Program Execution

A Micro800 cycle or scan consists of reading inputs, executing programs in sequential order, updating outputs and performing housekeeping (data log, 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 may 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, it displays the program icons based on this order. You can view and modify an order number for a program from the program's 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 is exceeded. This 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 just prior to the controller going into Fault mode.

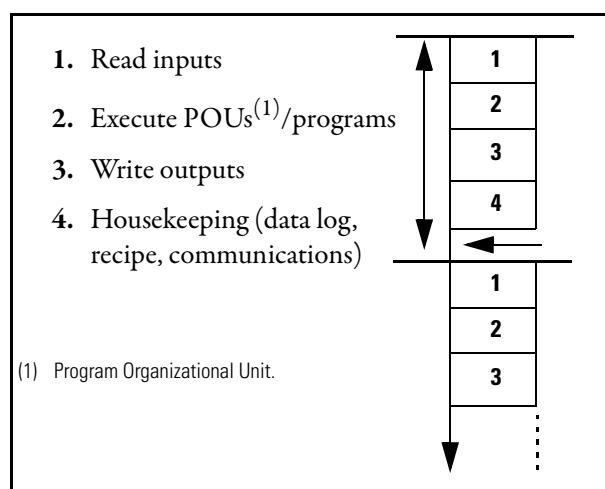
In addition to the User Fault Routine, Micro800 controllers also support two Selectable Timed Interrupts (STI). STIs execute assigned programs once every set point interval (1...65535 ms).

The Global System Variables 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 POU's 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.

Optional Module

Normally before the read inputs step, the controller will verify the presence of any configured plug-in and expansion I/O modules. If a plug-in or expansion I/O module is missing, the controller will fault. In Connected Components Workbench software release 10 or later, an Optional Module configuration option is added to prevent a missing plug-in I/O or expansion I/O module from faulting the controller if enabled. This option can be enabled separately for each plug-in I/O or expansion I/O module.



ATTENTION: If the optional module feature is enabled, use the MODULE_INFO instruction to verify that the module is present because the controller will not fault if the module is missing.

Controller Load and Performance Considerations

Within one program scan cycle, the execution of the main steps (as indicated in the Execution Rules diagram) could be interrupted by other controller activities which have higher priority than the main steps. Such activities include,

1. User Interrupt events, including STI, EII, and HSC interrupts (when applicable);
2. Communication data packet receiving and transmitting;
3. PTO Motion engine periodical execution (if supported by the controller).

When one or several of these activities occupy a significant percentage of the Micro800 controller execution time, the program scan cycle time will be prolonged. The Watchdog timeout fault (0xD011) could be reported if the impact of these activities is underestimated, and the Watchdog timeout is set marginally. The Watchdog setting defaults to 2 s and generally never needs to be changed.

Periodic Execution of Programs

For applications where periodic execution of programs with precise timing is required, such as for PID, it is recommended that STI (Selectable Timed Interrupt) be used to execute the program. STI provides precise time intervals.

It is not recommended that the system variable __SYSVA_TCYCYCTIME be used to periodically execute all programs as this also causes all communication to execute at this rate.



WARNING: Communication timeouts may occur if programmed cycle time is set too slow (for example, 200 ms) to maintain communications.

System Variable for Programmed Cycle Time

Variable	Type	Description
__SYSVA_TCYCYCTIME	TIME	Programmed cycle time. Note: Programmed cycle time only accepts values in multiples of 10 ms. If the entered value is not a multiple of 10, it will be rounded up to the next multiple of 10.

Power Up and First Scan

In Program mode, all analog and digital input variables hold their last state, and the LEDs are always updated. Also all analog and digital output variables hold their last state, but only the analog outputs hold their last state while the digital outputs are off.

When transitioning from Program mode to Run mode, all analog output variables hold their last state but all digital output variables are cleared.

Two system variables are available.

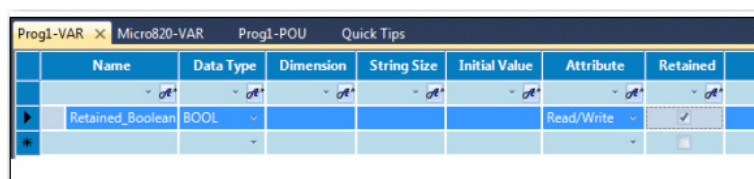
System Variables for Scan and Powerup

Variable	Type	Description
_SYSVA_FIRST_SCAN	BOOL	First scan bit. Can be used to initialize or reset variables immediately after every transition from Program to Run mode. Note: True only on first scan. After that, it is false.
_SYSVA_POWER_UP_BIT	BOOL	Powerup bit. Can be used to initialize or reset variables immediately after download from Connected Components Workbench or immediately after being loaded from memory backup module (for example, microSD card). Note: True only on the first scan after a powerup, or running a new ladder for the first time.

Variable Retention

Micro830 and Micro850 controllers retain all user-created variables after a power cycle, but the variables inside instances of instructions are cleared. For example: A user created variable called My_Timer of Time data type will be retained after a power cycle but the elapsed time (ET) within a user created timer TON instruction will be cleared.

Unlike Micro830/Micro850 controllers, Micro810 and Micro820 controllers can only retain a maximum of 400 bytes of user-created variable values. This means that after a power cycle, global variables are cleared or set to initial value, and only 400 bytes of user-created variable values are retained. Retained variables can be checked at the global variable page.



Name	Data Type	Dimension	String Size	Initial Value	Attribute	Retained
Retained_Boolean	BOOL				Read/Write	<input checked="" type="checkbox"/>

Memory Allocation

Depending on base size, available memory on Micro800 controllers are shown in the table below.

Memory Allocation for Micro800 Controllers

Attribute	10/16-point	20-point	24- and 48-points
Program steps ⁽¹⁾	4 K	10 K	10 K
Data bytes	8 KB	20 KB	20 KB

(1) Estimated Program and Data size are "typical" – program steps and variables are created dynamically.
1 Program Step = 12 data bytes.

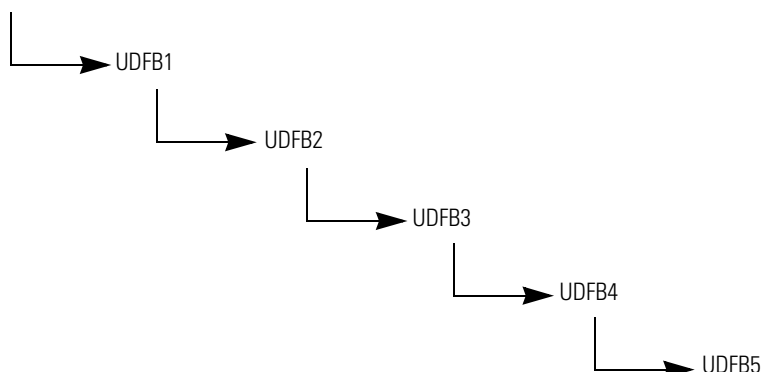
These specifications for instruction and data size are typical numbers. When a project is created for Micro800, memory is dynamically allocated as either program or data memory at build time. This means that program size can exceed the published specifications if data size is sacrificed and vice versa. This flexibility allows maximum usage of execution memory. In addition to the user defined variables, data memory also includes any constants and temporary variables generated by the compiler at build time.

The Micro800 controllers also have project memory, which stores a copy of the entire downloaded project (including comments), as well as configuration memory for storing plug-in setup information, and so on.

Guidelines and Limitations for Advanced Users

Here are some guidelines and limitations to consider when programming a Micro800 controller using Connected Components Workbench software:

- Each program/POU can use up to 64 Kb of internal address space. It is recommended that you split large programs into smaller programs to improve code readability, simplify debugging and maintenance tasks.
- A User Defined Function (UDF) uses significantly less memory than a User Defined Function Block (UDFB). For example, 30% less for a typical sized program compared to a UDFB with one instance. The savings increases as the number of UDFB instances increases.
- A User Defined Function Block (UDFB) can be executed within another UDFB, with a limit of five nested UDFBs. Avoid creating UDFBs with references to other UDFBs, as executing these UDFBs too many times may result in a compile error. This also applies to UDFs.

Example of Five Nested UDFBs

- Structured Text (ST) is much more efficient and easier to use than Ladder Logic, when used for equations. If you are used to using the RSLogix 500 CPT Compute instruction, then a great alternative is to use ST combined with either UDF or UDFB.

As an example, for an Astronomical Clock Calculation, Structured Text uses 40% less Instructions.

Display_Output LD:

Memory Usage (Code) : 3148 steps

Memory Usage (Data) : 3456 bytes

Display_Output ST:

Memory Usage (Code) : 1824 steps

Memory Usage (Data) : 3456 bytes

- You may encounter an Insufficient Reserved Memory error while downloading and compiling a program over a certain size. One workaround is to use arrays, especially if there are many variables.

Controller Security

Micro800 security generally has two components:

- **Exclusive Access** which prevents simultaneous configuration of the controller by two users
- **Controller Password Protection** which secures the Intellectual Property contained within the controller and prevents unauthorized access

Exclusive Access

Exclusive access is enforced on the Micro800 controller regardless of whether the controller is password-protected or not. This means that only one Connected Components Workbench session is authorized at one time and only an authorized client has exclusive access to the controller application. This ensures that only one software session has exclusive access to the Micro800 application-specific configuration.

Exclusive access is enforced on Micro800 firmware. When a Connected Components Workbench user connects to a Micro800 controller, the controller is given exclusive access to that controller.

Password Protection

By setting a password on the controller, a user effectively restricts access to the programming software connection of the controller to software sessions that can supply the correct password. Essentially, Connected Components Workbench operations such as upload and download are prevented if the controller is secured with a password and the correct password is not provided.

Micro800 controllers are shipped with no password but a password can be set through the Connected Components Workbench software (using firmware revision 2 or later).

In Connected Components Workbench software revision 10 or later, a stronger password algorithm is introduced to provide better security. To take full advantage of this enhancement, the Micro800 controller must have firmware revision 10 or later, and the project must also be version 10 or later.

The controller password is also backed up to the memory backup module (that is, 2080-MEMBAK-RTC for Micro830 and Micro850; 2080-LCD for Micro810; and microSD card for Micro820 controllers).

TIP

For instructions on how to set, change, and clear controller passwords, see [Configure Controller Password on page 158](#).

Compatibility

The Controller Password feature is supported on:

- Connected Components Workbench **revision 2** and later
- Micro800 controllers with at least **revision 2** firmware

For users with earlier versions of the software and/or hardware, refer to the compatibility scenarios below.

Connected Components Workbench revision 1 with Micro800 controller firmware revision 2 and later

Connection to a Micro800 controller with firmware revision 2 using an earlier version of the Connected Components Workbench software (revision 1) is possible and connections will be successful. However, the software will not be able to determine whether the controller is locked or not.

If the controller is not locked, access to the user application will be allowed, provided the controller is not busy with another session. If the controller is locked, access to the user application will fail. Users will need to upgrade to revision 2 of the Connected Components Workbench software.

Connected Components Workbench revision 2 and later with Micro800 controller firmware revision 1

Connected Components Workbench revision 2 is capable of "discovering" and connecting to Micro800 controllers with firmware revision earlier than revision 2 (that is, not supporting the Controller Password feature). However, the Controller Password feature will not be available to these controllers. The user will not be able to see interfaces associated with the Controller Password feature in the Connected Components Workbench session.

Users are advised to upgrade the firmware. See [Flash Upgrade Your Micro800 Firmware on page 147](#) for instructions.



ATTENTION: Connected Components Workbench revision 9 or earlier with Micro800 controller revision 10 or later

If a Micro800 controller with firmware revision 10 or later is locked using the new password algorithm introduced in Connected Components Workbench software revision 10 or later, it cannot be accessed using Components Workbench software revision 9 or earlier. Users are advised to upgrade to the latest version of Connected Components Workbench software.

Work with a Locked Controller

The following workflows are supported on compatible Micro800 controllers (firmware revision 2) and Connected Components Workbench software revision 2.

Upload from a Password-Protected Controller

1. Launch the Connected Components Workbench software.
2. In the Project Organizer, expand Catalog by clicking the + sign.
3. Select the target controller.
4. Select Upload.
5. When requested, provide the controller password.

IMPORTANT When using Connected Components Workbench software revision 9 or earlier:

- You cannot upload a revision 10 or later project from the controller.
- You can upload a revision 9 or earlier project from the controller if it was downloaded to the controller using Connected Components Workbench software revision 10 or later, but you cannot go online.

Debug a Password-Protected Controller

To debug a locked controller, you have to connect to the controller through the Connected Components Workbench software and provide the password before you can proceed to debug.

1. Launch the Connected Components Workbench software.
2. In the Project Organizer, expand Catalog by clicking the + sign.
3. Select the catalog number of your controller.
4. When requested, provide the controller password.
5. Build and save your project.
6. Debug.

Download to a Password-Protected Controller

1. Launch the Connected Components Workbench software.
2. Click Connect.
3. Select the target controller.
4. When requested, provide the controller password.
5. Build and save the project, if needed.

6. Click Download.
7. Click Disconnect.

IMPORTANT If the controller has a password locked revision 10 or later project, you cannot access the controller using Connected Workbench software revision 9 or earlier. If you use Connected Components Workbench software revision 10 or later to download a revision 9 or earlier project, the password in the controller will be automatically converted to the old algorithm.

IMPORTANT If the controller has a password locked revision 9 or earlier project and you use Connected Components Workbench software revision 10 or later, to download a revision 10 or later project, the password in the controller will be automatically converted to the new algorithm.

IMPORTANT If communication is lost during the download, repeat the download and verify that the controller is password protected.

Transfer Controller Program and Lock Receiving Controller

In this scenario, the user needs to transfer user application from controller1 (locked) to another Micro800 controller with the same catalog number. The transfer of the user application is done through the Connected Components Workbench software by uploading from controller1, then changing the target controller in the Micro800 project, and then downloading to controller2. Finally, controller2 will be locked.

1. In the Project Organizer, click the Discover icon.
The Browse Connections dialog appears.
2. Select target controller1.
3. When requested, enter the controller password for controller1.
4. Build and save the project.
5. Click Disconnect.
6. Power down controller1.
7. Swap controller1 hardware with controller2 hardware.
8. Power up controller2.
9. Click Connect.
10. Select target controller2.
11. Click Download.
12. Lock controller2. See [Configure Controller Password on page 158](#).

Back Up and Restore a Password-Protected Controller

In this workflow, user application will be backed up from a Micro800 controller that is locked to a memory plug-in device.

1. In the Project Organizer, click the Discover icon.
The Browse Connections dialog appears.
2. Select the target controller.
3. When requested, enter the controller password.
4. Back up controller contents to the memory module.
The project in the memory module is now password locked.
5. Remove the memory module from controller1 and insert into controller2.
6. Restore contents from the memory module to controller2.
This operation succeeds only if:
 - the controller has no password – the project can be restored to the controller by setting the “Load on power up” option for the memory module to Load Always.
 - the controller’s password matches the project’s password.

IMPORTANT Even though the password matches, the restore operation will fail if either one of the controller or project in the memory module is protected using the old password algorithm, and the other is protected using the new password algorithm. You can flash update the controller using the Reset option to clear the password before restoring the project to the controller.

Configure Controller Password

To set, change, and clear controller password, see the quickstart instructions [Configure Controller Password on page 158](#).

Recover from a Lost Password

If the controller is secured with a password and the password has been lost, then it is impossible to access the controller using the Connected Components Workbench software.

To recover, the controller must be set to Program Mode using the keyswitch for Micro830 and Micro850 controllers, the 2080-LCD for Micro810 controllers, or the 2080-REMLCD for Micro820 controllers. Then, ControlFlash can be used to update the controller firmware, which also clears the controller memory. In Connected Components Workbench software version 10 or later, the Reset option must be selected for the controller memory to be cleared during the firmware update. If the Upgrade or Downgrade option is selected, the password is retained.



ATTENTION: The project in the controller will be lost but a new project can be downloaded.

Using the Micro800 Remote LCD

This chapter provides a description of how you can use the Micro800 Remote LCD with the Micro820 controller. It has the following sections.

Topic	Page
Overview	83
Text Display Mode	85
USB Mode	84
Backup and Restore	91
Hardware Features, Installation, and Specifications	91

Overview

The 2080-REMLCD module serves as a simple IP65 text display that allows the configuration of such controller settings as IP address. It connects to the Micro820 controller through the RS232 port. The Remote LCD module has a dot matrix LCD with backlight and supports multilingual characters. The display size is 3.5 inches with 192 x 64 pixel resolution.

It also has:

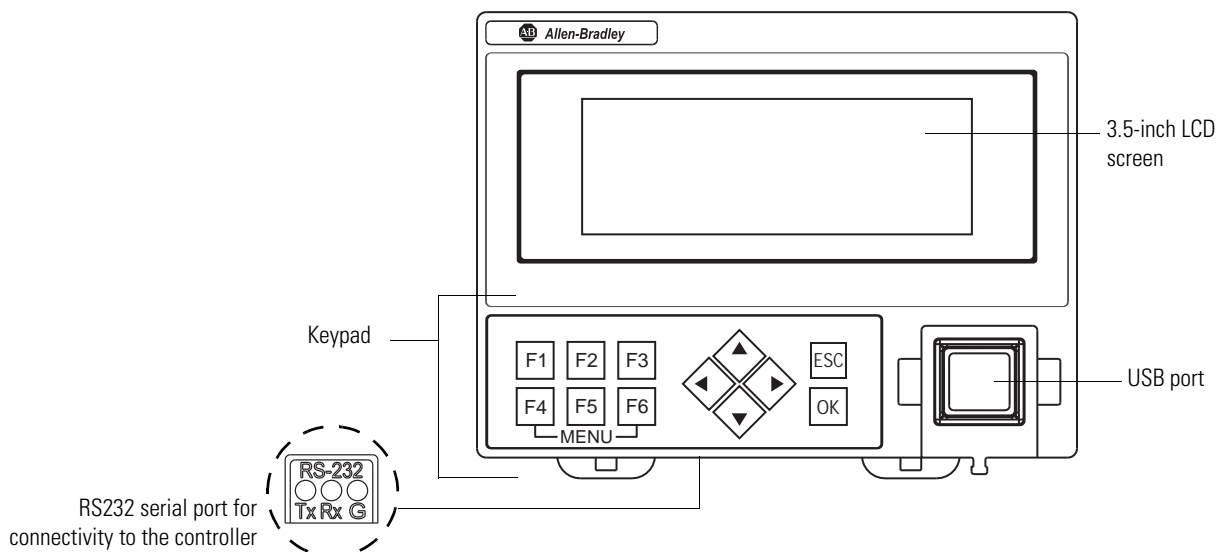
- Four arrow keys
- Six function keys
- ESC key
- OK key
- USB port for Connected Components Workbench connectivity

It supports:

- Small character set: 24 characters by 8 lines
- Large character set: 24 characters by 4 lines
- Extra large character set: displays 12 characters by 4 lines

The Remote LCD module supports English, French, Spanish, Italian and Simplified Chinese languages for the Main Menu.

Micro800 Remote LCD



The 2080-REMLCD module is IP65-rated and can be mounted through the front panel or on the same DIN rail as the Micro820 controller.

It has two modes of operation:

- USB Mode
- Text Display Mode
 - I/O Status and Main Menu operations (for example, change to RUN mode)
 - Optional user-defined screens (using the LCD_REM instructions)

USB Mode

In USB mode, the Remote LCD module acts as a USB pass-through for Connected Components Workbench. The Remote LCD module automatically enters USB mode when traffic is detected.

For example:

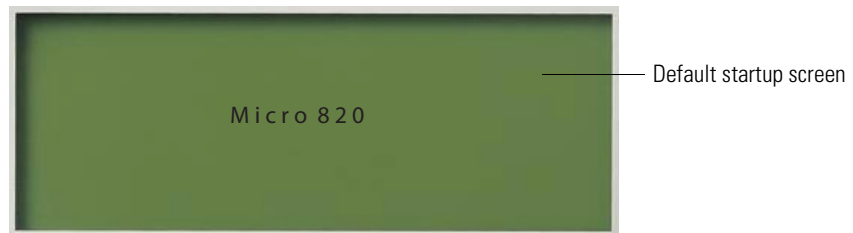
1. Remote LCD is in text display mode showing the I/O Status screen by default.
2. The user connects a USB cable between the PC and the Remote LCD.
3. Remote LCD is automatically detected by the PC as a USB device and the Remote LCD automatically goes to USB mode.
4. I/O Status screen is no longer shown. The user is now able to download program over USB using Connected Components Workbench.
5. When the USB cable is disconnected and no traffic is detected for 30 seconds, the Remote LCD automatically goes back to text display mode showing the I/O Status screen.

IMPORTANT Using the USB port is convenient when accessing the controller from the front of the cabinet without opening the door and when the IP address is unknown. For larger programs, it is recommended to use USB port through the Remote LCD to set the IP address and then use Ethernet to download. Ethernet is faster due to limitations of the USB to serial conversion.

Text Display Mode

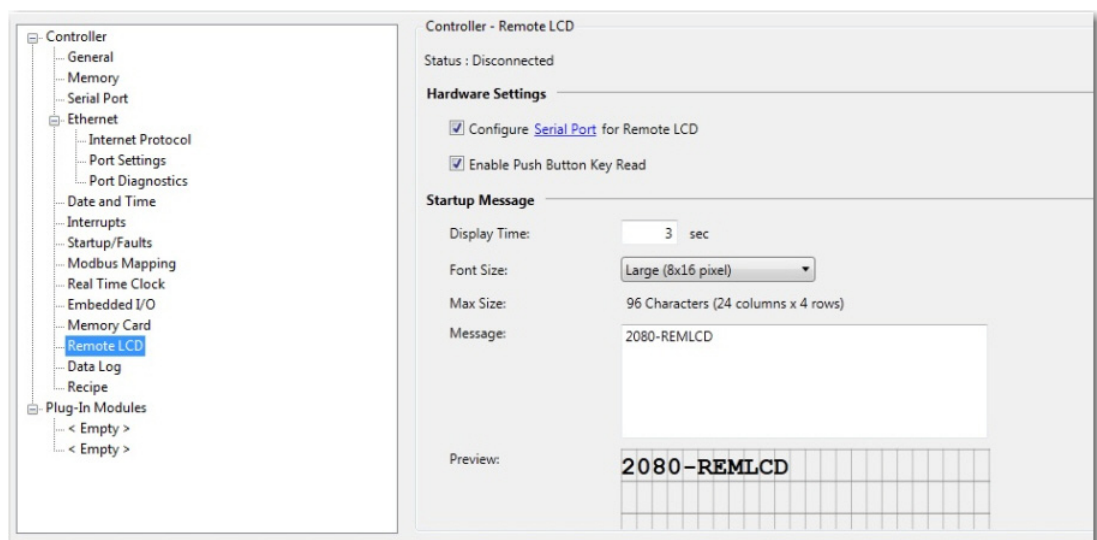
In text display mode, you are either in I/O Status, Main Menu, or executing Remote LCD instructions.

Startup Screen



On powerup, the Remote LCD module powers up with a splash screen that displays "Initializing". Then, it displays "Connecting to Controller" until the connection is established. The controller then displays the startup screen for 3 seconds by default or user-defined duration after the connection is established.

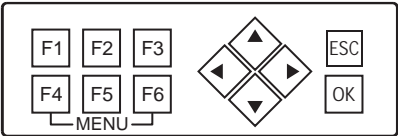
The user can customize this startup screen through Connected Components Workbench. The controller displays the default startup screen at powerup when the customized startup screen is blank.



After showing the startup message, the Remote LCD will show the I/O Status screen, assuming that no LCD_REM instructions are executing.

Navigate the Remote LCD

In text display mode, you can make use of available navigation keys (function keys, arrow keys, ESC and OK) to navigate through the menus.



The module has twelve keys with the following operations.

Function Keys Operation

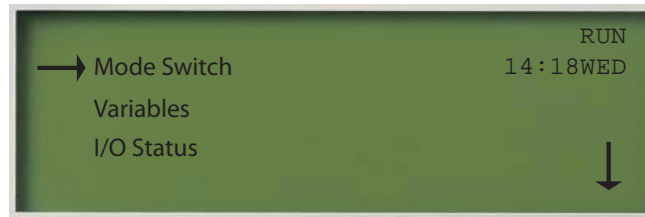
Button	Function
Arrow keys (cursor buttons)	Move cursor
	Select menu item
	Increment/Decrement Number
	Choose numbers, values, times, and so on
OK	Next menu level, store your entry
Esc	Previous menu level, cancel your entry.
F1	Variable (Shortcut)
F2	ENET Cfg (Shortcut)
F3	Mode Switch (Shortcut)
F4	Fault Mode (Shortcut)
F5	Security (Shortcut)
F6	Backlight (Shortcut)

Shortcut keys jump from the I/O Status screen to the specific main menu operation.

Main Menu

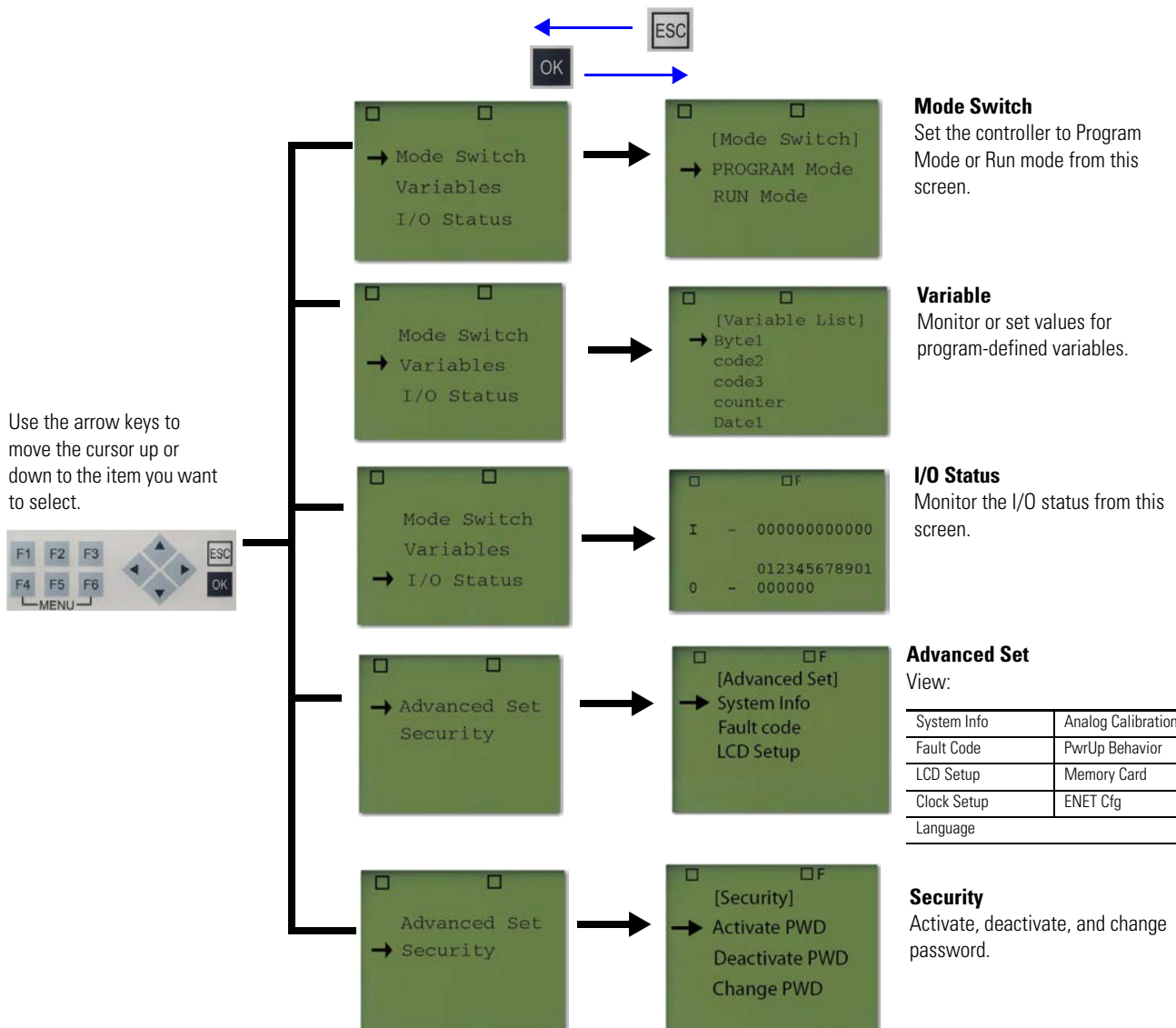
To access the Main Menu and available submenus, press F4 and F6 simultaneously. To exit the Main Menu, press ESC.

The Main Menu shows the following screen:



The following structure tree takes you through the different menus available in the Remote LCD module and their general description.

2080-REMLCD Menu Structure Tree



Main Menu Items

Menu Item	Description	
I/O Status	Shows the status of the local I/O.	
Mode switch	Change the mode switch selection.	
Variables	View and change the data value of a variable. Using Connected Components Workbench software, you can specify which variables in the program can be viewed and edited through the 2080-REMLCD module. See View and Edit Variable Values through the Remote LCD on page 88 .	
Security	Activate, deactivate and change password protection.	
Advanced Set	System Info	View system information such as operating systems series and firmware revision.
	Fault Code	View controller fault code information.
	LCD Setup	Adjust LCD contrast, backlight color and push button.
	Clock Setup	The real-time clock and daylight saving time.
	Language	Change menu language to French, Italian, Spanish, and Chinese.
	Analog Calibration	Configure calibration parameter of embedded analog inputs.
	PwrUp Behavior	Configure controller mode on powerup.
	Memory Card	Access the microSD card.
	ENET Cfg	View and change the Ethernet port configuration.

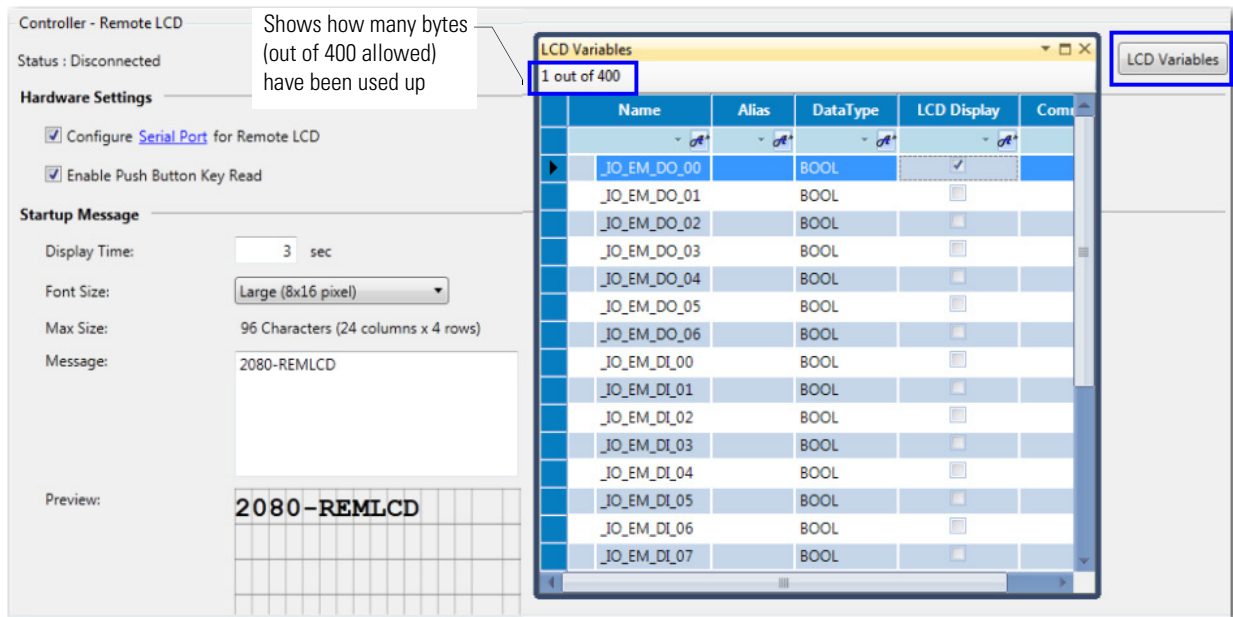
The controller limits certain operations according to controller mode, as shown in the following table.

Operational Limit on 2080-REMLCD

Operation	PROG Mode	RUN Mode
Variable Edit	NO	YES
Analog Calibration	YES	NO
Controller → Memory Card	YES	NO
Memory Card → Controller	YES	NO
Others	YES	YES

View and Edit Variable Values through the Remote LCD

Go to the 2080-REMLCD configuration window in Connected Components Workbench. Click LCD Variables and select which variables you would like to edit through the Remote LCD.



User-defined Screens

To create user-defined screens through Connected Components Workbench, you can program the Remote LCD module using the following function blocks.

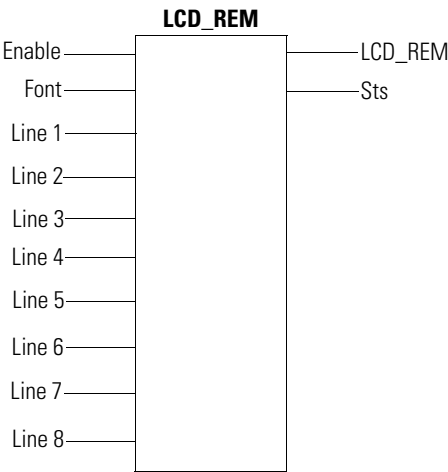
2080-REMLCD Function Blocks

Function Block Name	Description
LCD_REM	Used to display string or numbers on the Remote LCD.
KEY_READ_REM	Used to read keypad input on the Remote LCD.
LCD_BKLT_REM	Used to change the backlight color and mode of the Remote LCD screen.

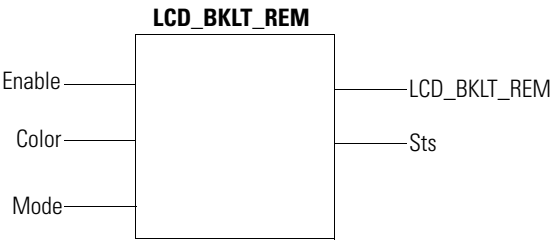
When the instructions are executing, the user-defined screen is shown, but when in the Main Menu, the Remote LCD instructions are disabled. For example, the KEY_READ_REM instruction will no longer read keypad input.

LCD_REM

The LCD_REM function block is used to display user strings on the REMLCD module when REMLCD is present and connected.



LCD_BKLT_REM

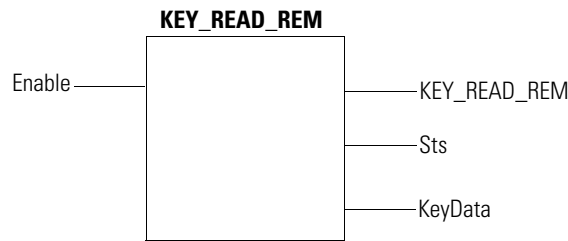


This function block is used to configure backlight parameters on the Remote LCD module.

Execution of the LCD_BKLT_REM takes precedence over current backlight settings in the Main Menu. When Enable input goes False and the instructions stop executing, the last Main Menu setting of the backlight takes effect.

The LCD_BKLT_REM instruction is only effective when displaying user-defined screen or default I/O Status screen. While in the Main Menu, backlight settings configured through the Main Menu take effect.

IMPORTANT When in the Main Menu, the LCD_BKLT_REM instruction will be disabled or ineffective.

KEY_READ_REM

This function block can be used to read key status on the Remote LCD module when the user-defined screen is active. When user-defined screen is not active, KEY_READ_REM instruction flags an error.

Note that the KEY_READ_REM instruction will always show key status as False if Push Button Key Read is disabled in Connected Components Workbench or the Remote LCD.

Backup and Restore

To initiate backup and restore through the REMLCD module, access the memory card by going to the Main Menu → Advanced Set → Memory Card.

See [Using microSD Cards on page 93](#) for information about project backup and restore on the microSD card.

Hardware Features, Installation, and Specifications

To learn about installation, hardware features, and specifications of the Micro800 Remote LCD, refer to the installation instructions, publication [2080-IN010](#), in the Literature Library.

Notes:

Using microSD Cards

This chapter provides a description of microSD card support on Micro820 controllers.

Topic	Page
Overview	93
Project Backup and Restore	94
Backup and Restore Directory Structure	95
Powerup Settings in ConfigMeFirst.txt	96
General Configuration Rules in ConfigMeFirst.txt	98
ConfigMeFirst.txt Errors	98
Data Log	99
Recipe	104
Quickstart Projects for Data Log and Recipe Function Blocks	108

The last section provides quickstart projects for the data log and recipe functions.

Overview

Micro820 controllers support microSD cards for the following purposes:

- Project backup and restore
- Data log and Recipe

IMPORTANT	For optimum performance, the microSD card should not be more than 90% full. Regularly check available space on your microSD card and ensure that the card is exclusively used for the Micro800 controller and no unnecessary files are present. Regularly delete old data log files and directories.
------------------	--

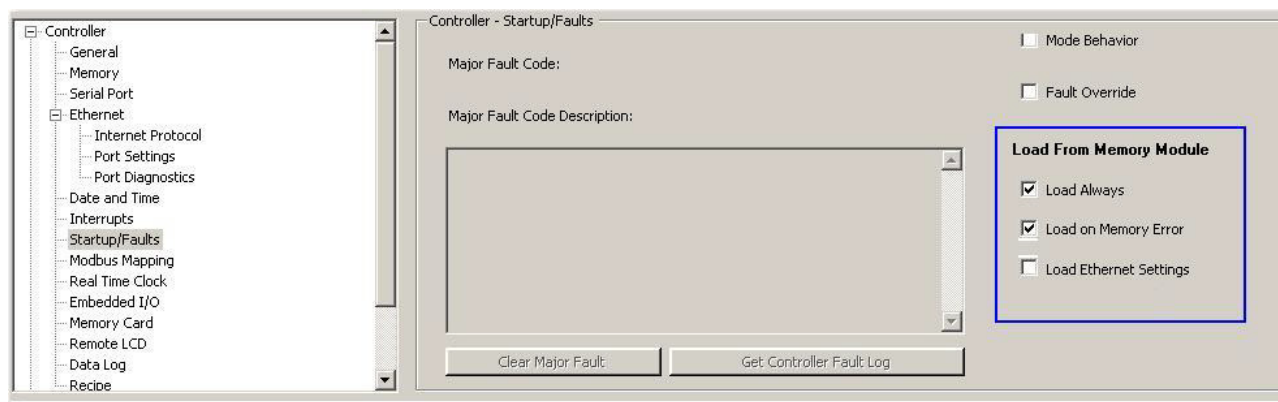
IMPORTANT	Do not remove the microSD card or power down while operations such as upload, download, delete, search, backup and restore are ongoing to prevent data loss. A blinking SD status LED indicates that these operations are ongoing. The SD status LED will not blink when flash upgrading the firmware from the microSD card.
------------------	---

IMPORTANT	To prevent data loss, recipe and data log function blocks must indicate Idle status before microSD card is removed.
------------------	---

Project Backup and Restore

Project backup and restore on Micro820 controllers are mainly supported through the microSD card. Both backup and restore can be initiated or manually triggered and configured through the Connected Components Workbench, the 2080-REMLCD module, and the ConfigMeFirst.txt file in the microSD card.

Backup and restore can only occur when the controller is in PROGRAM mode. On controller powerup, restore automatically occurs if the Load Always or Load on Memory Error option has been configured in Connected Components Workbench.



IMPORTANT To learn about restore and backup using the 2080-REMLCD module, see [Using the Micro800 Remote LCD on page 83](#).

To learn about restore and backup using the Connected Components Workbench, refer to the software Online Help.

IMPORTANT For Micro800 controllers that support microSD cards, IP protection of user project can only be achieved through the POU password protection mechanism in Connected Components Workbench (Developer Edition) and NOT via Controller Lock feature.

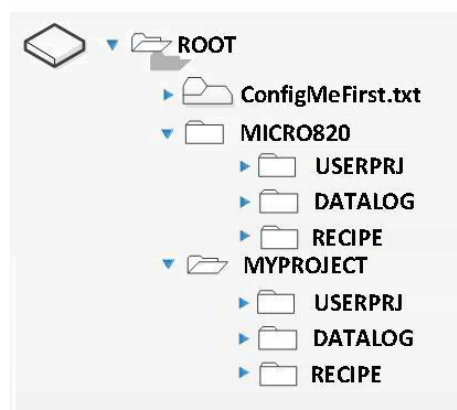
IMPORTANT If the Load Always setting is enabled and power is lost when restoring a project from the microSD card, the controller will attempt to load the project using the default project name and directory after power is restored. If your project is not using the default name and directory, the operation will fail and a fault occurs.

The microSD card stores the controller password in encrypted format. When the password is mismatched, the contents of the microSD card is not restored on the controller.

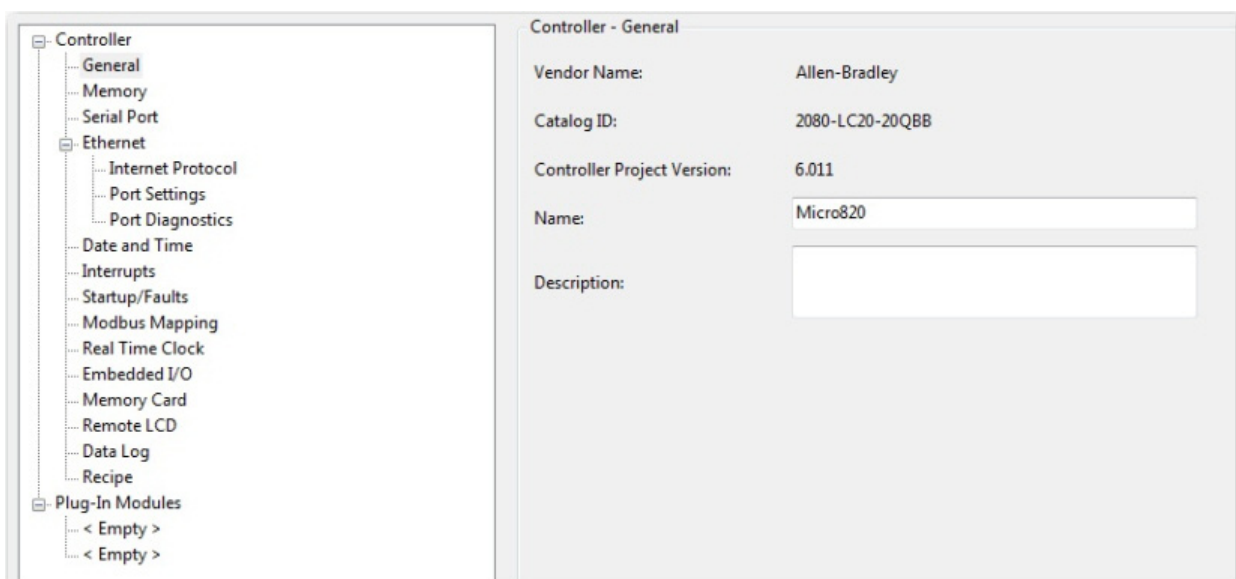
Backup and restore can be configured to trigger through the following ways:

Method	Backup	Restore
Online with Connected Components Workbench	Yes	Yes
2080-REMLCD	Yes	Yes
Project configuration on memory card at powerup	No	Load Always and/or Load on Memory Error options
ConfigMeFirst.txt at powerup	Yes (Through the [BKD] command)	Yes (Through the [RSD] command)

Backup and Restore Directory Structure



When a user project is backed up, a subdirectory named Micro820/USERPRJ is created on the microSD card. The folder name takes the name of the project specified in the General Page in Connected Components Workbench, which is Micro820 by default. However, if the ConfigMeFirst.txt file specifies a different subdirectory (example: MyProject), the project is backed up to that directory. See [General Configuration Rules in ConfigMeFirst.txt on page 98](#).



Project restore is done from the subdirectory specified in ConfigMeFirst.txt file or the Micro820/USERPRJ default folder, if none is specified in the ConfigMeFirst.txt file. The user needs to ensure that the directory is populated with correct contents before restoring.

The ConfigMeFirst.txt file is a configuration file stored on the microSD card that the user can optionally create to customize backup, restore, recipe and data log directories. The following sections include information on how to configure the ConfigMeFirst.txt properly.

IMPORTANT The Micro800 controller reports a major fault when project backup does not succeed because the memory card size is exceeded.

Powerup Settings in ConfigMeFirst.txt

On powerup, the Micro820 controller reads and carries out configuration settings described in the ConfigMeFirst.txt file, as shown in the following table.

ConfigMeFirst.txt Configuration Settings

Setting	Description
[FWFILE]	File path location of the firmware revision on the microSD card. The default location is in the following format: firmware\<catalog number>\<filename of firmware>
[FWBUS]	This setting is reserved and may be omitted.
[FWSLOT]	This setting is reserved and may be omitted.
[FWDOWN]	Sets whether to upgrade or downgrade the controller firmware from the current revision. 0 = Upgrade firmware; 1 = Downgrade firmware IMPORTANT: Firmware Upgrade will happen if [FWFILE] setting points to a newer version of firmware file compared to current firmware in the controller, irrespective of [FWDOWN] setting.
[PM]	Power up and switch to PROGRAM mode.
[CF]	Power up and attempt to clear fault.
[BKD = My Proj 1]	Power up and save the controller project into backup directory, My Proj 1\USERPRJ. Require extra power cycle to clear existing fault first using [CF] setting or other means.
[RSD = MyProj2]	Power up and read the project from restore directory MyProj2\USERPRJ into controller. Require extra power cycle to clear existing fault first using [CF] setting or other means. This setting overwrites UPD (or its default) load always or load on error restore function.
[UPD = My Proj]	For normal usage of backup and restore (that is, through Connected Components Workbench, 2080-REMLCD, Load Always, or Load on Memory Error settings), set the user project directory name. For example, My Proj, during powerup or when the microSD card is inserted. This directory is also used by data logging and recipe function.
[ESFD]	Embedded Serial Factory Defaults. Power up and revert embedded serial comms to factory defaults.
[IPA = xxx.xxx.xxx.xxx]	Power up and set IP address to xxx (must be numbers only).

ConfigMeFirst.txt Configuration Settings

Setting	Description
[SNM = xxx.xxx.xxx.xxx]	Power up and set subnet mask to xxx (must be numbers only).
[GWA = xxx.xxx.xxx.xxx]	Power up and set gateway address to xxx (must be numbers only).
[END]	End of setting. This setting is always required even when the ConfigMeFirst.txt file does not contain any other setting. The SD LED goes off when this setting is not present.

IMPORTANT Flash Upgrade Settings

With Connected Components Workbench release 8.0 onwards, you can flash upgrade your Micro820 controller from the microSD card in addition to using ControlFLASH. See [Flash Upgrade From MicroSD Card on page 150](#) for instructions.

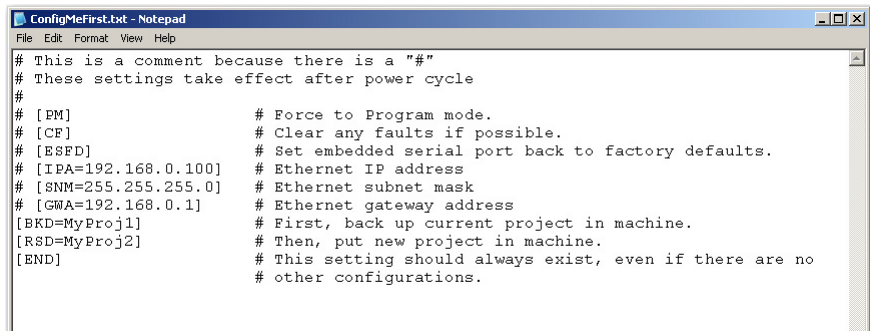
- [FWFILE], [FWBUS], [FWSLOT], and [FWDOWN] settings must be placed at the beginning of the file.

IMPORTANT Directory Settings

- If no directory has been specified in the ConfigMeFirst.txt file, then backup and restore will occur in the controller name directory (Micro820/USERPRJ, by default).
- If [UPD] is configured in the ConfigMeFirst.txt file, then backup and restore will occur in the [UPD] directory specified.
- [BKD] setting is implemented even when the controller is locked or password protected.
- [BKD] directory is automatically created if it does not yet exist.

IMPORTANT Powerup Network Parameter Settings

- [IPA], [SNM] and [GWA] follow the general IP configuration rules.
- [IPA], when set in ConfigMeFirst.txt, should always be configured with a valid [SNM] and vice versa.
- When optional [GWA] setting is used, make sure that [IPA] and [SNM] settings are also present in ConfigMeFirst.txt.
- The [ESFD], [IPA], [SNM], and [GWA] settings overwrite the respective communication settings from project restore due to [RSD], Load Always or Load on Memory Error.

Sample ConfigMeFirst.txt File


```

# This is a comment because there is a "#"
# These settings take effect after power cycle
#
# [PM]                # Force to Program mode.
# [CF]                # Clear any faults if possible.
# [ESFD]              # Set embedded serial port back to factory defaults.
# [IPA=192.168.0.100] # Ethernet IP address
# [SNM=255.255.255.0] # Ethernet subnet mask
# [GWA=192.168.0.1]  # Ethernet gateway address
[BKD=MyProj1]        # First, back up current project in machine.
[RSD=MyProj2]        # Then, put new project in machine.
[END]                # This setting should always exist, even if there are no
                    # other configurations.

```

General Configuration Rules in ConfigMeFirst.txt

- All settings must be in upper case and enclosed in brackets [].
- Each line must contain only one setting.
- Settings must always appear first in a line.
- Comments are started with the # symbol.
- No action related to the setting will be carried out when the setting does not exist, or a # symbol appears before the setting (example, #[PM]).

ConfigMeFirst.txt Errors

The SD status LED goes off when the microSD card is inserted during PROGRAM or RUN mode (or on powerup) and the ConfigMeFirst.txt file is either unreadable or invalid. The ConfigMeFirst.txt file will be invalid when it has the following errors:

- unrecognized setting (that is, the first three configuration rules have not been followed),
- the setting parameters after the = symbol is invalid, does not exist, or out of range,
- the same setting exists twice or more,
- one or more non-setting characters exist within the same bracket,
- space in between setting characters (example, [P M]), or
- space in between IP address, subnet mask, and gateway address (for example, xxx. x xx.xxx.xxx)
- only one of the network parameter settings ([IPA], [SNM], or [GWA]) is assigned
- [END] setting does not exist (even if there are no other settings in the configuration file).

The microSD card becomes unusable until the ConfigMeFirst.txt file becomes readable or the errors are corrected.

Data Log

The data logging feature allows you to capture global and local variables with timestamp from the Micro800 controller into the microSD card. You can retrieve the recorded datasets on the microSD card by reading the contents of the microSD card through a card reader or by doing an upload through the Connected Components Workbench software.

A maximum number of 10 datasets is supported for a Micro820 program. Each dataset can contain up to 128 variables, with a maximum of four (4) data string variables per dataset. String variables can have a maximum of 252 characters. All datasets are written to the same file. For more information on how data logs are stored on the microSD card, see the [Data Log Directory Structure on page 100](#).

You can retrieve data log files from the microSD card using a card reader or by uploading the data logs through Connected Components Workbench.

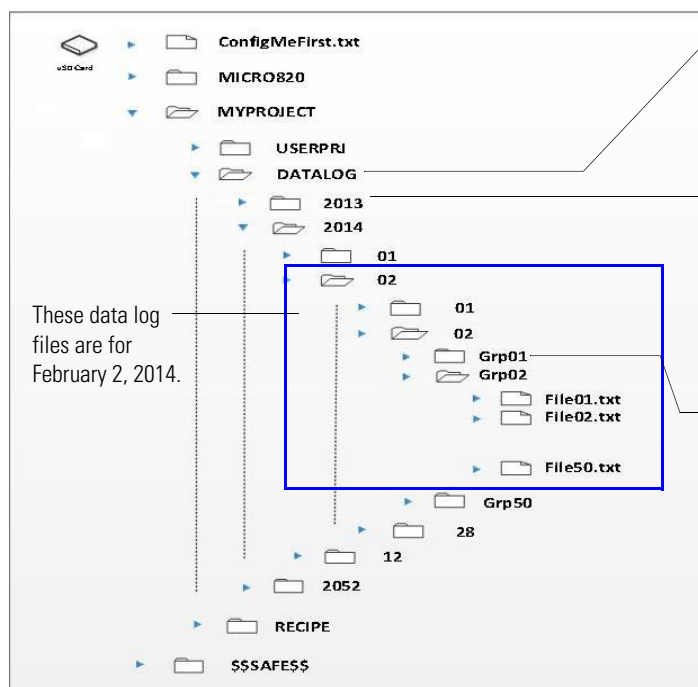
IMPORTANT Uploading data log files in PROGRAM mode is recommended for optimum performance and to prevent file access conflict. For example, if the data log instruction is executing, Connected Components Workbench will not upload the last data log file.

See the sample quickstart project to get you started on the Data Log feature, [Use the Data Log Feature on page 109](#).

IMPORTANT Data log execution time depends on the user application and its complexity. Users are advised to data log **no faster than every two seconds** for typical applications. Note that housekeeping takes at least 5 ms per program scan. See [Program Execution in Micro800 on page 71](#) for more information on program scan and execution rules and sequence. See also [Data Log – Data Payload vs. Performance Time on page 130](#).

IMPORTANT Note that in cases where there are simultaneous RCP and DLG function block execution or uploads/downloads/searches, the activities are queued up and handled one by one by the program scan. Users will notice a slowdown in performance in these cases.

Data Log Directory Structure



The DATALOG folder is created under the current project directory in the microSD card. In this example, the current project directory is MYPROJECT. By default, the current project directory name is taken from the downloaded project's controller name or from the ConfigMeFirst.txt. See [ConfigMeFirst.txt Configuration Settings on page 96](#).

Subdirectories are also created following the controller RTC timestamp. This means that if RTC date at the time of function block execution is February 02, 2013, the subfolder 2013 is created under DATALOG. Under the 2013 folder, the subfolder 02 (which stands for the month of February) is created. Under 02, another subfolder 02 is created, corresponding to the current date.

- Under the current working folder, the subfolder Grp01 is created. A maximum of 50 Grpxxx folders can be generated on the microSD card per day.

Under the current Grpxxx working folder, the data log file File01.txt is created. Once this file reaches more than 4 KB, another file, File02.txt, is automatically created to store data. The file size is kept small in order to minimize data loss in case the card is removed or when there is unexpected power off.

Each Grp xx folder can accommodate up to 50 files. This means that, for example, when the Grp01 folder already stores 50 files, a new folder Grp02 is automatically created to store the next data log files for that day. This automatic folder and file generation goes on until the Grp xx folder reaches 50 for that day.

When a microSD card is inserted, the DLG function block looks for the last Grp xx folder and file xx .txt file, and proceeds to do the data logging based on that information.

The following table summarizes data logging performance on Micro820 controllers.

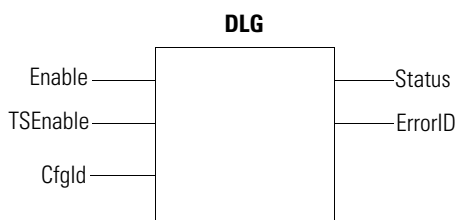
Data Log Specifications

Attribute	Value	
Maximum datasets	10	All datasets are stored in the same file.
Maximum variables per dataset	128	Configured in Connected Components Workbench software.
Minimum size per file	4 KB	
Maximum files per Grpxx folder ⁽¹⁾	50	When directory is full, a new directory is automatically created in RUN mode.
Maximum files (Filexx.txt) per day	50	When file reaches maximum size, a new file is automatically created in RUN mode.
Typical data per day	10 MB	

- (1) Once the data log limits is reached (that is, 50 Grpxx folders per day, then an error (ErrorID 3: DLG_ERR_DATAFILE_ACCESS) is returned.

Data Log Function (DLG) Block

The data logging function block lets a user program to write run-time global values into the data logging file in microSD card.



DLG Input and Output Parameters

Parameter	Parameter Type	Data Type	Description
Enable	INPUT	BOOL	Data logging write function enable. On rising edge (that is, Enable value is triggered from low to high), the function block executes. The precondition for execution is that the last operation has completed.
TSEnable	INPUT	BOOL	Date and timestamp logging enable flag.
Cfgld	INPUT	USINT	Configured dataset (DSET) number (1...10).
Status	OUTPUT	USINT	Data logging function block current status.
ErrorID	OUTPUT	UDINT	Error ID if DLG Write fails.

DLG Function Block Status

Status Code	Description
0	Data logging IDLE status.
1	Data logging BUSY status.
2	Data logging COMPLETE SUCCEED status.
3	Data logging COMPLETE ERROR status.

DLG Function Block Errors

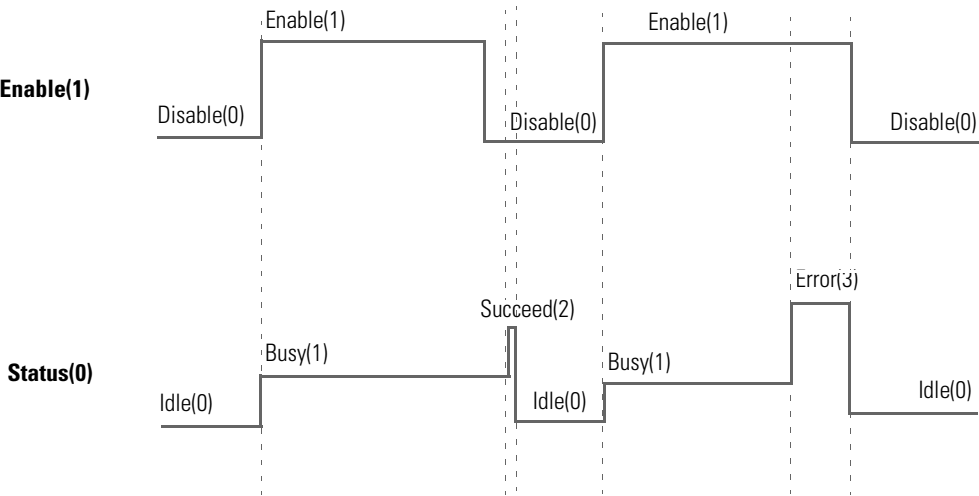
Status Code	Name	Description
0	DLG_ERR_NONE	No error.
1	DLG_ERR_NO_SDCARD	microSD card is missing.
2	DLG_ERR_RESERVED	Reserved.
3	DLG_ERR_DATAFILE_ACCESS	Error accessing data log file in microSD card.
4	DLG_ERR_CFG_ABSENT	Data log configuration file is absent.
5	DLG_ERR_CFG_ID	Configuration ID is missing in data log configuration file.
6	DLG_ERR_RESOURCE_BUSY	Same Configuration ID is used with other data log function block call at the same time

DLG Function Block Errors

Status Code	Name	Description
7	DLG_ERR_CFG_FORMAT	Data log configuration file format is wrong.
8	DLG_ERR_RTC	Real time clock is invalid.
9	DLG_ERR_UNKNOWN	Unspecified error has occurred.

IMPORTANT File access error will be returned during DLG function block execution when card is full.

Data Log Function Block Timing Diagram



IMPORTANT Data Log Function Block Execution

- There are three possible states for the Data Log function block: Idle, Busy and Complete (which includes Complete with Succeed and Complete with Error).
- For one Data Log function block execution, the typical status starts from Idle, then Busy and finishes with Complete. To trigger another function block execution, the status needs to go back to Idle first.
- Idle status changes to Busy status only when Enable input signal is in rising edge. Complete status enters Idle status when Enable input signal is Disable status only.
- TSEnable and Cfgld input parameters are only sampled at Enable input parameter's rising edge when a new function block execution starts. During function block execution, the input parameters of TSEnable and Cfgld are locked and any changes are ignored.
- When execution completes, the status changes from Busy to Complete. At this stage, if input Enable is False, status changes to Idle after indicating Complete for exactly one scan time. Otherwise function block status is kept as Complete until input Enable changes to False.
- The data log file can only be created by the DLG instruction block. Connected Components Workbench can only upload and delete the data log file.
- There are separators in between every data variable in the data file which is defined during configuration in Connected Components Workbench.
See [Supported Data Types for Data Log and Recipe Function Blocks on page 103](#).
- Data variable values are sampled when data logging function block is in Busy state. However, data logging file is only created when data logging function block is in Complete state.

Supported Data Types for Data Log and Recipe Function Blocks

Data Type	Description	Example format in output data log file
BOOL ⁽¹⁾	Logical Boolean with values TRUE and FALSE	0: FALSE 1: TRUE)
SINT	Signed 8-bit integer value	-128, 127
INT	Signed 16-bit integer value	-32768, 32767
DINT	Signed 32-bit integer value	-2147483648, 2147483647
LINT	Signed 64-bit integer value	-9223372036854775808, 9223372036854775807
USINT(BYTE)	Unsigned 8-bit integer value	0, 255
UINT(WORD)	Unsigned 16-bit integer value	0, 65535
UDINT(DWORD)	Unsigned 32-bit integer value	0, 4294967295
ULINT(LWORD)	Unsigned 64-bit integer value	0, 18446744073709551615

Supported Data Types for Data Log and Recipe Function Blocks

Data Type	Description	Example format in output data log file
REAL	32-bit floating point value	-3.40282347E+38, +3.40282347E+38
LREAL	64-bit floating point value	-1.7976931348623157E+308, +1.7976931348623157E+308
STRING ⁽²⁾	character string (1 byte per character)	"Rotation Speed"
DATE ⁽¹⁾	Unsigned 32-bit integer value	1234567 (Date variables are stored as 32-bit words, a positive number of seconds beginning at 1970-01-01 at midnight GMT.)
TIME ⁽¹⁾	Unsigned 32-bit integer value	1234567 (Time variables are stored as 32-bit words, positive number of milliseconds.)

- (1) BOOL, DATE, TIME data variables are presented in decimal digital format in the microSD Card. Users have the option to convert this format to a more friendly format. For example, use ANY_TO_STRING function block to convert BOOL data type (0, 1) to FALSE or TRUE. You can similarly do the same for DATE and TIME data types. DATE data type is presented in differential decimal digital value between system baseline time (1970/01/01,00:00:00) and current date value. Unit is millisecond. Time should be absolute time value. Unit is second.
- (2) String data variables are enclosed in double quotation marks in the data log file. The example below shows DSET1 using string variables and DSET2 using integers.

```
DSET1, "Temperature", "Humidity", "Pressure"
DSET2, 30, 50, 125
```

Recipe

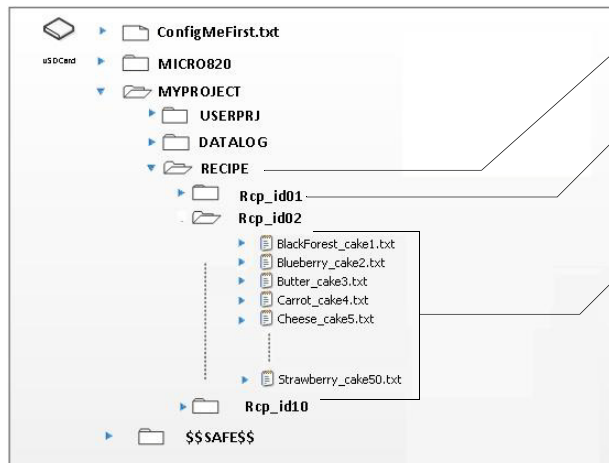
Micro820 controllers support the Recipe feature and allows users to store and load a list of data to and/or from recipe data files using the RCP instruction. It also allows users to download, upload, and delete Recipe data on the microSD card through Connected Components Workbench.

A maximum number of 10 recipe sets is supported for a Micro820 program. Each recipe can contain up to 128 variables, with a maximum of four (4) data string variables per recipe. String variables can have a maximum of 252 characters. Variations of the recipe are stored in separate files with unique file names. For more information on how recipes are stored on the microSD card, see the [Recipe Directory Structure on page 105](#).

Recipe Specifications

Attribute	Value	
Maximum number of recipe sets	10	Recipe sets are stored in 10 directories (Rcp_Id01...Rcp_Id10) with a maximum number of 50 recipe files in each directory.
Maximum number of recipes in each set	50	
Maximum number of variables per recipe	128	Configured in Connected Components Workbench software.
Maximum bytes per recipe file	4 KB	

Recipe Directory Structure



On first execution of RCP, it creates the RECIPE folder under the current project directory on the microSD card.

It also creates 10 subdirectories for each recipe set with a name following the CfgID input value (1...10). If the CfgID value is 1, then the subfolder Rcp_Id01 is created.

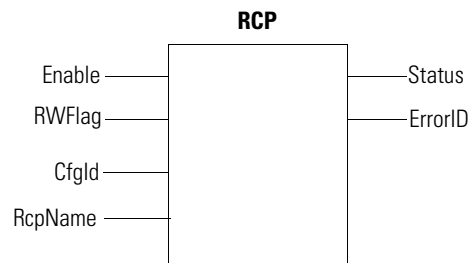
Recipe files are then created/written into the folder, with file names that correspond to the input value of RcpName parameter for the RCP function block, as configured in Connected Components Workbench. Each Recipe set can contain up to 50 recipe files or variations. Filenames for recipe files should not exceed 30 characters.

Recipe Configuration and Retrieval

You can retrieve recipe files from the microSD card using a card reader or by uploading and downloading the recipe sets through Connected Components Workbench.

Recipe Function (RCP) Block

The RCP function block allows a user program to read variable values from an existing recipe data file which is in the recipe folder of the microSD card and update run-time global or local variable values in the controller. The RCP function block also allows the user program to write run-time global or local variable values from smaller controller into the recipe data file in the microSD card.



RCP Input and Output Parameters

Parameter	Parameter Type	Data Type	Description
Enable	INPUT	BOOL	Recipe read/write function enable. If Rising Edge (Enable is triggered from "low" to "high"), starts recipe function block and the precondition is that last operation is completed.
RWFlag	INPUT	BOOL	TRUE: Recipe write data variables to recipe files into the microSD card. FALSE: Recipe reads saved data variables from the microSD card and update these variables accordingly.
CfgId	INPUT	USINT	Recipe set number (1...10).
RcpName	INPUT	STRING	Recipe data filename (maximum 30 characters).
Status	OUTPUT	USINT	Current state of Recipe function block.
ErrorID	OUTPUT	UDINT	Detailed error ID information if RCP read/write fails.

RCP Function Block Status

Status Code	Description
0	Recipe Idle status.
1	Recipe Busy status.
2	Recipe Complete Succeed status.
3	Recipe Complete Error status.

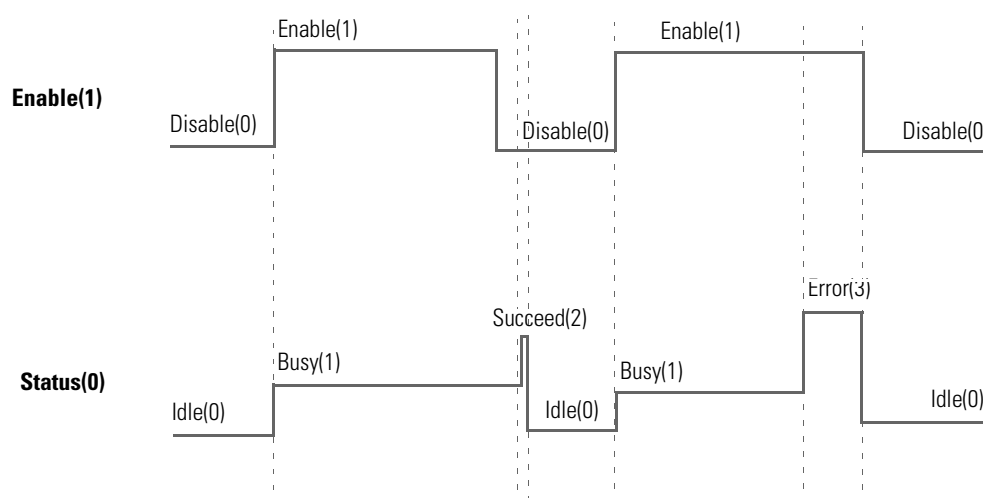
RCP Function Block Errors

Error ID	Error name	Description
0	RCP_ERR_NONE	No error.
1	RCP_ERR_NO_SD CARD	microSD card is absent.
2	RCP_ERR_DATAFILE_FULL	Recipe files exceed maximum number of files per recipe set folder.
3	RCP_ERR_DATAFILE_ACCESS	Error to access recipe data file in microSD card.
4	RCP_ERR_CFG_ABSENT	Recipe configuration file is absent.
5	RCP_ERR_CFG_ID	Configure ID is absent in recipe configuration file.
6	RCP_ERR_RESOURCE_BUSY	The Recipe operation resource linked to this Recipe ID is used by another function block operation.
7	RCP_ERR_CFG_FORMAT	Recipe configuration file format is invalid.
8	RCP_ERR_RESERVED	Reserved.
9	RCP_ERR_UNKNOWN	Unspecified error has occurred.
10	RCP_ERR_DATAFILE_NAME	Recipe data file name is invalid.
11	RCP_ERR_DATAFOLDER_INVALID	Recipe dataset folder is invalid.

RCP Function Block Errors

Error ID	Error name	Description
12	RCP_ERR_DATAFILE_ABSENT	Recipe data file is absent.
13	RCP_ERR_DATAFILE_FORMAT	Recipe data file contents are wrong.
14	RCP_ERR_DATAFILE_SIZE	Recipe data file size is too big (>4K).

IMPORTANT File access error will be returned during RCP function block execution when card is full.

Recipe Function Block Timing Diagram

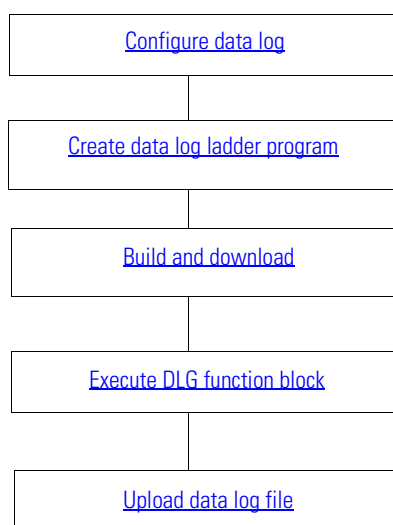
IMPORTANT RCP Function Block Execution

- There are three possible states for Recipe function block: Idle, Busy, Complete (Complete with Succeed and Complete with Error)
 - For one Recipe function block execution, the typical status starts from Idle then Busy and finishes with Complete. To trigger another function block execution, the status needs to go back to Idle first.
 - Idle status changes to Busy status only when Enable input signal is in rising edge. Complete status enters Idle status when Enable input signal is on Disable status.
 - RWFlag, CfgId and RcpName input parameters are only sampled at Enable input parameter's rising edge when a new function block execution starts. During function block execution, input parameters of RWFlag, CfgId and RcpName are locked and any changes are ignored.
 - When the function block execution finishes, the function block status changes from Busy to Complete. At this stage, if input Enable is False, function block status changes to Idle after staying as Complete for exactly one scan time. Otherwise, function block status remains Complete until input Enable changes to False.
 - Recipe function block file name supports a maximum of 30 bytes in length, and only supports upper and lower case letters Aa...Zz, numbers 0...9 and underscore (_).
 - The RcpName input parameter does not allow file extension (for example, .txt) to be added to its value. The recipe data file is written to the microSD card with the .txt extension.
 - There are separators in between every data variable in the recipe data file which is defined during configuration in Connected Components Workbench. Redundant tab, space, carriage return and line feed characters are strictly not allowed.
See [Supported Data Types for Data Log and Recipe Function Blocks on page 103](#).
 - Double quotes are not allowed within a string in a recipe file.
-

Quickstart Projects for Data Log and Recipe Function Blocks

The following sample quickstart projects provide step-by-step instructions on how to use the Data Log and Recipe function blocks in Connected Components Workbench to generate and manage your recipe files and data logs.

Use the Data Log Feature

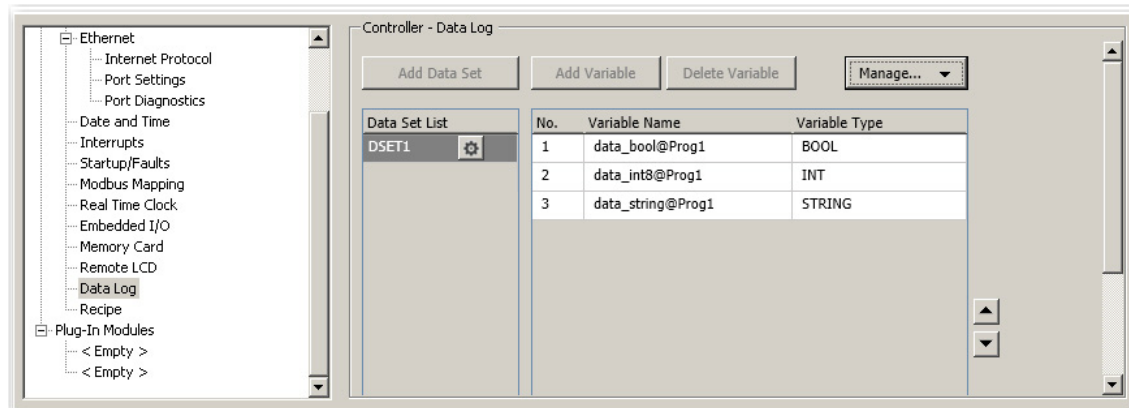


Configure data log

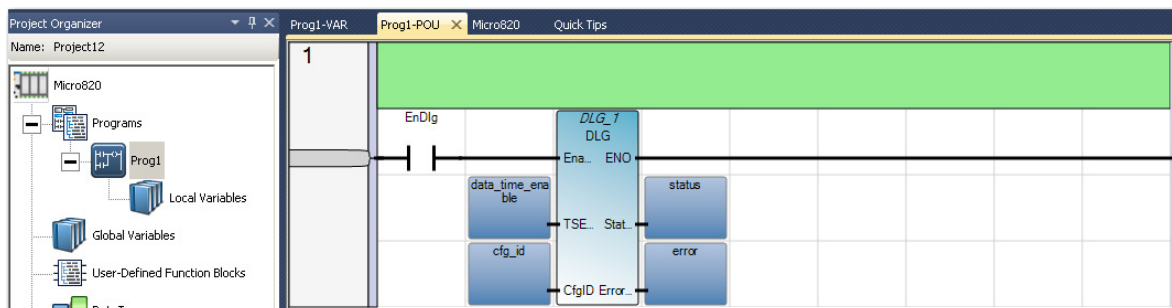
1. In Connected Components Workbench, go to the Properties pane to configure your data log.
2. Select Data Log. Click Add Dataset to add a dataset. Note that each dataset will be stored in the same file. You can add up to 10 datasets per configuration.
3. Click Add Variable to add variables to the dataset. You can add up to 128 variables to each dataset.
For this quickstart sample project, add the following variables that you have previously created to Dataset 1.

Local Variables

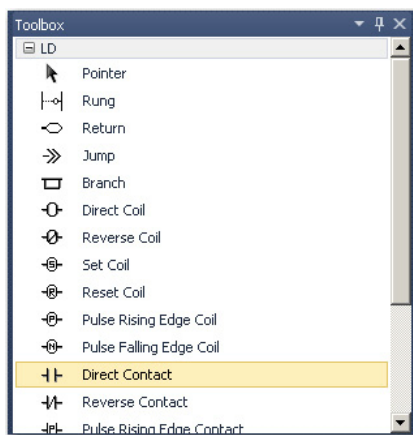
Variable Name	Data Type
data_bool	BOOL
data_int8	INT
data_string	STRING



Create data log ladder program

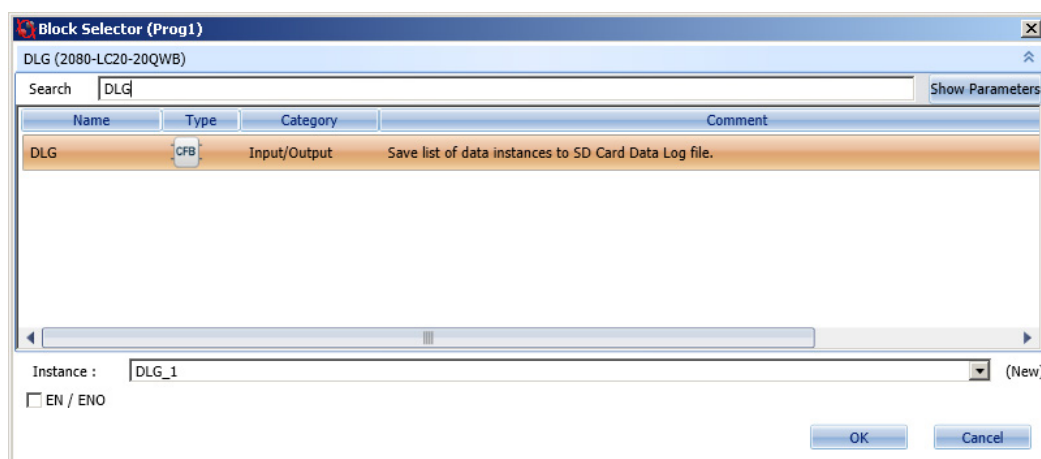


1. Launch Connected Components Workbench. Create a user program for your Micro820 controller.
2. Right-click Programs. Select Add New LD: Ladder Diagram. Name the Program (for example, Prog1).
3. From the Toolbox, double-click Direct Contact to add it to the rung.

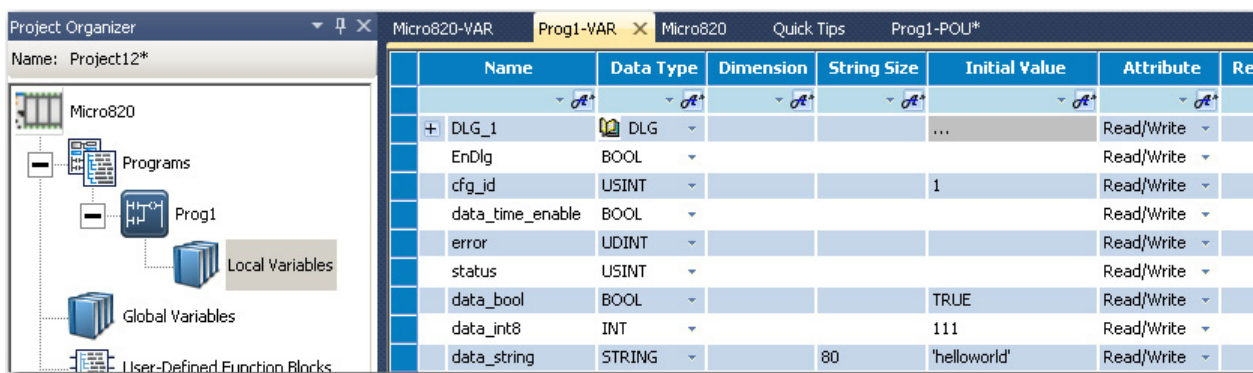


4. From the Toolbox, double-click Block to add it to the rung.

- On the Block Selector window that appears, type DLG to filter the DLG function block from the list of available function blocks. Click OK.



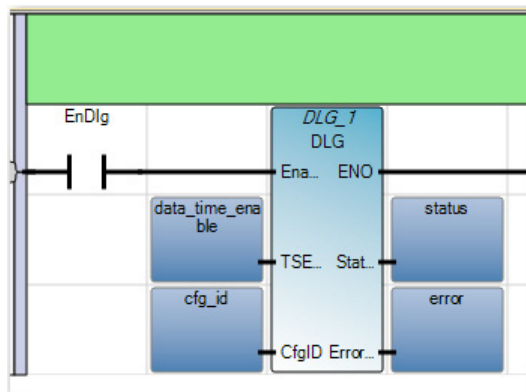
- Create the following local variables for your project.



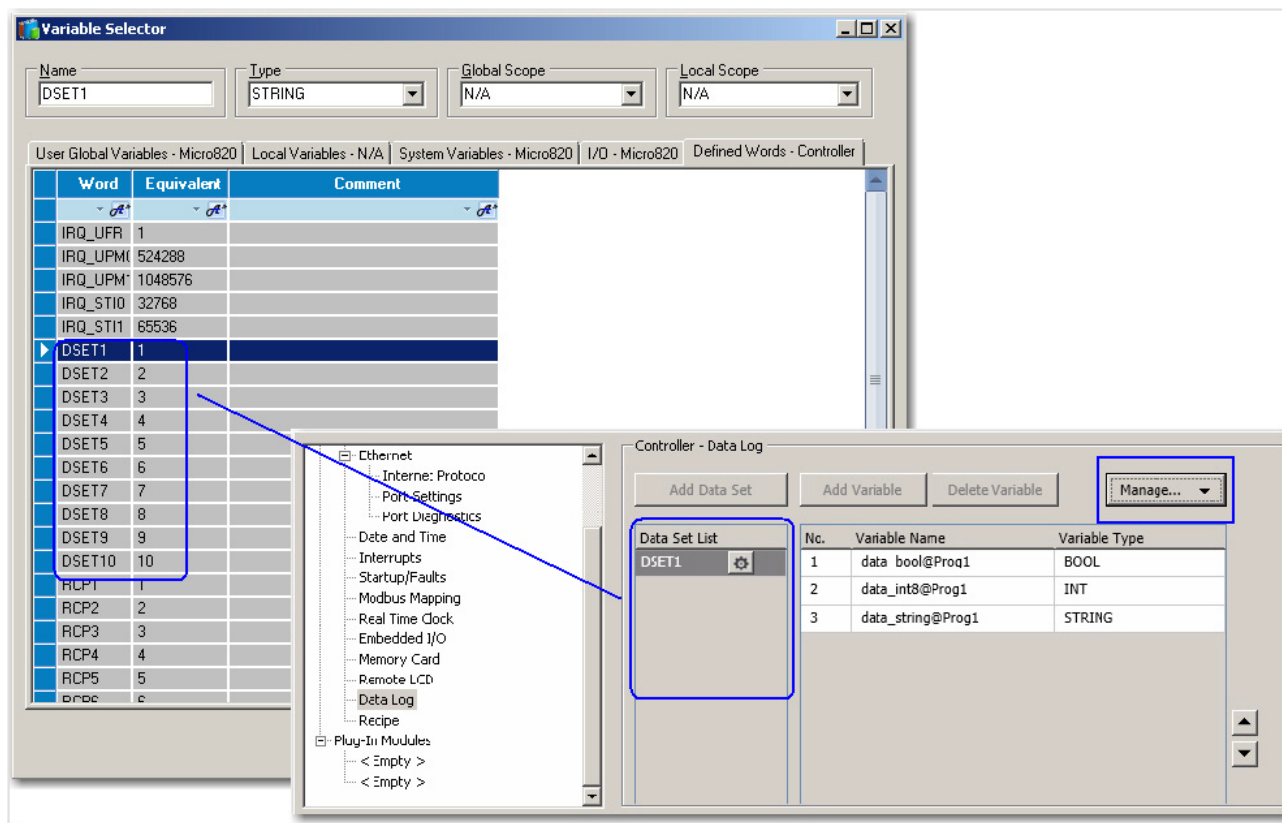
Local Variables

Variable Name	Data Type
EnDlg	BOOL
cfg_id	USINT
data_time_enable	BOOL
error	UDINT
status	USINT
data_bool	BOOL
data_int8	INT
data_string	STRING

7. Assign the variables to the DLG input and output parameters as follows:



Note: For CfgID input parameter, you can choose a predefined variable by choosing from the Defined Words in Connected Components Workbench. To do so, click the CfgID input box. From the Variable Selector window that appears, click the Defined Words tab and choose from the list of defined words (for example, DSET1 which corresponds to DSET1 in your recipe configuration). See the following screenshot.

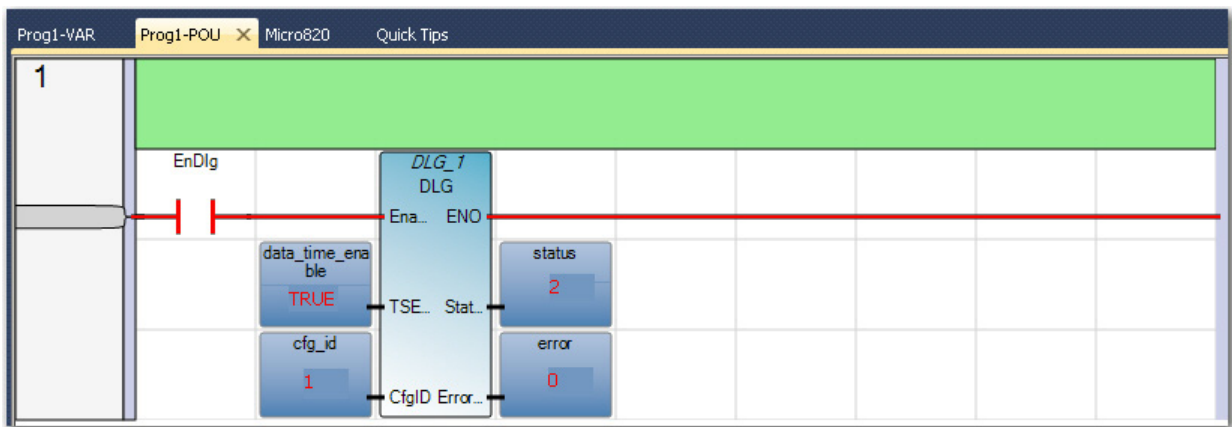


Build and download

After configuring data log properties, build the program and download to the controller.

Execute DLG function block

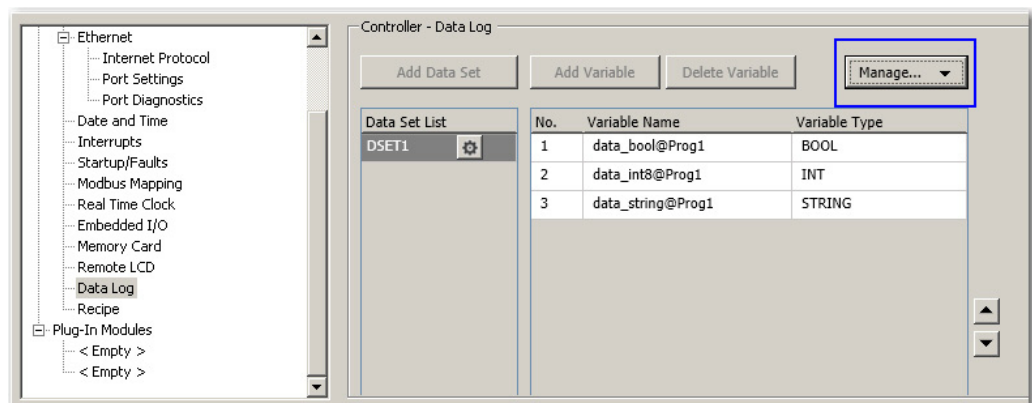
Execute the DLG function block. Notice the Status output go from 0 (Idle) to 1 (Enable), and 2 (Succeed).



Upload data log file

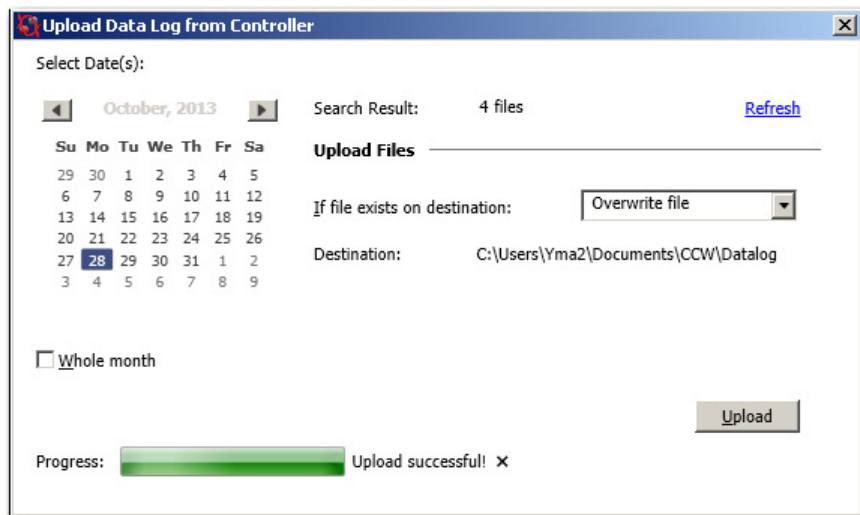
You can retrieve data log files from the microSD card using a card reader or by uploading the data logs through Connected Components Workbench.

1. To use the Upload feature, go to the Properties section of your project in Connected Components Workbench.
2. Select Data Log. Click Manage and then choose Upload.



IMPORTANT The Manage button is not available in DEBUG mode. You need to stop DEBUG mode to use the Manage button to upload data log files. Uploading data log files in PROGRAM mode is recommended for performance and file locking reasons.

- From the Upload window that appears, select the date of the data log files that you would like to upload. You can upload data logs for the entire month by clicking Whole Month option button.

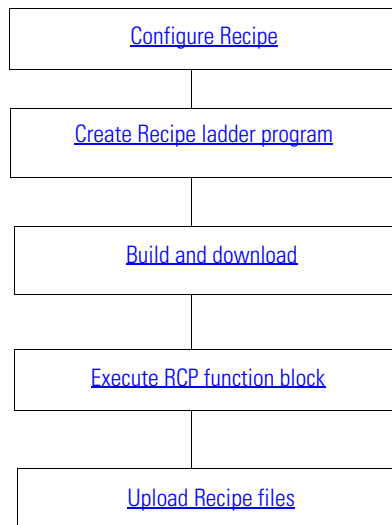


- If the file already exists in your destination folder, select whether you would like to Overwrite file, Skip file, or Preserve both files.
- Click Upload. The progress bar should tell you whether the upload is successful or not.

IMPORTANT Do not take out the microSD card from the slot while data is being written or retrieved from the card. Ongoing write and retrieval operations are indicated by a flashing SD status LED.

IMPORTANT For better data log file management, you can use a third-party tool or DOS CMD to merge all your data log files into a single file and import as a CSV file in Excel.

Use the Recipe Feature



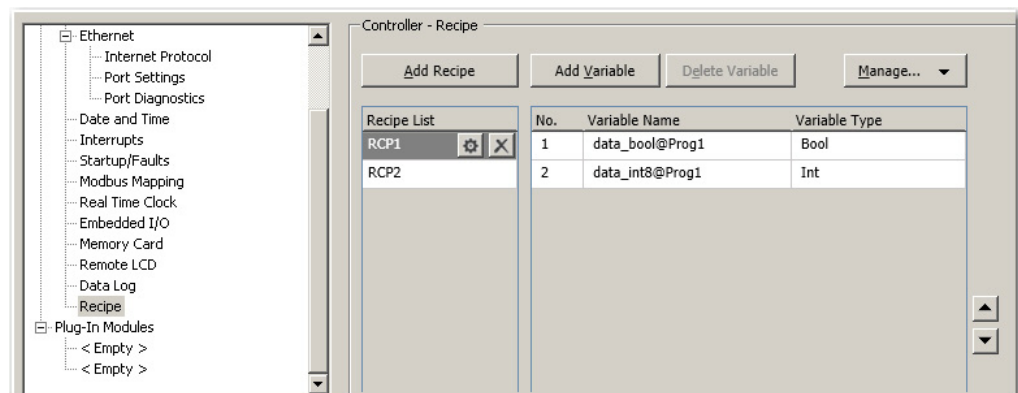
Configure Recipe

1. In Connected Components Workbench, go to the Properties pane to configure Recipe.
2. Select Recipe. Click Add Recipe to add a recipe. Note that each recipe will be stored in separate files. You can add up to 10 recipes per configuration.
3. Click Add Variable button to add variables to the recipe. You can add up to 128 variables to each recipe.

For this quickstart sample project, add the following variables that you have previously created to RCP 1:

Local Variables

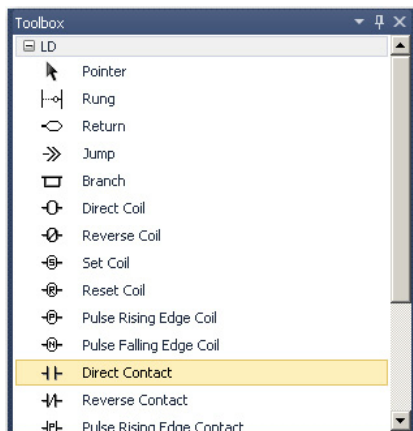
Variable Name	Data Type
data_bool	BOOL
data_int8	INT



Create Recipe ladder program

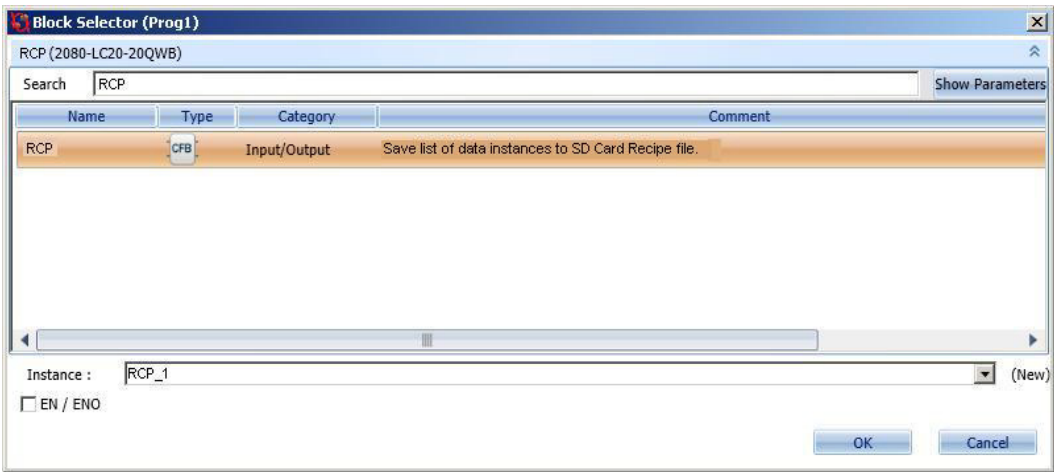


1. Launch Connected Components Workbench. Create a user program for your Micro820 controller.
2. Right-click Programs. Select Add New LD: Ladder Diagram. Name the Program (for example, Prog2).
3. From the Toolbox, double-click Direct Contact to add it to the first rung.



4. From the Toolbox, double-click Block to add it to the rung.

5. On the Block Selector window that appears, type RCP to filter the Recipe function block from the list of available function blocks. Click OK.



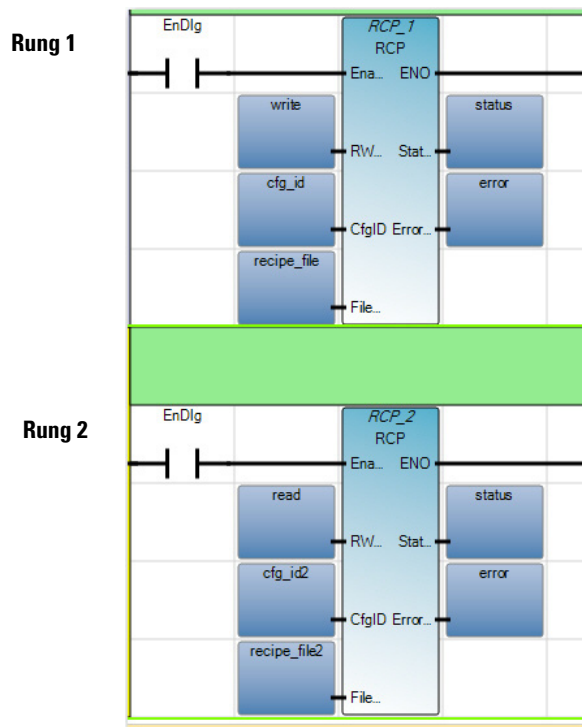
6. From the Toolbox, double-click rung to add another rung.
7. Add a Direct Contact and RCP function block to this second rung by following steps 3...5.
8. Create the following local variables for your program, in addition to the ones that you have already created for data log.

cfg_id2	USINT	2	Read/Write	<input type="checkbox"/>
+ RCP_2	RCP	...	Read/Write	<input type="checkbox"/>
recipe_file	STRING	80	'MyFirstRecipe'	Read/Write <input type="checkbox"/>
recipe_file2	STRING	80	'MySecondRecipe'	Read/Write <input type="checkbox"/>
read	BOOL	FALSE	Read/Write	<input type="checkbox"/>
write	BOOL	TRUE	Read/Write	<input type="checkbox"/>

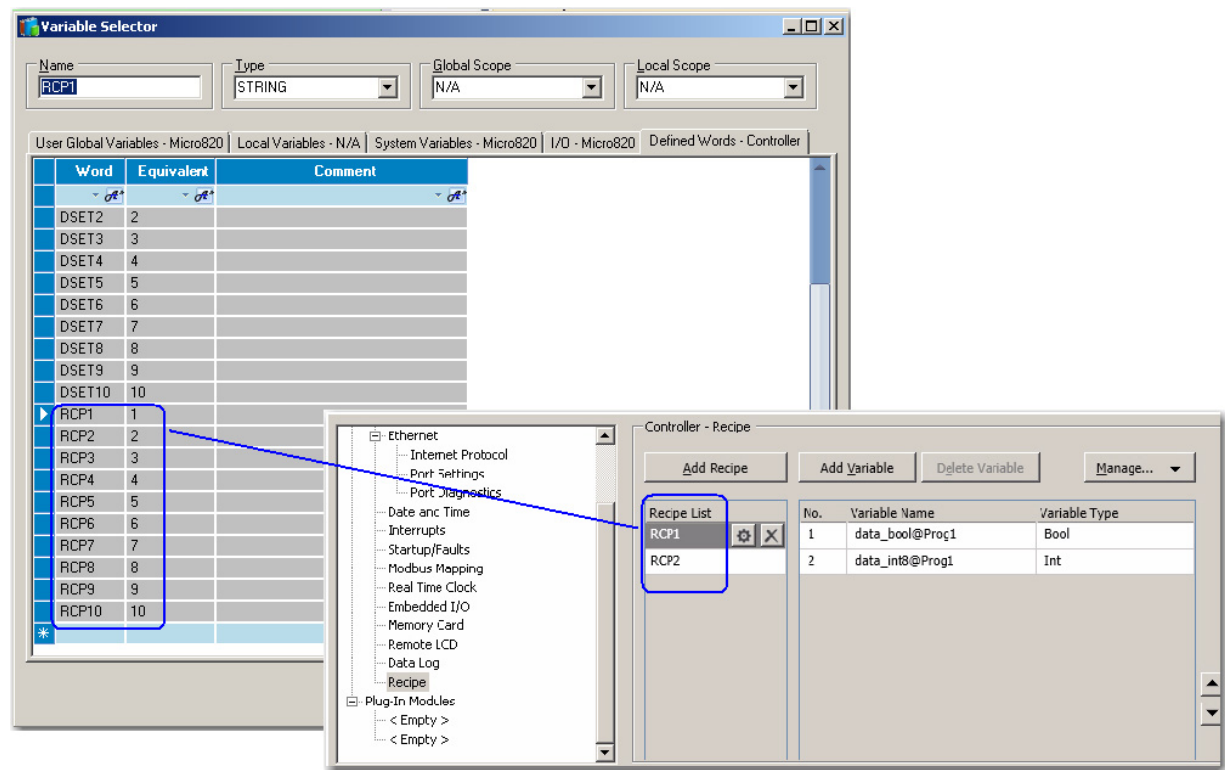
Local Variables

Variable Name	Data Type
recipe_file	STRING
recipe_file2	STRING
cfg_id2	USINT
read	BOOL
write	BOOL

9. Assign the variables to the RCP input and output parameters as follows:



Note: For CfgID input parameter, you can choose a predefined variable by choosing from the Defined Words in Connected Components Workbench. To do so, click the CfgID input box. From the Variable Selector window that appears, click the Defined Words tab and choose from the list of defined words (for example, RCP1 which corresponds to RCP1 in your recipe configuration). See the following screenshot.

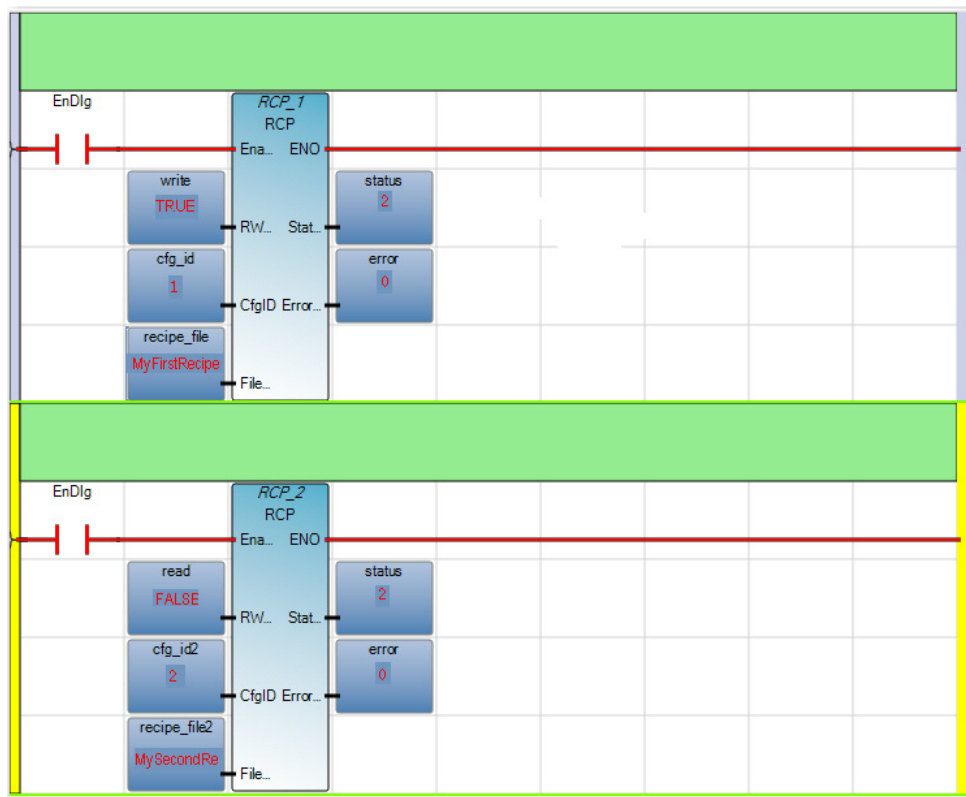


Build and download

After configuring Recipe, build the program and download to the controller.

Execute RCP function block

Execute the RCP function block. Notice the Status output go from 0 (Idle) to 1 (Enable), and 2 (Succeed).

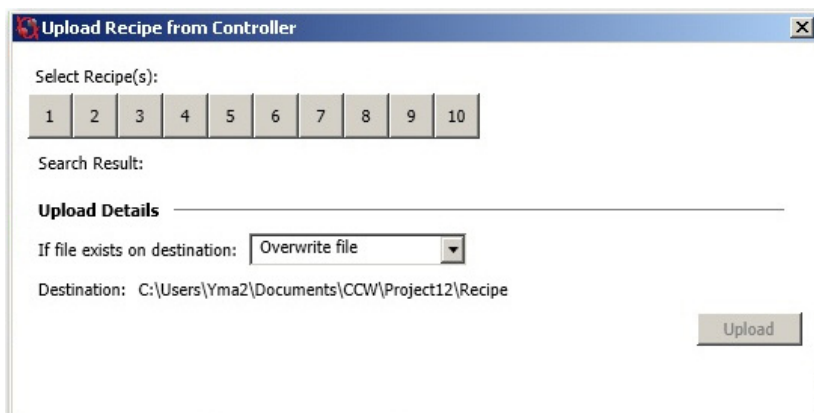


Upload Recipe files

You can retrieve recipe files from the microSD card using a card reader or by uploading the recipe files through Connected Components Workbench.

1. To use the Upload feature, go to the Properties section of your project in Connected Components Workbench.
2. Select Recipe. Click Manage and then choose Upload.
Through the Manage button, you can also choose to Download and Delete recipe files.

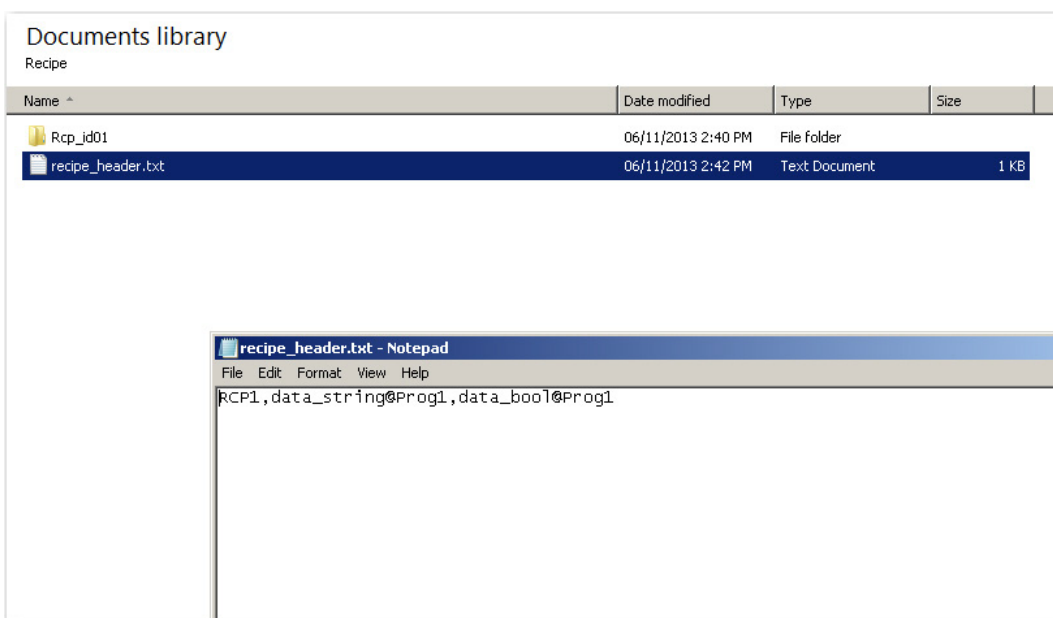
- From the Upload window that appears, select the batch of recipe files that you would like to upload.



- If the file already exists in your destination folder, select whether you would like to Overwrite file, Skip file, or Preserve both Files.
- Click Upload. The progress bar should tell you whether the upload is successful or not.

IMPORTANT Do not take out the microSD card from the slot while data is being written or retrieved from the card. Ongoing write and retrieval operations are indicated by a flashing SD status LED.

A recipe header file will be saved with the uploaded recipes.



Notes:

Specifications

General Specifications

The Micro820 controllers have the following specifications and certifications.

General Specifications

Attribute	2080-LC20-20AWB(R)	2080-LC20-20QBB(R)	2080-LC20-20QWB(R)
Number of I/O	12 inputs, 8 outputs		
Dimensions HxWxD	90 x 104 x 75 mm (3.54 x 4.09 x 2.95 in.)		
Shipping weight, approx.	0.38 kg (0.83 lb)		
Wire size	For fixed terminal blocks:		
		Min	Max
	Solid	0.14 mm ² (26 AWG)	2.5 mm ² (14 AWG)
	Stranded	0.14 mm ² (26 AWG)	1.5 mm ² (16 AWG)
	rated @ 90 °C (194 °F) insulation max		
	For removable terminal blocks:		
		Min	Max
	Solid and Stranded	0.2 mm ² (24 AWG)	2.5 mm ² (14 AWG)
	rated @ 90 °C (194 °F) insulation max		
	For RS232/RS485 serial port:		
		Min	Max
	Solid	0.14 mm ² (26 AWG)	1.5 mm ² (16 AWG)
	Stranded	0.14 mm ² (26 AWG)	1.0 mm ² (18 AWG)
	rated @ 90 °C (194 °F) insulation max		
Wiring category ⁽¹⁾	2 – on signal ports 2 – on power ports 2 – on communication ports		
Wire type	Use copper conductors or shielded cables		
Terminal screw torque	For removable and fixed terminal blocks: 0.5...0.6 Nm (4.4...5.3 lb-in.) using a 0.6 x 3.5 mm flat-blade screwdriver. Note: Use a handheld screwdriver to hold down the screws at the side. For RS232/RS485 serial port: 0.22...0.25 Nm (1.95...2.21 lb-in.) using 0.4 x 2.5 x 80 mm 2-component grip with non-slip grip screwdriver.		
Input circuit type	24V DC sink/source (standard) – for 2080-LC20-20QWB(R), 2080-LC20-20QBB(R) 120V AC – for 2080-LC20-20AWB(R) for inputs 4...11 only		
Output circuit type	Relay	24V DC source (standard and high-speed)	Relay
Power input	24V DC		
Power consumption	5.62 W (without plug-ins, max) ... 8.5 W (with plug-ins, max)		

General Specifications

Attribute	2080-LC20-20AWB(R)	2080-LC20-20QBB(R)	2080-LC20-20QWB(R)
Power dissipation	6 W		
Power supply voltage range	20.4...26.4 V DC, Class 2		
Auxiliary power supply output for thermistor	10V		
I/O rating	Input: 120V AC 16 mA Output: 2 A, 240 V AC 2A, 24V DC	Input: 24V DC, 8.8 mA Output: 24V DC, 1 A per point (Surrounding air temperature 30°C) 24 V DC, 0.3 A per point (Surrounding air temperature 65 °C)	Input: 24V DC, 8.8 mA Output: 2 A, 240 V AC, 2A, 24V DC
Isolation voltage	250V (continuous), Reinforced Insulation Type, Output to Aux and Network, Inputs to Outputs. 150V (continuous), Reinforced Insulation Type, Input to Aux and Network. Type tested for 60 s @ 3250 V DC Output to Aux and Network, Inputs to Outputs. Type tested for 60 s @ 1950 V DC Input to Aux and Network.	50V (continuous), Reinforced Insulation Type, I/O to Aux and Network, Inputs to Outputs. Type tested for 60 s @ 720 V DC, I/O to Aux and Network, Inputs to Outputs.	250V (continuous), Reinforced Insulation Type, Output to Aux and Network, Inputs to Outputs. 50V (continuous), Reinforced Insulation Type, Input to Aux and Network. Type tested for 60 s @ 720 V DC, Inputs to Aux and Network, 3250 V DC Outputs to Aux and Network, Inputs to Outputs.
Pilot duty rating	C300, R150	—	C300, R150
Insulation stripping length	7 mm for the removable and fixed terminal blocks 5 mm for the RS232/RS485 serial port		
Enclosure type rating	Meets IP20		
North American temp code	T4		

(1) Use this Conductor Category information for planning conductor routing. Refer to Industrial Automation Wiring and Grounding Guidelines, publication [1770-4.1](#).

Analog Input Specifications for I-00...I-03

Attribute	Value
Number of inputs	4
Type	Voltage (single-ended)
Data range	0...4095
Input voltage range	0...10V DC
Maximum input	26.4V DC
Input impedance	14.14 k Ω
Resolution	12-bit, 2.5 mV/count
Smoothing	None
Input time constant, typical	0.44 ms
Isolation	None
Accuracy (25...55 °C)	5% of full-scale (2% with calibration)

AC Input Specifications for I-04...I-11 for 2080-LC20-20AWB(R)

Attribute	Value
Number of inputs	8
On-state voltage	120V AC, nom 79V AC, min 125V AC, max
On-state current	5 mA, min 16 mA, max
Input frequency	50/60 Hz, nom 47 Hz, min 63 Hz, max
Off-state voltage, max	20V AC
Off-state current, max	2.5 mA
Inrush current, max	250 mA @ 125V AC
Inrush decay time constant, max	22 ms

DC Input Filter Settings for I-04...I-11 for 2080-LC20-20QWB(R), 2080-LC20-20QBB(R)

Nominal Filter Setting (ms) Inputs 4 and higher	Minimum ON Delay (ms)	Maximum ON Delay (ms)	Minimum OFF Delay (ms)	Maximum OFF Delay (ms)
0	0	0.1	0	0.1
8	5	8	5	8
16	10	16	10	16
32	20	32	20	32

DC Input Specifications

Attribute	Non-isolated, shared with analog inputs (Inputs 00...03)	Isolated inputs (Inputs 04...11) – for 2080-LC20-20QWB(R), 2080-LC20-20QBB(R) only
Voltage category	24V DC Sink	24V DC Sink/Source
On-state voltage, nom	12/24V DC	24V DC
On-state voltage range	9.8...26.4V DC	24V DC, nom 10...26.4V DC @ 65 °C (149 °F) 10...30V DC @ 30 °C (86 °F)
Off-state voltage, max	5V DC	
Off-state current, max	0.5 mA	1.5 mA
On-state current, min	0.75 mA @ 10.8V DC 1.0 mA @ 15V DC	1.8 mA @ 10.8V DC 2.7 mA @ 15V DC
On-state current, nom	2.1 mA @ 24V DC	8.5 mA @ 24V DC
On-state current, max	2.6 mA @ 26.4V	12.0 mA @ 30V DC
Nominal impedance	14.1 k Ω (non-isolated)	3.74 k Ω (isolated)
IEC input compatibility	Type 1	Type 3

Relay Output Specifications for 0-00...06 for 2080-LC20-20QWB(R), 2080-LC20-20AWB(R)

Attribute	Value
Voltage, min	5 V, AC 5 V, DC
Voltage, max	250 V, AC
Maximum switching time	10 ms, turn on 10 ms, turn off
Life	10,000,000 cycles (mechanical) 100,000 cycles (Electrical with UL test load)

Relay Contact Ratings

Maximum Volts	Amperes		Amperes Continuous	Volt-Amperes	
	Make	Break		Make	Break
120 V AC	15 A	1.5 A	2 A	1800	180
240 V AC	7.5 A	0.75 A			
24 V DC	1.0 A		1 A	28	
125 V DC	0.22 A				

Analog Output Specifications

Attribute	Value
Output count range	0...4008
Output type	Voltage
Output Voltage Range	0...10V
Voltage Output Maximum Load (Resistive)	>1000 Ohms
Accuracy	2% of full scale for voltage
Resolution	12-bit, 2.5 mV/count
Output update rate (with no output capacitance), max	20 ms
Channel-to-bus isolation	No isolation
Channel-to-channel isolation	No isolation

DC Output Specifications for 2080-LC20-20QBB(R)

Attribute	Standard Outputs (Outputs 0-00...0-05)	High Speed Output ⁽¹⁾ (Output 0-06)
User supply voltage	10V DC, min 26.4V DC, max	10V DC, min 26.4V DC, max
Load current, min	10 mA	
On state voltage drop	1V @ max load current 2.5V @ max surge current	1.5V @ max load current

DC Output Specifications for 2080-LC20-20QBB(R)

Attribute	Standard Outputs (Outputs 0-00...0-05)	High Speed Output ⁽¹⁾ (Output 0-06)
Current ratings per point	0.3 A @ 65 °C, max 1.0 A @ 30 °C, max 1.0 mA, max leakage	100 mA (high speed operation) 1.0 A @ 30 °C 0.3 A @ 65 °C (standard operation) 1.0 mA, max leakage
Surge current per point peak current max surge duration max rate of repetition @ 30 °C max rate of repetition @ 65 °C	4.0 A 10 ms once each second once every two seconds	
Controller current, max total	3 A	—
Turn-on time, max	0.1 ms	0.2 µs
Turn-off time, max	1.0 ms	2.5 µs
Response time, max	10 ms	
Frequency rate	NA	2%

(1) High speed output operation is greater than 5 KHz.

PWM Output Duty Cycle Error

Turn On/Off time for the Micro820 controllers for the PWM output port is 0.2 µs and 2.5 µs max, respectively. Duty cycle error is:

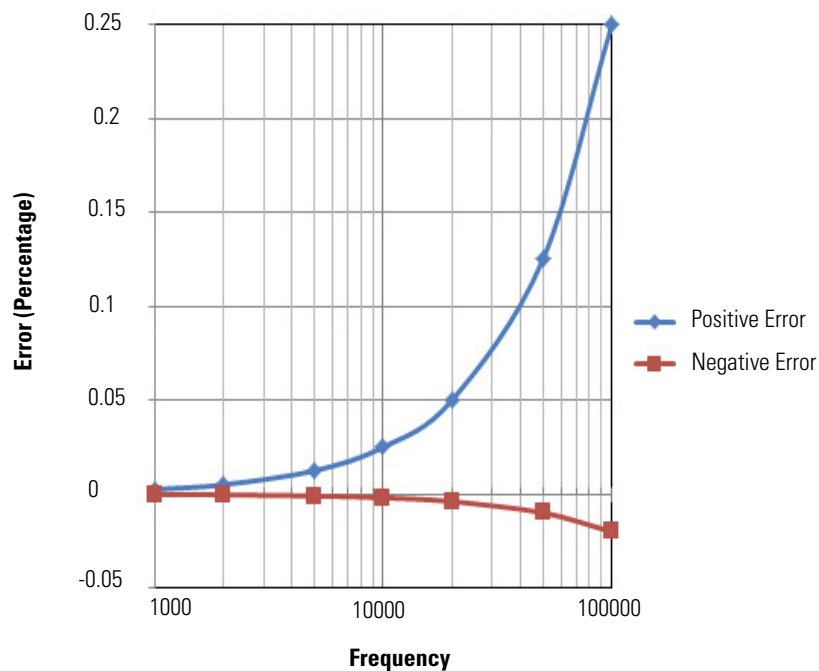
Positive error = $2.5 \mu s * F$

Negative error = $-0.2 \mu s * F$

The plot below shows duty cycle error vs. frequency.

To get the duty cycle error at a certain frequency, for example, the user sets frequency to 20 kHz, and sets duty cycle to 30% in Connected Components Workbench, then actual duty cycle is

30% $\begin{matrix} +5\% \\ -0.4\% \end{matrix}$.



PWM Typical Readings

PWM Typical Readings

		Expected Duty Cycle		Typical Duty Cycle (1.27 K Ω load)
Frequency (kHz)	%Duty Cycle	Minimum	Maximum	%Duty Cycle
5	5%	4.90%	6.25%	5.48
5	10%	9.90%	11.25%	10.5
5	20%	19.90%	21.25%	20.5
5	40%	39.90%	41.25%	40.5
5	55%	54.90%	56.25%	55.5
5	65%	64.90%	66.25%	65.5
5	75%	74.90%	76.25%	75.5
5	95%	94.90%	96.25%	95.5
10	5%	4.80%	7.50%	5.9
10	10%	9.80%	12.50%	11.0
10	20%	19.80%	22.50%	21.0
10	40%	39.80%	42.50%	40.9
10	55%	54.80%	57.50%	55.9
10	65%	64.80%	67.50%	65.9
10	85%	84.80%	87.50%	85.9
10	95%	94.80%	97.50%	95.9
25	5%	4.50%	11.25%	7.25

PWM Typical Readings

		Expected Duty Cycle		Typical Duty Cycle (1.27 KΩ load)
Frequency (kHz)	%Duty Cycle	Minimum	Maximum	%Duty Cycle
25	10%	9.50%	16.25%	12.3
25	20%	19.50%	26.25%	22.4
25	40%	39.50%	46.25%	42.3
25	55%	54.50%	61.25%	57.3
25	65%	64.50%	71.25%	67.3
25	85%	84.50%	91.25%	87.3
25	95%	94.50%	100.00%	97.0
50	5%	4.00%	17.50%	9.7
50	10%	9.00%	22.50%	14.8
50	20%	19.00%	32.50%	24.7
50	40%	39.00%	52.50%	44.7
50	55%	54.00%	67.50%	59.6
50	65%	64.00%	77.50%	69.6
50	85%	84.00%	97.50%	89.5
50	95%	94.00%	100.00%	98.1
100	5%	3.00%	30.00%	14.7
100	10%	8.00%	35.00%	19.5
100	20%	18.00%	45.00%	29.6
100	40%	38.00%	65.00%	49.3
100	55%	53.00%	80.00%	64.0
100	65%	63.00%	90.00%	73.8
100	85%	83.00%	100.00%	92.4
100	95%	93.00%	100.00%	98.0

Auxiliary Power Supply for Thermistor Applications

Attribute	Value
Output voltage	9.5V, min 10.04V, typical 10.5V, max
Output current	10 mA, typical 20 mA, max

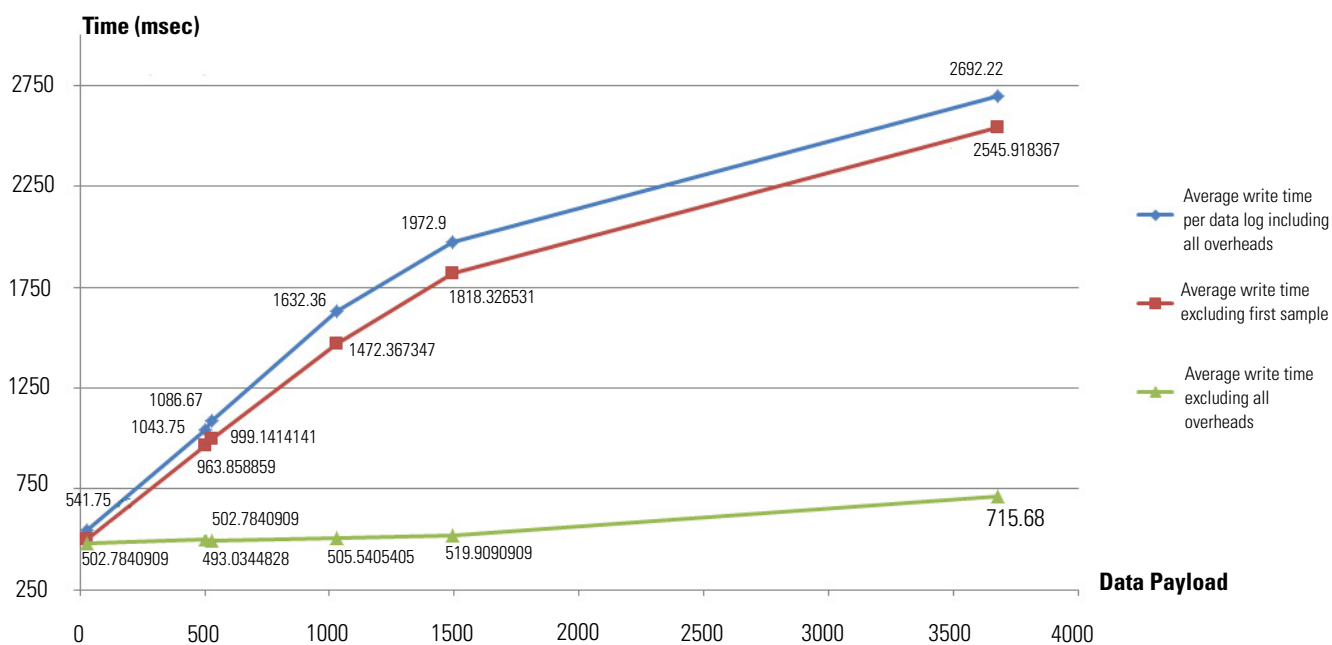
Embedded RTC

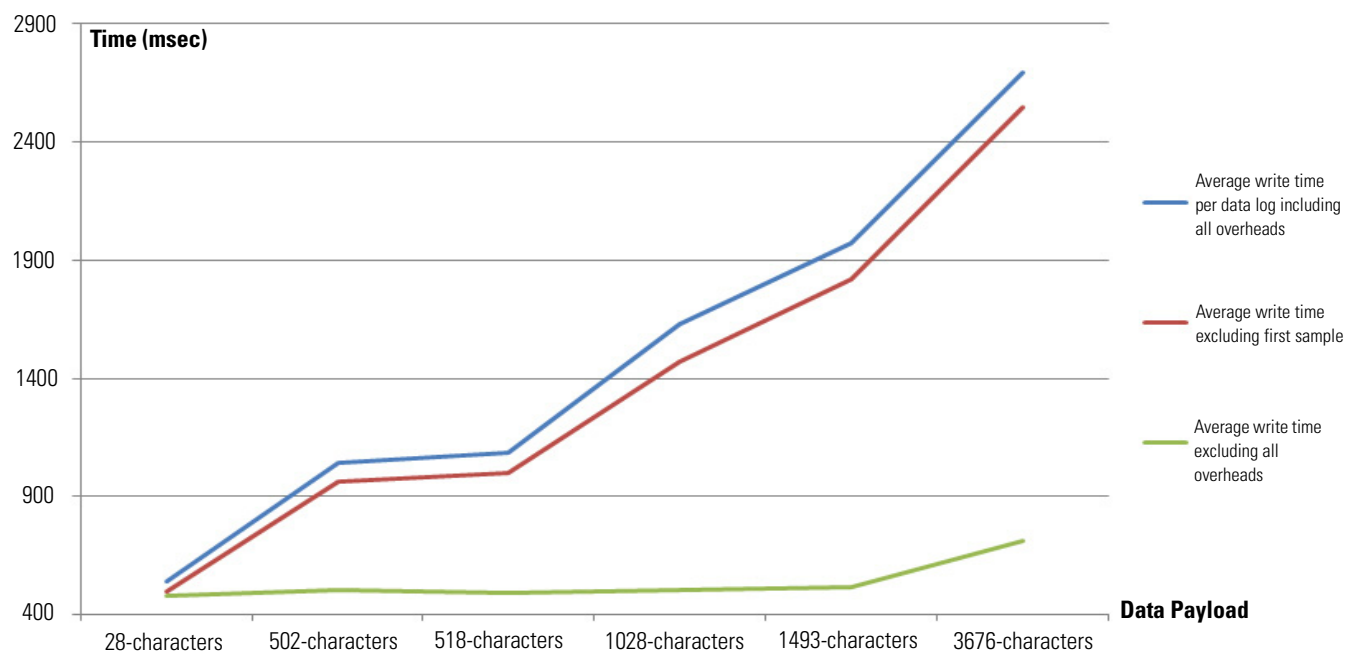
Attribute	Value
Resolution	1 sec
Accuracy, typical	± 60 sec/month @ 25 °C
Supercap life, typical	5 years @ 40 °C, 14.5 years @ 25 °C

Data Log Performance

Data Log – Data Payload vs. Performance Time

Parameter	Number of Characters					
	28	502	518	1028	1493	3676
Average write time per data log file including all overheads	541.77 ms	1043.75 ms	1086.67 ms	1632.36 ms	1972.9 ms	2696.22 ms
Average write time excluding first sample	500.40 ms	963.86 ms	999.14 ms	1472.36 ms	1818.33 ms	2545.92 ms
Average write time excluding all overheads	479.10 ms	502.78 ms	493.03 ms	505.54 ms	519.91 ms	715.68 ms





Environmental Specifications

Environmental Specifications

Attribute	Value
Temperature, operating	IEC 60068-2-1 (Test Ad, Operating Cold), IEC 60068-2-2 (Test Bd, Operating Dry Heat), IEC 60068-2-14 (Test Nb, Operating Thermal Shock): -20...65 °C (-4...149 °F)
Temperature, surrounding air, max	65 °C (149 °F)
Temperature, nonoperating	IEC 60068-2-1 (Test Ab, Unpackaged Nonoperating Cold), IEC 60068-2-2 (Test Bb, Unpackaged Nonoperating Dry Heat), IEC 60068-2-14 (Test Na, Unpackaged Nonoperating Thermal Shock): -40...85 °C (-40...185 °F)
Relative humidity	IEC 60068-2-30 (Test Db, Unpackaged Damp Heat): 5...95% non-condensing
Vibration	IEC 60068-2-6 (Test Fc, Operating): 2 g @ 10...500 Hz
Shock, operating	IEC 60068-2-27 (Test Ea, Unpackaged Shock): 25 g
Shock, non-operating	IEC 60068-2-27 (Test Ea, Unpackaged Shock): DIN mount: 25 g PANEL mount: 45 g
Emissions	CISPR 11 Group 1, Class A
ESD immunity	IEC 61000-4-2: 6 kV contact discharges 8 kV air discharges

Environmental Specifications

Attribute	Value
Radiated RF immunity	IEC 61000-4-3: 10V/m with 1 kHz sine-wave 80% AM from 80...2000 MHz 10V/m with 200 Hz 50% Pulse 100% AM @ 900 MHz 10V/m with 200 Hz 50% Pulse 100% AM @ 1890 MHz 10V/m with 1 kHz sine-wave 80% AM from 2000...2700 MHz
EFT/B immunity	IEC 61000-4-4: ±2 kV @ 5 kHz on power ports ±2 kV @ 5 kHz on signal ports ±1 kV @ 5 kHz on communication ports
Surge transient immunity	IEC 61000-4-5: ±1 kV line-line(DM) and ±2 kV line-earth(CM) on power ports ±1 kV line-line(DM) and ±2 kV line-earth(CM) on signal ports ±1 kV line-earth(CM) on communication ports
Conducted RF immunity	IEC 61000-4-6: 10V rms with 1 kHz sine-wave 80% AM from 150 kHz...80 MHz

Certifications**Certifications**

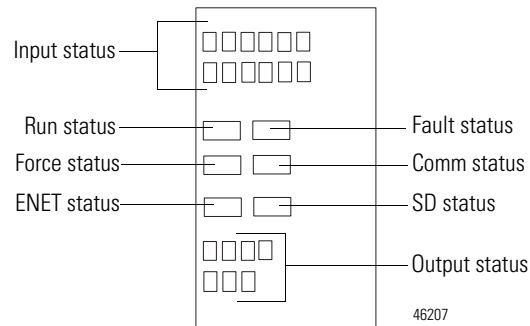
Certification (when product is marked) ⁽¹⁾	Value
c-UL-us	UL Listed Industrial Control Equipment, certified for US and Canada. See UL File E322657. UL Listed for Class I, Division 2 Group A,B,C,D Hazardous Locations, certified for U.S. and Canada. See UL File E334470.
CE	European Union 2004/108/EC EMC Directive, compliant with: EN 61326-1; Meas./Control/Lab., Industrial Requirements EN 61000-6-2; Industrial Immunity EN 61000-6-4; Industrial Emissions EN 61131-2; Programmable Controllers (Clause 8, Zone A & B) European Union 2006/95/EC LVD, compliant with: EN 61131-2; Programmable Controllers (Clause 11)
C-Tick	Australian Radiocommunications Act, compliant with: AS/NZS CISPR 11; Industrial Emissions
EtherNet/IP	ODVA conformance tested to EtherNet/IP specifications
KC	Korean Registration of Broadcasting and Communications Equipment, compliant with: Article 58-2 of Radio Waves Act, Clause 3

(1) See the Product Certification link at <http://www.rockwellautomation.com/products/certification> for Declaration of Conformity, Certificates, and other certification details.

Troubleshooting

Status Indicators on the Controller

Status indication on the Micro820 controller is as follows.



Status Indicator Description

	Description	State	Indicates
1	Input status	Off	Input is low.
		On	Input is energized (terminal status).
2	Fault status	Off	No fault detected.
		Red	Non-recoverable fault that requires a power cycle.
		Flashing red	Recoverable fault.
3	Run status	Green	Executing the user program in run mode.
		Flashing Green (1 Hz)	The controller is in program mode.
4	Serial communications status	Off	No traffic for RS232/RS485.
		Green	Traffic through RS232/RS485. The indicator only blinks when transmitting data. It does not blink when receiving data.
5	Force status	Off	No force conditions are active.
		Amber	Force conditions are active.

Status Indicator Description

	Description	State	Indicates
5	SD status	Off Uninitialized State	<ul style="list-style-type: none"> microSD card is not inserted. microSD card is inserted but medium is bad. microSD card is inserted but file system is bad.
		Off Error State	<ul style="list-style-type: none"> microSD card read/write failure. Failure to read ConfigMeFirst.txt in the root directory. Errors are detected in ConfigMeFirst.txt. See ConfigMeFirst.txt Errors on page 98 for list of errors.
		On Idle State	<ul style="list-style-type: none"> microSD card is initialized completely without read/write on SD card. microSD card read/write is complete.
		Blinking Operating State	microSD card is being read/written.
6	ENET status	Steady Off	Not powered, no connection. The device is powered off, or is powered on but no Ethernet link established.
		Flashing Green	No IP address. The device is powered on with Ethernet link established but no IP address is assigned yet. Duplicate IP. The device has detected that its IP address as being used by another device in the network. This status is applicable only if the device's duplicate IP address detection (ACD) feature is enabled.
		Steady Green	Operational. Ethernet link is active and the device has valid IP address.
7	Output status	Off	Output is not energized.
		On	Output is energized (logic status).

Normal Operation

The RUN indicator is on or flashing. If a force condition is active, the FORCE indicator turns on and remains on until all forces are removed.

Error Codes

This section lists possible error codes for your controller, as well as recommended actions for recovery. Information about the fault is stored in a fault log, which can be accessed from the Diagnostics page in Connected Components Workbench software. The fault log contains brief information about the last fault, and detailed information about the last 10 non-recoverable faults that occurred.

If an error persists after performing the recommended action, contact your local Rockwell Automation technical support representative. For contact information, go to

<http://support.rockwellautomation.com/MySupport.asp>

Fault Types

There are two basic types of faults that can occur:

- Recoverable – A recoverable fault can be cleared without having to power cycle the controller. The fault LED flashes red when a recoverable fault occurs.
- Non-recoverable – A non-recoverable fault requires the controller to be power cycled before clearing the fault. Non-recoverable controller faults may become recoverable after the controller has been power cycled, or an automatic reset is performed by the controller. If the controller performs an automatic reset and the fault becomes recoverable, the fault will not be logged. After the controller has been power cycled or reset, check the fault log in the Diagnostic page of the Connected Components Workbench software, then clear the fault. The fault LED is solid red when a non-recoverable fault occurs.

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xF000	Recoverable	The controller was unexpectedly reset due to a noisy environment or an internal hardware failure. If the system variable _SYSVA_USER_DATA_LOST is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Check wiring to eliminate any noise, refer to Wiring Requirements and Recommendation on page 39.
0xF001	Recoverable	The controller program has been cleared. This happened because: <ul style="list-style-type: none"> a power-down occurred during program download or data transfer from the memory module the cable was removed from the controller during program download. the RAM integrity test failed. 	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Transfer the program using the memory module restore utility.
0xF002	Non-recoverable	The controller hardware watchdog was activated. The controller hardware watchdog timeout happens if program scan is more than three seconds. If the system variable _SYSVA_USER_DATA_LOST is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.	See Corrective Actions for Non-recoverable Faults on page 143 .
0xF003	Recoverable	One of the following occurred: <ul style="list-style-type: none"> The memory module hardware faulted. The memory module connection faulted. The memory module was incompatible with the Micro800 controller's firmware revision. 	Perform one of the following: <ul style="list-style-type: none"> Remove the memory module and plug it in again. Obtain a new memory module. See Corrective Actions for Recoverable Faults on page 143. Upgrade the Micro800 controller's firmware revision to be compatible with the memory module. For more information on firmware revision compatibility, go to http://www.rockwellautomation.com/support/firmware.html
0xF004	Recoverable	A failure occurred during the memory module data transfer.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Attempt the data transfer again. If the error persists, replace the memory module. For Embedded RTC failure, restart the controller.
0xF005	Recoverable	The user program failed an integrity check while the Micro800 controller was in Run mode.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Refer to the Wire Your Controller on page 39.
0xF006	Recoverable	The user program is incompatible with the Micro800 controller's firmware revision.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Contact your local Rockwell Automation technical support representative.

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xF010	Recoverable	The user program contains a function/function block that is not supported by the Micro800 controller.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Contact your local Rockwell Automation technical support representative.
0xF014	Recoverable	A memory module memory error occurred.	<ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Reprogram the memory module. If the error persists, replace the memory module.
0xF015	Non-recoverable	An unexpected software error occurred.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Non-recoverable Faults on page 143. Refer to Wiring Requirements and Recommendation on page 39.
0xF016	Non-recoverable	An unexpected hardware error occurred.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Non-recoverable Faults on page 143. Refer to Wiring Requirements and Recommendation on page 39.
0xF017	Non-recoverable	An unexpected software error occurred due to unexpected hardware interrupt. If the system variable _SYSVA_USER_DATA_LOST is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Non-recoverable Faults on page 143. Refer to Wiring Requirements and Recommendation on page 39.
0xF018	Non-recoverable	An unexpected software error occurred due to SPI communication failure. If the system variable _SYSVA_USER_DATA_LOST is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Non-recoverable Faults on page 143. Refer to Wiring Requirements and Recommendation on page 39.
0xF019	Non-recoverable	An unexpected software error occurred due to memory or other controller resource issue.	See Corrective Actions for Non-recoverable Faults on page 143 .
0xF01A	Recoverable	The controller was unexpectedly reset during Run Mode Change (RMC) due to a noisy environment or an internal hardware failure. If the system variable _SYSVA_USER_DATA_LOST is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.	See Corrective Actions for Recoverable Faults on page 143 .
0xF020	Recoverable	The base hardware faulted or is incompatible with the Micro800 controller's firmware revision.	See Corrective Actions for Recoverable Faults on page 143 .
0xF021	Recoverable	The I/O configuration in the user program is invalid or does not exist in the Micro800 controller.	See Corrective Actions for Recoverable Faults on page 143 .
0xF022	Recoverable	The user program in the memory module is incompatible with the Micro800 controller's firmware revision.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Replace the memory module.

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xF023	Non-recoverable	The controller program has been cleared. This happened because: <ul style="list-style-type: none"> a power down occurred during program download or transfer from the memory module. the Flash Integrity Test failed (Micro810 only). 	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Non-recoverable Faults on page 143. Download or transfer the program.
0xF030 0xF031 0xF032 0xF033	Recoverable	Power down information in persistent memory may not be written properly due to a noisy environment or an internal hardware failure. If the system variable _SYSVA_USER_DATA_LOST is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.	See Corrective Actions for Recoverable Faults on page 143 .
0xF050	Recoverable	The embedded I/O configuration in the user program is invalid.	See Corrective Actions for Recoverable Faults on page 143 .
0xF100	Recoverable	There is general configuration error detected in the motion configuration downloaded from the Connected Components Workbench software, such as number of axis, or motion execution interval being configured out of range.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the axes configuration in the user program.
0xF110	Recoverable	There is motion resource missing, such as Motion_DIAG variable not defined.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the axes configuration in the user program.
0xF12z ⁽¹⁾	Recoverable	Motion configuration for axis z cannot be supported by this controller model, or the axis configuration has some resource conflict with some other motion axis, which has been configured earlier.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Remove all axes and re-configure motion with the guidance from the User Manual.
0xF15z ⁽¹⁾	Recoverable	There is a motion engine logic error (firmware logic issue or memory crash) for one axis detected during motion engine cyclic operation. One possible reason can be motion engine data/memory crash.	See Corrective Actions for Recoverable Faults on page 143 .
0xF210	Recoverable	The expansion I/O terminator is missing.	Perform the following: <ol style="list-style-type: none"> Power off the controller. Attach the expansion I/O terminator on the last expansion I/O module on the system. Power on the controller. See Corrective Actions for Recoverable Faults on page 143.
0xF230	Recoverable	The maximum number of expansion I/O modules has been exceeded.	Perform the following: <ol style="list-style-type: none"> Power off the controller. Check that the number of expansion I/O modules is not more than four. Power on the controller. See Corrective Actions for Recoverable Faults on page 143.
0xF250	Recoverable	There is a non-recoverable error and the expansion I/O module(s) could not be detected.	See Corrective Actions for Recoverable Faults on page 143 .
0xF26z ⁽²⁾	Recoverable	An expansion I/O master fault is detected on the system.	See Corrective Actions for Recoverable Faults on page 143 .

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xF27z ⁽²⁾	Recoverable	A non-recoverable communication fault has occurred on the expansion I/O module.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Replace the slot number z module.
0xF28z ⁽²⁾	Recoverable	Expansion I/O baudrate error.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Replace the slot number z module.
0xF29z ⁽²⁾	Recoverable	A module fault is detected on your expansion I/O module.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Replace the slot number z module.
0xF2Az ⁽²⁾	Recoverable	Expansion I/O power failure	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Replace the slot number z module.
0xF2Bz ⁽²⁾	Recoverable	Expansion I/O configuration fault.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the expansion IO module configuration in the user program to match that of the actual hardware configuration. Check the expansion I/O module operation and condition. Replace the expansion I/O module.
0xF300	Recoverable	The memory module is present but memory module is empty and restore operation is requested.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Check to make sure there is a valid project in the memory module. Download a user program and use the backup function to the memory module.
0xF301	Recoverable	The memory module's project is not compatible with the controller.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Check to make sure there is a user program with a controller that has the correct controller catalog configured. Download a user program and use the backup function to the memory module.
0xF302	Recoverable	The password is mismatched between memory module and controller. Only applies to Micro820 controller when Remote LCD performs the restore operation. This fault does not apply to Micro800 controller firmware revision 10 and later.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Check to make sure that the user program in the memory module has the correct password. Download a user program with a password and use the backup function to the memory module. Use Connected Components Workbench to enter the correct password into the controller and perform the restore operation again.
0xF303	Recoverable	The memory module is not present and restore operation is requested.	Check to make sure the memory module is present.

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xF0Az ⁽³⁾	Recoverable	The plug-in I/O module experienced an error during operation.	Perform the following: <ul style="list-style-type: none"> Check the condition and operation of the plug-in I/O module. See Corrective Actions for Recoverable Faults on page 143.
0xF0Bz ⁽³⁾	Recoverable	The plug-in I/O module configuration does not match the actual I/O configuration detected.	Perform one of the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the plug-in I/O module configuration in the user program to match that of the actual hardware configuration. Check the condition and operation of the plug-in I/O module. Replace the plug-in I/O module.
0xF0Dz ⁽³⁾	Recoverable	When power was applied to the plug-in I/O module or the plug-in I/O module was removed, a hardware error occurred.	Perform the following: <ol style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the plug-in I/O module configuration in the user program. Build and download the program using Connected Components Workbench. Put the Micro800 controller into Run mode.
0xF0Ez ⁽³⁾	Recoverable	The plug-in I/O module configuration does not match the actual I/O configuration detected.	Perform the following: <ol style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the plug-in I/O module configuration in the user program. Build and download the program using Connected Components Workbench. Put the Micro800 controller into Run mode.
0xF830	Recoverable	An error occurred in the EII configuration.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Review and change the EII configuration in the Micro800 controller properties.
0xF840	Recoverable	An error occurred in the HSC configuration.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Review and change the EII configuration in the Micro800 controller properties.
0xF850	Recoverable	An error occurred in the STI configuration.	Perform the following: <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Review and change the EII configuration in the Micro800 controller properties.
0xF860	Recoverable	A data overflow occurred. A data overflow error is generated when the ladder, structured text or function block diagram execution encounters a divide-by-zero.	Perform the following: <ol style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the program to ensure that there is no data overflow. Build and download the program using Connected Components Workbench. Put the Micro800 controller into Run mode.
0xF870	Recoverable	An index address was out of data space.	Perform the following: <ol style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Correct the program to ensure that there is no index used to access an array element beyond the array boundaries. Build and download the program using Connected Components Workbench. Put the Micro800 controller into Run mode.

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xF0878	Recoverable	An index used to access a bit is beyond the boundaries of the data type it is used on.	Perform the following: <ol style="list-style-type: none"> 1. See Corrective Actions for Recoverable Faults on page 143. 2. Correct the program to ensure that there is no index used to access a bit beyond the boundaries of the data type. 3. Build and download the program using Connected Components Workbench. 4. Put the Micro800 controller into Run mode.
0xF880	Recoverable	A data conversion error occurred.	Perform the following: <ol style="list-style-type: none"> 1. See Corrective Actions for Recoverable Faults on page 143. 2. Correct the program to ensure that there is no data conversion error. 3. Build and download the program using Connected Components Workbench. 4. Put the Micro800 controller into Run mode.
0xF888	Recoverable	The call stack of the controller cannot support the sequence of calls to function blocks in the current project. Too many blocks are within another block.	Perform the following: <ul style="list-style-type: none"> • See Corrective Actions for Recoverable Faults on page 143. • Change the project to reduce the quantity of blocks being called within a block.
0xF898	Recoverable	An error occurred in the user interrupt configuration for the plug-in I/O module.	Perform the following: <ul style="list-style-type: none"> • See Corrective Actions for Recoverable Faults on page 143. • Correct the user interrupt configuration for plug-in I/O module in the user program to match that of the actual hardware configuration.
0xF8A0	Recoverable	The TOW parameters are invalid.	Perform the following: <ol style="list-style-type: none"> 1. See Corrective Actions for Recoverable Faults on page 143. 2. Correct the program to ensure that there are no invalid parameters. 3. Build and download the program using Connected Components Workbench. 4. Put the Micro800 controller into Run mode.
0xF8A1	Recoverable	The DOY parameters are invalid.	Perform the following: <ol style="list-style-type: none"> 1. See Corrective Actions for Recoverable Faults on page 143. 2. Correct the program to ensure that there are no invalid parameters. 3. Build and download the program using Connected Components Workbench. 4. Put the Micro800 controller into Run mode.

List of Error Codes for Micro800 controllers

Error Code	Fault Type	Description	Recommended Action
0xFFzz ⁽⁴⁾	Recoverable	A user-created fault from Connected Components Workbench has occurred.	See Corrective Actions for Recoverable Faults on page 143 .
0xD00F	Recoverable	A particular hardware type (for example, embedded I/O) was selected in the user program configuration, but did not match the actual hardware base.	See Corrective Actions for Recoverable Faults on page 143 .
0xD011	Recoverable	The program scan time exceeded the watchdog timeout value.	<p>Perform the following:</p> <ul style="list-style-type: none"> See Corrective Actions for Recoverable Faults on page 143. Determine if the program is caught in a loop and correct the problem. <p>Fault may occur if your Structured Text program contains a For loop with the upper limit set to the maximum value of the variable. For example, the variable is a USINT and the limit is set to 255, or the variable is a UINT and the limit is set to 65535.</p> <p>To correct the fault, perform the following:</p> <ol style="list-style-type: none"> Correct the program to ensure that the upper limit is not reached. One method is to use a data type with a larger maximum value. Build and download the program using Connected Components Workbench. Put the Micro800 controller into Run mode. <p>If your program is designed to have a scan time of longer than three seconds, in the user program, increase the watchdog timeout value that is set in the system variable _SYSVA_TCYWDG and then build and download the program using Connected Components Workbench.</p>

(1) z indicates the logic axis ID. (0...3)

(2) z indicates the slot number of the expansion I/O. If z=0, then the slot number cannot be identified.

(3) z is the slot number of the plug-in module. If z = 0, then the slot number cannot be identified.

(4) zz indicates the last byte of the program number. Only program numbers up to 0xFF can be displayed. For program numbers 01x00 to 0xFFFF, only the last byte is displayed.)

Corrective Action for Recoverable and Non-recoverable Faults

Corrective Actions for Recoverable Faults

Perform the following:

1. Optionally save the fault log from Connected Components Workbench software.
2. Clear the recoverable fault using Connected Components Workbench software.
3. If problem persists, contact technical support with the fault log.

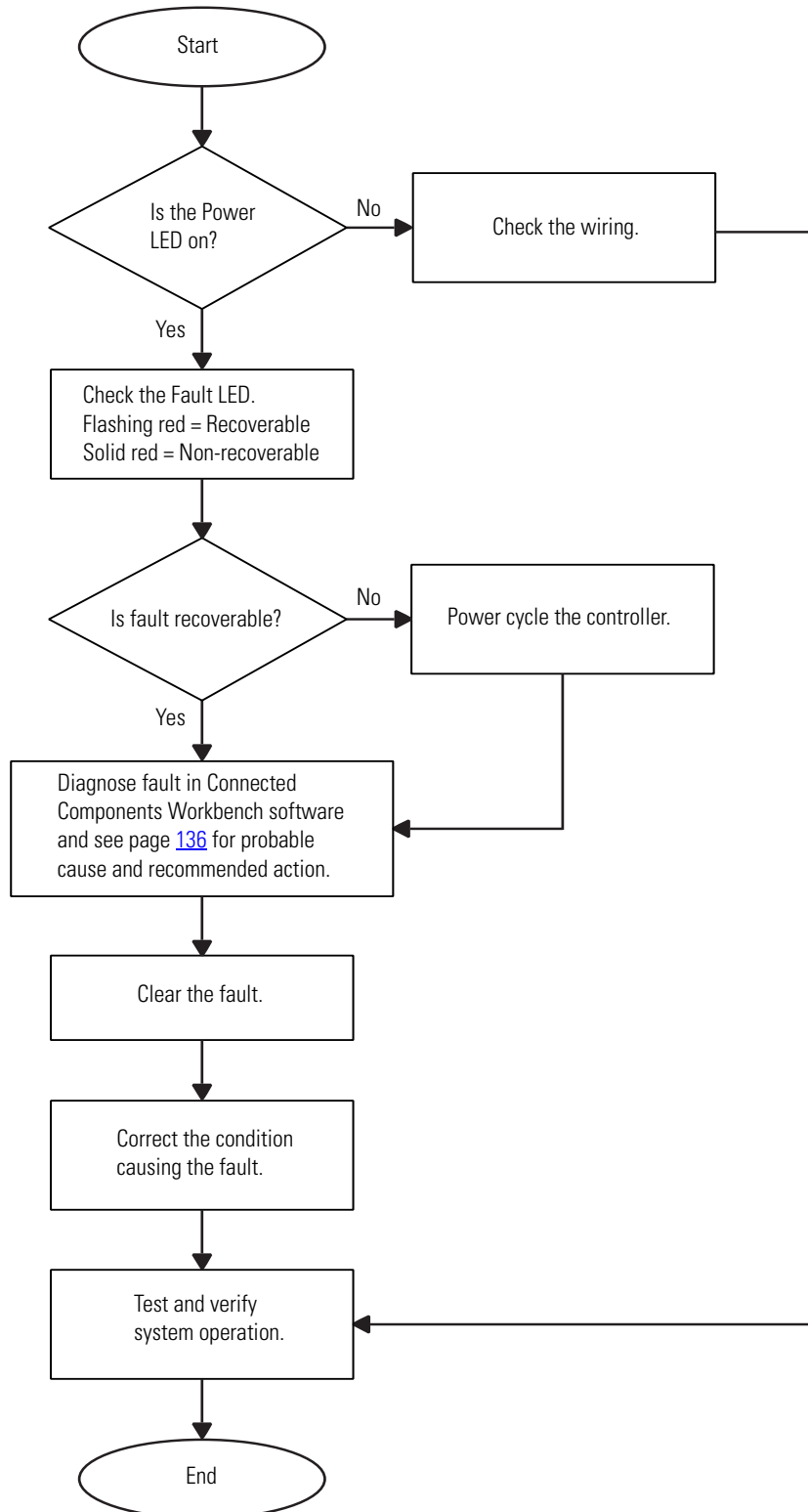
Corrective Actions for Non-recoverable Faults

Perform the following:

1. Power cycle your Micro800 controller.
2. Controller will go to recoverable fault. Optionally save the fault log from Connected Components Workbench software.
3. Clear the recoverable fault using Connected Components Workbench software.
4. If program is lost, build and download your program using Connected Components Workbench software.
5. If problem persists, contact technical support with the fault log.

Controller Error Recovery Model

Use the following error recovery model to help you diagnose software and hardware problems in the micro controller. The model provides common questions you might ask to help troubleshoot your system. Refer to the recommended pages within the model for further help.



Calling Rockwell Automation for Assistance

If you need to contact Rockwell Automation or local distributor for assistance, it is helpful to obtain the following (prior to calling):

- controller type, series letter, revision letter, and firmware (FRN) number of the controller
- controller indicator status

Notes:

Quickstarts

This chapter covers some common tasks and quickstart instructions that are aimed to make you familiar with the in Connected Component Workbench. The following quickstarts are included:

Topic	Page
Flash Upgrade Your Micro800 Firmware	147
Establish Communications between RSLinx and a Micro820 Controller through USB Port on 2080-REMLCD	154
Configure Controller Password	158
Forcing I/Os	162
Using Run Mode Change	164

Flash Upgrade Your Micro800 Firmware

This quick start will show you how to flash update the firmware for a Micro800 controller using Connected Components Workbench software release 10 or later.

From Connected Components Workbench software release 10 onwards, there are two options you can select when flash updating the firmware:

- Upgrade or Downgrade – This option retains the controller's existing configuration, Ethernet settings, and password.
- Reset – This option clears the controller's existing configuration, Ethernet settings, and password.

The procedure to flash update the controller is similar for both options.



ATTENTION: Retention of the controller's existing configuration, Ethernet settings, and password is only available when flash updating from firmware revision 10 to the same or later revision. If updating from firmware revision 10 to 9 or earlier, or updating to firmware revision 10 from an earlier revision, the controller's existing configuration, Ethernet settings, and password will be cleared.

IMPORTANT

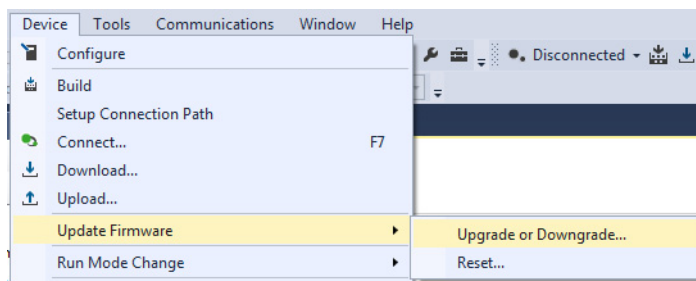
If you have forgotten the password for the controller, use the Reset option to clear the password.

On Micro820 controllers, users can use flash update their controllers through the Ethernet port, in addition to the USB of the 2080-REMLCD.

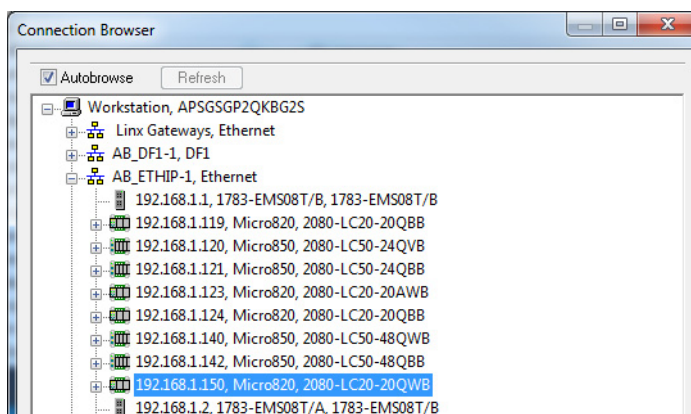
IMPORTANT To successfully flash update your controller over USB, connect only one controller to your computer, and do not perform the flash update in a virtual machine such as VMware.

To begin, launch the Connected Components Workbench software:

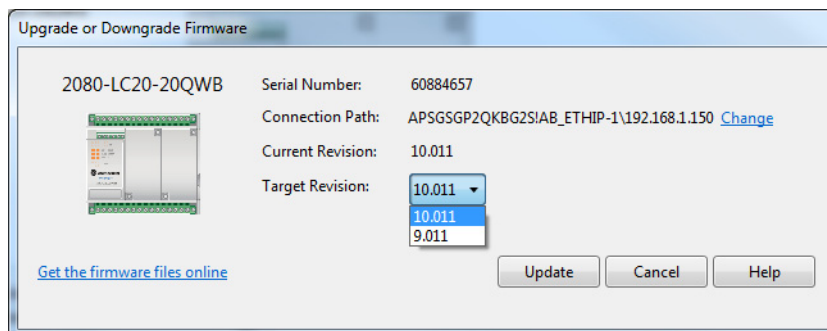
1. In the menu, select Device -> Update Firmware -> Upgrade or Downgrade...
Alternatively, in the Project Organizer, right-click the controller and select Update Firmware -> Upgrade or Downgrade...



2. If your project does not have a connection path to the controller, the Connection Browser dialog appears. Select your controller, then click OK.



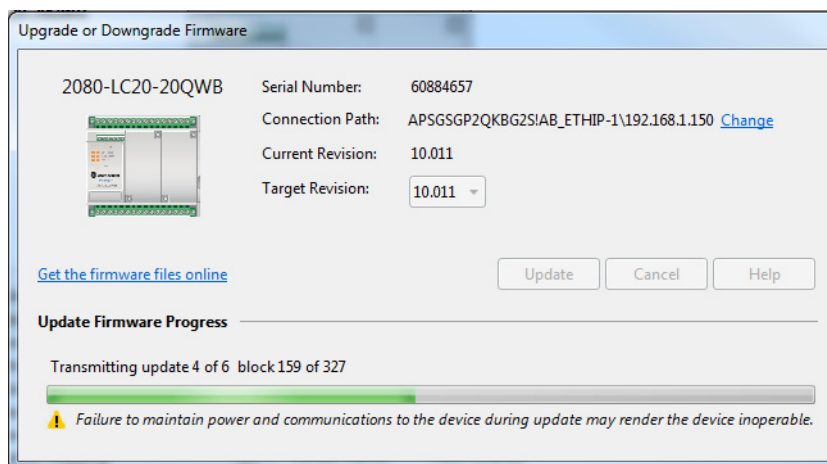
3. In the Upgrade or Downgrade Firmware dialog box, select the desired Target Revision to flash update the controller.



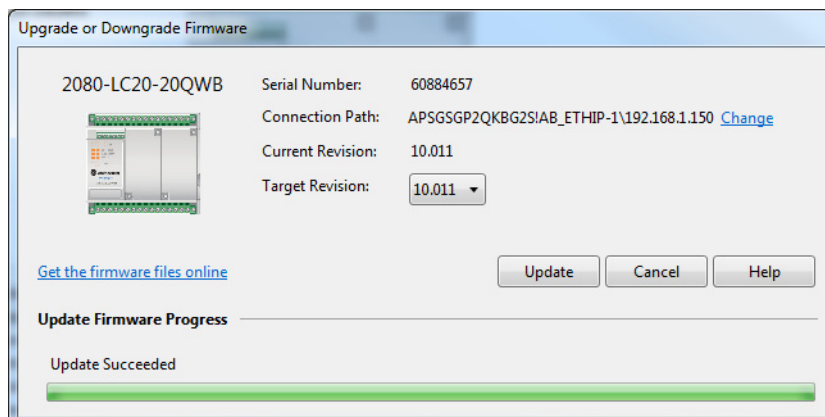
If the desired firmware revision is not shown in the drop-down list, you can download that firmware revision by clicking the “Get the firmware files online” link.

You can also change the Connection Path by clicking the “Change” link.

4. When you have confirmed the settings, click Update to begin flash updating the controller.
The update progress is shown in the dialog box.



5. After the update is completed, the status is shown in the dialog box.



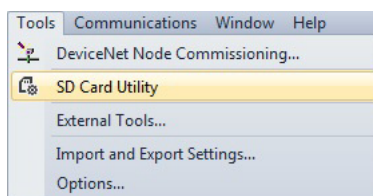
IMPORTANT After control flashing the controller, some microSD cards may not be detected. Remove and insert the microSD card, or power cycle the controller if this issue is encountered.

Flash Upgrade From microSD Card

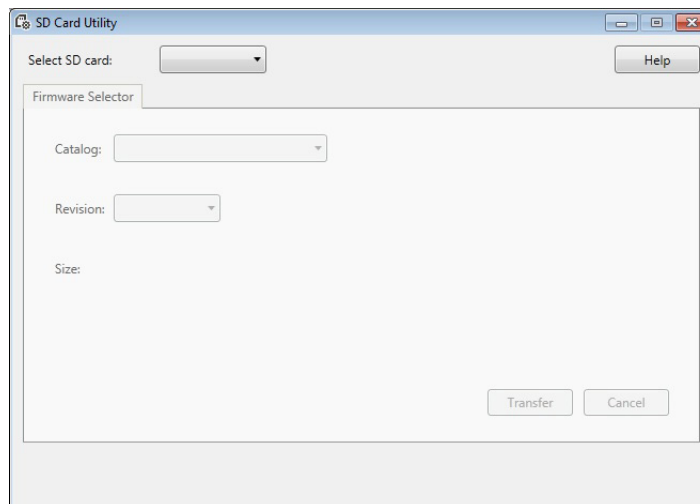
With Connected Components Workbench release 8.0 onwards, you can flash upgrade your Micro820 controller from the microSD card in addition to using ControlFLASH. This is two-step process – first you have to transfer the firmware to the microSD card using the SD Card Utility, then you need to edit the ConfigMeFirst.txt file to initiate the flash upgrade process. See the following instructions for performing the flash upgrade from the microSD card.

Step 1 – Transfer the Firmware to the microSD Card

1. Launch Connected Components Workbench.
2. Click Tools -> SD Card Utility.

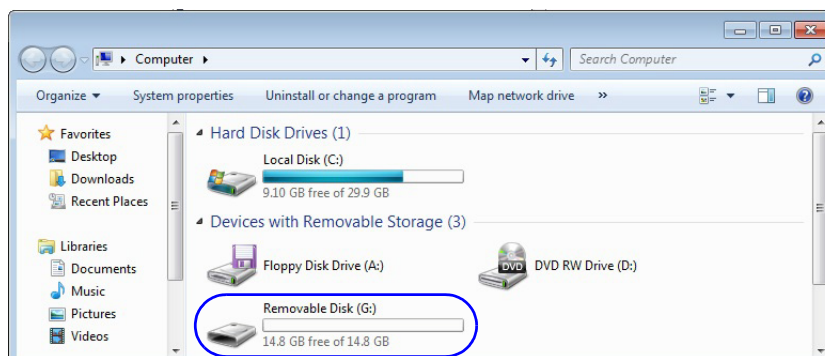


The SD Card Utility window appears.

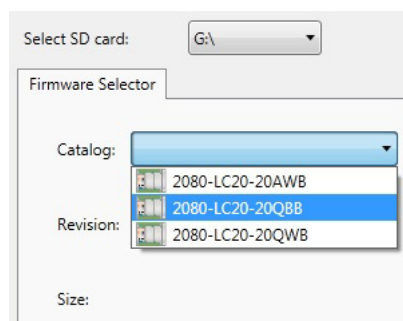


3. Select the drive letter that points to the microSD card on your computer from the pull-down list.

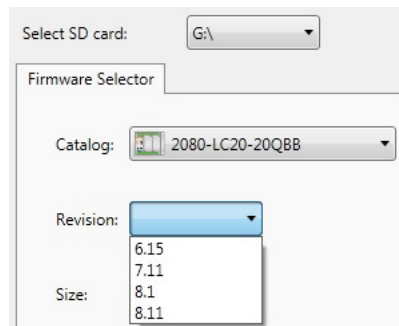
You can check the drive letter by looking in Windows Explorer. For this example, the microSD card is using the drive letter “G”.



4. Select the catalog number of your Micro820 controller.



5. Select the firmware revision you want to flash your Micro820 controller with.

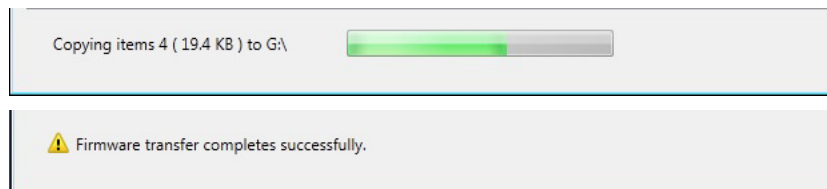


The list of firmware revisions are installed together with Connected Components Workbench. If you require a revision that is not listed, download the firmware from the Product Compatibility and Download Center (PCDC) at <http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page> and install the included ControlFLASH kit.

IMPORTANT You must sign in to the Rockwell Automation website before downloading a firmware revision.

Close and relaunch Connected Components Workbench, then open the SD Card Utility again. The revision should now appear in the list.

6. Click Transfer.
The file is copied to the microSD card.



7. Close the SD Card Utility and proceed to the next step to edit the ConfigMeFirst.txt file.

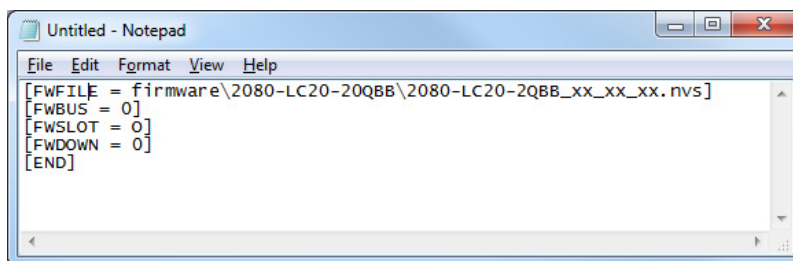
Step 2 – Edit the ConfigMeFirst.txt File

To flash upgrade the controller with the firmware that you have transferred to the microSD card, you need to edit the ConfigMeFirst.txt file with the settings listed below. These settings must be added at the beginning of the file.

New ConfigMeFirst.txt Configuration Settings for Flash Upgrade

Setting	Description
[FWFILE]	File path location of the firmware revision on the microSD card. The default location is in the following format: firmware\<catalog number>\<filename of firmware>
[FWBUS]	This setting is reserved and may be omitted.
[FWSLOT]	This setting is reserved and may be omitted.
[FWDOWN]	Sets whether to upgrade or downgrade the controller firmware from the current revision. 0 = Upgrade firmware; 1 = Downgrade firmware IMPORTANT: Firmware Upgrade will happen if [FWFILE] setting points to a newer version of firmware file compared to current firmware in the controller, irrespective of [FWDOWN] setting.

Example of ConfigMeFirst.txt File for Flash Upgrade



After you have edited the file, insert the microSD card into the controller. Power cycle the controller and the flash upgrade process will begin. Note that the SD status LED will not blink when flash upgrading the firmware from the microSD card is in progress.

Establish Communications between RSLinx and a Micro820 Controller through USB Port on 2080-REMLCD

This quick start shows you how to get RSLinx RSWho to communicate with a Micro820 controller through USB. Micro820 controller uses the 2080_REMLCD_xxxx driver

RSLinx Classic is installed as part of the Connected Components Workbench software installation process. The minimum version of RSLinx Classic with full Micro820 controller support is 3.60.01 (released on December 2013).

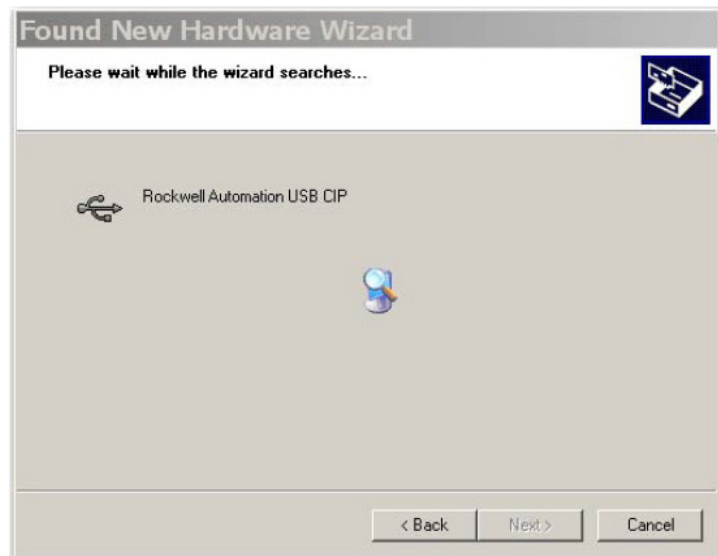
1. Power up the Micro820 controller.
2. Plug USB A/B cable directly between your PC and the USB port on the 2080-REMLCD.
3. Windows should discover the new hardware. Click No, not this time and then click Next.



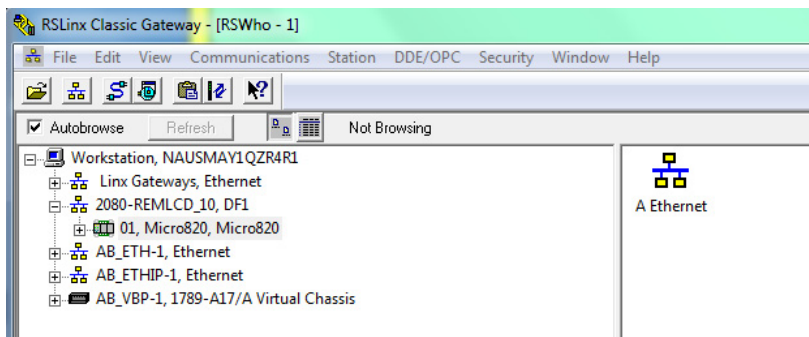
4. Click Install the software automatically (Recommended), and then click Next.



The Wizard searches for new hardware.



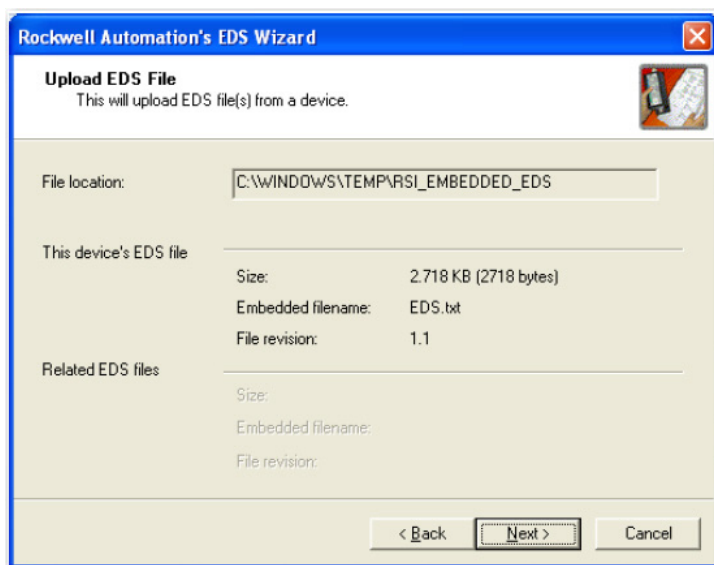
- Open RSLinx Classic and run RSWho by clicking the  icon.

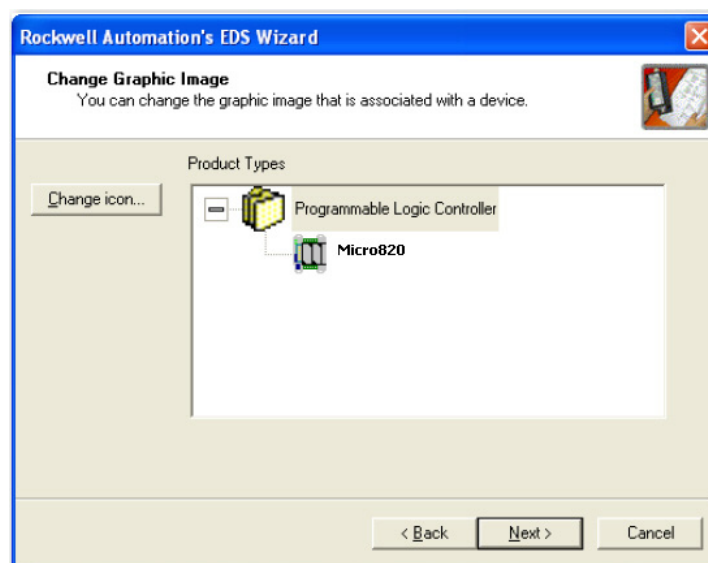
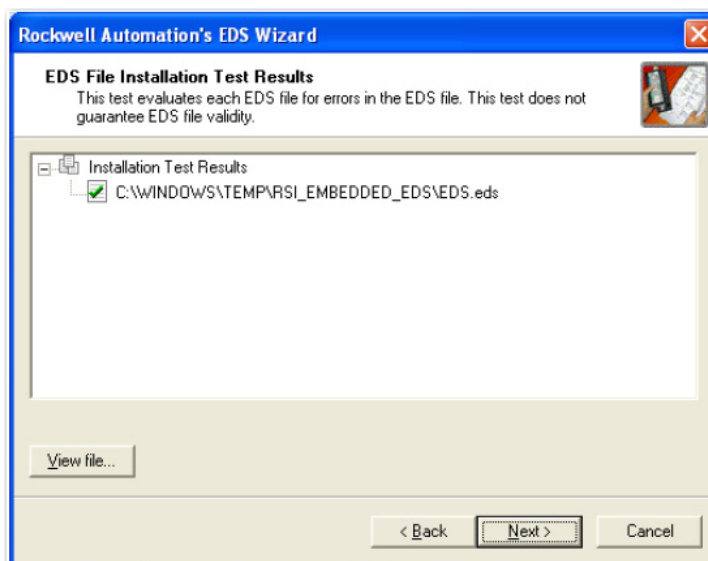


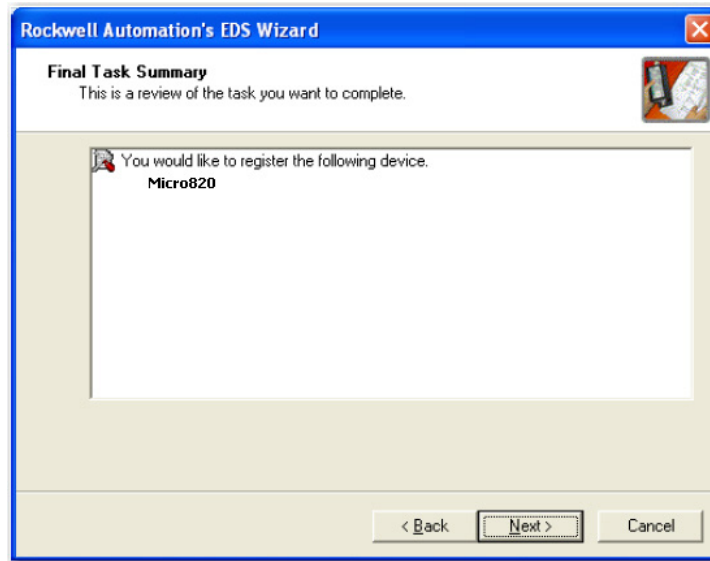
- On the EDS Wizard that appears, click Next to continue.



- Follow the prompts to upload and install the EDS file.







8. Click Finish to complete.

Configure Controller Password

Set, change, and clear the password on a target controller through the Connected Components Workbench software.

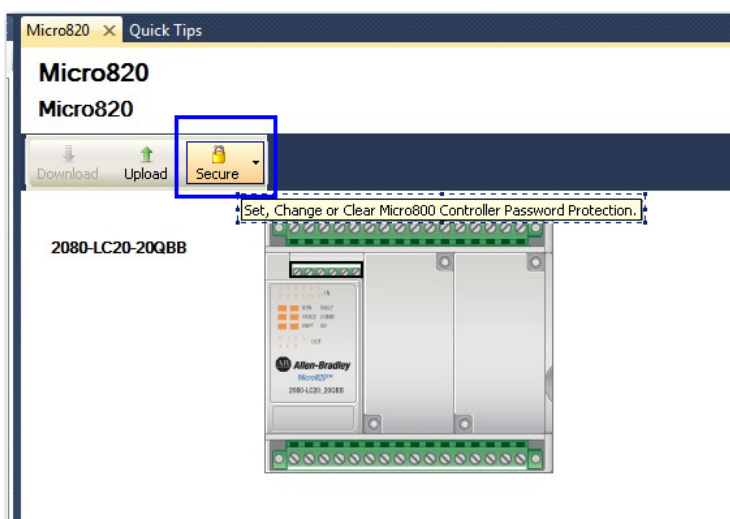
IMPORTANT The following instructions are supported on Connected Components Workbench revision 2 and Micro800 controllers with firmware revision 2. For more information about the controller password feature on Micro800 controllers, see [Controller Security on page 77](#).

Set Controller Password

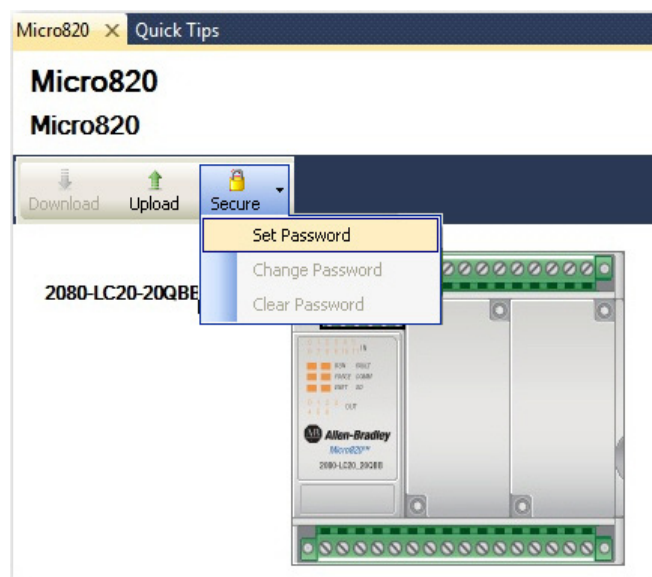
IMPORTANT After creating or changing the controller password, you need to power down the controller in order for the password to be saved.

In the following instructions, the Connected Components Workbench software is connected to the Micro800 controller.

1. On the Connected Components Workbench software, open the project for the target controller.
2. Click Connect to connect to the target controller.
On the Device Details toolbar, roll over the Secure button. The tooltip message “Set, Change, or Clear Micro800 Controller Password Protection” is displayed.



3. Click Secure button. Select Set Password.



4. The Set Controller Password dialog appears. Provide password. Confirm the password by providing it again in the Confirm field.



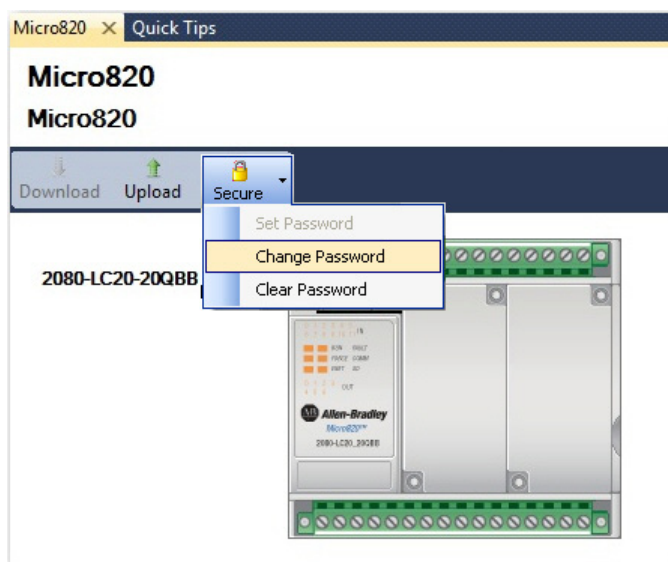
TIP Passwords must have at least eight characters to be valid.

5. Click OK.
Once a password is created, any new sessions that try to connect to the controller will have to supply the password to gain exclusive access to the target controller.

Change Password

With an authorized session, you can change the password on a target controller through the Connected Components Workbench software. The target controller must be in Connected status.

1. On the Device Details toolbar, click Secure button. Select Change Password.



2. The Change Controller Password dialog appears. Enter Old Password, New Password and confirm the new password.



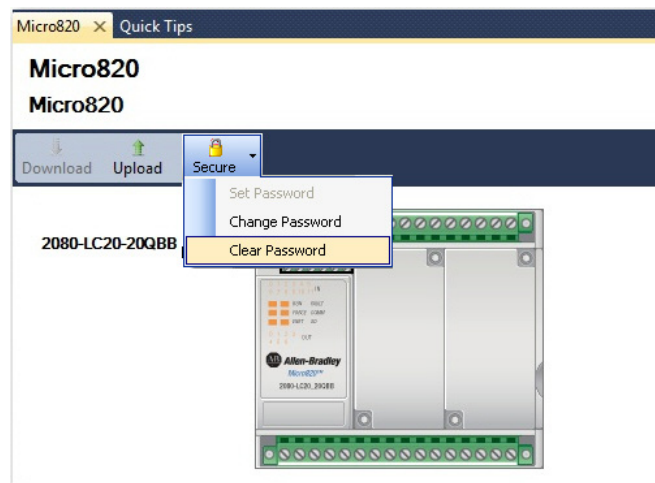
3. Click OK.

The controller requires the new password to grant access to any new session.

Clear Password

With an authorized session, you can clear the password on a target controller through the Connected Components Workbench software.

1. On the Device Details toolbar, click Secure button. Select Clear Password.



2. The Clear Password dialog appears. Enter Password.
3. Click OK to clear the password.

The controller will require no password on any new session.

Forcing I/Os

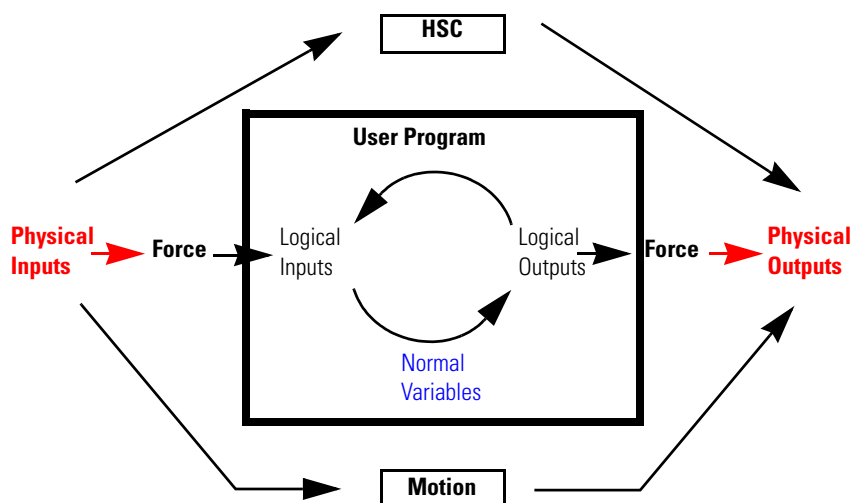
IMPORTANT This section generally talks about forcing I/O in Micro800 controllers. Some elements may not apply to certain models (for example, Micro810 and Micro820 controllers do not support PTO motion).

Inputs are logically forced. LED status indicators do not show forced values, but the inputs in the user program are forced.

Forcing is only possible with I/O and does not apply to user defined variables and non-I/O variables, and special functions such as HSC which execute independently from the User Program scan. For example, for motion, Drive Ready input cannot be forced.

Unlike inputs, outputs are physically forced. LED status indicators do show forced values and the user program does not use forced values.

The following diagram illustrates forcing behavior.



- LED status indicators always match the physical value of I/O
- Normal, non-physical internal variables cannot be forced
- Special functions such as HSC and Motion cannot be forced

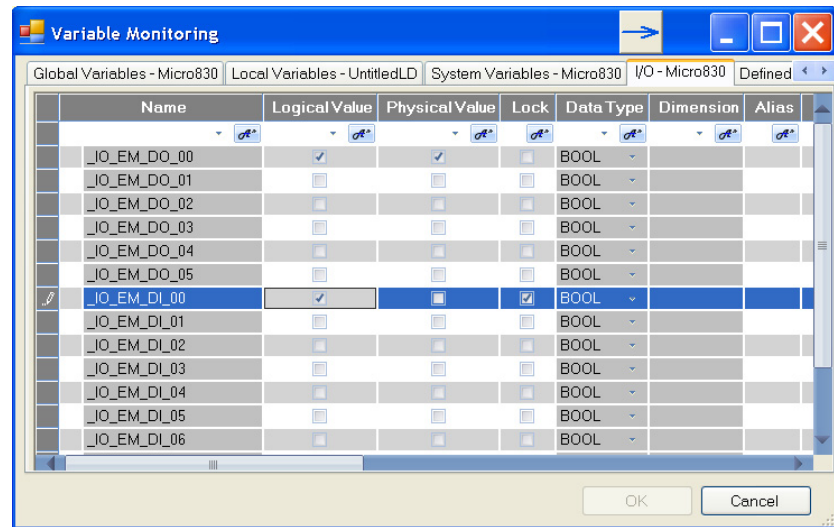


ATTENTION: Forcing variable can result in sudden machine movement, possibly injuring personnel or equipment. Use extreme caution when forcing variables.

Checking if Forces (locks) are Enabled

If Connected Components Workbench is available, check the Variable Monitor while debugging online. Forcing is performed by first locking an I/O variable and then setting the Logical Value for Inputs and Physical Value for Outputs.

Remember you cannot force a Physical Input and cannot force a Logical Output.



In many cases, the front of the controller is not visible to the operator and Connected Components Workbench is not online with the controller. If you want the force status to be visible to the operator, then the User Program must read the force status using the SYS_INFO function block and then display the force status on something that the operator can see, such as the human machine interface (HMI), or stack light. The following is an example program in Structured Text.

```

1  (* Read System Information including Force Enable bit *)
2  SYS_INFO_1(TRUE);
3
4  (* Turn on Warning Light if Forces are Enabled *)
5  If SYS_INFO_1.Sts.ForcesInstall = TRUE THEN
6    _IO_EM_DO_05 := TRUE;
7  ELSE
8    _IO_EM_DO_05 := FALSE;
9  END_IF;

```

If the front of the controller is visible, and not blocked by the cabinet enclosure, Micro830 and Micro850 controllers have a Force LED indicator.

I/O Forces After a Power Cycle

After a controller is power cycled, all I/O forces are cleared from memory.

Using Run Mode Change

Run Mode Change allows the user to make small changes to the logic of a running project and immediately testing it out on the controller, without having to go into Program mode or disconnecting from the controller.

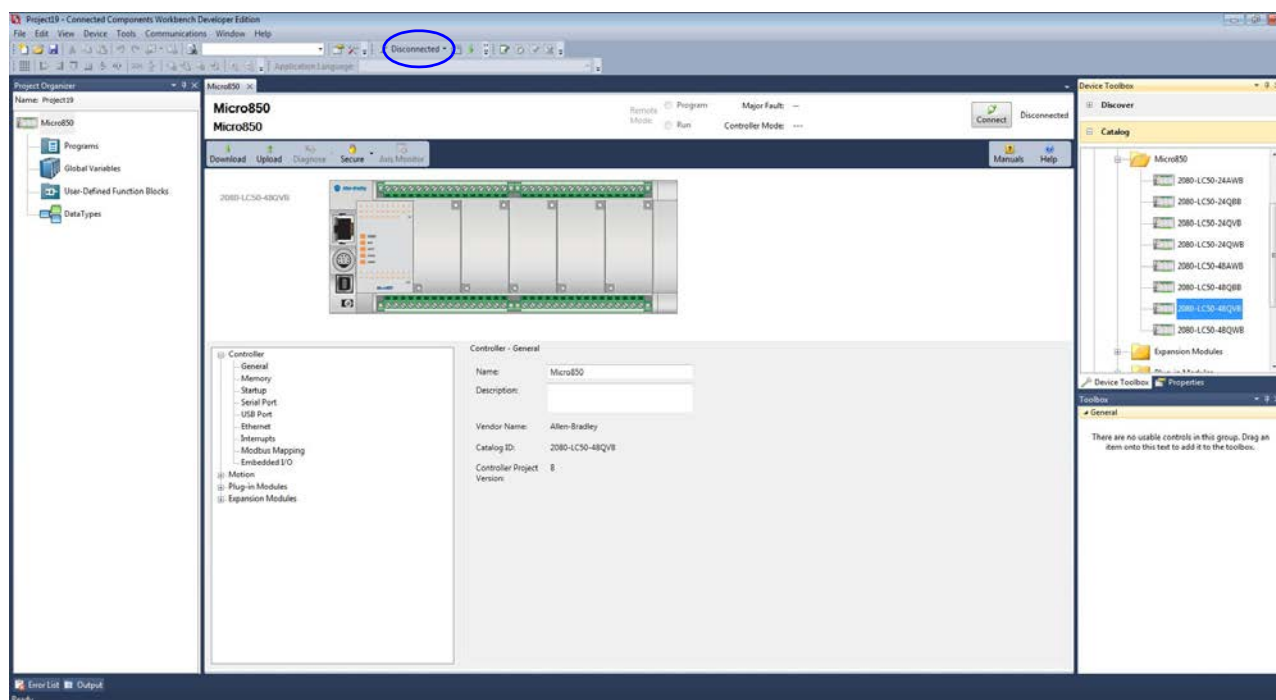
IMPORTANT The following requirements must be met to use Run Mode Change:

- Micro820/Micro830/Micro850 controller firmware revision 8.0 or higher, and
- Connected Components Workbench Developer Edition software, version 8.0 or higher.

The following sample project guides you through the creation of a simple application for a Micro850 controller without any plug-in modules, and how to use the Run Mode Change feature.

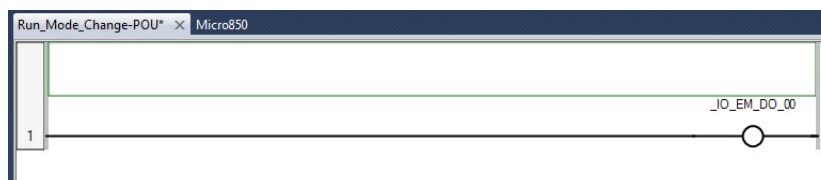
Create the Project

1. Create a new project for a Micro830/Micro850 controller without any plug-ins.
Observe that the controller is disconnected.

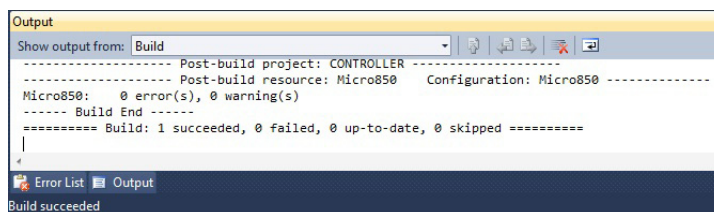


2. Right-click Programs and select Add -> New LD: Ladder Diagram.
3. From the Toolbox, double-click Direct Coil to add it to the rung, or drag and drop Direct Coil onto the rung.

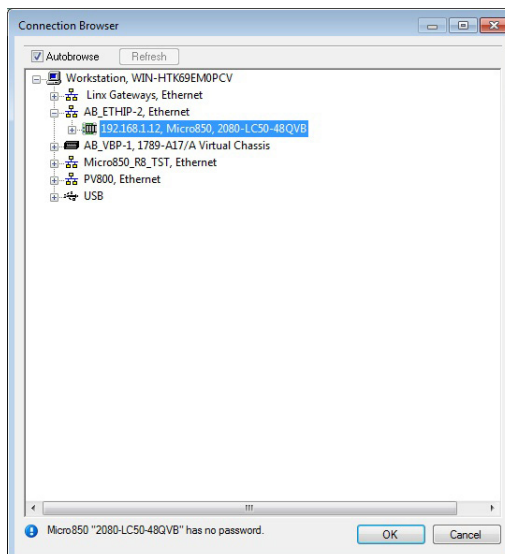
4. Double-click the newly added Direct Coil to bring up the Variable Selector dialog and select “_IO_EM_DO_00”.



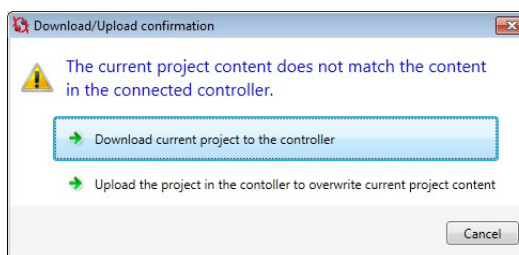
5. Build the project.



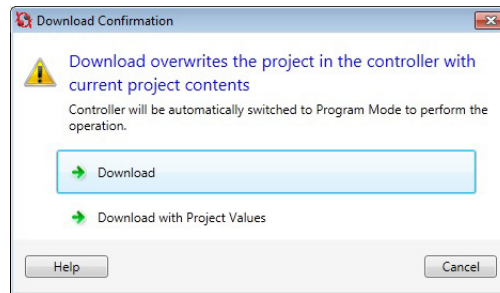
6. Download the project to the controller.
In the Connection Browser dialog, select the Micro850 controller.



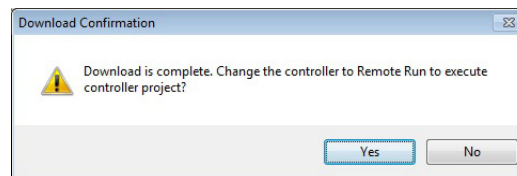
7. Select Download current project to the controller.



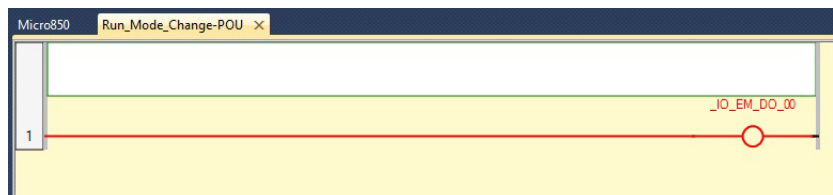
8. Select Download to confirm.



9. When the project has been downloaded to the controller, a prompt asking to change the controller to Remote Run mode appears. Click Yes.



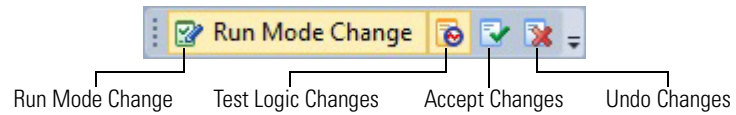
10. Observe that the controller is now in Debug mode.




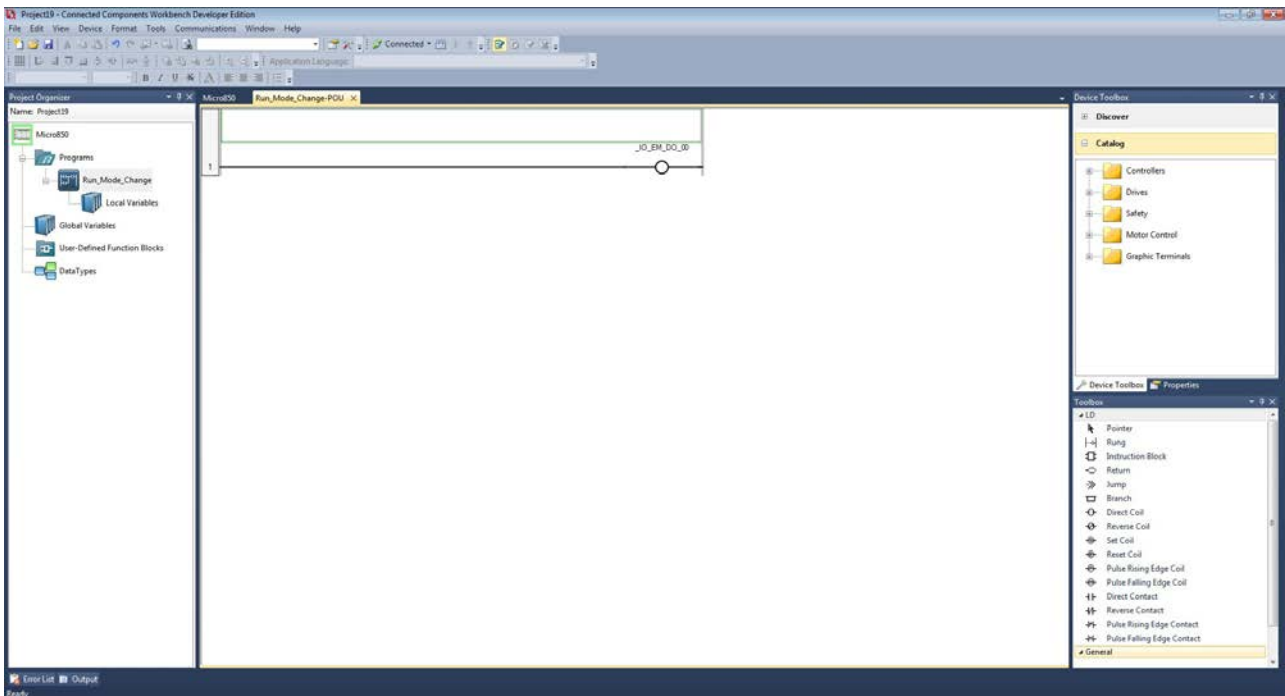
IMPORTANT From Connected Components Workbench version 8.0 onwards, selecting "Yes" to change the controller to Remote Run mode after a downloading a project automatically switches it to Debug mode.

Edit the Project Using Run Mode Change

Run Mode Change Toolbar



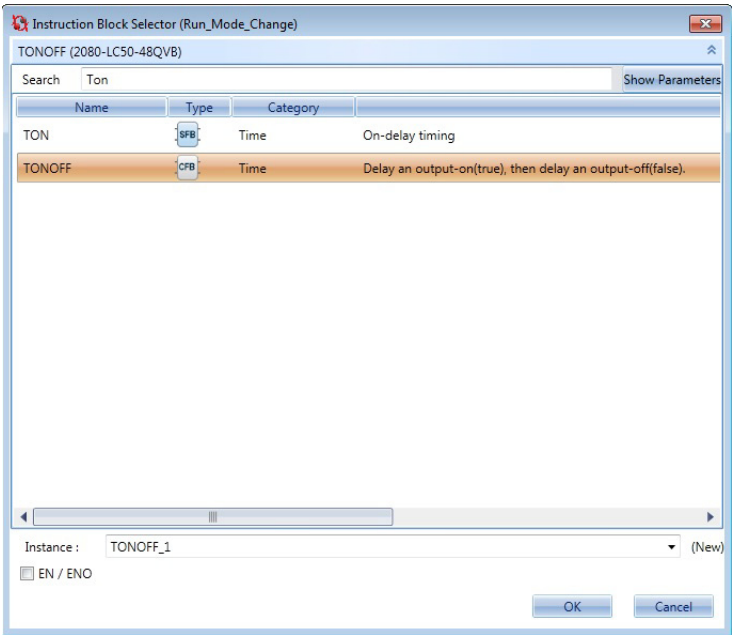
1. Click the Run Mode Change  icon.
Observe that the controller goes into Edit mode and is still connected.



If you add a new variable during RMC, external data access and changing the access type (default is Read/Write) of this new variable is not available until you have chosen to Accept or Undo the Test Logic changes.

2. From the Toolbox, double-click Instruction Block to add it to the rung, or drag and drop Instruction Block onto the rung.

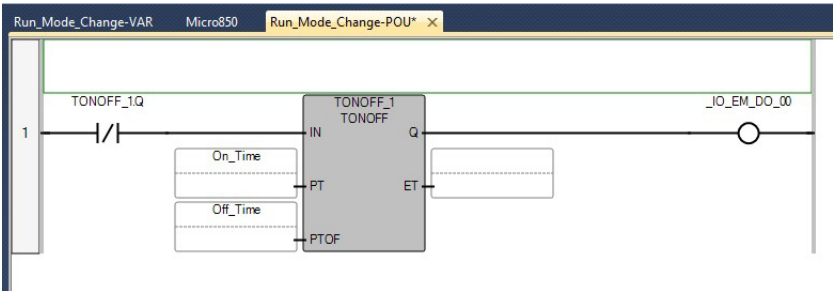
3. Double-click the newly added Instruction Block and select “Timer On/Off “(TONOFF).



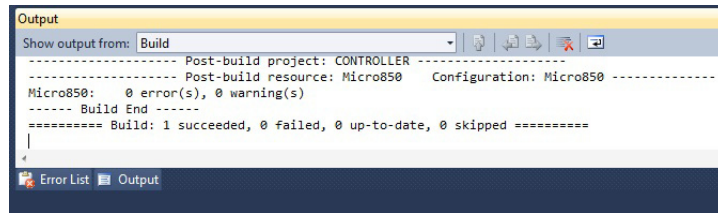
Configure the Instruction Block to trigger every one second.

Run_Mode_Change-POU*								
Name	Alias	Data Type	Dimension	Project Value	Initial Value	Comment	String Size	
TONOFF_1		TONOFF				
On_Time		TIME			T#1s			
Off_Time		TIME			T#1s			

4. From the Toolbox, double-click Reverse Contact to add it to the rung, or drag and drop Reverse Contact onto the run. Place it to left of the recently added Instruction Block.

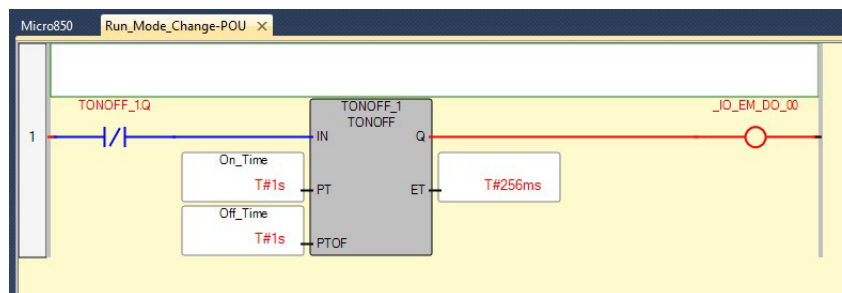


- Click the Test Logic Changes  icon to build the project and download it to the controller.




IMPORTANT When a Test Logic is performed, or undoing changes after the Test Logic is completed, any active communication instructions will be aborted while the changes are downloaded to the controller.

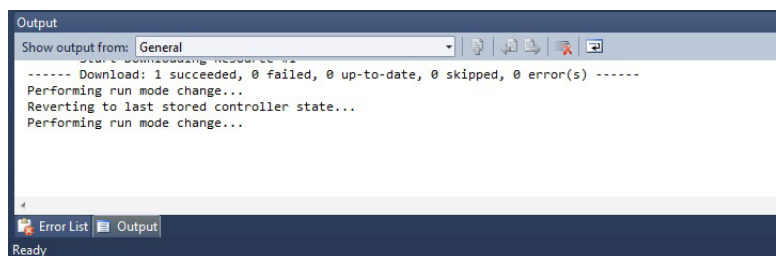
- The controller will automatically go into Debug mode and display the updated project.



- You can now choose to either Undo or Accept the changes to the project.

To Undo the Changes

- Click the Undo Changes  icon.
- The changes will be discarded and the original project will be restored to the controller.




IMPORTANT When a Test Logic is performed, or undoing changes after the Test Logic is completed, any active communication instructions will be aborted while the changes are downloaded to the controller.

Observe that original project is shown and the controller is in Debug mode.



To Accept the Changes

1. Click the Accept Changes  icon.
2. Observe that only the Run Mode Change icon is now enabled and the controller remains in Debug mode.



PID Function Blocks

The PID function block has parameter naming similar to RSLogix 500 and is recommended for users who are already familiar with programming in RSLogix 500. The IPIDCONTROLLER function block has the advantage of supporting auto tune.

Comparison Between IPIDCONTROLLER and PID

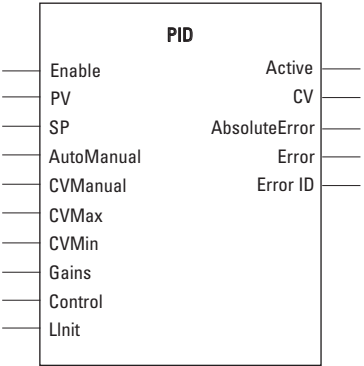
IPIDCONTROLLER	PID	Description
Common parameters		
Process	PV	Process Variable feedback
Setpoint	SP	Set Point input
Output	CV	CV output
Gains.DirectActing	Control	Control direction of process (cooling versus heating).
Gains.ProportionalGain	Gains.Kc	Controller gain for both P and I
Gains.TimeIntegral	Gains.Ti	Time integral value for I
Gains.TimeDerivative	Gains.Td	Time derivative value for D
Gains.DerivativeGain	Gains.FC	A higher filter constant makes CV output more responsive to error. Acts like a derivative gain.
AbsoluteError	AbsoluteError	Absolute value of error
PID specific parameters		
—	CVMin	For limiting CV
	CVMax	For limiting CV
	AutoManual	TRUE = Normal operation of PID. FALSE = Manual operation using CVManual.
	CVManual	CV when in manual mode.
	Enable	TRUE = Start execution with current input parameters. FALSE = CV equals zero.
	Active	TRUE = PID state is running. FALSE = PID state is stopped.
	Error	TRUE = PID has an error. FALSE = PID has no errors.
	ErrorID	PID Error ID
IPIDCONTROLLER specific parameters		

Comparison Between IPIDCONTROLLER and PID

IPIDCONTROLLER	PID	Description
Auto	—	TRUE = Normal operation of PID. FALSE = Output tracks Feedback.
Feedback		Feedback of the control being applied to the process. Usually it's the PID's CV after any limits or manual control has been applied.
AutoTune		TRUE = Autotune. FALSE = No Autotune.
ATParameters		Autotune parameters
ATWarning		Autotune warning
OutGains		Gains from Autotune
Initialize		Used for AutoTune

PID Function Block

This function block diagram shows the arguments in the PID function block.



The following table explains the arguments used in this function block.

PID Arguments

Parameter	Parameter Type	Data Type	Description
Enable	Input	BOOL	Enable instruction. TRUE = Start execution with current input parameters. FALSE = CV equals zero.
PV	Input	REAL	Process Value. This value is typically read from an analog input module. The SI unit must be the same as Setpoint.
SP	Input	REAL	The set point value for the process.
AutoManual	Input	BOOL	Auto or manual mode selection: TRUE = Normal operation of PID. FALSE = Manual operation using CVManual.
CVManual	Input	REAL	Control value input defined for manual mode operation. The valid range for CVManual is: CVMin < CVManual < CVMax

PID Arguments

Parameter	Parameter Type	Data Type	Description
CVMin	Input	REAL	Control value minimum limit. If CV < CVMin, then CV = CVMin. If CVMin > CVMax, and error occurs.
CVMax	Input	REAL	Control value maximum limit. If CV > CVMax, then CV = CVMax. If CVMax < CVMin, an error occurs.
Gains	Input	PID_GAIN S	Gains of PID for controller. Use the PID_GAINS data type to configure the Gains parameter.
Control	Input	BOOL	Control direction of the process: TRUE = Direct acting, such as Cooling. FALSE = Reverse acting, such as Heating.
LInit	Input	BOOL	Reserved for future use.
Active	Output	BOOL	Status of the PID controller: TRUE = PID state is running. FALSE = PID state is stopped.
CV	Output	REAL	The control value output. If any error occurred, CV is 0.
AbsoluteError	Output	REAL	Absolute error is the difference between process value (PV) and setpoint (SV) value.
Error	Output	BOOL	Indicates the existence of an error condition. TRUE = PID has an error. FALSE = PID has no errors.
ErrorID	Output	USINT	A unique numeric that identifies the error. The errors are defined in PID error codes.

GAIN_PID Data Type

Parameter	Parameter Type	Data Type	Description
Kc	Input	REAL	Controller gain for PID. Proportional and Integral are dependent on this gain (≥ 0.0001). Increasing Kc improves response time but also increases overshoot and oscillation of the PID. If Kc is invalid, an error occurs.

GAIN_PID Data Type

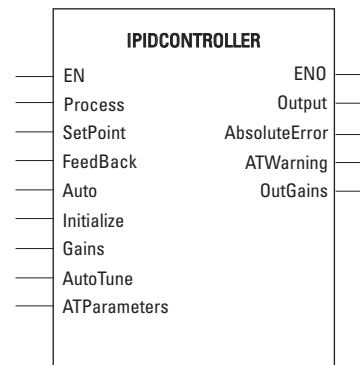
Parameter	Parameter Type	Data Type	Description
Ti	Input	REAL	Time integral constant in seconds (≥ 0.0001). Increasing Ti decreases overshoot and oscillation of the PID. If Ti is invalid, an error occurs.
Td	Input	REAL	Time derivative constant in seconds (≥ 0.0). When Td equals 0, then there is no derivative action and PID becomes a PI controller. Increasing Td reduces the overshoot and removes the oscillation of the PID controller. If Td is invalid, an error occurs.
FC	Input	REAL	Filter constant (≥ 0.0). Recommended range for FC is 0...20. Increasing FC smooths the response of the PID controller. If FC is invalid, an error occurs.

PID Error Codes

Error Code	Description
0	PID is working normally.
1	Kc is invalid.
2	Ti is invalid.
3	Td is invalid.
4	FC is invalid.
5	CVMin > CVMax, or CVMax < CVMin
6	CVManual < CVMin CVManual is invalid.
7	CVManual > CVMax CVManual is invalid.

IPIDCONTROLLER Function Block

This function block diagram shows the arguments in the IPIDCONTROLLER function block.



The following table explains the arguments used in this function block.

IPIDCONTROLLER Arguments

Parameter	Parameter Type	Data Type	Description
EN	Input	BOOL	Function block enable TRUE = Execute function. FALSE = Do not execute function. Applicable to Ladder Diagram programs.
Process	Input	REAL	Process value, which is the value measured from the process output.
SetPoint	Input	REAL	The set point value for the process.
Feedback	Input	REAL	Feedback signal, which is the value of the control variable applied to the process. For example, the feedback can be IPIDCONTROLLER output.
Auto	Input	BOOL	Operating modes of PID controller: TRUE = Normal operation of PID. FALSE = Output tracks Feedback.
Initialize	Input	BOOL	A change in value (TRUE to FALSE or FALSE to TRUE) causes the controller to eliminate any proportional gain during that cycle. It also initializes AutoTune sequences.
Gains	Input	GAIN_PID	Gains PID for IPIDCONTROLLER. Use the GAIN_PID data type to define the parameters for the Gains input.
AutoTune	Input	BOOL	TRUE = When AutoTune is TRUE, and Auto and Initialize are FALSE, the AutoTune sequence is started. FALSE = Do not start AutoTune.
ATParameters	Input	AT_Param	AutoTune parameters Use AT_Param data type to define the parameters for the ATParameters input.
Output	Output	Real	Output value from the controller.
AbsoluteError	Output	Real	Absolute error (Process - SetPoint) from the controller.

IPIDCONTROLLER Arguments

Parameter	Parameter Type	Data Type	Description
ATWarnings	Output	DINT	Warning for the AutoTune sequence. Possible value are: 0 = No auto tune done. 1 = In auto tune mode. 2 = Auto tune done. -1 = Error 1: Input automatically set to TRUE, no auto tune possible. -2 = Error 2: Auto tune error, the ATDynamSet expired.
OutGains	Output	GAIN_PID	Gains calculated from AutoTune Sequences. Use GAIN_PID data type to define the OutGains output.
ENO	Output	BOOL	Enable out. Applicable to Ladder Diagram programs.

GAIN_PID Data Type

Parameter	Type	Description
DirectActing	BOOL	Types of acting: TRUE = Direct acting, output moves same direction as error. That is, the actual process value is greater than the SetPoint and the appropriate controller action is to increase the output. For example, Chilling. FALSE = Reverse acting, output moves opposite direction as error. That is, the actual process value is greater than the Setpoint and the appropriate controller action is to decrease the output. For example, Heating.
ProportionalGain	REAL	Proportional gain for PID (≥ 0.0001). Proportional gain for PID (P_Gain) A higher proportional gain causes a larger change in the output based upon the difference between the PV (measured process value) and SV (set point value). The higher the gain, the faster the error is decreased, but this may result in instability such as oscillations. The lower the gain, the slower the error is decreased, but the system is more stable and less sensitive to large errors. The P_Gain usually is the most important gain to adjust and the first gain to adjust while tuning.

GAIN_PID Data Type

Parameter	Type	Description
TimeIntegral	REAL	Time integral value for PID (≥ 0.0001). Time integral value for PID A smaller integral time constant causes a faster change in the output based upon the difference between the PV (measured process value) and SV (set point value) integrated over this time. A smaller integral time constant decreases the steady state error (error when SV is not being changed) but increases the chances of instability such as oscillations. A larger integral time constant slows down the response of the system and make it more stable, but PV approaches the SV at a slower rate.
TimeDerivative	REAL	Time derivative value for PID (> 0.0). Time derivative value for PID (Td) A smaller derivative time constant causes a faster change in the output based upon the rate of change of the difference between PV (measured process value) and SV (set point value). A smaller derivative time constant makes a system more responsive to sudden changes in error (SV is changed) but increases the chances of instability such as oscillations. A larger time constant makes a system less responsive to sudden changes in error and the system is less susceptible to noise and step changes in PV. TimeDerivative (Td) is related to the derivative gain but allows the derivative contribution to PID to be tuned using time so the sample time must be taken into consideration.
DerivativeGain	REAL	Derivative gain for PID (≥ 0.0). Derivative gain for PID (D_Gain) A higher derivative gain causes a larger change in the output based upon the rate of change of the difference between the PV (measured process value) and SV (set point value). A higher gain makes a system more responsive to sudden changes in error but increases the chances of instability such as oscillations. A lower gain makes a system less responsive to sudden changes in error and makes the system less susceptible to noise and step changes in the PV. If derivative gain is set to zero, it disables the derivative portion of the PID.

AT_Param Data Type

Parameter	Type	Description
Load	REAL	Load parameter for auto tuning. This is the output value when starting AutoTune.
Deviation	REAL	Deviation for auto tuning. This is the standard deviation used to evaluate the noise band needed for AutoTune (noise band = $3 \times \text{Deviation}$) ⁽¹⁾
Step	REAL	Step value for AutoTune. Must be greater than noise band and less than $\frac{1}{2}$ load.
ATDynamSet	REAL	Waiting time in seconds before abandoning auto tune.
ATReset	BOOL	Determines whether the output value is reset to zero after an AutoTune sequence: TRUE = Reset output to zero. FALSE = Leaves output at Load value.

(1) The application engineer can estimate the value of ATParams.Deviation by observing the value of Proces input. For example, in a project that involves the control of temperature, if the temperature stabilizes around 22 °C, and a fluctuation of 21.7...22.5 °C is observed, the value of ATParams.Deviation will be $(22.5-21.7)/2=0.4$.

How to Autotune

Before you autotune, you need to:

- Verify that your system is constant when there is no control. For example, for temperature control, process value should remain at room temperature when there is no control output.
- Configure the set point to 0.
- Set Auto Input to False.
- Set the Gain parameter as follows:

GAIN Parameter Values

GAIN Parameter	Value
DirectActing	According to operation: TRUE (for example, Cooling), or FALSE (for example, Heating)
DerivativeGain	0.5
ProportionalGain	0.0001
TimeIntegral	0.0001
TimeDerivative	0.0

- Set the AT_Parameter as follows:

AT_Parameter Values

AT Parameter	Recommendation
Load	Every 'Load' provides a saturated process value over a period of time. Adjust the load to the value for the saturated process value you want. IMPORTANT: If a load of 40 gives you a process value of 30 °C over a period of time, and you want to tune your system to 30 °C, you should set the load to 40.
Deviation	This parameter plays a significant role in the autotune process. The method of deriving this value is explained later in this section. It is not necessary to set this parameter prior to autotuning. However, if you already know the deviation, it is fine to set it first.
Step	Step value should be between 3*Deviation and ½ load. The step provides an offset for the load during autotuning. It should be set to a value high enough to create a significant change in process value.
ATDynamSet	Set this value to a reasonably long time for the autotune process. Every system is different, so allow more time to a system with a process value that takes longer to react to change.
ATReset	Set this parameter to TRUE to reset the output to zero after the autotune process completes. Set this parameter to FALSE to leave the output at load value after the autotune process completes.

To autotune, perform the following steps:

1. Set the Initialize input to TRUE.
2. Set the AutoTune input to TRUE.
3. Wait for the Process input to stabilize or reach a steady state.
4. Note the temperature fluctuation of the process value.

5. Calculate deviation value with reference to the fluctuation. For example, if the temperature stabilizes around 22 °C (72 °F) with a fluctuation of 21.7...22.5 °C (71...72.5 °F), the value of 'ATParams.Deviation' is:

$$\text{For } ^\circ\text{C: } \frac{22.5 - 21.7}{2} = 0.4 \quad \text{For } ^\circ\text{F: } \frac{72.5 - 71}{2} = 0.75$$

6. Set the deviation value, if you have not set it yet.
7. Change the initialize input to FALSE.
8. Wait until the 'AT_Warning' shows 2. The autotune process is successful.
9. Get the tuned value from the 'OutGains'.

How Autotune Works

The auto tune process begins when the 'Initialize' is set to FALSE (Step 7.) At this moment, the control output increases by the amount of 'Step' and the process waits for the process value to reach or exceeds 'first peak'.

First peak is defined as:

For Direct Operation: First peak = PV1 - (12 x Deviation)

For Reverse Operation: First peak = PV1 + (12 x Deviation)

Where PV1 is the process value when Initialize is set to FALSE.

Once the process value reaches first peak, the control output reduces by the amount of Step and waits for the process value to drop to the second peak.

Second peak is defined as:

For Direct Operation: Second peak = PV1 - (3 x Deviation)

For Reverse Operation: Second peak = PV1 + (3 x Deviation)

Once the process value reaches or falls below second peak, calculations commence and a set of gain will be generated to parameter OutGains.

Troubleshooting an Autotune Process

You can tell what is going on behind the autotune process from the sequences of control output. Here are some known sequences of control output and what it means if autotune fails. For the ease of illustrating the sequence of control output, we define:

Load: 50

Step: 20

Output Sequence 1: 50 -> 70 -> 30

Sequence Condition	Autotune Result	Action for Autotune Fail
Process value reached 'first peak' and 'second' peak in time	Likely successful	NA

Output Sequence 2: 50 -> 70 -> 50

Sequence Condition	Autotune Result	Action for Autotune Fail
Process value not able to reach 'first peak'	Likely unsuccessful	Reduce Deviation or Increase Step

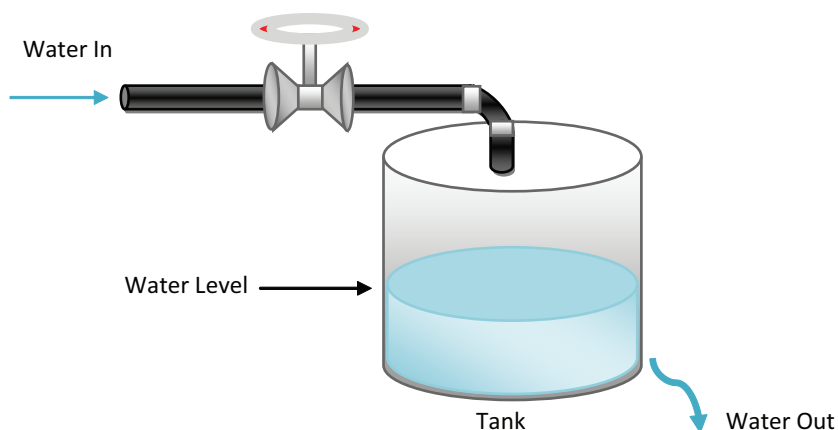
Output Sequence 3: 50 -> 70 -> 30 -> 50

Sequence Condition	Autotune Result	Action for Autotune Fail
Process value not able to reach second peak	Likely unsuccessful	Increase Deviation or increase Step

Output Sequence 4: 50 -> 70

Sequence Condition	Autotune Result	Action for Autotune Fail
Process value not able to reach First peak in time	Likely unsuccessful	Increase ATDynamSet

PID Application Example



The illustration above shows a basic water level control system, to maintain a preset water level in the tank. A solenoid valve is used to control incoming water, filling the tank at a preset rate. Similarly, outflowing water is controlled at a measureable rate.

IPID Autotuning for First and Second Order Systems

Autotune of IPID can only work on first and second order systems.

A first order system can be described by a single independent energy storage element. Examples of first order systems are the cooling of a fluid tank, the flow of fluid from a tank, a motor with constant torque driving a disk flywheel or an

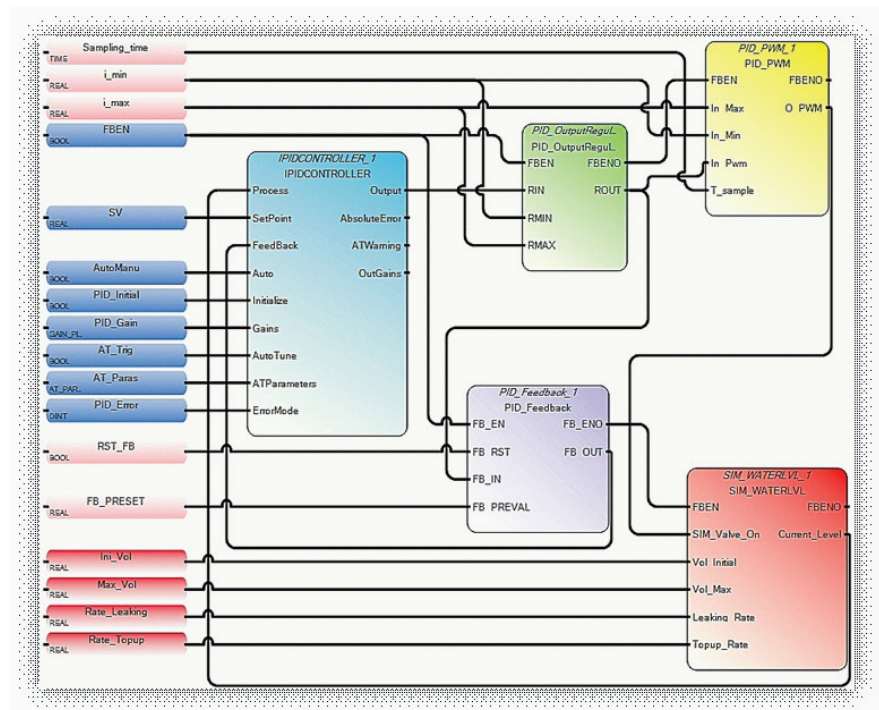
electric RC lead network. The energy storage element for these systems are heat energy, potential energy, rotational kinetic energy and capacitive storage energy, respectively.

This may be written in a standard form such as $f(t) = \tau dy/dt + y(t)$, where τ is the system time constant, f is the forcing function and y is the system state variable.

In the cooling of a fluid tank example, it can be modeled by the thermal capacitance C of the fluid and thermal resistance R of the walls of the tank. The system time constant will be RC , the forcing function will be the ambient temperature and the system state variable will be the fluid temperature.

A second order system can be described by two independent energy storage elements which exchange stored energy. Examples of second order systems are a motor driving a disk flywheel with the motor coupled to the flywheel via a shaft with torsional stiffness or an electric circuit composed of a current source driving a series LR (inductor and resistor) with a shunt C (capacitor). The energy storage elements for these systems are the rotational kinetic energy and torsion spring energy for the former and the inductive and capacitive storage energy for the latter. Motor drive systems and heating systems can be typically modeled by the LR and C electric circuit.

PID Code Sample



The illustration above shows sample code for controlling the PID application example shown before. Developed using Function Block Diagrams, it consists of

a pre-defined function block, IPIDCONTROLLER, and four user-defined function blocks. These four are:

- PID_OutputRegulator

This user-defined function block regulates the output of IPIDCONTROLLER within a safe range to ensure that there is no damage to the hardware used in the process.

IF $R_{MIN} \leq R_{IN} \leq R_{MAX}$, then $R_{OUT} = R_{IN}$,

IF $R_{IN} < R_{MIN}$, then $R_{OUT} = R_{MIN}$,

IF $R_{IN} > R_{MAX}$, then $R_{OUT} = R_{MAX}$.

- PID_Feedback

This user defined function block acts as a multiplexer.

IF "FB_RST" is false, $FB_OUT = FB_IN$;

If "FB_RST" is true, then $FB_OUT = FB_PREVAL$.

- PID_PWM

This user defined function block provides a PWM function, converting a real value to a time related ON/OFF output.

- SIM_WATERLVL

This user defined function block simulates the process depicted in the application example shown before.

IMPORTANT User Program Scan Time is Important

The autotuning method needs to cause the output of the control loop to oscillate. In order to identify the oscillation period, the IPID must be called frequently enough to be able to sample the oscillation adequately. The scan time of the user program must be less than half the oscillation period. In essence the Shannon, or Nyquist-Shannon, or the sampling theorem must be adhered to.

In addition, it is important that the function block is executed at a relatively constant time interval.

Modbus Mapping for Micro800

Modbus Mapping

All Micro800 controllers (except the Micro810 12-point models) support Modbus RTU over a serial port through the embedded, non-isolated serial port. The 2080-SERIALISOL isolated serial port plug-in module also supports Modbus RTU. Both Modbus RTU master and slave are supported. Although performance may be affected by the program scan time, the 48-point controllers can support up to six serial ports (one embedded and five plug-ins), and so consequently, six separate Modbus networks.

Endian Configuration

Modbus protocol is big-endian in that the most significant byte of a 16-bit word is transmitted first. Micro800 is also big-endian, so byte ordering does not have to be reversed. For Micro800 data types larger than 16-bits (for example, DINT, LINT, REAL, LREAL), multiple Modbus addresses may be required but the most significant byte is always first.

Mapping Address Space and supported Data Types

Since Micro800 uses symbolic variable names instead of physical memory addresses, a mapping from symbolic Variable name to physical Modbus addressing is supported in Connected Components Workbench software, for example, InputSensorA is mapped to Modbus address 100001.

By default Micro800 follows the six-digit addressing specified in the latest Modbus specification. For convenience, conceptually the Modbus address is mapped with the following address ranges. The Connected Components Workbench mapping screen follows this convention.

Variable Data Type	0 - Coils 000001 to 065536		1 - Discrete Inputs 100001 to 165536		3 - Input Registers 300001 to 365536		4 - Holding Registers 400001 to 465536	
	Supported	Modbus Address Used	Supported	Modbus Address Used	Supported	Modbus Address Used	Supported	Modbus Address Used
BOOL	Y	1	Y	1				
SINT	Y	8	Y	8				
BYTE	Y	8	Y	8				
USINT	Y	8	Y	8				
INT	Y	16	Y	16	Y	1	Y	1
UINT	Y	16	Y	16	Y	1	Y	1

Variable Data Type	0 - Coils 000001 to 065536		1 - Discrete Inputs 100001 to 165536		3 - Input Registers 300001 to 365536		4 - Holding Registers 400001 to 465536	
	Supported	Modbus Address Used	Supported	Modbus Address Used	Supported	Modbus Address Used	Supported	Modbus Address Used
WORD	Y	16	Y	16	Y	1	Y	1
REAL	Y	32	Y	32	Y	2	Y	2
DINT	Y	32	Y	32	Y	2	Y	2
UDINT	Y	32	Y	32	Y	2	Y	2
DWORD	Y	32	Y	32	Y	2	Y	2
LWORD	Y	64	Y	64	Y	4	Y	4
ULINT	Y	64	Y	64	Y	4	Y	4
LINT	Y	64	Y	64	Y	4	Y	4
LREAL	Y	64	Y	64	Y	4	Y	4

NOTE: Strings are not supported.

In order to make it easier to map variables to five-digit Modbus addresses, the Connected Components Workbench mapping tool checks the number of characters entered for the Modbus Address. If only five-digits are entered, the address is treated as a five-digit Modbus address. This means that the Coils are mapped from 00001...09999, Discrete Inputs are mapped from 10001...19999, Input Registers are mapped from 30001...39999, and Holding Registers are mapped from 40001...49999.

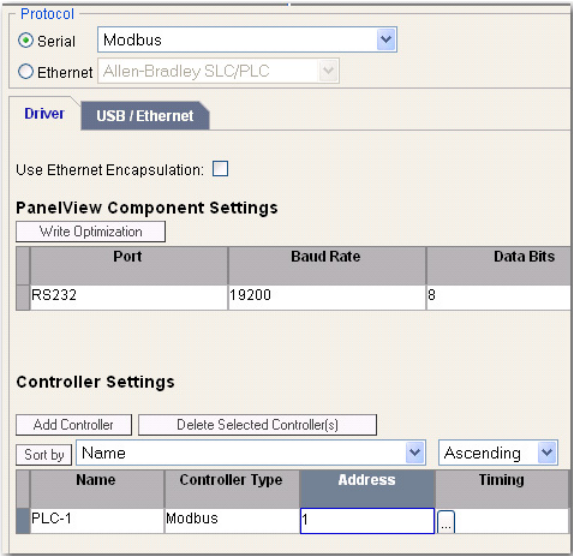
Example 1, PanelView Component HMI (Master) to Micro800 (Slave)

The embedded serial port is targeted for use with HMIs using Modbus RTU. The maximum recommended cable distance is 3 meters. Use the 2080-SERIALISOL serial port plug-in module if longer distances or more noise immunity is needed.

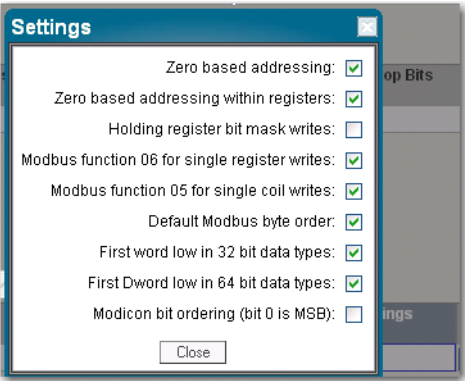
The HMI is typically configured for Master and the Micro800 embedded serial port is configured for Slave.

From the default Communications Settings for a PanelView Component HMI (PVC), there are three items that must be checked or modified in order to set up communications from PVC to Micro800.

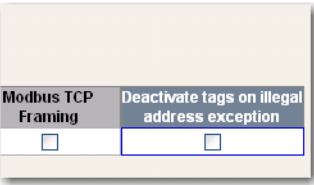
1. Change from DF1 to Modbus protocol.



2. Set the Address of Micro800 slave to match the serial port configuration for the controller.



3. Deactivate Tags on Error. This is to prevent the requirement of power cycling PVC when new Modbus Mappings are downloaded from Connected Components Workbench to Micro800 controller.



Example 2, Micro800 (Master) to PowerFlex 4M Drive (Slave)

The following is the overview of the steps to be taken for configuring a PowerFlex 4M drive.

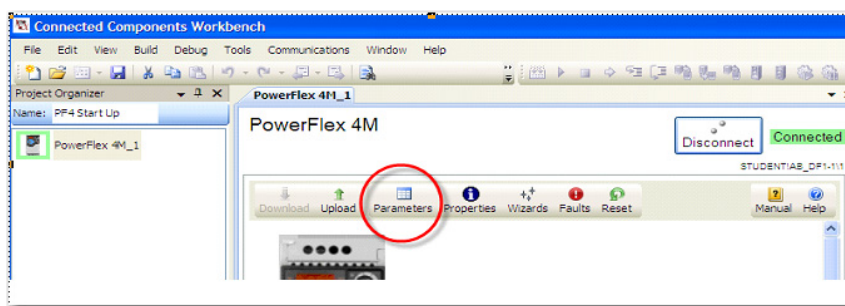
Parameter numbers listed in this section are for a PowerFlex 4M and will be different if you are using another PowerFlex 4-Class drive.

Parameter Name	Parameter Number						
	4M	4	40	40P	400	400N	400P
Start Source	P106	P36					
Speed Reference	P108	P38					
Comm Data Rate	C302	A103			C103		
Comm Node Addr	C303	A104			C104		
Comm Loss Action	C304	A105			C105		
Comm Loss Time	C305	A106			C106		
Comm Format	C306	A107			C102		

- Connect the 1203-USB to the PowerFlex Drive and to the Computer.
- Launch Connected Components Workbench, Connect to the Drive and set parameters.

To configure PowerFlex 4M, perform the following steps:

1. Double-click the PowerFlex 4M if it is not already open in Connected Components Workbench.
2. Click Connect.
3. In the Connection Browser, expand the AB_DF1 DH+ Driver. Select the AB DSI (PF4 Port) and click OK.
4. Once the Drive has connected and been read in, select the Start up wizard and change the following items. Select Finish to save the changes to the drive.
 - Select the Comm Port as the Speed Reference. Set P108 [Speed Reference] to 5 (Comm Port).
 - Set Start Source to Comm Port. Set P106 [Start Source] to 5 (Comm Port).
 - Defaults for the remaining Inputs
 - Accept Defaults for the remainder and click Finish.
5. Select Parameters from the Connected Components Workbench window.



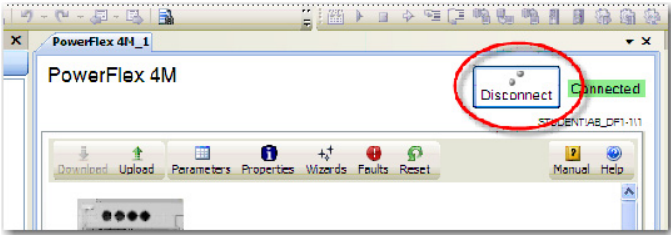
- The Parameter window opens. Resize it to view the parameters. From this window, you can view and set data values of Parameters.

#	Name	Value	Units	Internal Value	Default	Min	Max
1	Output Freq	0.0	Hz	0	0.0	0.0	999.9
2	Commanded Freq	0.0	Hz	0	0.0	0.0	999.9
3	Output Current	0.00	A	0	0.00	0.00	9.00
4	Output Voltage	0.0	V	0	0.0	0.0	999.9
5	DC Bus Voltage	314	V	314	0	0	1200
6	Drive Status	0000000000000010		2	00000000000000...	00000000000000...	0000000000001...

- From the Parameter window, change the following Parameters to set the communications for Modbus RTU so that the PowerFlex 4M Drive will communicate with Micro830/850 via Modbus RTU communication.

Parameter	Description	Setting
C302	Comm. Data Rate (Baud Rate) 4 = 19200 bps	4
C303	Communication Node Address (address range is 1...127)	2
C304	Comm. Loss Action (Action taken when loss communication) 0 = Fault with coast stop	0
C305	Comm. Loss Time (Time remain in communication before taking action set in C304) 5 sec (Max. 60)	5
C306	Comm. Format (Data/Parity/Stop) RTU:8 Data Bit, Parity None, 1 Stop bit	0

- Disconnect the Communications and save your project.



- Turn off the power to the drive until the PowerFlex 4M display blanks out completely, then restore power to the PowerFlex 4M.
The drive is now ready to be controlled by Modbus RTU communication commands initiated from the Micro830/850 controller.

Modbus devices can be 0-based (registers are numbered starting at 0), or 1-based (registers are numbered starting at 1). When PowerFlex 4-Class drives are used with Micro800 family controllers, the register addresses listed in the PowerFlex User Manuals need to be offset by n+1.

For example, the Logic Command word is located at address 8192, but your Micro800 program needs to use 8193 (8192+1) to access it.

Modbus Address (n+1 value shown)

8193 Logic Command word (Stop, Start, Jog, etc.)

8194	Speed Reference word xxx.x format for 4/4M/40, where "123" = 12.3 Hz xxx.xx format for 40P/400/400N/400P, where "123" = 1.23 Hz
8449	Logic Status word (Read, Active, Fault, and so on.)
8452	Speed Feedback word (uses same format as Speed Reference)
8450	Error Code word
(n+1)	To access Parameter 'n'

TIP

- If the respective PowerFlex drive supports Modbus Function Code 16 Preset (Write) Multiple Registers, use a single write message with a length of "2" to write the Logic Command (8193) and Speed reference (8194) at the same time.
- Use a single Function Code 03 Read Holding Registers with a length of "4" to read the Logic status (8449), Error Code (8450), and Speed Feedback (8452) at the same time.

Refer to the respective PowerFlex 4-Class drive User Manual for additional information about Modbus addressing. (See Appendix E – Modbus RTU Protocol, on publication [22C-UM001G](#)).

Performance

The performance of MSG_MODBUS (Micro800 is master) is affected by the Program Scan because messages are serviced when the message instruction is executed in a program. For example, if the program scan is 100 ms and six serial ports are used, then the theoretical maximum for serial ports is 60 messages/second total. This theoretical maximum may not be possible since MSG_MODBUS is a master/slave request/response protocol, so performance is affected by several variables such as message size, baud rate, and slave response time.

The performance of Micro800 when receiving Modbus request messages (Micro800 is slave) is also affected by the Program Scan. Each serial port is serviced only once per program scan.

Symbols

__SYSVA_CYCLECNT 72
__SYSVA_TCYCURRENT 72
__SYSVA_TCYMAXIMUM 72

Numerics

2080-LCD 77
2080-MEMBAK-RTC 1, 77
2080-PS120-240VAC 32
2080-REMLCD 1, 35, 82, 83
 advanced set 87
 analog calibration 88
 backlight parameters 90
 customization 85
 edit variables 88
 function blocks 89
 I/O Status 87
 main menu 86
 memory card 88
 mode switch 87
 modes of operation 84
 operational limit 88
 rating 84
 resolution 83
 security 87
 startup screen 85
 text display mode 84
 USB Mode 84
 user-defined screens 89
 variable 87
 variable edit 88

A

About Your Controller 7
Additional Resources iii
analog cable grounding 49
analog channel wiring guidelines 48
analog input
 from transmitters 54
analog inputs
 analog channel wiring guidelines 48
analog output 54
array of strings 59
arrays 59
ASCII 55, 57, 60, 62
 configuration 67
ASCII devices 57
AutoTune 178

B

backup 1, 3
baud rate 4, 64
before calling for assistance 145
BOOL 59

C

calibration 52
calling for assistance 145
CE mark 19
certifications 18
Checking if Forces (locks) are Enabled 162
CIP Client Messaging 59
CIP communications pass-thru 60
CIP Generic 59
CIP Serial 58, 62
 Parameters 64
CIP Serial Client/Server 55, 57
CIP Serial Driver
 configure 63
 parameters 64
CIP Symbolic 58, 59
 supported data types 59
CIP Symbolic Addressing 59
CIP Symbolic Client/Server 55, 58
communication connections 55
communication protocols 55
communications
 ports 55
CompactLogix 58
Compliance to European Union Directive
 EMC Directive 19
 Low Voltage Directive 19
Compliance to European Union Directives 19
ConfigMeFirst.txt 3
 errors 98
Connected Components Workbench iv, 7, 68, 71, 75, 77, 78, 79, 82, 84, 89, 99, 104
Connected Components Workbench Online 57
Connected Components Workbench. 85
connection limits 56
control line 64
ControlFlash 82
controller
 grounding 41
 I/O wiring 48
 minimizing electrical noise 48
 preventing excessive heat 25
Controller Error Recovery Model 144
controller load 73
controller lock 79

controller password 77, 78

ControlLogix 58

D

data log 1, 99

- data types 103
- directory structure 100
- execution rules 103
- specifications 100
- timing diagram 102

data types 59

datasets 99, 100

DATE 59

DF1 mode 64

DF1 point-to-point connection 61

DHCP Client 55

DIN rail mounting 32

DINT 59

disconnecting main power 22

discover 78

DLG

- function block status 101
- function error ID list 101
- input and output parameters 101

DLG_ERR_DATAFILE_ACCESS 100

duplicate packet detection 64

E

embedded responses 65

Embedded Serial Port Cables 5

EMC Directive 19

Endian Configuration 183

ENET status 2

ENQ retries 65

error codes 135, 136

error detection 64

error recovery model 144

Ethernet 58, 85

- configuration settings 68
- transfer rate 55

Ethernet support 5

EtherNet/IP 58

EtherNet/IP Client 58

EtherNet/IP Client/Server 55

EtherNet/IP Server 58

European Union Directive compliance 19

- EMC Directive 19

exclusive access 77

Execution Rules 72

execution rules 99

F

Fault status 2

Force status 2

force status 133

Forcing I/Os 162

G

general considerations 20

grounding the controller 41

Guidelines and Limitations for Advanced Users 75

H

hardware

- features 1

Hardware Features 1

Hardware Overview 1

heat protection 25

housekeeping 71, 99

I

I/O Status 84, 86

input states on power down 25

inputs

- embedded 1

installation

- DIN rail mounting 32
- Micro820 controller 31
- module spacing 32
- mounting dimensions 31
- panel mounting 32
- dimensions 33

Installation Considerations 20

Installing Your Controller 31

INT 59

IP address

- exclusions 69
- rules 69

IP65 83

IPIDCONTROLLER 175

- parameters 172, 173, 175

isolation transformers

- power considerations 24

K

KEY_READ_REM 89, 91
keyswitch 82

L

LCD_BKLT_REM 89, 90
LCD_REM 89, 90
LINT 59
Literature Library 35
LREAL 59

M

Mapping Address Space and supported Data Types 183
master control relay 25
 emergency-stop switches 26
 using ANSI/CSA symbols schematic 29
 using IEC symbols schematic 28
master control relay circuit
 periodic tests 23
Micro800 cycle or scan 71
Micro800 Remote LCD 83
Micro830 Controllers 2
microSD 2, 3, 77
 card slot 3
 Class 4 3
 Class 6 and 10 SDSC and SDHC 3
microSD card 100
 installation 34
minimizing electrical noise 48
minimizing electrical noise on analog channels 49
Modbus 57
Modbus Mapping 183
Modbus Mapping for Micro800 183
Modbus RTU 55, 57, 58, 62
 advanced parameters 66
 configuration 65
 parameters 66
Modbus TCP Client 58
Modbus TCP Client/Server 55, 58
Modbus/TCP Server 58
Modbus/TCP server 58
module spacing 32
motor starters (bulletin 509)
 surge suppressors 40
MSG_CIPGENERIC 59
MSG_CIPSYMBOLIC 59
MSG_MODBUS 57
MSG_MODBUS2 58

N

NAK retries 65
Normal Operation 134
North American Hazardous Location Approval 22

O

outputs
 embedded 1
Overview of Program Execution 71

P

panel mounting 32
PanelView Component 58
parity 64
password 77
password recovery 82
Performance, MSG_MODBUS 188
PID 172
PID Application Example 180
PID Code Sample 181
PID Function Blocks 171
POU (Program Organizational Unit) 72
power considerations
 input states on power down 25
 isolation transformers 24
 loss of power source 24
 other line conditions 25
 overview 24
 power supply inrush 24
power distribution 23
power source
 loss of 24
power supply 2
 requirement 2
power supply inrush
 power considerations 24
powerup settings 3
preventing excessive heat 25
program mode 99
program scan 99
program scan cycle 73

Q

Quickstarts 147

R

REAL 59

recipe 1, 104

- data types 103
- directory structure 105
- function block errors 106
- function block parameters 105
- function block status 106
- specifications 104

recipe sets 104

Remote LCD

- configuration 62

resistance input 52

restore 1, 3

RJ-45 connector

- location 34

RJ-45 Ethernet cable 5

RJ-45 Ethernet connector 2

RJ45 Ethernet port 55

RJ-45 ethernet port 5

RS232 4

RS-232/485 combo port 55

RS-232/485 serial port 55

RS232/RS485 2, 4, 41

- pin definition 4

RS232/RS485 serial port 4

RSLinux 58

Run Mode Change (RMC) 8

- benefits 8
- limitations 12
- RMC memory 10
- uncommitted changes 9
- using 164

Run Mode Configuration Change (RMCC) 13

- loop-back message 13
- using EtherNet/IP 17
- using Modbus RTU 15
- verify IP address change 18
- verify node address change 16

run status 2

S

safety circuits 23

Safety Considerations 22

safety considerations 22

- disconnecting main power 22
- hazardous location 22
- master control relay circuit

periodic tests 23

periodic tests of master control relay circuit 23

power distribution 23

safety circuits 23

security 77

serial communications status 133

serial port

- configure 62

serial port cables 4

Shutdown 62

SINT 59

Sockets Client/Server 55, 60

station address 64

status indicator 2

- Comm 2
- fault status 133
- input status 2, 133
- network status 134
- output 2
- run status 133
- SD 2
- serial communications 133

status indicators 2

Status Indicators on the Controller 133

STRING 59

surge suppression 38

surge suppressors

- for motor starters 40
- recommended 40
- using 38

T

terminal blocks 2

thermistor resistance 52

TIME 59

transmit retries 65

troubleshooting 133

U

UDINT 59

UINT 59

ULINT 59

USB mode 84

USB port 84, 85

User Defined Function (UDF) 71, 75

User Defined Function Block (UDFB) 71, 75

using emergency-stop switches 26

USINT 59

V

validate IP address 69

variable retainment 74

voltage input 52

W

wiring 37

fixed terminal blocks 38

removable terminal blocks 38

RS232/RS485 terminal block 38

wiring diagrams 41

Wiring Examples 50

wiring recommendation 37

Wiring Your Controller 37

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products.

At <http://www.rockwellautomation.com/support/>, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

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

Installation Assistance

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

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

New Product Satisfaction Return

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

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

Documentation Feedback

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

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

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

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

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

Rockwell Automation Publication 2080-UM005D-EN-E - November 2016

Supersedes Publication 2080-UM005C-EN-E - November 2015

Copyright © 2016 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.