User Manual

Allen-Bradley

Micro820 Programmable Controllers

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

Allen-Bradley • Rockwell Software

Rockwell Automation

Important User Information

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

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

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

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

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

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

⚠	**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
⚠	**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
⚡	**SHOCK HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
♨	**BURN HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

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 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.
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.

Resource	Description
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 SGI-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://ab.rockwellautomation.com/Programmable-Controllers/Connected-Components-Workbench-Software.

Preface

Who Should Use this Manual 7
Purpose of this Manual 7
Additional Resources 7

Chapter 1

Hardware Overview

Hardware Features 1
Inputs and Outputs. 3
Embedded microSD (Micro Secure Digital) Card Slot 3
Embedded RS232/RS485 Serial Port Combo 3
Embedded Ethernet Support 4

Chapter 2

About Your Controller

Programming Software for Micro800 Controllers 7
Obtain Connected Components Workbench 7
Use Connected Components Workbench 7
Agency Certifications. 7
Compliance to European Union Directives. 7
EMC Directive. 8
Low Voltage Directive 8
Installation Considerations 8
Environment and Enclosure 10
Preventing Electrostatic Discharge 10
Safety Considerations 10
North American Hazardous Location Approval. 11
Disconnecting Main Power. 11
Safety Circuits 12
Power Distribution 12
Periodic Tests of Master Control Relay Circuit 12
Power Considerations 12
Isolation Transformers. 13
Power Supply Inrush. 13
Loss of Power Source 13
Input States on Power Down 14
Other Types of Line Conditions 14
Preventing Excessive Heat 14
Master Control Relay. 14
Using Emergency-Stop Switches 15
Schematic (Using IEC Symbols) 17
Schematic (Using ANSI/CSA Symbols) 18

Chapter 3

Install Your Controller

Controller Mounting Dimensions 19
Module Spacing 20
DIN Rail Mounting 20
Panel Mounting 20
Panel Mounting Dimensions 21

Connect the Controller to an EtherNet/IP Network 21
Install the microSD Card 22
Install the 2080-REMLCD Module. 23

Chapter 4

Wire Your Controller

Wiring Requirements and Recommendation 25
Wire Requirements 26
Use Surge Suppressors 26
Recommended Surge Suppressors 28
Grounding the Controller. 29
Wiring Diagrams 29
Controller I/O Wiring. 30
Minimize Electrical Noise 31
Analog Channel Wiring Guidelines 31
Minimize Electrical Noise on Analog Channels 31
Grounding Your Analog Cable 32
Wiring Examples 32
Wiring Analog Channels. 33

Chapter 5

Communication Connections

Overview 37
Supported Communication Protocols. 37
Modbus RTU 37
Modbus/TCP Client/Server. 38
CIP Symbolic Client/Server 38
CIP Client Messaging. 39
ASCII 40
CIP Communications Pass-thru 40
Examples of Supported Architectures. 40
Use Modems with Micro800 Controllers. 41
Making a DF1 Point-to-Point Connection. 41
Construct Your Own Modem Cable 41
Configure Serial Port 42
Configure CIP Serial Driver 43
Configure Modbus RTU 45
Configure ASCII 47
Configure Ethernet Settings. 48
Ethernet Host Name. 49

Chapter 6

Program Execution in Micro800

Overview of Program Execution 51
Execution Rules 52
Controller Load and Performance Considerations 52
Periodic Execution of Programs 53

Power Up and First Scan 53
Variable Retention 54
Memory Allocation 54
Guidelines and Limitations for Advanced Users 54

Chapter 7

Controller Security

Exclusive Access 57
Password Protection 57
Compatibility 57
Work with a Locked Controller 58
Upload from a Password-Protected Controller 58
Debug a Password-Protected Controller 59
Download to a Password-Protected Controller 59
Transfer Controller Program and Lock Receiving Controller 59
Back Up a Password-Protected Controller 60
Configure Controller Password 60
Recover from a Lost Password 61

Chapter 8

Using the Micro800 Remote LCD

Overview 63
USB Mode 64
Text Display Mode 65
Startup Screen 65
Navigate the Remote LCD 66
Main Menu 66
User-defined Screens 69
Backup and Restore 71
Hardware Features, Installation, and Specifications 71

Chapter 9

Using microSD Cards

Overview 73
Project Backup and Restore 73
Backup and Restore Directory Structure 75
Powerup Settings in ConfigMeFirst.txt 76
General Configuration Rules in ConfigMeFirst.txt 77
ConfigMeFirst.txt Errors 77
Datalog 78
Datalog Directory Structure 79
Datalog Function (DLG) Block 80
Recipe 83
Recipe Directory Structure 84
Recipe Function (RCP) Block 84
Quickstart Projects for Datalog and Recipe Function Blocks 87
Use the Datalog Feature 88

Use the Recipe Feature 94

Appendix A
Specifications

. . . . 101

Appendix B
Troubleshooting

Status Indicators on the Controller 111
Normal Operation 112
Error Conditions 112
Error codes 113
Controller Error Recovery Model 119
Calling Rockwell Automation for Assistance 120

Appendix C
Quickstarts

Flash Upgrade Your Micro800 Firmware 121
Establish Communications between RSLinx and a Micro820 Controller through USB Port on 2080-REMLCD 124
Configure Controller Password 128
Set Controller Password 129
Change Password 130
Clear Password 131
Forcing I/Os 132
Checking if Forces (locks) are Enabled 132
I/O Forces After a Power Cycle 133

Appendix D
IPID Function Block

How to Autotune 137
How Autotune Works 138
Troubleshooting an Autotune Process 139
PID Application Example 140
PID Code Sample 141

Appendix F
Modbus Mapping for Micro800

Modbus Mapping 143
Endian Configuration 143
Mapping Address Space and supported Data Types 143
Example 1, PanelView Component HMI (Master) to Micro800 (Slave) 144
Example 2, Micro800 (Master) to PowerFlex 4M Drive (Slave) . . 145
Performance 148

Index

Chapter 1

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	3
Embedded Ethernet Support	4

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 datalog and recipe.

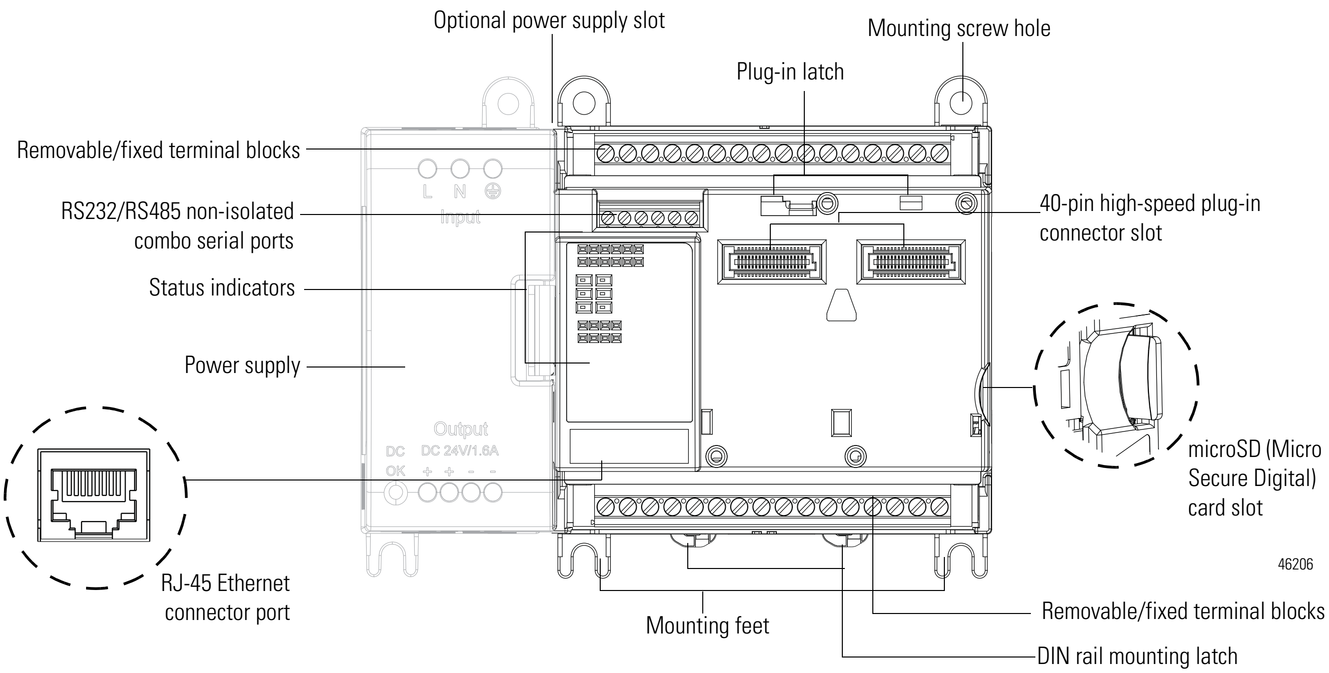
IMPORTANT The Micro820 controller supports all Micro800 plug-in modules, except for the 2080-MEMBAK-RTC.

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 63.

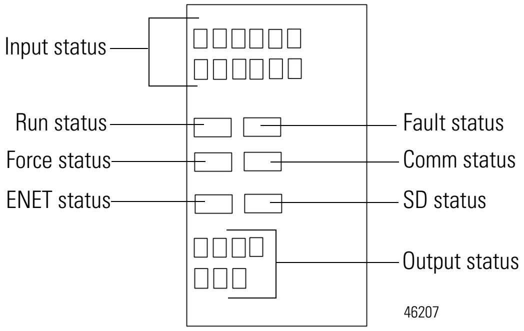
The controller also accommodates any class 2 rated 24V DC output power supply that meets minimum specifications such as the optional Micro800 power supply.

Micro820 Controllers


Optional power supply slot
Mounting screw hole
Plug-in latch
Removable/fixed terminal blocks
RS232/RS485 non-isolated combo serial ports
40-pin high-speed plug-in connector slot
Status indicators
Power supply
L N
Input
Output
DC OK
DC 24V/1.6A
microSD (Micro Secure Digital) card slot
46206
RJ-45 Ethernet connector port
Mounting feet
Removable/fixed terminal blocks
DIN rail mounting latch

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 111 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. It supports Class 6 and 10 SDSC and SDHC microSD cards, with FAT32/16 formats, 32 GB maximum size. Industrial grade cards such as Swissbit S-200u/S300u are recommended. 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 datalog 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 73.

To help you troubleshoot microSD card-related errors, see Troubleshooting on page 111.

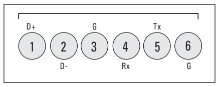
Embedded RS232/RS485 Serial Port Combo

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

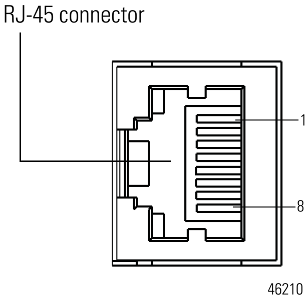
REMLCD Serial Port Terminal Block			Micro820 Serial Port Terminal Block	
Signal	Pin number		Pin number	Signal
RS232 TX	1	<-------->	4	RX RS232
RS232 RX	2	<-------->	5	TX RS232
RS232 G	3	<-------->	6	G RS232

Embedded Ethernet Support

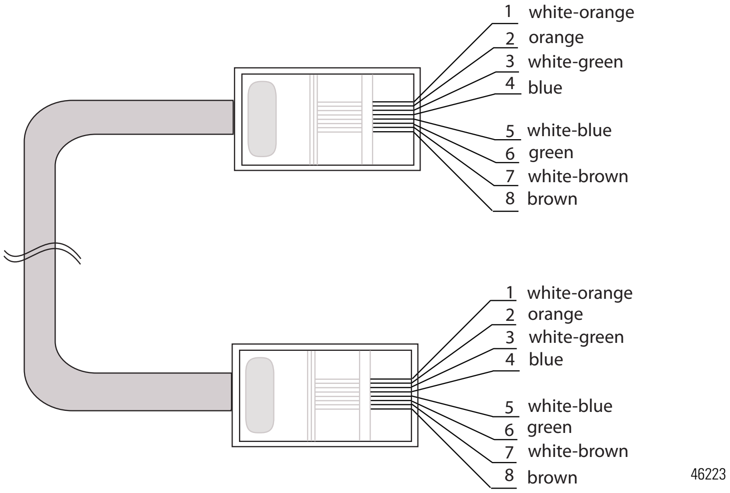
A 10/100 Base-T Port is available for connection to an Ethernet network through any standard RJ-45 Ethernet cable.

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	–	–	–



Ethernet port pin-to-pin connection



See Troubleshooting on page 111 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://ab.rockwellautomation.com/Programmable-Controllers/Connected-Components-Workbench-Software

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).

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/ECLow 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.

[1] Pollution Degree 2 is an environment where normally only non-conductive pollution occurs except that occassionally 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.

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.

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:**
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.	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.
⚠ **EXPLOSION HAZARD** • 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.	⚠ **RISQUE D'EXPLOSION** • 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 17 and 18.

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:

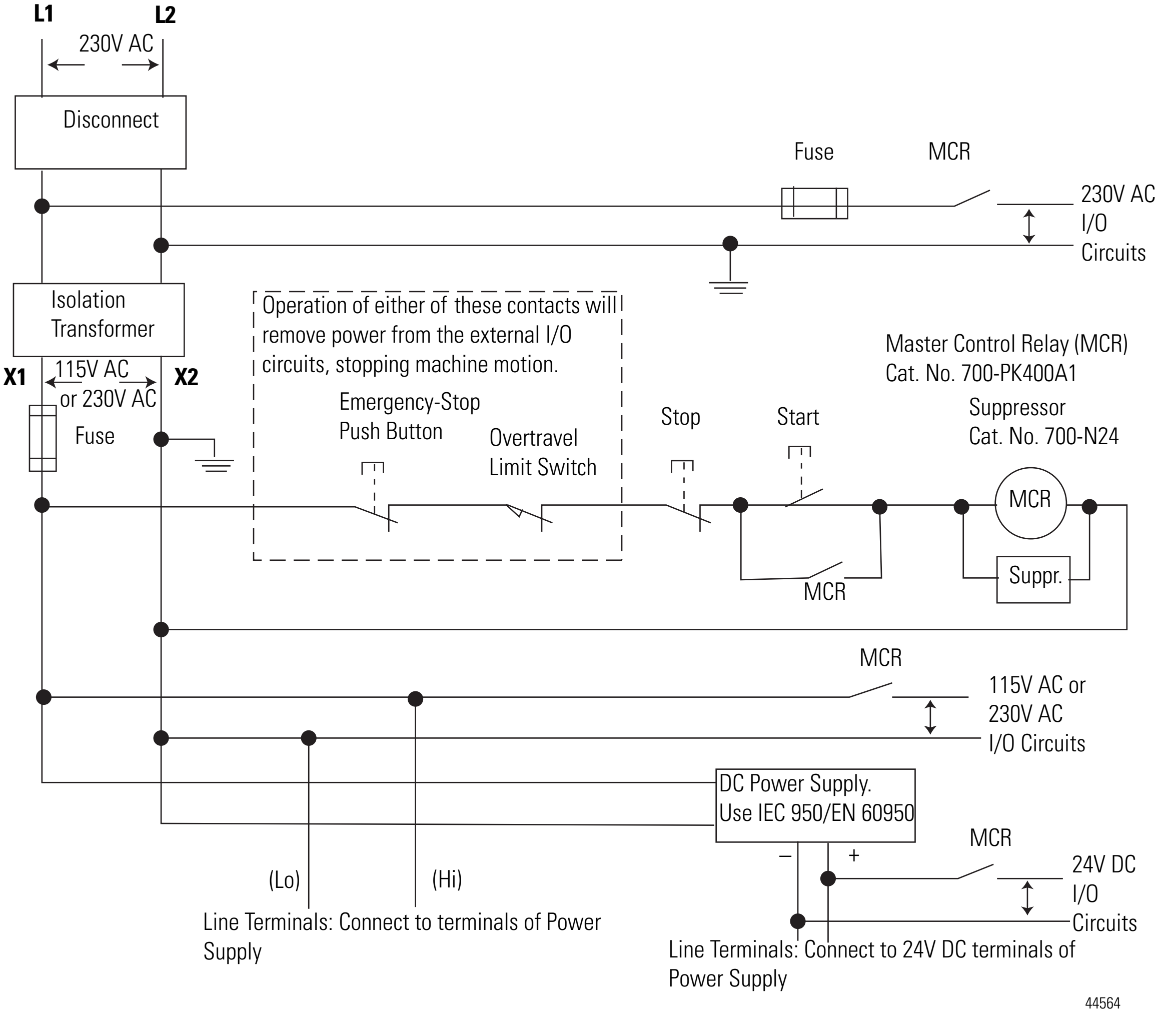
- Do not program emergency-stop switches in the controller program. Any emergency-stop switch should turn off all machine power by turning off the master control relay.
- Observe all applicable local codes concerning the placement and labeling of emergency-stop switches.

- 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.

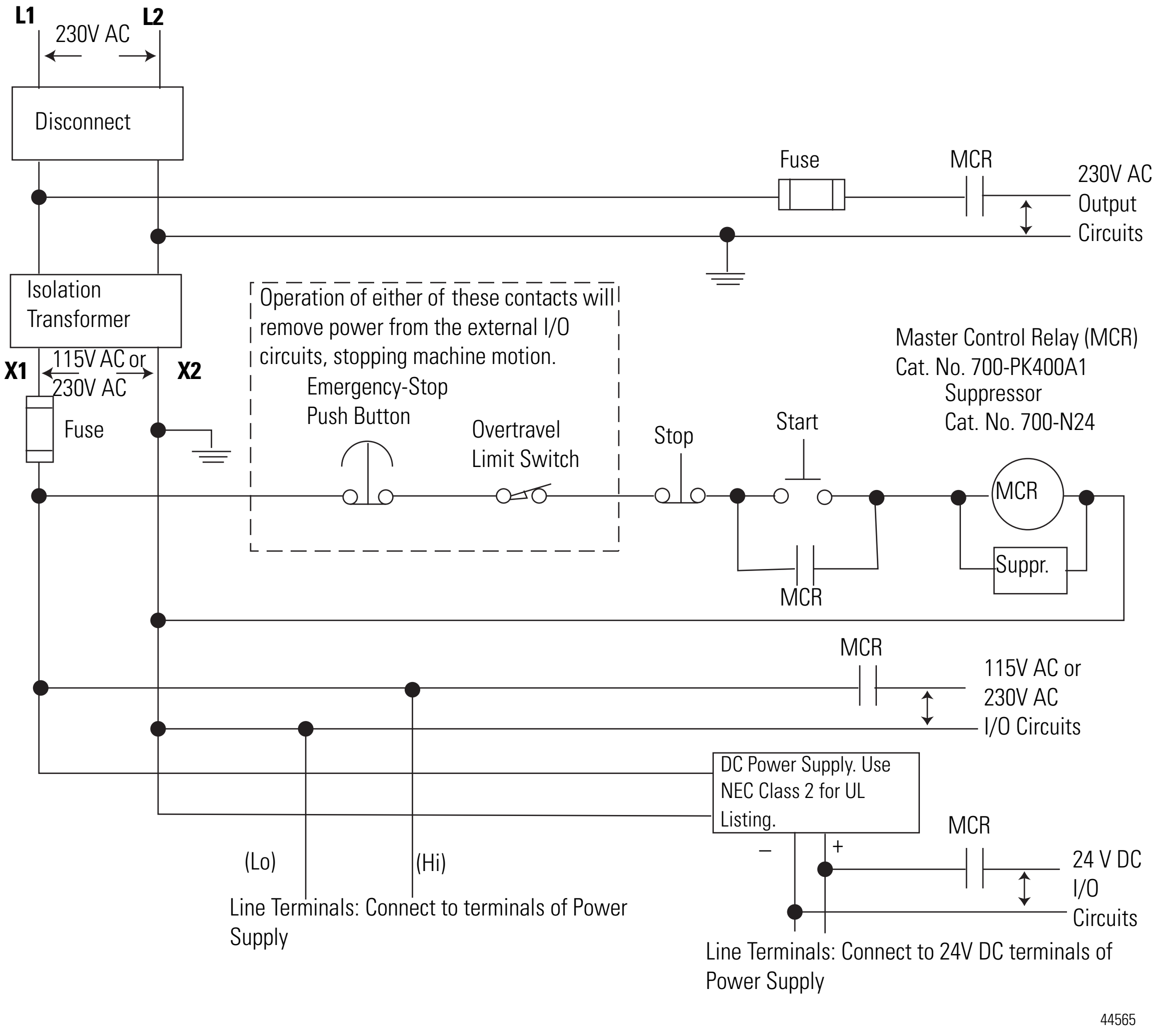
The following illustrations show the Master Control Relay wired in a grounded system.

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)



Schematic (Using ANSI/CSA Symbols)


L1
L2
230V AC
Disconnect
Fuse
MCR
230V AC
Output
Circuits
Isolation
Transformer
Operation of either of these contacts will remove power from the external I/O circuits, stopping machine motion.
Master Control Relay (MCR)
Cat. No. 700-PK400A1
Suppressor
Cat. No. 700-N24
X1
115V AC or
230V AC
X2
Emergency-Stop
Push Button
Fuse
Overtravel
Limit Switch
Stop
Start
MCR
Suppr.
MCR
MCR
115V AC or
230V AC
I/O Circuits
DC Power Supply. Use NEC Class 2 for UL Listing.
MCR
–
+
(Lo)
(Hi)
24 V DC
I/O
Circuits
Line Terminals: Connect to terminals of Power Supply
Line Terminals: Connect to 24V DC terminals of Power Supply
44565

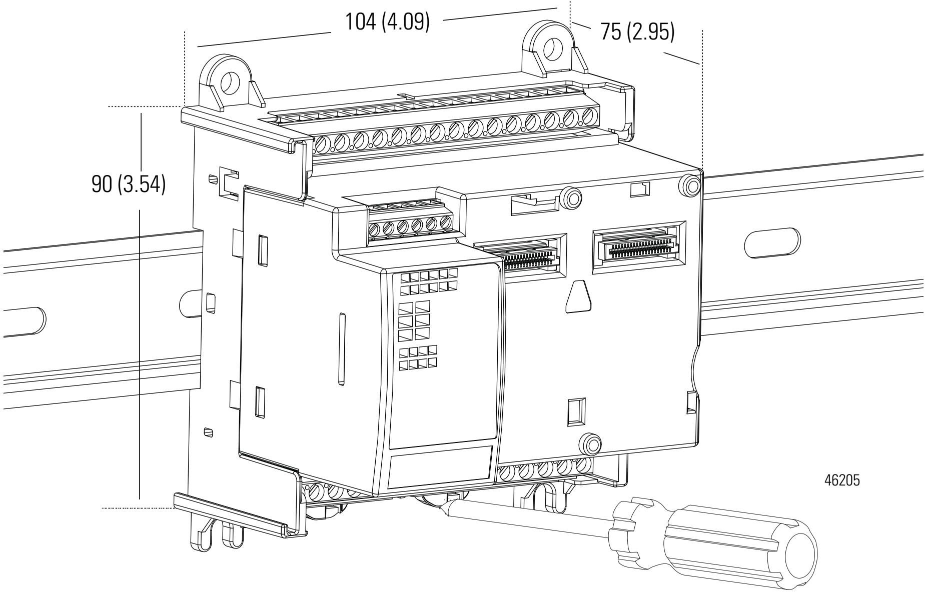
Install Your Controller

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

Topic	Page
Controller Mounting Dimensions	19
Connect the Controller to an EtherNet/IP Network	21
Module Spacing	20
DIN Rail Mounting	20
Panel Mounting	20
Panel Mounting Dimensions	21
Install the microSD Card	22
Install the 2080-REMLCD Module	23

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

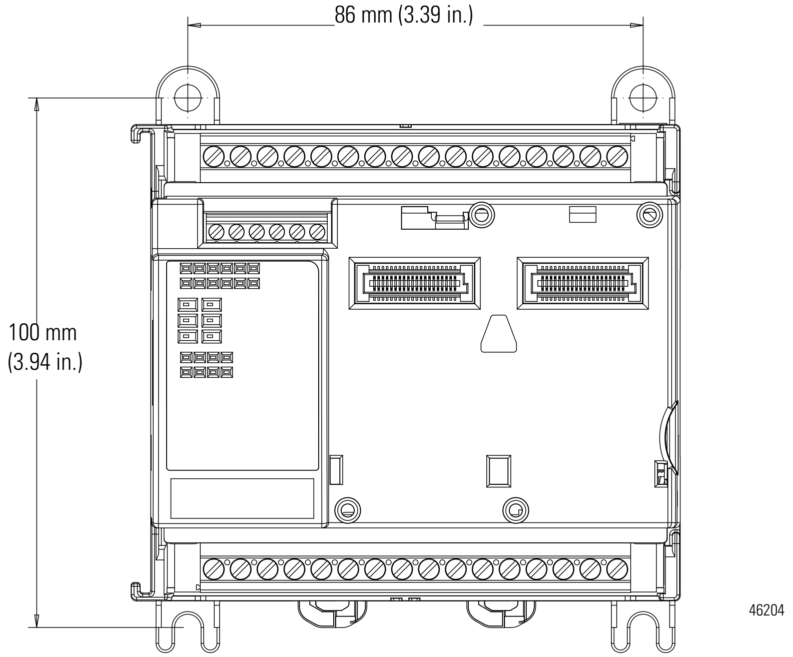
The preferred mounting method is to use four M4 (#8) screws per module. Hole spacing tolerance: ±0.4 mm (0.016 in.).

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

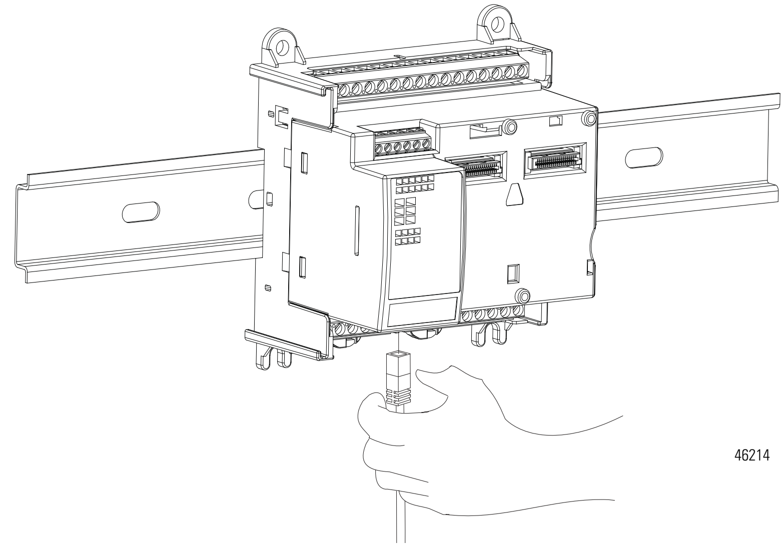
Micro820 20-point controllers
2080-LC20-20AWB, 2080-LC20-20QWB, 2080-LC20-20QBB
2080-LC20-20AWBR, 2080-LC20-20QWBR, 2080-LC20-20QBBR



Connect the Controller to an EtherNet/IP Network

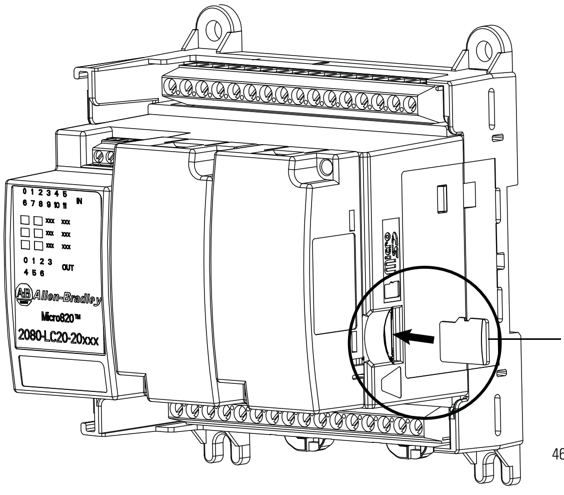
WARNING: If you connect or disconnect the communications cable with power applied to this module or any device on the network, an electrical arc can occur. This could cause an explosion in hazardous location installations.
Be sure that power is removed or the area is nonhazardous before proceeding.

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.



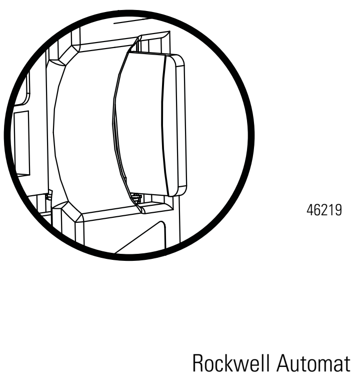
Install the microSD Card

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.



Insert the microSD card into the slot.

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



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 63.

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	25
Use Surge Suppressors	26
Recommended Surge Suppressors	28
Grounding the Controller	29
Wiring Diagrams	29
Controller I/O Wiring	30
Minimize Electrical Noise	31
Analog Channel Wiring Guidelines	31
Minimize Electrical Noise on Analog Channels	31
Grounding Your Analog Cable	32
Wiring Examples	32

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

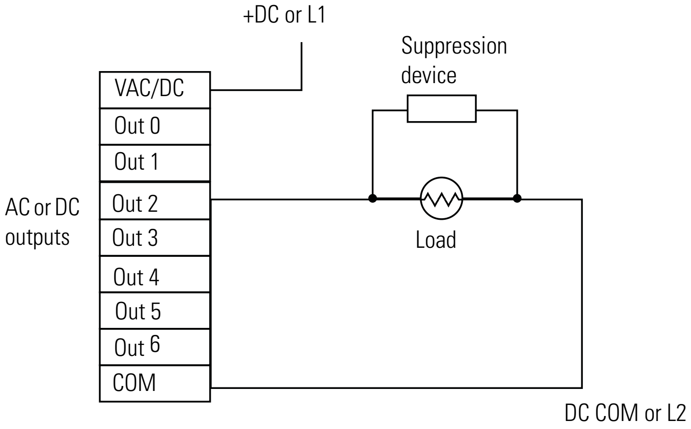
Wire requirements for RS232/RS485 serial port terminal block

	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)	

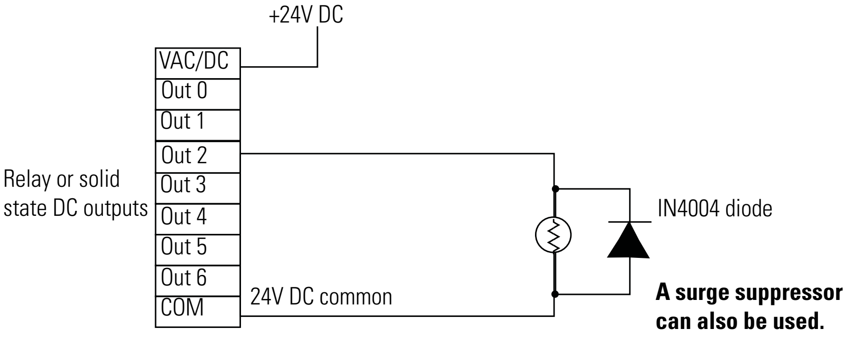
Use Surge Suppressors

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.

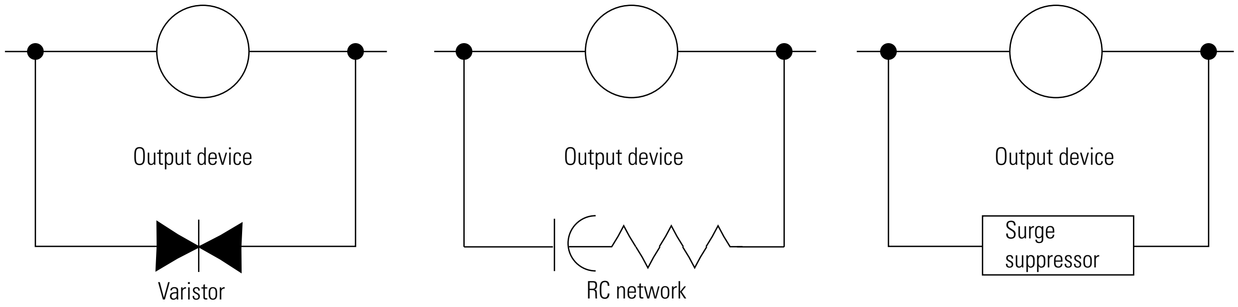


If the outputs are DC, we recommend that you use an 1N4004 diode for surge suppression, as shown below. For inductive DC load devices, a diode is suitable. A 1N4004 diode is acceptable for most applications. A surge suppressor can also be used. See Recommended Surge Suppressors on page 28. As shown below, 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 28 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 limted 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.

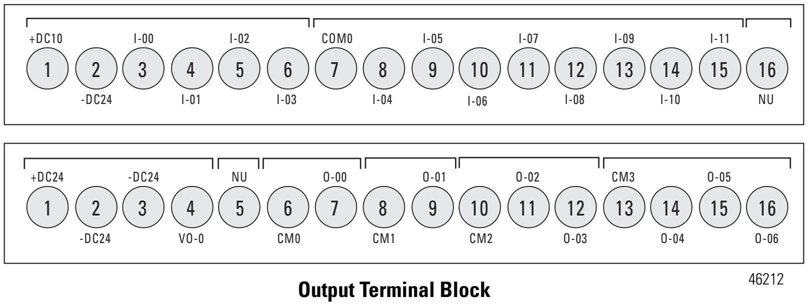
Wiring Diagrams

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

Input Terminal Block

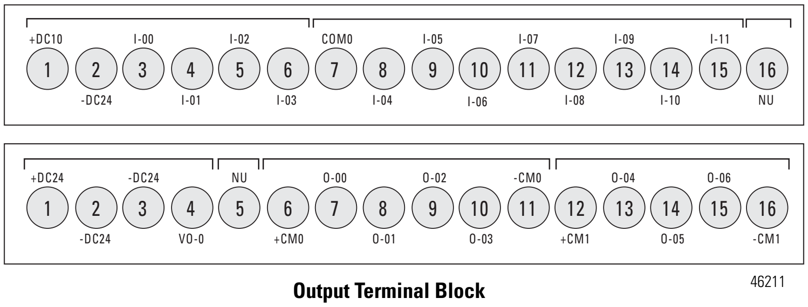


Output Terminal Block

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.

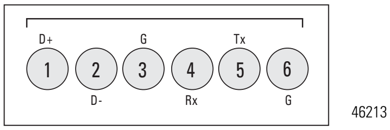
2080-LC20-20QBB / 2080-LC20-20QBBR

Input Terminal Block



Output Terminal Block

Serial Port Terminal Block



(View into terminal block)

Pin 1	RS485	Data +
Pin 2	RS485	Data -
Pin3	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

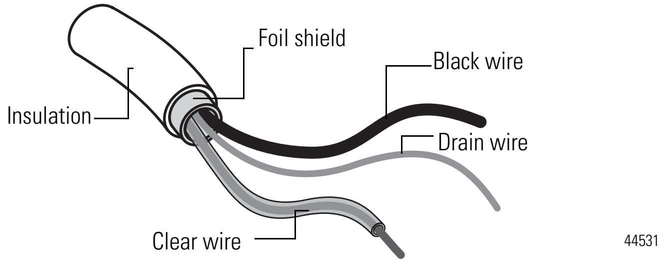
Inputs on analog channels employ digital high-frequency filters that significantly reduce the effects of electrical noise on input signals. However, because of the variety of applications and environments where analog controllers are installed and operated, it is impossible to ensure that all environmental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

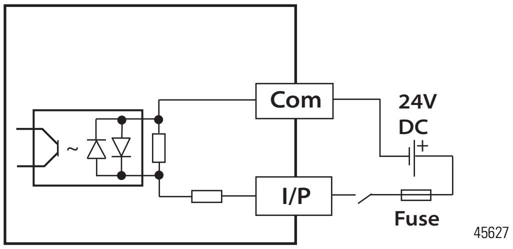


IMPORTANT Do not ground the drain wire and foil shield at both ends of the cable.

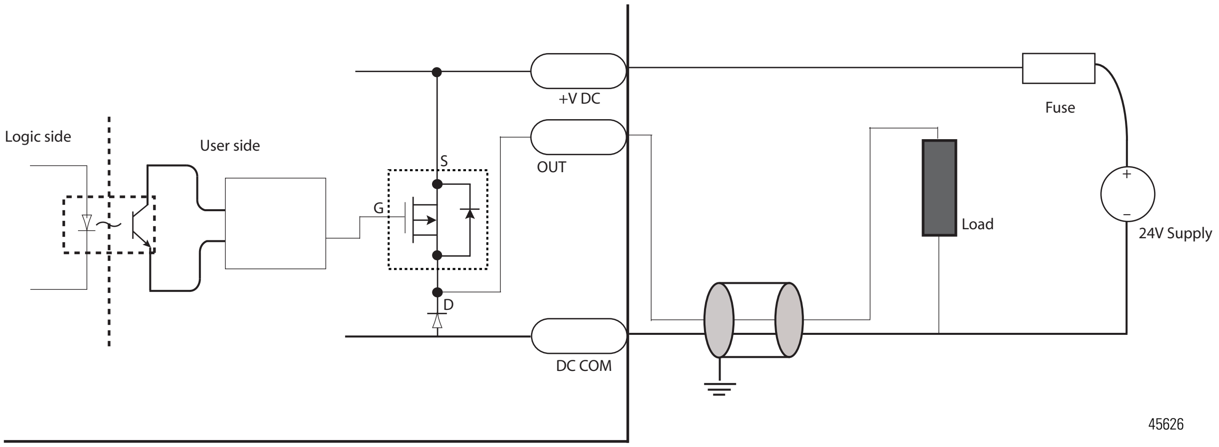
Wiring Examples

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

Sink Input Wiring Example

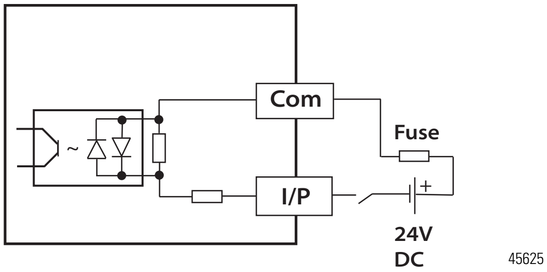


Source Output Wiring Example



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

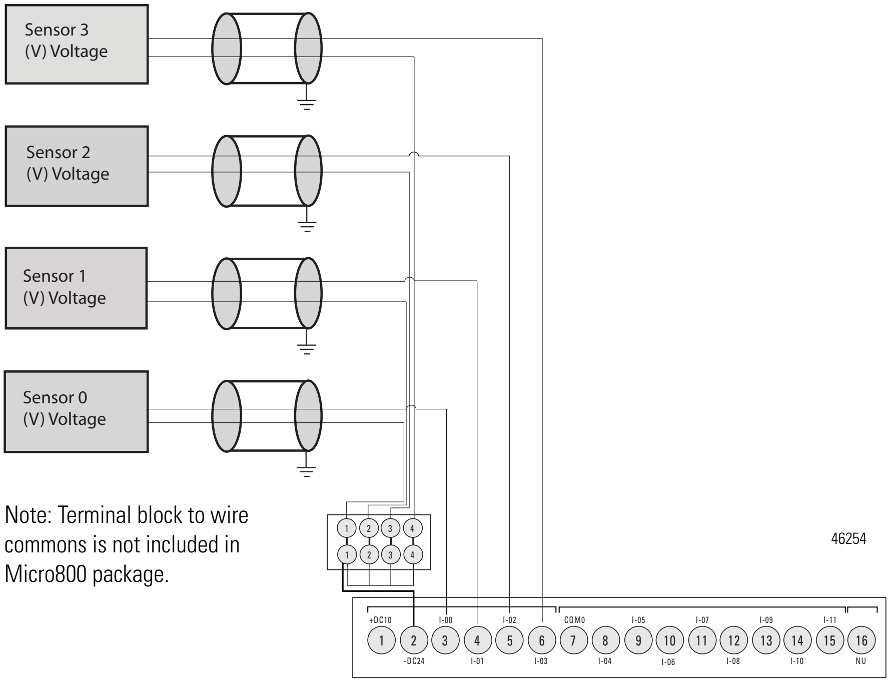


Wiring Analog Channels

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

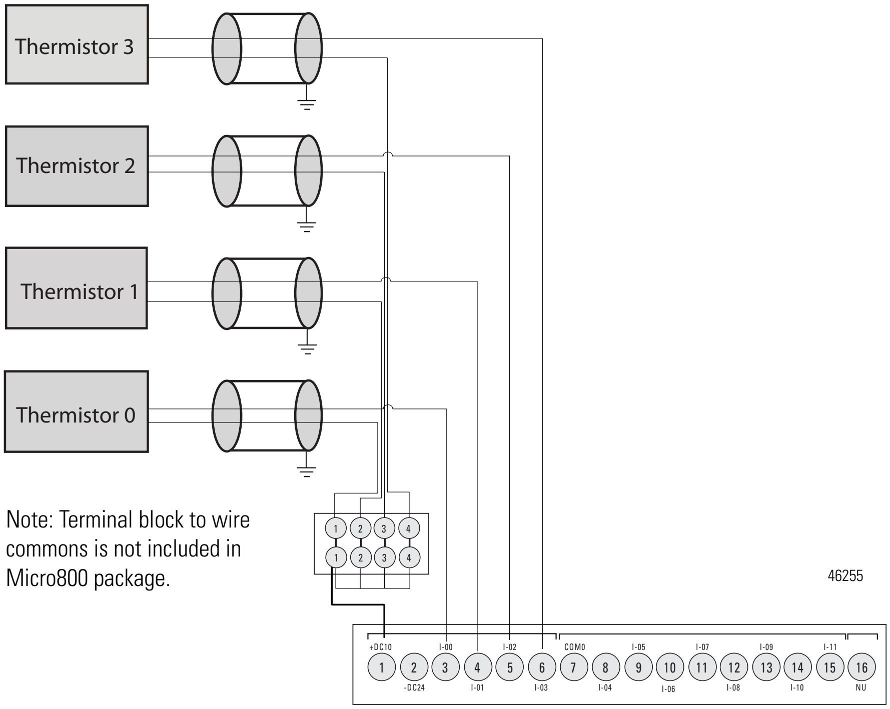
ATTENTION: Analog inputs and outputs are not isolated.

Analog input to sensors



Note: Terminal block to wire commons is not included in Micro800 package.

Analog input to thermistors



Note: Terminal block to wire commons is not included in Micro800 package.

Calculate for Thermistor Resistance

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

$$Vi = \frac{Ri}{Ri + Rt} * Vref$$

Where:
Vi = Voltage input (±5% without calibration; ±2% with calibration)
Ri = Resistance input (14.14 KΩ ±2%)
Rt = Thermistor resistance (10 KΩ Thermistor is recommended)
Vref = 10V ±0.5V

To calculate for thermistor resistance, use the following equation.

$$Rt= \frac{Vi\ Vref - Vi\ Ri}{Vi}$$

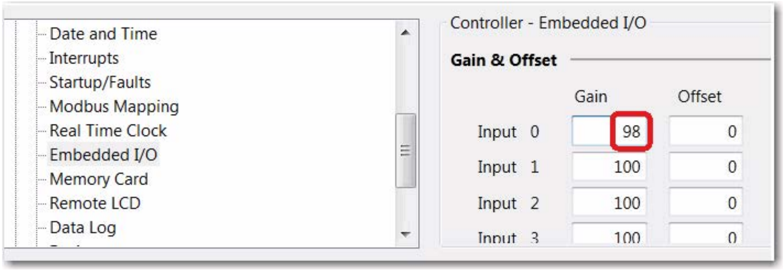
IMPORTANT	Micro820 controllers support 10 KΩ type thermistors. In order to get the best results, the system must be calibrated.

Calibrate Thermistor

1. Connect a resistor (10 KΩ is recommended) across Vref and Analog Input 00 of your Micro820 controller following the diagram, Analog input to thermistors on page 34. 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 KΩ / (14.14 KΩ + Ri) * 4095
3. Read the actual counts (C2) of Analog Input 00 from Connected Components Workbench.
4. Calculate for calibration Gain.
   Gain = C1/C2

   For example:
   If Ri is measured as 10.00 KΩ, then
   C1 = 14.14 / (14.14 + 10.00) * 4095 = 2399 counts;
   C2 is read from Connected Components Workbench as 2440; so
   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.



Controller - Embedded I/O

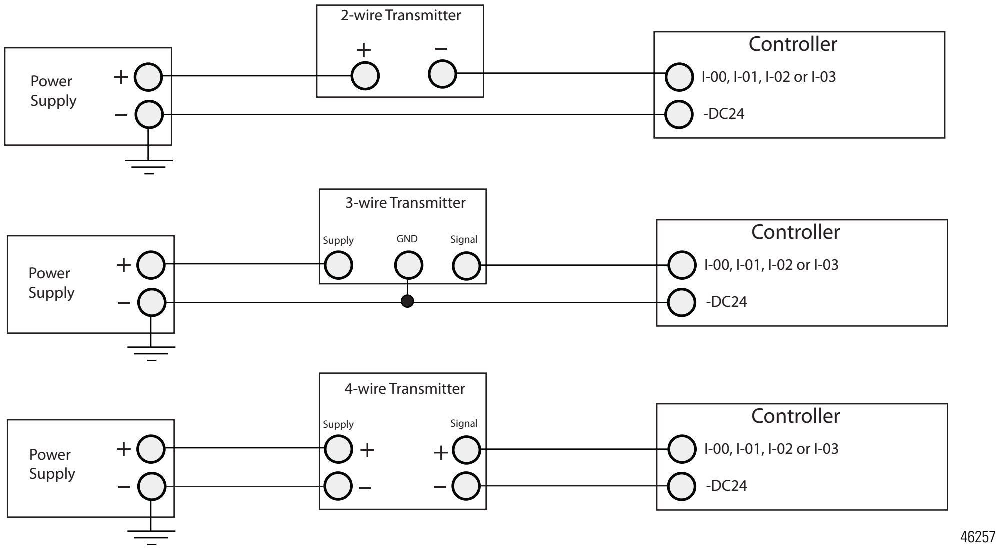
Gain & Offset

	Gain	Offset
Input 0	98	0
Input 1	100	0
Input 2	100	0
Input 3	100	0

No changes are required to the Offset parameter value.

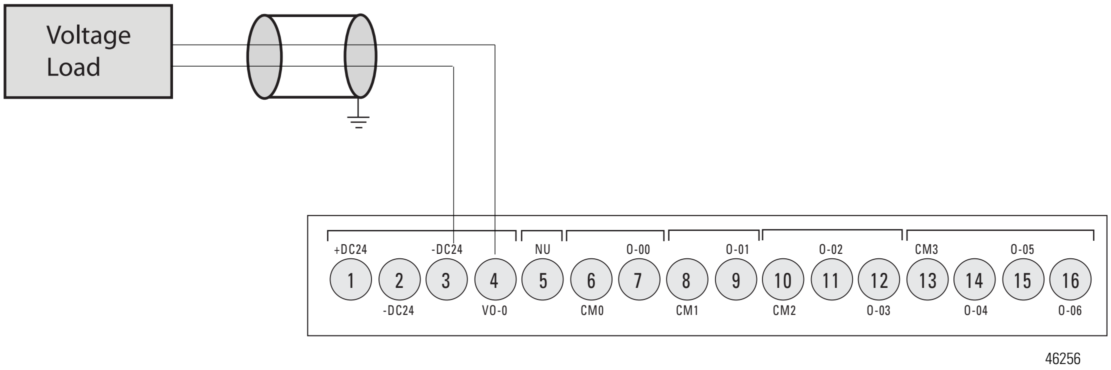
6. Repeat the same steps to calibrate all the other analog input channels.

Analog Input to Transmitters



Analog Output

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



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	37
Use Modems with Micro800 Controllers	41
Configure Serial Port	42
Configure Ethernet Settings	48

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 the following communication protocols through the embedded RS232/RS485 serial port as well as any installed serial port plug-in modules:

- Modbus RTU Master and Slave
- CIP Serial Client/Server (RS232 only)
- ASCII

In addition, the embedded Ethernet communication channel allows your Micro820 controller to be connected to a local area network for various devices providing 10 Mbps/100 Mbps transfer rate. Micro820 controllers support the following Ethernet protocols:

- EtherNet/IP Client/Server
- Modbus/TCP Client/Server
- DHCP Client

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 249 for information on Modbus mapping. To configure the Serial port as Modbus RTU, see Configure Modbus RTU on page 45.

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

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 16 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 249.

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), 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 16 simultaneous EtherNet/IP Server connections.

To configure CIP Serial, see Configure CIP Serial Driver on page 43.

To configure for EtherNet/IP, see Configure Ethernet Settings on page 48.

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 and FALSE
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)

[1] Logix MSG instruction can read/write SINT, INT, DINT, LINT and REAL datatypes using "CIP Data Table Read" and "CIP Data Table Write" message types.
BOOL, USINT, UINT, UDINT, ULINT, LREAL, STRING and SHORT_STRING datatypes are not accessible with the Logix MSG instruction.

[2] Not supported in PanelView Component.

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.

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.

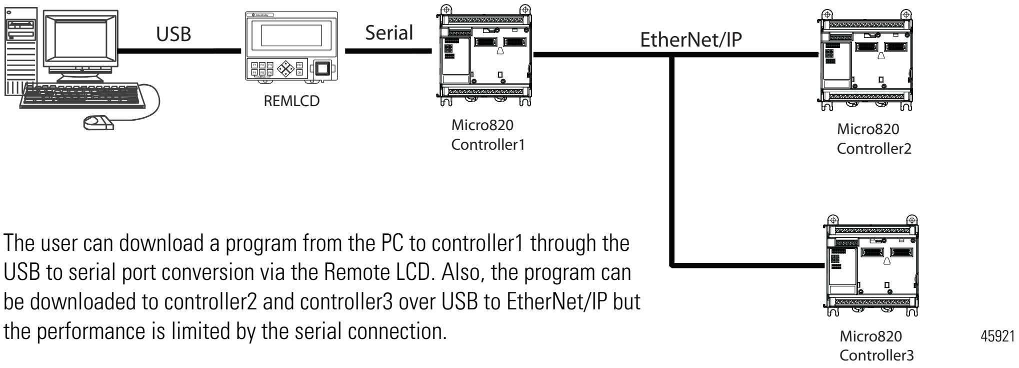
To configure the serial port for ASCII, see Configure ASCII on page 47.

CIP Communications Pass-thru

The Micro820 controllers support pass-thru on any communications port that supports Common Industrial Protocol (CIP). 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.

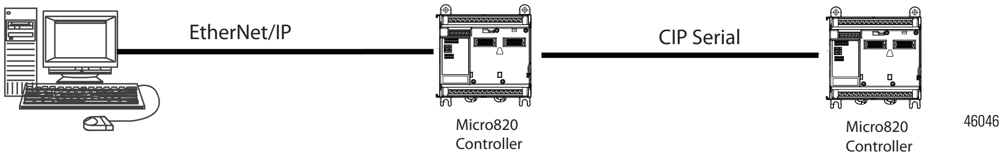
Examples of Supported Architectures

CIP Serial to EtherNet/IP



The user can download a program from the PC to controller1 through the USB to serial port conversion via the Remote LCD. Also, the program can be downloaded to controller2 and controller3 over USB to EtherNet/IP but the performance is limited by the serial connection.

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

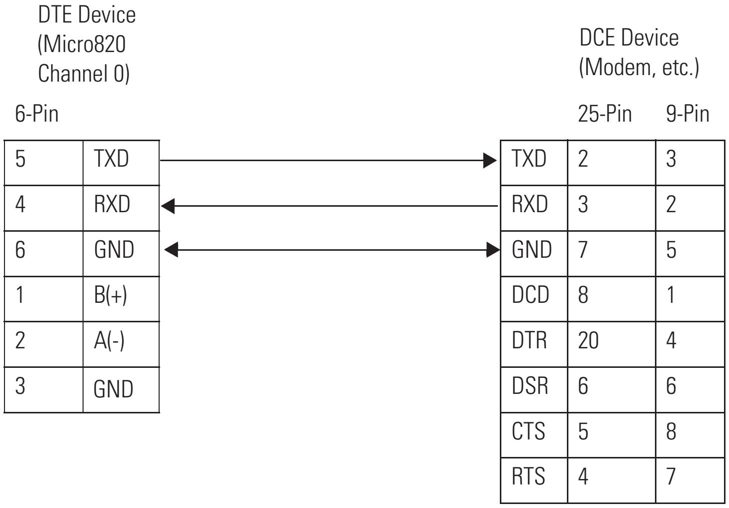
Serial modems can be used with the Micro820 controllers.

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)

6-Pin	
5	TXD
4	RXD
6	GND
1	B(+)
2	A(-)
3	GND

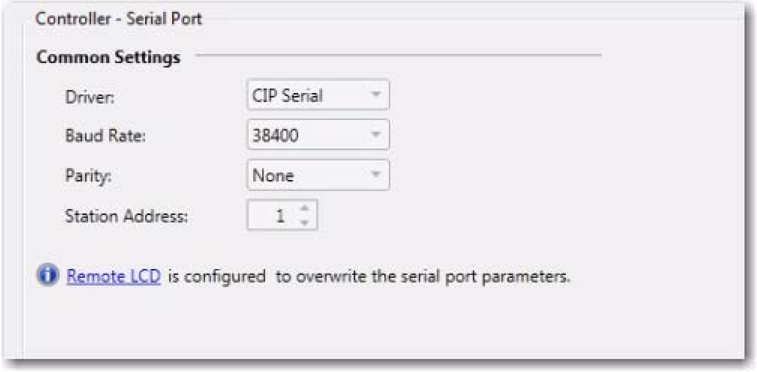
DCE Device
(Modem, etc.)

	25-Pin	9-Pin
TXD	2	3
RXD	3	2
GND	7	5
DCD	8	1
DTR	20	4
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.



Controller - Serial Port

Common Settings

Driver: CIP Serial

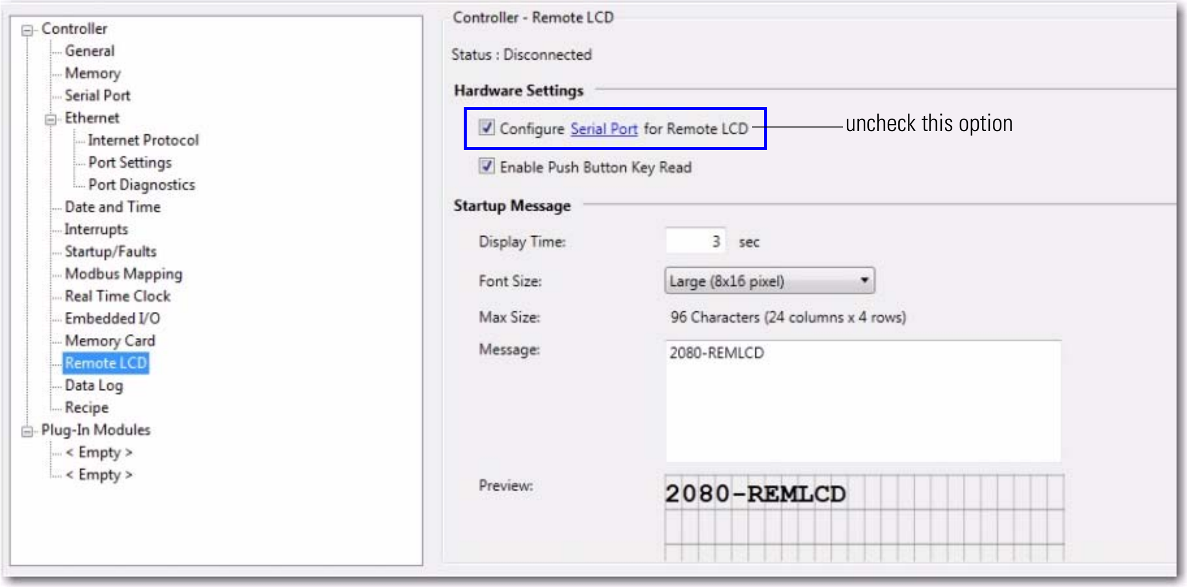
Baud Rate: 38400

Parity: None

Station Address: 1

Remote LCD is configured to overwrite the serial port parameters.

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



Controller - Remote LCD

Status : Disconnected

Hardware Settings

☑ Configure Serial Port for Remote LCD — uncheck this option

☑ Enable Push Button Key Read

Startup Message

Display Time: 3 sec

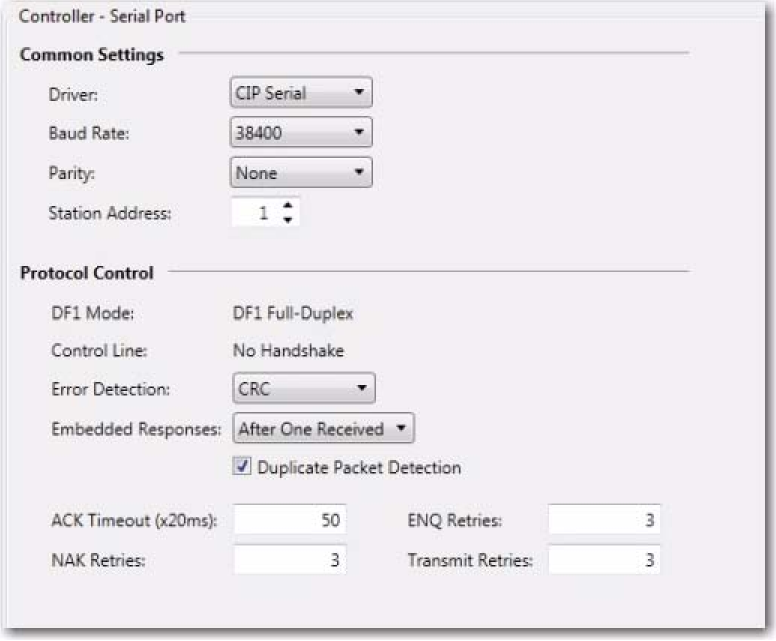
Font Size: Large (8x16 pixel)

Max Size: 96 Characters (24 columns x 4 rows)

Message: 2080-REMLCD

Preview: 2080-REMLCD

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



Controller - Serial Port

Common Settings

Driver: CIP Serial

Baud Rate: 38400

Parity: None

Station Address: 1

Protocol Control

DF1 Mode: DF1 Full-Duplex

Control Line: No Handshake

Error Detection: CRC

Embedded Responses: After One Received

☑ Duplicate Packet Detection

ACK Timeout (x20ms): 50

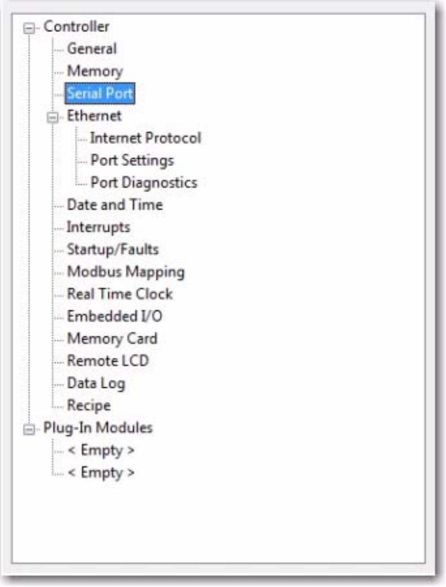
ENQ Retries: 3

NAK Retries: 3

Transmit Retries: 3

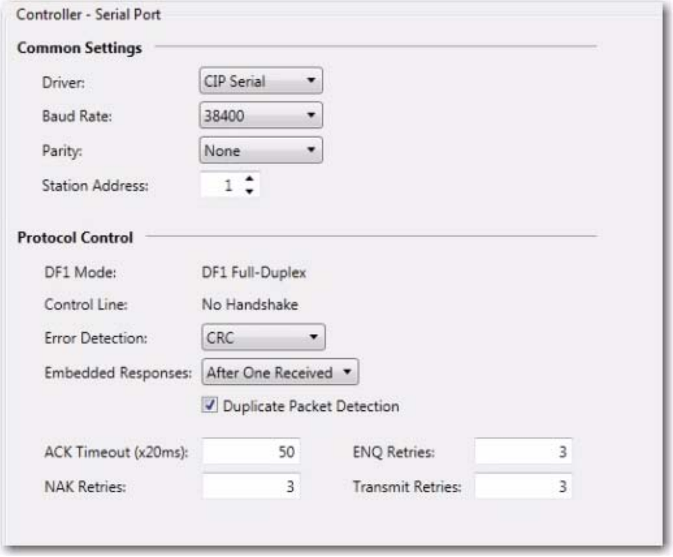
Configure CIP Serial Driver

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



```
Controller
    General
    Memory
    Serial Port
    Ethernet
        Internet Protocol
        Port Settings
        Port Diagnostics
    Date and Time
    Interrupts
    Startup/Faults
    Modbus Mapping
    Real Time Clock
    Embedded I/O
    Memory Card
    Remote LCD
    Data Log
    Recipe
Plug-In Modules
    < Empty >
    < Empty >
```

2. Select CIP Serial from the Driver field.



Controller - Serial Port

Common Settings

Driver: CIP Serial

Baud Rate: 38400

Parity: None

Station Address: 1

Protocol Control

DF1 Mode: DF1 Full-Duplex

Control Line: No Handshake

Error Detection: CRC

Embedded Responses: After One Received

☑ Duplicate Packet Detection

ACK Timeout (x20ms): 50

ENQ Retries: 3

NAK Retries: 3

Transmit Retries: 3

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 44 for a description of the CIP Serial parameters.

CIP Serial Driver Parameters

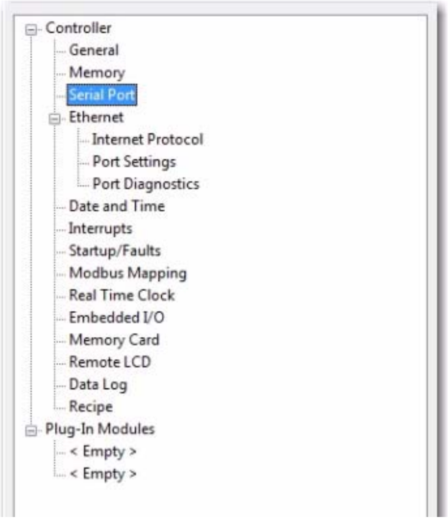
Parameter	Options	Default
Baud rate	Toggles between the communication rate of 1200, 2400, 4800, 9600, 19200, and 38400.	38400
Parity	Specifies the parity setting for the serial port. Parity provides additional message-packet error detection. Select Even, Odd, or None.	None
Station Address	The station 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

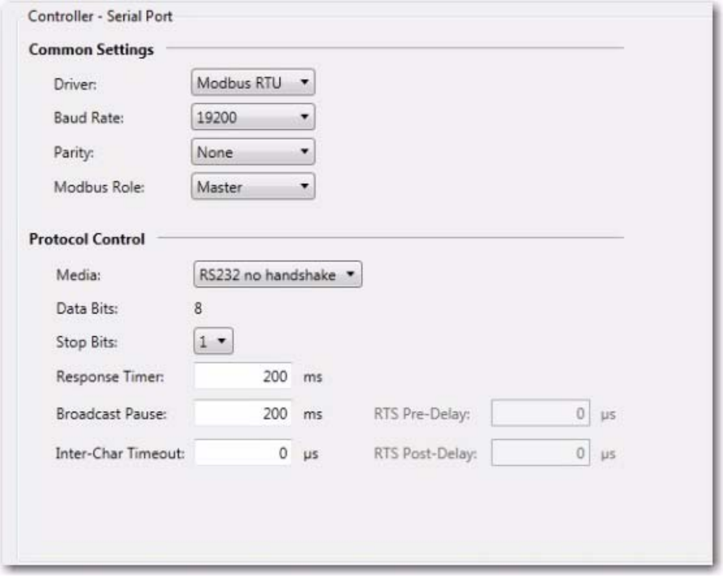
Embedded Responses	To use embedded responses, choose Enabled Unconditionally. If you want the controller to use embedded responses only when it detects embedded responses from another device, choose After One Received. If you are communicating with another Allen-Bradley device, choose Enabled Unconditionally. 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

RTS Pre-Delay: 0 µs

Inter-Char Timeout: 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

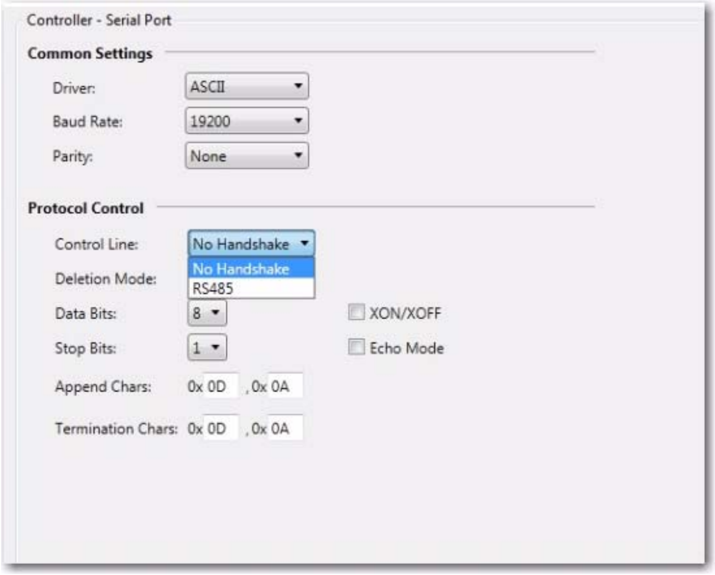
4. Click Advanced Settings to set advanced parameters.
   Refer to the table for available options and default configuration for advanced parameters.

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.



Controller - Serial Port

Common Settings

Driver: ASCII

Baud Rate: 19200

Parity: None

Protocol Control

Control Line: No Handshake

Deletion Mode:

Data Bits: 8

Stop Bits: 1

□ XON/XOFF

□ Echo Mode

Append Chars: 0x 0D , 0x 0A

Termination Chars: 0x 0D , 0x 0A

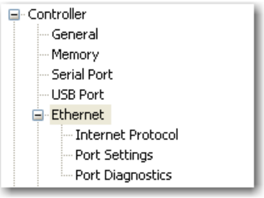
3. Specify baud rate and parity.

ASCII Parameters

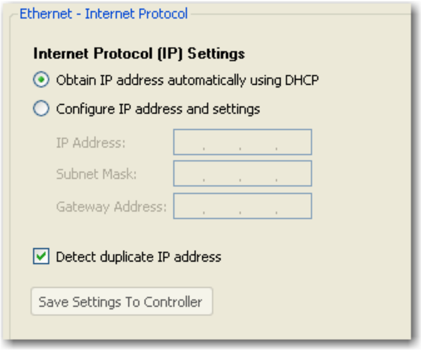
Parameter	Options	Default
Baud Rate	1200, 2400, 4800, 9600, 19200, 38400	19200
Parity	None, Odd, Even	None
Control Line	RS485 No Handshake	No Handshake
Deletion Mode	CRT Ignore Printer	Ignore
Data Bits	7, 8	8
XON/XOFF	Enabled or Disabled	Disabled
Stop Bits	1, 2	1
Echo Mode	Enabled or Disabled	Disabled
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.



Ethernet - Internet Protocol

Internet Protocol (IP) Settings

● Obtain IP address automatically using DHCP

○ Configure IP address and settings

IP Address: . . .

Subnet Mask: . . .

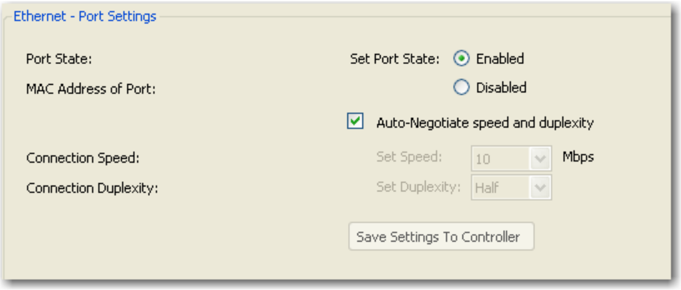
Gateway Address: . . .

☑ Detect duplicate IP address

TIP The Ethernet port defaults to the following out-of-the box settings:
- DHCP (dynamic IP address)
- Address Duplicate Detection: On

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

4. Under Ethernet, click Port Settings.



Ethernet - Port Settings

Port State:

MAC Address of Port:

Set Port State: ● Enabled ○ Disabled

☑ Auto-Negotiate speed and duplexity

Connection Speed:

Connection Duplexity:

Set Speed: 10 Mbps

Set Duplexity: Half

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.

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.

Notes:

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).

Overview of Program Execution

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

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

Up to 256 programs 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 Block (UDFB). 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.

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).

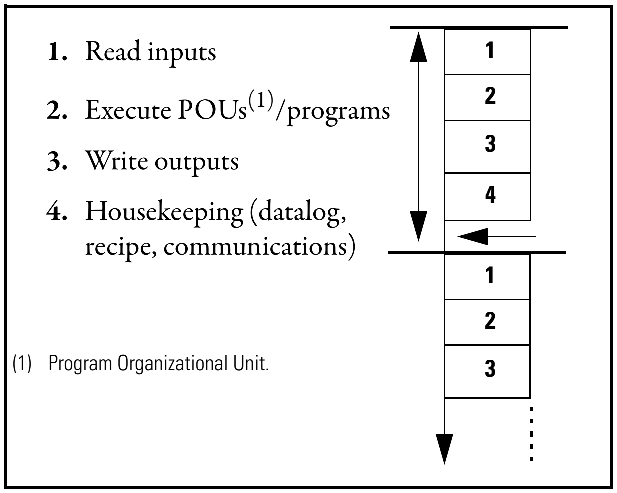
The Global System Variables associated with cycles/scans are:

- __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.



1. Read inputs
2. Execute POUs[1]/programs
3. Write outputs
4. Housekeeping (datalog, recipe, communications)

[1] Program Organizational Unit.

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

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

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

On firmware revision 2 and later, all digital output variables driven by the I/O scan gets cleared on powerup and during transition to RUN mode.

Two system variables are also available from revision 2 and later.

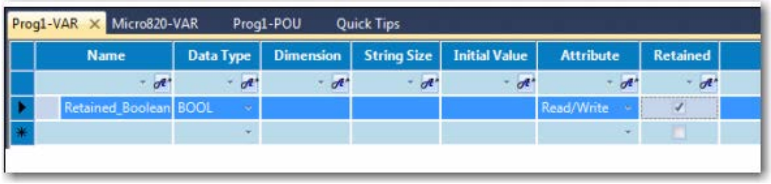
System Variables for Scan and Powerup on Firmware Release 2 and later

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	✓
*							□

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.

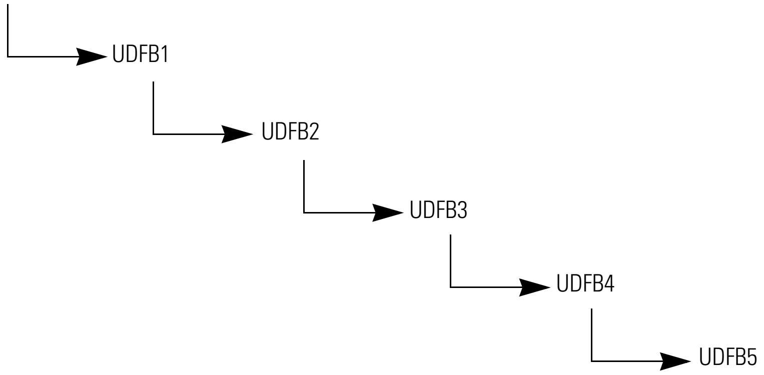
The Micro800 controllers also have project memory, which stores a copy of the entire downloaded project (including comments), as well as configuration memory for storing plug-in setup information, and so on.

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 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.

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, ST combined with UDFB is a great alternative.
  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.

Notes:

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).

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).

TIP For instructions on how to set, change, and clear controller passwords, see Configure Controller Password on page 128.

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 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 121 for instructions.

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. On the Device Toolbox, expand Catalog by clicking the + sign.
3. Select the target controller.
4. Select Upload.
5. When requested, provide the controller password.

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. On the Device Toolbox, 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.

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. On the Device Toolbox, open Discover and click Browse Connections.
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 128 .

Back Up 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. On the Device Toolbox, open Discover. Click Browse Connections.
2. Select the target controller.
3. When requested, enter the controller password.
4. Back up controller contents from the memory module.

Configure Controller Password

To set, change, and clear controller password, see the quickstart instructions Configure Controller Password on page 128.

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 the Micro820. Then, ControlFlash can be used to update the controller firmware, which also clears the controller memory.

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

Notes:

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	63
Text Display Mode	65
USB Mode	64
Backup and Restore	71
Hardware Features, Installation, and Specifications	71

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:

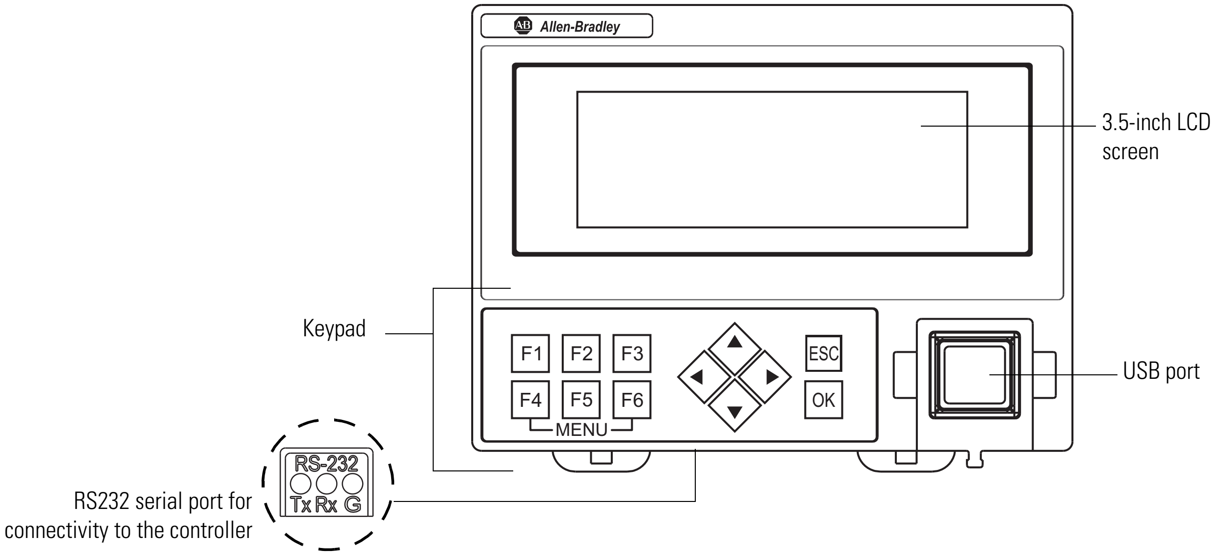
- Four arrow keys
- Six function keys
- ESC key
- OK key
- USB port for Connected Components Workbench connectivity

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

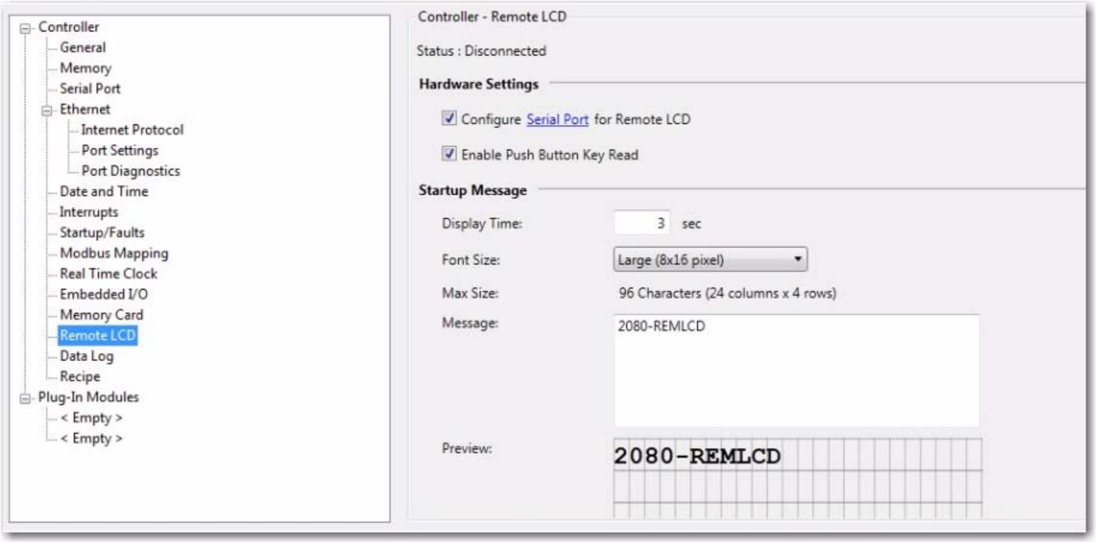
In text display mode, you are either in I/O Status, Main Menu, or executing Remote LCD instructions.

Startup Screen

Default 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.



Controller - Remote LCD

Status : Disconnected

Hardware Settings

☑ Configure Serial Port for Remote LCD

☑ Enable Push Button Key Read

Startup Message

Display Time: 3 sec

Font Size: Large (8x16 pixel)

Max Size: 96 Characters (24 columns x 4 rows)

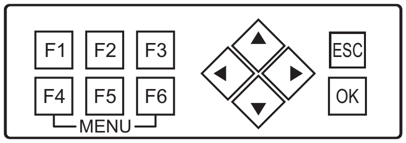
Message: 2080-REMLCD

Preview: 2080-REMLCD

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

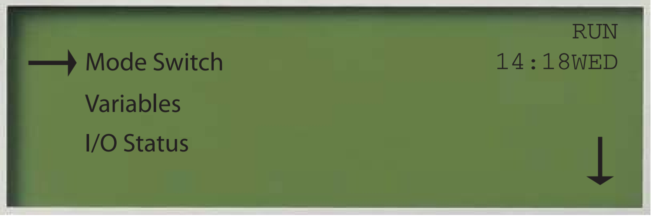
Button	Function
Arrow keys (cursor buttons)	Move cursor
	Select menu item
	Increment/Decrement Number
	Choose numbers, values, times, and so on
OK	Next menu level, store your entry
Esc	Previous menu level, cancel your entry.
F1	Variable (Shortcut)
F2	ENET Cfg (Shortcut)
F3	Mode Switch (Shortcut)
F4	Fault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

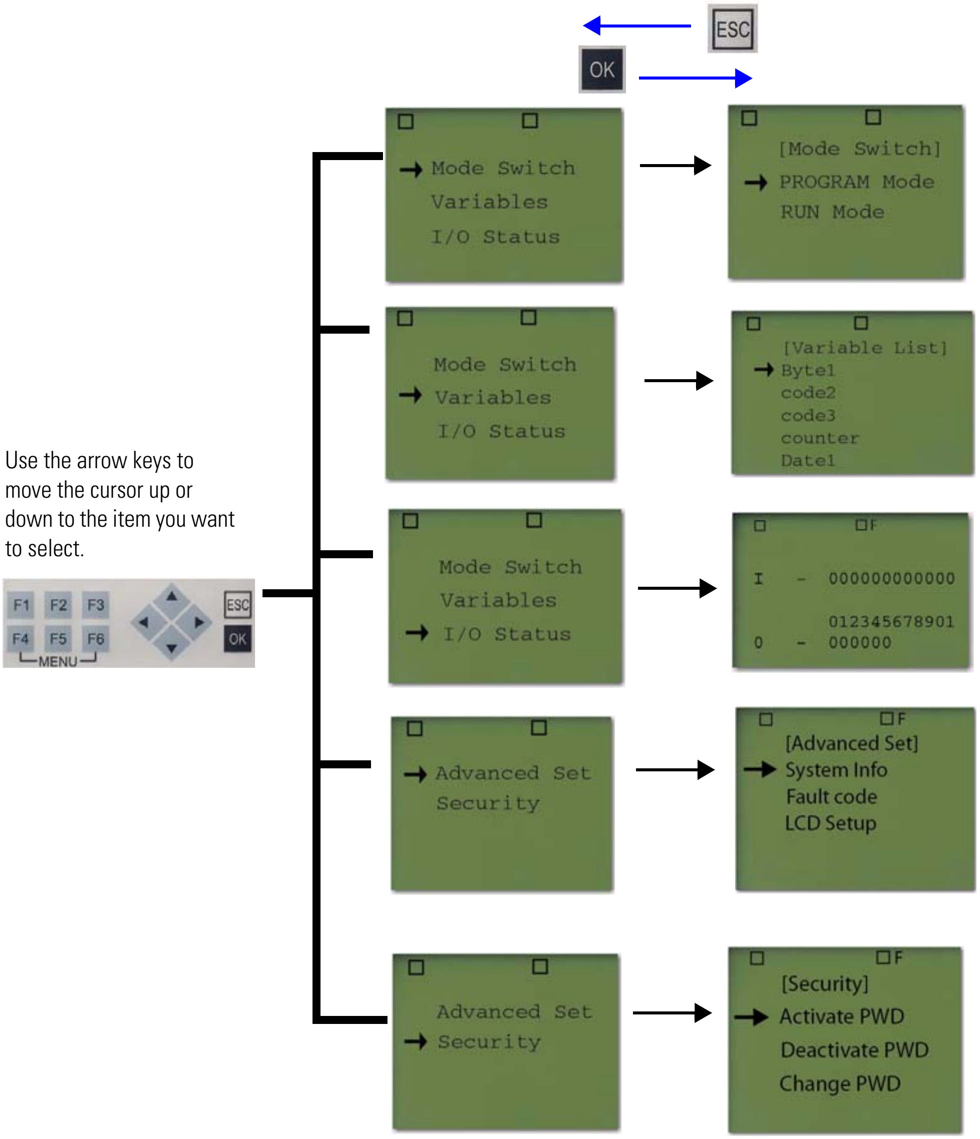
To access the Main Menu and available submenus, press F4 and F6 simultaneously. To exit the Main Menu, press ESC.

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



Use the arrow keys to move the cursor up or down to the item you want to select.

Mode Switch
Set the controller to Program Mode or Run mode from this screen.

Variable
Monitor or set values for program-defined variables.

I/O Status
Monitor the I/O status from this screen.

Advanced Set
View:

System Info	Analog Calibration
Fault Code	PwrUp Behavior
LCD Setup	Memory Card
Clock Setup	ENET Cfg
Language	

Security
Activate, deactivate, and change password.

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 68.	
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.

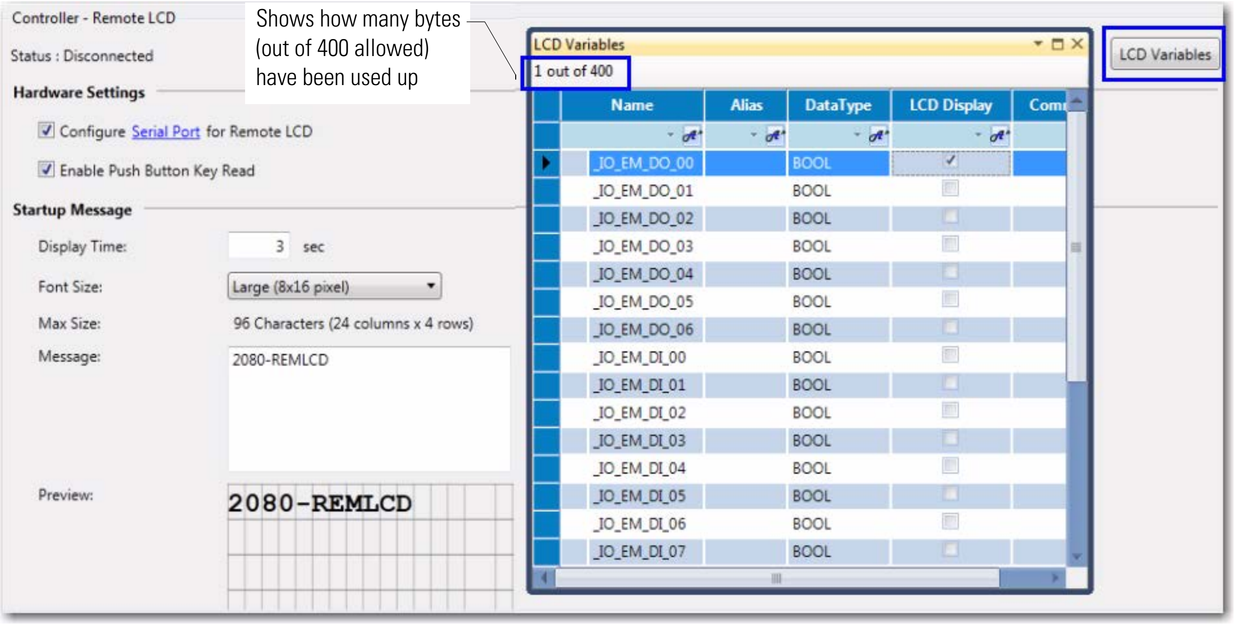
The controller limits certain operations according to controller mode, as shown in the following table.

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.



Shows how many bytes (out of 400 allowed) have been used up

Controller - Remote LCD

Status : Disconnected

Hardware Settings

☑ Configure Serial Port for Remote LCD

☑ Enable Push Button Key Read

Startup Message

Display Time: 3 sec

Font Size: Large (8x16 pixel)

Max Size: 96 Characters (24 columns x 4 rows)

Message: 2080-REMLCD

Preview: 2080-REMLCD

LCD Variables

1 out of 400

Name	Alias	DataType	LCD Display	Com
_IO_EM_DO_00		BOOL	☑	
_IO_EM_DO_01		BOOL	☐	
_IO_EM_DO_02		BOOL	☐	
_IO_EM_DO_03		BOOL	☐	
_IO_EM_DO_04		BOOL	☐	
_IO_EM_DO_05		BOOL	☐	
_IO_EM_DO_06		BOOL	☐	
_IO_EM_DI_00		BOOL	☐	
_IO_EM_DI_01		BOOL	☐	
_IO_EM_DI_02		BOOL	☐	
_IO_EM_DI_03		BOOL	☐	
_IO_EM_DI_04		BOOL	☐	
_IO_EM_DI_05		BOOL	☐	
_IO_EM_DI_06		BOOL	☐	
_IO_EM_DI_07		BOOL	☐	

User-defined Screens

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

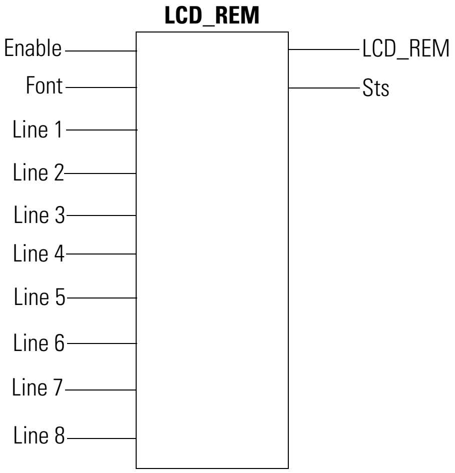
2080-REMLCD Function Blocks

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.

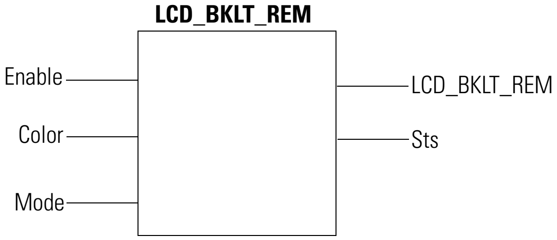
When the instructions are executing, the user-defined screen is shown, but when in the Main Menu, the Remote LCD instructions are disabled. For example, the KEY_READ_REM instruction will no longer read keypad input.

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



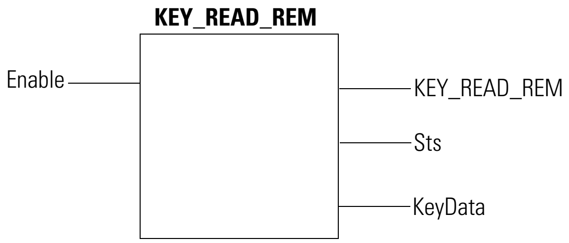
This function block is used to configure backlight parameters on the Remote LCD module.

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 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 73 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	73
Project Backup and Restore	73
Backup and Restore Directory Structure	75
Powerup Settings in ConfigMeFirst.txt	76
General Configuration Rules in ConfigMeFirst.txt	77
ConfigMeFirst.txt Errors	77
Datalog	78
Recipe	83
Quickstart Projects for Datalog and Recipe Function Blocks	87

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

Overview

Micro820 controllers support microSD cards for the following purposes:

- Project backup and restore
- Datalog and Recipe

IMPORTANT For optimum performance, 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 datalog files and directories.

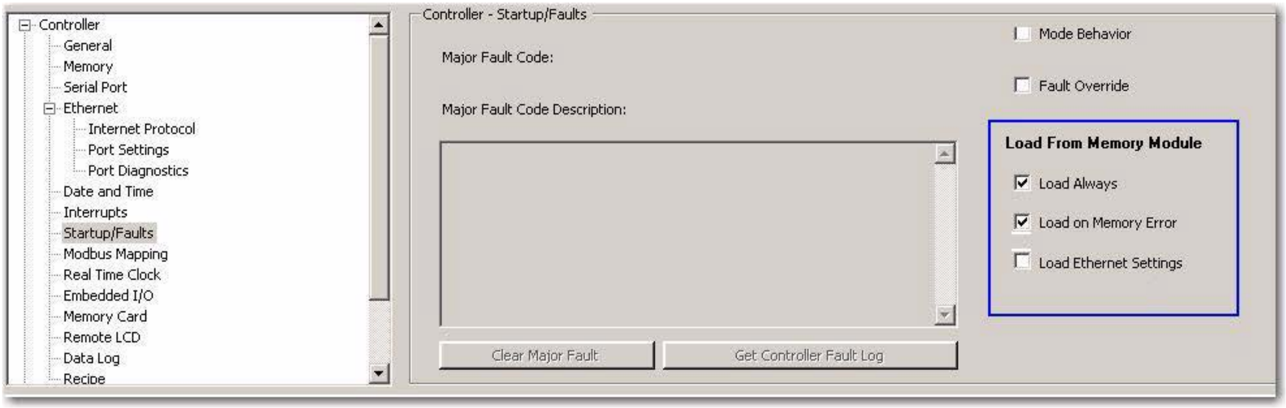
IMPORTANT Do not remove the microSD card or power down while operations such as upload, download, delete, search, backup and restore are ongoing to prevent data loss. A blinking SD status LED indicates that these operations are ongoing.

IMPORTANT To prevent data loss, recipe and datalog 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.



Controller - Startup/Faults

Major Fault Code:

Major Fault Code Description:

□ Mode Behavior

□ Fault Override

Load From Memory Module

☑ Load Always

☑ Load on Memory Error

□ Load Ethernet Settings

IMPORTANT To learn about restore and backup using the 2080-REMLCD module, see Using the Micro800 Remote LCD on page 63.

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

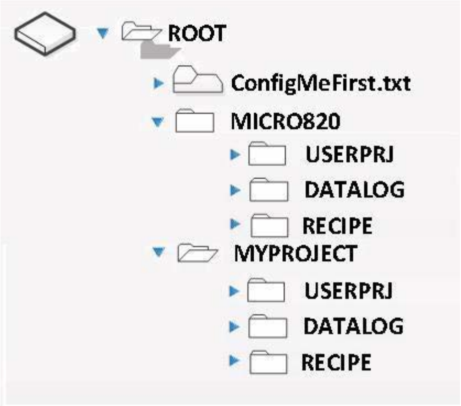
IMPORTANT For Micro800 controllers that support microSD cards, IP protection of user project can only be achieved through the POU password protection mechanism in Connected Components Workbench (Developer Edition) and NOT via Controller Lock feature.

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.

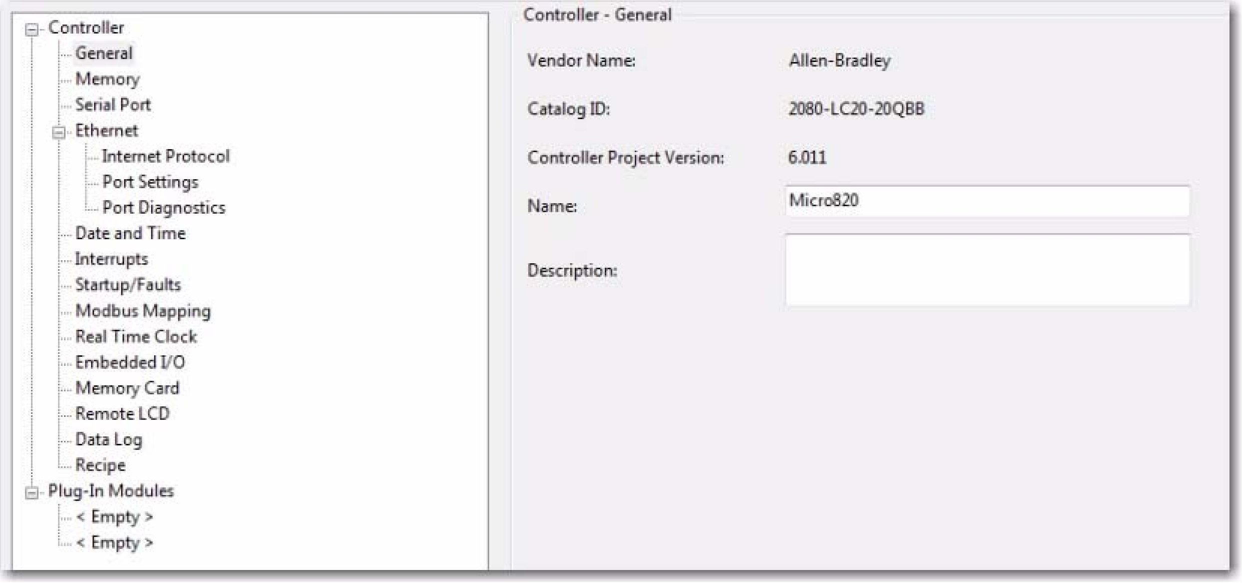
Backup and restore can be configured to trigger through the following ways:

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 77.



Controller - General

Vendor Name: Allen-Bradley

Catalog ID: 2080-LC20-20QBB

Controller Project Version: 6.011

Name: Micro820

Description:

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 datalog 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
[PM]	Power up and switch to PROGRAM mode.
[CF]	Power up and attempt to clear fault.
[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).
[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).
[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.
[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.

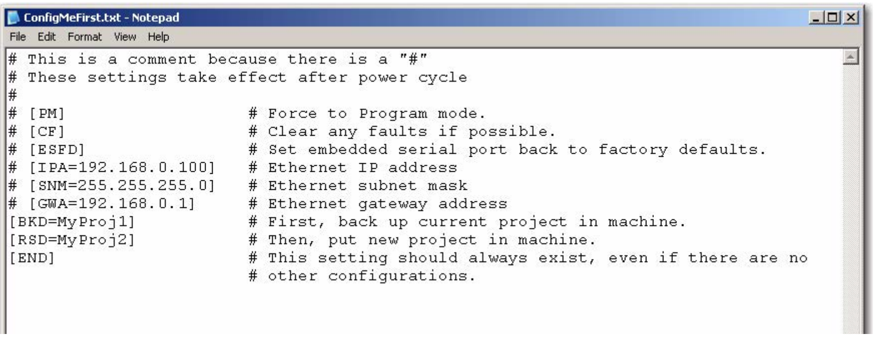
IMPORTANT

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)
- [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.

Datalog

The datalogging 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 datalogs are stored on the microSD card, see the Datalog Directory Structure on page 79.

Micro820 controllers typically support 10 MB of datalog per day.

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

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

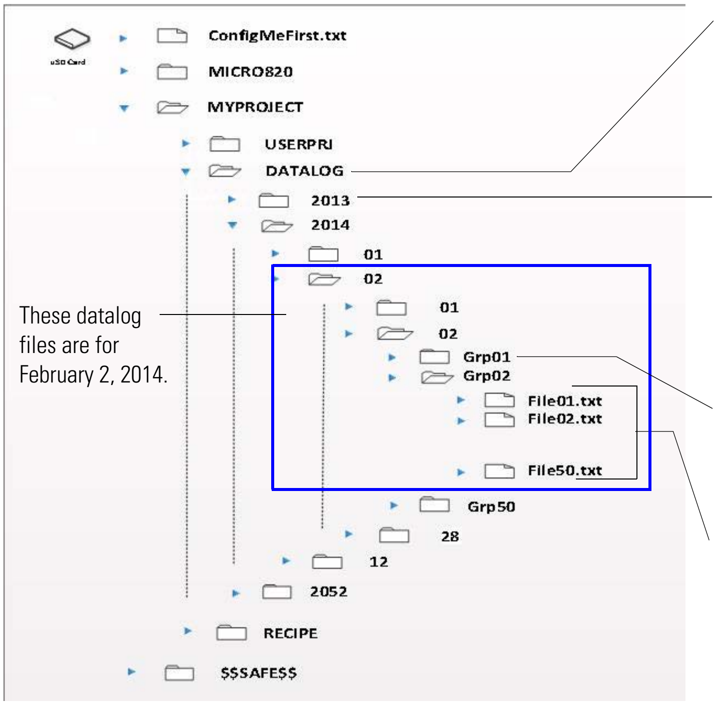
See the sample quickstart project to get you started on the Datalog feature, on page 87.

IMPORTANT Datalog execution time depends on the user application and its complexity. Users are advised to datalog once a minute for typical applications. Note that housekeeping takes at least 5 ms per program scan. See Program Execution in Micro800 on page 51 for more information on program scan and execution rules and sequence.

See also Datalog – Data Payload vs. Performance Time on page 108.

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.

Datalog Directory Structure



These datalog files are for February 2, 2014.

The DATALOG folder is created under the current project directory in the microSD card. This folder is created by default unless another directory has been specified in the ConfigMeFirst.txt. See [ConfigMeFirst.txt Configuration Settings on page 76](#).

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 datalog 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 Grpxx 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 datalog files for that day. This automatic folder and file generation goes on until the Grpxx folder reaches 50 for that day.
When a microSD card is inserted, the DLG function block looks for the last Grpxx folder and filexx.txt file, and proceeds to do the datalogging based on that information.

The following table summarizes datalogging performance onMicro820 controllers.

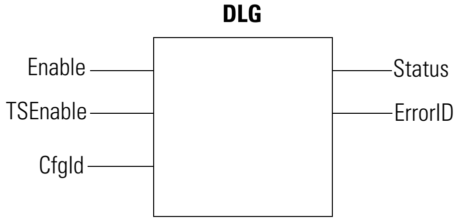
Datalog Specifications

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 datalog limits is reached (that is, 50 Grpxx folders per day, then an error (ErrorID 3: DLG_ERR_DATAFILE_ACCESS) is returned.

Datalog Function (DLG) Block

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



DLG Input and Output Parameters

Parameter	Parameter Type	Data Type	Description
Enable	INPUT	BOOL	Datalogging 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.
CfgId	INPUT	USINT	Configured dataset (DSET) number (1…10).
Status	OUTPUT	USINT	Datalogging function block current status.
ErrorID	OUTPUT	UDINT	Error ID if DLG Write fails.

DLG Function Block Status

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

DLG Function Block Errors

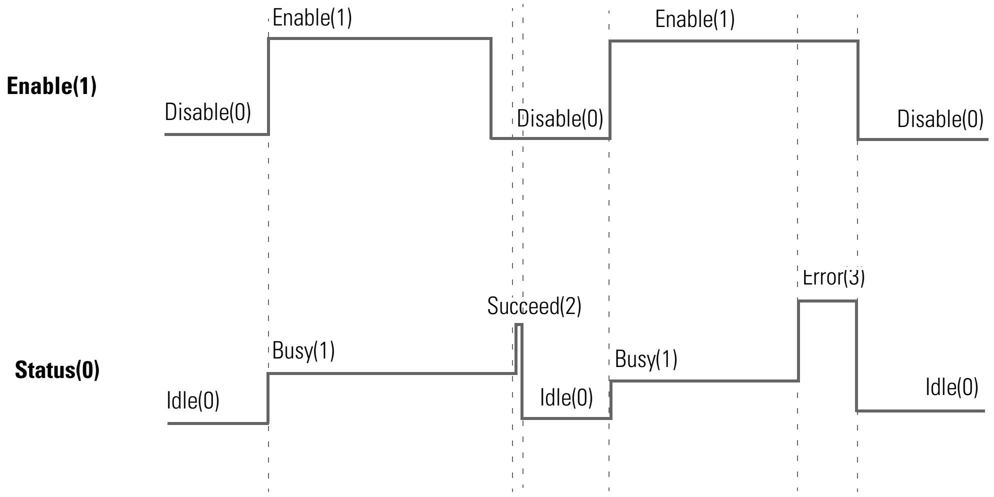
Status Code	Name	Description
0	DLG_ERR_NONE	No error.
1	DLG_ERR_NO_SDCARD	microSD card is missing.
2	DLG_ERR_RESERVED	Reserved.
3	DLG_ERR_DATAFILE_ACCESS	Error accessing datalog file in microSD card.
4	DLG_ERR_CFG_ABSENT	Datalog configuration file is absent.
5	DLG_ERR_CFG_ID	Configuration ID is missing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

DLG Function Block Errors

Status Code	Name	Description
7	DLG_ERR_CFG_FORMAT	Datalog 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.

Datalog Function Block Timing Diagram



IMPORTANT **Datalog Function Block Execution**

- There are three possible states for the Datalog function block: Idle, Busy and Complete (which includes Complete with Succeed and Complete with Error).
- For one Datalog 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 CfgId 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 CfgId 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 datalog file can only be created by the DLG instruction block. Connected Components Workbench can only upload and delete the datalog 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 Datalog and Recipe Function Blocks on page 82.
- Data variable values are sampled when datalogging function block is in Busy state. However, datalogging file is only created when datalogging function block is in Complete state.

Supported Data Types for Datalog and Recipe Function Blocks

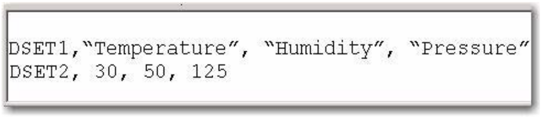
Data Type	Description	Example format in output datalog file
BOOL[1]	Logical Boolean with values TRUE and FALSE	0: FALSE 1: TRUE)
SINT	Signed 8-bit integer value	-128, 127
INT	Signed 16-bit integer value	-32768, 32767
DINT	Signed 32-bit integer value	-2147483648, 2147483647
LINT	Signed 64-bit integer 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 Datalog and Recipe Function Blocks

Data Type	Description	Example format in output datalog 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 datalog file.
The example below shows DSET1 using string variables and DSET2 using integers.



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

Recipe

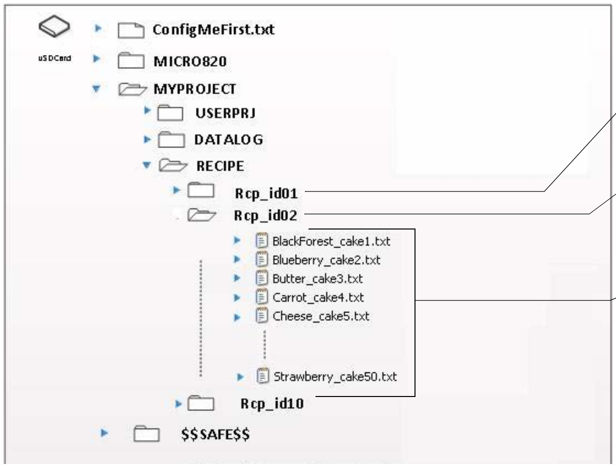
Micro820 controllers support the Recipe feature and allows users to store and load a list of data to and/or from recipe data files using the RCP instruction. It also allows users to download, upload, and delete Recipe data on the microSD card through Connected Components Workbench.

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 84.

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.

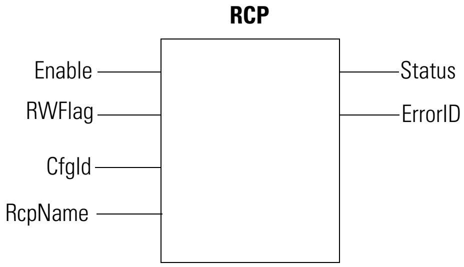
Recipe files are then created/written into the folder, with file names that correspond to the input value of RcpName parameter for the RCP function block, as configured in Connected Components Workbench. Each Recipe set can contain up to 50 recipe files or variations. Filenames for recipe files should 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

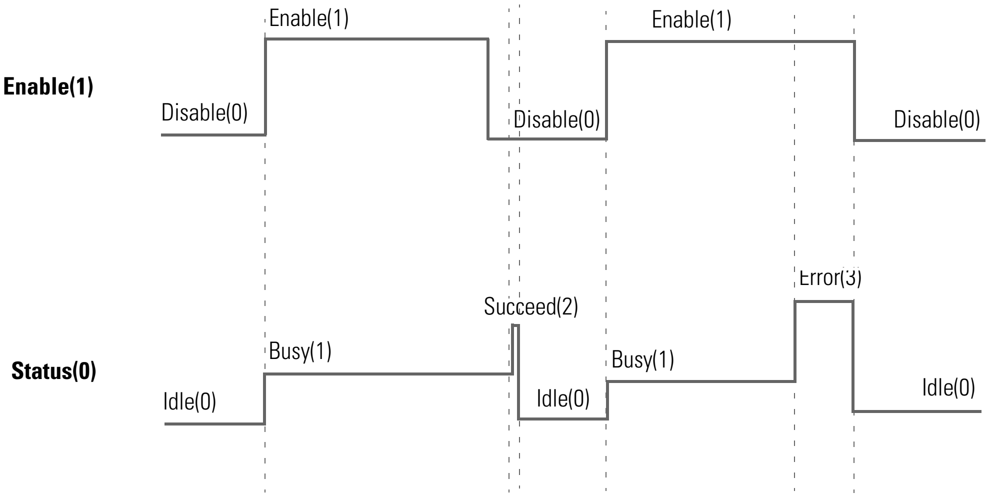
Error ID	Error name	Description
0	RCP_ERR_NONE	No error.
1	RCP_ERR_NO_SDCARD	microSD card is absent.
2	RCP_ERR_DATAFILE_FULL	Recipe files exceed maximum number of files per recipe set folder.
3	RCP_ERR_DATAFILE_ACCESS	Error to access recipe data file in microSD card.
4	RCP_ERR_CFG_ABSENT	Recipe configuration 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

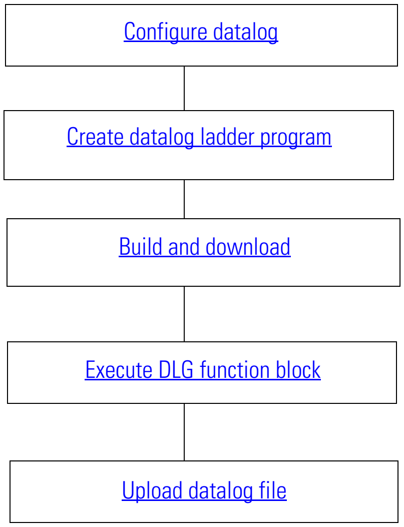


IMPORTANT	**RCP Function Block Execution**
	• There are three possible states for Recipe function block: Idle, Busy, Complete (Complete with Succeed and Complete with Error) • For one Recipe function block execution, the typical status starts from Idle then Busy and finishes with Complete. To trigger 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 Datalog and Recipe Function Blocks on page 82. • Double quotes are not allowed within a string in a recipe file.

Quickstart Projects for Datalog and Recipe Function Blocks

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

Use the Datalog Feature



Configure datalog

Create datalog ladder program

Build and download

Execute DLG function block

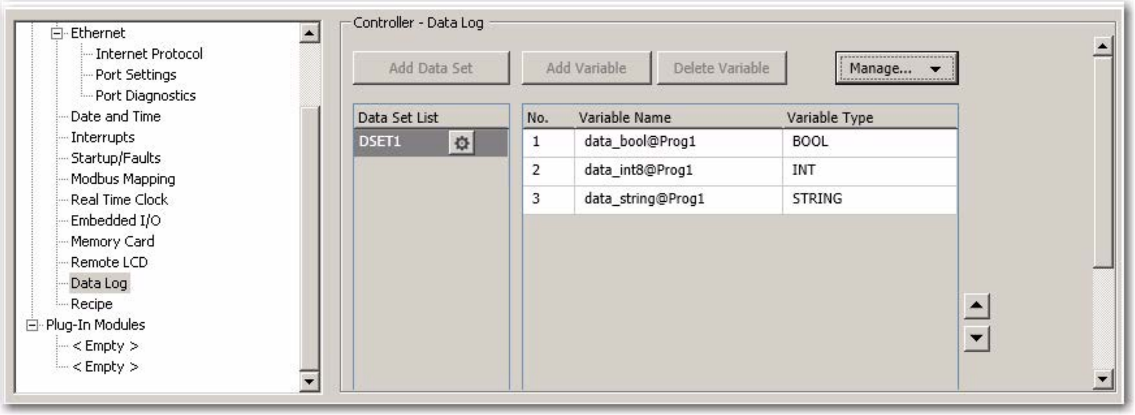
Upload datalog file

Configure datalog

1. In Connected Components Workbench, go to the Properties pane to configure your datalog.
2. Select Datalog. 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

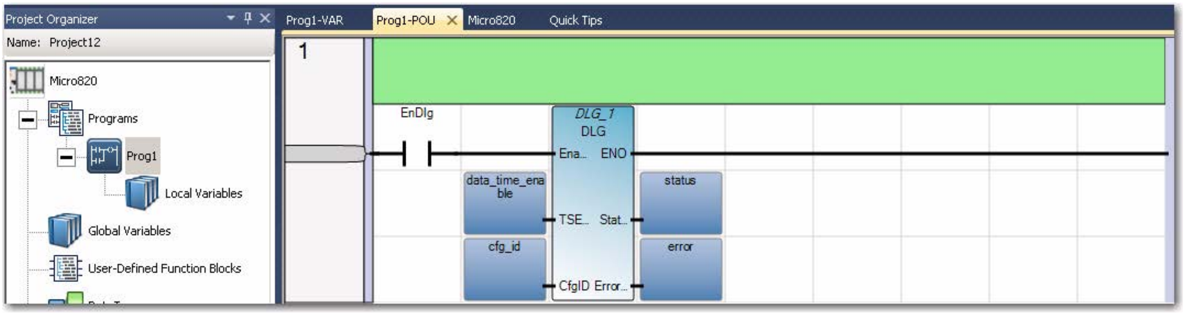
Variable Name	Data Type
data_bool	BOOL
data_int8	INT
data_string	STRING



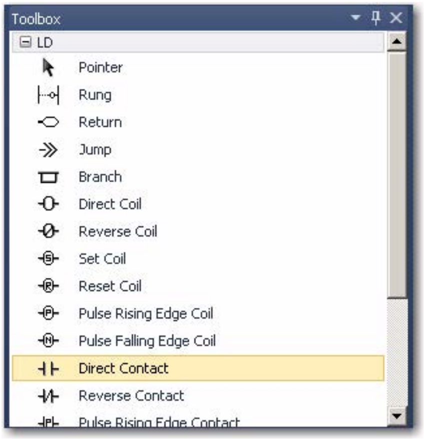
Data Set List
DSET1

No.	Variable Name	Variable Type
1	data_bool@Prog1	BOOL
2	data_int8@Prog1	INT
3	data_string@Prog1	STRING

Create datalog ladder program

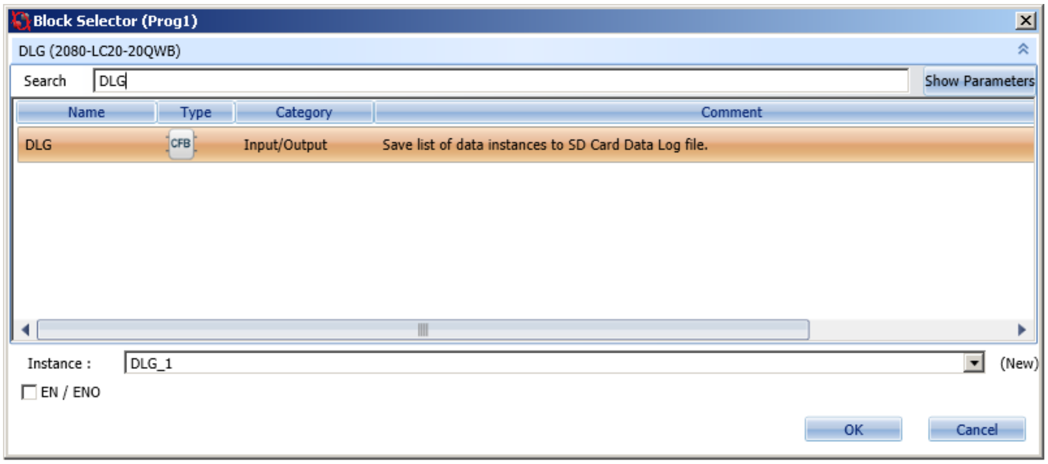


1. Launch Connected Components Workbench. Create a user program for your Micro820 controller.
2. Right-click Programs. Select Add New LD: Ladder Diagram. Name the Program (for example, Prog1).
3. From the Toolbox, double-click Direct Contact to add it to the rung.



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

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



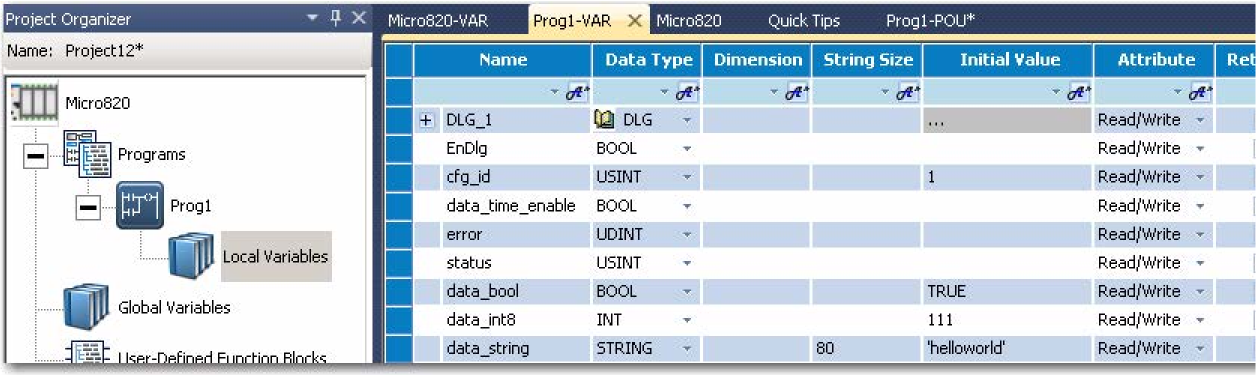
Search: DLG

Name	Type	Category	Comment
DLG	CFB	Input/Output	Save list of data instances to SD Card Data Log file.

Instance : DLG_1 (New)

□ EN / ENO

6. Create the following local variables for your project.

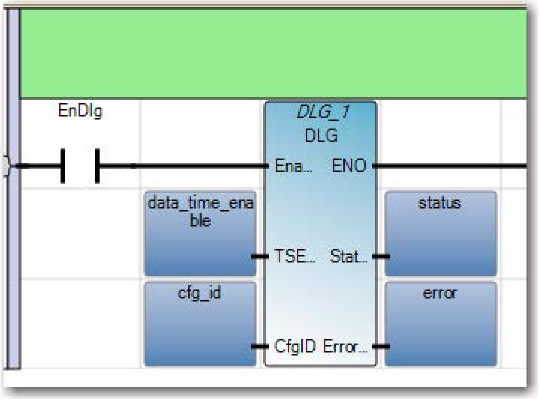


Name	Data Type	Dimension	String Size	Initial Value	Attribute	Ret
+ DLG_1	DLG			...	Read/Write	
EnDlg	BOOL				Read/Write	
cfg_id	USINT			1	Read/Write	
data_time_enable	BOOL				Read/Write	
error	UDINT				Read/Write	
status	USINT				Read/Write	
data_bool	BOOL			TRUE	Read/Write	
data_int8	INT			111	Read/Write	
data_string	STRING		80	'helloworld'	Read/Write	

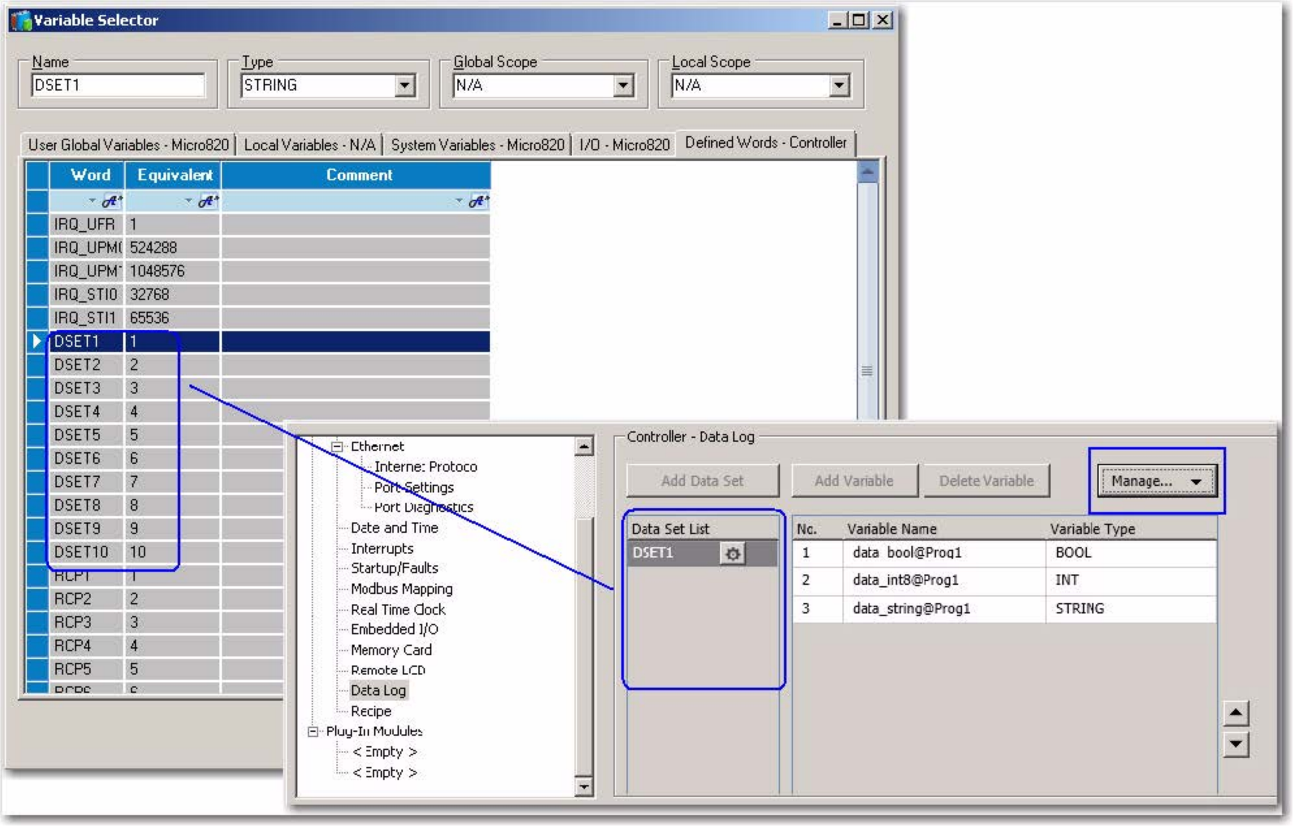
Local Variables

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.



Name: DSET1
Type: STRING
Global Scope: N/A
Local Scope: N/A

Word	Equivalent	Comment
IRQ_UFR	1	
IRQ_UPM(	524288	
IRQ_UPM'	1048576	
IRQ_STI0	32768	
IRQ_STI1	65536	
DSET1	1	
DSET2	2	
DSET3	3	
DSET4	4	
DSET5	5	
DSET6	6	
DSET7	7	
DSET8	8	
DSET9	9	
DSET10	10	
RCP1	1	
RCP2	2	
RCP3	3	
RCP4	4	
RCP5	5	
RCP6	6	

Data Set List: DSET1

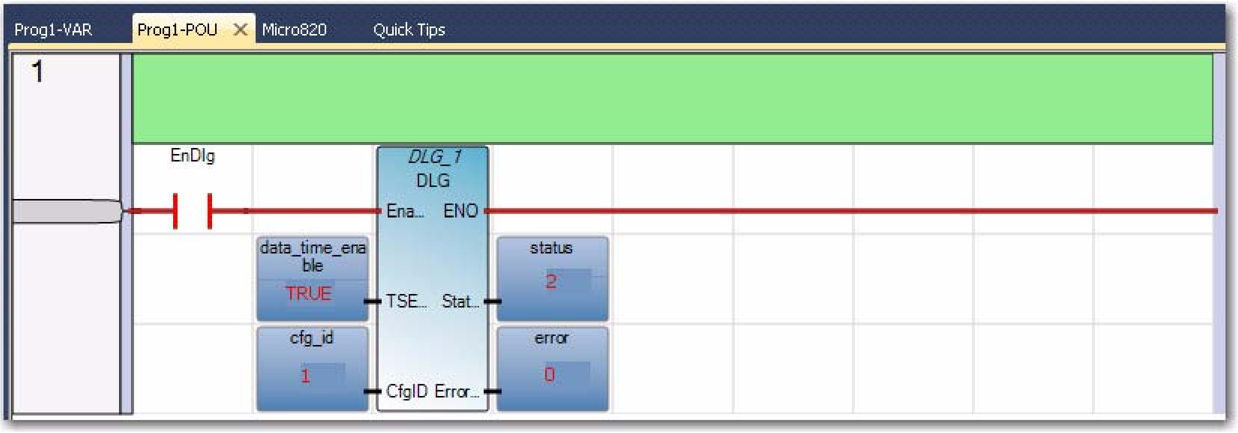
No.	Variable Name	Variable Type
1	data_bool@Prog1	BOOL
2	data_int8@Prog1	INT
3	data_string@Prog1	STRING

Build and download

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

Execute DLG function block

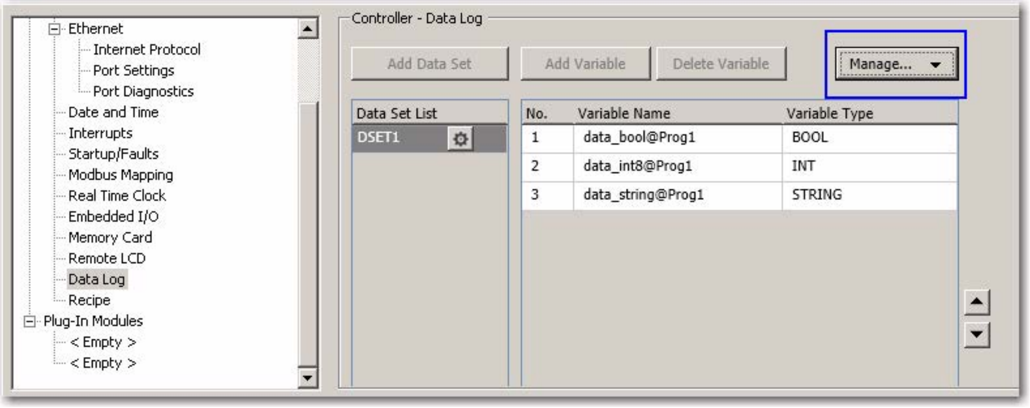
Execute the DLG function block. Notice the Status output go from 0 (Idle) to 1 (Enable), and 2 (Succeed).



Upload datalog file

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

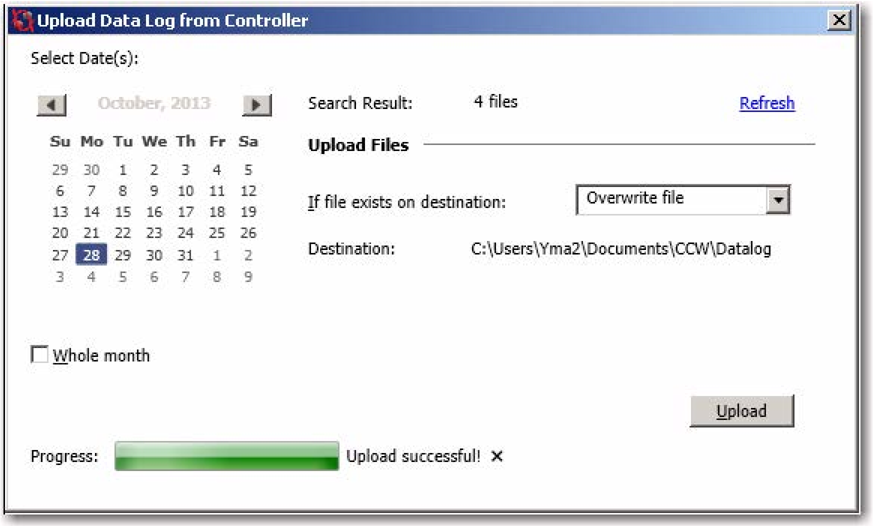
1. To use the Upload feature, go to the Properties section of your project in Connected Components Workbench.
2. Select Data Log. Click Manage and then choose Upload.



No.	Variable Name	Variable Type
1	data_bool@Prog1	BOOL
2	data_int8@Prog1	INT
3	data_string@Prog1	STRING

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

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



Su	Mo	Tu	We	Th	Fr	Sa
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Search Result: 4 files

If file exists on destination: Overwrite file

Destination: C:\Users\Yma2\Documents\CCW\Datalog

□ Whole month

Progress: Upload successful! ✕

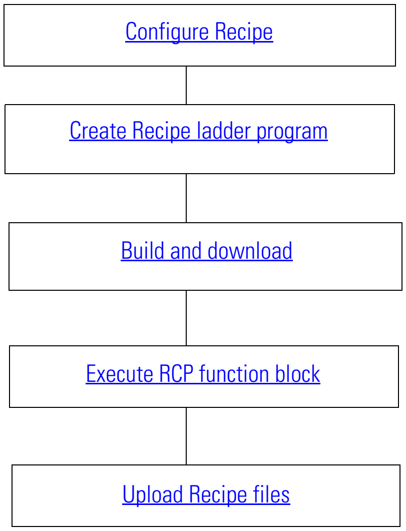
4. If the file already exists in your destination folder, select whether you would like to Overwrite file, Skip file, or Preserve both files.

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

Use the Recipe Feature

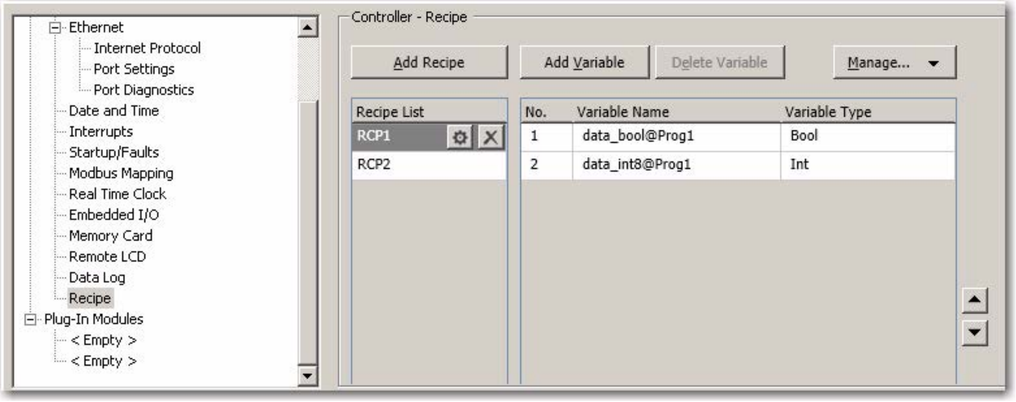


Configure Recipe

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

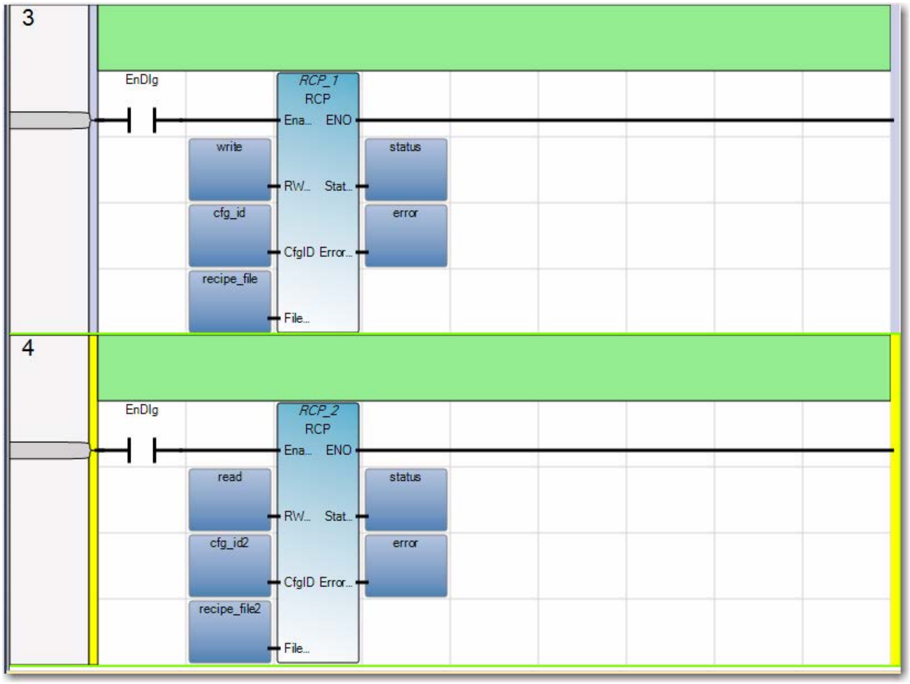
Variable Name	Data Type
data_bool	BOOL
data_int8	INT



Recipe List
RCP1
RCP2

No.	Variable Name	Variable Type
1	data_bool@Prog1	Bool
2	data_int8@Prog1	Int

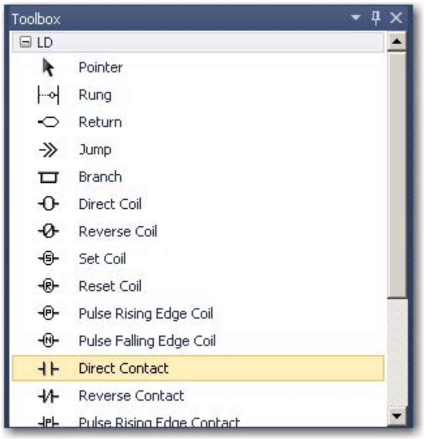
Create Recipe ladder program



1. Launch Connected Components Workbench. Create a user program for your Micro820 controller.

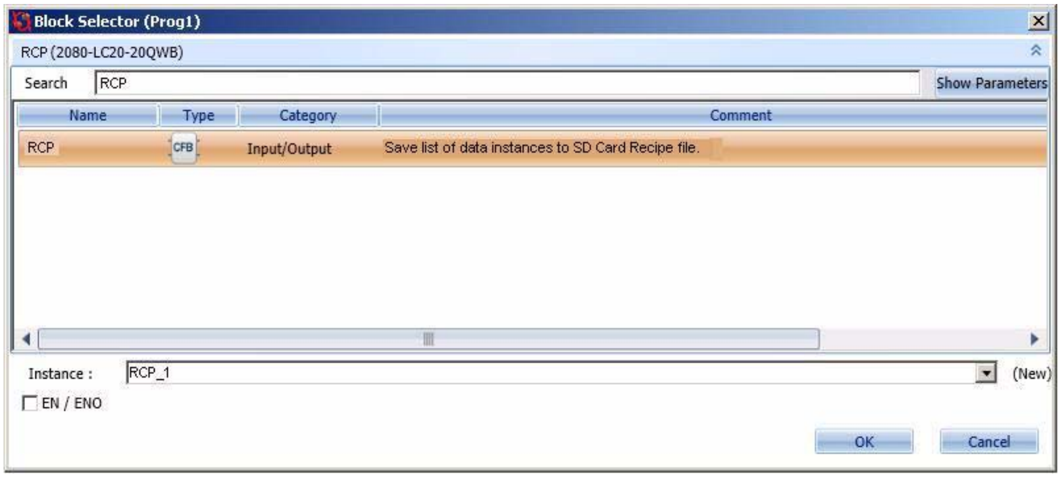
2. Right-click Programs. Select Add New LD: Ladder Diagram. Name the Program (for example, Prog2).

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.



Search: RCP

Name	Type	Category	Comment
RCP	CFB	Input/Output	Save list of data instances to SD Card Recipe file.

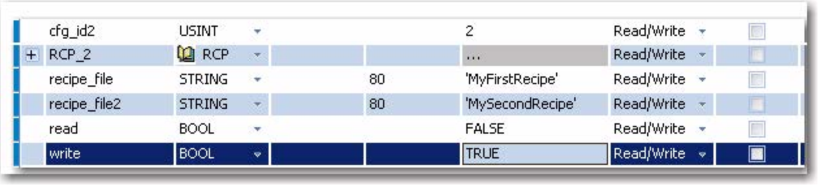
Instance : RCP_1 (New)

□ EN / ENO

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 datalog.



cfg_id2	USINT			2	Read/Write	□
+ RCP_2	RCP			...	Read/Write	□
recipe_file	STRING		80	'MyFirstRecipe'	Read/Write	□
recipe_file2	STRING		80	'MySecondRecipe'	Read/Write	□
read	BOOL			FALSE	Read/Write	□
write	BOOL			TRUE	Read/Write	□

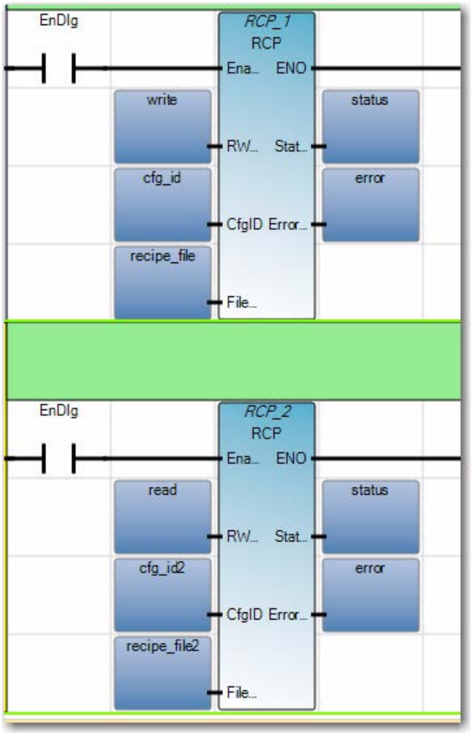
Local Variables

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

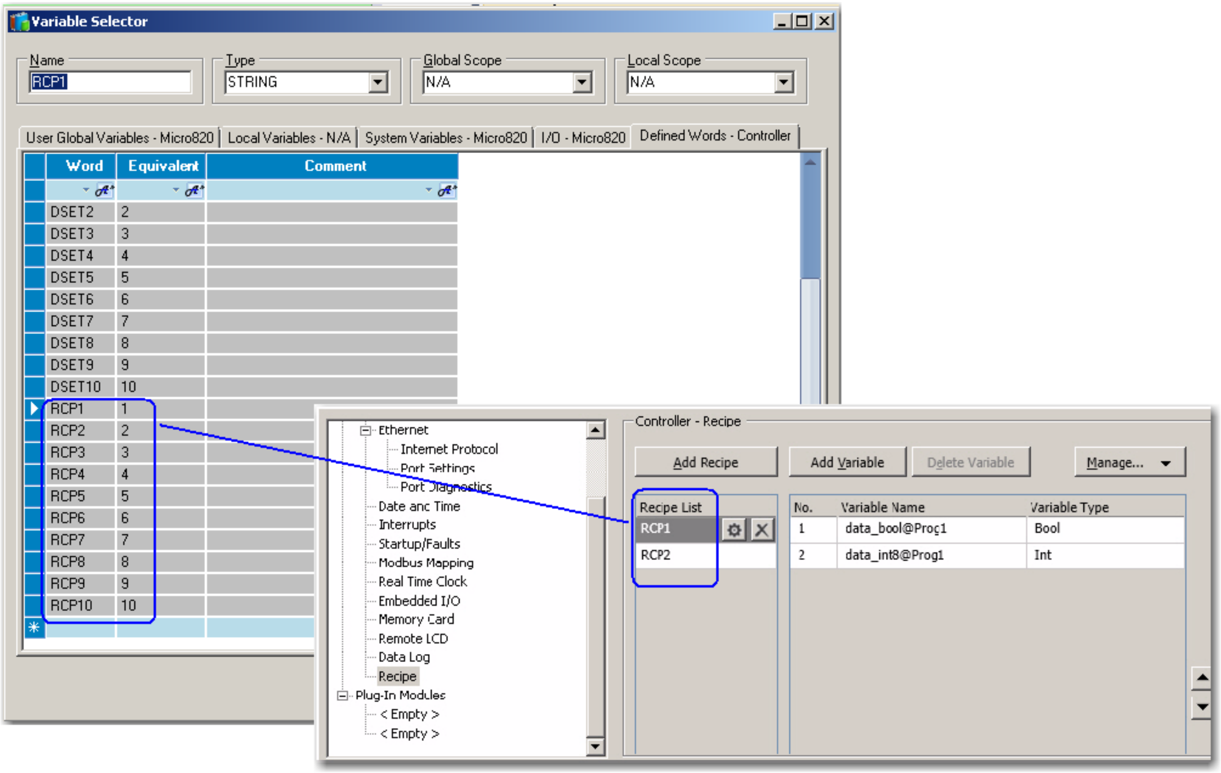
9. Assign the variables to the RCP input and output parameters as follows:

Rung 1

Rung 2



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.



Name	Type	Global Scope	Local Scope
RCP1	STRING	N/A	N/A

Word	Equivalent	Comment
DSET2	2	
DSET3	3	
DSET4	4	
DSET5	5	
DSET6	6	
DSET7	7	
DSET8	8	
DSET9	9	
DSET10	10	
RCP1	1	
RCP2	2	
RCP3	3	
RCP4	4	
RCP5	5	
RCP6	6	
RCP7	7	
RCP8	8	
RCP9	9	
RCP10	10	

Recipe List
RCP1
RCP2

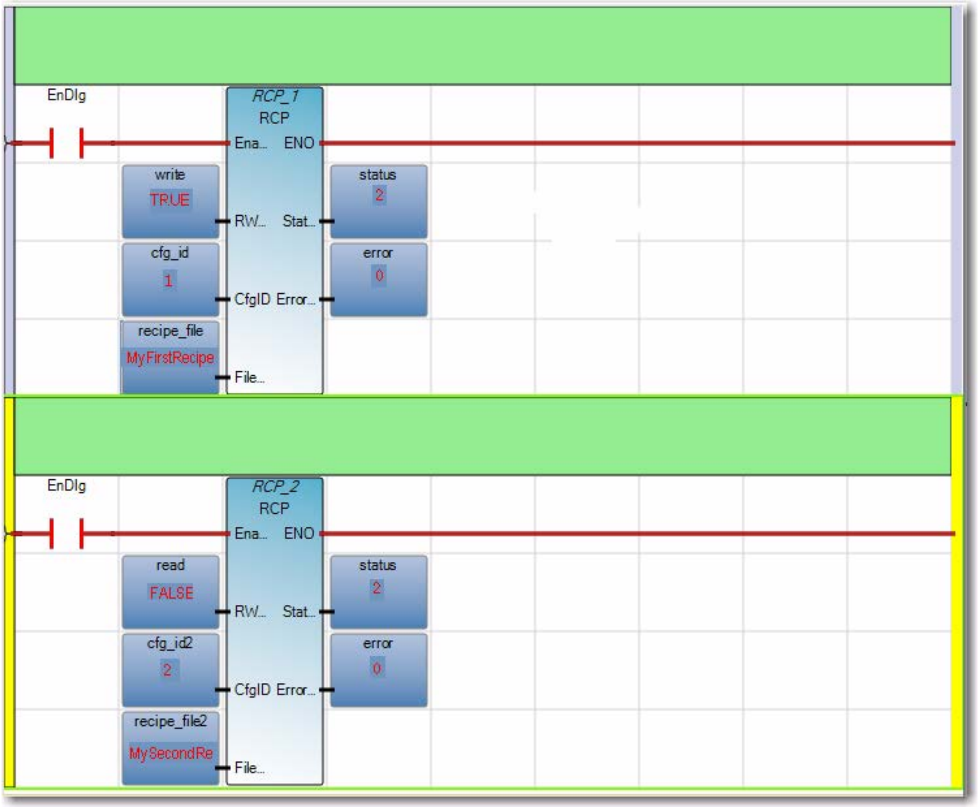
No.	Variable Name	Variable Type
1	data_bool@Prog1	Bool
2	data_int8@Prog1	Int

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).

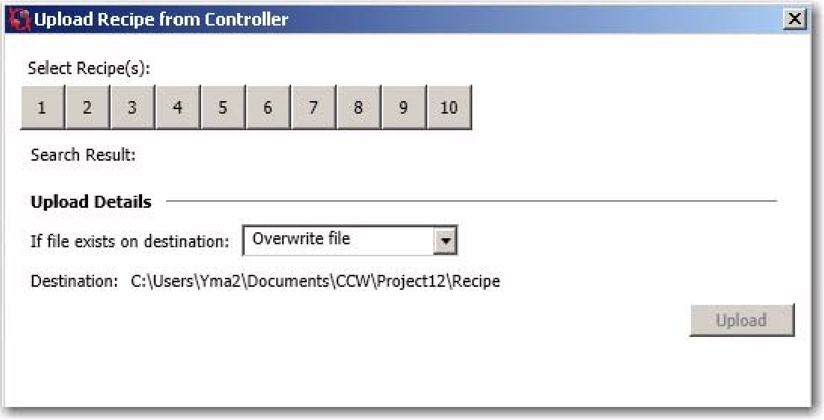


Upload Recipe files

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.

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



Upload Recipe from Controller

Select Recipe(s):

Search Result:

Upload Details

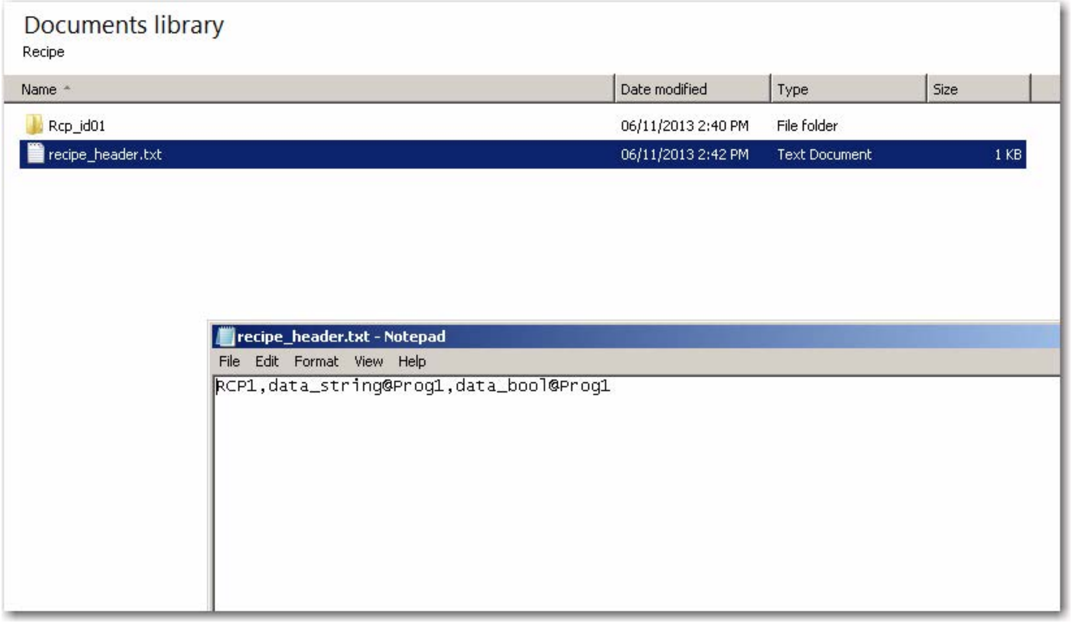
If file exists on destination: Overwrite file

Destination: C:\Users\Yma2\Documents\CCW\Project12\Recipe

4. If the file already exists in your destination folder, select whether you would like to Overwrite file, Skip file, or Preserve both Files.
5. 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.



Documents library
Recipe

Name	Date modified	Type	Size
Rcp_id01	06/11/2013 2:40 PM	File folder	
recipe_header.txt	06/11/2013 2:42 PM	Text Document	1 KB

recipe_header.txt - Notepad

```
RCP1,data_string@Prog1,data_bool@Prog1
```

Specifications

The Micro820 controllers have the following specifications and certifications.

General Specifications

<table>
<tr><th>Attribute</th><th>2080-LC20-20AWB(R)</th><th>2080-LC20-20QBB(R)</th><th>2080-LC20-20QWB(R)</th></tr>
<tr><td>Number of I/O</td><td colspan="3">12 inputs, 8 outputs</td></tr>
<tr><td>Dimensions
HxWxD</td><td colspan="3">90 x 104 x 75 mm
(3.54 x 4.09 x 2.95 in.)</td></tr>
<tr><td>Shipping weight, approx.</td><td colspan="3">0.38 kg (0.83 lb)</td></tr>
<tr><td>Wire size</td><td colspan="3"><b>For fixed terminal blocks:</b>
<table>
<tr><th></th><th>Min</th><th>Max</th><th></th></tr>
<tr><td>Solid</td><td>0.14 mm^2 (26 AWG)</td><td>2.5 mm^2 (14 AWG)</td><td>rated @ 90 °C
(194 °F) insulation max</td></tr>
<tr><td>Stranded</td><td>0.14 mm^2 (26 AWG)</td><td>1.5 mm^2 (16 AWG)</td><td></td></tr>
</table>
<b>For removable terminal blocks:</b>
<table>
<tr><th></th><th>Min</th><th>Max</th><th></th></tr>
<tr><td>Solid and Stranded</td><td>0.2 mm^2 (24 AWG)</td><td>2.5 mm^2 (14 AWG)</td><td>rated @ 90 °C
(194 °F) insulation max</td></tr>
</table>
<b>For RS232/RS485 serial port:</b>
<table>
<tr><th></th><th>Min</th><th>Max</th><th></th></tr>
<tr><td>Solid</td><td>0.14 mm^2 (26 AWG)</td><td>1.5 mm^2 (16 AWG)</td><td>rated @ 90 °C
(194 °F) insulation max</td></tr>
<tr><td>Stranded</td><td>0.14 mm^2 (26 AWG)</td><td>1.0 mm^2 (18 AWG)</td><td></td></tr>
</table></td></tr>
<tr><td>Wiring category[1]</td><td colspan="3">2 – on signal ports
2 – on power ports
2 – on communication ports</td></tr>
<tr><td>Wire type</td><td colspan="3">Use copper conductors or shielded cables</td></tr>
<tr><td>Terminal screw torque</td><td colspan="3"><b>For removable and fixed terminal blocks:</b>
0.5…0.6 Nm (4.4…5.3 lb-in.) using a 0.6 x 3.5 mm flat-blade screwdriver.
<b>Note:</b> Use a handheld screwdriver to hold down the screws at the side.
<b>For RS232/RS485 serial port:</b>
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.</td></tr>
<tr><td>Input circuit type</td><td colspan="3">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</td></tr>
<tr><td>Output circuit type</td><td>Relay</td><td>24V DC source
(standard and high-speed)</td><td>Relay</td></tr>
<tr><td>Power input</td><td colspan="3">24V DC</td></tr>
<tr><td>Power consumption</td><td colspan="3">5.62 W (without plug-ins, max)…8.5 W (with plug-ins, max)</td></tr>
</table>

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 kΩ
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 kΩ (non-isolated)	3.74 kΩ (isolated)
IEC input compatibility	Type 1	Type 3

Relay Output Specifications for O-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 O-00…O-05)	High Speed Output [(1)] (Output O-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 O-00…O-05)	High Speed Output [1] (Output O-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.

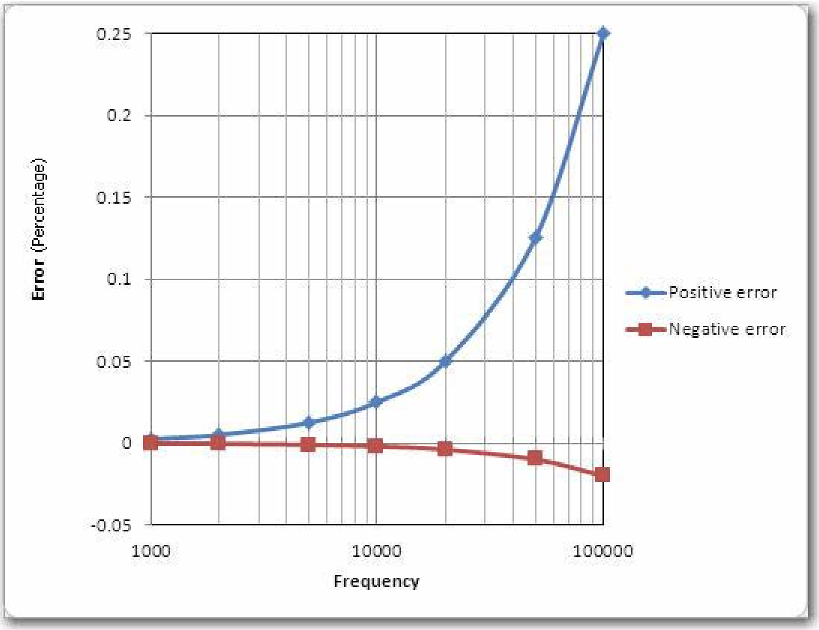
PWM Output Duty Cycle Error

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 μs * F
Negative error = -0.2 μs * 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

$30\%^{+5\%}_{-0.4\%}.$



Auxiliary Power Supply for Thermistor Applications

Attribute	Value
Output voltage	9.5V, min 10.04V, typical 10.5V, max
Output current	10 mA, typical 50 mA, max

Embedded RTC

Attribute	Value
Resolution	1 sec
Accuracy	± 52 sec/month @ 25 °C ± 160 sec/month @ 0...55 °C
Power off	Supercap – 4 days @ 25 °C Supercap life – 5 years @ 40 °C, 14.5 years @ 25 °C

PWM Typical Readings

PWM Typical Readings

		Expected Duty Cycle		Typical Duty Cycle (1.27 KΩ 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	75%	74.90%	76.25%	75.5
5	95%	94.90%	96.25%	95.5
5	65%	64.90%	66.25%	65.5
10	5%	4.80%	7.50%	5.9
10	10%	9.80%	12.50%	11
10	20%	19.80%	22.50%	21
10	40%	39.80%	42.50%	40.9
10	55%	54.80%	57.50%	55.9
10	65.00%	64.80%	67.50%	65.9
10	85.00%	84.80%	87.50%	85.9
10	95.00%	94.80%	97.50%	95.9
25	5.00%	4.50%	11.25%	7.25
25	10.00%	9.50%	16.25%	12.3
25	20.00%	19.50%	26.25%	22.4
25	40.00%	39.50%	46.25%	42.3
25	55.00%	54.50%	61.25%	57.3
25	65.00%	64.50%	71.25%	67.3
25	85.00%	84.50%	91.25%	87.3
25	95.00%	94.50%	100%	97
50	5%	4%	17.50%	9.7
50	10%	9%	22.50%	14.8
50	20%	19%	32.50%	24.7
50	40%	39%	52.50%	44.7
50	55%	54%	67.50%	59.6
50	65%	64%	77.50%	69.6
50	85%	84%	97.50%	89.5
50	95%	94%	100%	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

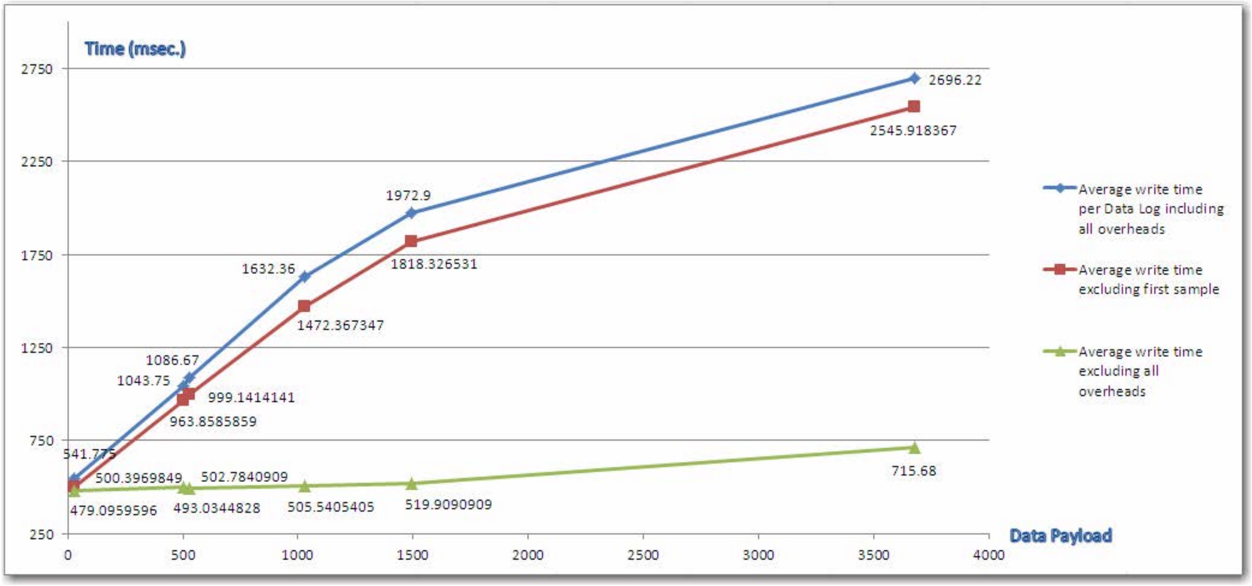
PWM Typical Readings

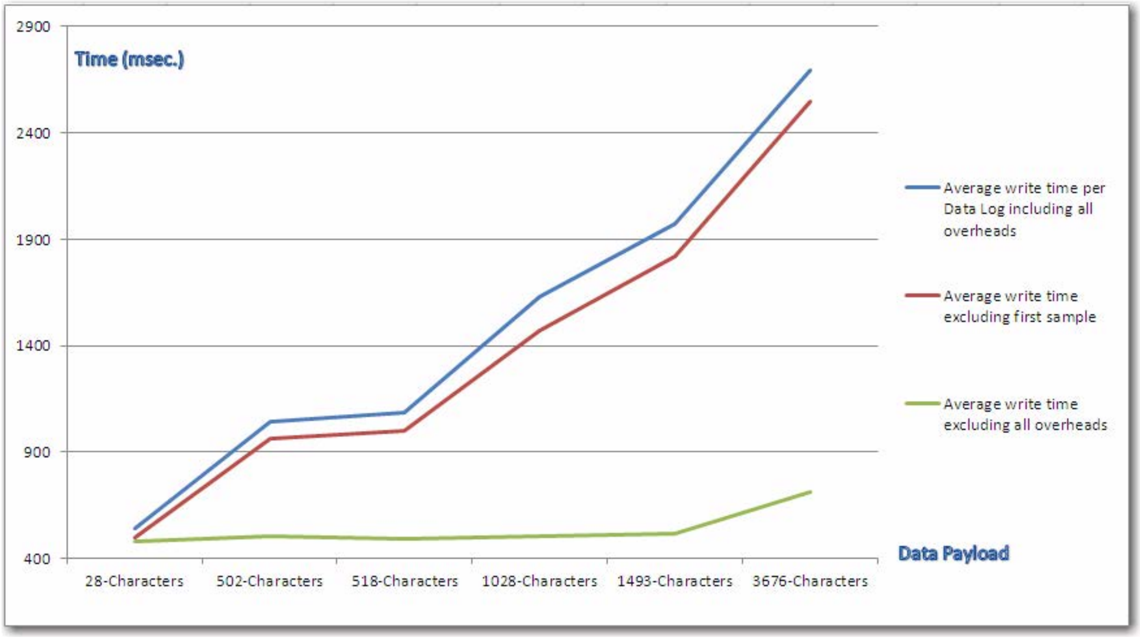
		Expected Duty Cycle		Typical Duty Cycle (1.27 KΩ load)
Frequency (Khz)	**%Duty Cycle**	**Minimum**	**Maximum**	**%Duty Cycle**
100	40%	38.00%	65.00%	49.3
100	55%	53.00%	80.00%	64
100	65%	63.00%	90.00%	73.8
100	85%	83.00%	100.00%	92.4
100	95%	93.00%	100.00%	98

Datalog Performance

Datalog – Data Payload vs. Performance Time

Parameter	Number of Characters					
	28	**502**	**518**	**1028**	**1493**	**3676**
Average write time per datalog 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

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
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

Environmental Specifications

Attribute	Value
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

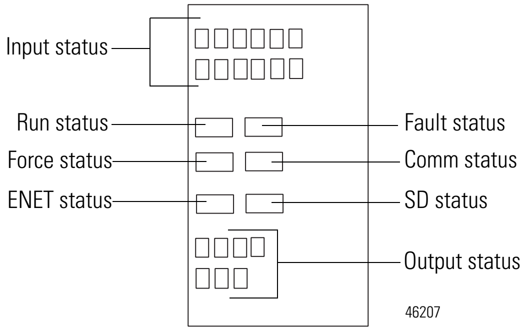
Certification (when product is marked)[1]	Value
c-UL-us	UL Listed Industrial Control Equipment, certified for US and Canada. See UL File E322657. UL Listed for Class I, Division 2 Group A,B,C,D Hazardous Locations, certified for U.S. and Canada. See UL File E334470.
CE	European Union 2004/108/EC EMC 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	Controller hard fault.
		Flashing red	Major fault detected.
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.
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	• 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	• 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 77 for list of errors.
		On Idle State	• 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 ink 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 Conditions

If an error exists within the controller, the controller indicators operate as described in the following table.

Indicator Behavior	Probable Error	Probable Cause	Recommended Action
All indicators off	No input power or power supply error	No line power	Verify proper line voltage and connections to the controller.
		Power supply overloaded	This problem can occur intermittently if power supply is overloaded when output loading and temperature varies.
Power and FAULT indicators on solid	Hardware faulted	Processor hardware error	Cycle power. Contact your local Allen-Bradley representative if the error persists.
		Loose wiring	Verify connections to the controller.
Power on with solid indicator and FAULT indicator flashing	Application fault	Hardware/software major fault detected	For error codes and status information, refer to the Connected Components Workbench online Help
Power on with solid indicator and FAULT indicator flashing	Operating system fault	Firmware upgrade unsuccessful	See Flash Upgrade Your Micro800 Firmware on page 121.

Error codes

This section lists possible error codes for your controller, as well as recommended actions for recovery.

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

List of Error Codes for Micro800 Controllers

Error Code	Description	Recommended Action
0xF000	The controller was unexpectedly reset due to a noisy environment or an internal hardware failure. • A **Micro800 controller revision 2 and later** attempts to save the program and clear the user data. 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. • A **Micro800 controller revision 1.xx** clears the program. Note that the system variable _SYSVA_USER_DATA_LOST is not available on Micro800 controllers revision 1.xx.	Perform one of the following: • Download the program through Connected Components Workbench. • Refer to Wiring Requirements and Recommendation on page 25. If the fault persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.
0xF001	The controller program has been cleared. This happened because: • 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: • Download the program using Connected Components Workbench software. • Transfer the program using the memory module restore utility or the microSD card. If the fault persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.
0xF002	The controller hardware watchdog was activated. • A **Micro800 controller revision 2 and later** attempts to save the program and clear the user data. 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. • A **Micro800 controller revision 1.xx** clears the program. Note that the system variable _SYSVA_USER_DATA_LOST is not available on Micro800 controllers revision 1.xx.	Perform the following: • Establish a connection to the Micro800 controller. • Download the program using Connected Components Workbench. If the fault persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.
0xD00F	A particular hardware type (for example, embedded I/O) was selected in the user program configuration, but did not match the actual hardware base.	Perform one of the following: • Connect to the hardware that is specified in the user program. • Reconfigure the program to match the target hardware type.
0xF003	One of the following occurred: • 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: • Remove the memory module and plug it in again. • Obtain a new memory module. • 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	There may be failure in either: • Memory module data transfer, or • Embedded RTC data	For Memory Module failure, retry data transfer. If the error persists, replace the memory module. For embedded RTC failure, cycle power the controller. If the error persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.

List of Error Codes for Micro800 Controllers

Error Code	Description	Recommended Action
0xF005	The user program failed an integrity check while the Micro800 controller was in Run mode.	Perform one of the following: • Cycle power on your Micro800 controller. Then, download your program using Connected Components Workbench and start up your system. • Refer to the Wiring Requirements and Recommendation on page 25.
0xF006	The user program is incompatible with the Micro800 controller's firmware revision.	Perform one of the following: • Upgrade the Micro800 controller's firmware revision using ControlFlash. • Contact your local Rockwell Automation technical support representative for more information about firmware revisions for your Micro800 controller. For more information on firmware revision compatibility, go to http://www.rockwellautomation.com/support/firmware.html
0xF010	The user program contains a function/function block that is not supported by the Micro800 controller.	Perform the following: • Modify the program so that all functions/function blocks are supported by the Micro800 controller. • Build and download the program using Connected Components Workbench. • Put the Micro800 controller into Run mode.
0xF014	A memory module memory error occurred.	Reprogram the memory module. If the error persists, replace the memory module.
0xF015	An unexpected software error occurred.	Perform the following: 1. Cycle power on your Micro800 controller. 2. Build and download your program using Connected Components Workbench, and then reinitialize any necessary data. 3. Start up your system. • Refer to the Wire Your Controller on page 25.
0xF016	An unexpected hardware error occurred.	Perform the following: 1. Cycle power on your Micro800 controller. 2. Build and download your program using Connected Components Workbench, and then reinitialize any necessary data. 3. Start up your system. • Refer to the Wire Your Controller on page 25.
0xF017	An unexpected software error occurred due to unexpected hardware interrupt. If the system variable __SYSVA_USER_DATA_LOST has been set, the controller will be able to recover the user program, but the user data will be cleared. If not, the Micro800 controller program will be cleared.	Perform one of the following: • Download the program using Connected Components Workbench software. • Transfer the program using the memory module restore utility or the microSD card. If the fault persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.
0xF018	An unexpected software error occurred due to SPI communication failure. If the system variable __SYSVA_USER_DATA_LOST has been set, the controller will be able to recover the user program but the user data will be cleared. If not, the Micro800 controller program will be cleared.	Perform one of the following: • Download the program using Connected Components Workbench software.what • Transfer the program using the memory module restore utility or the microSD card. If the fault persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.

List of Error Codes for Micro800 Controllers

Error Code	Description	Recommended Action
0xF019	An unexpected software error occurred due to memory or other controller resource issue.	Perform one of the following: • Download the program using Connected Components Workbench software. • Transfer the program using the memory module restore utility or the microSD card. 4. If the fault persists, contact your local Rockwell Automation technical support representative. For contact information, see: http://support.rockwellautomation.com/MySupport.asp.
0xF020	The base hardware faulted or is incompatible with the Micro800 controller's firmware revision.	Perform one of the following: • Upgrade the Micro800 controller's firmware revision using ControlFlash. • Replace the Micro800 controller. • Contact your local Rockwell Automation technical support representative for more information about firmware revisions for your Micro800 controller. For more information on firmware revision compatibility, go to http://www.rockwellautomation.com/support/firmware.html
0xF021	The I/O configuration in the user program is invalid or does not exist in the Micro800 controller.	Perform the following: • Verify that you have selected the correct Micro800 controller from the Device Toolbox. • Correct the plug-in I/O module configuration in the user program to match that of the actual hardware configuration. • Recompile and reload the program. • Put the Micro800 controller into Run mode. • If the error persists, be sure to use Connected Components Workbench programming software to develop and download the program.
0xF022	The user program in the memory module is incompatible with the Micro800 controller's firmware revision.	Perform one of the following: • Upgrade the Micro800 controller's firmware revision using ControlFlash to be compatible with the memory module. • Replace the memory module. • Contact your local Rockwell Automation technical support representative for more information about firmware revisions for your Micro800 controller. For more information on firmware revision compatibility, go to http://www.rockwellautomation.com/support/firmware.html
0xF023	The controller program has been cleared. This happened because: • 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: • Download the program. • Contact your local Rockwell Automation technical support representative if the error persists. For contact information, see: • http://support.rockwellautomation.com/MySupport.asp.
0xF030/ 0xF031/ 0xF032/ 0xF033	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 has been set, the controller will be able to recover the user program but the user data will be cleared. If not, the Micro800 controller program will be cleared.	Perform one of the following: • Download the program. • Contact your local Rockwell Automation technical support representative if the error persists. For contact information, see: • http://support.rockwellautomation.com/MySupport.asp.

List of Error Codes for Micro800 Controllers

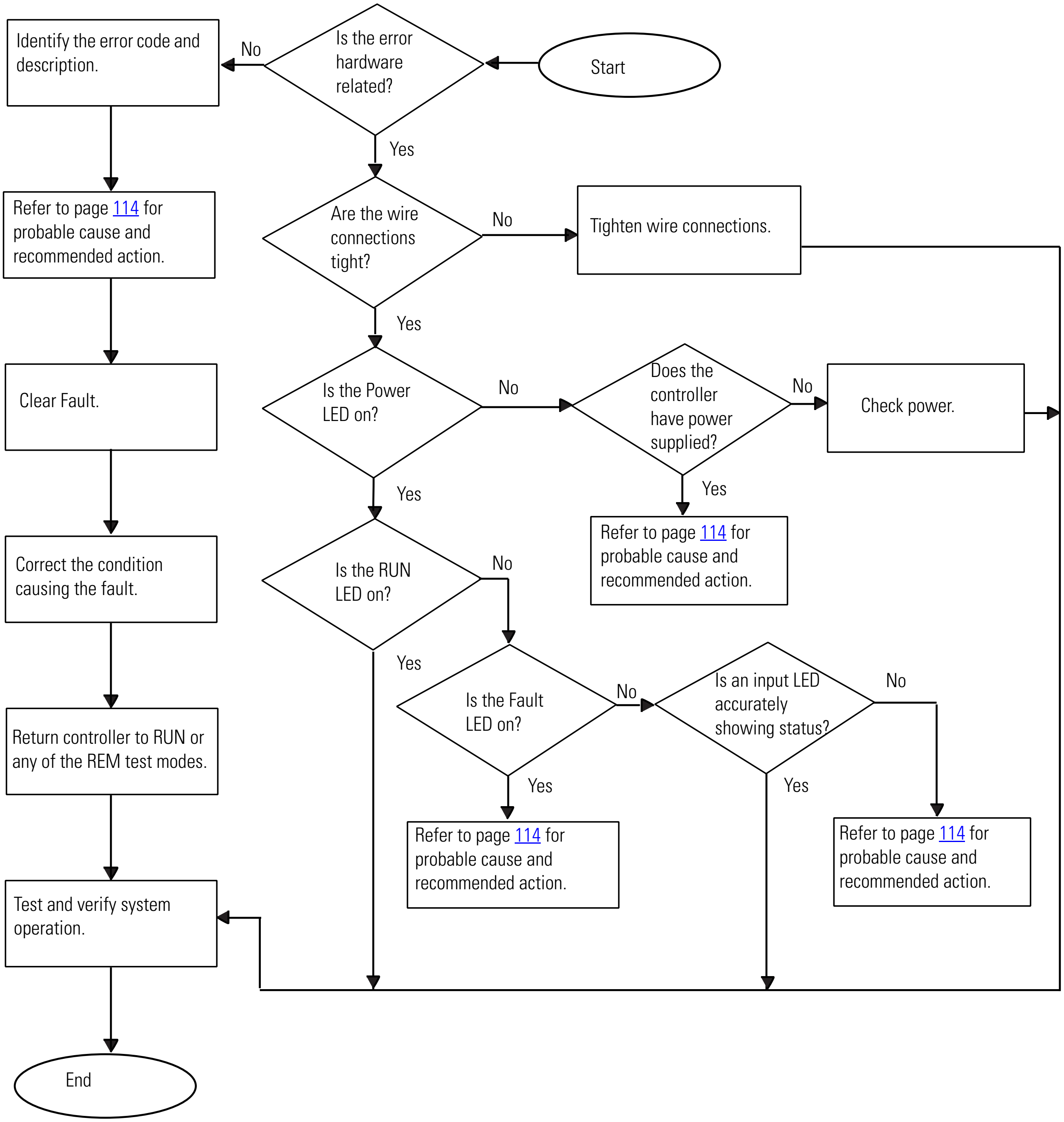
Error Code	Description	Recommended Action
0xF050	The embedded I/O configuration in the user program is invalid.	Perform the following: • Correct the embedded I/O configuration in the user program to match that of the actual hardware configuration. • Build and download the program using Connected Components Workbench. • Put the Micro800 controller into Run mode. • If the error persists, be sure to use Connected Components Workbench programming software to develop and download the program.
For the following four error codes, z is the slot number of the plug-in module. If z = 0, then the slot number cannot be identified		
0xF0Az	The plug-in I/O module experienced an error during operation.	Perform one of the following: • Check the condition and operation of the plug-in I/O module. • Cycle power to the Micro800 controller. • If the error persists, see the Micro800 Plug-in Modules User Manual, publication 2080-UM004.
0xF0Bz	The plug-in I/O module configuration does not match the actual I/O configuration detected.	Perform one of the following: • 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. • Cycle power to the Micro800 controller. • Replace the plug-in I/O module. • If the error persists, see the Micro800 Plug-in Modules User Manual, publication 2080-UM004.
0xF0Dz	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: • 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	The plug-in I/O module configuration does not match the actual I/O configuration detected.	Perform the following: • 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.
0xD011	The program scan time exceeded the watchdog timeout value.	Perform one of the following: • Determine if the program is caught in a loop and correct the problem. • 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.
0xF850	An error occurred in the STI configuration.	Review and change the STI configuration in the Micro800 controller properties.

List of Error Codes for Micro800 Controllers

Error Code	Description	Recommended Action
0xF860	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: • 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	An index address was out of data space.	Perform the following: • Correct the program to ensure that there is no index address out of data space. • Build and download the program using Connected Components Workbench. • Put the Micro800 controller into Run mode.
0xF880	A data conversion error occurred.	Perform the following: Correct the program to ensure that there is no data conversion error. • Build and download the program using Connected Components Workbench. • Put the Micro800 controller into Run mode.
0xF888	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.	Change the project to reduce the quantity of blocks being called within a block.
0xF898	An error occurred in the user interrupt configuration for the plug-in I/O module.	Correct the user interrupt configuration for plug-in I/O module in the user program to match that of the actual hardware configuration.
0xF8A0	The TOW parameters are invalid.	Perform the following: • Correct the program to ensure that there are no invalid parameters. • Build and download the program using Connected Components Workbench. • Put the Micro800 controller into Run mode.
0xF8A1	The DOY parameters are invalid.	Perform the following: • Correct the program to ensure that there are no invalid parameters. • Build and download the program using Connected Components Workbench. • Put the Micro800 controller into Run mode.
0xFFzz (Note: 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.)	A user-created fault from Connected Components Workbench has occurred.	Contact your local Rockwell Automation technical support representative if the error persists.

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.


Start
Is the error hardware related?
No
Identify the error code and description.
Refer to page 114 for probable cause and recommended action.
Clear Fault.
Correct the condition causing the fault.
Return controller to RUN or any of the REM test modes.
Test and verify system operation.
End
Yes
Are the wire connections tight?
No
Tighten wire connections.
Yes
Is the Power LED on?
No
Does the controller have power supplied?
No
Check power.
Yes
Refer to page 114 for probable cause and recommended action.
Yes
Is the RUN LED on?
No
Yes
Is the Fault LED on?
No
Is an input LED accurately showing status?
No
Refer to page 114 for probable cause and recommended action.
Yes
Yes
Refer to page 114 for probable cause and recommended action.

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

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	121
Configure Controller Password	128
Forcing I/Os	132

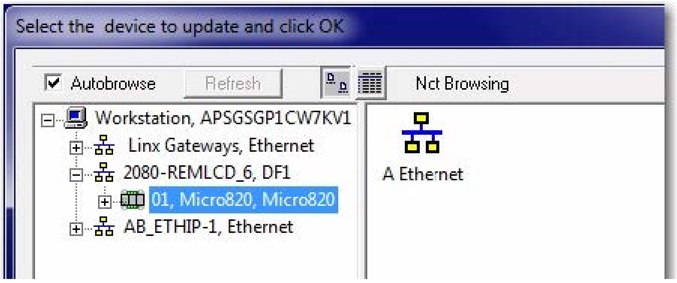
Flash Upgrade Your Micro800 Firmware

This quick start will show you how to flash update the firmware in a Micro800 controller using ControlFLASH. ControlFLASH is installed or updated with the latest Micro800 firmware when Connected Components Workbench software is installed on your computer.

ATTENTION: All Ethernet settings are reverted to factory default after a ControlFlash firmware upgrade. For users who need to use the same static IP address as previously set, for example, use the Memory Module to store project settings prior to a flash upgrade so that you can have the option to restore your original Ethernet settings.

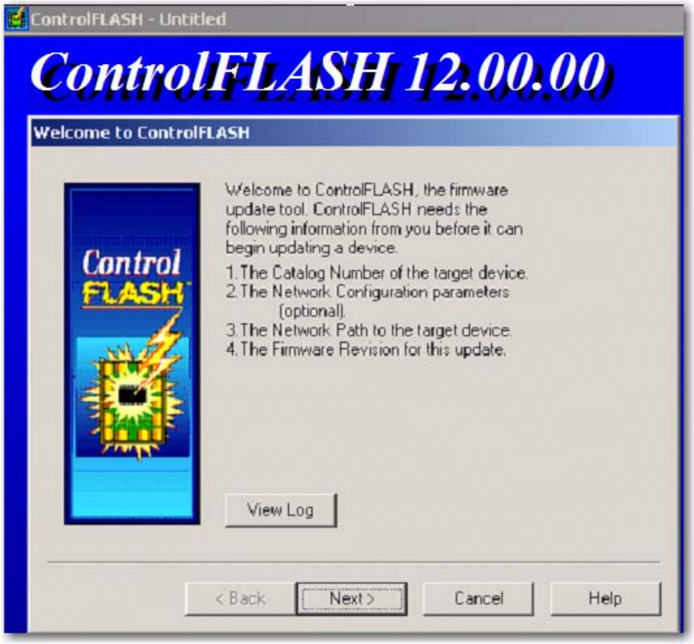
On Micro820 controllers, users can use flash upgrade their controllers through the Ethernet port, in addition to the USB port of the 2080-REMLCD.

1. **Through USB**: Verify successful RSLinx Classic communications with your Micro800 controller by USB using RSWho. Micro810 12-pt. controller uses the 12PtM810_xxxxx driver, the Micro820 uses the 2080_REMLCD_xxxx driver, and the Micro830/Micro850 uses the AB_VBP-x driver.



Select the device to update and click OK

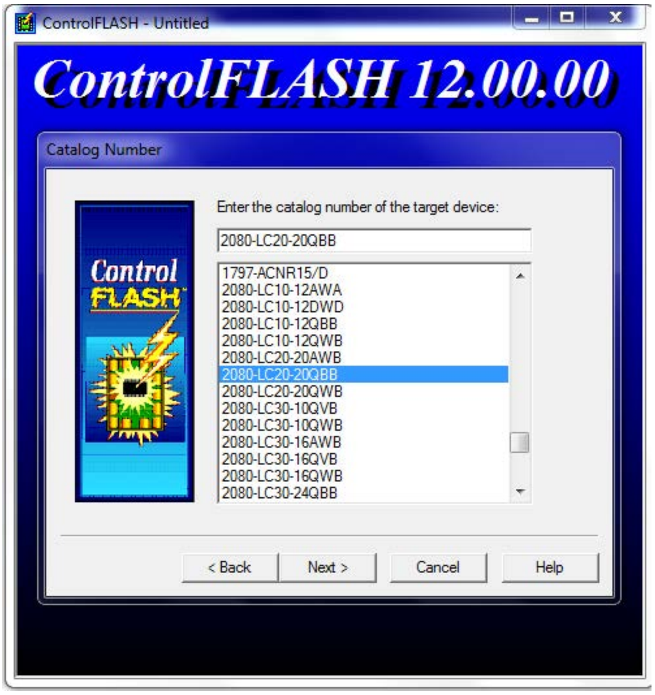
2. Start ControlFLASH and click Next.



Welcome to ControlFLASH, the firmware update tool. ControlFLASH needs the following information from you before it can begin updating a device.

1. The Catalog Number of the target device.
2. The Network Configuration parameters (optional).
3. The Network Path to the target device.
4. The Firmware Revision for this update.

3. Select the catalog number of the Micro800 controller that you are updating and click Next.

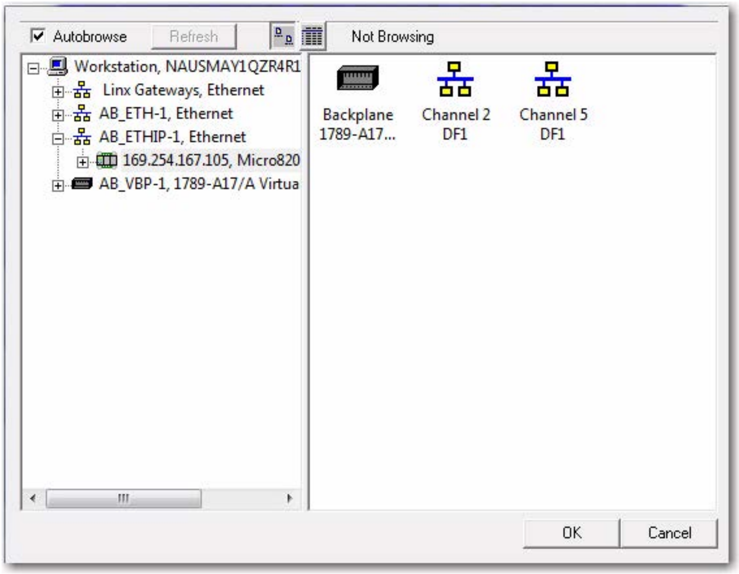


Enter the catalog number of the target device:

2080-LC20-20QBB

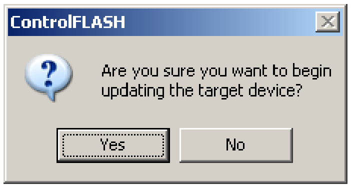
1797-ACNR15/D
2080-LC10-12AWA
2080-LC10-12DWD
2080-LC10-12QBB
2080-LC10-12QWB
2080-LC20-20AWB
2080-LC20-20QBB
2080-LC20-20QWB
2080-LC30-10QVB
2080-LC30-10QWB
2080-LC30-16AWB
2080-LC30-16QVB
2080-LC30-16QWB
2080-LC30-24QBB

4. Select the controller in the browse window and click OK.



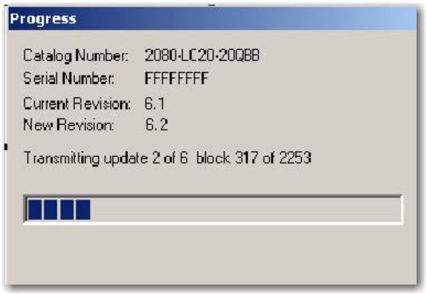
5. Click Next to continue, and verify the revision. Click Finish.

6. Click Yes to initiate the update.



Are you sure you want to begin updating the target device?

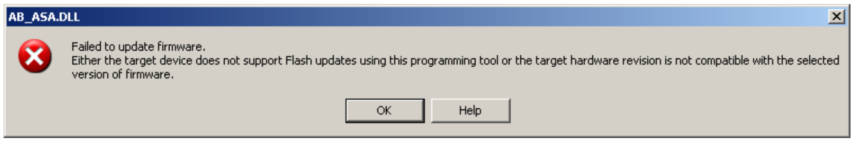
A screen showing the download progress appears.



Catalog Number: 2080-LC20-20QBB
Serial Number: FFFFFFFF
Current Revision: 6.1
New Revision: 6.2

Transmitting update 2 of 6 block 317 of 2253

If you see the following error message instead, check to see if the controller is faulted or in Run mode. If so, clear the fault or switch to Program mode, click OK and try again.



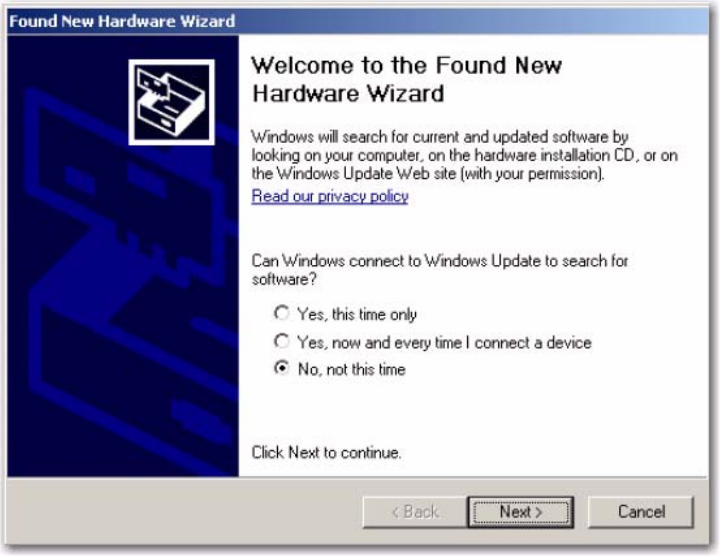
Failed to update firmware.
Either the target device does not support Flash updates using this programming tool or the target hardware revision is not compatible with the selected version of firmware.

7. When the flash update is complete, you see a status screen similar to the following. Click OK to complete the update.

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 a USB.

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



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

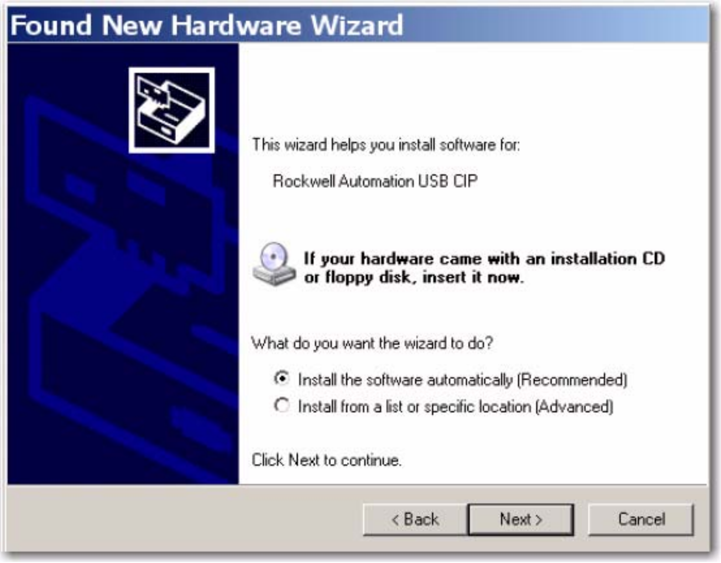
Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.

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



Found New Hardware Wizard

This wizard helps you install software for:

Rockwell Automation USB CIP

If your hardware came with an installation CD or floppy disk, insert it now.

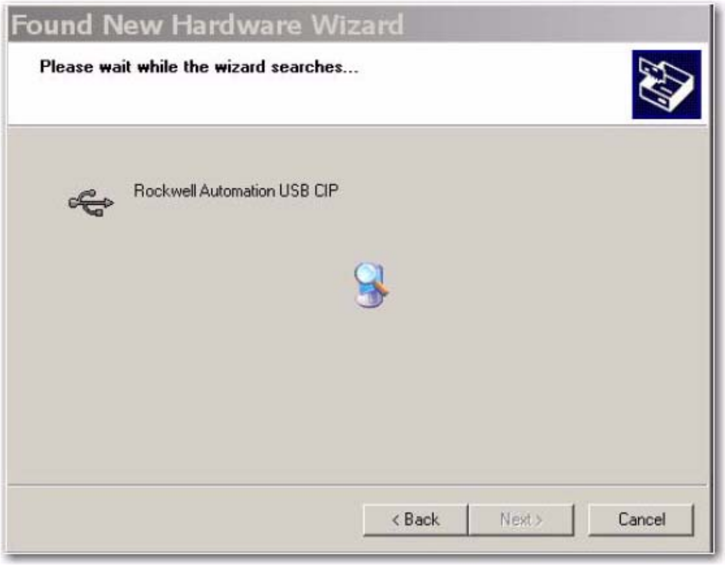
What do you want the wizard to do?

● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.

The Wizard searches for new hardware.

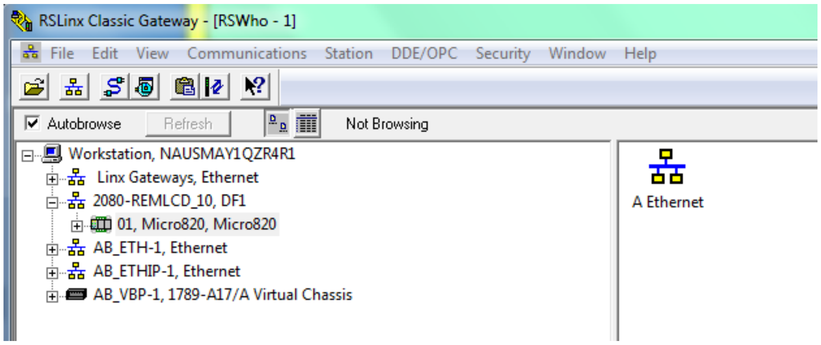


Found New Hardware Wizard

Please wait while the wizard searches...

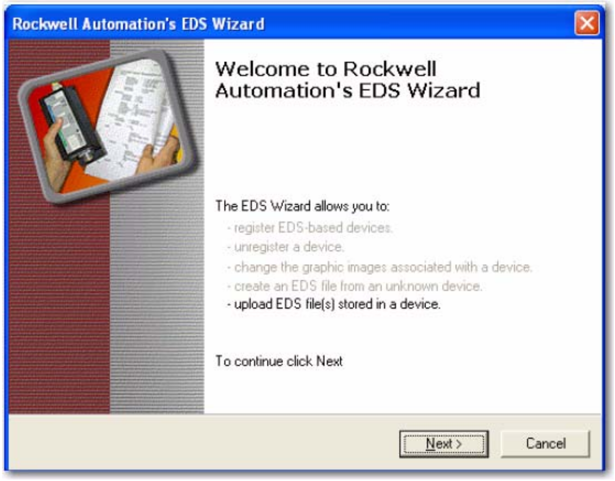
Rockwell Automation USB CIP

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


RSLinx Classic Gateway - [RSWho - 1]
File Edit View Communications Station DDE/OPC Security Window Help
☑ Autobrowse
Refresh
Not Browsing
A Ethernet

Workstation, NAUSMAY1QZR4R1
- Linx Gateways, Ethernet
- 2080-REMLCD_10, DF1
  - 01, Micro820, Micro820
- AB_ETH-1, Ethernet
- AB_ETHIP-1, Ethernet
- AB_VBP-1, 1789-A17/A Virtual Chassis

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

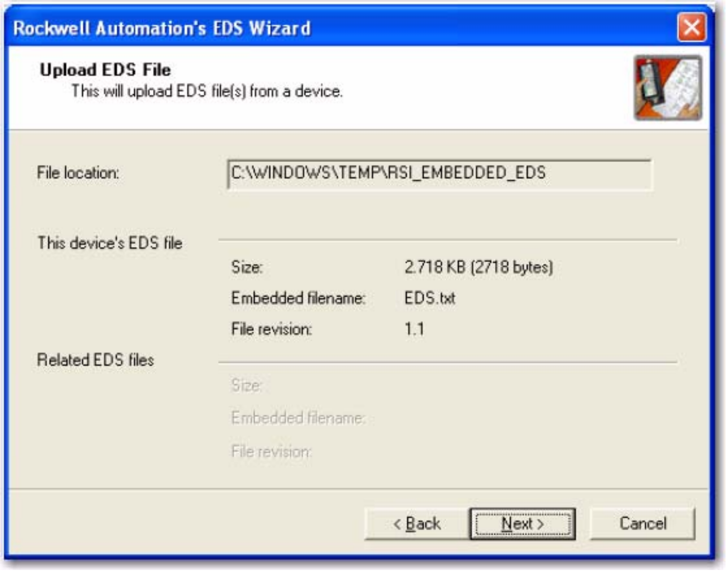


Welcome to Rockwell Automation's EDS Wizard

The EDS Wizard allows you to:
- register EDS-based devices.
- unregister a device.
- change the graphic images associated with a device.
- create an EDS file from an unknown device.
- upload EDS file(s) stored in a device.

To continue click Next

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



Upload EDS File
This will upload EDS file(s) from a device.

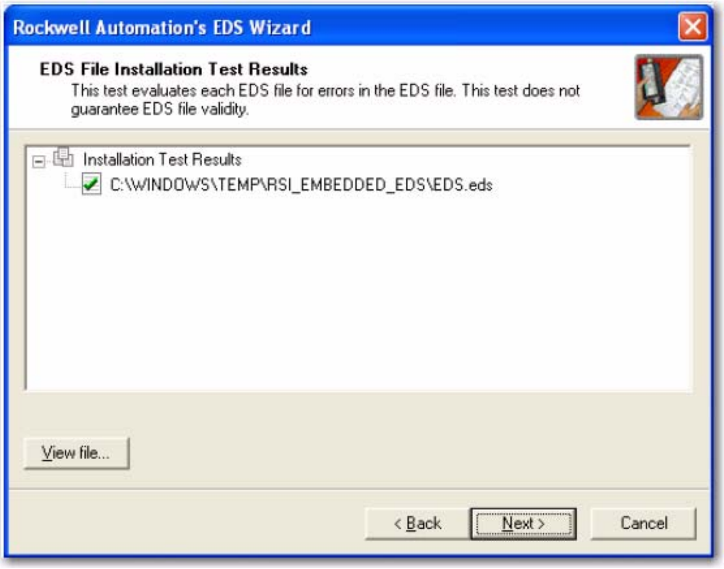
File location: C:\WINDOWS\TEMP\RSI_EMBEDDED_EDS

This device's EDS file

Size: 2.718 KB (2718 bytes)
Embedded filename: EDS.txt
File revision: 1.1

Related EDS files

Size:
Embedded filename:
File revision:

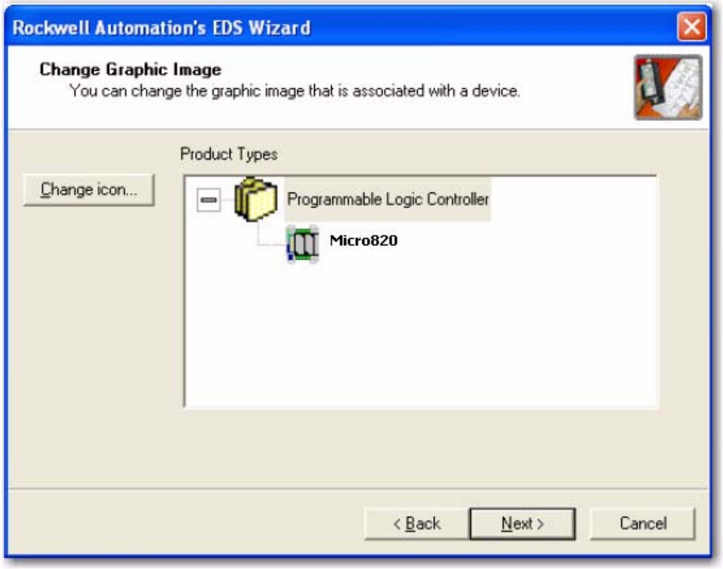


Rockwell Automation's EDS Wizard

EDS File Installation Test Results

This test evaluates each EDS file for errors in the EDS file. This test does not guarantee EDS file validity.

Installation Test Results
☑ C:\WINDOWS\TEMP\RSI_EMBEDDED_EDS\EDS.eds



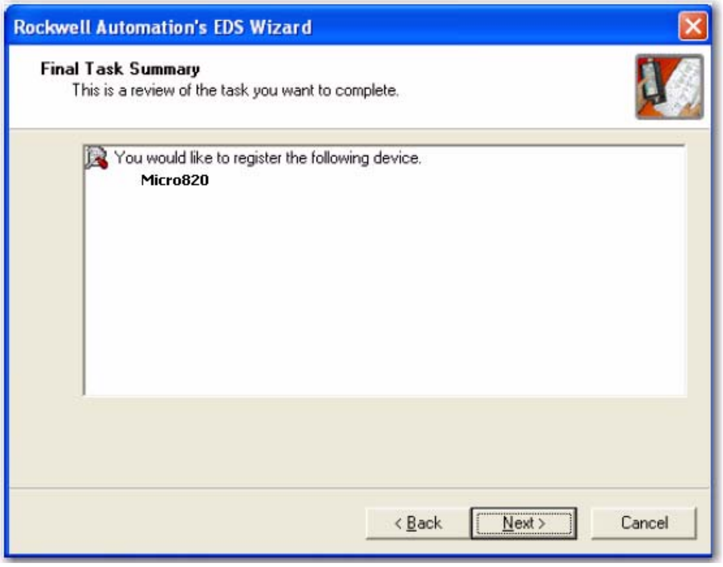
Rockwell Automation's EDS Wizard

Change Graphic Image

You can change the graphic image that is associated with a device.

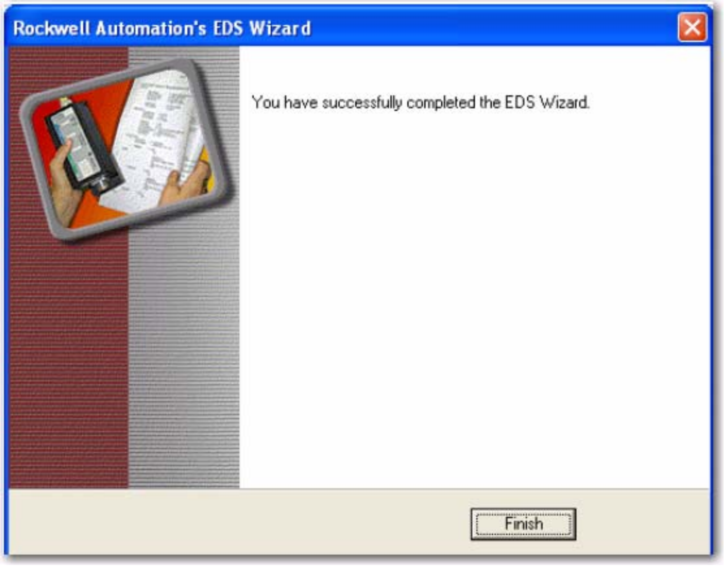
Product Types

Programmable Logic Controller
Micro820



Final Task Summary
This is a review of the task you want to complete.

You would like to register the following device.
Micro820



You have successfully completed the EDS Wizard.

9. Click Finish to complete.

Configure Controller Password

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

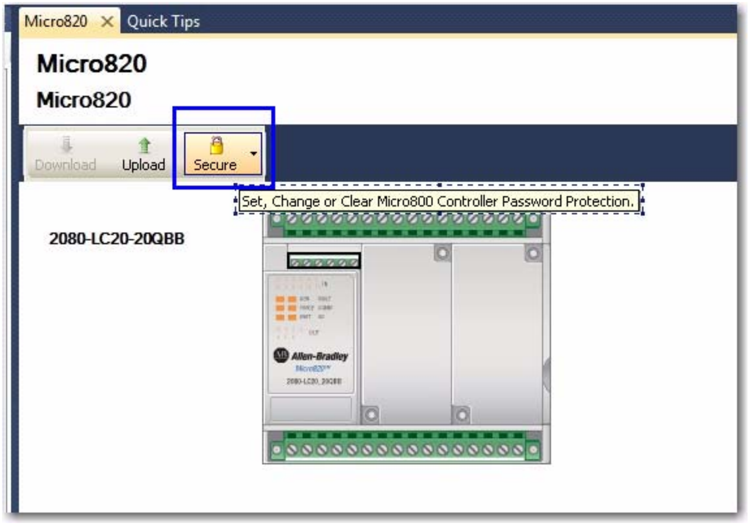
IMPORTANT	The following instructions are supported on Connected Components Workbench revision 2 and Micro800 controllers with firmware revision 2. For more information about the controller password feature on Micro800 controllers, see Controller Security on page 57.

Set Controller Password

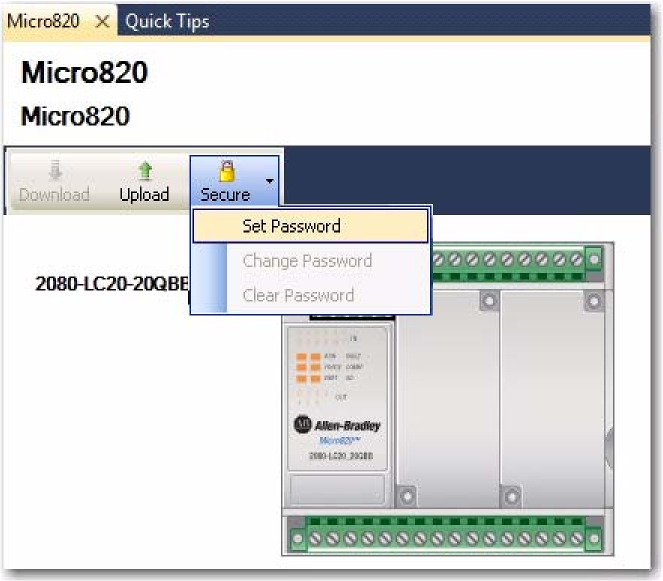
IMPORTANT	After creating or changing the controller password, you need to power down the controller in order for the password to be saved.

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

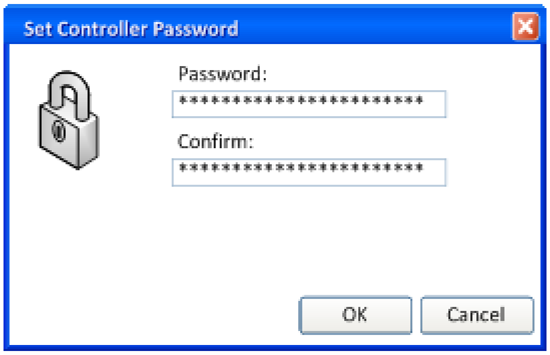
1. On the Connected Components Workbench software, open the project for the target controller.
2. Click Connect to connect to the target controller.
   On the Device Details toolbar, roll over the Secure button. The tooltip message "Set, Change, or Clear Micro800 Controller Password Protection" is displayed.



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.



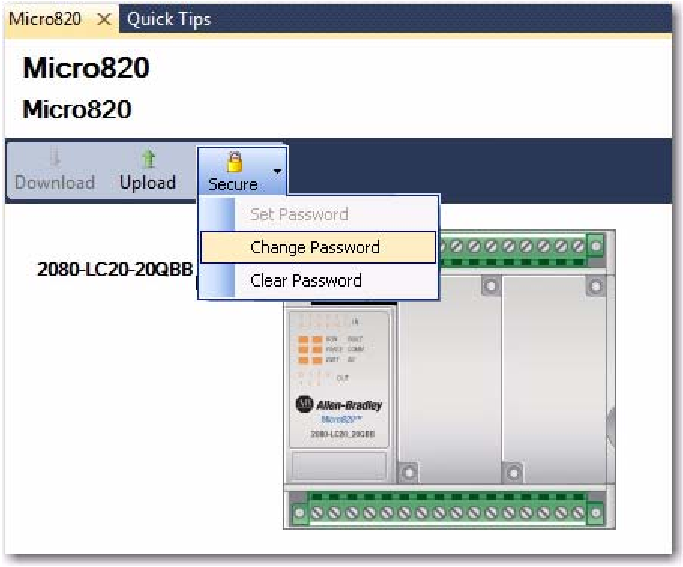
TIP Passwords must have at least eight characters to be valid.

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.

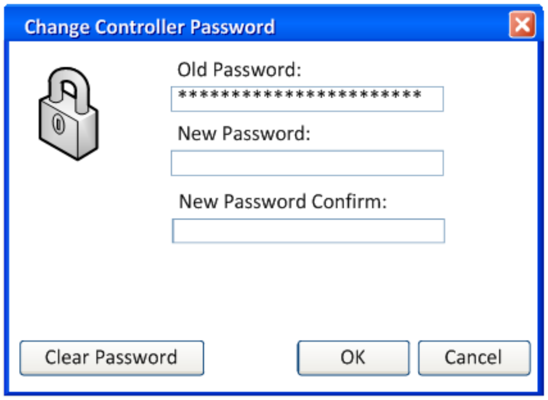
Change Password

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.



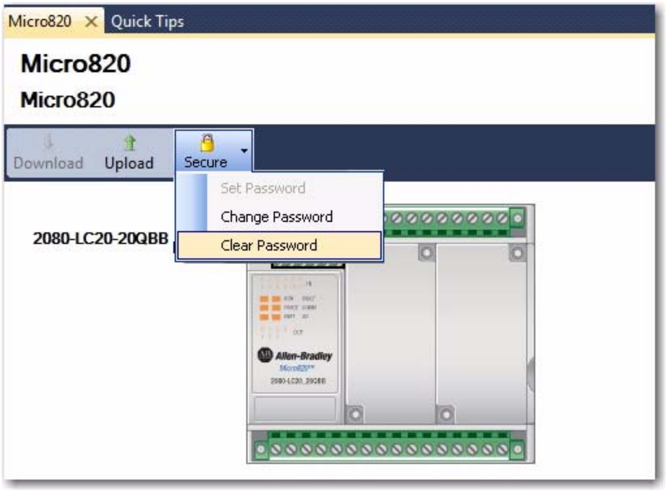
3. Click OK.

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

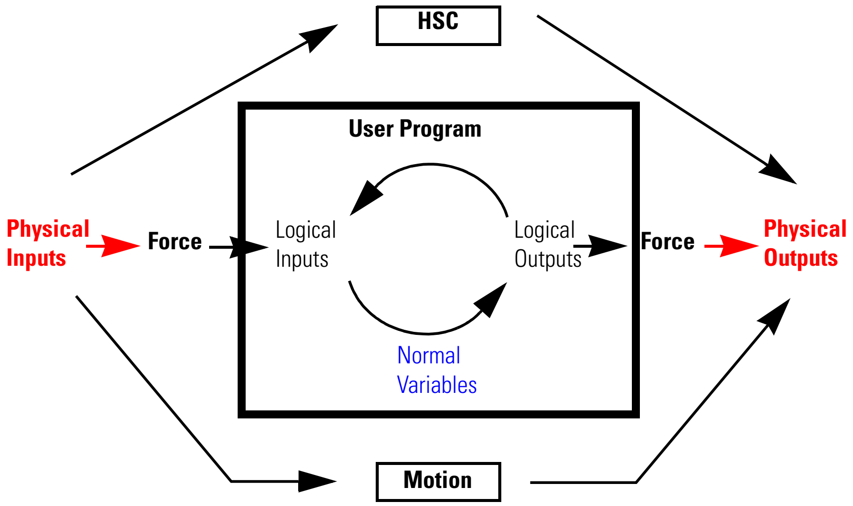
IMPORTANT This section generally talks about forcing I/O in Micro800 controllers. Some elements may not apply to certain models (for example, Micro810 and Micro820 controllers do not support PTO motion).

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.

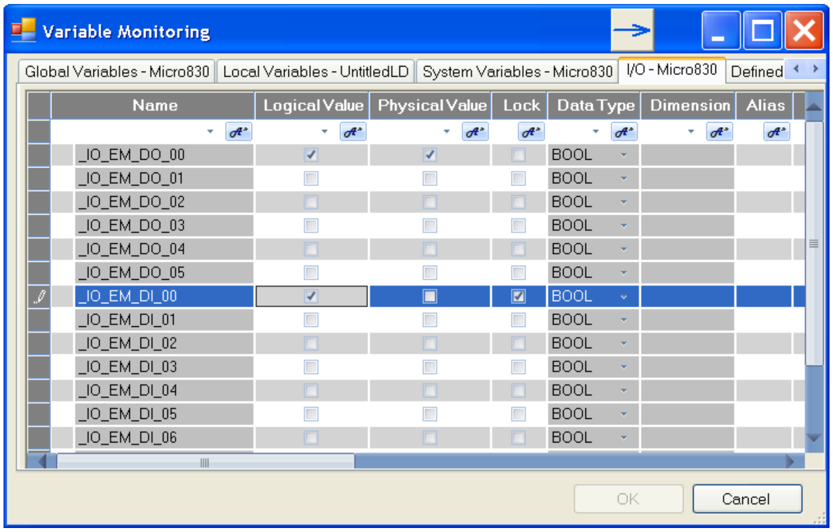


- LED status indicators always match the physical value of I/O
- Normal, non-physical internal variables cannot be forced
- Special functions such as HSC and Motion cannot be forced

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.


Variable Monitoring
Global Variables - Micro830
Local Variables - UntitledLD
System Variables - Micro830
I/O - Micro830
Defined
OK
Cancel

Name	Logical Value	Physical Value	Lock	Data Type	Dimension	Alias
_IO_EM_DO_00	☑	☑	□	BOOL		
_IO_EM_DO_01	□	□	□	BOOL		
_IO_EM_DO_02	□	□	□	BOOL		
_IO_EM_DO_03	□	□	□	BOOL		
_IO_EM_DO_04	□	□	□	BOOL		
_IO_EM_DO_05	□	□	□	BOOL		
_IO_EM_DI_00	☑	□	☑	BOOL		
_IO_EM_DI_01	□	□	□	BOOL		
_IO_EM_DI_02	□	□	□	BOOL		
_IO_EM_DI_03	□	□	□	BOOL		
_IO_EM_DI_04	□	□	□	BOOL		
_IO_EM_DI_05	□	□	□	BOOL		
_IO_EM_DI_06	□	□	□	BOOL		

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.

```
(* Read System Information including Force Enable bit *)
SYS_INFO_1(TRUE);

(* Turn on Warning Light if Forces are Enabled *)
If SYS_INFO_1.Sts.ForcesInstall = TRUE THEN
    _IO_EM_DO_05 := TRUE;
ELSE
    _IO_EM_DO_05 := FALSE;
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

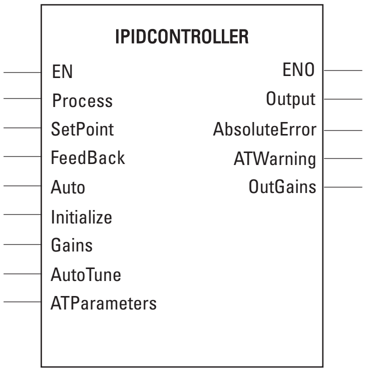
After a controller is power cycled, all I/O forces are cleared from memory.

Notes:

Appendix D

IPID 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 When EN = TRUE, execute function. When EN = FALSE, do not execute function. Only applicable to LD, EN is not required in FBD programming.
Process	Input	REAL	Process value, measured from the output of controlled process.
SetPoint	Input	REAL	Set point value for desired process
Feedback	Input	REAL	Feedback signal, measured from control input to a process.
Auto	Input	BOOL	Operating modes of PID controller: • TRUE —controller runs in normal mode • FALSE — controller out value equals to feedback value
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 for IPIDCONTROLLER See GAIN_PID Data type

IPIDCONTROLLER Arguments

Parameter	Parameter Type	Data Type	Description
AutoTune	Input	BOOL	Start AutoTune sequence
ATParameters	Input	AT_Param	Autotune parameters See AT_Param Data Type
Output	Output	Real	Output value from the controller
AbsoluteError	Output	Real	AbsoluteError is the difference between Process value and set point value
ATWarnings	Output	DINT	Warning for the Auto Tune sequence. Possible value are: • 0 — No auto tune done • 1 — Auto tuning in progress • 2 — Auto tuning done • -1 — Error 1: Controller input "Auto" is TRUE, please set it to False • -2 — Error 2: Auto tune error, the ATDynaSet time expired
OutGains	Output	GAIN_PID	Gains calculated from AutoTune Sequences. See GAIN PID Data type
ENO	Output	BOOL	Enable out. Only applicable to LD, "ENO" is not required in FBD programming.

GAIN_PID Data Type

Parameter	Type	Description
DirectActing	BOOL	Types of acting: • TRUE – Direct acting • FALSE – Reverse acting
ProportionalGain	REAL	Proportional gain for PID (>= 0.0001)
TimeIntegral	REAL	Time integral value for PID (>= 0.0001)
TimeDerivative	REAL	Time derivative value for PID (>= 0.0)
DerivativeGain	REAL	Derivative gain for PID (>= 0.0)

AT_Param Data Type

Parameter	Type	Description
Load	REAL	Initial controller value for autotuning process.
Deviation	REAL	Deviation for auto tuning. This is the standard deviation used to evaluate the noise band needed for AutoTune (noise band = 3* Deviation)[1]

AT_Param Data Type

Parameter	Type	Description
Step	REAL	Step value for AutoTune. Must be greater than noise band and less than ½ load.
ATDynamSet	REAL	Auto Tune time. Set the time to wait for stabilization after the step test (in seconds). Auto Tune process will be stopped when ATDynamSet time expires.
ATReset	BOOL	Determines whether the output value is reset to zero after an AutoTune sequence: • True – Reset IPIDCONTROLLER output to zero after Auto tune process. • 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	Typically set to 0.1 or 0.0
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:

$$\text{For °C: } \frac{22.5 - 21.7}{2} = 0.4 \qquad \text{For °F: } \frac{72.5 - 71}{2} = 0.75$$

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

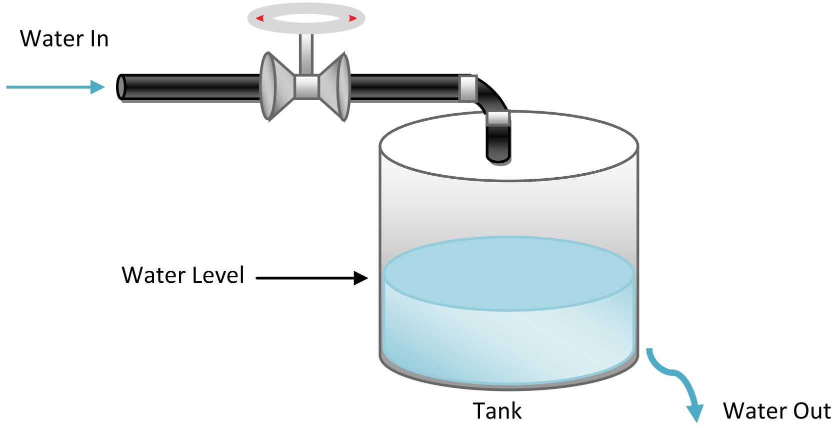
Output Sequence 3: 50 -> 70 -> 30 -> 50

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.

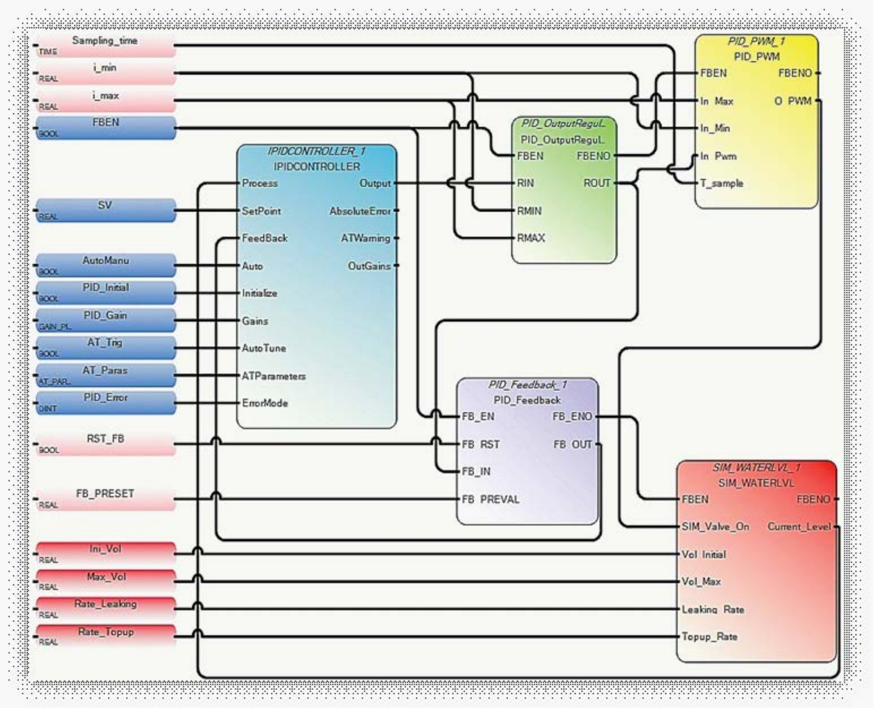
A first order system can be described by a single independent energy storage element. Examples of first order systems are the cooling of a fluid tank, the flow of fluid from a tank, a motor with constant torque driving a disk flywheel or an electric RC lead network. The energy storage element for these 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 RMIN ≤ RIN ≤ RMAX, then ROUT = RIN,
  IF RIN < RMIN, then ROUT = RMIN,
  IF RIN > RMAX, then ROUT = RMAX.

- 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

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
UINT	Y	16	Y	16	Y	1	Y	1
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 Discrete Inputs are mapped from 00001…09999, Coils are mapped from 10001…19999, Input Registers are mapped from 30001…39999, and Holding Registers are mapping from 40001…49999.

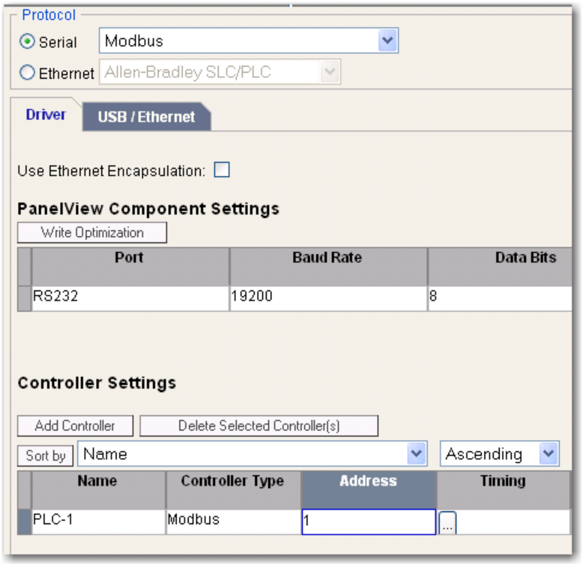
Example 1, PanelView Component HMI (Master) to Micro800 (Slave)

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.



Protocol

● Serial Modbus

○ Ethernet Allen-Bradley SLC/PLC

Use Ethernet Encapsulation: □

PanelView Component Settings

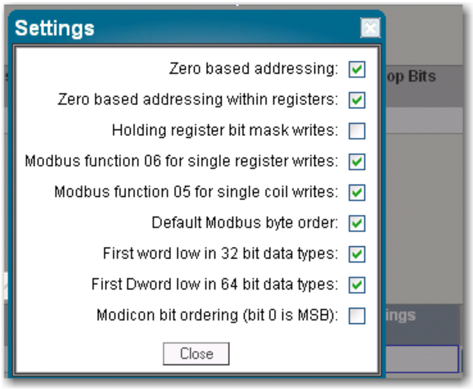
Port	Baud Rate	Data Bits
RS232	19200	8

Controller Settings

Sort by Name Ascending

Name	Controller Type	Address	Timing
PLC-1	Modbus	1	...

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



Settings

Zero based addressing: ☑

Zero based addressing within registers: ☑

Holding register bit mask writes: □

Modbus function 06 for single register writes: ☑

Modbus function 05 for single coil writes: ☑

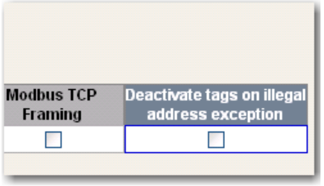
Default Modbus byte order: ☑

First word low in 32 bit data types: ☑

First Dword low in 64 bit data types: ☑

Modicon bit ordering (bit 0 is MSB): □

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.



Modbus TCP Framing	Deactivate tags on illegal address exception
□	□

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.

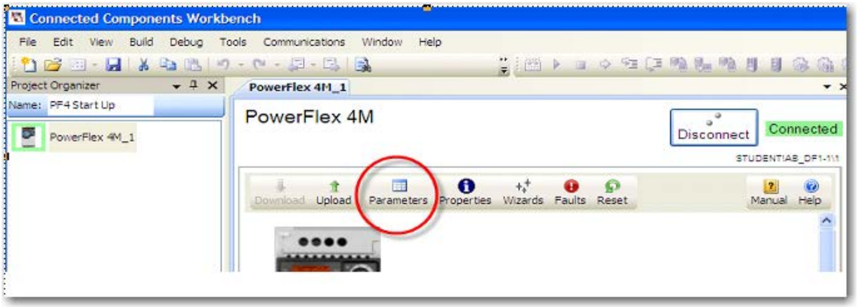
Parameter numbers listed in this section are for a PowerFlex 4M and will be different if you are using another PowerFlex 4-Class drive.

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		

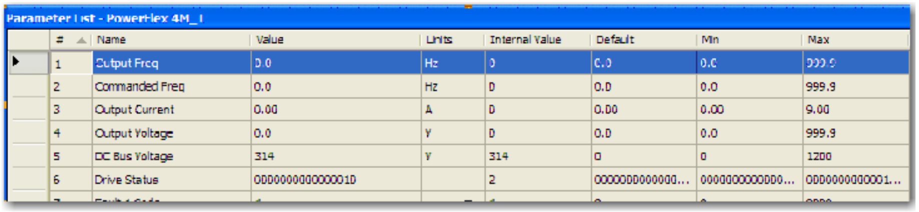
- Connect the 1203-USB to the PowerFlex Drive and to the Computer.
- Launch Connected Components Workbench, Connect to the Drive and set parameters.

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.



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



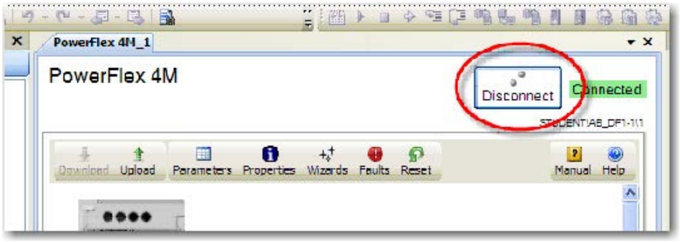
Parameter List - PowerFlex 4M_1

#	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	0000000000000...	0000000000000...	000000000001...

7. 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

8. Disconnect the Communications and save your project.



9. 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.

Numerics

2080-LCD 57
2080-MEMBAK-RTC 1, 57
2080-PS120-240VAC 20
2080-REMLCD 1, 23, 61, 63
- advanced set 67
- analog calibration 68
- backlight parameters 70
- customization 65
- edit variables 68
- function blocks 69
- I/O Status 67
- main menu 66
- memory card 68
- mode switch 67
- modes of operation 64
- operational limit 68
- rating 64
- resolution 63
- security 67
- startup screen 65
- text display mode 64
- USB Mode 64
- user-defined screens 69
- variable 67, 68

A

additional resources iii
analog cable grounding 32
analog channel wiring guidelines 31
analog input
- to transmitters 36
- analog channel wiring guidelines 31

analog output 36
array of strings 39
arrays 39
ASCII 37, 40, 42
- configuration 47

ASCII devices 40
autotune 137

B

backup 1, 3
baud rate 3, 44
before calling for assistance 120

C

calibration 35
calling for assistance 120
CE mark 7, 8
certifications 7
CIP Client Messaging 39
CIP communications pass-thru 40
CIP Generic 39
CIP Serial 38, 39, 42
- Parameters 44
- Driver
- configure 43
- parameters 44
- Server 37

CIP Symbolic 38, 39
- supported data types 39
- addressing 39
- Server 38

communication connections 37
communication protocols 37
communications
- ports 37

CompactLogix 38
Compliance to European Union Directive
- EMC Directive 8
- Low Voltage Directive 8

Compliance to European Union Directives 7
ConfigMeFirst.txt 3
- errors 77

Connected Components Workbench iv, 7, 48, 51, 54, 57, 58, 61, 64, 69, 78, 83
ControlFlash 61
controller
- Error Recovery Model 119
- grounding 29
- I/O wiring 30
- load 52
- lock 58
- minimizing electrical noise 31
- preventing excessive heat 14

password 57
ControlLogix 38

D

datalog 1, 78
- data types 82
- directory structure 79
- execution rules 82
- specifications 79
- timing diagram 81

datasets 78, 79
DF1 mode 44
DHCP Client 37
DIN rail mounting 20
disconnecting main power 11

DLG
function block status 80
function error ID list 80
input and output parameters 80
duplicate packet detection 44

E

embedded responses 45
Embedded Serial Port Cables 4
EMC Directive 8
Endian Configuration 143
ENET status 2
ENQ retries 45
error codes 113, 114
error conditions 112
error detection 44
error recovery model 119
Ethernet 38, 65
configuration settings 48
transfer rate 37
EtherNet/IP Client 39
EtherNet/IP Server 37, 39
European Union Directive compliance 7
EMC Directive 8
exclusive access 57

F

fault status 2
Force status 2, 111
Forcing I/Os 132

G

general considerations 8
grounding the controller 29
guidelines and limitations for advanced users 54

H

hardware
features 1
overview 1
heat protection 14
housekeeping 51, 78

I

I/O Status 64, 66
input states on power down 14
inputs
embedded 1
installation
considerations 8
DIN rail mounting 20
Micro820 controller 19
module spacing 20
mounting dimensions 19
panel mounting 20
dimensions 21
IPID Function Block 135
IPIDCONTROLLER 135
parameters 135
isolation transformers
power considerations 13

K

KEY_READ_REM 69, 71
keyswitch 61

L

LCD_BKLT_REM 69, 70
LCD_REM 69, 70
LINT 39
Literature Library 23
LREAL 39

M

master control relay 14
emergency-stop switches 15
using ANSI/CSA symbols schematic 18
using IEC symbols schematic 17
periodic tests 12
Micro800 cycle or scan 51
microSD 2, 3, 57
card slot 3
Class 4 3
Class 6 and 10 SDSC and SDHC 3
installation 22
minimizing electrical noise 31
Modbus 37
mapping 143
Modbus RTU 37, 38, 42
advanced parameters 46
configuration 45
parameters 46
Modbus TCP Client 38
Modbus/TCP Server 37, 38
module spacing 20

motor starters (bulletin 509)
surge suppressors 28
MSG_CIPGENERIC 39
MSG_CIPSYMBOLIC 39
MSG_MODBUS 38
MSG_MODBUS2 38

N

NAK retries 45
normal operation 112
North American Hazardous Location Approval 11

O

outputs
embedded 1

P

panel mounting 20
PanelView Component 38
password 57
Performance, MSG_MODBUS 148
PID Application Example 140
PID Code Sample 141
POU (Program Organizational Unit) 52
power considerations
input states on power down 14
isolation transformers 13
loss of power source 13
other line conditions 14
overview 12
power supply inrush 13
power supply 2
requirement 2
power supply inrush
power considerations 13
powerup settings 3
preventing excessive heat 14
program scan cycle 52, 78

Q

Quickstarts 121

R

REAL 39
recipe 1, 83
data types 82
directory structure 84
function block errors 85
function block parameters 84
function block status 85
specifications 83
recipe sets 83
Remote LCD
configuration 42
resistance input 35
restore 1, 3
RJ-45 connector
location 22
Ethernet cable 4
ethernet port 4, 37
RS-232/485 combo port 37
RS232/RS485 2, 3, 4, 29
pin definition 4
RSLinx 38
run status 2

S

safety circuits 12
safety considerations 10
disconnecting main power 11
hazardous location 11
master control relay circuit
periodic tests 12
periodic tests of master control relay circuit 12
power distribution 12
safety circuits 12
security 57
serial communications status 111
serial port
cables 3
configure 42
station address 44
status indicator 2
Comm 2
fault status 111
input status 2, 111
network status 112
output 2
run status 111
SD 2
serial communications 111
surge suppression 26

surge suppressors
for motor starters 28
recommended 28
using 26
__SYSVA_CYCLECNT 51
__SYSVA_TCYCURRENT 52
__SYSVA_TCYMAXIMUM 52

T

terminal blocks 2
thermistor resistance 35
transmit retries 45
troubleshooting 111

U

USB mode 64
USB port 64, 65
User Defined Function Block (UDFB) 51, 55
using emergency-stop switches 15

V

variable retention 54
voltage input 35

W

wiring 25
fixed terminal blocks 26
removable terminal blocks 26
RS232/RS485 terminal block 26
wiring recommendation 25

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