Micro800 Plug-in Modules

Catalog Numbers 2080-IQ4, 2080-IQ4OB4, 2080-IQ4OV4, 2080-OB4, 2080-OV4, 2080-OW4I, 2080-IF2, 2080-IF4, 2080-OF2, 2080-TC2, 2080-RTD2, 2080-MEMBAK-RTC, 2080-MEMBAK-RTC2, 2080-TRIMPOT6, 2080-SERIALISOL, 2080-DNET20, 2080-MOT-HSC

Allen-Bradley
by ROCKWELL AUTOMATION

User Manual

Original Instructions

Important User Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The following icon may appear in the text of this document.

Identifies information that is useful and can help to make a process easier to do or easier to understand.

Table of Contents

Preface

About This Publication 7
Download Firmware, AOP, EDS, and Other Files 7
Summary of Changes 7
Additional Resources 7

Chapter 1

Micro800 Plug-in Modules

Digital Plug-ins 10
- 12/24V Digital Plug-ins – 2080-IQ4, 2080-IQ4OB4, 2080-IQ4OV4, 2080-OB4, 2080-OV4 10
- AC/DC Relay Output Module – 2080-OW4I 10

Analog Plug-ins 10
- Non-isolated Unipolar Analog Input and Output – 2080-IF2, 2080-IF4, 2080-OF2 10

Specialty Plug-ins 10
- Non-isolated Thermocouple and RTD – 2080-TC2 and 2080-RTD2 10
- Memory Backup and High Accuracy RTC – 2080-MEMBAK-RTC and 2080-MEMBAK-RTC2 10
- Six-channel Trimpot – 2080-TRIMPOT6 11
- High-speed Counter – 2080-MOT-HSC 11

Communication Plug-ins 11
- RS232/RS485 Isolated Serial Port – 2080-SERIALISOL 11
- DeviceNet Scanner – 2080-DNET20 12

Chapter 2

Install and Wire Your Module

Hardware Features 13
Insert Module into Controller 13
Wiring 14
Wiring Considerations and Applications for 2080-TC2 17
- Type of CJC Sensor 18
- Wire the CJC Thermistor on the 2080-TC2 Module 18

Wiring Considerations and Applications for 2080-RTD2 19
- Wire the RTD Sensors 19
- Wire the RTD Module and RTD Sensor in the Field 20

Wiring Applications for 2080-MOT-HSC 21

Chapter 3

Non-isolated Thermocouple and RTD Plug-in Modules – 2080-TC2 and 2080-RTD2

Thermocouple Module 23
- Thermocouple Sensor Types and Ranges 23

RTD Module 24
- RTD Sensor Types and Ranges 24

Connected Components Workbench Global Variables Data Maps 25
- Temperature Conversion – Data to Degree Celsius (°C) 26

Chapter 4

High-speed Counter – 2080-MOT-HSC

Overview 27
Differences Between Embedded HSC and Plug-in Module 27
Counter Specifications 28
- Number of Counters: 1 to 2 28
- Up Counter 29
- Counter with External Direction 30
- Understanding Rates 35

User Defined Function Blocks 36
- RA_HSCPlugIn 36
- RA_EncoderFDBK 37
- RA_ServoFDBK 38
- Use the 2080-MOT-HSC Module 38

Chapter 5

DeviceNet Plug-in – 2080-DNET20

Overview 39
Status Indicators 39
Network Configuration 40
- Network Wiring 40
- DeviceNet Switches 40
- Power Supply 41

User-defined Function Blocks 43
- RA_DNET_MASTER 44
- RA_DNET_NODE_STATUS 45
- RA_DNET_LDX_DISCRETE 45
- RA_DNET_LDX_ANALOG 46
- RA_DNET_LDX_TC_RTD 47
- RA_DNET_TOWERLIGHT 47
- RA_PF_DNET_STANDARD 47
- RA_PF_DNET_MULTIDRIVE 48
- RA_DNET_OVERLOAD 49
- RA_DNET_GENERIC 50
- RA_DNET_EXPLICIT 50

Send Explicit Messages to 2080-DNET20 Plug-in Using Micro800 Pass Through 52
Error Codes 53
- Use the 2080-DNET20 Plug-in 53

Appendix A

Quick Start

Add and Configure Plug-ins in Connected Components Workbench Software 55
Browse Your 2080-DNET20 Plug-in Using RSLinx Software 57
- Browse Using the DeviceNet Network 57
- Browse Using the Micro800 Pass Through Feature 58

Update Your 2080-DNET20 Plug-in Firmware 59
Quick Start Project for 2080-DNET20 Plug-in 63
- Setup and Wiring 63
- Configuration 64
- Build and Download 66
- Execute Program 66

Quick start Projects for 2080-MOT-HSC Plug-in 67
Setup and Wiring 67
Configuration for UDFB 1: RA_HSCPlugIn 68
Build and Download 68
Execute the Function Block 69
Configuration for UDFB 2: RA_EncoderFDBK 70
Build and Download 71
Execute the Function Block 71
Configuration for HSC UDFB 3: RA_ServoFDBK 72
Build and Download 73
Execute the Function Block 74

Appendix B

Error Codes

Troubleshooting 75
Error Codes for Micro800 Plug-ins 75
Calling Rockwell Automation for Assistance 75

Index 77

Notes:

About This Publication

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

This manual is a reference guide for Micro800 expansion I/O modules. It describes the procedures that you use to install, wire, and troubleshoot your expansion I/O. This manual:

- Gives you an overview of expansion I/O features and configuration parameter
- Gives you an overview of the Micro800 controller system

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.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Download Firmware, AOP, EDS, and Other Files

Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes from the Product Compatibility and Download Center at rok.auto/pcdc.

Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

Topic	Page
Updated template	throughout
Added Inclusive Language Acknowledgment	7
Updated Additional Resources	7

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation. You can view or download publications at rok.auto/literature.

Additional Resources

Resource	Description
Micro800 Programmable Controller Family Selection Guide, publication 2080-SG001	Provides information to help you select the Micro800 controller, plug-ins, expansion I/O, and accessories, based on your requirements.
Micro800 Programmable Controllers Technical Data, publication 2080-TD001	Provides detailed specifications for Micro800 controllers, expansion I/O modules, plug-in modules, and accessories.
Micro830, Micro850, and Micro870 Programmable Controllers User Manual, publication 2080-UM002	Describes how to install, configure, use, and troubleshoot your Micro830®, Micro850®, and Micro870® controllers.
Micro800 16-point and 32-point 12/24V Sink/Source Input Modules Installation Instructions, publication 2085-IN001	Provides information on mounting and wiring the expansion I/O modules (2085-IQ16, 2085-IQ32T).
Micro800 Bus Terminator Module Installation Instructions, publication 2085-IN002	Provides information on mounting and wiring the expansion I/O bus terminator (2085-ECR).
Micro800 16-point Sink and 16-point Source 12/24V DC Output Modules Installation Instructions, publication 2085-IN003	Provides information on mounting and wiring the expansion I/O modules (2085-OV16, 2085-OB16).
Micro800 8-point and 16-point AC/DC Relay Output Modules Installation Instructions, publication 2085-IN004	Provides information on mounting and wiring the expansion I/O modules (2085-OW8, 2085-OW16).
Micro800 8-point Input and 8-point Output AC Modules Installation Instructions, publication 2085-IN005	Provides information on mounting and wiring the expansion I/O modules (2085-IA8, 2085-IM8, 2085-OA8).
Micro800 4-channel and 8-channel Analog Voltage/Current Input and Output Modules Installation Instructions, publication 2085-IN006	Provides information on mounting and wiring the expansion I/O modules (2085-IF4, 2085-IF8, 2085-OF4).
Micro800 4-channel Thermocouple/RTD Input Module Installation Instructions, publication 2085-IN007	Provides information on mounting and wiring the expansion I/O module (2085-IRT4).
Micro870 Programmable Controllers 24V DC Expansion Power Supply Installation Instructions, publication 2085-IN008	Provides information on mounting and wiring the 24V DC expansion power supply (2085-EP24VDC).

Additional Resources (Continued)

Resource	Description
Safety Guidelines for the Application, Installation, and Maintenance of Solid-state Control, publication SGI-1.1	Designed to harmonize with NEMA Standards Publication No. ICS 1.1-1987 and provides general guidelines for the application, installation, and maintenance of solid-state control in the form of individual devices or packaged assemblies incorporating solid-state components.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Selection and Configuration webpage, rok.auto/systemtools	Helps configure complete, valid catalog numbers and build complete quotes based on detailed product information.
Product Certifications website, rok.auto/certifications	Provides declarations of conformity, certificates, and other certification details.

Micro800 Plug-in Modules

Plug-in modules enhance the functionality of a base unit controller. With these modules, you can:

- Extend the functionality of embedded I/O without increasing the footprint of your controller
- Improve performance by adding additional processing power or capabilities
- Add additional communication functionality

Micro800 controllers support the following plug-in modules:

Table 1 - Micro800 Plug-in Modules

Module	Type	Description
2080-IQ4	Digital	4-point, 12/24V DC sink/source input
2080-IQ4OB4	Digital	8-point, Combo, 12/24V DC sink/source input 12/24V DC Source output
2080-IQ4OV4	Digital	8-point, Combo, 12/24V DC sink/source input 12/24V DC Sink output
2080-OB4	Digital	4-point, 12/24V DC Source output
2080-OV4	Digital	4-point, 12/24V DC Sink output
2080-OW4I	Digital	4-point, AC/DC Relay output
2080-IF2	Analog	2-channel, Non-isolated unipolar voltage/current analog input
2080-IF4	Analog	4-channel, Non-isolated unipolar voltage/current analog input
2080-OF2	Analog	2-channel, Non-isolated unipolar voltage/current analog output
2080-TC2	Specialty	2-channel, non-isolated thermocouple module
2080-RTD2	Specialty	2-channel, non-isolated RTD module
2080-MEMBAK-RTC(1)	Specialty	Memory backup and high accuracy RTC, 1 MB
2080-MEMBAK-RTC2(1)	Specialty	Memory backup and high accuracy RTC, 4 MB
2080-TRIMPOT6	Specialty	6-channel trimpot analog input
2080-MOT-HSC	Specialty	High-speed counter
2080-DNET20	Communication	20-node DeviceNet® scanner
2080-SERIALISOL	Communication	RS-232/RS-485 isolated serial port

(1) 2080-MEMBAK-RTC and 2080-MEMBAK-RTC2 are not supported on Micro820 controllers.
2080-MEMBAK RTC is not supported on Micro870 controllers.

The number of supported Micro800 plug-ins on the controllers are summarized in Table 2.

Table 2 - Plug-in Slots on Micro800 Controllers

Controller	Number of Plug-in Slots
Micro810®	0
Micro820®	2
Micro830	2 (10/16 points) 3 (24 points) 5 (48 points)
Micro850	3 (24 points) 5 (48 points)
Micro870	3

ATTENTION: Removal and Insertion Under Power (RIUP) is not supported on all Micro800 plug-in modules, except on the 2080-MEMBAK-RTC and 2080-MEMBAK-RTC2 modules.

ATTENTION: Micro800 plug-in modules can be installed on any plug-in slot on the controller, except for the 2080-MEMBAK-RTC and 2080-MEMBAK-RTC2 modules, which can only be installed on the leftmost plug-in slot.

Digital Plug-ins

12/24V Digital Plug-ins – 2080-IQ4, 2080-IQ4OB4, 2080-IQ4OV4, 2080-OB4, 2080-OV4

These digital plug-in modules provide transistor outputs for switching various 12/24V DC voltages to field loads and for detecting 12/24V signals from field devices.

AC/DC Relay Output Module – 2080-OW4I

The 2080-OW4I is a 4-channel relay output and provides dry contact relay closure outputs for switching various AC and DC voltages to field loads.

Analog Plug-ins

The following analog plug-ins are supported by most Micro800 controllers.

Non-isolated Unipolar Analog Input and Output – 2080-IF2, 2080-IF4, 2080-OF2

These plug-in modules add extra embedded non-isolated unipolar (0...10V, 0...20 mA) analog I/O and offer 12-bit resolution.

Specialty Plug-ins

Non-isolated Thermocouple and RTD – 2080-TC2 and 2080-RTD2

These non-isolated plug-in modules help to make temperature control possible when used with PID (Proportional Integral Derivative).

See Non-isolated Thermocouple and RTD Plug-in Modules - 2080-TC2 and 2080-RTD2 on page 23 for more information.

Memory Backup and High Accuracy RTC – 2080-MEMBAK-RTC and 2080-MEMBAK-RTC2

These plug-in modules allow you to make a backup copy of the project in your controller, and adds precision real-time clock function without needing to calibrate or update periodically.

They can also be used to clone/update Micro800 application code. The 2080-MEMBAK-RTC2 has larger memory to support clone/update for Micro870 application code. However, these

plug-in modules cannot be used as additional Run-Time Program or Data Storage for recipe and datalog.

Table 3 - Status Indicators

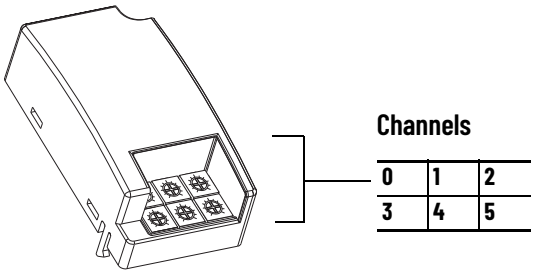
State	Description
Steady red (2 s)	Startup cycle test in progress
Flashing red	Back up in progress
Steady red (continuous)	Battery low

Project Backup and Restore

The project can be backed up and restored using Connected Components Workbench™ software.

Six-channel Trimpot – 2080-TRIMPOT6

This trimpot plug-in offers an affordable method of adding six analog presets for speed, position, and temperature control.



High-speed Counter – 2080-MOT-HSC

This plug-in module provides enhanced high-speed counter capabilities to the Micro800 controller. It supports the same functionalities of an embedded HSC on the Micro800 controllers but is enhanced to support up to 250 kHz 5V differential line driver for improved noise immunity and provides additional dedicated I/O.

For more information, see High-speed Counter – 2080-MOT-HSC on page 27.

Communication Plug-ins

RS232/RS485 Isolated Serial Port – 2080-SERIALISOL

The 2080-SERIALISOL plug-in supports CIP™ Serial (RS-232 only), Modbus RTU (RS-232 and RS-485), and ASCII (RS-232 and RS-485[a]) protocols. Unlike the embedded Micro800 serial port, this port is electrically isolated, making it ideal for connecting to noisy devices, such as variable frequency and servo drives, and for communications over long cable lengths. Depending on the application and baud rate setting, you can extend this length.

IMPORTANT	2080-SERIALISOL is suitable for communication over longer cable length of up to 1000 m (3280 ft) using RS-485, with up to 19,200 bps baud rate. The electrical characteristics of the cable that is used and good wiring practices are critical in achieving reliable communication performance over longer cable length. A shielded twisted-pair RS-485 0.34 mm^2 (22 AWG) cable (example: 3106A from Belden) is recommended. Terminate both ends of the cable with 120 ohm resistance.

(a) RS-485 support is only available from Connected Components Workbench software version 6 or later.

DeviceNet Scanner – 2080-DNET20

The Micro800 DeviceNet plug-in module serves as a scanner and client for explicit messaging to remote devices including I/O and drives, using a proven and well-accepted fieldbus/network. It also provides better performance than using serial and Ethernet (EtherNet/IP™ Class 3) communications.

For more information, see the DeviceNet Plug-in – 2080-DNET20 on page 39.

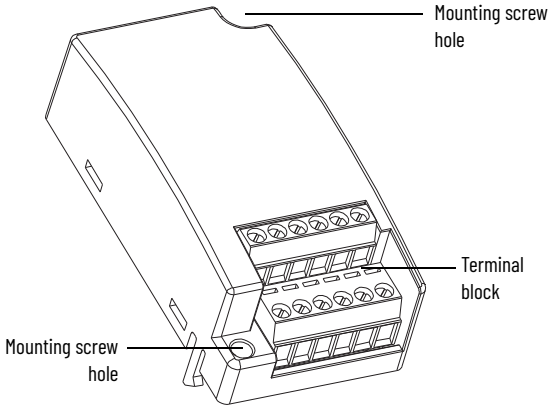
Install and Wire Your Module

This chapter provides hardware features, installation, and wiring connection diagrams for all Micro800 plug-in modules.

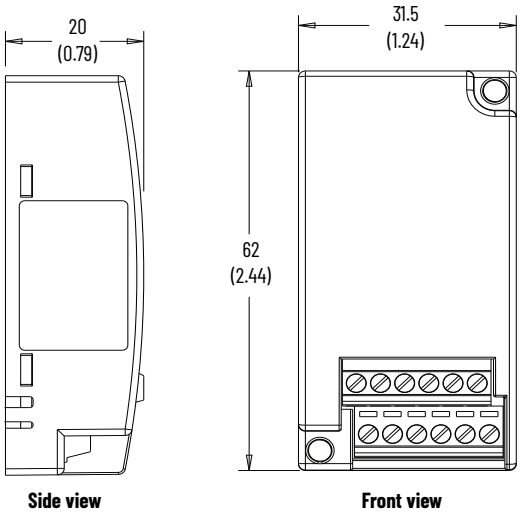
Hardware Features

The plug-in modules, except for the 2080-MEMBAK-RTC and 2080-MEMBAK-RTC2, can be plugged into any plug-in slots on the Micro800 controllers.

2080-RTD2 shown



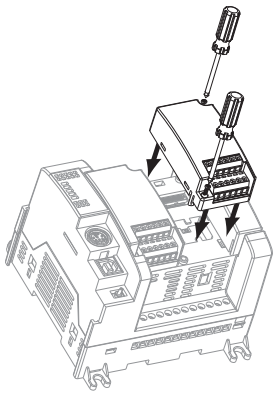
Measurements in millimeters (inches)



Side view **Front view**

Insert Module into Controller

Follow the instructions to insert and secure the plug-in module to the controller.



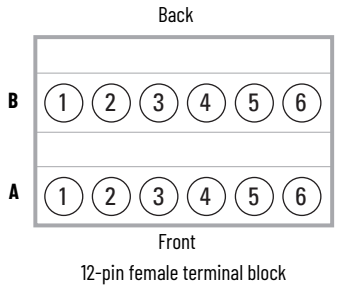
1. Position the plug-in module with the terminal block facing the front of the controller as shown.
2. Snap the module into the module bay.
3. Using a screwdriver, tighten the 10...12 mm (0.39...0.47 in.) M3 self-tapping screw to torque specifications. See the Micro800 Programmable Controllers Technical Data, publication 2080-TD001 for more information.

IMPORTANT	Analog I/O performance depends on the application. For better noise immunity, cable length should ideally be less than 3 m (10 ft) because the plug-ins are non-isolated. For longer cable length requirements, use the 2085 expansion I/O modules instead.

Wiring

The following plug-in modules have 12-pin female terminal blocks:

- 2080-IQ4
- 2080-IQ40B4, 2080-IQ40V4
- 2080-OB4, 2080-OV4, 2080-OW4I
- 2080-IF2, 2080-IF4
- 2080-TC2, 2080-RTD2



12-pin female terminal block

Table 4 - Pin Designations for 12-pin Female Terminal Block Modules

Pin	2080-IQ4	2080-IQ40B4, 2080-IQ40V4	2080-OB4, 2080-OV4	2080-OW4I	2080-IF2	2080-IF4	2080-TC2	2080-RTD2
A1	I-02	I-02	Not used	COM3	COM	COM	CH0+	CH0+
A2	I-03	I-03	Not used	O-3	Not used	VI-2	CH0-	CH0-
A3	COM	COM	-24V DC	Not used	Not used	CI-2	CJC+	CH0L (Sense)
A4	COM	-24V DC	-24V DC	Not used	COM	COM	Not used	Not used
A5	Not used	O-02	O-02	Not used	Not used	VI-3	Not used	Not used
A6	Not used	O-03	O-03	Not used	Not used	CI-3	Not used	Not used
B1	I-00	I-00	Not used	COM0	VI-0	VI-0	CH1+	CH1+
B2	I-01	I-01	Not used	O-0	CI-0	CI-0	CH1-	CH1-
B3	COM	COM	+24V DC	COM1	COM	COM	CJC-	CH1L (Sense)
B4	COM	+24V DC	+24V DC	O-1	VI-1	VI-1	Not used	Not used
B5	Not used	O-00	O-00	COM2	CI-1	CI-1	Not used	Not used
B6	Not used	O-01	O-01	O-2	COM	COM	TH	Not used

Figure 1 - Example Wiring for 2080-IQ40B4

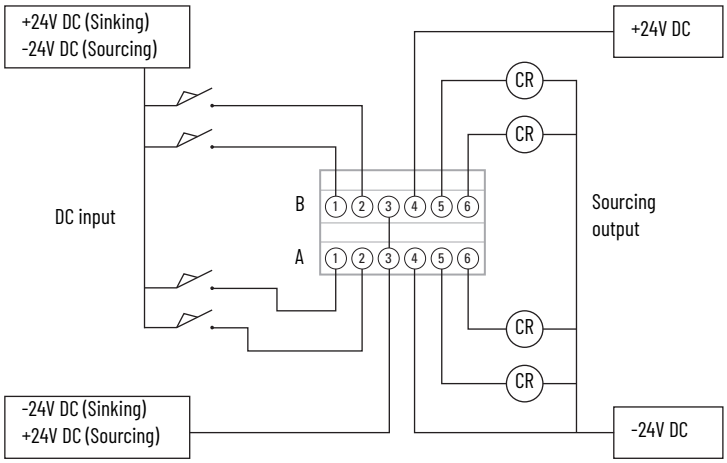


Figure 2 - Example Wiring for 2080-IQ40V4

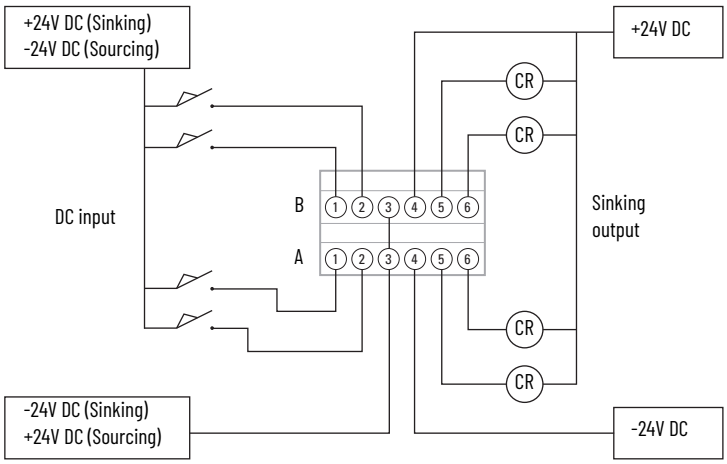
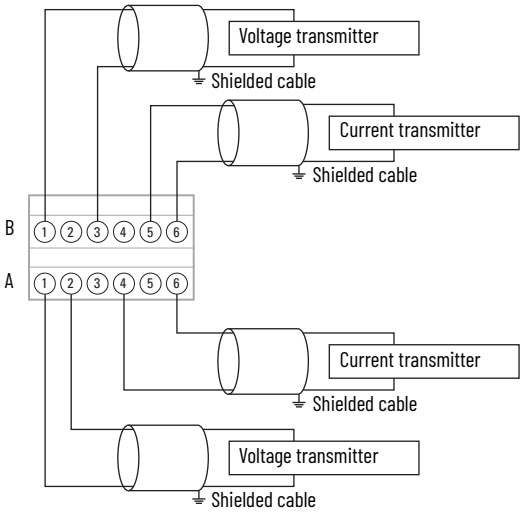
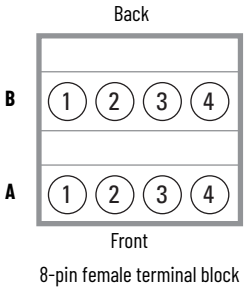


Figure 3 - Example Wiring for 2080-IF4



The following plug-in modules have 8-pin female terminal blocks:

- 2080-OF2
- 2080-SERIALISOL
- 2080-MOT-HSC



8-pin female terminal block

Table 5 - Pin Designations for 8-pin Female Terminal Block Modules

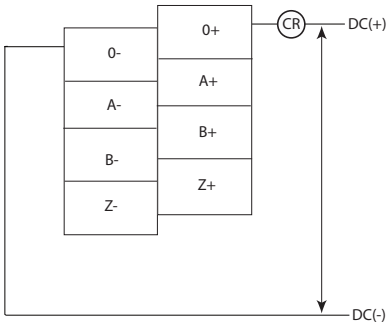
Pin	2080-OF2	2080-SERIALISOL	2080-MOT-HSC(1) (2)
A1	COM	RS-485 B+	O-
A2	COM	GND	A-
A3	COM	RS-232 RTS	B-
A4	COM	RS-232 CTS	Z-
B1	VO-O	RS-232 DCD	O+

Table 5 - Pin Designations for 8-pin Female Terminal Block Modules (Continued)

Pin	2080-OF2	2080-SERIALISOL	2080-MOT-HSC(1) (2)
B2	CO-0	RS-232 RXD	A+
B3	VO-1	RS-232 TXD	B+
B4	CO-1	RS-485 A-	Z+

(1) **IMPORTANT**: Individually shielded, twisted-pair cable (or the type that is recommended by the encoder or sensor manufacturer) should be used for the 2080-MOT-HSC plug-in.
(2) Sinking output/sourcing output wiring for the 2080-MOT-HSC plug-in is shown as follows.

Sinking Output Wiring



Sourcing Output Wiring

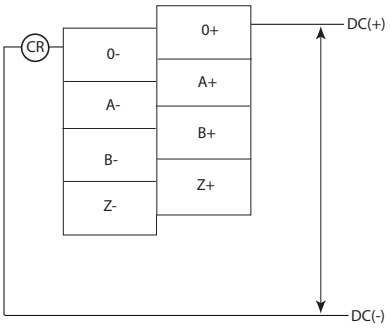
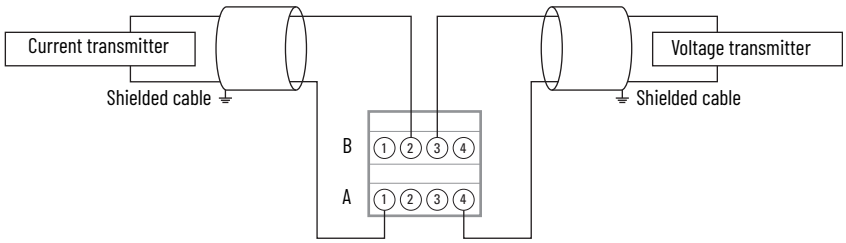
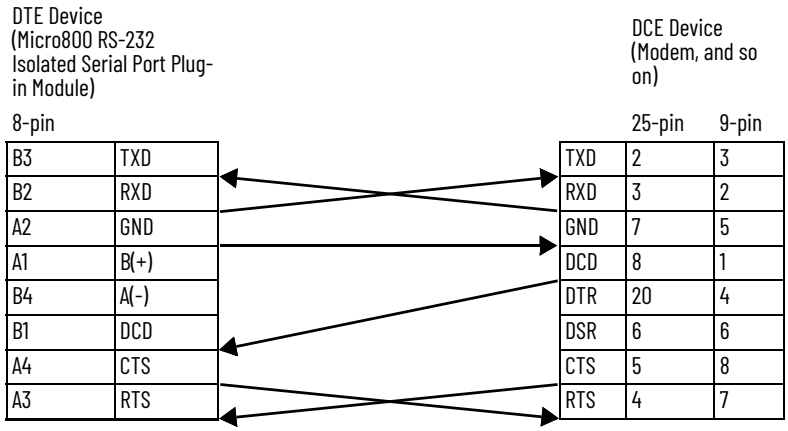


Figure 4 - Example Wiring for 2080-OF2



Serial Port to Modem Cable Pinout

When connecting Micro800 to a modem using an RS-232 cable, the maximum that the cable length may be extended is 15.24 m (50 ft).



ATTENTION: Do not connect to pins A1 and B4 for RS-232 connections. This connection causes damage to the RS-232/485 communication port.

2080-DNET20 – 6-pin Female Terminal Block

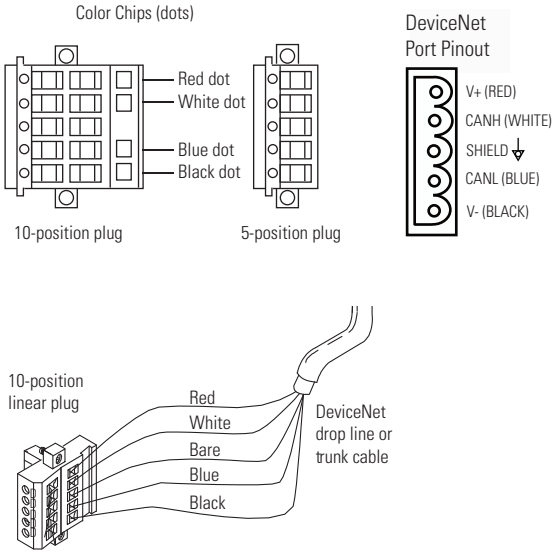
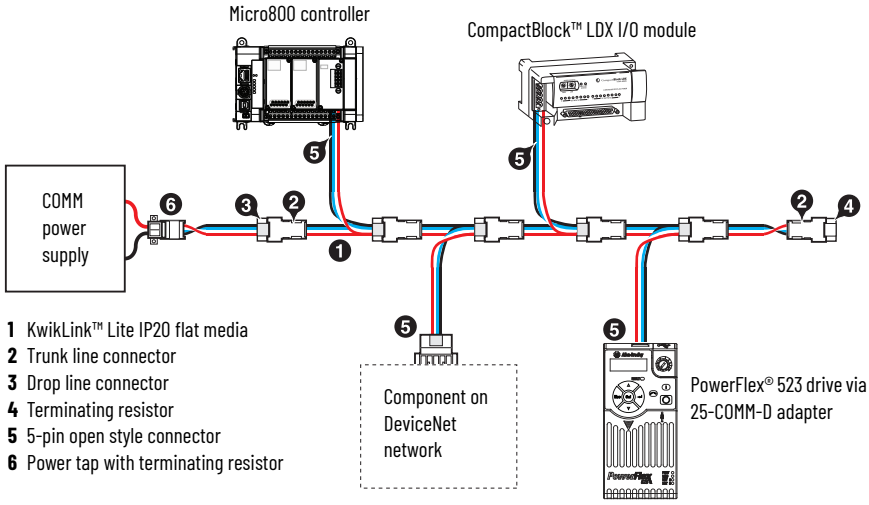


Figure 5 - 2080-DNET20: Sample Network Wiring Using KwikLink Lite Flat Media

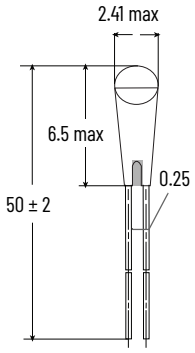


IMPORTANT Individually shielded, twisted-pair cable (or the type that is recommended by the encoder or sensor manufacturer) should be used for the 2080-MOT-HSC plug-in.

Wiring Considerations and Applications for 2080-TC2

ATTENTION: The module currently ships with the CJC thermistor fixed to the module. Do not remove or unscrew the CJC thermistor.
The following sections apply to the previous version of the module.

Type of CJC Sensor



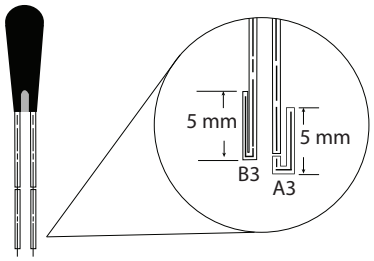
The CJC sensor is a non-polarized, passive negative temperature co-efficient thermistor (EPCOS B57869S0502F140). It is readily available in the market with most third-party suppliers/ vendors.

IMPORTANT	CJC Channel Error
	The CJC channel on 2080-TC2 has a worst-case error of ±1.2 °C @ 25 °C. This error excludes the manufacturer-specified sensor error ±0.2 °C @ 25 °C.

Wire the CJC Thermistor on the 2080-TC2 Module

To wire the thermistor on to the 2080-TC2 module, do the following.

1. Connect the thermocouples to channel 0 and 1, respectively.
2. Connect and screw the thermistor to terminals A3 and B3.



3. Once fitted, bend the black bead of the thermistor such that it contacts the A2 screw securely.

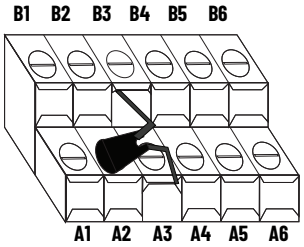
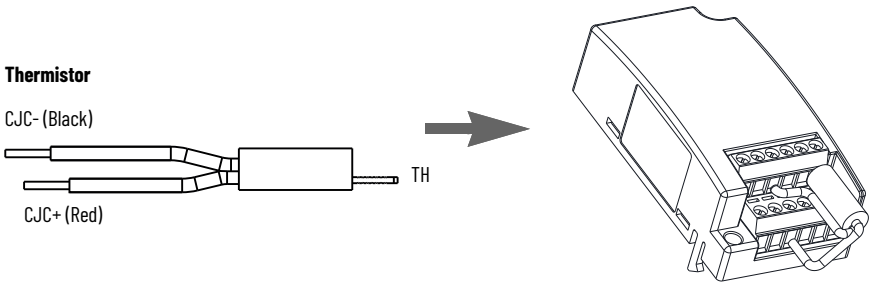
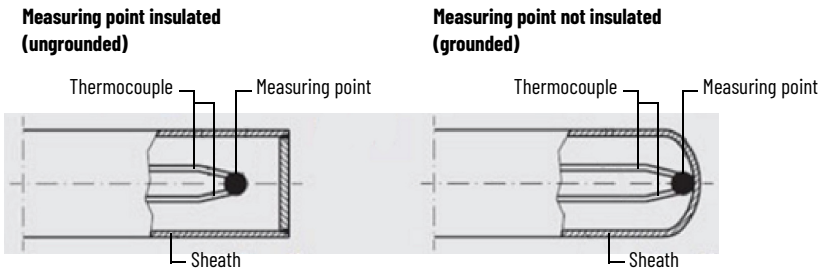


Figure 6 - Position of Thermistor on Plug-in Module



The position for the thermistor, as shown in Figure 6, helps to compensate for thermoelectric voltages developed at screw junction equally for thermocouples that are connected to channels 0 and 1. If the bead is not in proper contact with the screw, there is deviation in readings due to inadequate isothermal compensation.

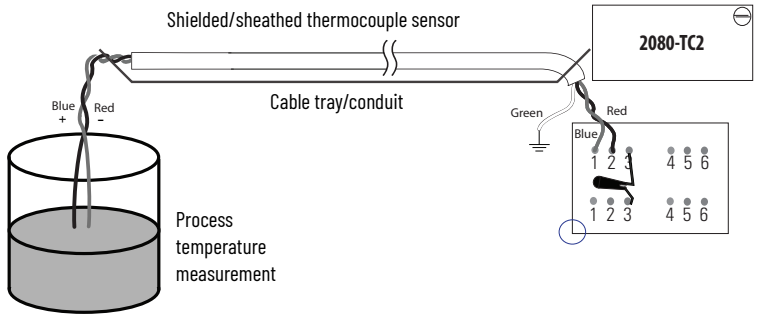
Figure 7 - Tip Designs of Thermocouple Sensors



Wire the Thermocouple Module and Thermocouple Sensor in the Field

Connect the thermocouple sensors directly to the module terminals.

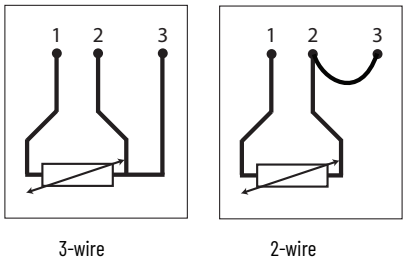
Figure 8 - Direct sensor Wiring



ATTENTION: Direct wiring is the preferred method of wiring for thermocouples.

Wiring Considerations and Applications for 2080-RTD2

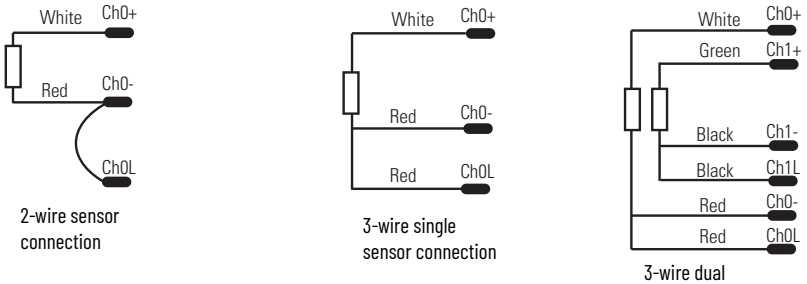
Figure 9 - 2-wire and 3-wire Wiring



Wire the RTD Sensors

In an RTD sensor, the sensing element is always connected between two wires of different colors. Wires of the same color are shorted and form the compensation leads. Measuring resistance between these wires confirms the position of sensing element and compensation elements. Compensation elements will always show 0 ohms.

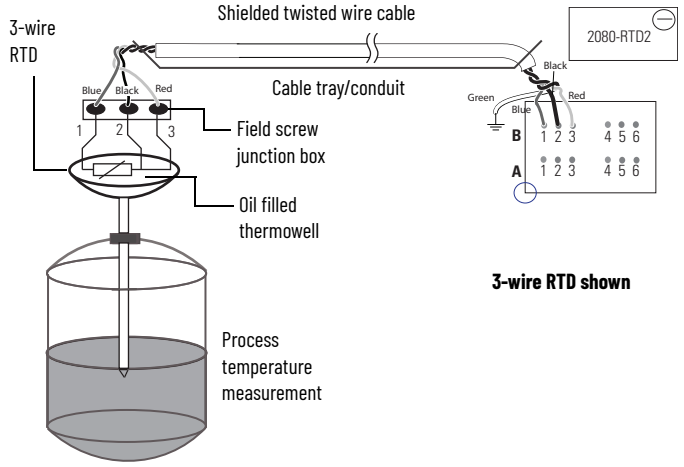
Figure 10 - Wire the Sensors



NOTE: This illustration for channel 0 shows only 2-wire and 3-wire single sensor connections. The wire colors illustrate a particular type of RTD sensor available in the market.

For better accuracy in noisy industrial environments, 3-wire or 4-wire RTD sensors are mostly used. While using these sensors, the resistance added by lead lengths is compensated by an additional third wire in case of 3-wire RTD, and two additional wires in bridge configuration in case of 4-wire RTD. For 2-wire RTD sensor in this module, this lead compensation is provided by using an external 50 mm (2 in.), 0.34 mm^2 (22 AWG) shorting wire between terminals A2-A3 and B2-B3 for channel 0 and 1, respectively. Shielded twisted-pair cables are to be used for remote use of these sensors with cable shield grounded at the controller end.

Wire the RTD Module and RTD Sensor in the Field



The RTD sensing element should always be connected between terminals B1(+) and B2(-) for channel 1, and A1(+) and A2(-) for channel 0 in the module. Terminals B3 and A3 should always be shorted to B2 and A2, respectively, to complete the constant current loop. Mismatch in wiring can cause erroneous, overrange, or underrange readings.

IMPORTANT	Cabling that is used with the 2080-TC2/RTD2 modules have to be shielded twisted cores with the shield wire shorted to chassis ground at the controller end. It is advisable to use 0.34 mm^2 (22 AWG) wires to connect the sensors to the module. Use sensors dipped in oil-filled thermowells for stable and uniform readings. Recommended cable type: Alpha wire P/N 5471C. Performance is dependent on the application. For better noise immunity, cable length should ideally be less than 3 m (10 ft) because the plug-ins are non-isolated. For longer cable length requirements, use the 2085 expansion I/O modules instead.

Wiring Applications for 2080-MOT-HSC

Figure 11 and Figure 12 show wiring applications for the 2080-MOT-HSC plug-in with Kinetix® servo drives.

Figure 11 - Kinetix 3 Servo Drive in Feedback Configuration to 2080-MOT-HSC

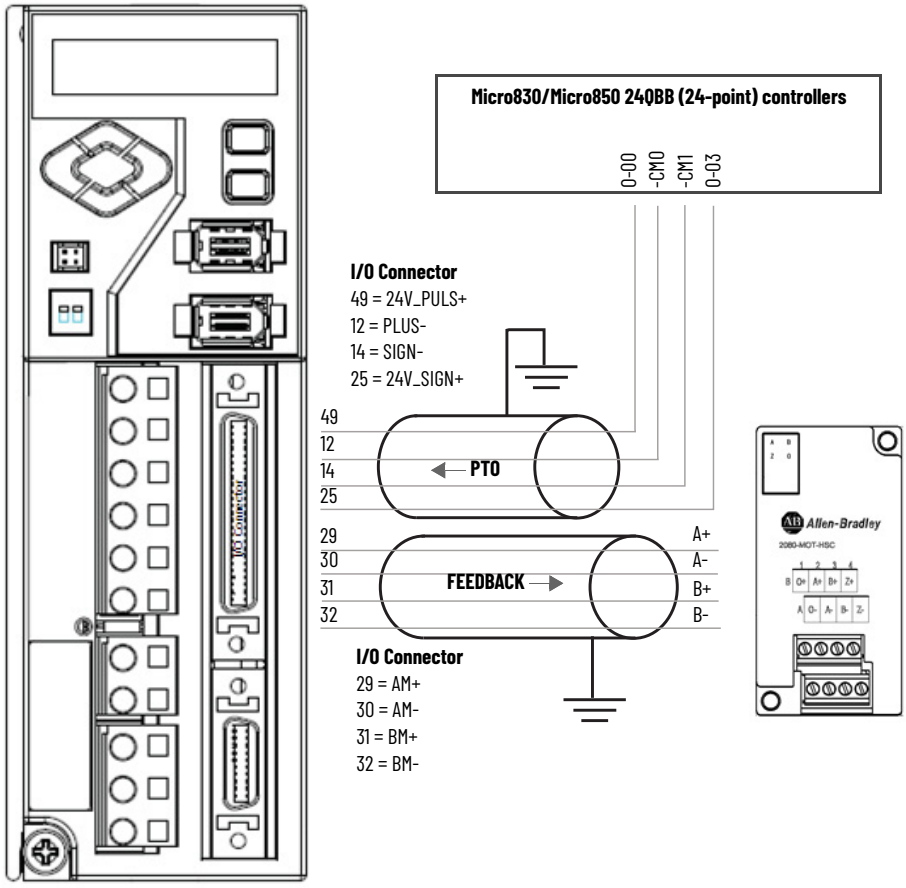
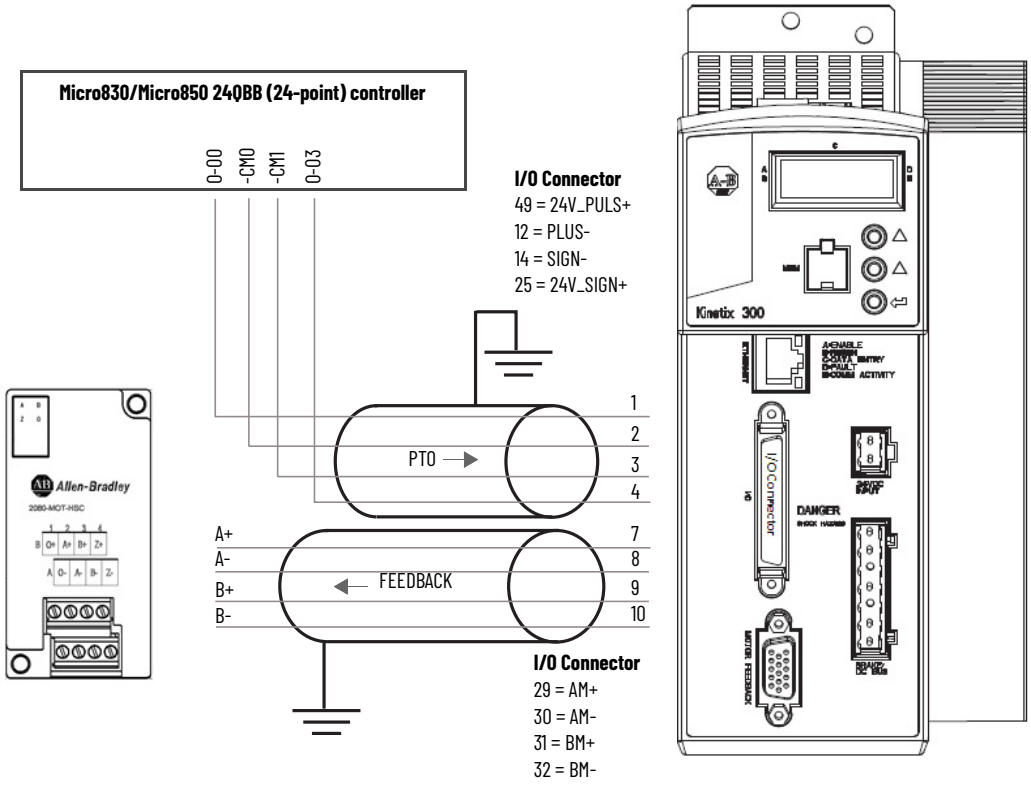


Figure 12 - Kinetix 300 Servo Drive in Feedback Configuration to 2080-MOT-HSC



Non-isolated Thermocouple and RTD Plug-in Modules – 2080-TC2 and 2080-RTD2

The Thermocouple (2080-TC2) and RTD (2080-RTD2) plug-in modules allow for temperature measure and control when used with PID.

This plug-in can be used in any slot of your Micro800 controller. Removal and Insertion Under Power (RIUP) is not supported.

Thermocouple Module

The 2080-TC2 two-channel plug-in module supports thermocouple measurement. It digitally converts and transmits temperature data from any combination of up to eight types of thermocouple sensors. Each input channel is individually configurable through the Connected Components Workbench software for a specific sensor, filter frequency.

Thermocouple Sensor Types and Ranges

The module supports B, E, J, K, N, R, S, T types of thermocouple sensors. The module channels are referred to as Channel 0, Channel 1, and CJC, respectively. The cold junction compensation is provided by an external NTC thermistor, which comes with the module. The thermistor has to be fitted to the screw terminals A3 and B3 of the module. This CJC is common to channel 0 and 1 thermocouple sensors and provides open-circuit, overrange, and underrange detection and indication.

Overrange and Underrange Conditions

If the channel temperature input is below the minimum value of its normal temperature range for the represented sensor, the module reports an underrange error through the Connected Components Workbench global variables. If the channel reads above the maximum value of its normal temperature range for the represented sensor, an overrange error is flagged.

Table 6 defines thermocouple types and their associated full-scale temperature ranges.

Table 6 - Thermocouple Sensor Types and Temperature Ranges

Thermocouple Type	Temperature Range °C (°F)		Accuracy °C (°F)		ADC Update Rate in Hz (Accuracy °C)
	Min	Max	±1.0 °C	±3.0 °C	
B	40 (104)	1820 (3308)	90…1700 (194…3092)	< 90 (194) > 1700 (3092)	(±1.0) 4.17, 6.25, 10, 16.7 (±3.0) 19.6, 33, 50, 62, 123, 242, 470
E	-270 (-454)	1000 (1832)	-200…+930 (-328…+1706)	< -200 (-328) > 930 (1706)	
J	-210 (-346)	1200 (2192)	-130…+1100 (-202…+2012)	< -130 (-202) > 1100 (2012)	
K	-270 (-454)	1370 (2498)	-200…+1300 (-328…+2372)	< -200 (-328) > 1300 (2372)	
N	-270 (-454)	1300 (2372)	-200…+1200 (-328…+2192)	< -200 (-328) > 1200 (2192)	
R	-50 (-58)	1760 (3200)	40…1640 (104…2984)	< 40 (104) > 1640 (2984)	
S	-50 (-58)	1760 (3200)	40…1640 (104…2984)	< 40 (104) > 1640 (2984)	
T	-270 (-454)	400 (752)	-220…+340 (-364…+644)	< -220 (-364) > 340 (644)	

To configure Thermocouple type and update rate in Connected Components Workbench software, see Quick Start on page 55.

RTD Module

The 2080-RTD2 module supports RTD measurement applications that support up to two channels. The module digitally converts analog data and transmits the converted data in its image table.

The module supports connections from any combination of up to eleven types of RTD sensors. Each channel is individually configurable through the Connected Components Workbench software. When configured for RTD inputs, the module can convert the RTD readings into temperature data. See Temperature Conversion – Data to Degree Celsius (°C) on page 26, for converting temperature data to actual temperature degree.

RTD Sensor Types and Ranges

Each channel provides open-circuit (all wires), short-circuit (excitation and return wires only), and overrange and underrange detection and indication. The 2080-RTD2 module supports 11 types of RTD sensors:

Table 7 - Types of RTD Sensors

PT100 385	PT1000 385	PT500 392	Ni120 672
PT200 385	PT100 392	PT1000 392	NiFe604 518
PT500 385	PT200 392	Cu10 427	

It supports 2-wire and 3-wire type of RTD sensor wiring.

RTD Compatibility

An RTD consists of a temperature-sensing element that is connected by two, three, or four wires that provide resistance input to the module. The following table lists the RTD types that you can use with the module, including their temperature range, accuracy, and ADC update rate.

Overrange and Underrange Conditions

If the channel temperature input is below the minimum value of its normal temperature range for the represented sensor, the module reports an underrange error through the Connected Components Workbench global variables. If the channel temperature input is above the maximum value of its normal temperature range for the represented sensor, an overrange error is flagged.

Table 8 - RTD Sensor Types and Temperature Ranges

RTD Type	Temperature Range °C (°F)		Accuracy °C (°F)		ADC Update Rate in Hz (Accuracy °C)
	Min	Max	±1.0 °C	±3.0 °C	
PT100 385	-200 (-328)	660 (1220)	-150...+590 (-238...+1094)	< -150 (-238) > 590 (1094)	**3-wire others** 4.17, 6.25, 10, 16.7,19.6, 33, 50 (±1.0) 62, 123, 242, 470 (±3.0) 2-wire and 3-wire Cu10[1] 4.17, 6.25, 10, 16.7 (>±1.0 < ±3.0) 19.6, 33, 50, 62, 123, 242, 470 (> ±3.0) 2-wire others 4.17, 6.25, 10, 16.7 (±1.0) 19.6, 33, 50, 62, 123, 242, 470 (±3.0)
PT200 385	-200 (-328)	630 (1166)	-150...+570 (-238...+1058)	< -150 (-238) > 570 (1058)	
PT500 385	-200 (-328)	630 (1166)	-150...+580 (-238...+1076)	< -150 (-238) > 580 (1076)	
PT1000 385	-200 (-328)	630 (1166)	-150...+570 (-238...+1058)	< -150 (-238) > 570 (1058)	
PT100 392	-200 (-328)	660 (1220)	-150...+590 (-238...+1094)	< -150 (-238) > 590 (1094)	
PT200 392	-200 (-328)	630 (1166)	-150...+570 (-238...+1058)	< -150 (-238) > 570 (1058)	
PT500 392	-200 (-328)	630 (1166)	-150...+580 (-238...+1076)	< -150 (-238) > 580 (1076)	
PT1000 392	-50 (-58)	500 (932)	-20...+450 (-4...+842)	< - 20 (-4) > 450 (842)	

Table 8 - RTD Sensor Types and Temperature Ranges (Continued)

RTD Type	Temperature Range °C (°F)		Accuracy °C (°F)		ADC Update Rate in Hz (Accuracy °C)
	Min	Max	±1.0 °C	±3.0 °C	
Cu10 427	-100 (-148)	260 (500)	(1)	< -70 (-94) > 220 (428)	
Ni120 672	-80 (-112)	260 (500)	-50...+220 (-58...+428)	< -50 (-58) > 220 (428)	
NiFe604 518	-200 (-328)	200 (392)	-170...+170 (-274...+338)	< -170 (-274) > 170 (338)	

(1) For Cu10 427, the accuracy range is within >±1.0 < ±3.0 for -70...+220 °C (-94...+428 °F). Above this temperature range, it is > ±3.0 °C as shown in the table.

Connected Components Workbench Global Variables Data Maps

The following bit/words describe the information that is read from the Thermocouple and RTD plug-in modules in the Connected Components Workbench global variables.

Table 9 - Mapping Table

Word Offset	Bit															
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
00 (example: _IO_P1_AI_00)	Channel 0 Temperature Data															
01 (example: _IO_P1_AI_01)	Channel 1 Temperature Data															
02 (example: _IO_P1_AI_02)	Channel 0 Information															
	UKT	UKR	Reserved				Reserved		OR	UR	OC	DI	CC	Reserved		
03 (example: _IO_P1_AI_03)	Channel 1 Information															
	UKT	UKR	Reserved				Reserved		OR	UR	OC	DI	CC	Reserved		
04 (example: _IO_P1_AI_04)	System Information															
	Reserved				SOR	SUR	COC	CE	Reserved							

Table 10 - Bit Definitions

Bit Name	Description
Channel Temperature Data	The temperature count mapped from temperature Celsius degree with one decimal. Check the section, Temperature Conversion – Data to Degree Celsius (°C) on page 26, for the mapping formula.
UKT (Unknown Type)	Bit set to report an unknown sensor type error in configuration
UKR (Unknown Rate)	Bit set to report an unknown update rate error in configuration
OR (Overrange)	Bit set to indicate overrange on channel input. The Channel Temperature Data shows the maximum temperature count for the individual type of sensor that is used and the value does not change until the overrange error is clear.
UR (Underrange)	Bit set to indicate that the channel input underrange happens. The Channel Temperature Data shows the minimum temperature count for the individual type of sensor that is used and the value does not change until the underrange error is clear.
OC (Open Circuit)	Bit set to indicate open-circuit on the channel input sensor
DI (Data Illegal)	The data in the channel data field is illegal and cannot be used. This bit is set when temperature data is not ready for use.
CC (Code Calibrated)	Bit set indicates that temperature data is calibrated by the system calibration coefficient.
SOR (System Overrange)	Bit set to indicate system overrange error with environment temperature over 70 °C (158 °F)
SUR (System Underrange)	Bit set to indicate system underrange error with environment temperature under -20 °C (-4 °F)
COC (CJC open-circuit)	Bit set to indicate CJC sensor not connected for thermocouple module, open circuit. This bit is for the thermocouple module only.
CE (Calibration Error)	Bit set indicates that the module is not accurate. This bit is set to 0 by default and remains as 0. Contact Technical Support when the value is otherwise.

Temperature Conversion – Data to Degree Celsius (°C)

To keep the precision of temperature value from the Thermocouple and RTD plug-in modules, there is a general data-mapping conversion in the firmware before the actual temperature is sent to the Connected Components Workbench software.

The following equation shows how the Connected Components Workbench software data count is mapped from temperature Celsius degree by the firmware:

Connected Components Workbench software Data Count = (Temp (°C) + 270.0)*10

IMPORTANT	This conversion formula applies to all types of Thermocouple and RTD sensors.

This equation illustrates how the Connected Components Workbench data count does not use the full range of 0...65535 of the data word.

Derive Actual Temperature (°C) From Connected Components Workbench Data Count

The following formula shows how to derive temperature Celsius degree from the temperature data word in the Connected Components Workbench software:

Temp (°C) = (Data - 2700)/10

Examples:

1234 -> (1234 - 2700)/10 -> -146.6 °C
8000 -> (8000 - 2700)/10 -> 530.0 °C

IMPORTANT	Underrange and overrange error report checking is **not** based on the temperature data count in Connected Components Workbench software, but the actual temperature (°C) or the voltage going into the plug-in module.

High-speed Counter – 2080-MOT-HSC

Overview

The 2080-MOT-HSC plug-in module provides enhanced high-speed counter capabilities to the Micro800 controller. It supports the same functionalities of an embedded high-speed counter on the Micro800 controllers but is enhanced to support up to 250 KHz 5V differential line driver for improved noise immunity and provides additional dedicated I/O.

The 2080-MOT-HSC module supports most commercial encoders (5V differential or 24V single-ended).

IMPORTANT	To configure the plug-in module, you need to download and use the HSC user-defined function block (UDFBs) from the Rockwell Automation Sample Code Library. From Connected Components Workbench software version 7 or later, the sample code is included in the installation and is located in the folder: \documents\public documents\ccw\samples\rockwell automation\udfb See Quick start Projects for 2080-MOT-HSC Plug-in on page 67 for step-by-step instructions on how to use the plug-in with a sample project.

IMPORTANT	From Connected Components Workbench software version 8 or later, support has been added for a HSC Feedback Axis which uses the same instructions as the PTO Motion Axis. UDFBs are still supported (you can use either one but you cannot select both for the same plug-in).

IMPORTANT	From Connected Components Workbench software version 11 or later, support has been added for native HSCE instructions which can be used in place of the UDFBs. With native HSCE instructions, you can configure the plug-in offline using a graphical user interface. With UDFBs, configuration is done at runtime using instructions.

IMPORTANT	2080-MOT-HSC modules with hardware revision 1.xxx only supports a value of one for Number of Pulses for rate calculation. 2080-MOT-HSC modules with hardware revision 2.xxx enhances support by enabling you to choose a value from 1...255. The hardware revision is found on the label on the module.

Differences Between Embedded HSC and Plug-in Module

Table 11 lists the differences between the embedded HSC and the 2080-MOT-HSC plug-in module.

Table 11 - Differences Between Embedded HSC and HSC Plug-in Module

Embedded HSC	2080-MOT-HSC Plug-in Module
Limited to 100 kHz	Up to 250 kHz
12/24V input	12/24V input or 5V differential line driver
None	Embedded rate calculation using "Per Pulse" and "Cyclic" methods
Dedicated preset and hold inputs	Configure either preset or hold inputs
None	One 5/24V output

IMPORTANT When using the 2080-MOT-HSC module, the high and low preset status does not automatically reset. When using the embedded HSC, it automatically resets when the high and low preset condition no longer exists.

Counter Specifications

Filter and decode inputs: 3 input points A, B, Z

These input points may come from different types and configurations of sensors. The user must configure the module to respond to the type of sensor connected to the module as described below. This can be configured in the 2080-MOT-HSC UDFB. From Connected Components Workbench software version 8.0 or later, if you have configured the plug-in for Feedback Axis, you can also edit the input filter values in the plug-in configuration module properties.

Nominal Filter Settings	Maximum Guaranteed Block Pulse Width	Minimum Guaranteed Pass Pulse Width
No Filter – Default	–	–
250 kHz (DC 2 µs)	617 kHz (DC 0.81 µs)	263kHz (DC 1.9 µs)
200 kHz (DC 2.5 µs)	352 kHz (DC 1.42 µs)	201 kHz (DC 2.48 µs)
80 kHz (DC 6.25 µs)	135 kHz (DC 3.7 µs)	86.2 kHz (DC 5.8 µs)
40 kHz (DC 12.5 µs)	62.5 kHz (DC 8.0 µs)	42.5 kHz (DC 11.76 µs)
13.3 kHz (DC 37.6 µs)	20.8 kHz (DC 24.1 µs)	14.1 kHz (DC 35.5 µs)
10 kHz (DC 50 µs)	15.7 kHz (DC 31.8 µs)	10.5 kHz (DC 47.2 µs)
4 kHz (DC 125 µs)	6.3 kHz (DC 79.3 µs)	4.2 kHz (DC 119 µs)
2 kHz (DC 250 µs)	3.2 kHz (DC 156 µs)	2.1 kHz (DC 237 µs)
1 kHz (DC 0.5 ms)	1.6 kHz (DC 0.31 ms)	1.0 kHz (DC 0.5 ms)
500 Hz (DC 1 ms)	778 Hz (DC 0.642 ms)	526 Hz (DC 0.95 ms)
250 Hz (DC 2 ms)	389 Hz (DC 1.285 ms)	263 Hz (DC 1.9 ms)
125 Hz (DC 4 ms)	195 Hz (DC 2.6 ms)	131 Hz (DC 3.81 ms)
62.5 Hz (DC 8 ms)	97.3 Hz (DC 5.14 ms)	65.8 Hz (DC 7.6 ms)
31.25 Hz (DC 16 ms)	48.5 Hz (DC 10.3 ms)	32.9 Hz (DC 15.2 ms)
50% duty		

IMPORTANT For low frequency pulses, filter times should be set appropriately to avoid extra pulses from a noisy environment. For high frequency pulses, shielded cable must always be used.

IMPORTANT When the controller is power cycled, the value of the counters are reset to zero.

The counters are not reset to zero for program download. For example, if using the feedback axis, use the MC_SetPosition function block to reset the position to zero.

Number of Counters: 1 to 2

The module may be configured, using HSC_Mode, to use the inputs as 1 or 2 counters.
1 counter: A, B, Z = Counter 0
2 counters: A, Z = Counter0; B = Counter 1

Figure 13 - Counter Pin Usage

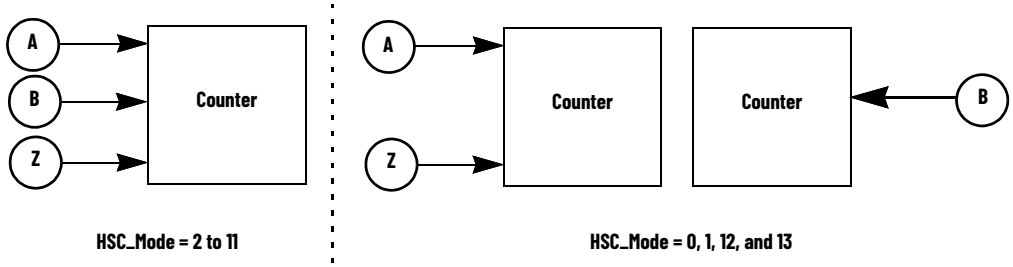
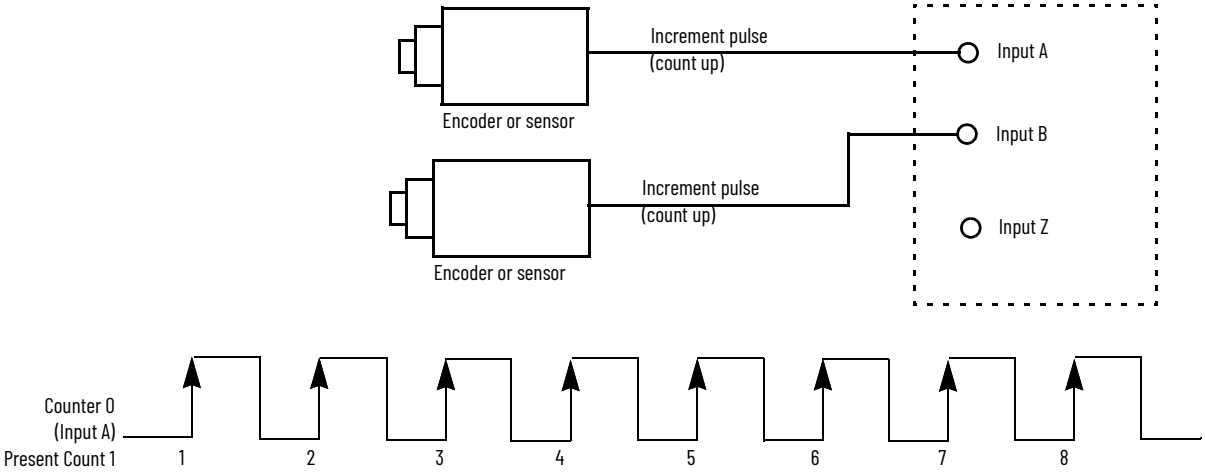


Table 12 - Input Operational Modes

Mode	Description
0	Up Counter – The accumulator is immediately cleared (0) when it reaches the high preset. A low preset cannot be defined in this mode.
1	Up Counter with external reset and hold – The accumulator is immediately cleared (0) when it reaches the high preset. A low preset cannot be defined in this mode.
2	Counter with external direction
3	Counter with external direction, reset, and hold
4	Two input counter (up and down)
5	Two input counter (up and down) with external reset and hold
6	Quadrature counter (phased inputs A and B)
7	Quadrature counter (phased inputs A and B) with external reset and hold
8	Quadrature X4 counter (phased inputs A and B)
9	Quadrature X4 counter (phased inputs A and B) with external reset and hold
10	Quadrature X2 counter (phased inputs A and B)
11	Quadrature X2 counter (phased inputs A and B) with external reset and hold
12	Down Counter
13	Down Counter with external reset and hold

Up Counter

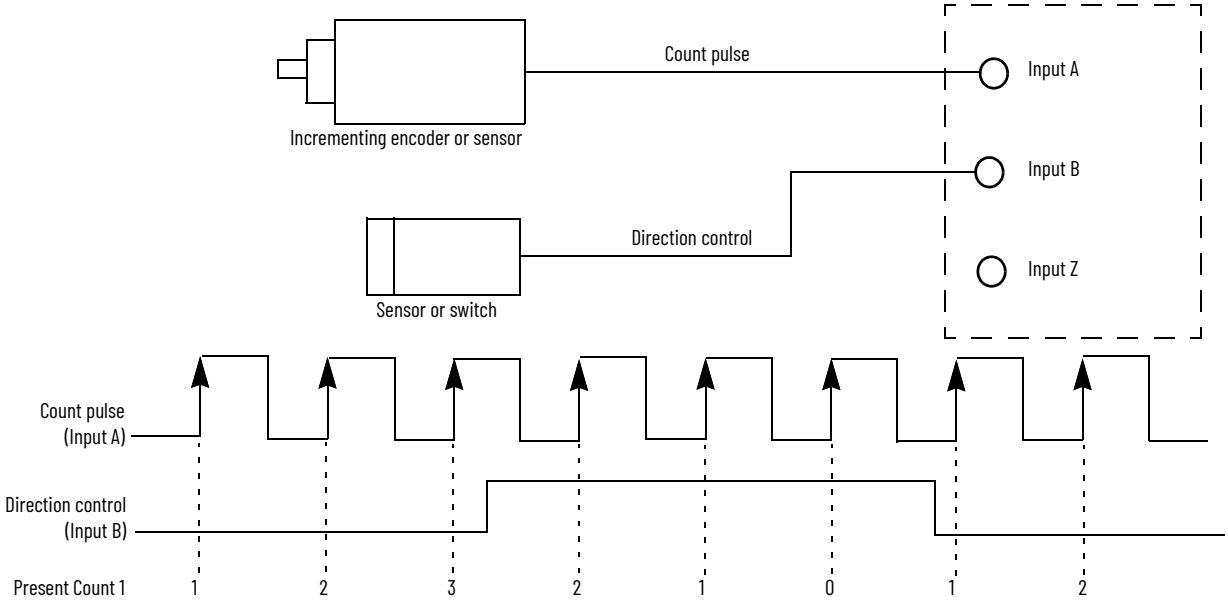
Pulses on A cause the up counter (Counter 0). Pulses on B also cause the up counter (Counter 1).



Counter with External Direction

Pulses on A cause the counter to increment when B is low and decrement when B is high. When B is open or undriven, the counter increments. See Pulse External Direction Counting on page 30.

Figure 14 - Pulse External Direction Counting



A	B	Change in Count Value
↑	0 (Open or No Connection)	+1
↑	1	-1
1	X (don't care)	0
0	X (don't care)	0

Two input counter (Up/Down pulses)

Pulses on A cause the counter to increment. Pulses on B cause the counter to decrement. Pulses may occur at any time. Note that pulses can occur very closely (that is, much faster than plug-in scan time) so that the plug-in does not capture the change in count. In such cases, both counts may be ignored (the net change being zero). A pulse that is not captured is not considered lost. See Figure 15.

Figure 15 - Two Input Counting

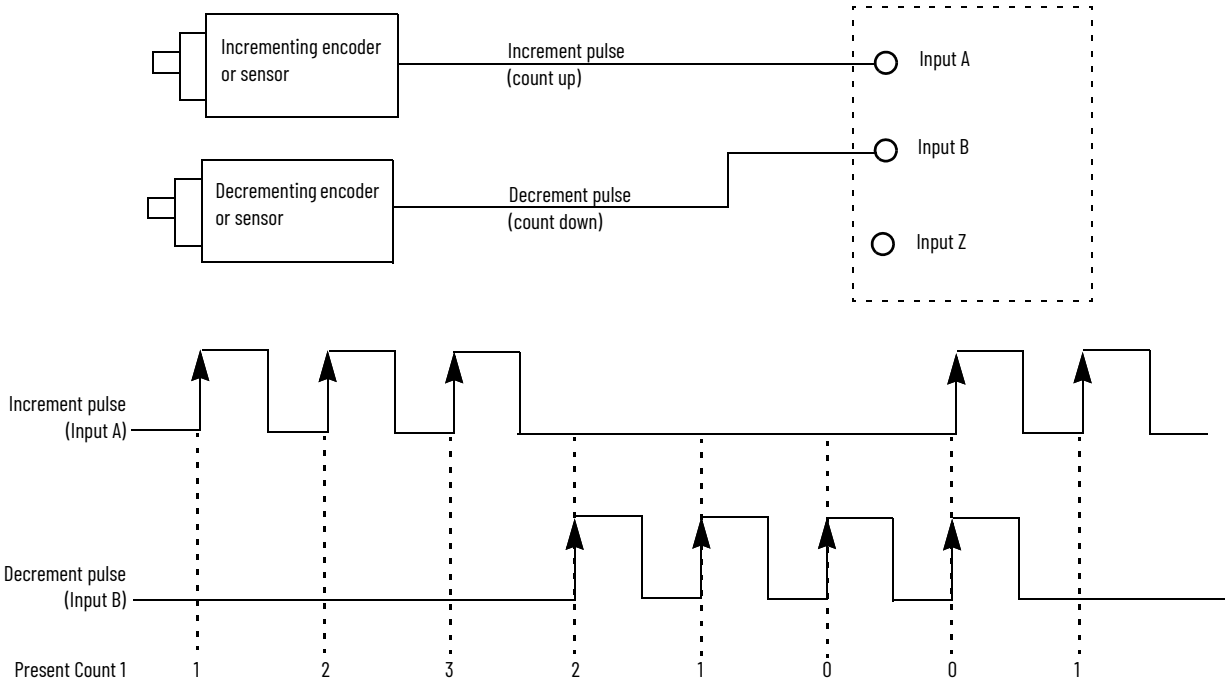


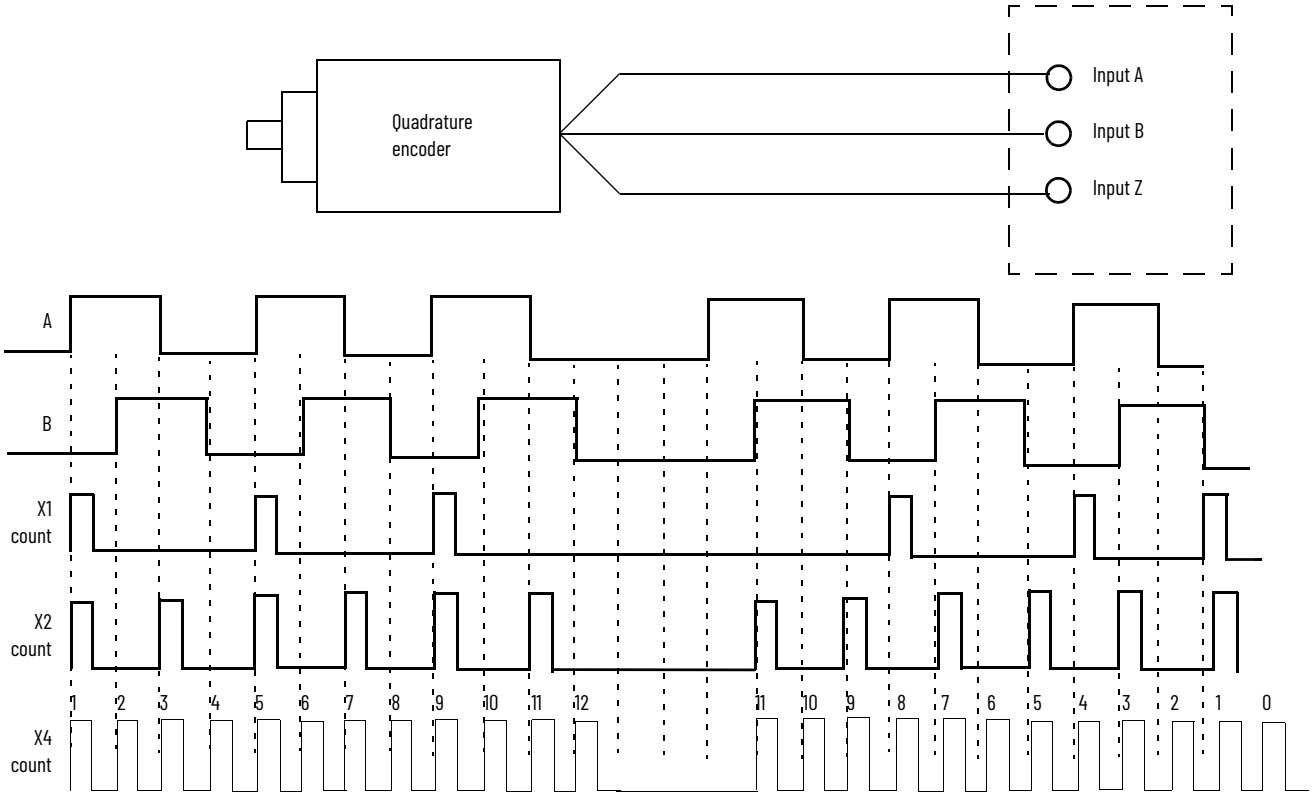
Table 13 - Up/Down Counting

A	B	Change in Count Value
↑	0 or 1	+1
0 or 1	↑	-1
↑	↑	0
0	0	0

Quadrature Counter (X1)

The module is compatible with 2 and 3 signal quadrature, or incremental encoders. The A and B signals are offset by 90 degrees and encode the direction of the rotation. The third signal, Z, occurs once per revolution and is often used as a home reference. The module's use of this signal is discussed below in the Z input section.

Figure 16 - Quadrature Counting



Quadrature X4 Counter

Counter shall increment or decrement on each edge of the A and B pulses when the signal is in the positive or negative direction respectively. See Figure 16.

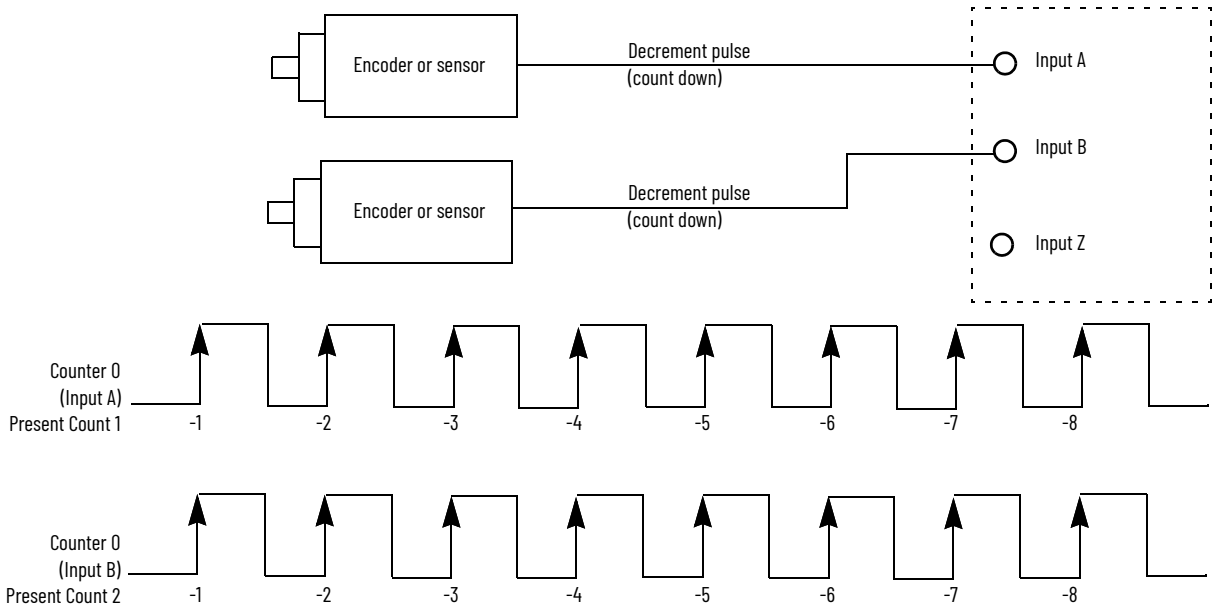
Quadrature X2 Counter

The counter increments or decrements on each edge of the A pulse when the signal is in the positive or negative direction respectively. See Figure 16.

Down Counter

Pulses on A cause the down counter (Counter 0). Pulses on B also cause the down counter (Counter 1).

Figure 17 - Down Counting



Z Input (Gate) Function/Touch Probe

This signal functionality supports:

- **Touch Probe** the present count value on the rising edge of IntZ_n to the HSC_Touch Probe term in the backplane input file
- **Hold** the counter at its present count value while IntZ_n = 1
- **Reset** the present count value on rising edge of IntZ_n

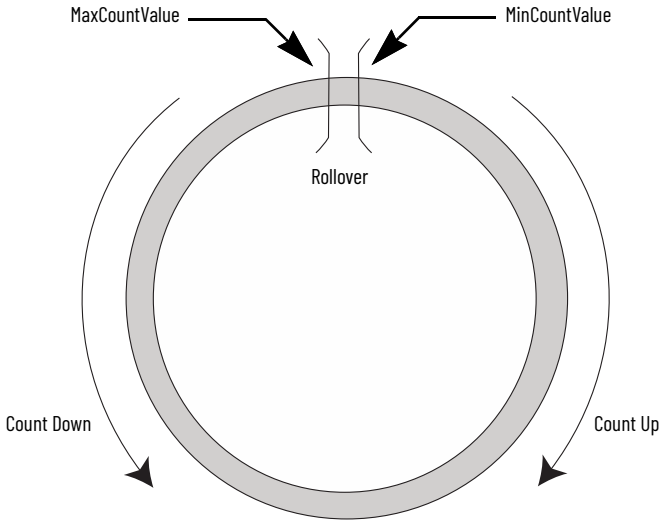
IMPORTANT	If the module gets two or more Z pulses during a single plug-in scan, the HSC_TouchProbe is overwritten with the last stored value. There is no indication that more than one store has occurred.

Ring or Linear Counter

The counter may be configured with the RingOrLinearCnt_n control bit to rollover at its limits (ring counter) or to stop counting and set a flag (linear counter).

0: Ring counter. When the counter is a ring counter and the present count value is equal to MaxCountValue_n, the next input count in the up direction causes the PresentCount_n to become the MinCountValue_n. This action is known as rollover. And the CountOverflow_n flag is set to indicate that a rollover has happened. It is reset using the ResetCountOverflow bit.

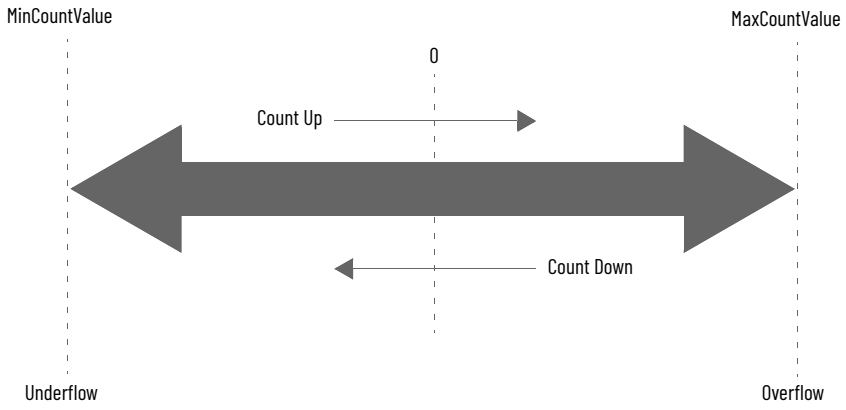
Conversely, when the PresentCount_n is equal to MinCountValue_n the next input count in the down direction causes the PresentCount_n to become the MaxCountValue_n. This action is known as rollunder. The CountUnderflow_n flag is set to indicate that a rollunder has occurred. It is reset using the ResetCountUnderflow_n bit.



1: Linear counter. When the counter is a linear counter and the present count value is equal to MaxCountValue_n the next input count in the up direction activates the CountOverflow_n bit and also the PresentCount_n remains at the MaxCountValue_n. CountOverflow_n is reset using the ResetCountOverflow_n bit.

Conversely, when the PresentCount_n is equal to MinCountValue_n the next input count in the down direction activates the CountUnderflow_n bit and the PresentCount_n remains at MinCountValue_n. CountUnderflow_n is reset using the ResetCountUnderflow_n bit.

IMPORTANT	The counts in overflow and underflow are not accumulated. That is, even if 1000 pulses are applied while in overflow, the first pulse with the opposite direction (down in this case) causes the counter to be decremented by 1. The CountOverflow_n bit is only reset using the ResetCountUnderflow_n bit.



Enabling and Disabling a Counter using the HSC_EN bit

Disabling the counter does not inhibit any HSC_ACC_Bn loading functions (preset or direct write) or any Z function.

The module continuously calculates rates for each of the counters regardless of input operational mode.

Timer

For the first two counters, a timer is used to measure the time between two successive pulses. This value is reported to the backplane as HSC_PULSE_WIDTH_Bn.

Understanding Rates

There are different applications which require rate information but there is no one perfect method for all. Generally, the user must weigh rate accuracy with the need for new information quickly.

Broadly, there are two different ways to calculate rates and optimize accuracy and speed of the rate of calculation:

- Per Pulse
  1/HSC_PULSE_WIDTH_B (supported through 2080-MOT-HSC plug-in)
  HSC_PULSE_WIDTH_Bn is reported to the user in the input array.
- Cyclic
  Number of Pulses/User Defined Time Interval (supported through Connected Components Workbench)
  PresentRate_n is reported to the user in the input array.

Per Pulse

The Per Pulse rate method can be very accurate if the time between pulses is large compared to the timer clock (1 µs for 2080-MOT-HSC). A timer is used to measure the time between the two successive pulses. This value is reported to the backplane as HSC_PULSE_WIDTH_Bn after each pulse. The user may invert this value to derive a rate.

Per Pulse rate = 1 / HSC_PULSE_WIDTH_B

However, when the time between pulses shrinks, two factors can distort the Per Pulse calculation of rate values:

- The time between pulses is closer to measuring the clock's frequency, making the granularity of the time increments have a greater effect on rate inaccuracy.
- Also, the rate may be calculated multiple times during the course of one backplane scan time. This means that the rate data is obtained at a backplane scan is only that of the very last pair of pulses and disregards the other rate calculations that have happened during that interval. This is especially problematic if the pulses during the update time are unevenly spaced, the reported rate could be based entirely on two pulses which are extremely close together (a very high rate) but a third pulse was separated by a greater time (low rate).

You must understand these limitations when using HSC_PULSE_WIDTH_Bn to derive a rate.

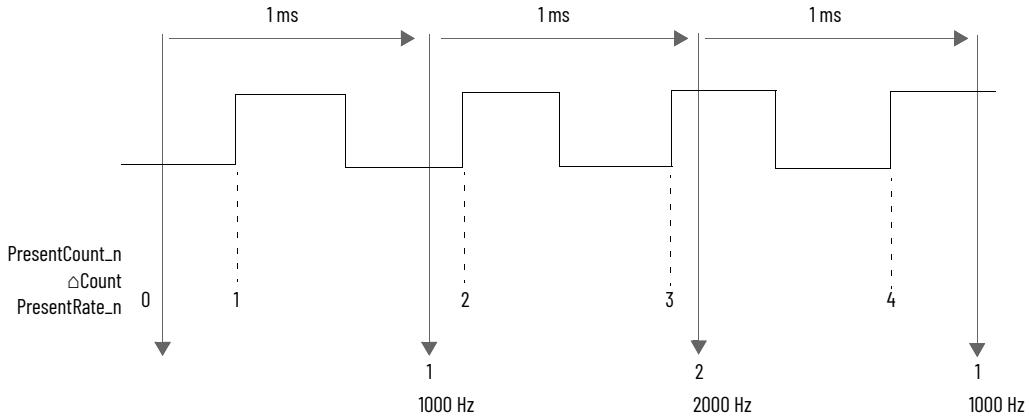


Table 14 - Per Pulse Errors(1)

Real Pulses (1.9999 can be rounded to 2)	Pulses Reported By Module	Real Frequency	Reported Frequency	% Error
2	1	500 kHz	1 MHz	100%
9	10	111 kHz	100 kHz	11.1%
101	100	9.901 kHz	10.000 kHz	1.00%
1001	1000	999 Hz	1000 Hz	0.10%
9,999	10,000	100.01 Hz	100.00 Hz	0.010%
99,999	100,000	10.00010 Hz	10.00000 Hz	0.001%

(1) This table does not represent accuracy per pulse but repeatability. This repeatability can be applied in the No Filter setting.

Table 15 - Maximum Cyclic Rate Errors

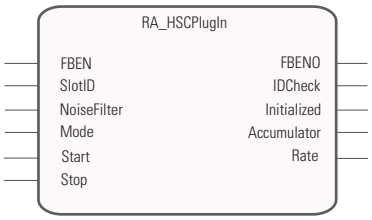
Update Time Value Scalar	Frequency				
	100 Hz	1 kHz	10 kHz	100 kHz	1 MHz
1	NA	NA	20.02%	20.02%	0.210%
10	NA	20.11%	2.020%	0.210%	0.030%
100	20.01%	2.110%	0.220%	0.031%	0.012%
1000	3.010%	0.310%	0.040%	0.013%	0.010%
10,000	1.210%	0.130%	0.022%	0.011%	0.010%

IMPORTANT For low frequency pulses, set filter times appropriately to avoid extra pulses from a noisy environment. For high frequency pulses, you must use shielded cables.

User Defined Function Blocks

UDFBs only apply if UDFB mode is selected in Connected Components Workbench software. It is recommended to use Feedback Axis (Version 8.0 or later) or native HSCE instructions (Version 11.0 or later) to configure the 2080-MOT-HSC plug-in module.

RA_HSCPlugIn



The purpose of this UDFB is to get high-speed counter accumulator value and current pulse frequency.

Table 16 - RA_HSCPlugIn: Input and Output Parameters

Parameter	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input
SlotID	INPUT	UINT	Plug-in slot number Slot ID = 1…5 (starting with the far left slot 1)
NoiseFilter	INPUT	USINT	00: No filter 01: 250 kHz 02: 200 kHz 03: 80 kHz 04: 40 kHz 05: 13.3 kHz 06: 10 kHz 07: 4 kHz 08: 2 kHz 09: 1 kHz 10: 500 Hz 11: 250 Hz 12: 125 Hz 13: 63.5 Hz 14: 31.25 Hz
HSCMode	INPUT	USINT	0, 2, 4, 6, 8, 10, 12
Start	INPUT	BOOL	Start counter
Stop	INPUT	BOOL	Stop the counter and clear MaxDPos and MaxDSpd value
FBENO	OUTPUT	BOOL	Function block enable output

Table 16 - RA_HSCPlugIn: Input and Output Parameters (Continued)

Parameter	Type	Data Type	Description
IDCHeck	OUTPUT	BOOL	TRUE: HSC plug-in is at selected slot FALSE: Wrong plug-in or no plug-in at selected slot
Initialized	OUTPUT	BOOL	TRUE: HSC plug-in initialization finished and ready to execute FALSE: HSCplug-in initialization not yet finished
Accumulator	OUTPUT	LINT	Accumulator value
Rate	OUTPUT	REAL	Current pulse rate The rate calculation is based on how many pulses have been counted every 10 ms.

RA_EncoderFDBK

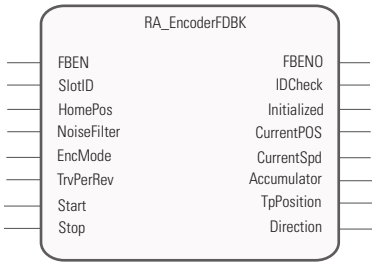
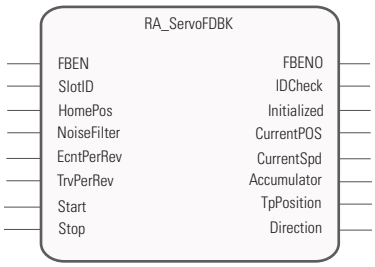


Table 17 - RA_EncoderFDBK: Input and Output Parameters

Parameter	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input
SlotID	INPUT	UINT	Plug-in slot number Slot ID = 1...5 (starting with the far left slot 1)
HomePos	INPUT	REAL	Home position Same value indicated in MC_Home instruction
NoiseFilter	INPUT	USINT	00 - No filter 01 - 250 kHz 02 - 200 kHz 03 - 80 kHz 04 - 40 kHz 05 - 13.3 kHz 06 - 10 kHz 07 - 4 kHz 08 - 2 kHz 09 - 1 kHz 10 - 500 Hz 11 - 250 Hz 12 - 125 Hz 13 - 62.5 Hz 14 - 31.25 Hz
EncMode	INPUT	USINT	Encoder mode, 1-X1, 2- X2, 4-X4
ECntPerRev	INPUT	REAL	User input to indicate how many X1 counts will be generated when Encoder disk turns one revolution.
TrvPerRev	INPUT	REAL	The actual distance travelled when motor turns one revolution
Start	INPUT	BOOL	Start counter
Stop	INPUT	BOOL	Stop the counter and clear MaxDPos and MaxDSpd value
FBENO	OUTPUT	BOOL	Function block enable output
IDCheck	OUTPUT	BOOL	TRUE: HSC plug-in is at selected slot FALSE: Wrong plug-in or no plug-in at selected slot
Initialized	OUTPUT	BOOL	TRUE: Indicates HSC initialization has finished FALSE: Indicates HSC initialization has not finished
CurrentPos	OUTPUT	REAL	Current position
CurrentSpd	OUTPUT	REAL	Current speed (Unit = user distance per second)
Accumulator	OUTPUT	LINT	Accumulator value
TpPosition	OUTPUT	REAL	Position recorded when the latest touch probe is triggered
Direction	OUTPUT	SINT	1 = Forward -1 = Reverse 0 = Not moving



RA_ServoFDBK

Table 18 - RA_ServoFDBK: Input and Output Parameters

Parameter	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input
SlotID	INPUT	UINT	Plug-in slot number Slot ID = 1...5 (starting with the far left slot 1)
HomePos	INPUT	REAL	Home position Same value indicated in MC_Home instruction
NoiseFilter	INPUT	USINT	00: No filter 01: 250 kHz 02: 200 kHz 03: 80 kHz 04: 40 kHz 05: 13.3 kHz 06: 10 kHz 07: 4 kHz 08: 2 kHz 09: 1 kHz 10: 500 Hz 11: 250 Hz 12: 125 Hz 13: 62.5 Hz 14: 31.25 Hz
EncMode	INPUT	USINT	Encoder Mode, 1-X1, 2- X2, 4-X4
ECntPerRev	INPUT	REAL	User input to indicate how many X1 counts will be generated when Encoder disk turns one revolution.
TrvPerRev	INPUT	REAL	The actual distance travelled when motor turns one revolution
Start	INPUT	BOOL	Start counter
Stop	INPUT	BOOL	Stop the counter and clear MaxDPos and MaxDSpd value
FBENO	OUTPUT	BOOL	Function block enable output
IDCheck	OUTPUT	BOOL	TRUE: HSC plug-in is at selected slot FALSE: Wrong plug-in or no plug-in at selected slot
Initialized	OUTPUT	BOOL	TRUE: Indicates HSC initialization has finished FALSE: Indicates HSC initialization has not finished
CurrentPos	OUTPUT	REAL	Current position
CurrentSpd	OUTPUT	REAL	Current speed (Unit = user distance per second)
Accumulator	OUTPUT	LINT	Accumulator value
TpPosition	OUTPUT	REAL	Position recorded when the latest touch probe is triggered
Direction	OUTPUT	SINT	1 = Forward -1 = Reverse 0 = Not moving

Use the 2080-MOT-HSC Module

For a step-by-step guide on how to use the Micro800 High-speed Counter plug-in, see Quick start Projects for 2080-MOT-HSC Plug-in on page 67.

DeviceNet Plug-in – 2080-DNET20

Overview

The DeviceNet plug-in serves as scanner and client for explicit messaging to remote devices. The module is designed to scan devices such as:

- CompactBlock LDX I/O
- PowerFlex drives
- E1 Plus overloads
- Stack lights

User-defined function blocks (UDFB) are required to enable interaction between these devices.

The 2080-DNET20 DeviceNet scanner supports a maximum of 20 nodes. For example, if the scanner ID is configured to zero, the scanner would scan from 1...20. It is supported on Micro800 controllers with available plug-in slots. Only one 2080-DNET20 DeviceNet scanner is supported per controller.

IMPORTANT	Rockwell Automation recommends that only one 2080-DNET20 DeviceNet scanner be used for each network trunk-line.

IMPORTANT	If the 2080-DNET20 DeviceNet scanner is updated to a new major firmware revision (for example, from 1.xxx to 2.xxx or vice versa), once the plug-in is successfully upgraded, power cycle the controller.

IMPORTANT	If RSLinx® browsing is enabled, the CIP client messages can get timed out because the DeviceNet buffers are fully occupied by RSLinx messages. It is recommended not to have RSLinx browsing to the DeviceNet bridge if Client Messaging is required.

Status Indicators

The DeviceNet plug-in module supports two standard DeviceNet green and red LED indicators:

- Module status
- Network status

Table 19 - Module Status Indicator

LED State	Module Status	Description
Off	No power	There is no power present.
Flashing green	Operational	Unit is starting up.
Green	Unit operational	Device is operating normally.
Flashing red	Minor fault	A recoverable fault is present or the module is undergoing firmware update.
Red	Unrecoverable fault	A nonrecoverable fault is detected.

Table 20 - Network Status Indicator

LED State	Module Status	Description
Off	No power or offline	There is no network power or the device is not operating.
Flashing green	Idle	No valid network connection has been made.

Table 20 - Network Status Indicator (Continued)

LED State	Module Status	Description
Green	Online	The plug-in module is operating normally and receiving messages.
Flashing red	Connection timeout	One or more network connections have timed out.
Red	Critical link failure	The plug-in module has detected an error that makes it incapable of communicating on the link (Bus Off or duplicate MAC_ID).

Network Configuration

To configure the DeviceNet plug-in and scan the network, you must import user-defined function blocks (UDFBs) into your Micro800 project in Connected Components Workbench software. Autoscan is used to add nodes into the scan list.

IMPORTANT It is recommended that when Autoscan is running for the nodes in range, or for the connection to be established, the nodes should be idle without any pre-occupied connections requests.

Network Wiring

The DeviceNet specifications provide for maximum network distances for the main trunk line and drop lines, depending upon the baud rate used on the network.

Table 21 - Network Specifications

Baud Rate	Trunk Line Length		Drop Length			
	Maximum Distance		Maximum		Cumulative	
	Meters	Feet	Meters	Feet	Meters	Feet
125k baud	420	1377	6	20	156	512
250k baud	200	656.17	6	20	78	256
500k baud	75	246	6	20	39	128

IMPORTANT Maximum power supply drop cable length is 3 m (10 ft).

Recommended Cable
Flat cable (KwikLink lite)
- Class 1 cable, maximum allowable current 8 A (NEC/CE code)
- Class 2 cable, maximum allowable current 4 A (NEC/CE code)

DeviceNet Switches

Figure 18 - 2080-DNET20 Assembly Diagram

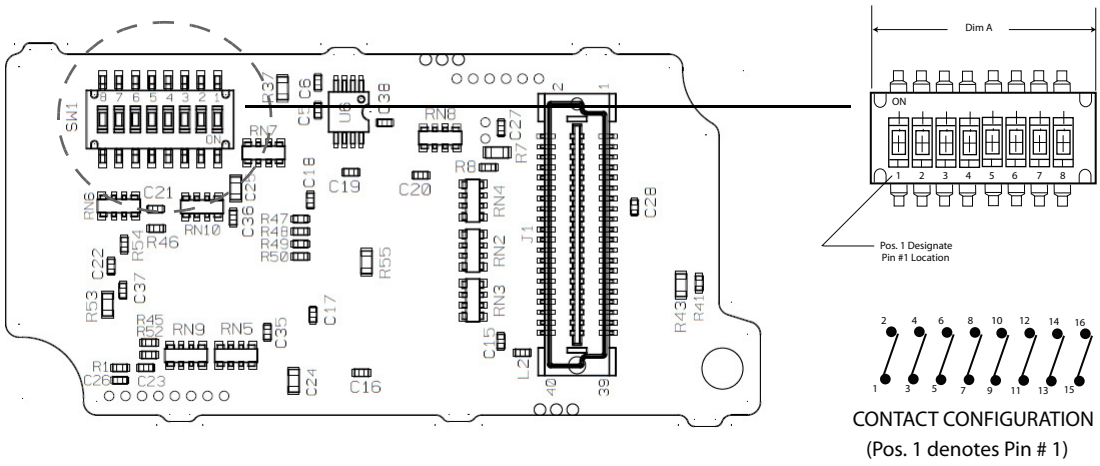


Table 22 - DeviceNet Address (MAC_ID) Switch Definitions

Node Address	SW1 Switch Positions					
	3	4	5	6	7	8
	Switch Position Values					
	32	16	8	4	2	1
0 (default)	OFF	OFF	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	OFF	ON	ON
4	OFF	OFF	OFF	ON	OFF	OFF
5	OFF	OFF	OFF	ON	OFF	ON
...						
62	ON	ON	ON	ON	ON	OFF
63	ON	ON	ON	ON	ON	ON

Table 23 - DeviceNet Baud Rate Switch Definitions

Baud Rate DR (Data Rate)	SW1 Switch Position	
	1	2
125k	OFF	OFF
250k	OFF	ON
500k (default)	ON	OFF
Autobaud	ON	ON

IMPORTANT For most applications, Rockwell Automation recommends that you use default node and baud rate settings. The DeviceNet scanner plug-in is at node 0 and the devices are at nodes 1...20. The baud rate is at 500k baud and the maximum trunkline length is 75 m (246 ft) (KwikLink lite).

Power Supply

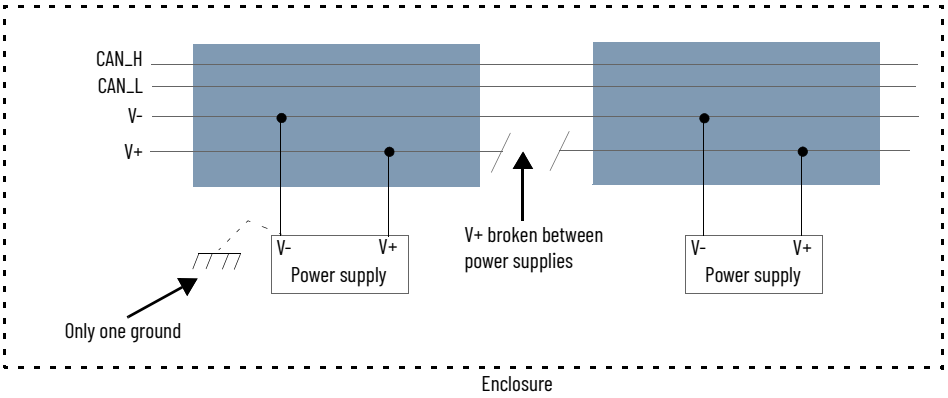
The plug-in module gets its power from the Micro800 backplane. However, the DeviceNet interface is isolated from the Micro800 system. Therefore, network power to operate the DeviceNet transceiver on the plug-in module is supplied by an external DeviceNet power supply.

If using one power supply in the network, calculate the total current requirement of all devices in the network and add +10% for current surge. Recommended power supply is the 1606-XLSDNET4.

Table 24 - Power Supply Cable Dropline Length

Dropline Length	Allowable Current
1.5 m (5 ft)	3 A
2 m (6 ft)	2 A
3 m (10 ft)	1.5 A
4.5 m (15 ft)	1 A
6 m (20 ft)	0.75 A

If two or more power supplies are connected to the KwikLink lite media (trunk cable) V+ should be broken between the two power supplies.



Grounding the network

If grounding at only one location, it is recommended that you ground at the center of the network.

Single Source Power Supply (End segment) KwikLink Lite Cable

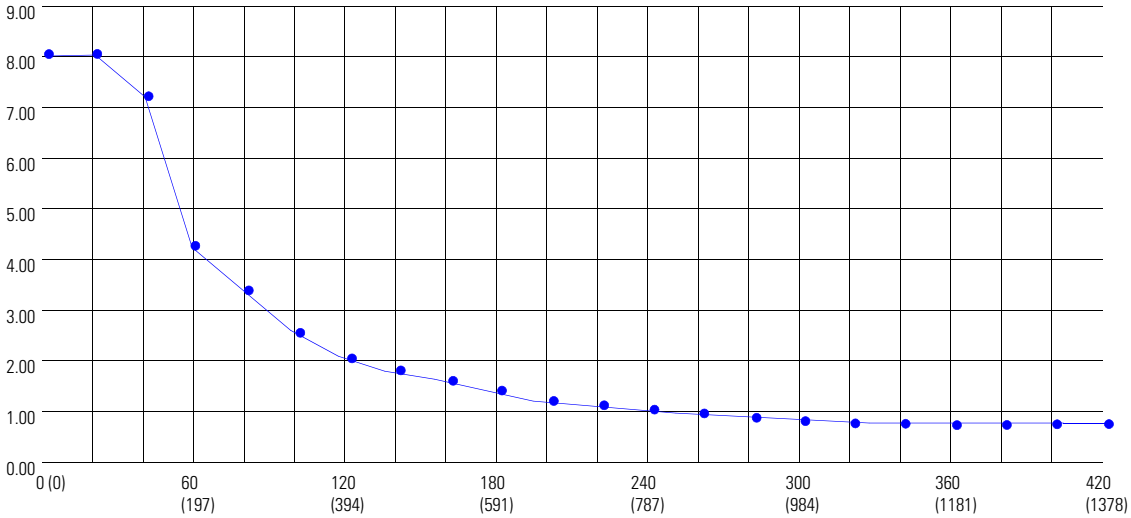


Table 25 - Single Source Power Supply – Trunkline Length and Maximum Current

Network Length in Meters (ft)	Current, Max
0 (0)	8.00(1)
20 (66)	8.00(1)
40 (131)	7.01(1)
60 (197)	4.72(1)
80 (262)	3.56
100 (238)	2.86
120 (394)	2.39
140 (459)	2.05
160 (525)	1.79
180 (591)	1.60
200 (656)	1.44

Network Length in Meters (ft)	Current, Max
220 (722)	1.31
240 (787)	1.20
260 (853)	1.11
280 (919)	1.03
300 (984)	0.96
320 (1050)	0.90
340 (1115)	0.85
360 (1181)	0.80
380 (1247)	0.76
400 (1312)	0.72
420 (1378)	0.69

(1) Exceeds NEC CL2/CE Code 4 A limit.

Dual Source Power Supply (both ends – KwikLink Lite Cable)

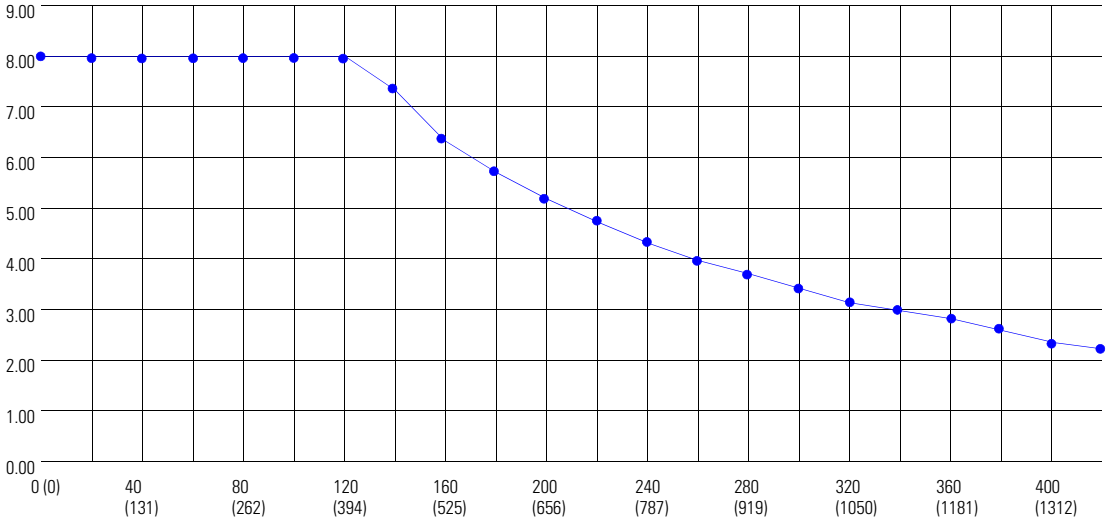


Table 26 - Dual source power supply (both ends – KwikLink Lite Cable)

Network Length in Meters (ft)	Current, Max	Network Length in Meters (ft)	Current, Max
0 (0)	8.00[1]	220 (722)	4.69
20 (66)	8.00	240 (787)	4.30
40 (131)	8.00	260 (853)	3.97
60 (197)	8.00	280 (919)	3.69
80 (262)	8.00	300 (984)	3.44
100 (328)	8.00	320 (1050)	3.23
120 (394)	8.00	340 (1115)	3.04
140 (459)	7.35	360 (1181)	2.87
160 (525)	6.43	380 (1247)	2.72
180 (591)	5.72	400 (1312)	2.59
200 (656)	5.16	420 (1378)	2.46

(1) Exceeds NEC CL2/CE Code 4 A limit.

Calculate Voltage Requirement

SUM {[(Ln * (Rc)) + (Nt * (0.005))] * In} < 4.65V

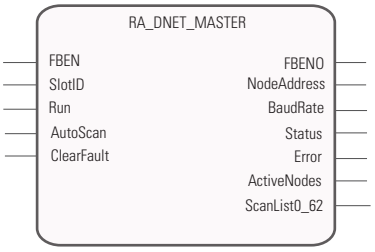
Where:
Ln = Length in meters or feet
Rc = Resistance of the cable per meter or feet
(KwikLink flat media = 0.019 ohms/meter or 0.0058 ohms/feet)
Nt = Number of the node starting from 1 that is closest to the power supply and increasing
0.005 = Nominal contact resistance that is used for every connection to the trunkline
In = Current that is drawn from the cable system by the device

IMPORTANT To calculate for percentage of loading, divide the total voltage that is calculated from the previous formula by 4.65.

User-defined Function Blocks

Download the following 2080-DNET20 user-defined function blocks (UDFB) from the Rockwell Automation Sample Code Library website.

RA_DNET_MASTER



This UDFB sets the 2080-DNET20 scanner to RUN mode.

Table 27 - RA_DNET_MASTER: Input and Output Parameters

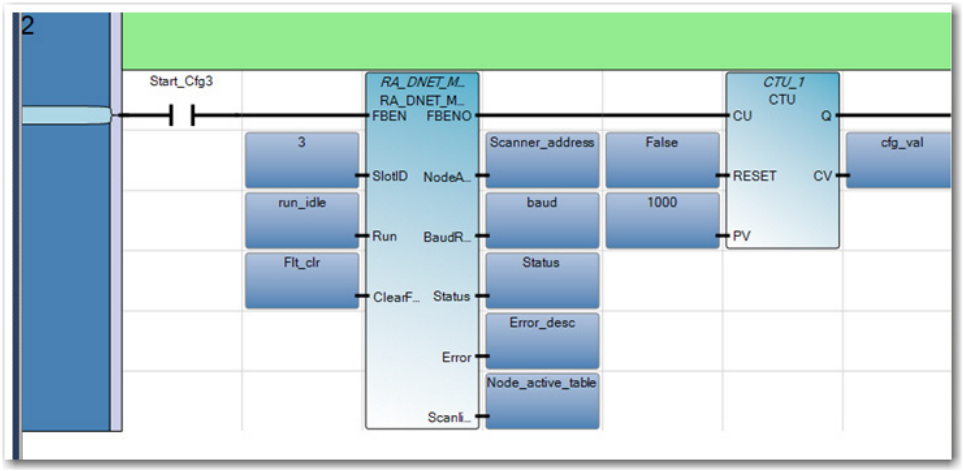
Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	TRUE: To continue reading and writing the scanner status. FBEN changed to level Triggered.
SlotID	INPUT	UINT	Plug-in slot number (1...5)
Run	INPUT	BOOL	TRUE: Set the scanner to RUN mode. FALSE: Scanner is in IDLE mode. AutoScan is enabled.
AutoScan	INPUT	BOOL	TRUE: AutoScan is enabled.
ClearFault	INPUT	BOOL	TRUE: Clear scanner fault. FALSE: No action
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
NodeAddress	OUTPUT	USINT	Scanner node address. Default node address is 0.
BaudRate	OUTPUT	USINT	Network baud rate: 0: 125K 1: 250K 2: 500K 3: AutoBaud Default baud rate is 500K.
Status	OUTPUT	USINT	Scanner fault status: 0: No error
Error	OUTPUT	STRING	Scanner error description
ActiveNodes	OUTPUT	USINT	Number of slave nodes in the network
Scanlist0_62	OUTPUT	LWORD	Details on active node table, bit 0...62. Bit 0: Represent Node 0 Bit 62: Represent Node 62

Table 28 - Sequence of Operation: RA_DNET_MASTER

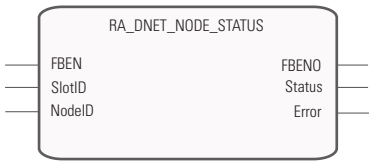
Sequence	Run	Autoscan	Description
1	False	False	Reinitializes scan list from the plug-in scanner if FBEN = TRUE
2	False	True	Triggers autoscan to scan the network after clearing scan list
3	False	False	Puts scanner to IDLE mode by disabling autoscan if active node number = number of nodes in network
4	True	False	Puts scanner to RUN mode

Upon powerup, the scanner is in IDLE Mode for the autoscan to start. Wait until the autoscan process is complete before turning the scanner to RUN Mode (that is, Run bit is TRUE).

Sample Code



RA_DNET_NODE_STATUS

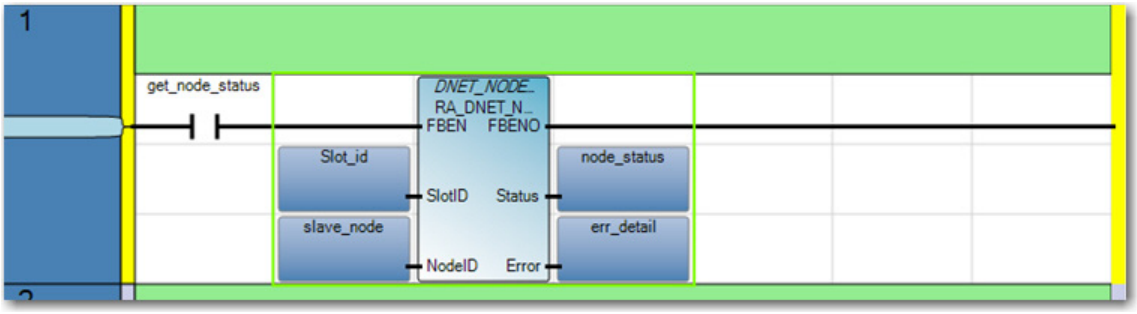


This UDFB is used to read the node status of slave nodes in a DeviceNet network where the 2080-DNET20 scanner is connected.

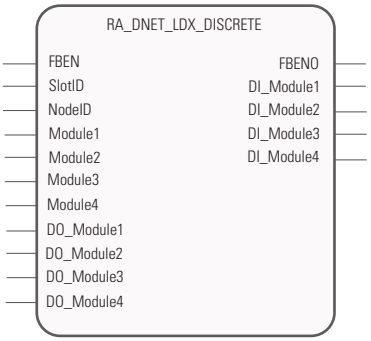
Table 29 - RA_DNET_NODE_STATUS: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function
SlotID	INPUT	UINT	Plug-in slot number (1...5)
NodeID	INPUT	USINT	Slave node address
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
Status	OUTPUT	USINT	Scanner fault status 0: No errors
Error	OUTPUT	STRING	Description of the node status error

Sample Code: RA_DNET_NODE_STATUS



RA_DNET_LDX_DISCRETE



This UDFB is used for I/O data exchange with digital CompactBlock LDX I/O modules.

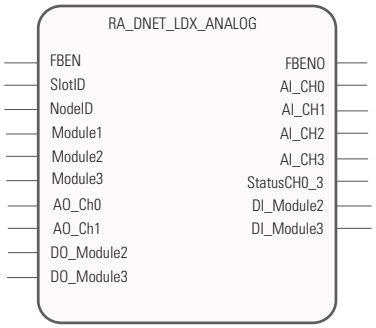
Table 30 - RA_DNET_LDX_DISCRETE: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function block
SlotID	INPUT	UINT	Plug-in slot number (1...5)
NodeID	INPUT	USINT	Node address of the digital CompactBlock LDX I/O slave node
Module1	INPUT	STRING	Base module I/O configuration INPUT X OUTPUT Channels For example: 16X0 (16 input / 0 output is physically present as base module) Valid String: 32X0, 0X32, 16X0, 0X16, 16X16, 8X8, 8X0, 0X8, 0X6 NOTE: X should always be uppercase.
Module2	INPUT	STRING	Expansion module 1 I/O configuration INPUT X OUTPUT channels For example: 16X0 (16 input / 0 output is physically present as base module) Valid String: 32X0, 0X32, 16X0, 0X16, 16X16, 8X8, 8X0, 0X8, 0X6 NOTE: X should always be uppercase.
Module3	INPUT	STRING	Expansion module 2 I/O configuration INPUT X OUTPUT Channels For example: 16X0 (16 input / 0 Output is physically present as base module) Valid String: 32X0, 0X32, 16X0, 0X16, 16X16, 8X8, 8X0, 0X8, 0X6 NOTE: X should always be uppercase.

Table 30 - RA_DNET_LDX_DISCRETE: Input and Output Parameters (Continued)

Variable Name	Type	Data Type	Description
Module4	INPUT	STRING	Expansion module 3I/O configuration INPUT X OUTPUT Channels For example: 16X0 (16 input / 0 output is physically present as base module) Valid String: 32X0, 0X32, 16X0, 0X16, 16X16, 8X8, 8X0, 0X8, 0X6 NOTE: X should always be uppercase.
DO_Module1	INPUT	UDINT	Output data for base module
DO_Module2	INPUT	UDINT	Output data for expansion module 1
DO_Module3	INPUT	UDINT	Output data for expansion module 2
DO_Module4	INPUT	UDINT	Output data for expansion module 3
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
DI_Module1	OUTPUT	UDINT	Input data from base module (Module 1)
DI_Module2	OUTPUT	UDINT	Input data from expansion module 1 (Module 2)
DI_Module3	OUTPUT	UDINT	Input data from expansion module 2 (Module 3)
DI_Module4	OUTPUT	UDINT	Input data from expansion module 3 (Module 4)

RA_DNET_LDX_ANALOG



This UDFB is used for data exchange with analog CompactBlock LDX I/O modules.

Table 31 - RA_DNET_LDX_ANALOG: Input and Output Parameters

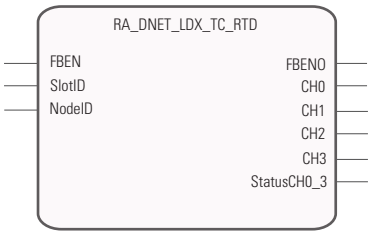
Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function
SlotID	INPUT	UINT	Plug-in slot number (1...5)
NodeID	INPUT	USINT	Slave node address
Module1[(1)]	INPUT	STRING	Analog base module I/O configuration INPUT X OUTPUT Channels For example: 4X0 (4 input analog module is physically present as base module) Valid String: 0X2, 4X0 NOTE: X should always be upper case.
Module2[(1)]	INPUT	STRING	Digital expansion module 1 I/O configuration INPUT X OUTPUT Channels For example: 16X16 (16 input / 16 output is physically present as expansion module 1) Valid String: 16X0, 0X16, 16X16, 8X8, 8X0, 0X8, 0X6 NOTE: X should always be upper case.
Module3[(1)]	INPUT	STRING	Digital expansion module 2 I/O configuration INPUT X OUTPUT channels For example: 16X16 (16 input / 16 output is physically present as expansion module 2) Valid String: 16X0, 0X16, 16X16, 8X8, 8X0, 0X8, 0X6 NOTE: X should always be upper case.
AO_Ch0	INPUT	WORD	Analog Output Channel 0 value This value is valid only if Module1 = '0X2'
AO_Ch0	INPUT	WORD	Analog Output Channel 1 value. This value is valid only if Module1 = '0X2'
DO_Module2	INPUT	UINT	Output data for expansion I/O module 1
DO_Module3	INPUT	UINT	Output Data for expansion I/O module 2
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
AI_CH0	OUTPUT	WORD	Analog Input Channel 0 value This value is valid only if Module 1 = '4X0'
AI_CH1	OUTPUT	WORD	Analog Input Channel 1 value. This value is valid only if Module 1 = '4X0'
AI_CH2	OUTPUT	WORD	Analog Input Channel 2 value. This value is valid only if Module 1 = '4X0'

Table 31 - RA_DNET_LDX_ANALOG: Input and Output Parameters (Continued)

Variable Name	Type	Data Type	Description
AI_CH3	OUTPUT	WORD	Analog Input Channel 3 Value This value is valid only if Module 1 = '4X0'
StatusCH0_3	OUTPUT	WORD	Analog input channel 0...3 status
DI_Module2	OUTPUT	UINT	Digital Expansion Module 1 Input Data Applicable only if catalog is with digital inputs.
DI_Module3	OUTPUT	UINT	Digital Expansion Module 2 Input Data Applicable only if catalog is with digital inputs.

(1) Use only valid strings combinations as mentioned in the table. If the physical I/O of Module 1, Module 2, and Module 3 do not match the physical I/O present in the base and expansion, then the incorrect sequence is written.

RA_DNET_LDX_TC_RTD

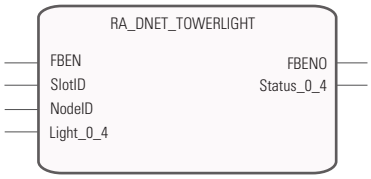


This UDFB is used to read input data from the Thermocouple/RTD module.

Table 32 - RA_DNET_LDX_TC_RTD: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable function
SlotID	INPUT	UINT	Plug-in slot number (1...5)
NodeID	INPUT	USINT	Node address of the digital Compact I/O slave node
FBENO	OUTPUT	BOOL	Function block enable output
CH0	OUTPUT	WORD	RTD/Thermocouple input channel 0 value
CH1	OUTPUT	WORD	RTD/Thermocouple input channel 1 value
CH2	OUTPUT	WORD	RTD/Thermocouple input channel 2 value
CH3	OUTPUT	WORD	RTD/Thermocouple input channel 3 value
StatusCH0_3	OUTPUT	WORD	RTD/Thermocouple Input channel 0...3 status

RA_DNET_TOWERLIGHT

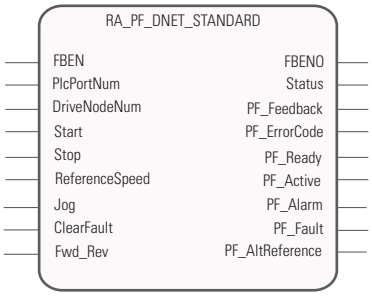


This UDFB is used for data exchange with a tower light or stack light.

Table 33 - RA_DNET_TOWERLIGHT: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function
SlotID	INPUT	UINT	Plug-in slot number (1...5)
NodeID	INPUT	USINT	Tower light node address
Light_0_4	INPUT	USINT	Light 0...4, for example: Bit 0: Blue Bit 1: Yellow Bit 2: Red
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
Status_0_4	OUTPUT	USINT	Light 0...4 status

RA_PF_DNET_STANDARD



This UDFB is used for I/O data exchange with standard PowerFlex drives configured as single mode.

Table 34 - RA_PF_DNET_STANDARD: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function
PlcPortNum	INPUT	UINT	Plug-in slot number (1...5 for plug-in slots)
DriveNodeNum	INPUT	USINT	Slave node address for PowerFlex drive
Start	INPUT	BOOL	TRUE to start PowerFlex drive

Table 34 - RA_PF_DNET_STANDARD: Input and Output Parameters (Continued)

Variable Name	Type	Data Type	Description
Stop	INPUT	BOOL	TRUE to stop PowerFlex drive
ReferenceSpeed	INPUT	REAL	Reference speed for the device Configure PowerFlex drive speed
Jog	INPUT	BOOL	TRUE to enable jog in PowerFlex drive
ClearFault	INPUT	BOOL	TRUE to clear fault in PowerFlex drive
Fwd_Rev	INPUT	BOOL	TRUE to configure PowerFlex drive for forward motion FALSE to configure PowerFlex drive for reverse motion
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
Status	OUTPUT	BOOL	PowerFlex drive status
PF_Feedback	OUTPUT	REAL	Feedback from the PowerFlex drive
PF_ErrorCode	OUTPUT	INT	For future use
PF_Ready	OUTPUT	BOOL	Ready bit from PowerFlex drive
PF_Active	OUTPUT	BOOL	Active bit from PowerFlex drive
PF_Alarm	OUTPUT	BOOL	Alarm bit from PowerFlex drive
PF_Fault	OUTPUT	BOOL	Fault bit from PowerFlex drive
PF_AltReference	OUTPUT	BOOL	Alt Reference bit from PowerFlex drive

IMPORTANT Ensure that your PowerFlex drives settings are correct. For basic setup configuration, see the user manual for your PowerFlex drives. You can download the manual from rok.auto/literature.

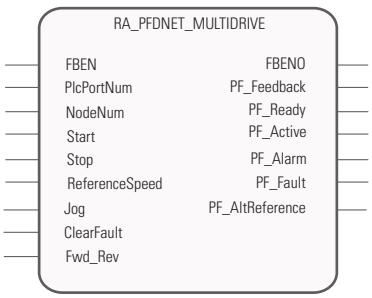
With PowerFlex 523 drives, you must multiply the speed reference and divide the speed feedback by a factor of 10.0 to get the correct value. The PowerFlex 4-series and PowerFlex 520-series drives have a different multiplier.

Table 35 - Reference and Feedback for the Different PowerFlex Drives

Drive Type	Number	Reference	Feedback
PowerFlex 4M	132	x10	x0.1
PowerFlex 4	39	x10	x0.1
PowerFlex 40	40	x100	x0.1
PowerFlex 40P	41	x100	x0.01
PowerFlex 400	129	x100	x0.01
PowerFlex 523	8	x100	x0.01
PowerFlex 525	9	x100	x0.01

For example, if you set reference speed at 50, command speed is 50 Hz for PowerFlex 4M drives and only 5 Hz for PowerFlex 523 and PowerFlex 525 drives.

RA_PF_DNET_MULTIDRIVE



This UDFB is used for I/O data exchange with standard PowerFlex drives, which are configured as multi-drive.

Table 36 - RA_PF_DNET_MULTIDRIVE: Input and Output Parameters

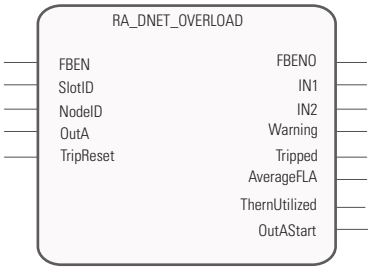
Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function
PlcPortNum	INPUT	UINT	Plug-in slot number (1...5 for plug-in slots)
NodeNum	INPUT	USINT	DeviceNet node address for PowerFlex drive (connected as master in multi-drive setup)
Start	INPUT	BOOL[1...5]	TRUE to start each element of the array. Corresponds to each drive. For example: Start[1] for Drive 1 and Start[5] for Drive5.

Table 36 - RA_PF_DNET_MULTIDRIVE: Input and Output Parameters (Continued)

Variable Name	Type	Data Type	Description
Stop	INPUT	BOOL[1...5]	TRUE to stop each element of the array. Corresponds to each drive, for example, Stop [1] for Drive 1 and Stop [5] for Drive5.
ReferenceSpeed	INPUT	REA[1...5]L	Reference speed to set the device speed. Each element of the array corresponds to each drive, for example, Reference Speed [1] for Drive 1 and Reference Speed [5] for Drive5.
Jog	INPUT	BOOL[1...5]	TRUE to enable jog in PowerFlex drive. Each element of the array corresponds to each drive, for example, Reference Jog [1] for Drive 1 and Jog [5] for Drive5.
ClearFault	INPUT	BOOL[1...5]	TRUE to clear fault in PowerFlex drive. Each element of the array corresponds to each drive, for example, ClearFault [1] for Drive 1 and ClearFault [5] for Drive5.
Fwd_Rev	INPUT	BOOL[1...5]	TRUE to configure PowerFlex drive for forward motion. FALSE to configure PowerFlex drive for Reverse motion. Each element of the array corresponds to each drive, for example, Fwd_Rev [1] for Drive 1 and Fwd_Rev [5] for Drive5.
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
PF_Feedback	OUTPUT	REAL[1...5]	Speed reference from the PowerFlex drive. Each element of the array corresponds to each drive, for example, PF_Feedback[1] for Drive 1 and PF_Feedback[5] for Drive5.
PF_Ready	OUTPUT	BOOL[1...5]	Ready bit from PowerFlex drive. Each element of the array corresponds to each drive, for example, PF_Ready[1] for Drive 1 and PF_Ready[5] for Drive5.
PF_Active	OUTPUT	BOOL[1...5]	Active bit from PowerFlex drive. Each element of the array corresponds to each drive, for example, PF_Active[1] for Drive 1 and PF_Active[5] for Drive5.
PF_Alarm	OUTPUT	BOOL[1...5]	Alarm bit from PowerFlex drive. Each element of the array corresponds to each drive, for example, PF_Alarm[1] for Drive 1 and PF_Alarm[5] for Drive5.
PF_Fault	OUTPUT	BOOL[1...5]	Fault bit from PowerFlex drive. Each element of the array corresponds to each drive, for example, PF_Fault[1] for Drive 1 and PF_Fault[5] for Drive5.
PF_AltReference	OUTPUT	BOOL[1...5]	Alt Reference bit from PowerFlex drive. Each element of the array corresponds to each drive, for example, PF_AltReference[1] for Drive 1 and PF_AltReference[5] for Drive5.

IMPORTANT Verify that your PowerFlex drives settings are correct. For basic setup configuration, see the user manual for your PowerFlex drives. You can download the manual from rok.auto/literature.

RA_DNET_OVERLOAD



This UDFB is used for I/O data exchange with an overload relay.

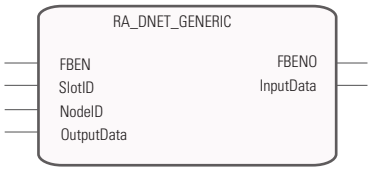
Table 37 - RA_DNET_OVERLOAD: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable the function
SlotID	INPUT	UINT	Plug-in slot number (1...5 for plug-in slots)
NodeID	INPUT	USINT	DeviceNet node address of the slave node
OutA	INPUT	BOOL	TRUE to turn on Output A

Table 37 - RA_DNET_OVERLOAD: Input and Output Parameters (Continued)

Variable Name	Type	Data Type	Description
TripReset	INPUT	BOOL	TRUE to enable Trip Reset
FBENO	OUTPUT	BOOL	Function block enable output TRUE upon exit
IN1	OUTPUT	BOOL	Input 1 from overload relay
IN2	OUTPUT	BOOL	Input 2 from overload relay
Warning	OUTPUT	BOOL	TRUE if warning is enabled
Tripped	OUTPUT	BOOL	TRUE if tripped
AverageFLA	OUTPUT	WORD	Average FLA % value from overload relay
ThermUtilized	OUTPUT	WORD	Therm Utilized value from overload relay
OutAStatus	OUTPUT	WORD	Average FLA value from overload relay

RA_DNET_GENERIC

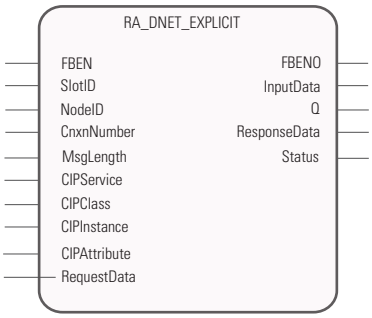


This UDFB is used for I/O data exchange with generic I/O devices.

Table 38 - RA_DNET_GENERIC: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable function
SlotID	INPUT	UINT	Plug-in slot number (1...5 for plug-in slots)
NodeID	INPUT	USINT	Slave node address
OutputData	INPUT	USINT[1...64]	Slave output data
FBENO	OUTPUT	BOOL	Function block enable output TRUE to enable function
InputData[1...64]	OUTPUT	USINT[1...64]	Input data from slave

RA_DNET_EXPLICIT



This UDFB is used for sending explicit messages to the slave node.

IMPORTANT For DNET explicit messages, the maximum payload supported is 256 bytes.

Table 39 - RA_DNET_EXPLICIT: Input and Output Parameters

Variable Name	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input TRUE to enable function
SlotID	INPUT	UINT	Plug-in slot number (1...5 for plug-in slots)
NodeID	INPUT	USINT	Node address of slave node
CnxnNumber	INPUT	USINT	Connection number values 1, 2, 3, 4, 5
MsgLength	INPUT	USINT	Specifies the size of the CIP message in the transaction block
CIPService	INPUT	UINT	CIP service code
CIPClass	INPUT	UINT	CIP Class code (valid values 0...65535)
CIPInstance	INPUT	UINT	CIP instance (valid values 0...65535)
CIPAttribute	INPUT	UINT	CIP attribute (valid values 0...65535)
RequestData	INPUT	USINT[1...54]	Request data from slave
FBENO	OUTPUT	BOOL	Function block enable output
Q	OUTPUT	BOOL	TRUE when the message is sent out successfully

Table 39 - RA_DNET_EXPLICIT: Input and Output Parameters (Continued)

Variable Name	Type	Data Type	Description
Error	OUTPUT	BOOL	TRUE when message transmits error
ResponseData	OUTPUT	USINT [1...50]	CIP Response error Response Data[1]: Extended Error ID Response Data[2]: Error ID See Explicit Message Request Format on page 51. See Explicit Message Status Codes on page 51.
Status	OUTPUT	USINT	See Explicit Message Status Codes on page 51.

Table 40 - Transaction Block Format

Byte Offset	Contents
0	Status
1	Transaction ID
2	Size
3	Reserved
4	MAC ID
5	Service
6...115	Transaction Body (110 bytes)

Table 41 - Explicit Message Request Format

Byte Offset	Contents
0	Status
1	Transaction ID
2	Size
3	Reserved
4	MAC ID
5	Service
6...7	Class
8...9	Instance
10...115	Service Data (106 Bytes)

Table 42 - Explicit Message Response Format

Byte Offset	Contents
0	Status
1	Transaction ID
2	Size
3	Reserved
4	MAC ID
5	Service

Byte Offset	Contents	Request Data
0	Status	Can be read from UDFB status
6...115	ServiceData	Can be read from UDFB response data Response data shows CIP error code

Table 43 - Explicit Message Status Codes

Status Code	Description
0	Ignore transaction block (block empty)
1	Transaction completed successfully
2	Transaction in progress (not ready)
4	Error – node offline
5	Error – DeviceNet port disabled/offline
6	Error – Transaction TXID unknown

Table 43 - Explicit Message Status Codes (Continued)

Status Code	Description
7	Error – Duplicate TXID
9	Error – Scanner out of buffers
12	Error – Response data too large for block
14	Error – Invalid size specified
15	Error – Device timed out
16	Block queued
17	Block allocated
18	Connection in progress
3, 8, 10, 11, 13, 19...255	Reserved

Send Explicit Messages to 2080-DNET20 Plug-in Using Micro800 Pass Through

MSG_CIPGENERIC instruction can be used to send Explicit messages to the 2080-DNET20 plug-in and the Slave nodes on the DeviceNet network.

IMPORTANT For DeviceNet messaging using MSG_CIPGENERIC, only unconnected messaging (connection mode 0) is supported. CIP connection type must be configured as 0 – unconnected.

In MSG_CIPGENERIC, configure the target path as follows:

- To access the plug-in, the format of the target path is "1, Slot number"
- To access a Slave device through the plug-in, the format of the target path is "1, Slot number, 2, DeviceNet node address"

For example, if the 2080-DNET20 plug-in is connected at physical slot 3 and the Slave device of address 40 is present in the DeviceNet network, then:

- Using MSG_CIPGENERIC to access the plug-in, the target path is "1, 3"
- Using MSG_CIPGENERIC to access the Slave node through the plug-in, the target path is "1, 3, 2, 40"

The number "1" refers to the Virtual backplane port number and "2" refers to the 2080-DNET20 plug-in's DeviceNet port number. These are fixed values. The slot number starts from 1 up to the maximum number of slots physically present in the controller.

When the controller pass-through feature is used in the following example:

Micro850 -> Micro820 [192.168.1.100] -> 2080-DNET20 plug-in [Slot 3] -> Slave device [Node 40]

- Using MSG_CIPGENERIC starting from Micro850 to access the plug-in, the target path would be "4, 192.168.1.100, 1, 3"
- Using MSG_CIPGENERIC starting from Micro850, the Slave device cannot be accessed because the MSG_CIPGENERIC target path configuration is limited to one hop. This path works if you use a Logix controller/PC instead of Micro850.

The number "4" refers to the EtherNet/IP port number (for Serial the port number varies from 2...5. See the controller's user manual for detailed information).

IMPORTANT To use the controller pass through feature, the following firmware revisions are required:

- Micro820, Micro830, Micro850, or Micro870 controller firmware revision 8.011 or later
- 2080-DNET20 plug-in firmware revision 2.011 or later

Error Codes

Table 44 - DeviceNet Plug-in Error Codes and Descriptions

ErrorID	Description
0	No errors
1	Node number not in scanlist
65	AutoScan active
70	Scanner failed DUP MAC check
71	Illegal value in scanlist
72	Device stopped communicating
73	Device does not match scanlist
74	Scanner has detected data overrun
75	No network traffic detected
76	No network traffic detected for scanner
77	Data size returned does not match scanlist
78	Device on scanlist not active on subnet
79	Scanner failed to transmit a message
80	Scanner is in Idle mode operation
81	Scanner is in fault mode operation
82	I/O fragment out of sequence
83	Device refused to be initialized
84	Device not yet initialized
85	Incorrect data size upon connection with device
86	Device/Slave went into Idle
87	Shared master has not allocated slave
88	Shared master has not allocated required choices
89	Keeper download failed
90	User has disabled scanner
91	Bus Off detected on scanner
92	No network power detected
93	CRC failure detected on one or more configuration blocks
95	Scanner application program flash is being updated
96	Port is in test mode
97	Scanner is halted by user
98	ESC, overflow, divide, or other processor error
99	Scanner watchdog has timed out

Use the 2080-DNET20 Plug-in

For a step-by-step guide on how to use the DeviceNet plug-in, see Quick Start Project for 2080-DNET20 Plug-in on page 63.

Notes:

Quick Start

This chapter provides quick start instructions on how to use your 2080-DNET20 and 2080-MOT-HSC plug-in modules.

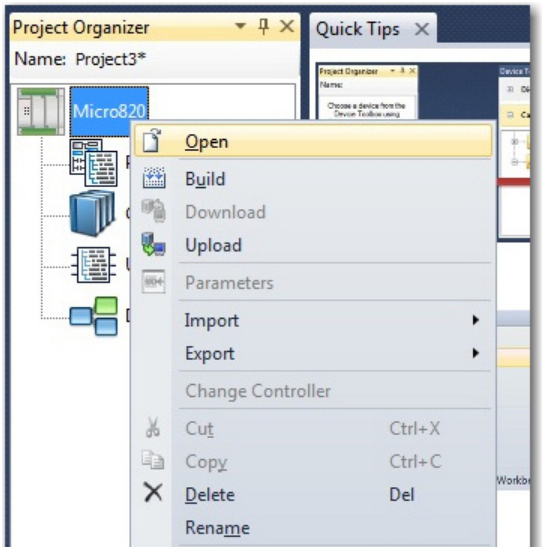
Add and Configure Plug-ins in Connected Components Workbench Software

This section shows you an example of how to configure the plug-ins through the Connected Components Workbench software.

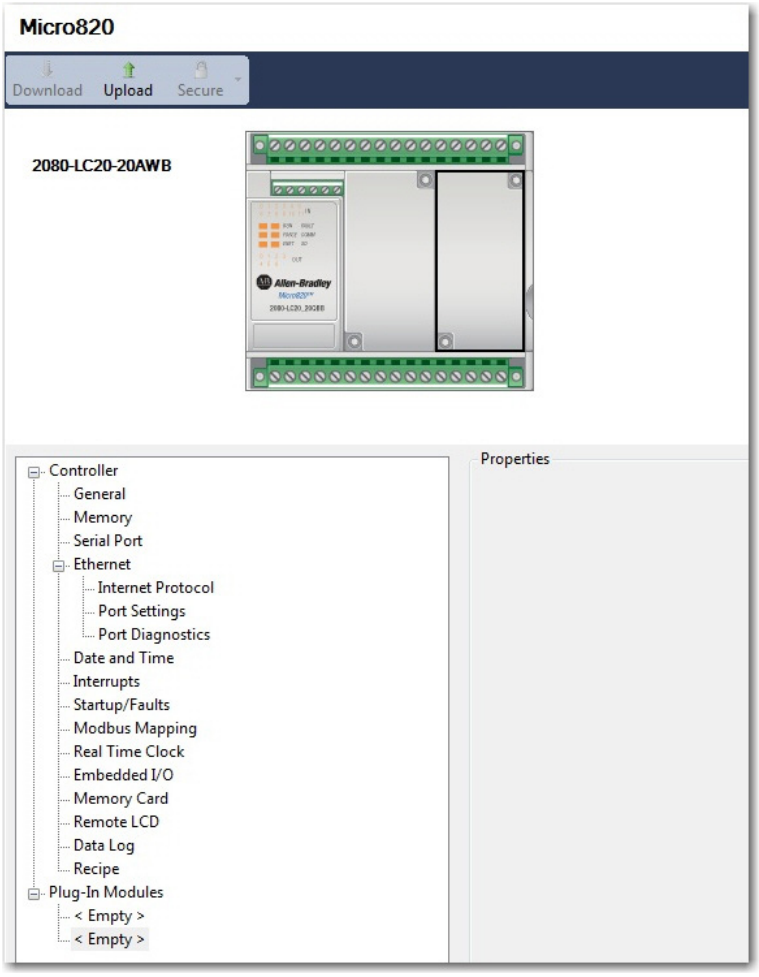
For more information about using Connected Components Workbench software, you can check out the Connected Components Workbench Online Help (it comes with the software).

The following steps show a Micro820 controller.

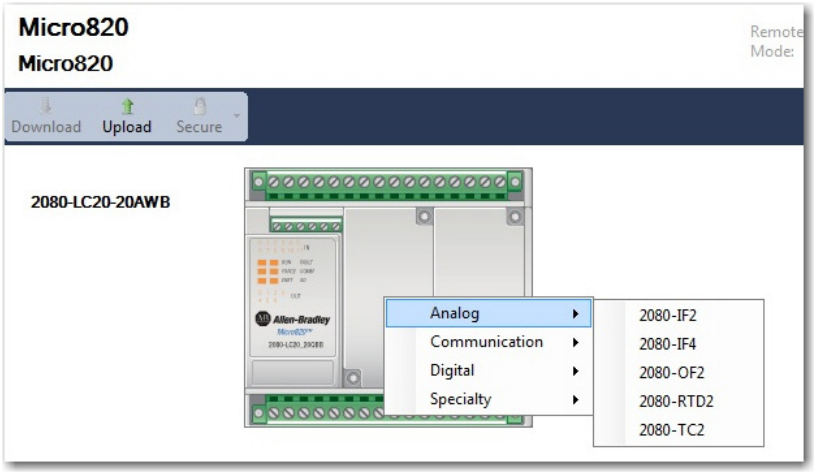
1. Launch the Connected Components Workbench software and open your Micro800 project. On the Project Organizer pane, right-click the project name and select Open.



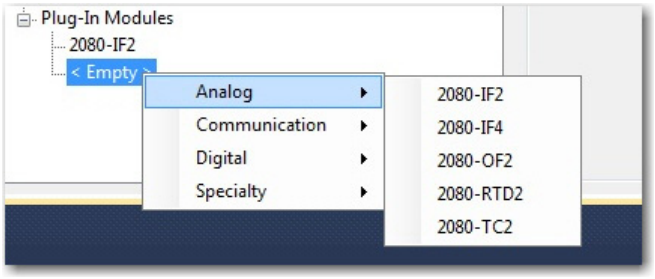
The Controller Properties page appears.



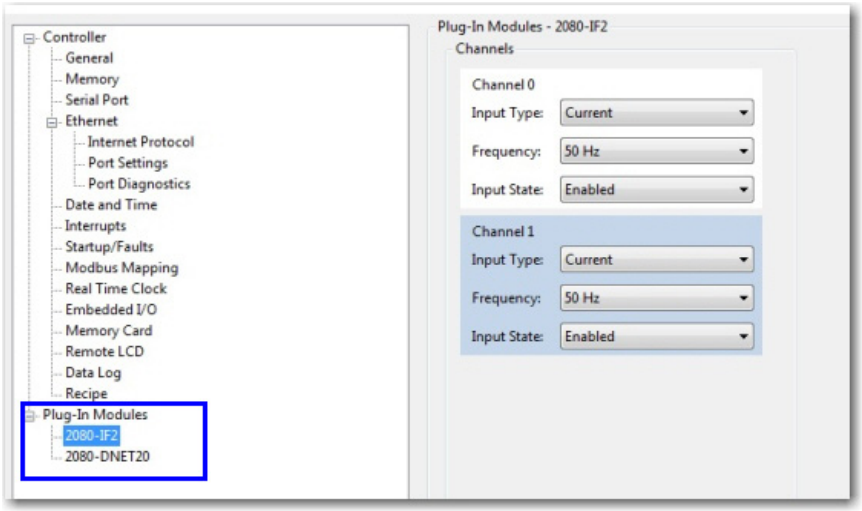
2. To add a Micro800 plug-in, you can do any of the following:
   - Right-click the plug-in slot that you would like to configure and choose the plug-in, which is shown as follows.



   - Right-click the plug-in slot in the Controller Properties tree and choose the plug-in you would like to add.



The device configuration window should show the added plug-in modules:



Browse Your 2080-DNET20 Plug-in Using RSLinx Software

There are two methods that you can use to browse for your 2080-DNET20 plug-in using RSLinx software. The first method is browsing directly to the plug-in through the DeviceNet network. The second method is browsing through a Micro820, Micro830, Micro850, or Micro870 controller using the pass through feature. This allows you to upgrade the firmware of the plug-in.

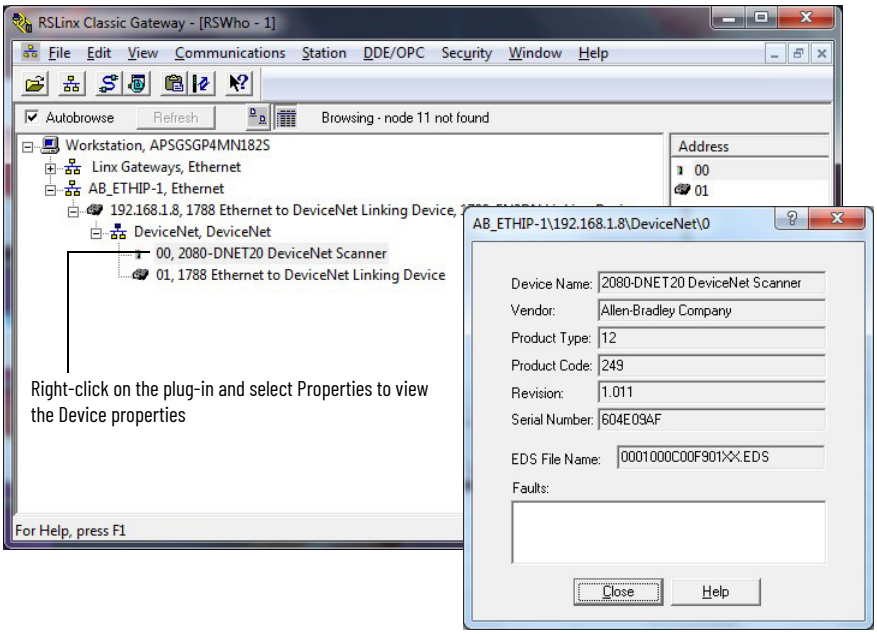
IMPORTANT To use the controller pass through feature, the following firmware revisions are required:

- Micro820, Micro830, or Micro850 controller firmware revision 8.011 or later
- 2080-DNET20 plug-in firmware revision 2.011 or later

Browse Using the DeviceNet Network

From the computer, the 2080-DNET20 plug-in can be browsed through the DeviceNet network. This method requires an additional device to connect the computer to the DeviceNet network. For example, you can use the 1788-EN2DN or 1784-U2DN devices. For instructions on connecting the plug-in to the DeviceNet network, see Setup and Wiring on page 63.

Figure 19 - Browsing the 2080-DNET20 Plug-in Through the DeviceNet Network



Browse Using the Micro800 Pass Through Feature

IMPORTANT To use the controller pass through feature, the following firmware revisions are required:

- Micro820, Micro830, or Micro850 controller firmware revision 8.011 or later
- 2080-DNET20 plug-in firmware revision 2.011 or later

Figure 20 - Browsing the 2080-DNET20 Plug-in from the Controller Backplane Through EtherNet/IP

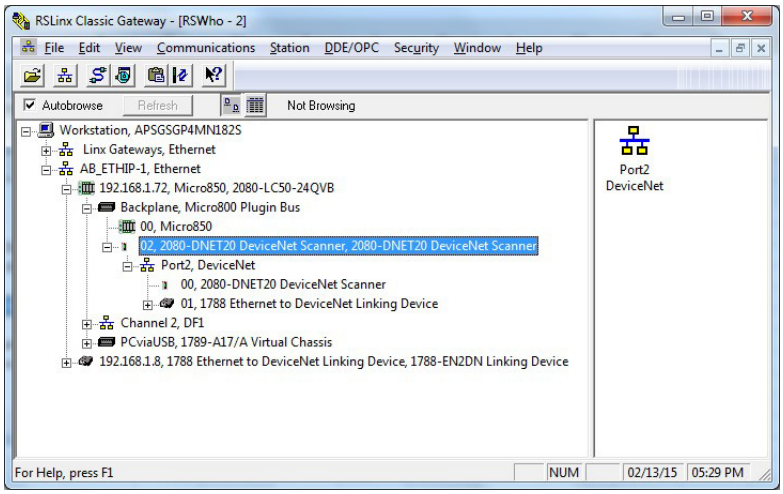


Figure 21 - Browsing the 2080-DNET20 Plug-in from the Controller Backplane Through USB

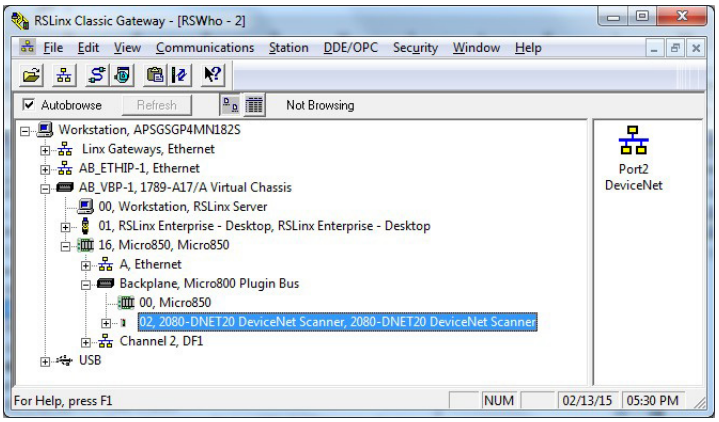
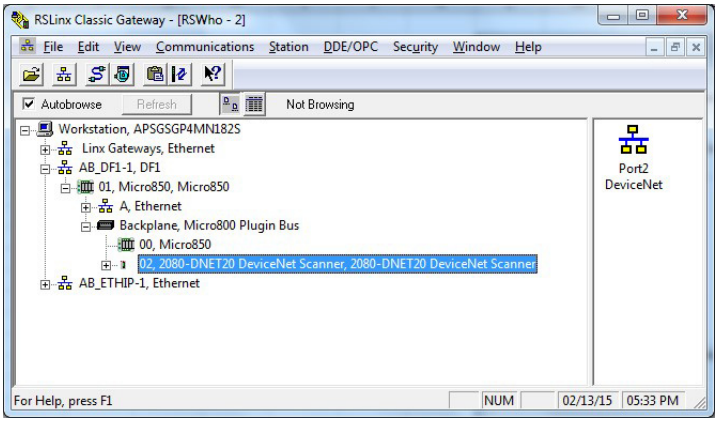


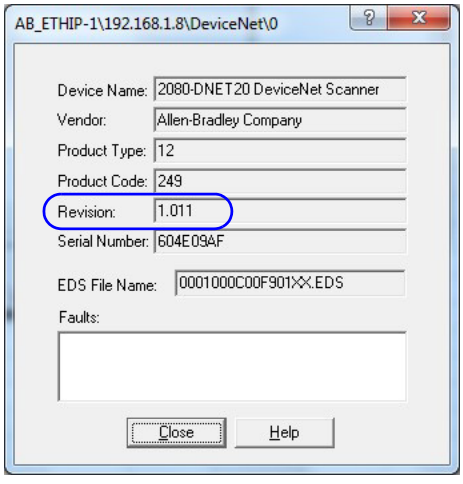
Figure 22 - Browsing the 2080-DNET20 Plug-in from the Controller Backplane Through Serial DF1



Update Your 2080-DNET20 Plug-in Firmware

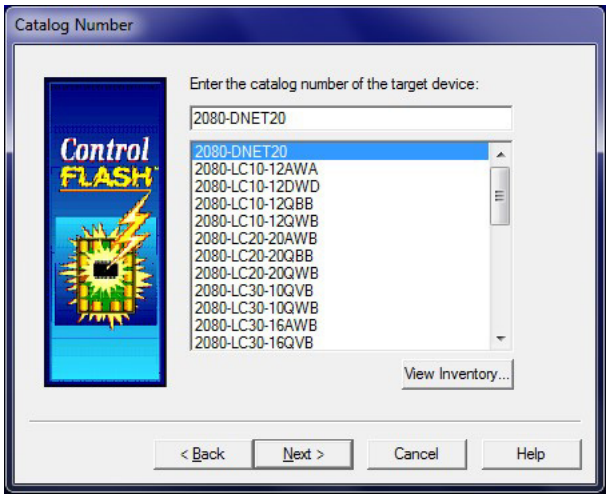
This quick start shows you how to update the firmware in a 2080-DNET20 plug-in using the ControlFLASH™ software. The ControlFLASH software is installed or updated when Connected Components Workbench software is installed on your computer. It is recommended that the controller is in Program mode and the plug-in is in Idle mode when performing the update.

1. Check the firmware revision of the plug-in.
   To do this, check the Device Properties of the plug-in RSLinx software.

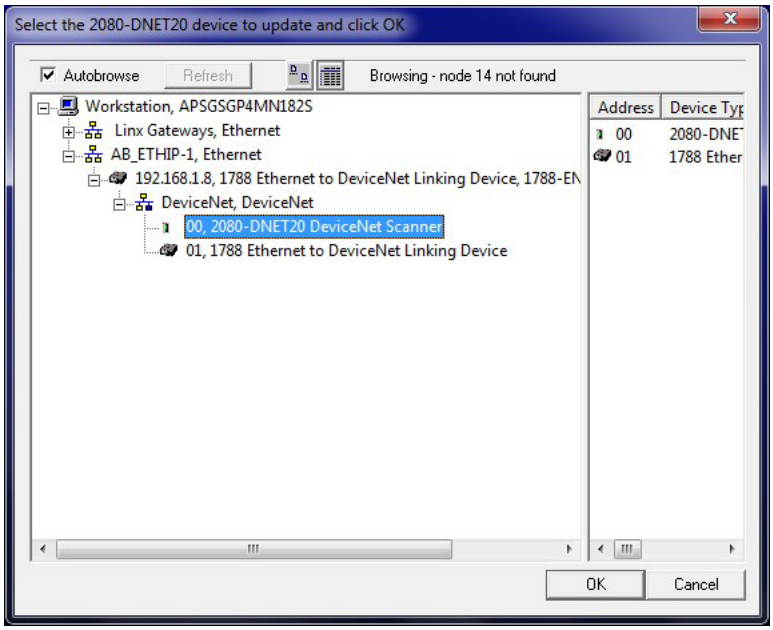


2. Launch the ControlFLASH software and select Next.

3. In the Catalog Number dialog, select the 2080-DNET20 plug-in and select Next.

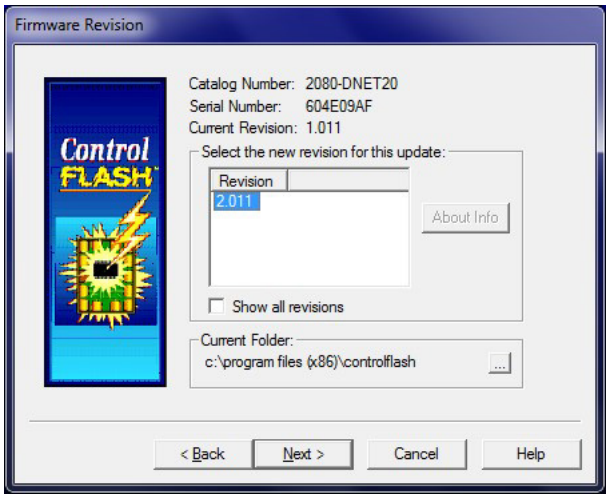


4. The Connection Browser dialog appears, select the 2080-DNET20 plug-in and select OK.

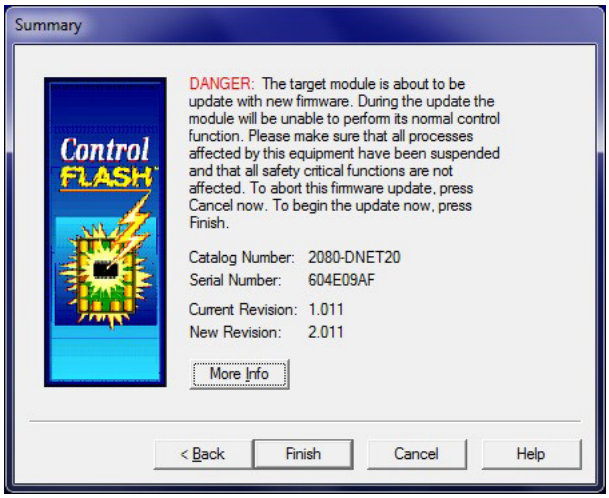


IMPORTANT To update from firmware revision 1.012 to 2.011, the DeviceNet network should be used. From revision 2.011 or later, you can update the firmware using the controller pass through feature.

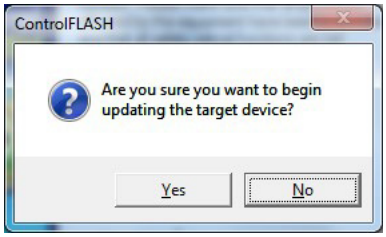
5. Select the firmware revision to update and select Next.



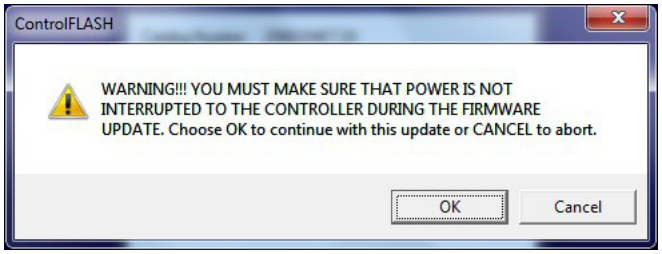
6. Verify your selection and select Finish.



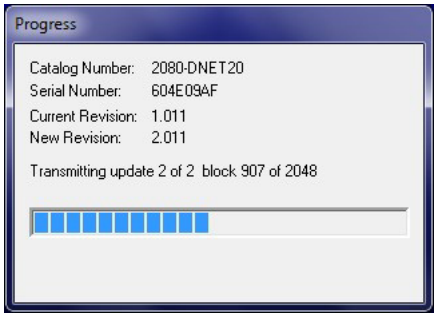
7. Select Yes to continue.



8. Select OK to begin the upgrade process.



A dialog showing the progress appears.



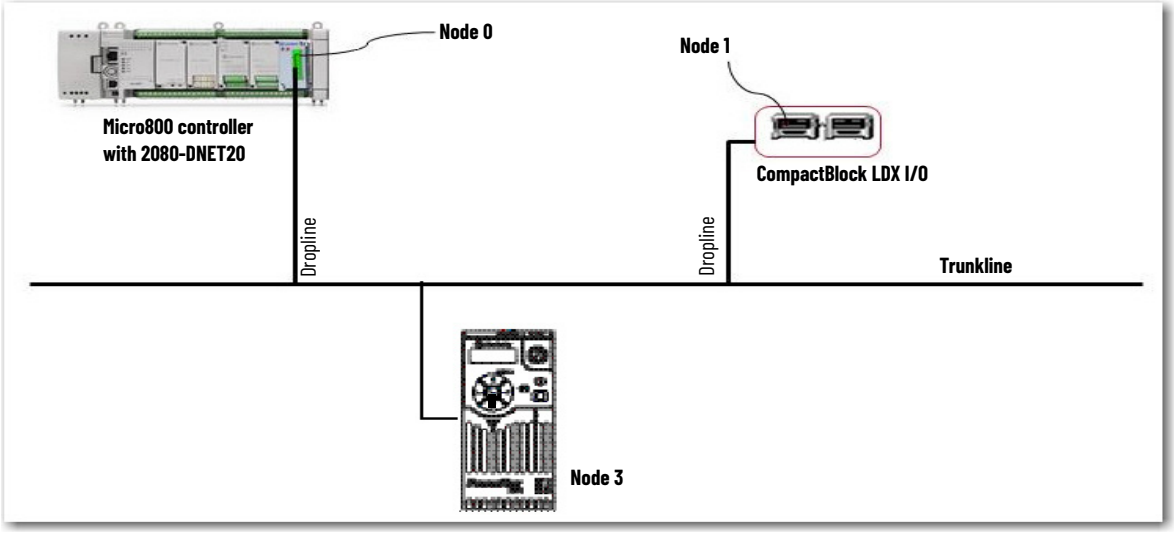
9. When the update is complete, the Update Status dialog appears. Check that the status is green and the update is complete. If the status is unsuccessful, power cycle the controller and restart the upgrade process.

10. After a successful firmware update, cycle power to the controller.

Quick Start Project for 2080-DNET20 Plug-in

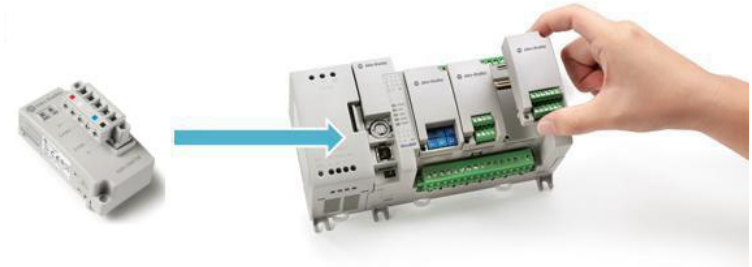
Using your 2080-DNET20 plug-in module and user-defined function blocks in Connected Components Workbench software, the following quick start project shows you how to turn on all outputs of a CompactBlock LDX I/O module connected in Node 3.

IMPORTANT	Quick Start Prerequisite
	For the following quick start projects, you must first download the DNET UDFBs and sample project from the Rockwell Automation Sample Code Library.

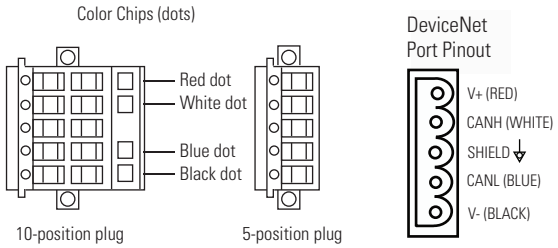


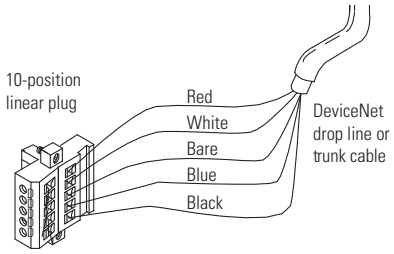
Setup and Wiring

1. Insert your 2080-DNET20 module into the designated plug-in slot in your Micro800 controller.

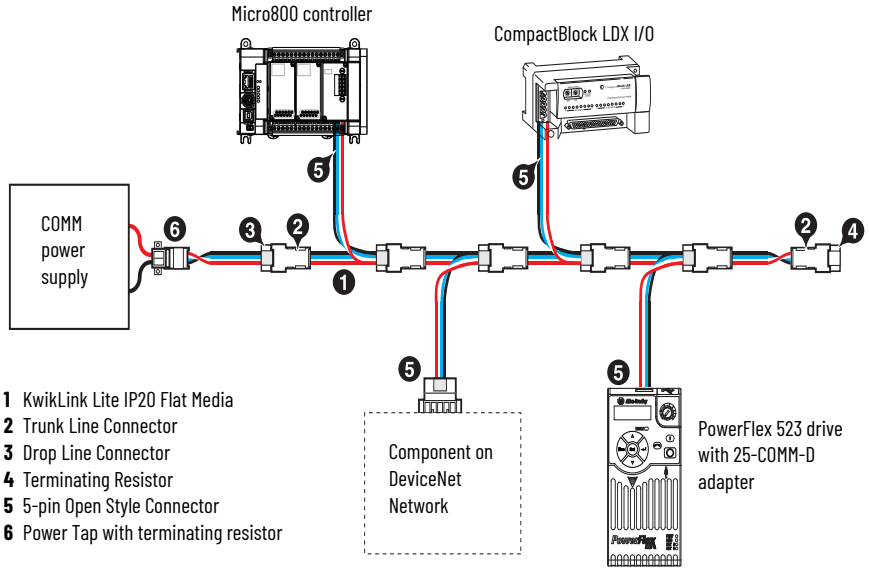


2. Next, wire your 2080-DNET20 plug-in as shown in the following diagram.





3. Configure the DeviceNet devices as shown.



4. Set the node address for the CompactBlock LDX I/O module through the node switches to node 1.
5. Set the node address for the PowerFlex drive through the 25-COMM-D adapter to node 2. Set the baud rate to autobaud.

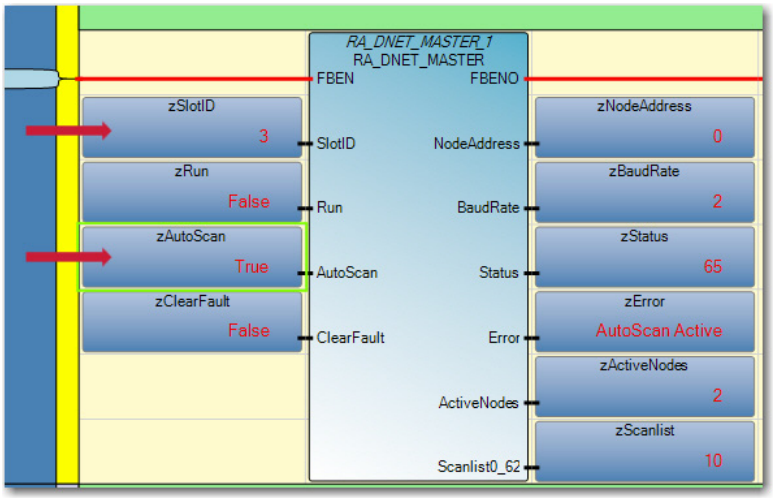
Configuration

WARNING: Do not change the configuration when the Controller is in RUN mode.

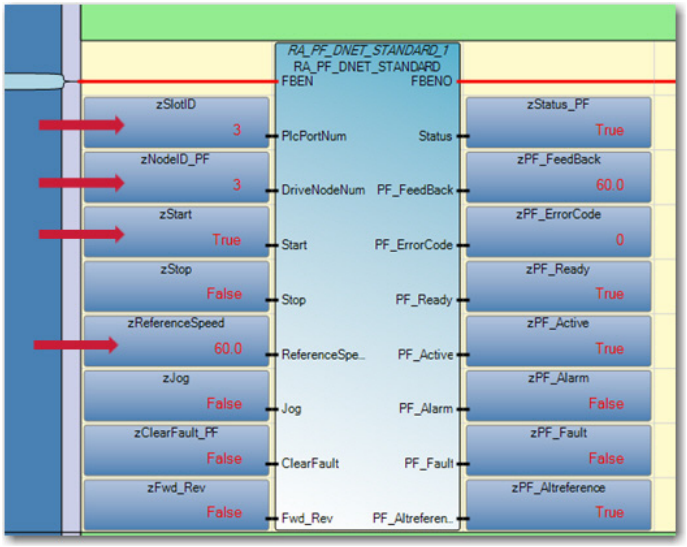
1. Launch Connected Components Workbench software and open the sample project that you have downloaded from the Sample Code Library.
2. Import the following DNET UDFBs into your project:
   - RA_DNET_MASTER
   - RA_DNET_LDX_DISCRETE
   - RA_PF_DNET_STANDARD

See User Defined Function Blocks on page 47 for information on input and output parameters for these UDFBs.

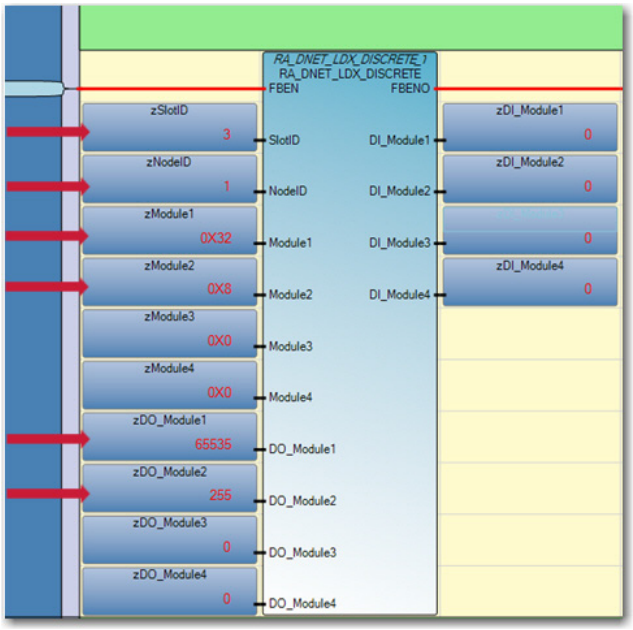
3. Configure the RA_DNET_MASTER UDFB as shown.



4. Configure the PowerFlex UDFB as shown (that is, set the node address, plug-in slot ID, and so on).



5. Configure the CompactBlock LDX I/O module UDFB as shown (that is, set the node address, plug-in slot ID, Module1...Module4).

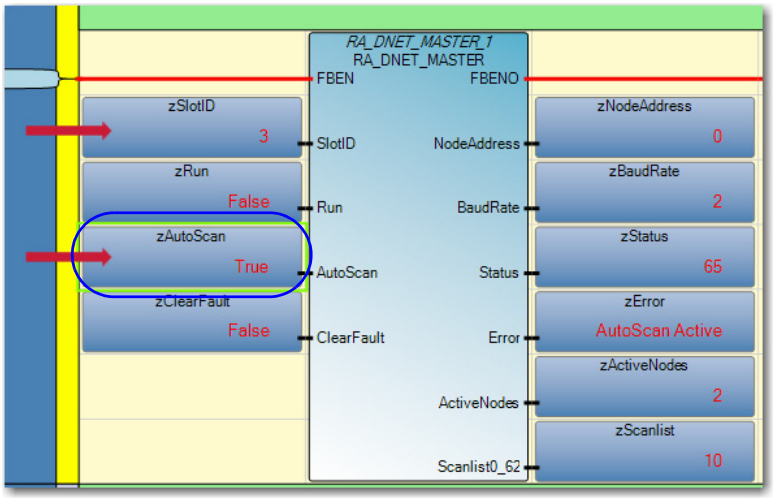


Build and Download

Build and download the project into the controller.

Execute Program

1. Set Micro800 controller to RUN mode.
2. Enable AutoScan in the DeviceNet Scanner UDFB.
   This scans all active nodes and populate the scan list.



3. Set the Scanner to RUN mode.

IMPORTANT	It is recommended that when Autoscan is running for the nodes in range, or for the connection to be established, the nodes should be idle without any pre-occupied connections requests.

Quick start Projects for 2080-MOT-HSC Plug-in

The following quick start projects will show you how to write and use three user-defined function blocks to configure and use your Micro800 High-speed Counter plug-in.

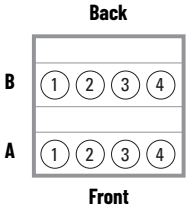
IMPORTANT **Quick Start Prerequisite**

For the following quick start projects, you must first download the HSC UDFBs and sample project from the Rockwell Automation Sample Code Library.

Setup and Wiring

1. Insert the high-speed counter plug-in module into the designated slot in your Micro800 controller.

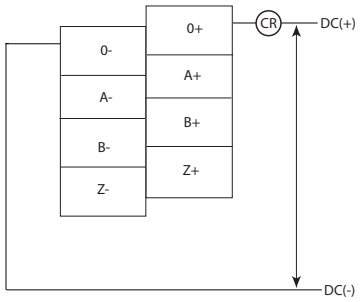
2. Wire your plug-in to your controller as shown in the following diagram.



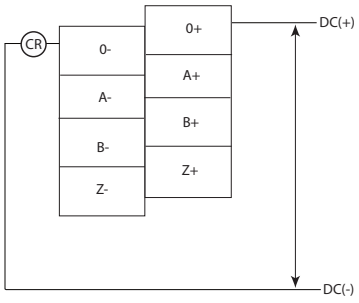
(View into terminal block)

Pin A1	0-	Pin B1	0+
Pin A2	A-	Pin B2	A+
Pin A3	B-	Pin B3	B+
Pin A4	Z-	Pin B4	Z+

Sinking Output Wiring

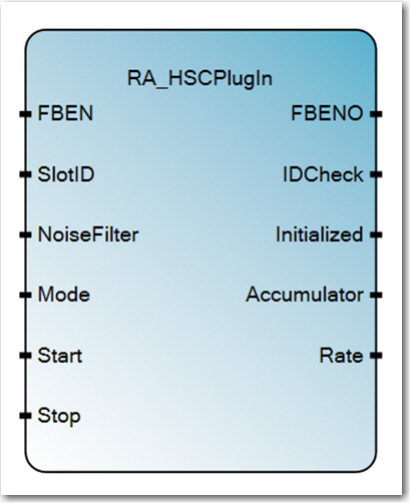


Sourcing Output Wiring



NOTE: Output functionality is not currently supported and is dependent on availability of UDFB support.

Configuration for UDFB 1: RA_HSCPlugIn



NOTE: The purpose of this UDFB is to get high-speed counter accumulator value and current pulse frequency.

1. Launch Connected Components Workbench software. Open the sample project that you have downloaded from the Sample Code Library.
2. Import the RA_HSCPlugIn UDFB that you have downloaded into this project. The sample project has the following input and output parameters.

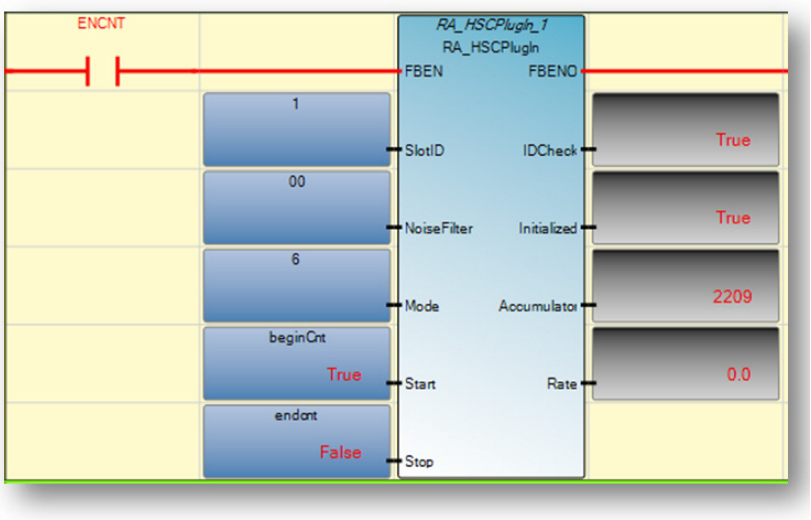
Table 45 - Input and Output Parameters

Parameter	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input
SlotID	INPUT	UINT	Plug-in slot number Slot ID = 1...5 (starting with the far left slot 1)
NoiseFilter	INPUT	USINT	00: No filter 01: 250 kHz 02: 200 kHz 03: 80 kHz 04: 40 kHz 05: 13.3 kHz 06: 10 kHz 07: 4 kHz 08: 2 kHz 09: 1 kHz 10: 500 Hz 11: 250 Hz 12: 125 Hz 13: 63.5 Hz 14: 31.25 Hz
HSCMode	INPUT	USINT	0, 2, 4, 6, 8, 10, 12
Start	INPUT	BOOL	Start counter
Stop	INPUT	BOOL	Stop the counter and clear MaxDPos and MaxDSpd value
FBENO	OUTPUT	BOOL	Function block enable output
IDCHeck	OUTPUT	BOOL	TRUE: HSC plug-in is at the selected slot FALSE: Wrong plug-in or no plug-in at the selected slot
Initialized	OUTPUT	BOOL	TRUE: HSC plug-in initialization finished and ready to execute FALSE: HSC plug-in initialization not yet finished
Accumulator	OUTPUT	LINT	Accumulator value
Rate	OUTPUT	Real	Current pulse rate. The rate calculation is based on how many pulses have been counted every 10 ms.

Build and Download

Build and download the program into the controller.

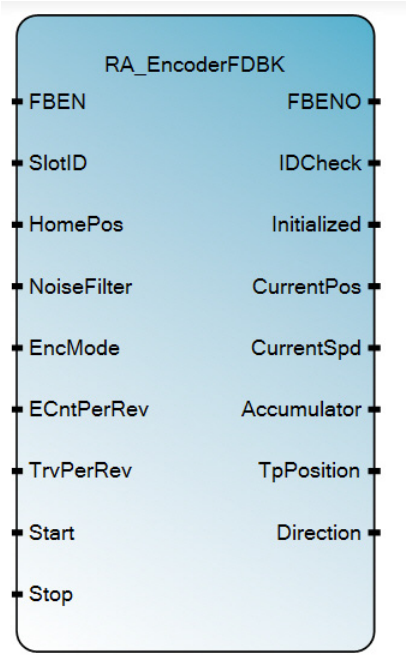
Execute the Function Block



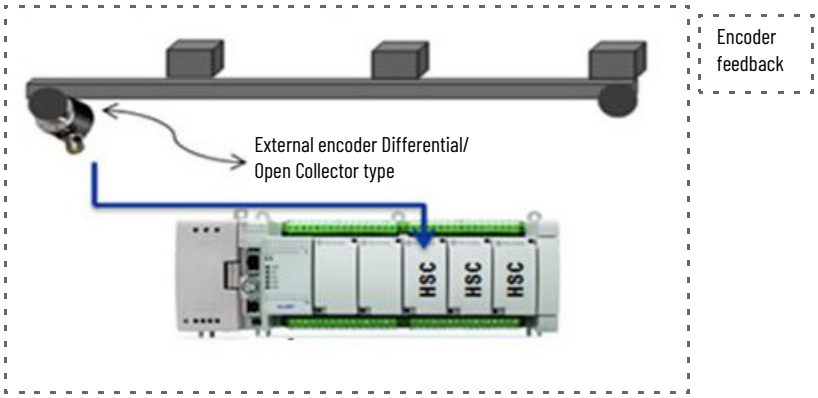
Operation Sequence for RA_HSCPlugIn

- A rising edge of FBEN causes the input Start to initialize steps. When initialization is done, Output Initialized changes to TRUE.
- You can start the feedback process (Start counting) after Initialized output becomes TRUE. A rising edge of Start triggers the feedback process. If you want to know the positioning information, then you must perform the following steps to make sure that position information is in sync:
  - Initialize the RA_HSCPlugIn function block
  - Start the counting process when necessary
- You can stop/pause the process by giving a rising edge of Stop. If you want to disable the function block, set Stop to true first.
- If FBEN is True and you starts the feedback process from previous Stop state, the function block will not be reinitialized. It will resume count from previous accumulator value.
- IF FBEN is False, then all outputs are cleared and accumulator is cleared. When FBEN goes True again, the HSC will be reinitialized.

Configuration for UDFB 2: RA_EncoderFDBK



NOTE: This UDFB gets the positioning information of a motion axis based on pulses that are counted from the encoder.



1. Launch Connected Components Workbench software and open the HSC project you have downloaded from the Sample Code Library.
2. Import the downloaded RA_EncoderFDBK into this project.
   Note that the UDFB comes with the following input and output variables.

Table 46 - Input and Output Parameters

Parameter	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input
SlotID	INPUT	UINT	Plug-in slot number Slot ID = 1...5 (starting with the far left slot 1)
HomePos	INPUT	REAL	Home position Same value indicated in MC_Home instruction
NoiseFilter	INPUT	USINT	00 - No filter 01 - 250 kHz 02 - 200 kHz 03 - 80 kHz 04 - 40 kHz 05 - 13.3 kHz 06 - 10 kHz 07 - 4 kHz 08 - 2 kHz 09 - 1 kHz 10 - 500 Hz 11 - 250 Hz 12 - 125 Hz 13 - 62.5 Hz 14 - 31.25 Hz
EncMode	INPUT	USINT	Encoder Mode, 1-X1, 2- X2, 4-X4
ECntPerRev	INPUT	REAL	User input to indicate how many X1 counts will be generated when the Encoder disk turns one revolution
TrvPerRev	INPUT	REAL	The actual distance traveled when the motor turns one revolution
Start	INPUT	BOOL	Start counter
Stop	INPUT	BOOL	Stop the counter and clear MaxDPos and MaxDSpd value
FBENO	OUTPUT	BOOL	Function block enable output
IDCheck	OUTPUT	BOOL	TRUE: HSC plug-in is at the selected slot FALSE: Wrong plug-in or no plug-in at the selected slot

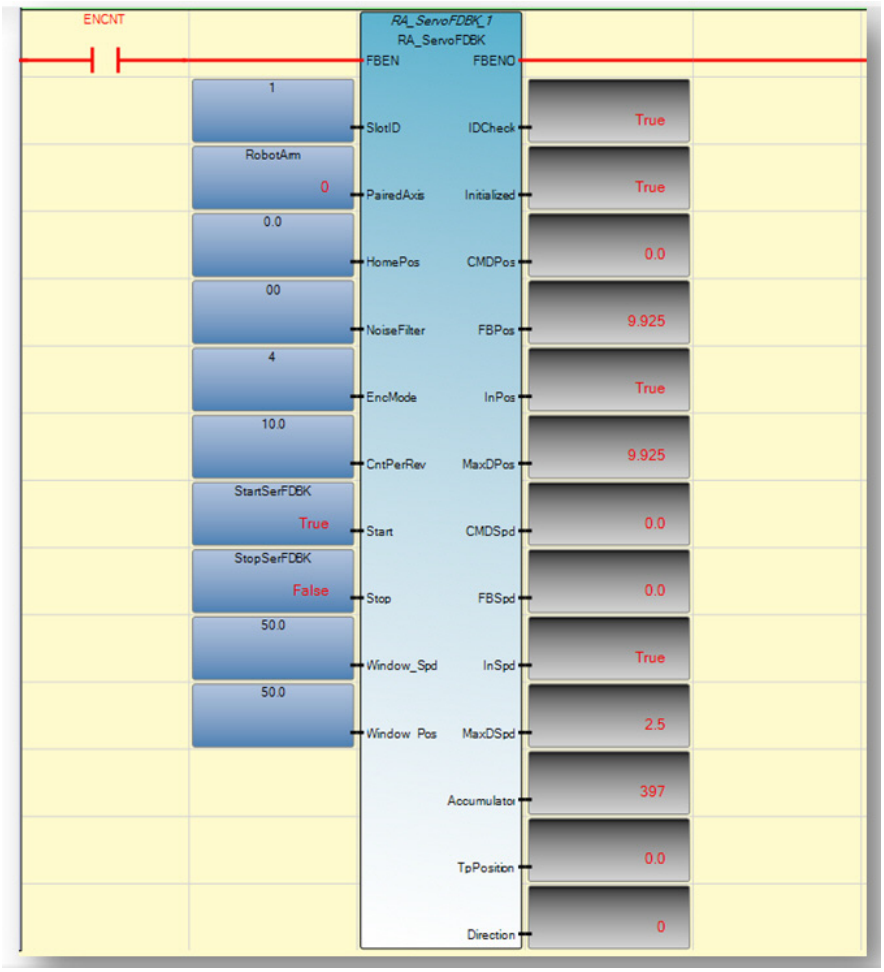
Table 46 - Input and Output Parameters (Continued)

Parameter	Type	Data Type	Description
Initialized	OUTPUT	BOOL	TRUE: Indicates that HSC initialization has finished FALSE: Indicates that HSC initialization has not finished
CurrentPos	OUTPUT	REAL	Current position
CurrentSpd	OUTPUT	REAL	Current speed (Unit = user distance per second)
Accumulator	OUTPUT	LINT	Accumulator value
TpPosition	OUTPUT	REAL	Position recorded when the latest touch probe is triggered
Direction	OUTPUT	SINT	1 = Forward -1 = Reverse 0 = Not moving

Build and Download

Build and download the program into the controller.

Execute the Function Block



Operation Sequence

- A rising edge of FBEN causes the function block Start initialize steps. When initialized done, Output Initialized changes to TRUE.
- You can start the feedback process (Start counting) after Initialized output becomes TRUE. A rising edge of Start triggers the feedback process. If you want to know the positioning information, you must do the following to make sure that position information is in sync:

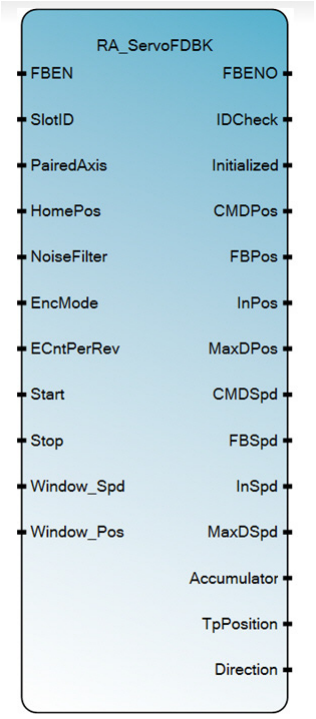
a. Initialize the RA_FDBKAxis function block.

b. Home the PTO axis to be monitored.

c. Once home is done, start the feedback process.

- You can stop/pause the process and clear the MaxDPos and MaxDSpd by given a rising edge of Stop. To disable the function block, you must first set Stop to false.
- If FBEN is True and you start the feedback process from previous Stop state, the function block will not be reinitialized. Count resumes from previous accumulator value.
- IF FBEN is False, then all outputs are cleared and the accumulator is cleared and when FBEN goes true again, the HSC is reinitialized.

Configuration for HSC UDFB 3: RA_ServoFDBK



NOTE: This UDFB gets positioning information of a PTO axis that is controlled by a Micro800 controller with HSC plug-in installed.

1. Launch Connected Components Workbench software and open the sample project that you have downloaded from the Sample Code Library.
2. Import the RA_ServoFDBK UDFB into this project.
   The UDFB comes with the following input and output variables.

Table 47 - Input and Output Parameters

Parameter	Type	Data Type	Description
FBEN	INPUT	BOOL	Function block enable input
SlotID	INPUT	UINT	Plug-in slot number Slot ID = 1...5 (starting with the far left slot 1)
HomePos	INPUT	REAL	Home position Same value indicated in MC_Home instruction

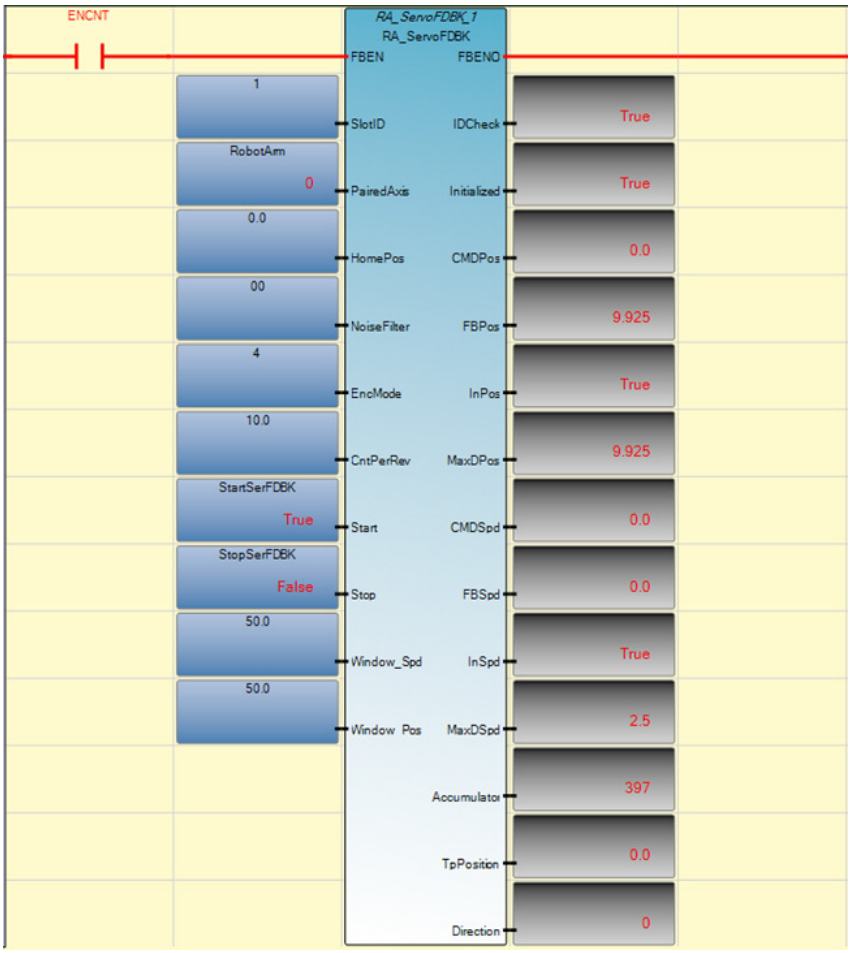
Table 47 - Input and Output Parameters (Continued)

Parameter	Type	Data Type	Description
NoiseFilter	INPUT	USINT	00 - No filter 01 - 250 kHz 02 - 200 kHz 03 - 80 kHz 04 - 40 kHz 05 - 13.3 kHz 06 - 10 kHz 07 - 4 kHz 08 - 2 kHz 09 - 1 kHz 10 - 500 Hz 11 - 250 Hz 12 - 125 Hz 13 - 62.5 Hz 14 - 31.25 Hz
EncMode	INPUT	USINT	Encoder Mode, 1-X1, 2- X2, 4-X4
ECntPerRev	INPUT	REAL	User input to indicate how many X1 counts will be generated when Encoder disk turns one revolution
TrvPerRev	INPUT	REAL	The actual distance traveled when the motor turns one revolution
Start	INPUT	BOOL	Start counter
Stop	INPUT	BOOL	Stop the counter and clear MaxDPos and MaxDSpd value
FBENO	OUTPUT	BOOL	Function block enable output
IDCheck	OUTPUT	BOOL	TRUE: HSC plug-in is at the selected slot FALSE: Wrong plug-in or no plug-in at the selected slot
Initialized	OUTPUT	BOOL	TRUE: Indicates that HSC initialization has finished FALSE: Indicates that HSC initialization has not finished
CurrentPos	OUTPUT	REAL	Current position
CurrentSpd	OUTPUT	REAL	Current speed (Unit = user distance per second)
Accumulator	OUTPUT	LINT	Accumulator value
TpPosition	OUTPUT	REAL	Position recorded when the latest touch probe is triggered
Direction	OUTPUT	SINT	1 = Forward -1 = Reverse 0 = Not moving

Build and Download

Build and download the program into the controller.

Execute the Function Block



Operation Sequence

- A rising edge of FBEN causes the function block to start initializing steps. When Initialized is done, Output Initialized changes to TRUE.
- You can start the feedback process (Start counting) after Initialized is TRUE. A rising edge of Start triggers the feedback process.
  If you want to get positioning information, you must perform the following steps to verify that position information is in sync:
  a. Initialize the RA_FDBKAxis function block.
  b. Home the PTO axis to be monitored.
  c. Once home is done, start the feedback process.
- You can stop/pause the process and clear the MaxDPos and MaxDSpd by giving a rising edge of Stop. If you want to disable the function block, you need set it to Stop state first.
- If FBEN is True and you start the feedback process from the previous Stop state, the function block is not reinitialized. It resumes the count from the previous accumulator value.
- IF FBEN is false, then all outputs are cleared and the accumulator is cleared. When FBEN goes true again, the HSC is reinitialized.

Error Codes

Troubleshooting

For troubleshooting your Micro800 controller system, see the User Manual for your controller:

- Micro830, Micro850, and Micro870 Programmable Controllers User Manual, publication 2080-UM002
- Micro820 Programmable Controllers User Manual, publication 2080-UM005

Error Codes for Micro800 Plug-ins

This section lists possible error codes for your plug-in modules and 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 rok.auto/support.

Table 48 - List of Error Codes for Micro800 Plug-ins

Error Code	Description	Recommended Action
For the following four error codes, z refers to 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 has encountered 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.
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.
0xF0Dz	A hardware error has occurred while power was being applied to the plug-in module or the plug-in module was removed.	Perform the following: • Correct the plug-in I/O module configuration in the user program. • Build and download the program using Connected Components Workbench software. • 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 software. • Put the Micro800 controller into Run mode.

Calling Rockwell Automation for Assistance

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

- Controller type, series letter, revision letter, and firmware revision of the controller
- Controller indicator status

Notes:

Numerics

2080-DNET20 12
wiring 17
2080-IF2 10
wiring 14
2080-IF4 10
wiring 14
2080-IQ4 10
wiring 14
2080-IQ4OB4 10
wiring 14
2080-IQ4OV4 10
wiring 14
2080-MEMBAK-RTC 10, 13
2080-MEMBAK-RTC2 10, 13
2080-MOT-HSC 11
wiring 15
2080-OB4 10
wiring 14
2080-OF2 10
wiring 15
2080-OV4 10
wiring 14
2080-OW4I 10
wiring 14
2080-RTD2 10, 23, 24
data maps 25
wiring 14
2080-SERIALISOL 11
wiring 15
2080-TC2 10, 18, 23
data maps 25
features 23
thermocouple sensor types and ranges 23
wiring 14
2080-TRIMPOT6 11

A

About Plug-In Modules and Accessories 23

C

cable pinout
controller to modem cable 16
calibration error bit (CE) 25
calling for assistance 75
channel temperature data 25
CJC open-circuit bit (COC) 25
CJC Sensor 18
CJC thermistor 18
code calibrated bit 25
Connected Components Workbench 23, 24, 25, 26

D

data illegal (DI) 25

E

error codes 75

H

hardware features 13

I

installation 13

M

Micro800 plug-ins 9
6-channel trimpot analog input plug-in module 11
DeviceNet scanner plug-in module 12
digital I/O plug-in module 10
high-speed counter plug-in module 11
memory backup and high accuracy RTC plug-in module 10
non-isolated RTD plug-in module 10
non-isolated unipolar analog I/O plug-in module 10
relay output plug-in module 10
RS232/RS485 isolated serial port plug-in module 11

O

open circuit 25
open-circuit 23
overrange 23, 24, 25

P

plug-in modules 13

R

Removal and Insertion Under Power 23
RS-232 cable 16
RTD 23, 26
sensor types 24

S

system overrange bit (SOR) 25
system underrange bit (SUR) 25

T

temperature conversion 26
thermistor 23
thermocouple 23, 26
thermocouple sensor types 23

U

underrange 23, 24, 25

W

wiring 13

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, and product notification updates.	rok.auto/support
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Technical Documentation Center	Quickly access and download technical specifications, installation instructions, and user manuals.	rok.auto/techdocs
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental compliance information on its website at rok.auto/pec.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.

rockwellautomation.com — expanding **human possibility**®

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2663 0600

ASIA PACIFIC: Rockwell Automation SEA Pte Ltd, 2 Corporation Road, #04-05, Main Lobby, Corporation Place, Singapore 618494, Tel: (65) 6510 6608

UNITED KINGDOM: Rockwell Automation Ltd., Pitfield, Kiln Farm, Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908) 838-800

Publication 2080-UM004E-EN-E - March 2024

Supersedes Publication 2080-UM004D-EN-E - March 2018