

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

Fieldbus

MC68331-Based Fieldbus Round Card User Manual

January 1998 Edition
Part Number 321866A-01

© Copyright 1998 National Instruments Corporation. All rights reserved.



Internet Support

support@natinst.com
E-mail: info@natinst.com
FTP Site: ftp.natinst.com
Web Address: http://www.natinst.com



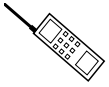
Bulletin Board Support

BBS United States: (512) 794-5422
BBS United Kingdom: 01635 551422
BBS France: 01 48 65 15 59



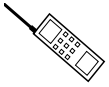
Fax-on-Demand Support

(512) 418-1111



Telephone Support (U.S.)

Tel: (512) 795-8248
Fax: (512) 794-5678



International Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 288 3336,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00,
Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30,
Hong Kong 2645 3186, Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970,
Korea 02 596 7456, Mexico 5 520 2635, Netherlands 0348 433466, Norway 32 84 84 00,
Singapore 2265886, Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51,
Taiwan 02 377 1200, United Kingdom 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway Austin, TX 78730-5039 Tel: (512) 794-0100

Important Information

Warranty

The Fieldbus Round Card is warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

NI-FBUS™ is a trademark of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

About This Manual

How to Use the Manual Set ix
Organization of This Manual x
Conventions Used in This Manual x
Related Documentation xi
Customer Communication xii

Chapter 1
Introduction

What You Need to Get Started 1-1
Hardware Overview 1-1
Software Overview 1-2

Chapter 2
Hardware Installation

Install the Hardware 2-1

Chapter 3
Software Installation

Fieldbus Round Card Interface Kit Components 3-1
Install the Software 3-1

Chapter 4
Interfacing to Your Round Card

Interfacing to the Hardware 4-1
- Hardware Description 4-1
- Daughter Card Interface Signal Connections 4-2

Using the Software 4-6
- Overview 4-6
- Developing Your Round Card Application 4-6
  - Writing Device Templates 4-7
  - Converting a Device Template to C Code 4-7

Writing Function Block Callbacks 4-8
Writing userStart and Registering Callbacks..... 4-8
Specifying the System Clock Speed and RAM Size 4-9
Defining and Installing Interrupt Handlers..... 4-9
Generating Your Device Configuration..... 4-10
Converting Your Device Configuration to C Code 4-10
Compiling, Linking, and Downloading Your Program 4-10
Burn Your Flash 4-12

Chapter 5
Migrating from a Development Round Card to an OEM Round Card

Differences Between the Development and OEM Round Cards 5-1
System Clock Speed Selection 5-2

Appendix A
Data Link Configuration Section Format

Appendix B
System Management Configuration Section Format

Appendix C
Specifications

Appendix D
Customer Communication

Glossary

Figures

Figure 2-1. Top Side of Fieldbus Round Card 2-2
Figure 2-2. Top Side of Serial Programming Daughter Card 2-3

Figure 4-1. Pin Assignments for the Daughter Card Interface Header, J4 4-3
Figure 4-2. Pin Assignments for the Daughter Card BDM Connector, W3 4-5

Tables

Table 4-1. Signal Descriptions of Daughter Card Connector, J4 4-3
Table 4-2. Pinout of Daughter Card BDM Header, J2 4-5
Table 4-3. Memory Map of Fieldbus Round Card 4-12

Table 5-1. Features of the Development Round Card and OEM Round Card 5-1

Table A-1. Valid Variable Names and Values for the Data Link Configuration A-1

Table B-1. Valid Variable Names and Values for the MIB Sections B-2
Table B-2. Valid Variable Names and Values for the FB Schedule Sections B-2

Table C-1. Fieldbus Round Card Specifications C-1
Table C-2. Fieldbus Round Card Components C-2
Table C-3. Serial Programming Daughter Card Specifications and Components ... C-2

This manual contains instructions for installing, interfacing to, and programming the National Instruments Fieldbus Round Card. The Round Card software is intended for use with Windows 3.*x*, Windows 95, or Windows NT.

This manual assumes that you are already familiar with the Windows operating system you are using.

How to Use the Manual Set

Use the *Getting Started with Fieldbus* manual to install and configure your Fieldbus hardware, the Fieldbus Stack Interface Library, and the NI-FBUS Function Block Shell software.

Use this *MC68331-Based Fieldbus Round Card User Manual* manual to install the Motorola MC68331-based Fieldbus Round Card.

Use the *Intel 80188EB-Based Fieldbus Round Card User Manual* manual to install the Intel 80188EB-based Fieldbus Round Card.

Use the *NI-FBUS Function Block Shell Reference Manual* to learn about writing Function Block server applications that are embedded in the Fieldbus Round card.

Use the *NI-FBUS Monitor User Manual* to learn to use the interactive NI-FBUS Monitor utility with your Fieldbus Round Card.

Use the *NI-FBUS Communications Manager User Manual* to learn to use the interactive Fieldbus dialog system with your Fieldbus Round Card.

Use the *NI-FBUS Configurator User Manual* to learn to use the NI-FBUS Configurator to configure your Fieldbus network.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, lists what you need to get started and includes a brief description of the Fieldbus Round Card hardware and supplied software.
- Chapter 2, *Hardware Installation*, contains instructions to help you install your Fieldbus Round Card.
- Chapter 3, *Software Installation*, contains instructions for installing the software that came with your Fieldbus Round Card.
- Chapter 4, *Interfacing to Your Round Card*, describes how to connect the Fieldbus Round Card to any external electronics, and how to develop your Field Device application to interface to the NI-FBUS Function Block Shell.
- Chapter 5, *Migrating from a Development Round Card to an OEM Round Card*, explains how to migrate from the development version of the Fieldbus Round Card to the OEM version of the Fieldbus Round Card.
- Appendix A, *Data Link Configuration Section Format*, explains how to structure the Data Link Configuration section of your Device Configuration `.ini` file.
- Appendix B, *System Management Configuration Section Format*, explains how to structure the System Management Configuration section of your Device Configuration `.ini` file.
- Appendix C, *Specifications*, describes the characteristics of the Fieldbus Round Card and the Serial Programming Daughter Card.
- Appendix D, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.

Conventions Used in This Manual

The following conventions are used in this manual:

<>

Angle brackets enclose the name of a key on the keyboard—for example, <shift>. Angle brackets containing numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DBIO<3..0>.

	This icon to the left of bold italicized text denotes a note, which alerts you to important information.
	This icon to the left of bold italicized text denotes a warning, which advises you of precautions to take to avoid being electrically shocked.
bold	Bold text denotes the names of menus, menu items, parameters, dialog box, dialog box buttons or options, icons, windows, Windows 95 tabs, or LEDs.
bold italic	Bold italic text denotes a note, caution, or warning.
italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept. This font also denotes text from which you supply the appropriate word or value, as in Windows 3.*x*.
`italic monospace`	Italic text in this font denotes that you must supply the appropriate words or values in the place of these items.
`monospace`	Text in this font denotes text or characters that you should literally enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and for statements and comments taken from programs.

Related Documentation

The following document contains information that you may find helpful as you read this manual:

- *Fieldbus Foundation Specification*, which includes the following items:
  - *Fieldbus Foundation System Management Services*
  - *Function Block Application Process, Part 1*
  - *Function Block Application Process, Part 2*

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix D, *Customer Communication*, at the end of this manual.

Chapter 1

Introduction

This chapter lists what you need to get started and includes a brief description of the Fieldbus Round Card hardware and supplied software.

What You Need to Get Started

To install your Fieldbus Round Card Interface Kit, you need the following items:

- Fieldbus Round Card
- Fieldbus Round Card Interface Kit distribution disk
- Windows 3.*x*, Windows 95, or Windows NT installed on your computer
- Serial Programming Daughter Card
- Standard serial cable

Hardware Overview

The Fieldbus Round Card is a stand-alone card that allows you to interface to a network that complies with the Fieldbus Foundation H1 specification. The Fieldbus Round Card is powered from the Fieldbus. The Fieldbus Round Card uses the Motorola MC68331 embedded processor and a programmable 128 KB X 16 Flash to run the Stack Interface Library, Function Block Shell, and user applications. A 128 KB X 16 SRAM device on the card provides volatile memory.

The Fieldbus Round Card also has a 4 KB X 8 Serial ROM (SROM) that serves as non-volatile memory. This non-volatile memory is used to store the Fieldbus MIB parameters and block parameters.

The Fieldbus Round Card provides an interface to the Fieldbus. To run real-world applications, the Round Card can be connected to a Daughter Card. On the Daughter Card, you can implement DIO, A/D, D/A functionality. The Fieldbus Round Card provides the necessary interface signals via a 2 X 21 position header (W1) that can be

connected to the Daughter Card. Refer to Figure 4-1, *Pin Assignments for the Daughter Card Interface Header, J4,*in Chapter 4, *Interfacing to Your Round Card,* for the pin assignments of the W1 header.

When developing your application, you will use the development version of the Fieldbus Round Card. When you have finished developing and debugging your application, you need to migrate to the OEM version of the Round Card. Chapter 5, *Migrating from a Development Round Card to an OEM Round Card,* explains how to migrate from the development version to the OEM version of the Round Card.

Software Overview

The software supplied with the Fieldbus Round Card Interface Kit includes the NI-FBUS Function Block Shell, which is an Application Programmer's Interface (API) designed to simplify Fieldbus device development by providing a high-level interface to the Fieldbus communications stack. In addition, a serial driver API is provided to allow you to make use of the Round Card's interrupt-driven serial port. The serial driver supports standard HART commands, as well as generic access to the serial port to allow any other serial protocols. A linkable library version of the Fieldbus protocol stack is also supplied. Link your Function Block application with the Function Block Shell and the protocol stack before downloading it to your Fieldbus Round Card.

Chapter 2

Hardware Installation

This chapter contains instructions to help you install your Fieldbus Round Card.

Warning: ***Electrostatic discharge can damage several components on your Fieldbus Round Card. To avoid such damage in handling the board, touch the antistatic plastic package to a metal part of your computer chassis before removing the board from the package.***

Install the Hardware

Perform the following steps to install the Fieldbus Round Card:

1. Connect the Fieldbus cable to terminal J1 on the Round Card. The positive (+) end of the cable should be connected to terminal F+ and the negative (–) end of the cable should be connected to terminal F– (see Figure 2-1). Ensure that the Fieldbus cable is properly terminated.
2. Connect the Serial Programming Daughter Card to the Fieldbus Round Card by mating the W1 and W2 connectors on the Round Card to the J2 and J4 connectors on the Daughter Card. (See Figure 2-2 for the locations of the connectors on the Daughter Card.)
3. Connect a standard serial cable from the serial port of the host PC to the J1 connector on the Daughter Card. This link can be used to download the application to the Flash on the Fieldbus Round Card.
4. You can debug the application in RAM by using the BDM port of the processor. Refer to Chapter 4, *Interfacing to Your Round Card*, for a description of the BDM mode. Connect the 3 V ICD cable that came with the software debugger from the parallel port of the host PC to connector W3 on the Daughter Card. Confirm that pin 1 of the 3 V ICD cable matches pin 1 on the W3 connector.

☞ **Note:** ***Remember to order a 3 V ICD cable when you buy the software debugger. One suggested vendor for the ICD cable is: P&E Microcomputer Systems, Woburn, MA Web site: http://www.pemicro.com Low Voltage In-Circuit Debugger Cable (CPU16/CPU32)***

5. Press the reset button on the Round Card, S1, to reset the processor.

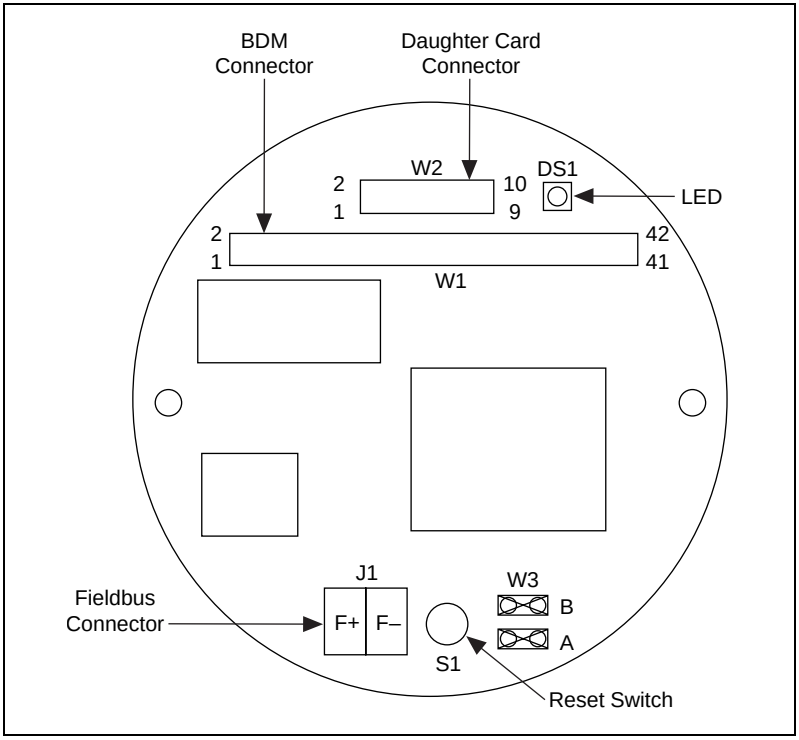


Figure 2-1. Top Side of Fieldbus Round Card

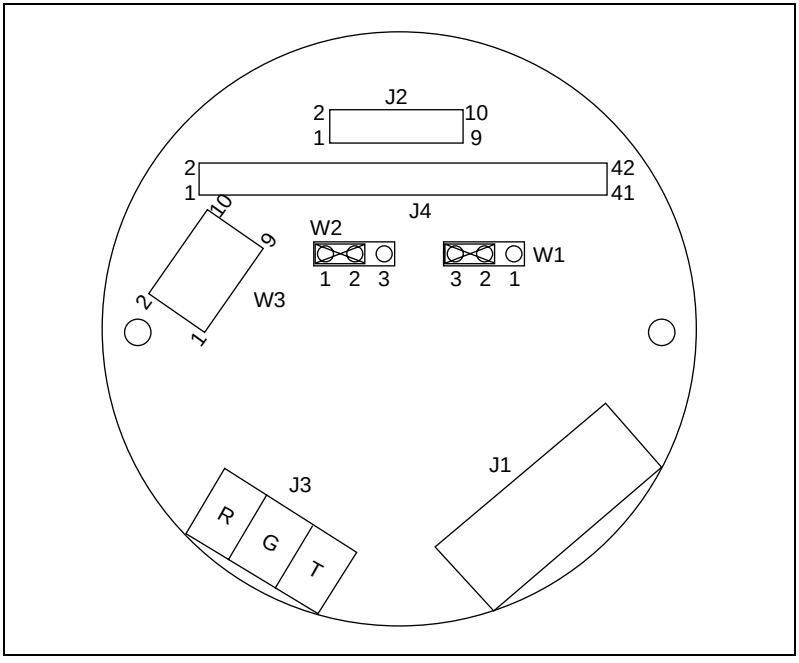


Figure 2-2. Top Side of Serial Programming Daughter Card

Your hardware is now installed. Proceed to Chapter 3, *Software Installation*, for instructions on installing the software.

Chapter 3

Software Installation

This chapter contains instructions for installing the software that came with your Fieldbus Round Card.

Fieldbus Round Card Interface Kit Components

The MC68331-Based Fieldbus Round Card Interface Kit contains the following software components:

- Function Block Shell Code Generation and Configuration Generation utilities for Windows
- Combined protocol stack, NI-FBUS Function Block Shell, and Serial Driver linkable library for SDS CrossCode Compiler (version 7.11)
- Sample function block templates for all standard Foundation Fieldbus function blocks
- Sample function block configuration files
- C language header files for interfacing to the NI-FBUS Function Block Shell

Install the Software

Complete the following steps to run the software installation program:

1. Insert the Fieldbus Round Card Interface Kit distribution disk into an unused drive.
2. Windows 95 or Windows NT 4.*x*: Choose **Run...** from the **Start** menu.

   Windows 3.*x* or Windows NT 3.5 or earlier: Choose **Run...** from the **File** menu in the **Program Manager** window.
3. Type the following command into the dialog box that appears:

   x:\setup

   where *x* is the letter of the drive containing the distribution disk (usually a or b).

4. The interactive installation program installs the software into the directory of your choice.

Your software installation is now complete. Proceed to Chapter 4, *Interfacing to Your Round Card*.

Interfacing to Your Round Card

Chapter 4

This chapter describes how to connect the Fieldbus Round Card to any external electronics, and how to develop your Field Device application to interface to the NI-FBUS Function Block Shell.

Interfacing to the Hardware

Hardware Description

The Fieldbus Round Card provides an interface to the Fieldbus. To run real-world applications, the Round Card can be connected to a Daughter Card. On the Daughter Card you can implement DIO, A/D, D/A functionality. The Fieldbus Round Card provides the necessary interface signals via a 2 X 21 position header (W1) that can be connected to the Daughter Card. Refer to Figure 4-1 for the pin assignments of the W1 header.

The Fieldbus Round Card is also capable of providing a 5 V at 4 mA power supply to power your electronics.

The Fieldbus Round Card can be placed in Simulate mode by removing jumper A of W3. In Simulate mode you can run a software simulation of your function blocks without doing any actual input/output. Under normal operating conditions, place jumper A of the W3 header on the Fieldbus Round Card.

The MC68331 processor has a Background Debug Mode (BDM) port, as shown in Figure 2-1, *Top Side of Fieldbus Round Card*, in Chapter 2, *Hardware Installation*. The BDM port can be used for debugging your application in RAM. The BDM port has an SPI interface that can be connected to the parallel port of the host computer running the software debugger. Over the BDM port you can monitor registers, read or write to memory, and single step through your code. The stack library has been developed and debugged with SDS SingleStep software, version 7.11.

To use your software debugger to debug your application, connect the Daughter Card to the Fieldbus Round Card. Connect the 3 V ICD cable that came with the software debugger from the parallel port of the host PC to the W3 connector on the Daughter Card. Connector W3 on the Daughter Card is connected to the BDM port on the processor. Refer to Figure 4-2 for the pin assignments of header W3.

The development version of the Fieldbus Round Card operates under one of two operating modes: PROG mode and RUN mode. When power is applied to the Fieldbus Round Card, the processor reads the position of jumper B of W3 to check the mode (see Figure 2-1, *Top Side of Fieldbus Round Card*, in Chapter 2, *Hardware Installation*, for the location of W3).

To select PROG mode, place jumper B of W3. Connect the Serial Programming Daughter Card to the Fieldbus Round Card. Attach a standard serial port cable from the COM port of the host computer to connector J1 on the Daughter Card. When the card is powered up in PROG mode, the processor downloads the user application into the Flash, from the COM port of the host computer to the serial port of the processor.

To select RUN mode, remove jumper B of W3 from the Round Card. When the card is powered up in RUN mode, the processor begins executing the application that is loaded in the Flash.

The serial port of the processor can alternatively used to run the serial HART protocol or any other proprietary protocol. Refer to the pin assignments of the header W1 in Figure 4-1 for the serial port signals.

Daughter Card Interface Signal Connections

You can use the hardware on the Daughter Card to implement DIO, A/D, D/A functionality. The Fieldbus Round Card to Daughter Card interface is over a 2 X 21-position header. Figure 4-1 shows the pin assignments for the J4 Daughter Card connector.

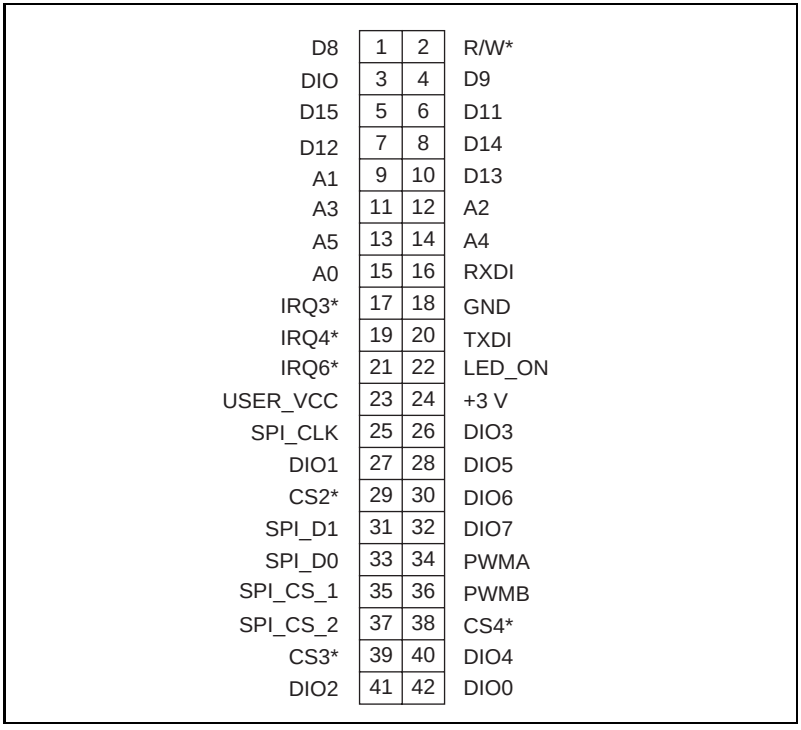


Figure 4-1. Pin Assignments for the Daughter Card Interface Header, J4

Table 4-1 lists the signal descriptions of the Daughter Card interface connector, J4.

Table 4-1. Signal Descriptions of Daughter Card Connector, J4

Signal Name	Description
3 V	3 V power supply output
A<5..0>	Address lines driven by the MC68331 processor
CS<4..2>*	Programmable chip-select lines driven by the processor
D<15..8>	Upper data bus lines of the processor
DIO<7..0>	General-purpose digital I/O lines connected to the PORT GP of the processor

Table 4-1. Signal Descriptions of Daughter Card Connector, J4 (Continued)

Signal Name	Description
GND	Ground
IRQ<6,4,3>*	IRQ lines that provide IRQ signals to the processor
LED_ON	Output to d rive an LED for monitoring status of the Fieldbus Round Card
PWMA	Output from the pulse width modulator on the processor
PWMB	Output from the pulse width modulator on the processor
R/W*	R/W* line driven by the processor
RXD1	3 V-compatible serial port line
SPI_CLK	Clock output from the SPI port of the processor
SPI_D0	Serial output from the SPI slave device
SPI_D1	Serial input to the SPI slave device
SPI_CS_1	SPI chip select driven by the processor to select the slave SPI devices
SPI_CS_2	SPI chip select driven by the processor to select the slave SPI devices
TXD1	3 V-compatible serial port line
USER_VCC	5 V at 4 mA power supply output to power user electronics

Figure 4-2 shows the pin assignments for the Serial Programming Daughter Card BDM connector, J2.

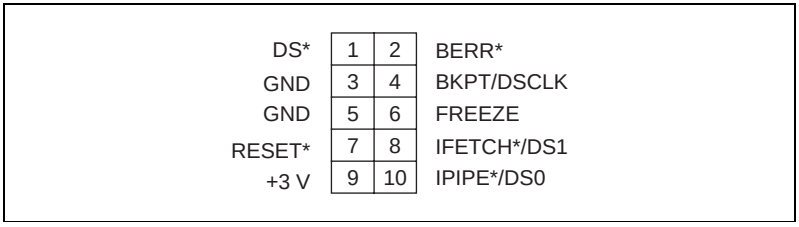


Figure 4-2. Pin Assignments for the Daughter Card BDM Connector, W3

Table 4-2 lists the signal descriptions of the BDM connector, J2, on the Serial Programming Daughter Card.

Table 4-2. Pinout of Daughter Card BDM Header, J2

Signal Name	Description
+ 3 V	3 V power supply output
BERR*	Bus error
BKPT/DSCLK	Hardware breakpoint signal or BDM clock source
DS*	Data strobe signal from processor
GND	Ground
FREEZE	Indicates processor has entered background mode
IFETCH*/DS1	Indicates instruction pipeline activity or BDM data input signal
IPIPE*/DS0	Indicates instruction pipeline activity or BDM data output signal
RESET*	Processor reset

Using the Software

Overview

Most of the code that will be running on your Fieldbus Round Card has already been written for you. It includes the Fieldbus protocol stack, Function Block Shell, and Serial Driver, which are provided in the form of a linkable library with your Fieldbus Round Card Interface Kit. This library provides Fieldbus communications and an API (the Function Block Shell) designed to isolate your application as much as possible from the specifics of the Fieldbus. For more specific information about the Function Block Shell API, consult the *NI-FBUS Function Block Shell Reference Manual*.

In addition, the library contains an API to allow HART or direct serial access to the Round Card's serial port. This API facilitates communications with a HART or serial transducer external to the Round Card. This Serial Driver API is also described in Chapter 8, *Serial Functions*, of the *NI-FBUS Function Block Shell Reference Manual*.

Developing Your Round Card Application

Complete the following steps to develop your application, after you have installed the hardware and software:

1. Write a device template for your device.
2. Convert the device template to C code using the Device Code Generator.
3. Write your Function Block Callbacks, algorithms, and device interface code.
4. Write your `userStart` function to register your callbacks.
5. Specify the system clock speed and RAM size.
6. Define and install any interrupt handlers you may need.
7. Write your Device Configuration.
8. Convert your Device Configuration to C code using the Configuration Code Generator.
9. Compile, link, and download your program on Flash for installation on the Round Card.
10. Burn your Flash on the Round Card.

These steps are documented in more detail in the following sections.

Writing Device Templates

You must create a device template to describe the network-visible structure of your device and the parameters of your function blocks to the Function Block Shell. The device template is an ASCII file that is divided into various sections containing numerical and string parameters. Sample device templates are in the `\samples` subdirectory of your installation directory for devices containing function blocks of the standard types.

The simplest way to create your device template is to modify a copy of one of the sample device templates using a text editor such as MS-DOS Edit or Windows Notepad. Choose the sample device template that most closely matches your device. For example, if the main function of your device is analog input, start with the AI Device Template. If you want your device to contain multiple function blocks, you need to paste several `BLOCK` sections from the sample files into your device template file.

The templates contain information about the device identification, the physical and function blocks in the device, and the device parameters. The device template syntax is described in Chapter 3, *Registration Functions*, of the *NI-FBUS Function Block Shell Reference Manual.*

Your final device templates are converted to C code using the Device Code Generator, described in the next section.

Converting a Device Template to C Code

Before you compile your Function Block device, you must convert the device template to C code using the Device Code Generator. The Device Code Generator resides in the `\utils` subdirectory of your installation directory. The Device Code Generator takes the following command line arguments:

`codegen` *`deviceTemplate outputFile symbolFile`*

where *`deviceTemplate`* is the name of your device template file, and *`outputFile`* is the name that you want to call the output file. Make sure *`outputFile`* ends in `.c`.

If there are syntax errors in your device template, `codegen` tells you where they are. When you have corrected all syntax errors, *`outputFile`* is created. *`outputFile`* contains code representing the structure of your device. `outputFile` is used again when you compile and link your device application.

`symbolFile` contains a reference to DD information for the parameters defined in your DD and template file. If you are using only the standard function blocks and their parameters, *`symbolFile`* is the standard symbols file, `nifb.sym`, located in the `\samples` subdirectory. If you have defined your own blocks or parameters, *`symbolFile`* must be set to the output of the DD tokenizer.

Writing Function Block Callbacks

The callback functions that you must develop are responsible for the following main functions:

- Executing your function block algorithm
- Handling alarm confirmations and acknowledgments (if you are using alarms)

The Function Block Shell calls your execution callback whenever your Block is scheduled to execute. This callback performs whatever algorithm you want your function block to perform. The other callbacks, which are called after your device sends an alarm, allow you to perform device-specific processing upon alarm confirmations (notifications that the alarm was received) and alarm acknowledgments (notifications that a user has seen the alarm).

Other optional callbacks are provided for other purposes. See Chapter 4, *Callback Functions*, of the *NI-FBUS Function Block Shell Reference Manual* for more details.

Writing userStart and Registering Callbacks

Your `userStart` function is called by the stack during startup. It is your chance to perform your own initialization tasks. `userStart` also initializes the Function Block Shell and registers your callbacks with the Function Block Shell. See Chapter 7, *Miscellaneous Functions*, of the *NI-FBUS Function Block Shell Reference Manual* for more details.

Specifying the System Clock Speed and RAM Size

You must specify the system clock speed (4 MHz or 8 MHz) and RAM size (128 KB or 256 KB) you are using so that the protocol stack is set up correctly. To do so, you must declare and define two global variables called `SystemClockSpeed` and `RamSize`, both `unsigned char`, in your source code. Failing to declare these global variables may result in link time errors.

You must initialize these variables when you declare them. The valid values for these variables are as follows:

For `SystemClockSpeed`, choose `1` for a 4 MHz clock or `2` for an 8 MHz clock.

For `RamSize`, choose `1` for 128 KB or `2` for 256 KB.

If these variables are uninitialized or are initialized with any other values, a clock speed of 4 MHz and RAM size of 128 KB are assumed. Refer to Chapter 5, *Migrating from a Development Round Card to an OEM Round Card*, for more information on clock speed and RAM size.

Defining and Installing Interrupt Handlers

The Round Card provides interrupt pins at priority levels 3, 4, and 6, which are available to any interrupt sources on the Daughter Card. You may also use any internal modules of the microcontroller to generate interrupts that are necessary for you. (Notice, however, that you must not use the output compare OC1 of the general-purpose timer module to generate any interrupts. The protocol stack uses it for its timer handler routines.)

You must install interrupt handlers for the interrupts handled by you. To install an interrupt handler, `myinthandler`, at vector number *`vnum`*, your application must make the following stack function call:

`niInstallIntHandler(`*`vnum`*`,(unit32)myinthandler)`

where the value of `vnum` is one of the following: 27, 28, 30, or any number between 85 and 255, inclusive.

`niInstallInstHandler` returns zero upon success and nonzero upon failure. Notice that vector numbers 27, 28, and 30 are level 3, level 4, and level 6 interrupt autovectors, respectively. A vector number

between 85 and 255 is for an interrupt whose priority level is encoded in the interrupt source.

Generating Your Device Configuration

The initial configuration of your device includes the configuration of items such as the starting node address, device identification, and, optionally, the function block schedules. You must specify the parameters in the standard Windows `.ini` file format in a configuration `.ini` file. Sample configuration files are included in the `\samples` subdirectory of your installation directory. You might want to start with one of these files and edit it according to your needs. The entries in the data link configuration and system management configuration sections of your configuration `.ini` file are described in Appendix A, *Data Link Configuration Section Format* and Appendix B, *System Management Configuration Section Format.*

After you have generated your configuration `.ini` file, you must run the Configuration Generator to create a C source file that contains your configuration. This step is described in the next section, *Converting Your Device Configuration to C Code.*

Converting Your Device Configuration to C Code

To convert your Device Configuration to a compilable and linkable `.c` file, you must use the Configuration Generator utility. The Configuration Generator requires the following syntax:

`cfggen` *`iniFile cFile`*

where *`iniFile`* is the name of your configuration file, and *`cFile`* is the name of the C source file for the output C code. If your `.ini` file contains errors, the Configuration Generator halts and informs you where the errors are located. Otherwise, it creates a `.c` source file, which you use in the final step to create your binary file.

Compiling, Linking, and Downloading Your Program

During the development stage of your application, you should run your application in RAM by downloading the linker output file from your debugging environment. After the application is fully debugged, generate a binary image that can be downloaded to the Flash using the `niBurn` utility.

Note: ***The*** `nistack.lib` ***library file, which your code must link to, was created with SDS CrossCode C/C++ Compiler, version 7.11. Use either the same compiler or a compatible compiler to ensure that your code works correctly with*** `nistack.lib`***. National Instruments recommends that you use SDS CrossCode C/C++ Compiler, version 7.11, for maximum compatibility with our library file.***

The first step in the final stage is compilation. The following files must be successfully compiled to `.o` format:

- The `.c` file generated by the Device Code Generator.
- The `.c` file generated by the Configuration Generator.
- Your own `.c` file that contains your `userStart` function, your callbacks, declaration of the global variables `SystemClockSpeed` and `RamSize`, and any interrupt handlers.

You should compile the files to `.o` format using compiler options to meet the following conditions:

- You must allow single-byte enumeration where possible. For the SDS CrossCode C/C++ compiler, version 7.11, this option is `-s enum=1`.
- You must set the size of the pointer to data and the size of the pointer to function to 4 each. For the SDS CrossCode C/C++ compiler, version 7.11, these options are `-s ptrd=4` and `-s ptrf=4`, respectively.

After you have compiled the files, you must link them with the National Instruments Round Card library, `nistack.lib`. This file contains the communications stack and Function Block Shell. Make sure to do the following when linking to `nistack.lib`:

- Specify case-sensitive public and external symbols. For the SDS CrossCode linker, do *not* use the `-u` option.
- Give the correct linker specification file (using the `-f` option) that maps different regions of your object files to the correct places in RAM and ROM. See the `\samples` directory for example linker specification (`.spc`) files for the SDS CrossCode linker.
- Do not use the `-l` option to specify `nistack.lib` to the linker, because `nistack.lib` is actually a relocatable object file. Instead, specify `nistack.lib` as you would any other object file when linking.

The following step is necessary only when you are ready to download your application to the Flash.

After successfully linking your application with `nistack.lib`, extract object bytes from the linker output file and convert them into a binary image that is ready to be downloaded into the memory of the target processor. To complete this step, you need a downloader utility such as SDS CrossCode's `down`.

The memory locations of the Flash, RAM, and Frontier-1 are shown in Table 4-3.

Table 4-3. Memory Map of Fieldbus Round Card

Component	Size	Memory Window (hex)
SRAM	128 KB × 16	C0000–FFFFF
SRAM	64 KB × 16	C0000–DFFFF
Flash	128 KB × 16	00000–3FFFF
Frontier-1	2 KB × 8	80000–807FF*

* The Frontier-1 needs a 32-byte window for accessing its registers, but the minimum memory window size that can be allocated is 2 KB, so the Frontier-1 registers are aliased within this 2 KB memory space.

Burn Your Flash

After you have run the downloader, you are ready to burn your Flash and test your program.

The Fieldbus Round Card provides a method for you to burn the Flash on the Round Card without an external PROM burner device. This method involves the use of the `niBurn` utility, an RS-232 port on the host, and the serial port on the Round Card. See the *Interfacing to the Hardware* section, earlier in this chapter, for the connections of the serial port on the Serial Programming Daughter Card.

The user program must be located at physical address 0x4000, because this is the address the `niBurn` utility jumps to when the board is reset in RUN mode, or when `niBurn` has successfully downloaded the user program into the Flash in PROG mode.

Complete the following steps to burn the Flash with the `niBurn` utility:

1. Power up the Round Card in PROG Mode (see the *Hardware Description* section, earlier in this chapter, for more information about PROG Mode).
2. Confirm that pins W2(1)–W2(2) and W1(3)–W1(2) are jumpered to select RS-232 level signals.
3. Make sure that the user program is located at physical address 0x4000, because this is the address the `niBurn` utility jumps to when the board is reset in RUN mode, or when `niBurn` has successfully downloaded the user program into the Flash in PROG mode.
4. Wait for the DS1 LED to turn on.
5. Launch `niBurn`. The `niBurn` utility is located in the **NI Fieldbus Round Card** program group on the host. When you launch the `niBurn` utility, it prompts you for the name of your binary file, and the COM port to use.

   A standard serial cable leading to the Round Card must be attached to the COM port you specified. The `niBurn` utility contacts the Round Card and downloads your program into the Flash on the Round Card. The utility informs you when it has completed.
6. After the application has downloaded, remove jumper B of W3 before resetting the Round Card.

Migrating from a Development Round Card to an OEM Round Card

Chapter 5

This chapter explains how to migrate from the development version of the Fieldbus Round Card to the OEM version of the Fieldbus Round Card.

When the development and debugging phase of your application, described in the previous chapters, is complete, you are ready to move your application over to the OEM version of the Fieldbus Round Card. This move involves programming the Flash with your application during manufacturing.

Differences Between the Development and OEM Round Cards

Table 5-1 shows the differences between the development Round Card and the OEM Round Card.

Table 5-1. Features of the Development Round Card and OEM Round Card

Feature	Development Round Card	OEM Round Card
Operating Frequency	4 or 8 MHz	4 or 8 MHz
Operation At 4 MHz At 8 MHz	 9–32 V at 20 mA 9–32 V at 20 mA	 9–32 V at10 mA 9–32 V at 20 mA
RAM	256 KB	128 KB
Flash	256 KB	256 KB
Serial ROM	4 KB	4 KB

The development Round Card has 256 KB RAM so that the entire application can be moved into the RAM and debugged.

System Clock Speed Selection

You can run the Fieldbus Round Card at two speeds: 4 MHz and 8 MHz. The reason to select one clock speed over the other is the power draw from the Fieldbus. A Round Card running at 8 MHz needs more current from the Fieldbus.

The Fieldbus driver circuitry on the Round Card governs the current draw from the Fieldbus. Refer to Table 5-1 for the current draw and speed specifications of the development and OEM versions of the Round Card.

The Fieldbus driver circuitry on the development Round Card is designed to draw current needed for the 8 MHz system clock. Selecting a slower clock speed on the development Round Card does not lower the current draw. However, when you are ready to migrate to the OEM Round Card, depending on your power budget, you need to order the OEM Round Card at the correct system clock speed.

Before you compile your application, you need to set the `SystemClockSpeed` variable to the speed at which you want to run the Round Card. When the Round Card initially powers up, it starts running at 4 MHz. During the execution of your program, your desired operating frequency is set. If the `SystemClockSpeed` variable is set to 8 MHz, the Round Card operates at 8 MHz. If the `SystemClockSpeed` variable is set to 4 MHz, the Round Card operates at 4 MHz. Refer to the *Specifying the System Clock Speed and RAM Size* section of Chapter 4, *Interfacing to Your Round Card*, for information on setting the `SystemClockSpeed` variable.

Data Link Configuration Section Format

Appendix
A

This appendix explains how to structure the Data Link Configuration section of your Device Configuration `.ini` file.

The Data Link Configuration section of your Windows `.ini` Device Configuration file must be converted to C code and linked with your application before the Round Card can communicate on the Fieldbus network. The code generated by running this file through the Configuration Generator automatically configures your board. When a parameter is changed over the Fieldbus, the parameter is updated in nonvolatile memory.

Following is a description of the format of the Data Link Configuration section.

The first line of the Data Link Configuration section is as follows:

```
[Data Link]
```

The general line format for all other lines is as follows:

variable=value

where the valid variable names and values are defined in Table A-1.

Table A-1. Valid Variable Names and Values for the Data Link Configuration

Variable Name	Valid Values	Default
`devClass`	BASIC LINKMASTER	none
`nodeAddress`	0x10–0xfb	none

`devClass` indicates whether the device functions as a basic device or a link master device.

`nodeAddress` is the address of the device on the Fieldbus network. It ranges from 0x10 to 0xff. According to the *Fieldbus Foundation Specification*, addresses between 0x10 and 0xf7 are fixed addresses. A

device with a fixed address can be in operational state. You will normally configure your device to have a fixed address. Addresses between 0xf8 and 0xfb are temporary addresses. A device with a temporary address on the bus will eventually be assigned a fixed address to be operational. Addresses between 0xfc and 0xff are visitor addresses. You should not assign your device a visitor address.

A Sample Data Link Configuration section follows:

```
[Data Link]
; Comments are allowed on lines starting with a
; semicolon
devClass=BASIC
nodeAddress=0x20
```

If you specify a node address in the range 0xf8 through 0xfb, your device may show up on the bus at any default address. You may use a system configurator such as NI-FBUS Configurator to assign a fixed address to your device.

System Management Configuration Section Format

Appendix
B

This appendix explains how to structure the System Management Configuration section of your Device Configuration `.ini` file.

The System Management Configuration section of your Windows `.ini` Device Configuration file must be converted to C code and linked with your application before the Round Card can communicate on the Fieldbus network. The code generated by running this file through the Configuration Generator automatically configures your card. When a parameter is changed over the Fieldbus, the parameter is updated in nonvolatile memory.

Following is a description of the format of the System Management Configuration section:

The names of the sections in System Management Configuration are as follows:

```
[MIB]
...
[FB Schedule 0]
...
[FB Schedule 1]
...
...
[FB Schedule N]
...
```

The general line format for all other lines is as follows:

`variable=value`

where the valid variable names and values are defined in Table B-1.

☞ **Note:** ***Most of the variables in Table B-1 are optional. In fact, only the*** `devID` ***is required. For other variables, default values will be used if values are not defined in the configuration file.***

Table B-1. Valid Variable Names and Values for the MIB Sections

Variable Name	Valid Values	Implied Units	Default
clockSyncInterval	0–255	seconds	10
macrocycleDuration	32-bit unsigned integer	1/32 ms	0x8000
primaryTimeMaster	0–255	n/a	none
devID	(ASCII string identifier of this device)	n/a	none
pdTag	(ASCII string tag for this device)	n/a	blank tag
T1	0–0xffffffff	1/32 ms	0x40000
T2	0–0xffffffff	1/32 ms	0x40000
T3	0–0xffffffff	1/32 ms	0x8000

Each [FB Schedule] section denotes a single entry in the FB Schedule.

Table B-2. Valid Variable Names and Values for the FB Schedule Sections

Variable Name	Valid Values	Implied Units	Default
offset	32-bit unsigned integer	1/32 ms	none
index	0–65535	n/a	none
vfdRef	32-bit unsigned integer	n/a	none

Specifications

Appendix
C

This appendix describes the characteristics of the Fieldbus Round Card and the Serial Programming Daughter Card.

Table C-1. Fieldbus Round Card Specifications

Characteristic	Specification
Dimensions	6.2 x 1.52 cm (2.44 x 0.6 in.)
Processor	Motorola MC68331, 8/4 MHz system clock
Crystal Clock Frequency	25 kHz
Fieldbus Interface	Fuji Electric Frontier-1, bus-powered, 31.25 kb/s
Fieldbus Power Supply	9–32 V at 10 mA at 4 MHz, 50 Ω terminated or 9–32 V at 20 mA at 8 MHz, 50 Ω terminated
Power Supply to External Electronics	5 V at 4 mA
Intrinsic Safety	CENELEC: EEx ia IIC, Temperature Code: T4 at 80° C ambient

Table C-2. Fieldbus Round Card Components

Component	Location on Round Card
Fieldbus Connection	2-position terminal block JI
Daughter Card Interface	Header W1 (2 X 21 position)
BDM Port	Header W2 (2 X 5 position)
Reset Switch	Switch S1
Simulate Mode Jumper	Jumper A on W3
Program Mode Jumper	Jumper B on W3

Table C-3. Serial Programming Daughter Card Specifications and Components

Characteristic	Specification
Dimensions	6.2 x 1.52 cm (2.44 x 0.6 in.)
Daughter Card Interface	Header J4 (2 X 21 position)
Host PC to BDM Port Connector	Header W3 (2 X 5 position)
3 V/5 V/RS-232 Serial Port Connections	D-SUB 9 female connector J1 and 3-position terminal block J3
Daughter Card BDM Port Interface	Header J2
3 V/5 V/RS-232 Transmit Signal Selector	Jumper W1 (1 X 3 position) Jumper across W1(1) and W1(2)—3 V/5 V tolerant transmit signal Jumper across W1(3) and W1(2)—RS-232 transmit signal

Table C-3. Serial Programming Daughter Card Specifications and Components (Continued)

Characteristic	Specification
3 V/5 V/RS-232 Receive Signal Selector	Jumper W2 (1 X 3 position) Jumper across W2(2) and W2(3)—3 V/5 V tolerant receive signal Jumper across W2(1) and W2(2)—RS-232 receive signal

Customer Communication

Appendix
D

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation. When you contact us, we need the information on the Technical Support Form and the configuration form, if your manual contains one, about your system configuration to answer your questions as quickly as possible.

National Instruments has technical assistance through electronic, fax, and telephone systems to quickly provide the information you need. Our electronic services include a bulletin board service, an FTP site, a Fax-on-Demand system, and e-mail support. If you have a hardware or software problem, first try the electronic support systems. If the information available on these systems does not answer your questions, we offer fax and telephone support through our technical support centers, which are staffed by applications engineers.

Electronic Services



Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call (512) 795-6990. You can access these services at:

United States: (512) 794-5422
Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 01 48 65 15 59
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity



FTP Support

To access our FTP site, log on to our Internet host, `ftp.natinst.com`, as `anonymous` and use your Internet address, such as `joesmith@anywhere.com`, as your password. The support files and documents are located in the `/support` directories.



Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at (512) 418-1111.



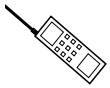
E-Mail Support (currently U.S. only)

You can submit technical support questions to the applications engineering team through e-mail at the Internet address listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

`support@natinst.com`

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the list below to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.





	Telephone	Fax
Australia	03 9879 5166	03 9879 6277
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Brazil	011 288 3336	011 288 8528
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Quebec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	01 48 14 24 24	01 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Israel	03 6120092	03 6120095
Italy	02 413091	02 41309215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	5 520 2635	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
United Kingdom	01635 523545	01635 523154
United States	512 795 8248	512 794 5678

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___)_____ Phone (___) _____

Computer brand _____ Model _____ Processor_____

Operating system (include version number) _____

Clock speed _____MHz RAM _____MB Display adapter _____

Mouse ___yes ___no Other adapters installed _____

Hard disk capacity _____MB Brand _____

Instruments used _____

_____

National Instruments hardware product model _____ Revision _____

Configuration _____

National Instruments software product _____ Version _____

Configuration _____

The problem is: _____

_____

_____

_____

_____

List any error messages: _____

_____

_____

The following steps reproduce the problem:_____

_____

_____

_____

_____

_____

Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

Interrupt level of hardware _____

DMA channels of hardware _____

Base I/O address of hardware _____

Other Products

Computer make and model _____

Microprocessor _____

Clock frequency or speed _____

Type of video board installed _____

Operating system version _____

Operating system mode _____

Programming language _____

Programming language version _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: *MC68331-Based Fieldbus Round Card User Manual*

Edition Date: January 1998

Part Number: 321866A-01

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (___)_____ Fax (___) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
(512) 794-5678

Glossary

Prefix	Meanings	Value
m-	milli-	10^{-3}
c-	centi-	10^{-2}
k-	kilo-	10^{3}
M-	mega-	10^{6}

°	Degrees
Ω	Ohms
A	Amperes
A/D	Analog-to-digital
AI	Analog input
API	Application Programmer's Interface
ASCII	American Standard Code for Information Interchange
baud	The signaling rate of a line; bits per second
BDM	Background Debug Mode
bit	A binary digit; a digit (1 or 0) in the representation of a number in binary notation
byte	Eight related bits of data
C	Celsius
D/A	Digital-to-analog
DIO	Digital input/output

DMA	Direct-memory access
EPROM	Erasable programmable read-only memory
FB	Function Block
FTP	File Transfer Protocol
GND	Ground
HART	HART Field Communications Protocol
hex	Hexadecimal
Hz	Hertz
in.	inches
I/O	Input/Output
IRQ	Interrupt request
KB	Kilobytes of memory
LED	Light-emitting diode
MB	megabytes of memory
OEM	Original equipment manufacturer
PC	Personal computer
PROM	Programmable read-only memory
RAM	Random-access memory
ROM	Read-only memory
RXD	Receive data
s	Seconds
snap	Read from the communications stack
SPI	Serial programming interface
SRAM	Serial random-access memory

SROM	Serial read-only memory
TTL	Transistor-transistor logic
TXD	Transmit data
V	volts
VCR	Virtual Communication Relationship
VFD	Virtual Field Device

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP