CORELIS An EWA Company

NetUSB II

NetUSB II
High-Speed JTAG Controller
User's Manual

13100 Alondra Blvd. Suite 102
Cerritos, CA 90703-2262
Telephone: (562) 926-6727 • Fax.: (562) 404-6196

Printing History

Edition 1: April 25, 2019

General Notice

Information contained in this document is subject to change without notice. CORELIS shall not be liable for errors contained herein for incidental or consequential damages in connection with the furnishing, performance, or use of material contained in this manual. This document contains proprietary information, which is protected by copyright. All rights reserved. No part of this document may be reproduced or translated to other languages without the prior written consent of CORELIS.

CORELIS assumes no responsibility for the use of or reliability of its software on equipment that is not furnished by CORELIS.

GPL Code Statement

The NetUSB II firmware includes compiled source code licensed under the GNU General Public License ("GPL") or GNU Lesser General Public License ("LGPL"). A copy of these licenses is included in **Appendix A: GNU Licenses**.

As part of Corelis conformance with copyright law and in accordance with the GPL and LGPL, the source code is available for general distribution WITHOUT ANY WARRANTY for a nominal fee to cover shipping and media charges. You may obtain the complete corresponding source code for a period of three years after our last shipment of this product, which will be no earlier than April 25, 2019. Please direct all inquiries to:

Email: gplcode@corelis.com

Postal Mail:
Attn: GPLSOURCE REQUEST
Corelis, Inc.
13100 Alondra Blvd. Suite 102
Cerritos, CA 90703

Environmental Notice

This product must be disposed of in accordance with the WEEE directive.

This product does not release any substances into the environment during normal use.

Trademark Notice

All products and services named in this book are trademarks or registered trademarks of their respective companies. All trademarks and registered trademarks in this book are the property of their respective holders.

Product Warranty

For product warranty and software maintenance information, see the PRODUCT WARRANTY AND SOFTWARE MAINTENANCE POLICY statement included with your product shipment.

Exclusive Remedies

THE REMEDIES CONTAINED HEREIN ARE THE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES.CORELIS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Product maintenance agreements and other customer assistance agreements are available for Corelis products. For assistance, contact your nearest Corelis Sales and Service Office.

Return Policy

No items returned to CORELIS for warranty, service, or any other reason shall be accepted unless first authorized by CORELIS, either direct or through its authorized sales representatives. All returned items must be shipped pre-paid and clearly display a Returned Merchandise Authorization (RMA) number on the shipping carton. Freight collect items will NOT be accepted. Customers or authorized sales representatives must first contact CORELIS with notice of request for return of merchandise. RMA's can only originate from CORELIS. If authorization is granted, an RMA number will be forwarded to the customer either directly or through its authorized sales representative.

Contact Information

For sales inquiries, please contact sales@corelis.com.

For any support related questions, please enter a support request at www.corelis.com/support or email support@corelis.com.

For more information about other products and services that Corelis offers, please visit www.corelis.com.

Table of Contents

Table of Figures

Table of Tables

Chapter 1: Product Overview

Introduction

The NetUSB II 4-TAP and 8-TAP High-Speed JTAG Controllers are members of the Corelis ScanExpress™ family of scan-based test, analysis, and diagnostic tools. The NetUSB II interfaces between a PC equipped with a USB2.0 or Ethernet port and any IEEE Standard 1149.1 compatible target. The NetUSB II is designed to control the operation of an IEEE Standard 1149.1 boundary-scan (JTAG) test access port (TAP) by generating the proper signals under software control to interface with the target device. It supports scan operations at continuous JTAG clock (TCK) speeds of up to 100 MHz.

Figure 1-1. NetUSB II 4-TAP (left) and NetUSB II 8-TAP (right) JTAG Controllers.

The NetUSB II facilitates software-controlled boundary-scan operations defined by IEEE Standard1149.1. It provides command access to the target's Test Access Port (TAP), accessing device internal registers and on-chip debugger, verifying PCB interconnects, performing functional testing, and debug without manual probing. The JTAG interface also provides access to internal device functions that are not accessible via external probing, enabling fault isolation within the device itself. The JTAG interface also enables programming target flash and CPLD devices, as well as data download and upload to and from the target memory devices.

What is IEEE Standard 1149.1?

The IEEE-1149.1 Test Access Port Interface (TAP) shown in Figure 1-2consists of four required signals:

- Test Data In (TDI)
- Test Data Out (TDO)
- Test Mode Select (TMS)
- Test Clock (TCK)

A fifth signal is defined as optional:

- Test Reset (TRST*)

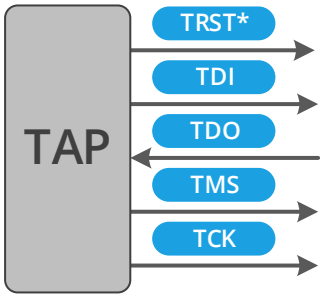

Figure 1-2. JTAG Controller Test Access Port (TAP)

Product Versions

The NetUSB II is available in a 4-TAP or 8-TAP configuration, as described in Table 1-1 below. The 4-TAP front and rear panels are pictured in Figure 1-3 below.

Table 1-1. NetUSB II Model Descriptions

Part Number	Product Name	Description
10504	NetUSB II 4-TAP	NetUSB II high-speed JTAG controller with 4 TAP connectors
10508	NetUSB II 8-TAP	NetUSB II high-speed JTAG controller with 8 TAP connectors

Figure 1-3. Net USB II 4-TAP front panel (left) and rear panel (right).

Features of the NetUSB II

The Corelis NetUSB II is a sophisticated test controller that can access devices, boards or systems compliant with IEEE Standard 1149.1.

TAP Interface

Configurable JTAG, SPI, I²C, & GPIO Signals

Each TAP includes 8 configurable signal pins. These pins can be configured to provide JTAG, serial peripheral interface (SPI), Inter-Integrated Circuit (I²C), or general-purpose input/output (GPIO) capabilities. Pin order can be customized, and each TAP can be configured with different signal types.

In addition to general functions, the GPIO signals may be designated as flash ready/busy handshake input or as external write pulse signal to significantly improve in-system-programming (ISP) speeds.

Adjustable Voltage

The software-programmable voltage level of the TAP interfaces and discrete I/O can be set to any voltage between 1.00 V and 3.30 V in increments of 0.05 V.

Programmable Clock

The NetUSB II's programmable TCK output to the IEEE Standard 1149.1 compatible target system can be generated with a wide range of frequencies using the programmable clock generator.

Pin Protection

The NetUSB II features 5 V tolerant signals, board-level ESD protection, and voltage detection circuitry to prevent the drivers from being damaged if the outputs are inadvertently shorted to power or ground.

Power Supply Short Detection

Each TAP on the NetUSB II can detect a target power supply short to ground for up to 2 voltages. This allows the NetUSB II to discover potentially damaging conditions before the target system is powered up.

Serial TAP Chaining
The NetUSB II TAPs can be programmed to daisy-chain sub-sets of the TAP ports into single chains.

Concurrent TAP Operation
During gang (concurrent) TAP operations, all targets can be scanned simultaneously. While scanning, each TAP returns scanned in information that is compared against expected patterns for verification and possible mismatch error detection. An unexpected pattern on any individual TAP will be logged but need not necessarily stop the scan session for the others. This method can be used for rapid flash memory content validation on a large set of targets concurrently.

Delay Compensation
The NetUSB II delay compensation feature allows higher clock frequencies on TCK to be used by adjusting when data on TDO coming back from the target is sampled by the hardware in half clock increments.

USB 2.0 Port Interface
The NetUSB II USB interface complies with the version 2.0 standard with backward compatibility to version 1.1 (excluding low speed). Speed adjustment is automatic per the standard. The hot-plug feature of this standard is fully supported such that attachment is easily made to a running PC, which will automatically sense its presence. For optimal performance, a host PC with a USB 2.0 port is strongly recommended.

Ethernet Port Interface
The NetUSB II Ethernet interface supports 10/100/1000Base-T with automatic speed switching and is IEEE 802.3, 802.3u and 802.3ab compatible.

Power Connector & Power Switch
The NetUSB II connects to a 12 V DC Power Supply (included). The power switch is used to turn the NetUSB II on when connected to power.

Power & Status Indicator LED
The front panel includes a green LED labeled **Power** to indicate unit status. When powered on, the NetUSB II undergoes a boot process when powered on that typically lasts approximately 30 seconds. After the boot process is completed, the LED remains on steadily to indicate that the unit is ready for use. Table 1-2 below describes the LED status modes.

Table 1-2. Power & Status Indicator States

Status	Description
Off	The NetUSB II is powered off.
Blinking green	The NetUSB II is currently booting. The boot process can take up to 30 seconds.
Solid green	The NetUSB II is ready for use.

Active Cooling
The NetUSB II includes an automatically controlled internal fan to control the unit temperature. Ventilation holes on the left and right panels of the NetUSB II allow for air flow.

> **Note:** The ventilation holes must not be obscured to allow proper airflow. Inadequate ventilation could result in damage to the NetUSB II.

Absolute Maximum Ratings

Table 1-3. Absolute Maximum Ratings

Symbol	Description	Limit Min	Limit Max	Units
V_I	DC Input Voltage	-0.30	+5.25	V
I_O	DC Output Current	-40	+40	mA
T_{STG}	Ambient Storage Temperature	-65	150	ºC
T_A	Ambient Operating Temperature	-40	85	ºC

Recommended Operating Conditions

Table 1-4. Recommended Operating Conditions

Symbol	Description	Limit Min	Limit Max	Units
V_{IH}	High Level Input Voltage	V_{ADJ} * 0.5	5.00	V
V_{IL}	Low Level Input Voltage	0.0	V_{ADJ} * 0.5	V
R_O	DC Output Resistance	45	55	Ω
I_O	DC Output Current	-25	+25	mA
T_{STG}	Ambient Storage Temperature	-40	85	ºC
T_A	Ambient Operating Temperature	0	45	ºC

Specifications

Host Computer System Requirements

Interface	Available USB 2.0-compatible or 10/100/1000Base-T Ethernet port
Operating System	Microsoft Windows 7 Microsoft Windows 8/8.1 Microsoft Windows 10 Linux (Scan Function Library only)

Host Interface Compliance

USB	2.0 High Speed
Ethernet	10/100/1000Base-T (IEEE 802.3, 802.3u or 802.3ab)

I/O Cable Connectors

USB Connector	Standard USB type B socket
Ethernet Connector	Standard RJ-45 socket
TAP Connectors	Four (4-TAP) or Eight (8-TAP) 2 × 10 pin shrouded headers 0.1 inch × 0.1 inch spacing with long ejectors (3M 3428-6302 or equivalent)

JTAG Interface

TCK frequency range	0.1 MHz to 100 MHz

SPI Interface

SCK frequency	0.1 MHz to 50 MHz

I²C Interface

SCL frequency	100 kHz

Power Requirements

12 V ±5%	4 A maximum

Physical Dimensions

NetUSB II 4-TAP	5.71 inches × 6.24 inches × 2.14 inches
NetUSB II 8-TAP	5.71 inches × 6.24 inches × 3.09 inches

Chapter 2: NetUSB II Installation & Use

Package Contents

The standard NetUSB II product package contains the following components:

- One NetUSB II High-Speed JTAG Controller Hardware (4-TAP or 8-TAP)
- One 12 V power adapter
- Four (4-TAP) or eight (8-TAP) 30 cm 20-pin shielded TAP cables
- One 2-meter USB Cable (A-to-B type)
- One 2-meter Ethernet Cable (RJ45 compatible)

Ensure all materials listed are present and free from visible damage or defects before proceeding. If anything appears to be missing or damaged, please contact Corelis at support@corelis.com or the number listed on the front cover.

It is suggested that you preserve the original packing material in the event the product must be shipped back to Corelis for service.

Microsoft Windows Software Installation

Refer to the instructions included with your software package for software installation instructions. The NetUSB II is supported under ScanExpress Tools version 8.6 or later and Corelis Scan Function Library (SFL) version 2.20 or later.

The NetUSB II uses the Remote Network Driver Interface Specification (RNDIS) included with current versions of Microsoft Windows for communication over USB. When plugged into a USB port, the controller will be listed in the device manager as a **Remote NDIS Compatible Device** as shown in Figure 2-1 below.

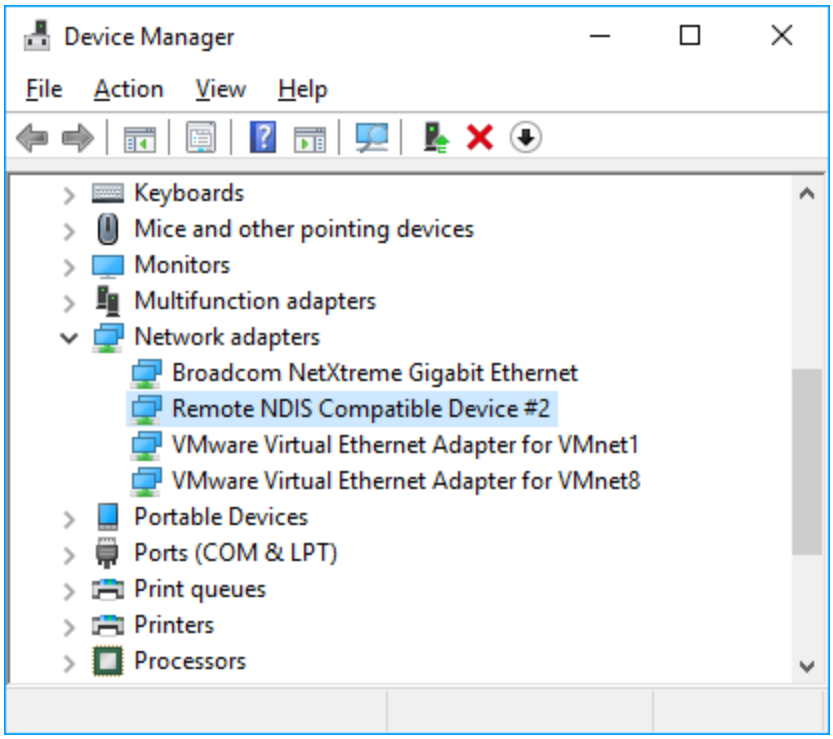

Figure 2-1 Example Entry in Windows Device Manager

> **Note:** A device driver is not required when using the NetUSB II controller in Microsoft Windows 7, 8, or 10. Unsupported device drivers are available for use with legacy versions of Microsoft Windows.

Linux Software Installation

Limited support for Linux environments is supported using the Scan Function Library (SFL). Consult the **NetUSB II SFL User's Manual** for more details on using the NetUSB II SFL.

> **Note:** The NetUSB II requires a Linux kernel with TCP/IP and RNDIS support. The NetUSB II was verified to be compatible with Ubuntu 14.04 LTS.

Hardware Installation

Using the NetUSB II Over USB

To install the NetUSB II on PC using a USB port:

1. Connect the NetUSB II to an available USB port on your PC using the supplied USB cable.
2. Connect the power adapter to the NetUSB II power connected labeled **12V DC** and plug the power adapter into an electrical outlet.
3. Toggle the switch labeled **Power** to the **ON** position.

The front panel Power LED will blink during the power up sequence and remain on when the NetUSB II is ready for operation.

Using the NetUSB II Over Ethernet

To use the NetUSB II over network connection, network parameters must first be configured using a USB connection. A configuration utility is included with the ScanExpress Tools or Scan Function Library software. To configure the network parameters:

1. Follow the steps described in the section titled **Using the NetUSB II Over USB** above.
2. Wait 30 seconds for the power up sequence to complete.
3. Execute **"C:\Program Files (x86)\Common Files\Corelis Shared\12.0\NetUSB-II_Config.exe"**.

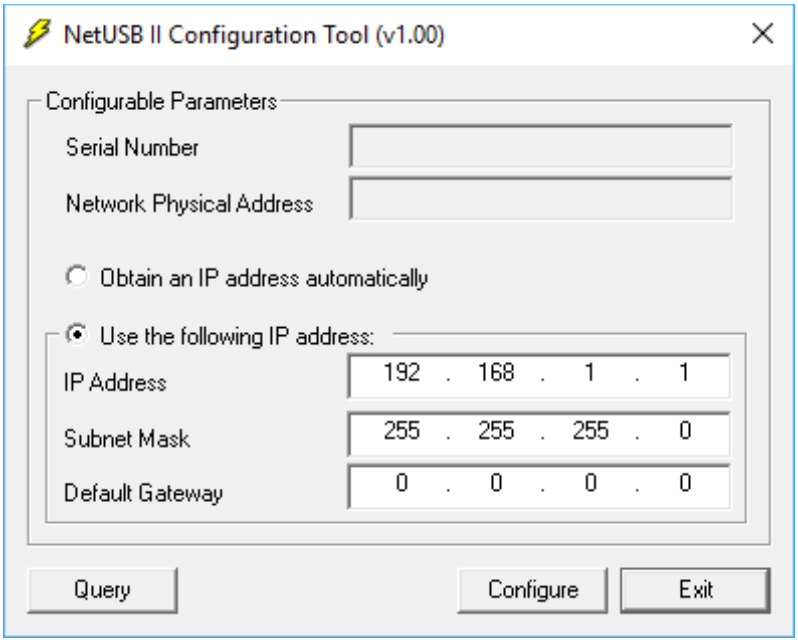

Figure 2-2. NetUSB II Network Configuration Tool

4. Enter the appropriate network parameters. For example:
   - IP Address: default value = 192.168.1.1
   - Subnet Mask: default value = 255.255.255.0
   - Default Gateway: default value = 0.0.0.0
5. Click the **Configure** button.
6. Cycle power by toggling the **Power** switch to the **OFF** position and then back to the ON position. The new settings will take effect when the NetUSB II boots.
7. Click the **Query** button to read the current configuration.

> **Note:** The NetUSB II uses port 6471 when communicating over the network. If the NetUSB II is behind a firewall, port 6471 must be opened.

Chapter 3: Use with ScanExpress Tools

Overview

The NetUSB II controller is compatible with all ScanExpress applications. The following steps are provided for ScanExpress Runner; configuration of the NetUSB II controller in the other ScanExpress applications is done in a similar fashion. Some controller configuration options are available as test step options in ScanExpress Runner. Refer to the ScanExpress Runner User's Manual for details.

Hardware Setup

You must configure the NetUSB II controller inside the ScanExpress application before the application can use it. This chapter uses ScanExpress Runner as an example to illustrate the configuration process.

NetUSB II Controller Selection

1. Make sure that the NetUSB II controller is plugged in to network port and is properly configured with desired network settings as detailed in the section titled "**Microsoft Windows Software Installation**".
2. Wait 30 seconds before starting any ScanExpress applications if you just powered up the controller.
3. launch the **ScanExpress Runner** application.
4. Click the **Setup** menu item and then select the **Controller** entry to display the **Controller Configuration** dialog.
5. Select the **NetUSB II** controller from the icons on the left. Adjust the settings to the desired values. Note that the controller uses port 6471 when communicating over the network; if the controller is behind a firewall, make sure to open port 6471.
6. After making your selections, click **Apply** to test and save the settings. When the program saves the settings successfully, it displays the controller in the **Current Controller** box. If ScanExpress Runner cannot find the controller, it will display an error dialog.

NetUSB II Controller Configuration

The main parameters of the NetUSB II controller are configured in the **Controller Configuration** dialog shown in Figure 3-1 below.

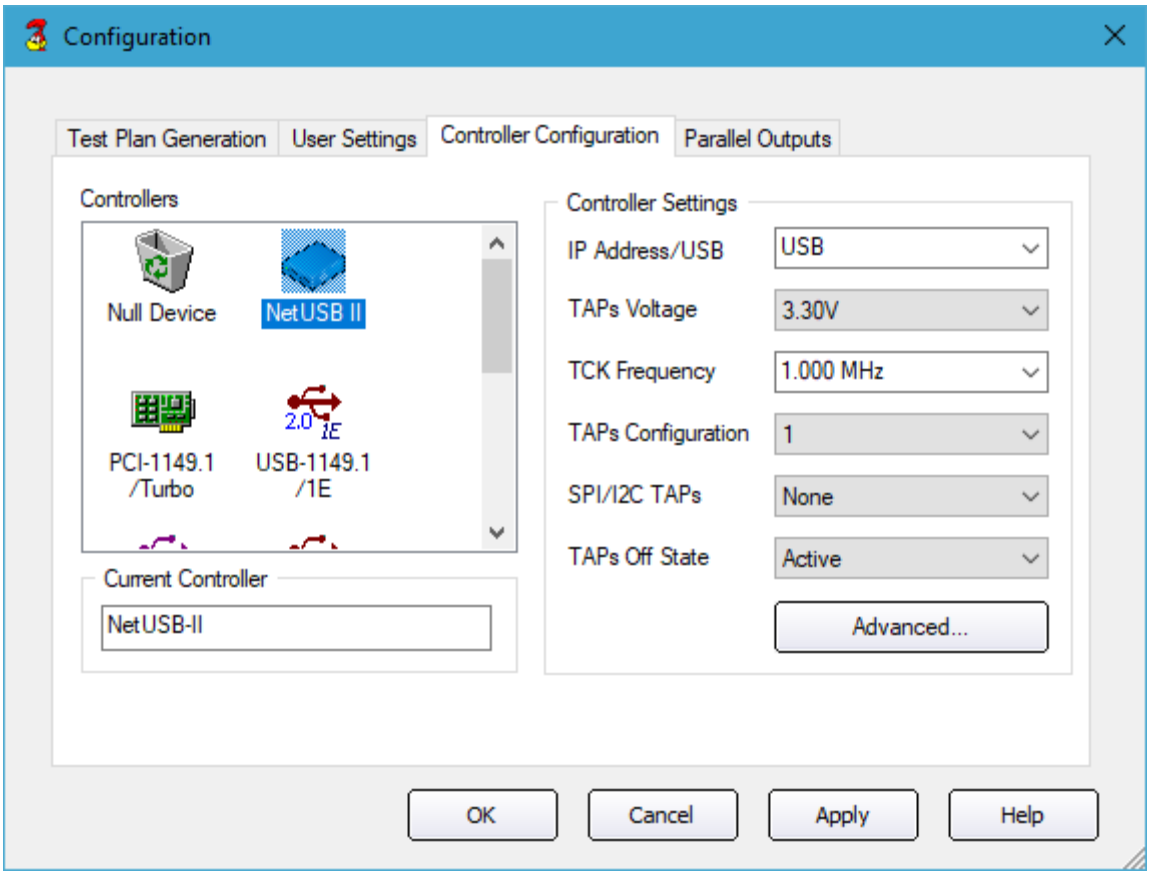

Figure 3-1. NetUSB II Controller Configuration

IP Address/USB

Select "USB" when using the USB interface or enter a network IP address when using the Ethernet interface.

TAPs Voltage

The "TAPs Voltage" setting configures all TAP signals and GPIO to a fixed voltage between 1.00 V and 3.30 V. For multiple TAP voltages, select "Advanced" and select the voltage for each TAP in the **NetUSB II Advanced Configuration** dialog.

Table 3-1. TAPs Voltage options

TAPs Voltage Selection	**Description**
1.00V to 3.30V	All TAPs configured with the selected voltage.
Advanced	Individual TAPs configured with the voltage selected in the **NetUSB II Advanced Configuration** dialog.

TCK Frequency

The "TCK Frequency" setting sets the clock rate for TCK (JTAG) and SCK (SPI). Valid selections are described in Table 3-2 below.

Table 3-2. Programmable TCK Frequencies

TCK Range (MHz)	Frequency Resolution (MHz)
0.1 to 0.9	0.1
1.0 to 100.0	1.0

> **Note:** The clock (SCL) for I2C Direct Programming is always fixed at 100 kHz.

TAPs Configuration

The "TAPs Configuration" setting selects the TAP(s) to be used. Valid selections are described in Table 3-3 below.

Table 3-3. TAPs Configuration options

TAPs Configuration	Description
1	TAP 1 (default)
2	TAP 2
3	TAP 3
4	TAP 4
5	TAP 5
6	TAP 6
7	TAP 7
8	TAP 8
1+2	Chain TAP 1 – TAP 2
1+2+3	Chain TAP 1 – TAP 3
1+2+3+4	Chain TAP 1 – TAP 4
1+…+5	Chain TAP 1 – TAP 5
1+…+6	Chain TAP 1 – TAP 6
1+…+7	Chain TAP 1 – TAP 7
1+…+8	Chain TAP 1 – TAP 8

SPI/I2C TAPs

The "SPI/I2C TAPs" setting configures a TAP for I²C or SPI Direct Programming. Valid selections are shown in Table 3-4 below.

Table 3-4. SPI/I2C TAPs options

SPI/I2C TAPs	Description
None	None (default)
1	TAP 1 used for I2C/SPI
2	TAP 2 used for I2C/SPI
3	TAP 3 used for I2C/SPI
4	TAP 4 used for I2C/SPI
5	TAP 5 used for I2C/SPI
6	TAP 6 used for I2C/SPI
7	TAP 7 used for I2C/SPI
8	TAP 8 used for I2C/SPI

TAPs Off State

When set to "Active," the boundary-scan controller will drive TAP signals between tests and when set to "Tri-State," the TAP signals will be tri-stated (not driven) between tests. Valid selections are shown in Table 3-5 below.

Table 3-5. TAPs Off State options

TAPs Off State	Description
Active	ScanExpress Runner will not tri-state the JTAG interface upon completion of a test plan execution.
Tri-State	ScanExpress Runner will tri-state the JTAG interface upon completion of a test plan execution. Enables TAP signal protection.

A TAP signal protection feature is enabled when the off state is set to "Tri-State." Under these conditions, ScanExpress Runner will test the TAP output signals (TMS, TDI, and TCK) to determine if they are shorted to power. If ScanExpress Runner detects that any of the TAP output signals are shorted to power, it will stop the execution of the test to prevent damage to the hardware.

NetUSB II Advanced Configuration

Click **Advanced** under **Controller Settings** to bring up the **NetUSB II Advanced Configuration** dialog shown in Figure 3-2 below.

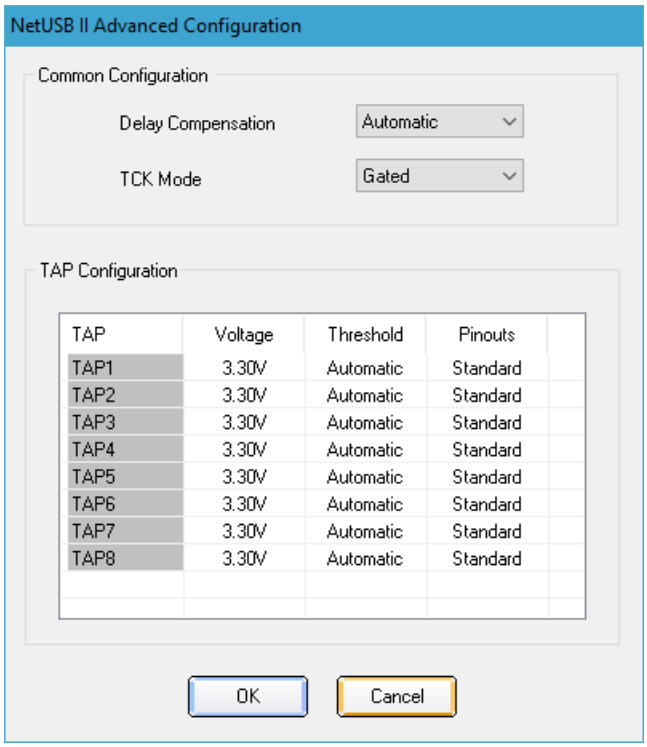

Figure 3-2. NetUSB II Advanced Configuration

Delay Compensation

The NetUSB II's delay compensation feature allows higher clock frequencies on TCK to be used by adjusting when data on TDO coming back from the target is sampled by the hardware in half clock increments. Valid selections are shown in Table 3-6 below.

Table 3-6. Delay Compensation options

Delay Compensation	Description
Automatic	Delay compensation will be automatically selected based on the selected TCK frequency. This is the default option.
No Delay	TDO will be sampled with no delay.
0.5 Clock	TDO will be sampled after 0.5 clock periods delay.
1 Clock	TDO will be sampled after 1 clock period delay.
1.5 Clocks	TDO will be sampled after 1.5 clock periods delay.
2 Clocks	TDO will be sampled after 2 clock periods delay.
2.5 Clocks	TDO will be sampled after 2.5 clock periods delay.
3 Clocks	TDO will be sampled after 3 clock periods delay.
3.5 Clocks	TDO will be sampled after 3.5 clock periods delay.

TCK Mode

The "TCK Mode" setting configures the behavior of TCK in-between scan operations and the JTAG bus is parked in one of the four stable states (`Test-Logic-Reset`, `Run-Test/Idle`, `Pause-DR`, or `Pause-IR`). Valid selections are shown in Table 3-7 below.

Table 3-7. TCK Mode options

TCK Mode	Description
Gated (default)	TCK will be held low between scan operations. This is the default option.
Continuous	The state machine will be moved to one of the four stable states (`Test-Logic-Reset`, `Run-Test/Idle`, `Pause-DR`, or `Pause-IR`) and TCK will continuously toggle between scan operations.

> **Note:** Continuous TCK mode does guarantee that TCK will always remain active. The NetUSB II may stop the clock under certain conditions.

TAP Configuration – Voltage

This set of options configures separate voltages for each TAP when the **TAPs Voltage** setting in the main controller dialog is set to "Advanced". Valid selections include "0.100V" to "3.30V."

TAP Configuration – Threshold

This set of options configures separate input signal threshold voltages for each TAP. When set to "Automatic," a threshold level equal to one half of the selected **Voltage** setting. Valid selections are shown in Table 3-8 below.

Table 3-8. TAP Configuration - Threshold options

Threshold	Description
Automatic	The input threshold will be one half of the selected **Voltage** setting.
0.50V to 2.00V	The input threshold will be equal to the selected value.

TAP Configuration – Pinouts

This set of options configures the pinout for each TAP. Valid selections are shown in Table 3-9 below.

Table 3-9. TAP Configuration - Pinouts options

Pinouts	Description
Standard	The Corelis standard pinouts will be used for JTAG and SPI/I2C TAPs. These pinouts are shown in Table 3-10.
Custom	A custom pinout will be used for JTAG and SPI/I2C TAPs. When this option is selected, the **Pinout Configuration** dialog shown in Figure 3-3 will be displayed and allow a pin to be selected for each JTAG or SPI/I2C signal.

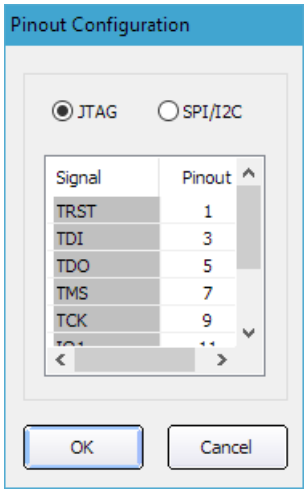

Figure 3-3. Pinout Configuration Dialog

The default pinouts for JTAG, SPI, and I2C are shown in Table 3-10 below.

Table 3-10. Standard Pinouts

Pin	JTAG	I2C	SPI
1	TRST*	SDA	SCK
2	GND	GND	GND
3	TDI	SCL	MOSI
4	GND	GND	GND
5	TDO		MISO
6	GND	GND	GND
7	TMS		CS0*
8	GND	GND	GND
9	TCK		CS1*
10	GND	GND	GND

Pin	JTAG	I2C	SPI
11	Write_Strobe* / GPIO1		CS2*
12	GND	GND	GND
13	GPIO2		CS3*
14	GND	GND	GND
15	Ready_Busy* / GPIO3		CS4*
16	GND	GND	GND
17	VCC1	VCC1	VCC1
18	NC	NC	NC
19	VCC2	VCC2	VCC2
20	NC	NC	NC

Parallel Outputs

The **Parallel Outputs** tab configures the GPIO before and after a test has run as well as upon termination of ScanExpress Runner.

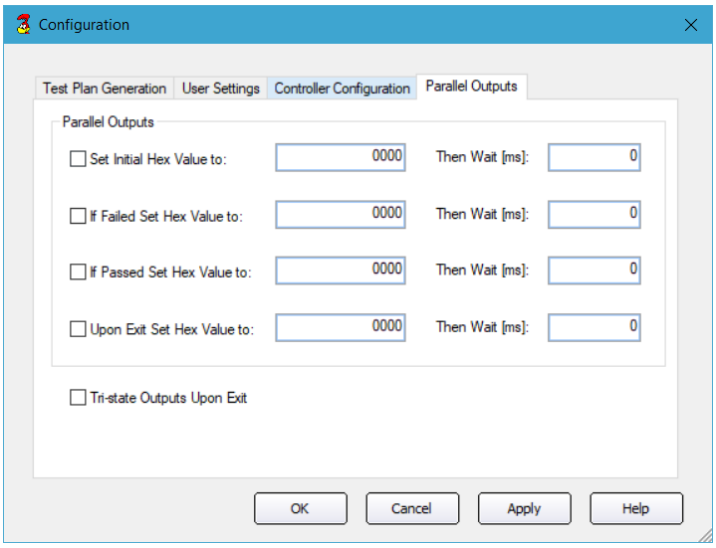

Figure 3-4. NetUSB II Parallel Outputs

Set Initial Hex Value to

Check this box to set the parallel outputs to the specified hex value then pause for the specified number of milliseconds before running a Test Plan.

If Failed Set Hex Value to

Check this box to set the parallel outputs to the specified hex value then pause for the specified number of milliseconds when a Test Plan completes with **Failed** results.

If Passed Set Hex Value to

Check this box to set the parallel outputs to the specified hex value then pause for the specified number of milliseconds when a Test Plan completes with **Passed** results.

Upon Exit Set Hex Value to

Check this box to set the parallel outputs to the specified hex value then pause for the specified number of milliseconds when ScanExpress Runner terminates.

Parallel Output Value Fields

Each TAP of a Corelis high-speed boundary-scan controller supports three individual GPIO (General Purpose I/O) signals. **Error! Reference source not found. Error! Reference source not found.** identifies the mapping of the **Parallel Outputs** window values to the physical pins on the controller. Note that when a Flash Programming (.fpi) test step uses external control signals such as `Write_Strobe*` and/or `Ready_Busy*`, flash control signal functionality takes precedence over values in the **Parallel Outputs** window. It is recommended that the user avoid using these signals to control other functions.

Table 3-11. NetUSB II Parallel Outputs Mapping

Bit	TAP	TAP Pin	Function	Alternate Function
0**Error! Reference**	1	11	`TAP 1 - GPIO1`	`Write_Strobe*`

Bit	TAP	TAP Pin	Function	Alternate Function
source not found.				
1	1	13	TAP 1 - GPIO2	
2	1	15	TAP 1 - GPIO3	Ready_Busy*
3	2	11	TAP 2 - GPIO1	Write_Strobe*
4	2	13	TAP 2 - GPIO2	
5	2	15	TAP 2 - GPIO3	Ready_Busy*
6	3	11	TAP 3 - GPIO1	Write_Strobe*
7	3	13	TAP 3 - GPIO2	
8	3	15	TAP 3 - GPIO3	Ready_Busy*
9	4	11	TAP 4 - GPIO1	Write_Strobe*
10	4	13	TAP 4 - GPIO2	
11	4	15	TAP 4 - GPIO3	Ready_Busy*

[1] Bit 0 is the least significant bit (LSB).

For example, if the initial hex value is set to "4321" and a NetUSB II controller is being used, GPIO1 of TAP1, GPIO3 of TAP2, GPIO3 of TAP3 and GPIO1 of TAP4 will be driven high and the remaining GPIO signals will be driven low. The most significant digit of the hex value ("4" in this case) is ignored.

Tri-state Outputs Upon Exit

Check this box to tri-state all GPIO pins after setting the exit value when ScanExpress Runner terminates.

Chapter 4: Third Party Application Interface

ScanExpress applications provide general purpose, third-party application programming interfaces (APIs). A "controller specification" string is used to specify the controller and settings when using these APIs. This section clarifies the requirements related to the NetUSB II controller. The examples below reference ScanExpress Runner; please refer to the specific ScanExpress application user's manual for further information.

Using the NetUSB II with ScanExpress Runner API

The "controller specification" string consists of the name of a Corelis boundary-scan controller followed by a comma-separated list of integer numbers. Each of the numbers represents a selection for different parameters of the specified controller. Please consult the **ScanExpress Runner User's Manual** for more detail.

> **Note:** The controller specification parameters correspond to the settings on the **Controller Configuration** window of the ScanExpress Runner application graphical user interface.
>
> The relevant Controller Specification String is written to the Runner test plan (.tsp) file for convenience. When using ScanExpress Runner and saving a test plan, the controller specification string is recorded with all parameters as a comment line in the header at the top of the test plan (.tsp) file. Open the .tsp file with your favorite text editor and look for the line that begins with *"ControllerParam"*. This line is write-only and is does not affect controller operation.

The NetUSB II controller uses 32 parameters described in Table 4-1 below.

Controller keyword: **NetUSB II**

Table 4-1. NetUSB II Controller Parameters

Position	Parameter	Value	Setting
1	IP Address	(string)	IP Address (i.e.: 192.168.1.2) or string "USB"
2	TAPs Voltage	1 2 … 46 47 48	1.00 V 1.05 V … (0.05 volts per step) 3.25 V 3.30 V Use advanced voltage settings in position 8 through 11
3	Clock Frequency	1 … 100 … 109	100 MHz … (1 MHz increment) 1 MHz … (0.1 MHz increment) 0.1 MHz
4	TAPs Config	1 2 … 7 8 9 10	Use TAP1 Use TAP2 … Use TAP7 Use TAP8 Use TAPs 1 and 2 in series Use TAPs 1, 2, and 3 in series

Position	Parameter	Value	Setting
		… 14 15	… Use TAPs 1 through 7 in series Use TAPs 1 through 8 in series
5	SPI/I2C TAP Config	1 2 3 … 8 9	No direct programming TAP Use TAP1 Use TAP2 … Use TAP7 Use TAP8
6	TAPs Off State	1 2	Active (JTAG signals are driven when tests are not running) Tri-State
7	Delay Compensation	1 2 3 4 5 6 7 8 9	Automatic No Delay 0.5 Clock Delay 1.0 Clock Delay 1.5 Clock Delay 2.0 Clock Delay 2.5 Clock Delay 3.0 Clock Delay 3.5 Clock Delay
8	TAP1 Voltage	1 2 … 46 47	1.00 V 1.05 V … (0.05 V per step) 3.25 V 3.30 V
9	TAP2 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
10	TAP3 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
11	TAP4 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
12	TAP5 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
13	TAP6 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
14	TAP7 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
15	TAP8 Voltage	1 … 47	1.00 V … (0.05 V per step) 3.30 V
16	TAP1	1	Automatic

Position	Parameter	Value	Setting
	Input Threshold	2 … 32	0.50 V … (0.05 V per step) 2.00 V
17	TAP2 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
18	TAP3 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
19	TAP4 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
20	TAP5 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
21	TAP6 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
22	TAP7 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
23	TAP8 Input Threshold	1 2 … 32	Automatic 0.50 V … (0.05 V per step) 2.00 V
24	TAP1 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
25	TAP2 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
26	TAP3 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*

Position	Parameter	Value	Setting
27	TAP4 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
28	TAP5 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
29	TAP6 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
30	TAP7 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
31	TAP8 Pinout Configuration	0 324508639 355965919 … 4256789809	Standard pinout configuration* (0x13579BDF): Standard pinout configuration* (0x15379BDF): Custom pinout configuration* … (Custom pinout configurations. See the *NOTE below) (0xFDB97531): Custom pinout configuration*
32	TCK Mode	1 2	Gated Continuous

Example

To select the NetUSB II controller using:

- A USB connection
- All TAP voltages set to 3.30 V
- TCK frequency set to1 MHz
- TAP1 used for JTAG
- TAP2 used for direct SPI/I2C programming
- Tri-state the JTAG signal when test is finished
- Automatic delay compensation
- Automatic input threshold
- Default pinout

To configure the NetUSB II for these options, use this "controller specification" string:

```
-controller "NetUSB-II,USB,47,31,1,3,2,1,47,47,47,47,47,47,47,1,1,1,1,1,1,1,1,
324508639,324508639,324508639,324508639,324508639,324508639,324508639,324508639,1,"
```

> **Note:** The pinout configuration value in decimal format represents the encoded pin Numbers for the JTAG, SPI, or I²C signals. Each digit of the value in hex format corresponds to the pin number of the signals.

Table 4-2. Pinout Values

Hex Digits	1st	2nd	3rd	4th	5th	6th	7th	8th
JTAG	TRST	TDI	TDO	TMS	TCK	IO1	IO2	IO3
SPI	SCK	MOSI	MISO	CS0	CS1	CS2	CS3	CS4
I2C	SDA	SCL	-	-	-	-	-	-

For example, the standard pinout configuration for JTAG is:

```
[Signal:Pin#] = [TRST:1] [TDI:3] [TDO:5] [TMS:7] [TCK:9] [IO1:11] [IO2:13] [IO3:15]
```

This would be represented as:

```
0x13579BDF (324508639 in decimal format)
```

The custom pinout configuration of

```
[Signal:Pin#] = [TRST:1] [TDI:5] [TDO:7] [TMS:9] [TCK:3] [IO1:11] [IO2:13] [IO3:15]
```

would be represented as:

```
0x15793BDF (360266719 in decimal format)
```

If an invalid number is used, the configuration will be set to the standard pinouts by default. Therefore, if you want to use the standard pinout configuration, you may just use the value *"0"* instead of *"324508639"*.

Using the NetUSB II with the Scan Function Library

The **Scan Function Library** (**SFL**) is a C interface to Corelis boundary-scan controllers which allows users to perform data and instruction scans via the IEEE Standard 1149.1 test bus. Consult the **NetUSB II SFL User's Manual** for more details on using the NetUSB II SFL.

Chapter 5: Recommended Target Connectors

10-pin TAP Connector

The Boundary-Scan TAP is a well-defined IEEE-1149.1-compatible electrical interface between boundary-scan test equipment and the boundary-scan compatible devices in the user's target board. Boundary-scan based test equipment, such as the Corelis ScanExpress family of products, utilize one or more TAPs to interface to the UUT. This section explains how to design in a simple TAP connector for your target that is compatible with most standard test equipment. The cables provided with the NetUSB II provide a connector compatible with this standard.

The controller TAP contains 5 signals: `TCK`, `TMS`, `TDO`, `TDI` and `TRST*`. It also contains ground signal(s). Corelis recommends the standard TAP connector shown in Figure 5-1 , which is widely regarded as the industry standard.

> **Note:** Corelis recommends that each signal be terminated with a resistor to improve noise immunity.

The connector on the user's target should have a standard flat cable compatible pin-out to match the TAP connector described in Table 5-1. Figure 5-1 shows the top component-side view of the basic target 10-pin connector header (0.100 × 0.100 in. spacing).

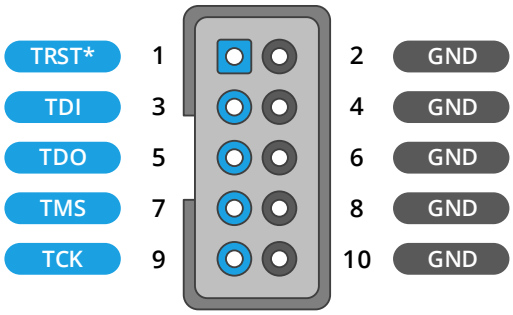

Figure 5-1. Standard 10-pin TAP Connector (top view)

Table 5-1 describes the 10-pin TAP connector signals and the Corelis recommended values of terminating resistors.

Table 5-1. 10-pin TAP Signal Description and Termination

Pin	Signal	Direction	Recommended Termination
1	TRST*	Input to the UUT	4.7 kΩ pull-up
2	GND		
3	TDI	Input to the UUT	4.7 kΩ pull-up
4	GND		
5	TDO	Output of the UUT	33 Ω series
6	GND		
7	TMS	Input to the UUT	4.7 kΩ pull-up
8	GND		
9	TCK	Input to the UUT	4.7 kΩ pull-up
10	GND		

Note: Some target boards may require a pull-down resistor on the TRST* signal to ensure normal device operations when not in boundary-scan test mode.

Table 5-2 summarizes the specifications for the 10-pin TAP connector. Equivalent connectors are available from other manufacturers.

Table 5-2. Standard 10-Pin TAP Connector

Reference	Description	Manufacturer	Part Number
10-Pin Target TAP	Straight header, 10-pin, 4-wall, with center notch	3M	3473-6610

Figure 5-2 shows a typical schematic of the target TAP connector with the recommended termination resistors. The 4.7 kΩ pull-up resistors should connect to the target V_{cc} supply corresponding to the boundary-scan device interface voltage (programmable on the NetUSB II controller from 1.00 V to 3.30V). Recommended resistor values are ±5%. Good design practices specific to the UUT and the device manufacturer recommendations take precedence over Corelis recommended values.

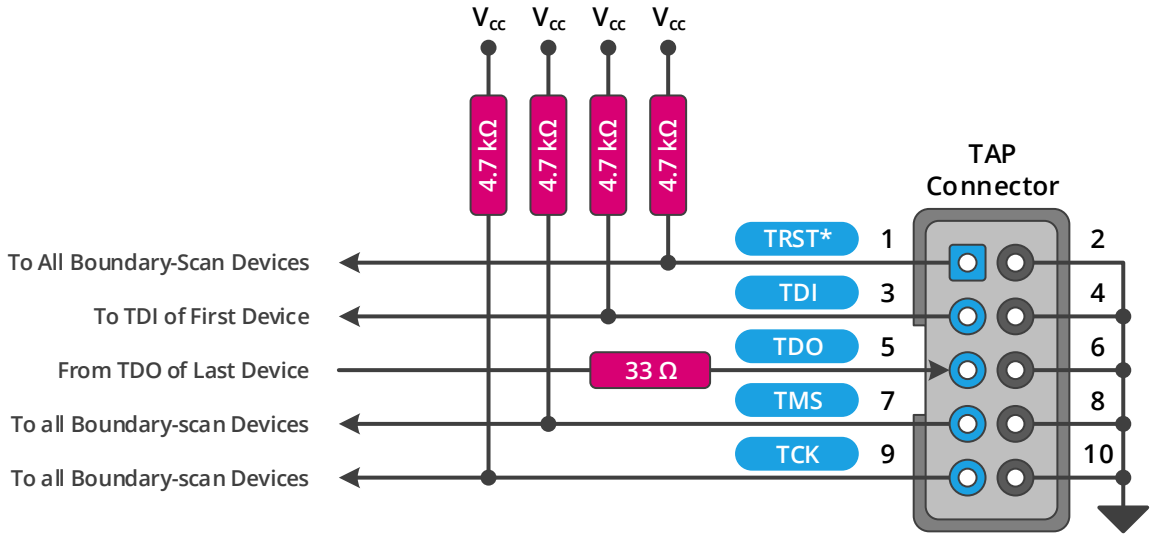

Figure 5-2. 10-pin TAP Connector Schematic

16-pin Flash Programming TAP Connector

To support expedited in-circuit programming of flash or microprocessor devices, Corelis recommends including supplemental control signals in the TAP interface. These will exploit the Corelis controllers' advanced methods to accelerate the target device programming session. The ScanExpress Programmer, for example, can use the added signals of a 16-pin TAP, like Figure 5-3, to improve programming time. Cables complying with this topology are available for the NetUSB II hardware.

This interface adds `Write_Strobe*`, `Ready_Busy*`, and additional ground signals to the standard 5-signal interface. Using the termination resistors recommended in Table 5-4 below can further improve signal quality.

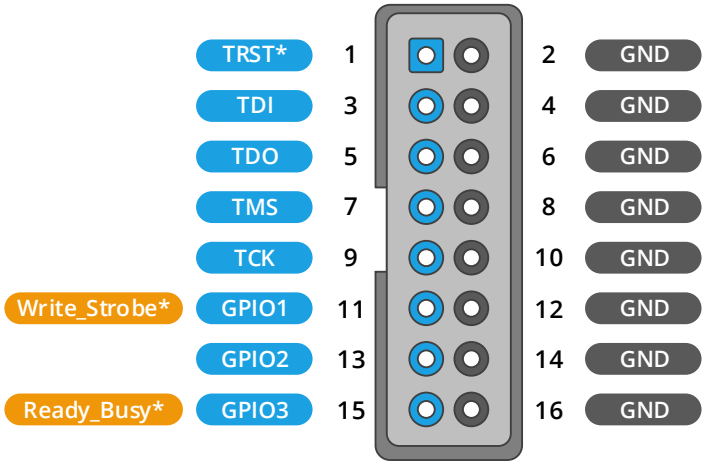

Figure 5-3. Boundary-scan Flash Programming 16-pin TAP Connector (top view)

Corelis flash programming software supports the external signals `Write_Strobe*` and `Ready_Busy*`, while still offering the standard—but slower—fully scanned out/in signals approach.

The `Write_Strobe*` signal is active low and should be pulled up with a 4.7 kΩ resistor on the target board. It should be logically *OR-ed* with the flash Write-Enable (`WE*`) signal and both inputs to the *OR* should be pulled up. This way either the target's flash `WE*` signal or the controller's external `Write_Strobe*` going low will assert the flash WE* input.

The active low `Ready_Busy*` signal is typically an open-collector/open-drain signal that ties directly to the same signal(s) on the flash device(s). This enables multiple devices to drive it toward the NetUSB II controller.

Table 5-3 summarizes the specifications for a 16-pin TAP connector without latch ejector. Equivalent connectors are available from other manufacturers.

Table 5-3. Flash Programming TAP 16-Pin Connector

Reference	Description	Manufacturer	Part Number
Flash TAP	Straight header, 16-pin, 4 wall, with center notch	3M	2516-6002UG

Table 5-4 describes the signals and Corelis recommended values of terminating resistors for a 16-pin TAP connector.

Table 5-4. 16-pin TAP Signal Description and Termination

Pin	Signal	Direction	Recommended Termination
1	TRST*	Input to the UUT	4.7K pull-up
2	GND		
3	TDI	Input to the UUT	4.7K pull-up
4	GND		
5	TDO	Output from UUT	33 Ω series
6	GND		
7	TMS	Input to the UUT	4.7K pull-up
8	GND		
9	TCK	Input to the UUT	4.7K pull-up
10	GND		
11	Write_Strobe*	Input to the UUT	4.7K pull-up
12	GND		
13	Reserved		
14	GND		
15	Ready_Busy*	Output from UUT	4.7K pull-up
16	GND		

Note: Some target boards may require a pull-down resistor on the TRST* signal to assure normal device operations when not in boundary-scan test mode.

Figure 5-4 shows a typical schematic of the target TAP connector with termination resistors. The 4.7 kΩ pull-up resistors should connect to the target V_{cc} supply corresponding to the boundary-scan device interface voltage (programmable on the NetUSB II controller from 1.00 V to 3.30 V). Recommended resistor values are ±5%. Good design practices specific to the UUT and the device manufacturer recommendations take precedence over Corelis recommended values.

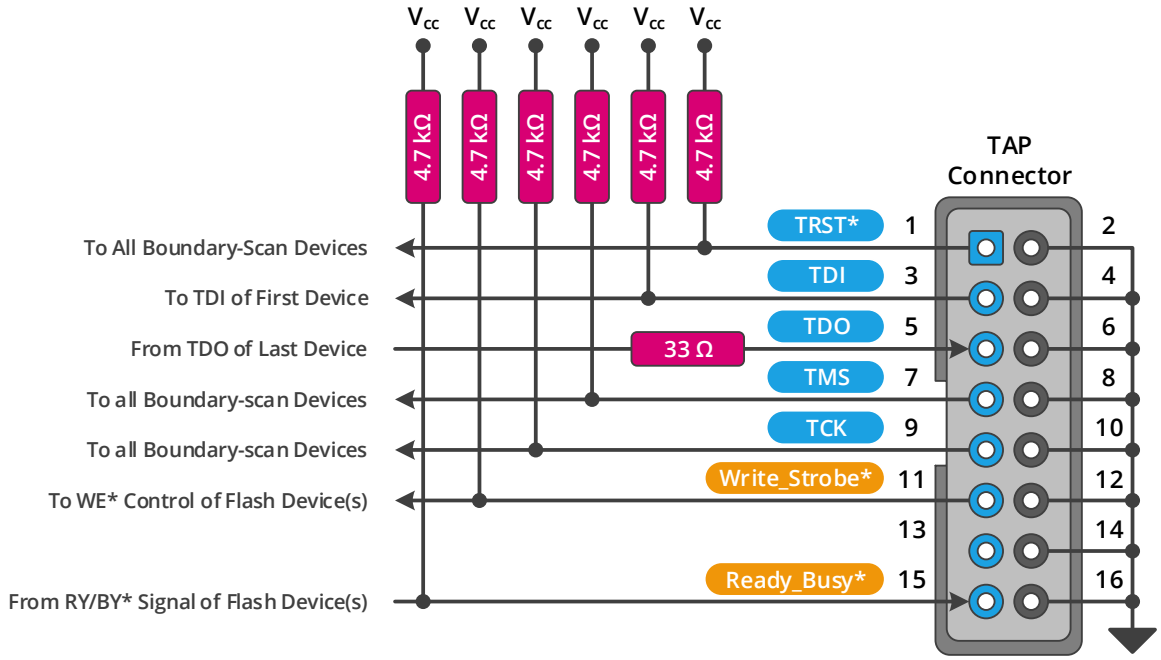

Figure 5-4. 16-pin Flash Programming TAP Connector Schematics

20-pin Flash Programming & Power Test TAP Connector

To support expedited in-circuit programming of flash or microprocessor devices, Corelis recommends including supplemental control signals in the TAP interface. These will exploit the Corelis controllers' advanced methods to accelerate the target device programming session. The ScanExpress Programmer, for example, can use the added signals of a 20-pin TAP, similar to Figure 5-5 below, to improve programming time. Cables complying with this topology are available for the NetUSB II hardware.

This interface adds `Write_Strobe*`, `Ready_Busy*`, and two power test point pins as well as additional ground signals to the standard 5-signal interface. Terminating resistors (see Table 5-5) can further improve signal quality.

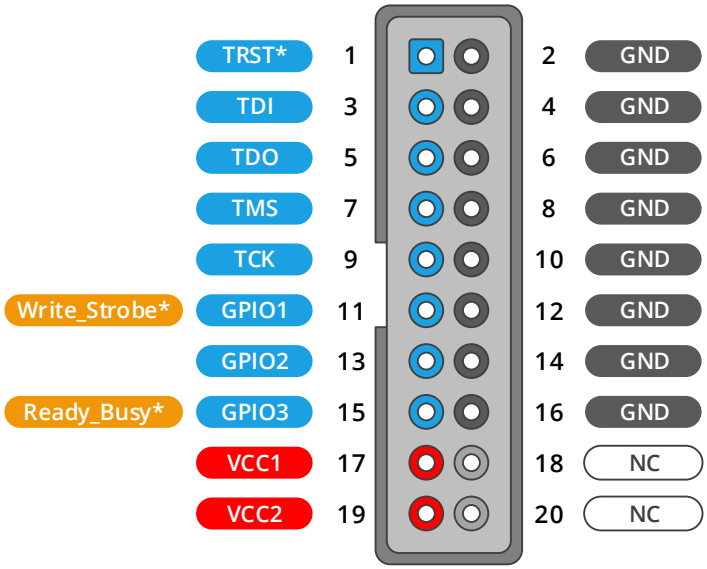

Figure 5-5. Boundary-scan Flash Programming with Power Test Points 20-pin TAP Connector (top view)

Corelis' Flash Programming software supports the external signals `Write_Strobe*` and `Ready_Busy*`, while still offering the standard, but slower, fully scanned out/in signals approach.

The `Write_Strobe*` signal is active low and should be pulled up with a 4.7 kΩ resistor on the target board. It should be logically *OR-ed* with the flash Write-Enable (WE*) signal. This way either the target's flash WE* signal or the controller's external `Write_Strobe*` going low will assert the flash WE* input.

The active low `Ready_Busy*`signal is typically an open-collector/open-drain signal that ties directly to the same signal(s) on the flash device(s). This enables multiple devices to drive it toward the NetUSB II controller.

Table 5-5summarizes the specifications for a 20-pin TAP connector without latch ejector. Equivalent connectors are available from other manufacturers.

Table 5-5. Flash Programming TAP 20-pin Connector

Reference	Description	Manufacturer	Part Number
Flash TAP	Straight header, 20-pin, 4 wall, with center notch	3M	2516-6002UG

Table 5-6 describes the signals and Corelis recommended values of terminating resistors:

Table 5-6. 20-pin TAP Signal Description and Termination

Pin	Signal	Direction	Termination
1	TRST*	Input to the UUT	4.7 kΩ pull-up
2	GND		
3	TDI	Input to the UUT	4.7 kΩ pull-up
4	GND		
5	TDO	Output from UUT	33 Ω series
6	GND		
7	TMS	Input to the UUT	4.7 kΩ pull-up
8	GND		
9	TCK	Input to the UUT	4.7 kΩ pull-up
10	GND		
11	Write_Strobe* / GPIO1	Input to the UUT	4.7 kΩ pull-up
12	GND		
13	GPIO2	Input to the UUT	4.7 kΩ pull-up
14	GND		
15	Ready_Busy* / GPIO3	Output from UUT	4.7 kΩ pull-up
16	GND		
17	VCC1	Power Test Point	None
18	NC	No Connect	None
19	VCC2	Power Test Point	None
20	NC	No Connect	None

Figure 5-6 below shows a typical schematic of the target TAP connector with termination resistors. The 4.7 kΩ pull-up resistors should connect to the target V_{cc} supply corresponding to the boundary-scan device interface voltage (programmable on the NetUSB II controller from 1.00 V to 3.30 V). Recommended resistor values are ±5%. Good design practices specific to the UUT and the device manufacturer recommendations take precedence over Corelis recommended values.

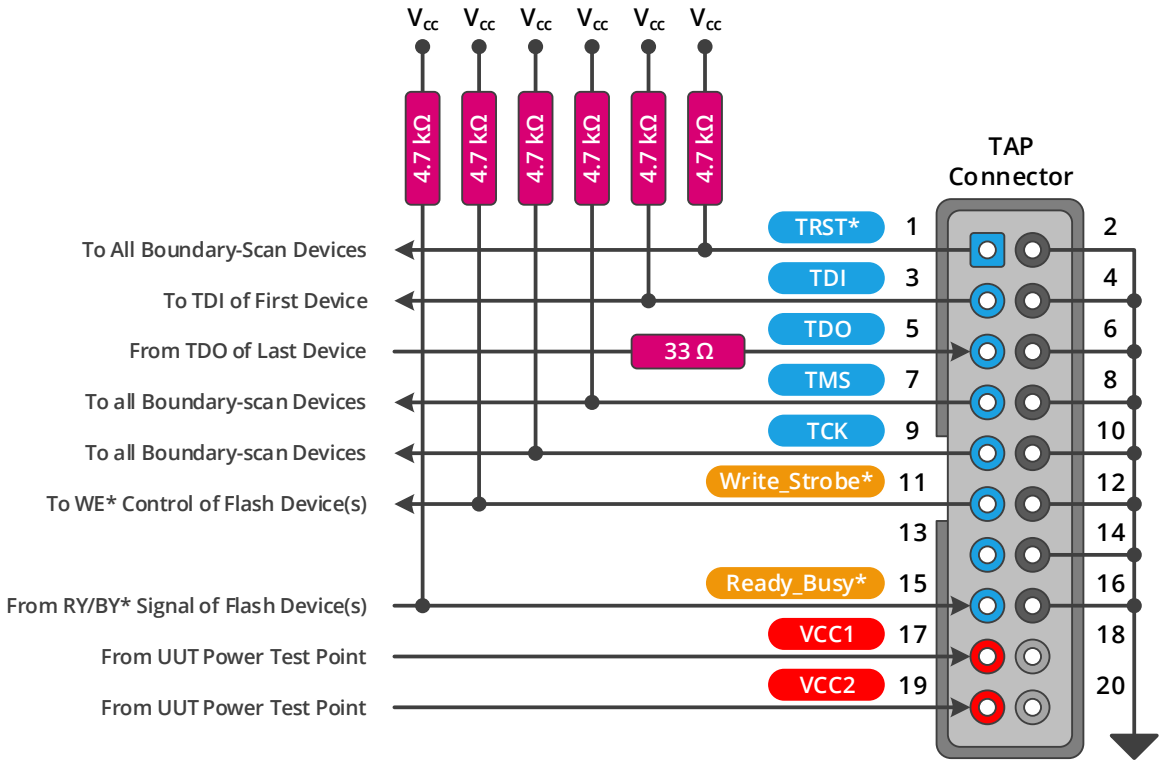

Figure 5-6. 20-pin Flash Programming with Power Test Points Connector Schematics

Direct I2C Direct Programming Connector

The default pinout for Direct I2C Programming is shown in Table 5-7 below.

Table 5-7. I2C Signal Description and Recommended Termination

Pin	Signal	Direction	Termination
1	SDA	Bidirectional	1 kΩ pull-up
2	GND		
3	SCL	Input to the UUT	1 kΩ pull-up
4	GND		

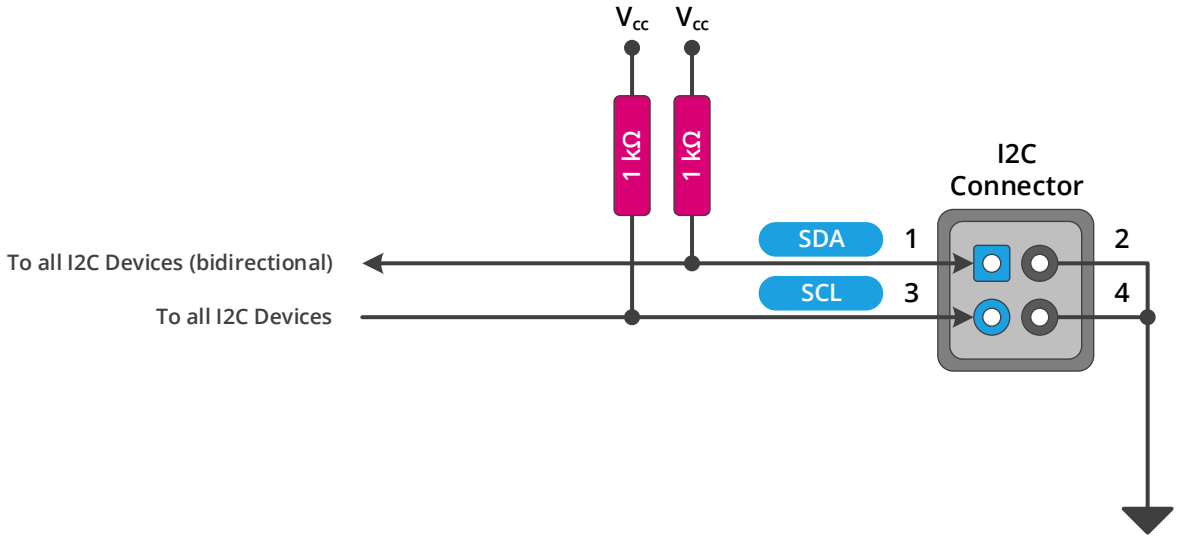

Figure 5-7. Direct I2C Programming Connector Schematics

Direct SPI Programming Connector

The default pinout for Direct SPI Programming is shown in Table 5-8 below.

Table 5-8. SPI Signal Description and Recommended Termination

Pin	Signal	Direction	Termination
1	SCK	Input to the UUT	4.7 kΩ pull-up
2	GND		
3	MOSI	Input to the UUT	4.7 kΩ pull-up
4	GND		
5	MISO	Output from UUT	33 Ω series
6	GND		
7	CS0*	Input to the UUT	4.7 kΩ pull-up
8	GND		
9	CS1*	Input to the UUT	4.7 kΩ pull-up
10	GND		
11	CS2*	Input to the UUT	4.7 kΩ pull-up
12	GND		
13	CS3*	Input to the UUT	4.7 kΩ pull-up
14	GND		
15	CS4*	Input to the UUT	4.7 kΩ pull-up
16	GND		

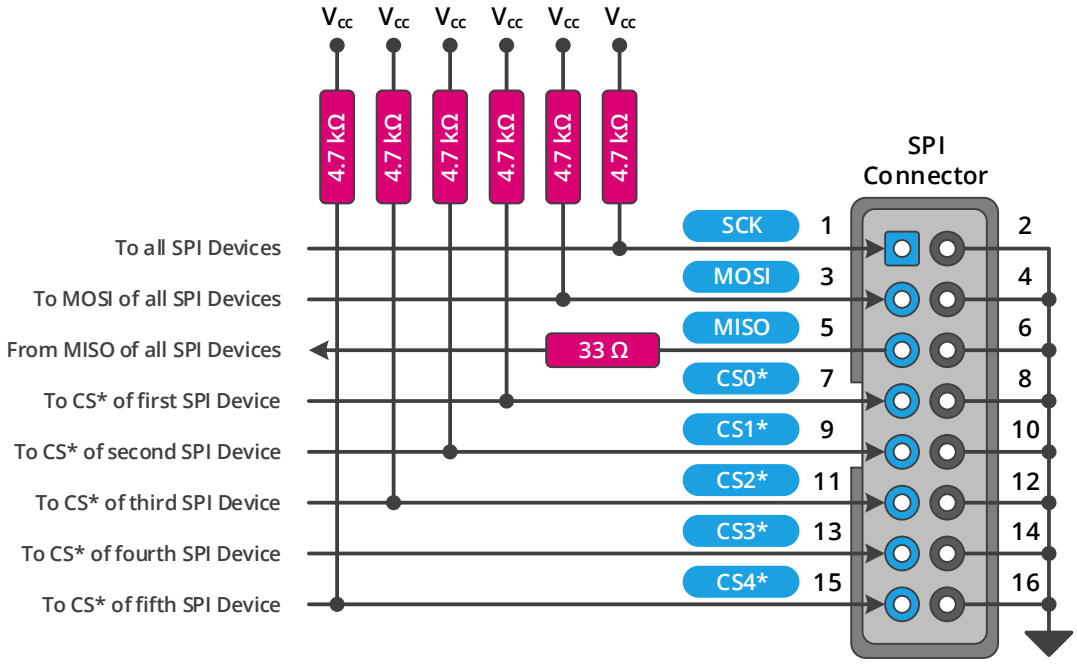

Figure 5-8. SPI Direct Programming Connector Schematics

Chapter 6: Self-Test Application

Overview

A self-test application is provided to run basic trouble-shooting diagnostics on the hardware.

Loopback Cables

The self-test includes an optional loopback capability to verify proper operation of the TAP signals.

Connect 1:1 20-pin cables (Corelis P/N 15466) between the TAPs shown in Table 6-1 or disable the loopback test steps.

Table 6-1. Self-Test Loopback Cable Connections.

Connection 1	Connection 2	NetUSB II 4-TAP	NetUSB II 8-TAP
TAP 1	TAP 3	x	x
TAP 2	TAP 4	x	x
TAP 5	TAP 7	n/a	x
TAP 6	TAP 8	n/a	x

Note: Disconnect any other targets as they will interfere with the self-test.

Running the Test

The self-test application is installed in the same folder as the ScanExpress Applications (ScanExpress Runner, ScanExpress Debugger and ScanExpress Programmer).

1. Connect the NetUSB II to an available USB or Ethernet port.
2. Connect the power adapter to the NetUSB II power connected labeled **12V DC** and plug the power adapter into an electrical outlet.
3. Toggle the switch labeled **Power** to the **ON** position. Wait approximately 30 seconds for the Status LED stops blinking.
4. Execute "**NetUSB_II_SelfTest.exe**" from "**C:\Program Files (x86)\Common Files\Corelis Shared\12.0**".
5. Select **USB** or **Net** (and enter the IP Address).
6. Enable or disable the "**Loopback Test**".
7. Click **Run**.

The program should respond with results similar to those shown in Figure 6-1.

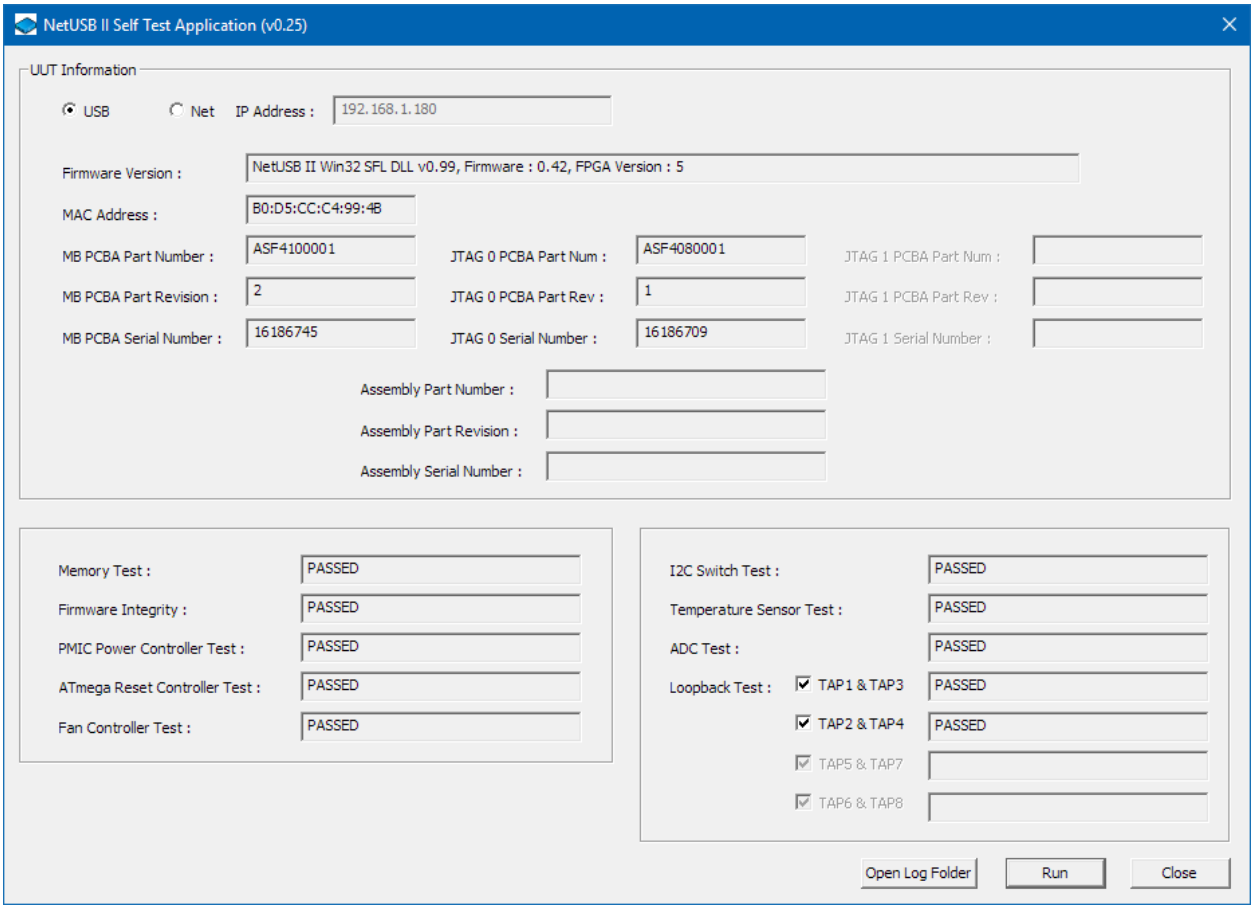

Figure 6-1. NetUSB II Self-Test Results

Chapter 7: Firmware Update Utility

Overview

The NetUSB II includes field-upgradable firmware. A firmware update utility is included with ScanExpress Tools and the Scan Function Library CD.

> **Note:** The firmware should not normally need to be modified unless the ScanExpress application or a Corelis support engineer asks you to do so.

Updating the Firmware

The firmware update utility is installed by with the ScanExpress CD or the Boundary-Scan Tools CD. The "\Firmware\NetUSB II" subdirectory contains the update utility and the new firmware file.

1. Execute "*C:\Program Files (x86)\Corelis\ScanExpress Runner v6\Firmware\NetUSB II\NetUSB-II_Flash.exe*". The Corelis NetUSB II Firmware Update Utility shown in Figure 7-1 below will be displayed.

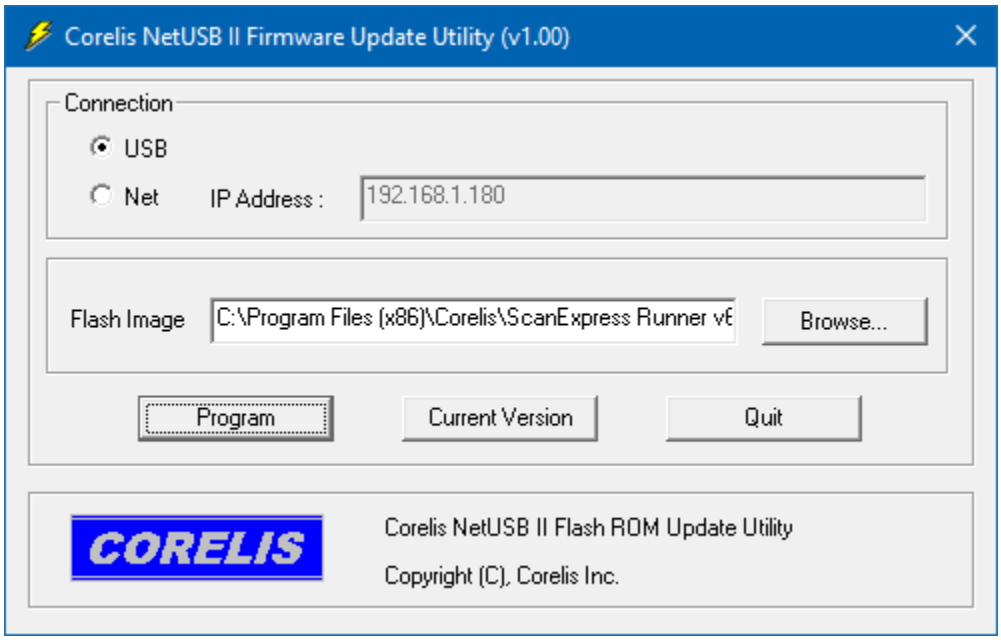

Figure 7-1. NetUSB II Firmware Update Utility

2. Select **USB** or **Net** (and enter the IP Address).
3. Click the **Browse** button and select the new firmware flash image file (.frm).
4. Click the **Program** button to start updating the firmware. The application will ask for a confirmation to proceed, as shown in Figure 7-2 below.

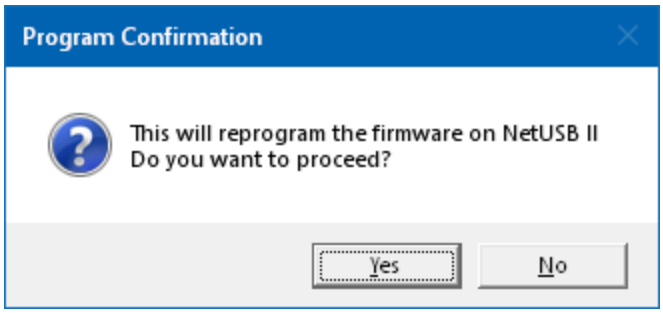

Figure 7-2. Firmware Update Confirmation Dialog

5. Click **Yes** to start the update process. The utility will show a progress bar while updating the firmware.

> **Note:** Do not power off the unit or disconnect the Ethernet/USB cable until the firmware update process is complete.

6. When finished a dialog box will appear. Click **OK**.

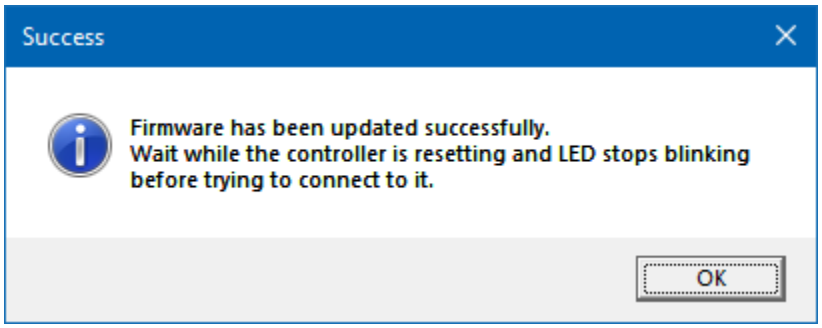

Figure 7-3. Firmware Update Success Dialog

7. Close the firmware update utility and cycle power on the NetUSB II unit.

Appendix A: GNU Licenses

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a

portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who

receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C) yyyy  name of author

This program is free software; you can redistribute it and/or modify it under the
terms of the GNU General Public License as published by the Free Software Foundation;
either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.  See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this
program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street,
Fifth Floor, Boston, MA  02110-1301, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision'
(which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

```
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

   A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

   The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

   "Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

   In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they

refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

   If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

   However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

   When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

   If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

   Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

   You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work

based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

    If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

    It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

    This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions

for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the library's name and an idea of what it does.
Copyright (C) year  name of author

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in
the library `Frob' (a library for tweaking knobs) written
by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!