BCI-1553

Product Documentation

REV	DATE	DESCRIPTION	PREPARED BY	APPROVED BY
1.0	08/17/2015	Initial version.	J. Ryan Vallieres	-

Table of Contents

List of Figures

1 Product Description

The Bloomy Controls BCI-1553 module is a redundant multi-function MIL-STD-1553 terminal designed for use in a National Instruments cRIO chassis. The BCI-1553 module supports operation as a Remote Terminal, Dual Remote Terminals, Bus Controller, or a Bus Monitor. Redundancy on the module is handled by having up to two transmission buses for any of the operation modes, each being independently configurable.

1.1 Features

1.1.1 Single or Dual Remote Terminals

The BCI-1553 supports the functionality of up to two fully independent and configurable Remote Terminals. Each Remote Terminal is addressable and maintains separate data buffers for transmit and receive data. All transmit and receive commands are supported for subaddresses 1-30. All mode codes defined and undefined by the MIL-STD-1553 protocol are supported.

1.1.2 Bus Controller

TBD

1.1.3 Bus Monitor

TBD

1.1.4 Dual Redundant Bus

The BCI-1553 module has two bus interfaces available for operating on. When used in the same network, Bus B can be used as a redundant secondary bus. Alternatively, Bus A and Bus B can be independently inhibited in each operation mode. This selective inhibit can be leveraged in the Remote Terminal operation mode to have independent Remote Terminal on two separate buses, without redundancy.

1.1.5 Low Latency I/O

Communication with the BCI-1553 is maintained over high-speed 10MHz SPI communication lines. The high speed SPI communication affords quick data transfer between custom FPGA applications and the BCI-1553.

1.2 Setup & Installation

1.2.1 How to Setup Hardware for Use

The BCI-1553 module requires several external components for interfacing and power.

1. A compatible National Instruments cRIO chassis (Minimum Suggested: Kintex 7 70T FPGA).
2. An external power source capable of providing the voltage and current outlined in section 2.1.1.
3. Cable XYZ.ABC for interfacing with the DB-15 connector.

After the BCI-1553 drivers have been installed, the BCI-1553 module can be placed in an empty slot in the cRIO chassis. Power can be connected to the front of the module and cabling run from the DB-15 interface onto the 1553 Bus. Once this is complete, the module can be discovered on a LabVIEW FPGA application.

1.2.2 How to Install Software for Use

When receiving a BCI-1553 module, drivers will accompany the module via flash drive, CD, or hyperlink. These various sources will contain an installer executable that will place the required drivers onto a target computer, allowing an installed version of LabVIEW to recognize the BCI-1553 module when it is present in a cRIO chassis.

1.2.2.1 Before Installing Drivers

Before attempting to install the BCI-1553 drivers, make sure that all versions of LabVIEW are closed on the target computer and no other install processes are actively running. Further, the following applications and software are required to successfully install the BCI-1553 drivers.

1. National Instruments LabVIEW 2014.
2. National Instruments LabVIEW 2014 FPGA and Real Time modules.

1.2.2.2 Installing BCI-1553 Drivers from a Flash Drive

1. Insert the flash drive into an available USB port on the target computer.
2. Navigate to the flash drive through Windows Explorer.
3. Run the *Setup.exe* file located at the root of the flash drive.
4. Follow the prompts presented by the installer.

1.2.2.3 BCI-1553 and LabVIEW FPGA Projects

Once the driver software is installed on a target computer, the BCI-1553 module can be discovered and/or added to an FPGA project. There are no slot requirements for the module and the number of modules allowable on a single system depends entirely on the FPGA space available. See Appendix B: FPGA Utilization for more information on the FPGA utilization of each operation mode.

The BCI-1553 module **requires a cRIO chassis to be running in FPGA mode** to be discovered or even added. In other words, the BCI-1553 module is **incompatible with National Instruments Scan Engine**.

1.2.2.4 Adding the BCI-1553 Module to a LabVIEW FPGA Project

The BCI-1553 module supports auto-discovery and manual addition to a project. The below steps outline the basic procedure for manually adding the BCI-1553 module to a LabVIEW FPGA project. Use the 'Existing target or device' option at step 2 for auto-discovery of the BCI-1553 module.

1. Add a new C Series module to the FPGA target.

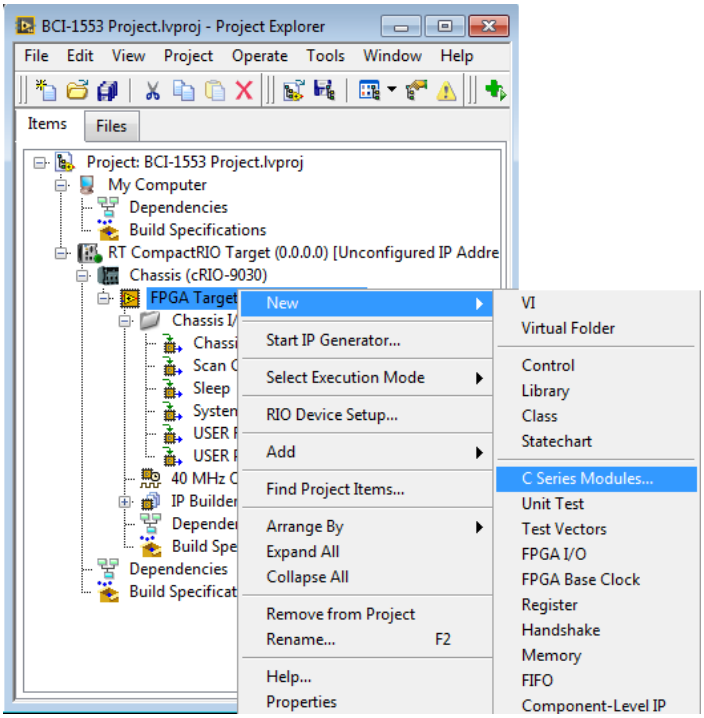

Figure 1-A: New C Series module.

2. Select to add a new target or device.

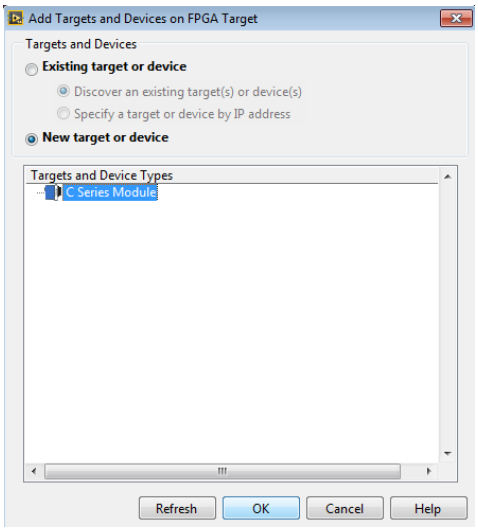

Figure 1-B: New target or device.

3. Select the module name, set the type to BCI-1553, and set the slot.

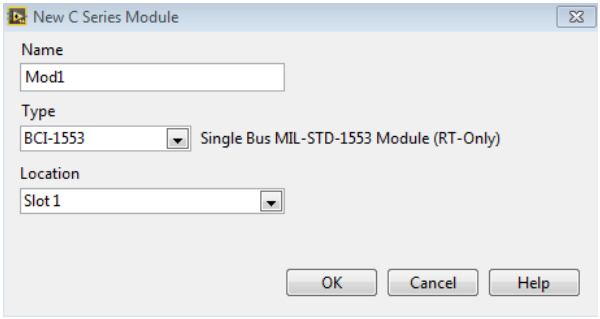

Figure 1-C: Setting the details of the BCI-1553 module to add.

4. Find the BCI-1553 module in the project.

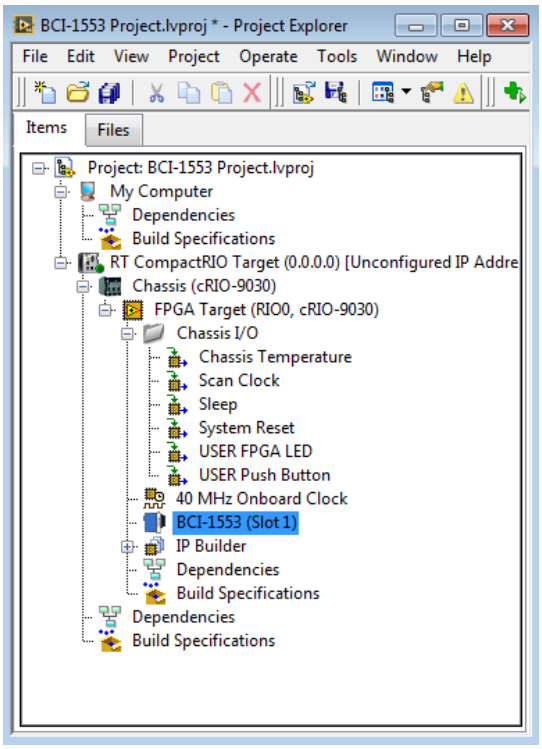

Figure 1-D: BCI-1553 module present in the project.

2 Hardware

2.1 Hardware General Overview

2.1.1 4-Contact Power Connector

This connector is the interface for an external power supply to the BCI-1553 module. An external power supply is required for the BCI-1553 to operate.

2.1.1.1 Voltage + (9V - 30V)

Pin(s): 4

The BCI-1553 module requires a stable supply of 9V - 30V for proper operation. Further, power required to maintain the operation of the module varies as the bus traffic increases relevant to the BCI-1553. When a remote terminal on the BCI-1553 is at near 100% bus load, the module requires a power supply that can maintain at least 270 mA at 24V (approx. 6.5 W).

2.1.1.2 Common

Pin(s): 3

2.1.1.3 No Connect

Pin(s): 1, 2

2.1.2 DB-15 1553 Interface

This is the primary interface to the MIL-STD-1553 buses on the BCI-1553. Additional utility functions may be accessed on this interface.

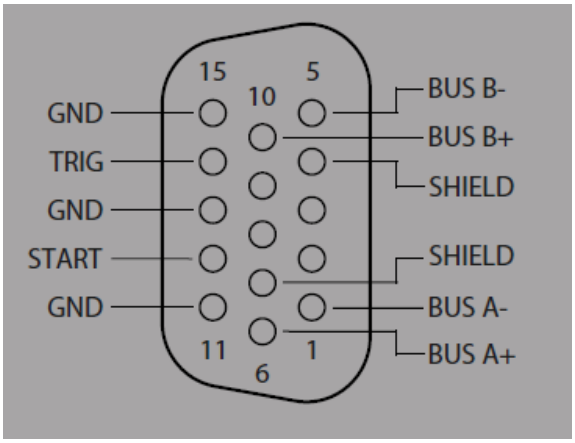

Figure 2-A: DB-15 output diagram for the BCI-1553.

2.1.2.1 Bus A+, Bus A-, Shield

Pin(s): 6, 1, 7

The primary MIL-STD-1553 bus on the BCI-1553 module. Depending on the operation mode of the BCI-1553 module, transmission on this bus can be selectively inhibited. Bus A is typically the main bus for communication with the BCI-1553 module. Behavior regarding activity on Bus A can be modified in software.

2.1.2.2 Bus B+, Bus B-, Shield

Pin(s): 10, 5, 6

The secondary MIL-STD-1553 bus on the BCI-1553 module. Depending on the operation mode of the BCI-1553 module, transmission on this bus can be selectively inhibited. Bus B is typically the redundant bus for communication with the BCI-1553 module. Behavior regarding activity on Bus B can be modified in software.

2.1.2.3 External Start

Pin(s): 12

Currently disabled for future implementation.

2.1.2.4 External Trigger

Pin(s): 14

The External Trigger requires a 3.3V TTL signal and is active high. This signal utilized exclusively by the Bus Controller operation mode of the BCI-1553 module. When this line is set active, a Bus Controller that has been initialized, started, and had a trigger configured will begin transmission of frames.

2.1.2.5 Ground

Pin(s): 11, 13, 15

2.1.3 LED Indicators

The LEDs present on the face of the BCI-1553 module are a quick way to discern the state of the module (Figure 2-B). As the module is interfaced through software, the state of each of these LEDs may change to indicate activity on the BCI-1553.

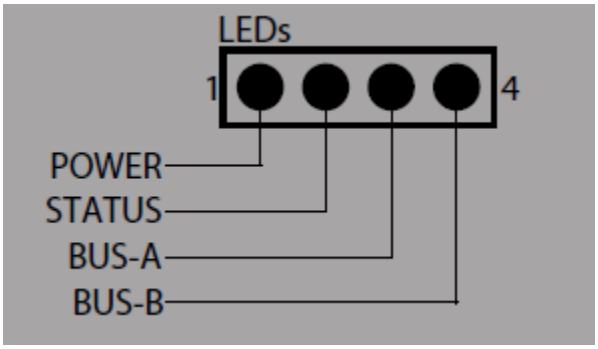

Figure 2-B: BCI-1553 LED Assignment.

2.1.3.1 Power (LED 1)

When the BCI-1553 module is connected to the backplane of a C-Series chassis and is receiving adequate power, this red LED will turn on. If there is no FPGA application running on the interfacing chassis, this LED will not turn on.

2.1.3.2 Status (LED 2)

The Status LED is used to indicate whether the BCI-1553 module is ready to be started for a particular operation mode. When in Remote Terminal mode, this LED will turn on when the 'Initialize RT' (Figure 3-C) method is called and will turn off when the 'Release RTs' (Figure 3-J) method is called.

2.1.3.3 Bus-A (LED 3)

The Bus-A green LED will flash on for brief periods of time. In general, this LED indicates that the BCI-1553 module is processing a command. This LED will not flash when Bus A is inhibited in software by the BCI-1553 module.

2.1.3.4 Bus-B (LED 4)

The Bus-B green LED will flash on for brief periods of time. In general, this LED indicates that the BCI-1553 module is processing a command. This LED will not flash when Bus B is inhibited in software by the BCI-1553 module.

3 Software

3.1 General Overview

The BCI-1553 module is capable of operating in any one of three modes: Remote Terminal, Bus Controller, and Bus Monitor. Any module can only operate in **one mode at a time**. It is not possible to have a single module operate as a Bus Controller and a Remote Terminal at the same time. However, it is possible to have three separate BCI-1553 modules present in a single chassis, each operating in one of the available modes.

Each mode has its own set of API calls and its own variable FPGA utilization. To access the API for a module, it is necessary to place an I/O Device on the block diagram of an FPGA VI (Figure 3-A). Once placed, method and property nodes can be connected to the I/O Device refnum to access the API. Please note that a 1-second delay is required to accompany the I/O Device refnum instantiation to allow the BCI-1553 module the requisite time to finish internal initialization.

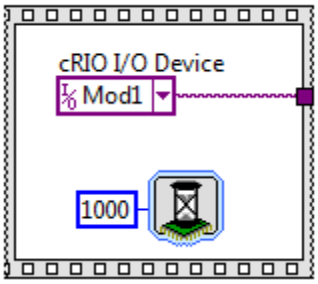

Figure 3-A: Creating a BCI-1553 Module Reference.

3.2 Common API

These methods and properties are constant across all operation modes. Most of these are read-only properties.

3.2.1 Serial Number (Property)

Returns the serial number associated with the BCI-1553 module present in the chassis (Figure 3-B).

3.2.2 Module ID (Property)

Returns the module ID associated with the BCI-1553 module present in the chassis (Figure 3-B).

3.2.3 Vendor ID (Property)

Returns the vendor ID associated with the BCI-1553 module present in the chassis (Figure 3-B).

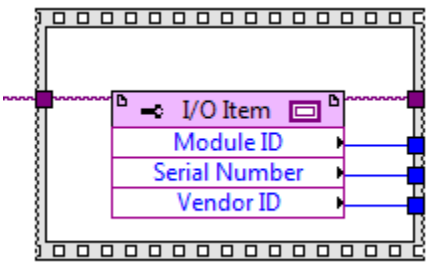

Figure 3-B: Reading the Module ID, Serial Number, and Vendor ID.

3.3 Remote Terminal

The Remote Terminal mode sets up the BCI-1553 module to act as one or two remote terminals with dual redundancy. By default, both remote terminals with transmit and receive on both Bus A and Bus B, using Bus B as the redundant bus.

3.3.1 Initialize RT (Method)

This function (Figure 3-C) is used to initialize the BCI-1553 module into Remote Terminal mode. The number of Remote Terminals can be specified, as well as the addresses of the terminals. Upon completion, this function will return two references. These are used as specifiers for each Remote Terminal used in the application. If only a single remote terminal is used, the second RT Reference will be returned as a value of 255. Reference values outside of 1 or 2 will default to referencing remote terminal 1. For every call of the 'Initialize RT' function, there should be a corresponding 'Release RTs' call after the application is complete.

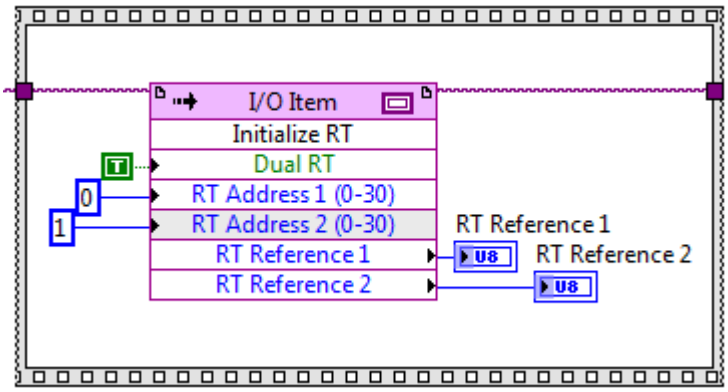

Figure 3-C: Initializing a BCI-1553 Module in dual RT mode.

3.3.2 Configure RT (Method)

Configures the general behavior of a Remote Terminal (Figure 3-D). Inhibiting either bus will prevent the Remote Terminal from attempting to transmit out on that bus. Inhibits are specific to each Remote Terminal.

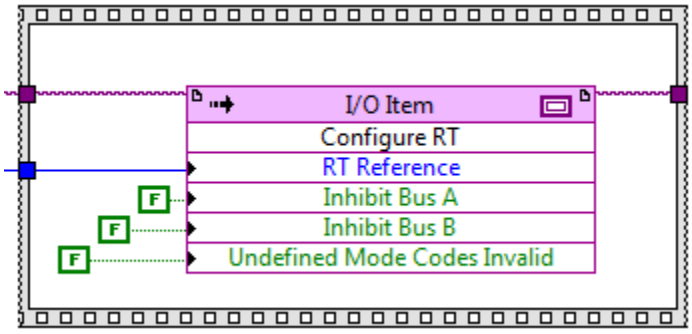

Figure 3-D: Configure the general behavior of a Remote Terminal.

Setting the 'Undefined Mode Codes Invalid' flag to true results in a subset of mode codes being unsupported by the Remote Terminal. These are mode codes that have no definition in the 1553

protocol. If this flag is set true, the Remote Terminal will only return an error status to the Bus Controller when an undefined mode code is received. These are the undefined mode codes:

Mode Codes 0-15: $T/\bar{R} = 0$

Mode Codes 16, 18, 19: $T/\bar{R} = 0$

Mode Codes 17, 20, 21: $T/\bar{R} = 1$

3.3.3 Configure Tx (Method)

Configures data to be transmitted by a Remote Terminal. Transmit data is single-buffered, unlike receive data. Further, this function is pipelined, requiring several sequential calls to push all the transmit data into memory. This is best done in a loop, setting the 'Init?' terminal to true on the first loop execution. The loop should only execute as many times as necessary to push the required data into memory. For example, see Figure 3-E.

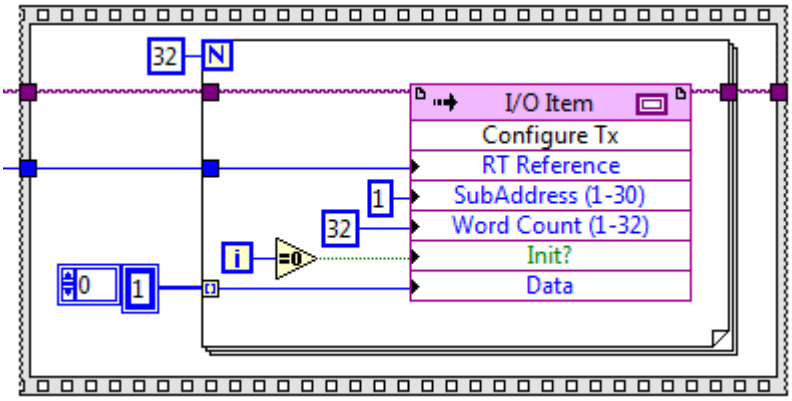

Figure 3-E: Configuring data for transmission.

The method node is placed into a FOR loop, set to statically execute 32 times (Word Count). The Data array auto-indexes into the FOR loop, retrieving a new value on each execution cycle. On the first execution of the FOR loop, the 'Init?' terminal is set to true. Every subsequent call sets the 'Init?' terminal to false. All other terminals should be wired to their corresponding value. After execution of the FOR loop, transmission memory will look like this:

>Transmit Data
[Addr0] - Data[0]
[Addr1] - Data[1]
...
[Addr31] - Data[31]

3.3.3.1 A Note on Pipelined Functions and Parallel Execution

Due to how resources are shared across all methods and properties, it is critical to ensure that all pipelined functions completely finish execution before another function is called. For example in Figure 3-E, all the FOR loop executions need to be completed before the next Remote Terminal method or property executes. Failure to do so may result in unexpected data and performance from the BCI-1553

module. This only pertains to a single instance of the BCI-1553 I/O Device refnum. Multiple BCI-1553 modules may execute in parallel freely.

3.3.4 Configure Rx (Method)

Configures Remote Terminal buffers for receiving data. The buffer length and the word length determine how much space is to be set aside for the received data. For example, see (Figure 3-F). This example allocates 88 locations in memory for word storage, including 8 utility words.

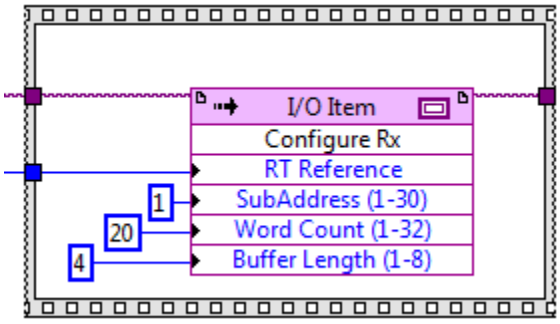

Figure 3-F: Allocating memory for a receive buffer.

The word length should match the number of words that will be received over that subaddress. The buffer length determines how many instances of received data should be stored in memory before overwriting previously received data. A buffer length of 1 results in each instance of received data overwriting the previously received instance. A buffer length of 8 requires that 8 instances of data be received before the oldest data is overwritten.

Note that per protocol requirements, each buffer instance results in two extra words allocated outside of the data received. These words include message information and timing details and are always the first two words stored in memory before an instance of received data. For example:

>Data Received (Length N)
[Addr0] - Message Information
[Addr1] - Timing Information
[Addr2] - Data0
...
[AddrN] – DataN

3.3.4.1 A Note on Configuration Methods

The two data configure methods (3.3.3, 3.3.4) can be executed after a Remote Terminal has been started (3.3.5). However, this functionality is limited in that they **can only be used for a subaddress that has been configured BEFORE the target remote terminal has been started**.

For example, if Remote Terminal 1 has transmit subaddress 1 configured before Remote Terminal 1 is started, the Configure Tx method can only be used against transmit subaddress 1 after the remote terminal has started. **In short, all Tx and Rx subaddresses that are expected to be used during command processing should be configured before the remote terminal starts**. Not following this rule will result in inconsistent behavior when transmitting or receiving data.

3.3.5 Start RT (Method)

Starts a Remote Terminal (Figure 3-G). Should be called only after the Remote Terminal is completely configured. The 'Configure Tx' and function can still be called after the Remote Terminal is started to modify the data being transmitted, however the Word Length property should remain the pre-started value since it could result in unstable transmission behavior.

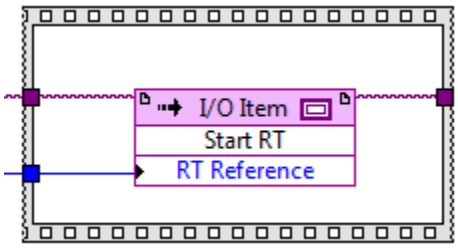

Figure 3-G: Starts a single Remote Terminal.

3.3.6 Read Rx Data (Method)

Reads data received by a Remote Terminal (Figure 3-H). This is a pipelined function that retrieves sequential data points that have been received by the Remote Terminal. It should be used in a FOR or WHILE loop, where the loop execution is limited to the data of interest. Only the first execution of the loop should set the 'Init?' terminal to TRUE.

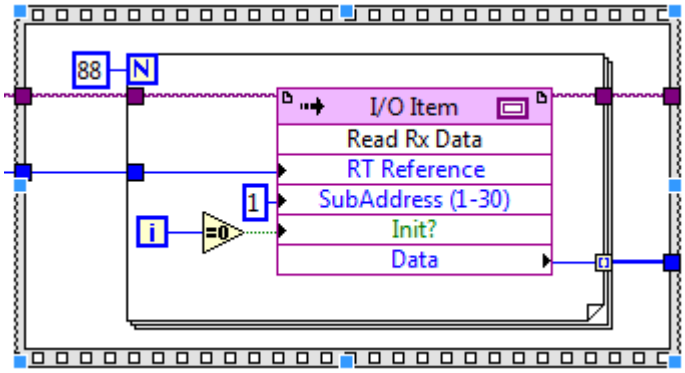

Figure 3-H: Iterating across the contents of a receive buffer.

Consider that the 'Configure Rx' function was called prior to this function, per the arguments in (Figure 3-F).To read all the data stored across all 88 possible memory locations, it would be necessary to iterate on this function 88 times in a FOR or WHILE loop. Each subsequent call would access the next memory location and return the data received. To only read the contents of the first buffer location, the loop should be set to execute 22 times. The first word read will always be a Message Information Word. The second word read will always be a timing word, giving a relative metric for when the command was received by the Remote Terminal.

3.3.7 Stop RT (Method)

Stops a Remote Terminal from transmitting or receiving data on the bus (Figure 3-I). Any data received by a remote terminal is still available to the ‘Read Rx Data’ method.

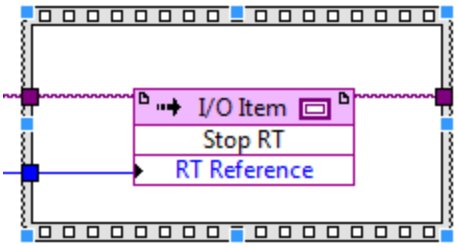

Figure 3-I: Stops a single Remote Terminal.

3.3.8 Release RTs (Method)

Releases the configured Remote Terminals and sets the BCI-1553 module into a default state (Figure 3-J). Any data received will no longer be retrievable. If Remote Terminals are running on the module, they will be stopped. This should be called at the end of the application to restore the BCI-1553 module to a known state. Not calling this function results in the BCI-1553 memory remaining in its last known state. Removing power from the module will also reset the memory to a known state.

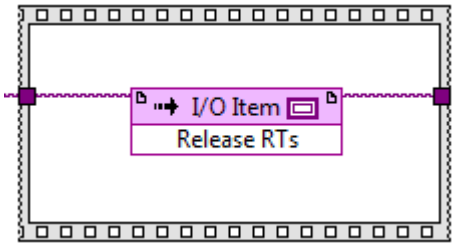

Figure 3-J: Releases a BCI-1553 module into a default state.

3.4 Bus Controller

TBD

3.5 Bus Monitor

TBD

3.6 Swapping Between Modes

TBD

3.7 Example Code

The BCI-1553 installer ships with several examples for using the BCI-1553. All of these examples assume the BCI-1553 is in the first slot of a National Instruments 9030 C-Series chassis. All examples can be found in C:\Program Files\Bloomy Controls\BCI-1553\Examples.

4 Troubleshooting

4.1 Common Scenarios

General		
Issue	**Cause**	**Fix**
Red Power LED does not turn on.	FPGA code is not running on the host C-Series chassis.	When an application is not running on a C-Series FPGA, the chassis holds all modules in a sleep state. Running a simple application can verify proper power is being supplied to the BCI-1553.
	Incorrect power being supplied to the BCI-1553 module.	Make sure that the power supplied to the BCI-1553 module conforms to the details in section 2.1
BCI-1553 module does not transmit.	Incorrect bus termination.	Check that the MIL-STD-1553 bus has been setup properly.
	Operation has not been started.	Call the start function relevant for the mode of operation the BCI-1553 module is in.
	Module not initialized	Call the initialize function relevant for the mode of operation the BCI-1553 module is in.
BCI-1553 module is not discoverable in a cRIO chassis	Incorrect or no power being supplied to module.	Make sure that the power supplied to the BCI-1553 module conforms to the details in section 2.1.
Remote Terminal data transmissions are inconsistent	Subaddress not configured before starting the Remote Terminal.	Ensure that all subaddresses relevant to the application have been configured before starting the remote terminal.

5 Maintenance

TBD

6 Appendix A: API Summary

Remote Terminal (*Bold = Output*)			
Function	**I/O**	**I/O Data Type**	**Description Section**
Initialize RT	Dual RT	Boolean	Section 3.3.1
	RT Address 1 (0-30)	U8	
	RT Address 2 (0-30)	U8	
	RT Reference 1	**U8**	
	RT Reference 2	**U8**	
Configure RT	RT Reference	U8	Section 3.3.2
	Inhibit Bus A	Boolean	
	Inhibit Bus B	Boolean	
	Undefined Mode Codes Invalid	Boolean	
Configure Tx	RT Reference	U8	Section 3.3.3
	SubAddress (1-30)	U8	
	Word Count (1-32)	U8	
	Init?	Boolean	
	Data	U16	
Configure Rx	RT Reference	U8	Section 3.3.4
	SubAddress (1-30)	U8	
	Word Count (1-32)	U8	
	Buffer Length (1-8)	U8	
Start RT	RT Reference	U8	Section 3.3.5
Read Rx Data	RT Reference	U8	Section 3.3.6
	SubAddress (1-30)	U8	
	Init?	Boolean	
	Data	**U16**	
Stop RT	RT Reference	U8	Section 3.3.7
Release RTs	Void	Void	Section 3.3.8

7 Appendix B: FPGA Utilization

Below are the typical results of using the entire API for a single BCI-1553 module in the various operation modes. These results may vary based on the optimizations used when compiling FPGA code with the BCI-1553 module present. When using multiple BCI-1553 modules in a single cRIO chassis, it is expected that FPGA utilization will increase linearly, assuming full API usage.

Just having the module present in a LabVIEW project will result in a baseline impact on FPGA utilization (approx. 50% of the below estimates). For any FPGA applications not using the BCI-1553, it is strongly suggested to make sure it is removed from the LabVIEW project before compiling FPGA applications.

Remote Terminal			
FPGA Component	**Number Used**	**% Utilized Kintex 7 70T FPGA**	**% Utilized Kintex 7 160T FPGA**
Slice Registers	13755	16.8	6.8
Slice LUTs	12208	29.8	12.9
Block RAMs	3	2.2	0.9
DSP48s	6	2.5	1.0
Bus Controller			
Slice Registers	TBD	TBD	TBD
Slice LUTs	TBD	TBD	TBD
Block RAMs	TBD	TBD	TBD
DSP48s	TBD	TBD	TBD
Bus Monitor			
Slice Registers	TBD	TBD	TBD
Slice LUTs	TBD	TBD	TBD
Block RAMs	TBD	TBD	TBD
DSP48s	TBD	TBD	TBD