BCI-1553

Remote Terminal Documentation

REV	DATE	DESCRIPTION	PREPARED BY	APPROVED BY
0.10	08/17/2015	Initial release	J. Ryan Vallieres	-
0.20	10/05/2015	Added 'Rx Data Available' function. Modified FPGA utilizations. Modification of the 'Read Rx Data' function.	J. Ryan Vallieres	-
0.30	11/10/2015	Added some changes to internal timing for reads and writes to mitigate race conditions and data integrity issues. Added an option at configuration for using receive buffering.	J. Ryan Vallieres	-
0.40	12/03/2015	Refactor of core functionality. Reduced FPGA utilization. New 'Data Transmitted' function. Modified interfaces to remove pipelined behaviors. Now utilizes small arrays for bulk data transfer. New information added to the 'Troubleshooting' section.	J. Ryan Vallieres	-
0.41	12/10/2015	Added troubleshooting information for error 65221.	J. Ryan Vallieres	-
1.0	04/19/2016	Full revision to 1.0	J. Ryan Vallieres	-
1.01	05/18/2016	Updated Common API to include 'Change Mode' method. Updated troubleshooting section for operating mode mismatch.	J. Ryan Vallieres	-

Table of Contents

List of Figures **4**

1 Product Description **5**

1.1 Features 5

1.1.1 Single or Dual Remote Terminals 5

1.1.2 Dual Redundant Bus 5

1.2 Setup & Installation 5

1.2.1 How to Setup Hardware for Use 5

1.2.2 How to Install Software for Use 5

1.2.2.1 Before Installing Drivers 6

1.2.2.2 Installing BCI-1553 Drivers from a Flash Drive 6

1.2.2.3 BCI-1553 and LabVIEW FPGA Projects 6

1.2.2.4 Adding the BCI-1553 Module to a LabVIEW FPGA Project 6

2 Hardware **9**

2.1 Hardware General Overview 9

2.1.1 4-Contact Power Connector 9

2.1.1.1 Voltage + (9V - 24V) 9

2.1.1.2 Common 9

2.1.1.3 No Connect 9

2.1.2 DB-15 1553 Interface 9

2.1.2.1 Bus A+, Bus A-, Shield 9

2.1.2.2 Bus B+, Bus B-, Shield 10

2.1.2.3 External Start 10

2.1.2.4 External Trigger 10

2.1.2.5 Ground 10

2.1.3 LED Indicators 10

2.1.3.1 Power (LED 1) 10

2.1.3.2 Act (LED 2) 10

2.1.3.3 Bus-A (LED 3) 11

2.1.3.4 Bus-B (LED 4) 11

3 Software **12**

3.1 General Overview 12

3.2 Common API 12

3.2.1 Serial Number (Property) 12

3.2.2 Module ID (Property) 12

3.2.3 Vendor ID (Property) 12

3.2.4 Change Mode (Method) 13

3.3 Remote Terminal 13
3.3.1 Initialize RT (Method) 13
3.3.2 Configure RT (Method) 14
3.3.3 Configure Rx (Method) 15
3.3.3.1 Message Information Word 15
3.3.3.2 Timing Information Word 15
3.3.3.3 A Note on Configuration Methods 16
3.3.4 Configure Tx (Method) 16
3.3.5 Start RT (Method) 16
3.3.6 Rx Data Available (Method) 17
3.3.7 Read Rx Data (Method) 17
3.3.8 Stop RT (Method) 18
3.3.9 Data Transmitted (Method) 18
3.3.10 Release RTs (Method) 18
3.4 Example Code 19
4 Troubleshooting 20
4.1 Common Scenarios 20
5 Appendix A: API Summary 22
6 Appendix B: Message Information Word 24
7 Appendix C: FPGA Utilization 25

List of Figures

Figure 1-A: New C Series module. 7
Figure 1-B: New target or device. 7
Figure 1-C: Setting the details of the BCI-1553 module to add. 8
Figure 1-D: BCI-1553 module present in the project. 8
Figure 2-A: DB-15 output diagram for the BCI-1553. 9
Figure 2-B: BCI-1553 LED Assignment. 10
Figure 3-A: Creating a BCI-1553 Module Reference. 12
Figure 3-B: Reading the Module ID, Serial Number, and Vendor ID. 13
Figure 3-C: A typical use of the Change Mode method. 13
Figure 3-D: Initializing a BCI-1553 Module in dual RT mode. 14
Figure 3-E: Configure the general behavior of a Remote Terminal. 14
Figure 3-F: Allocating memory for a receive buffer. 15
Figure 3-G: Configuring data for transmission. 16
Figure 3-H: Starts a single Remote Terminal. 17
Figure 3-I: Checks if unread receive data is available for a given SubAddress and buffer location. 17
Figure 3-J: Reading data received by the remote terminal. 17
Figure 3-K: Stops a single Remote Terminal. 18
Figure 3-L: Checks if the data stored at SubAddress 1 has been transmitted at least once. 18
Figure 3-M: Releases a BCI-1553 module into a default state. 18

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. This document specifically defines the operation of the Remote Terminal mode of the BCI-1553.

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 SubAddresses 1-30. All mode codes defined and undefined by the MIL-STD-1553 protocol are supported.

1.1.2 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.

1.2 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.

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 2015.
2. National Instruments LabVIEW 2015 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 C: 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.

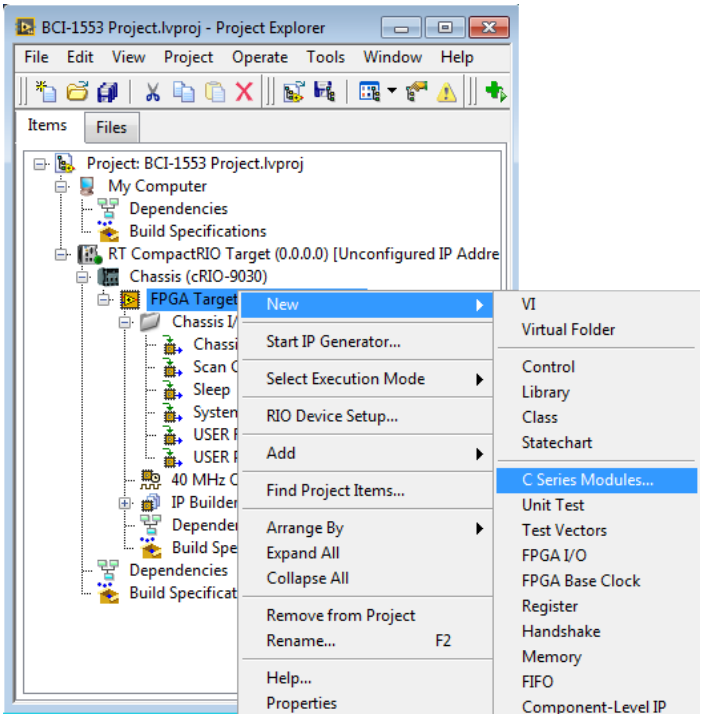

Figure 1-A: New C Series module.

2. Select to add a new target or device.

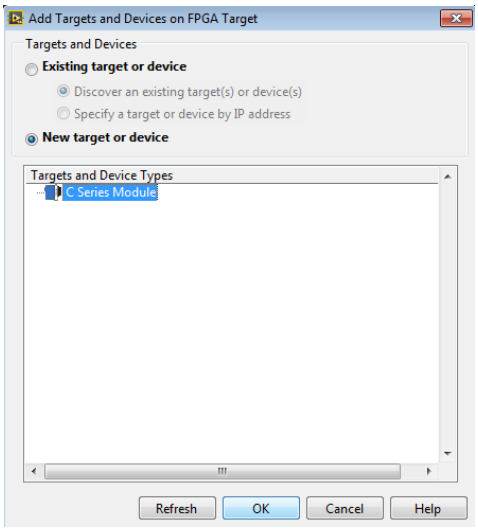

Figure 1-B: New target or device.

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

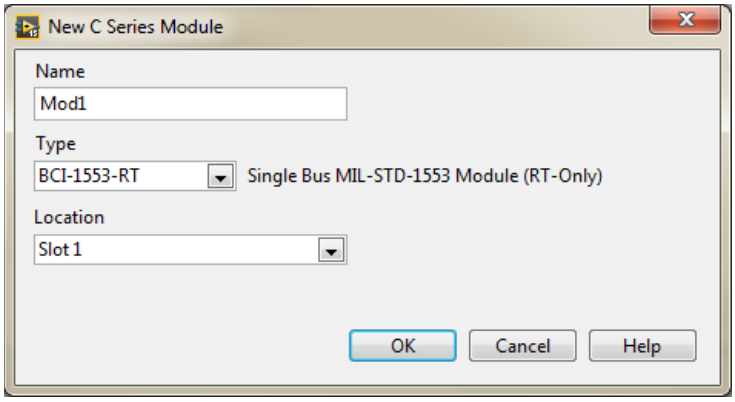

Figure 1-C: Setting the details of the BCI-1553 module to add.

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

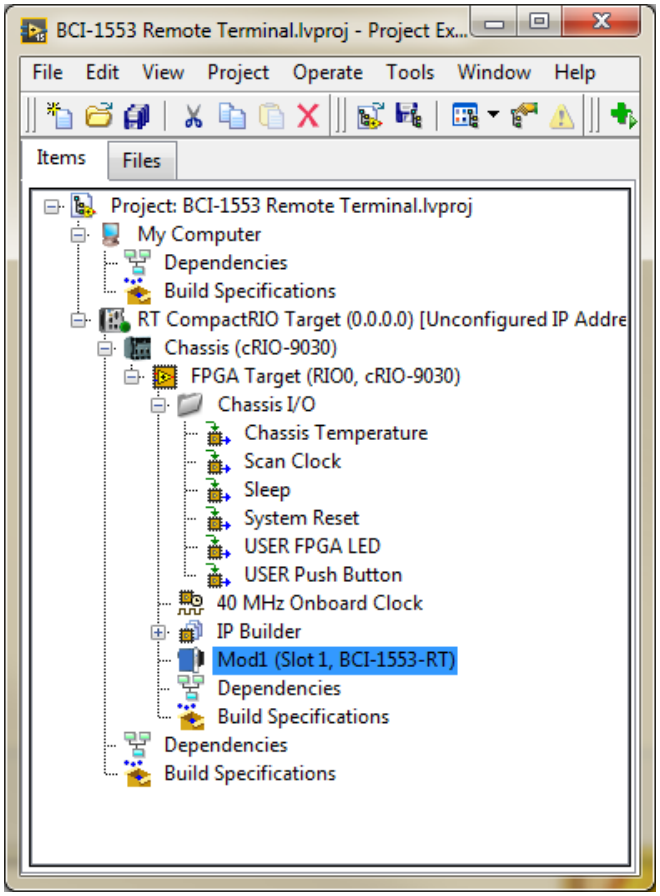

Figure 1-D: BCI-1553 module present in the project.

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 - 24V)

Pin(s): 4

The BCI-1553 module requires a stable supply of 9V - 24V 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.

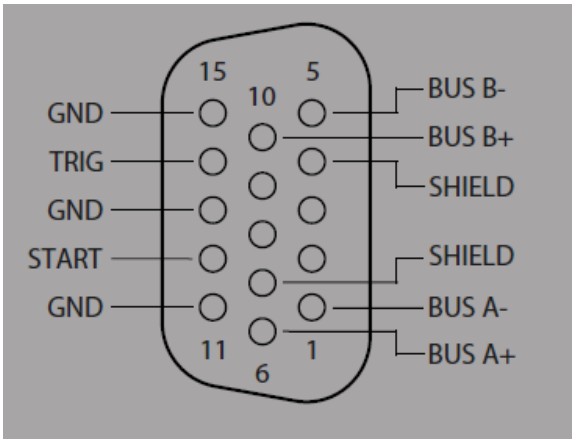

Figure 2-A: DB-15 output diagram for the BCI-1553.

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.

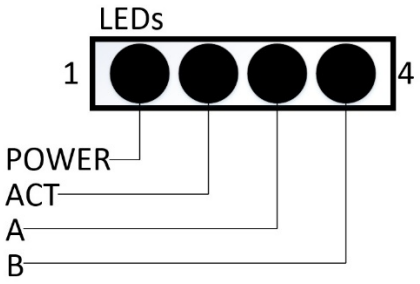

Figure 2-B: BCI-1553 LED Assignment.

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 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 Act (LED 2)

The Act 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-D) method is called and will turn off when the 'Release RTs' (Figure 3-M) method is called.

2.1.3.3 Bus-A (LED 3)

The Bus-A 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 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 four modes: Remote Terminal, Bus Controller, Bus Monitor, and Advanced. It is not currently possible to operate a module in multiple operating modes at the same time. For example, the BCI-1553 module may not be in a remote terminal mode and a bus controller mode at the same time, while using the basic Bloomy drivers. The exception to this is when using the 'Advanced' operating mode. This mode allows for extremely precise control of the internal state of the BCI-1553 module, enabling the simultaneous operation of all three operating modes at once. When in one of the basic operating modes, it is still possible to populate a single chassis with several BCI-1553 modules, each in their own operating mode.

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.

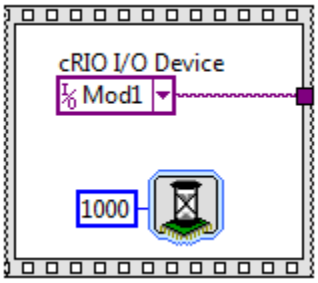

Figure 3-A: Creating a BCI-1553 Module Reference.

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).

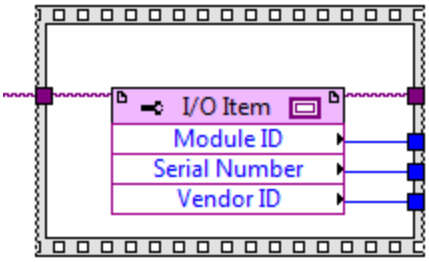

Figure 3-B: Reading the Module ID, Serial Number, and Vendor ID.

3.2.4 Change Mode (Method)

A utility for changing the operating mode of the module. This method should never be used in a final application without extreme caution.

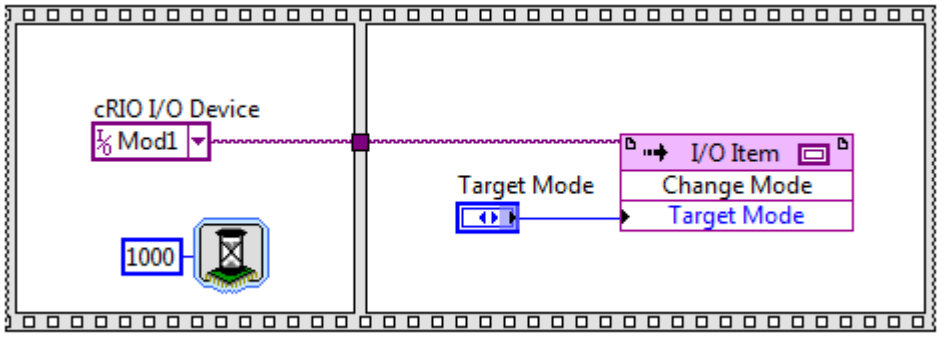

Figure 3-C: A typical use of the Change Mode method.

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-D) 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.

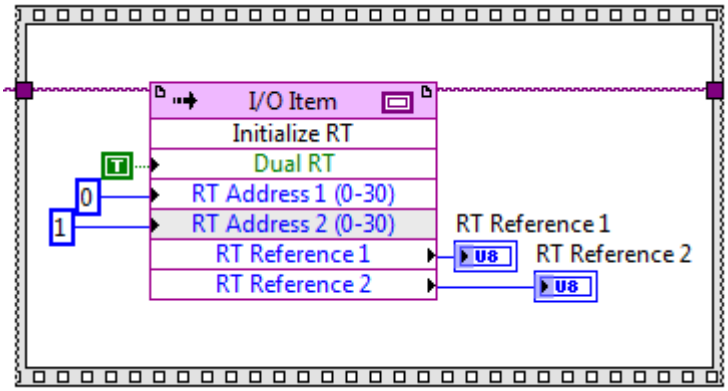

Figure 3-D: Initializing a BCI-1553 Module in dual RT mode.

3.3.2 Configure RT (Method)

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

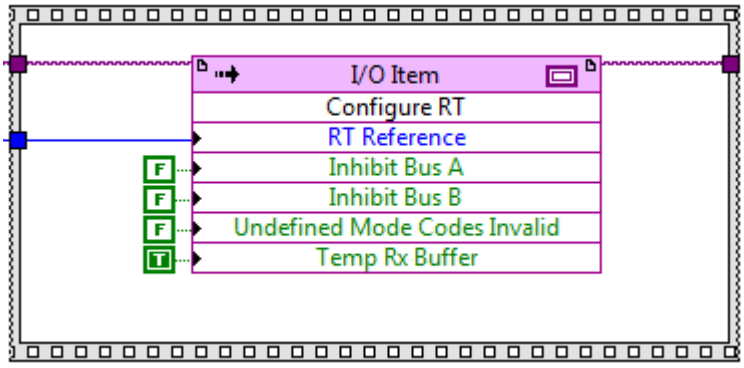

Figure 3-E: Configure the general behavior of a Remote Terminal.

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: $T/\bar{R} = 1$

Setting the 'Temp Rx Buffer' flag true sets a special form of receive data buffering. When active, data is first collected into a buffer. When the data has been completely received with no error, the data will then be pushed to readable memory. When set inactive, data is moved from temporary storage out to readable memory every 20 µs, regardless of how much data has been received.

3.3.3 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.

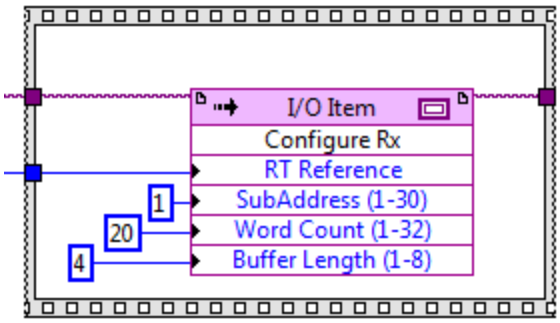

Figure 3-F: Allocating memory for a receive buffer.

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 Word
[Addr1] - Timing Information Word
[Addr2] - Data0
...
[AddrN] – DataN

3.3.3.1 Message Information Word

The Message Information Word (MIW) is stored as a part of the operation of the BCI-1553. The MIW should not be confused with the Command Word that is defined by the MIL-STD-1553 protocol. The MIW serves to provide information on the data that has been transacted between a bus controller and the receiving remote terminal. See *Appendix B: Message Information Word* for an exact breakdown of each bit definition for the MIW. This information is typically useful for debugging bus controller behavior and prototyping.

3.3.3.2 Timing Information Word

The Timing Information Word (TIW) is another word added to receive data by the BCI-1553 module. It is a relative timestamp that helps identify unique data that has been received in close proximity to each other. For example, if a receive buffer of size 4 is configured, the TIW can be used to help identify which

word was received first. A common clock is used across all remote terminals on a single BCI-1553. The current TIW only has a timing resolution of 64µs.

3.3.3.3 A Note on Configuration Methods

The two data configure methods (3.3.4, 3.3.3) 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.

3.3.4 Configure Tx (Method)

The Data terminal requires an array of static length for 32 U16 words. Creating a constant on this terminal will ensure that an array of the proper length is used. If a SubAddress requires less than 32 words of data, the Word Count terminal should be used to control the data pushed to the BCI-1553.

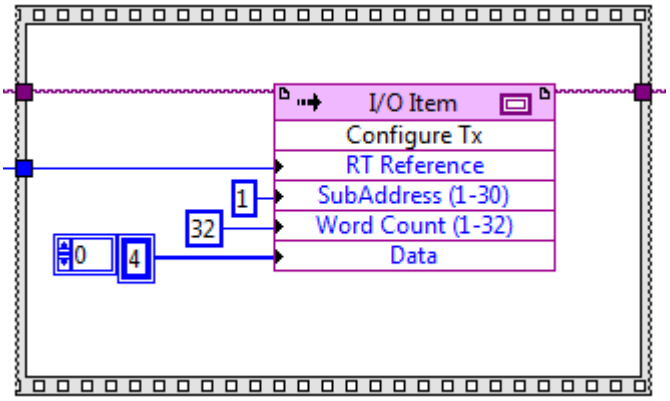

Figure 3-G: Configuring data for transmission.

3.3.5 Start RT (Method)

Starts a Remote Terminal (Figure 3-H). Should be called only after the Remote Terminal is completely configured. The 'Configure Tx' 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.

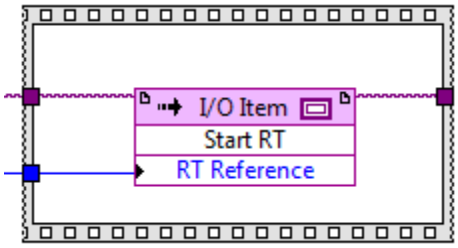

Figure 3-H: Starts a single Remote Terminal.

3.3.6 Rx Data Available (Method)

Determines if new data has been received by a Remote Terminal (Figure 3-I). This function is typically used for conditionally calling the 'Read Rx Data' function. The 'Read Rx Data' function may also be called without being pre-empted by this method.

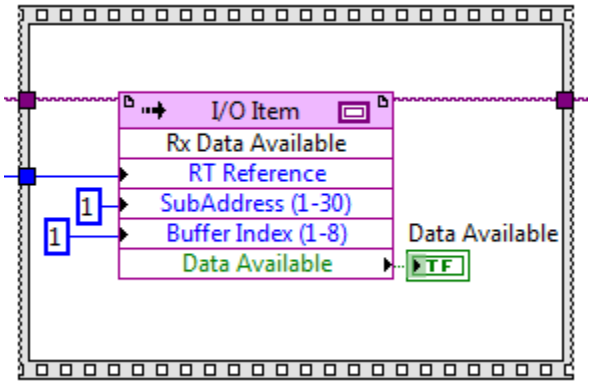

Figure 3-I: Checks if unread receive data is available for a given SubAddress and buffer location.

3.3.7 Read Rx Data (Method)

Reads data received by a Remote Terminal (Figure 3-J). The first word read will always be a Message Information Word (MIW). The second word read will always be a Time Information Word (TIW), giving a relative metric for when the command was received by the Remote Terminal. The TIW returned with this function is cleared from the BCI-1553 memory after execution. This function will always output an array with a static length of 34 elements. The word length determines how many words are placed in that array in addition to the MIW and TIW.

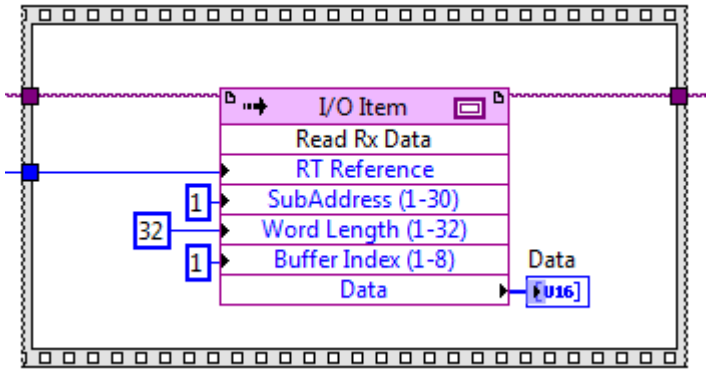

Figure 3-J: Reading data received by the remote terminal.

3.3.8 Stop RT (Method)

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

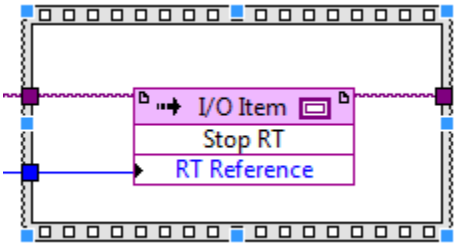

Figure 3-K: Stops a single Remote Terminal.

3.3.9 Data Transmitted (Method)

Informs the user when data has been transmitted at least once from the remote terminal and SubAddress of interest (Figure 3-L). Upon calling this method, the user can optionally clear the flag and continue to check for the next transmission instance. The transmission flag is forced to reset when the 'Configure Tx' method is called (3.3.4).

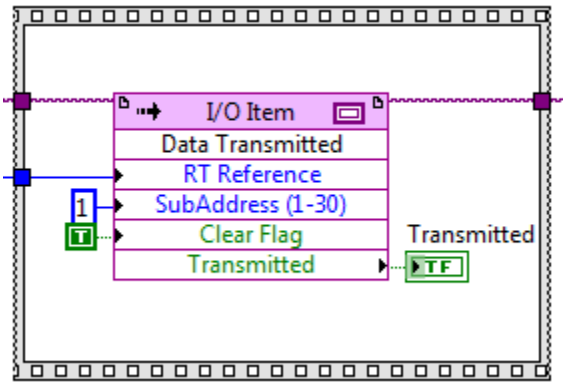

Figure 3-L: Checks if the data stored at SubAddress 1 has been transmitted at least once.

3.3.10 Release RTs (Method)

Releases the configured Remote Terminals and sets the BCI-1553 module into a default state (Figure 3-M). 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.

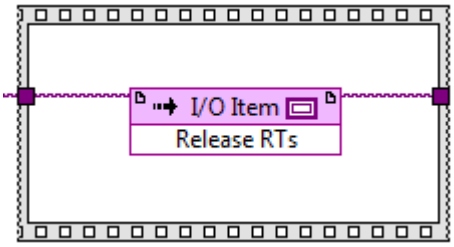

Figure 3-M: Releases a BCI-1553 module into a default state.

3.4 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 <LabVIEW 2015 Directory>\examples\Bloomy Controls\BCI-1553-RT.

4 Troubleshooting

4.1 Common Scenarios

General		
Issue	**Cause**	**Fix**
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.
	Driver not installed.	Make sure that the BCI-1553 driver is installed on the target machine for the correct version of LabVIEW.
	Module not added to the FPGA of the C-Series chassis.	Make sure that the BCI-1553 module is added as a module to the FPGA resource. The module does not support being added to the chassis, which is a scan-engine interface.
Error 7 Encountered at FPGA compile time	Conflicting driver versions.	This error typically only occurs when two conflicting versions of the RT driver set are present on the same machine. This issue can be avoided completely by uninstalling the original driver ***BEFORE*** installing a new version. If encountering this issue, please check that only the latest version of the RT driver has been installed to this location: ..\ National Instruments\LabVIEW 2014\Targets\NI\FPGA\cRIO\other\

		Delete the 'BCI-1553-RT' directory here and attempt to reinstall the driver.
Error 65221 Encountered at FPGA compile time	Conflicted method node state.	When updating from a previous driver version to a new one, LabVIEW does not immediately detect changes in the API. This results in behavior where method nodes and property nodes can have incorrect data wired to them, or indicators wired to input terminals, etc… To resolve this, replace the BCI-1553-RT device node on the block diagram and reselect all affected methods. The methods can be selected by clicking on the method node and selecting the same method name (or the equivalent) from the dropdown list. Once this selection is made, you should be able to detect the error by a broken run arrow.
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.
Module does not appear to perform any methods or properties properly.	Conflicting operating mode.	The vast majority of BCI-1553 modules are shipped in the Remote Terminal operating mode and must be transitioned to the target operating mode. To do this, verify the current operating mode of the module by using auto-discovery in your LabVIEW project. Once the current operating mode has been determined, use the driver for that operating mode to transition the module to the desired operating mode. Verify the module transitioned mode as intended by using auto-discovery in the LabVIEW project again.

5 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	
	Temp Rx Buffer	Boolean	
Configure Rx	RT Reference	U8	Section 3.3.3
	SubAddress (1-30)	U8	
	Word Count (1-32)	U8	
	Buffer Length (1-8)	U8	
Configure Tx	RT Reference	U8	Section 3.3.4
	SubAddress (1-30)	U8	
	Word Count (1-32)	U8	
	DataArray32	U16	
Start RT	RT Reference	U8	Section 3.3.5
Rx Data Available	RT Reference	U8	Section 3.3.6
	SubAddress (1-30)	U8	
	Buffer Index (1-8)	U8	
	Data Available	**Boolean**	
Read Rx Data	RT Reference	U8	Section 3.3.7
	SubAddress (1-30)	U8	
	Word Length (1-32)	U8	
	Buffer Index (1-8)	U8	
	DataArray34	**U16**	
Read MCRx Data	ProjectItemID	U8	Section 3.3.7
	RT Reference	U8	
	Mode Code (0-31)	U8	
	Data	**U16**	
Stop RT	RT Reference	U8	Section 3.3.8
Data Transmitted	RT Reference	U8	Section 3.3.9
	SubAddress (1-30)	U8	
	Clear Flag?	Boolean	
	Transmitted?	**Boolean**	

Release RTs	Void	Null	Section 3.3.10

6 Appendix B: Message Information Word

The Message Information Word (MIW) is used by the BCI-1553 to provide details and debug information on data received by the remote terminal of interest. It is not the same as the Command Word that is defined by the MIL-STD-1553 protocol. Each bit in the MIW serves to flag a certain type of failure in received data. In the case of receive data failure, the MIW will be set and no data will be received. If there is no failure in the receive command, all data will be received and the MIW will have a value of zero. The table below describes the various flags and bit definitions in the MIW. All flags are active high.

Message Information Word Bit Assignment		
Bit(s)	**Name**	**Description**
15	Time-Out Error	Set active during RT-RT transmission when the transmitting remote terminal does not start data transmission before timeout conditions occur.
14	Invalid Word Error	Set active when a Manchester encoding error or parity error occurs in the received word.
13	Gap Error	Set active when there is an unexpected delay in the transmission of consecutive data words or if extra data was detected immediately after all expected words were received.
12	Word Count Error	Set active when the data following a command word does not match the defined length. For example, the command word sets a word length of 5, but only 3 data words are received by the remote terminal.
11	Sync Error	Set active when there is an error in the sync type (command/status) of the received data words.
10	Message Error	Set active when there is an error in processing a received command. See bits 7 and 11-15.
9	Was Busy	Set high when a remote terminal responds to a receive command with a 'BUSY' status. Data words should still be buffered as normal.
8	Illegal Command Received	Does not apply to this driver version.
7	RT-RT Error	Set active when an error occurs in transmission of data from one remote terminal to another. These are typically caused by incorrect command word formatting or parity errors.
6	RT-RT Success	Set active when RT-RT transmission was successful.
5	Bus ID	Set active when the message was transmitted on BUS B.
4-0	Word Count	Contains the word count extracted from the command word. Zero indicates 32 words.

7 Appendix C: FPGA Utilization

Below are the typical results of using the entire API for a single BCI-1553 module in the remote terminal operating mode. 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**
Total Slices	3368	32.9	12.5
Slice Registers	11293	13.8	5.6
Slice LUTs	8816	21.5	8.7
Block RAMs	3	2.2	0.9
DSP48s	8	3.3	1.3