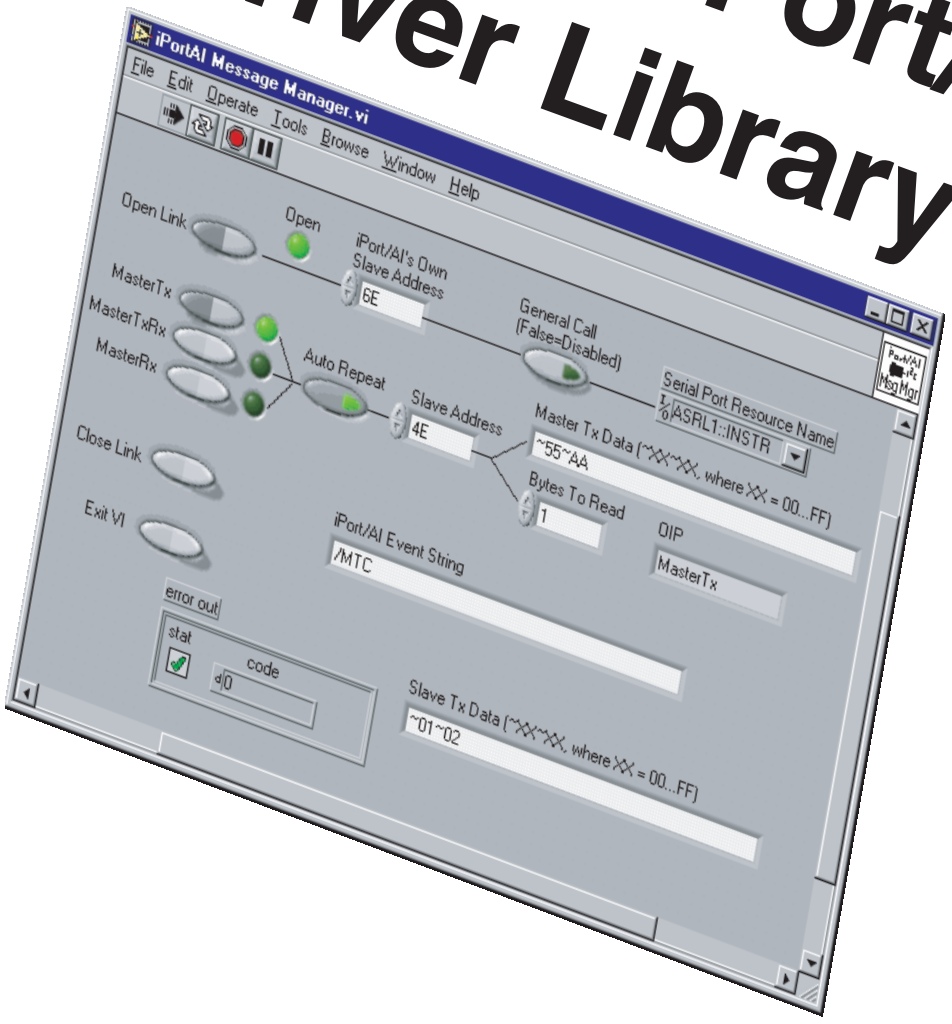
Programmer's Guide

LabVIEW iPort/AI Driver Library



MCC

www.mcc-us.com

DISCLAIMER: Micro Computer Control Corporation makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Micro Computer Control Corporation reserves the right to revise the product described in this publication and to make changes from time to time in the content hereof without the obligation to notify any person of such revisions or changes.

Life Support Applications

MCC Products are not designed for use in life support appliances, devices, or systems where the malfunction of a MCC Product can reasonably be expected to result in a personal injury.

WARNING: This equipment can radiate levels of radio frequency energy that may cause interference to communications equipment. Operation of this equipment may cause interference with radio, television, or other communications equipment. The user is responsible for correcting such interference at the expense of the user.

Printed in the United States of America

MCC products are licensed to use the I²C Bus.

Purchase of Philips I²C components conveys a license under the Philips I²C patent to use the components of the I²C system, provided the system conforms to the I²C specifications defined by Philips.

I²C is a trademark of Philips Corporation.

LabVIEW™ is a registered trademark of National Instruments

Table of Contents

Part 1

Part 2

Part 3

Part 4

Part 1

Introduction

Introduction

LabVIEW iPort/AI Driver Library

LabVIEW™ is the industry standard integrated development environment for test and control. MCC's LabVIEW iPort/AI Driver Library accelerates the development and deployment of systems incorporating I^2C Bus small area networks for configuration, testing, control, security, and monitoring activities.

The LabVIEW iPort/AI Driver Library is designed to assist LabVIEW developers in integrating I^2C Bus capabilities into test and control applications. The library includes a set of LabVIEW VIs (Virtual Instruments) for configuring and controlling the iPort/AI I^2C Bus Host Adapter as a bus master or slave device. Also included with the library are several LabVIEW introductory projects to help understand basic concepts, and two full-featured LabVIEW utility programs for communicating with I^2C Bus devices.

The LabVIEW drivers are implemented as a standard LabVIEW VIs, and by default are installed in the <LabVIEW>\instr.lib folder. This means that the iPort VIs appear on the Functions palette in LabVIEW, and can be easily added to an application's block diagram.

Part 2

System Requirements

System Requirements

Software

Windows 98/NT/2000+.

LabVIEW V6.0 or later.

Hardware

RS-232 Serial port available.

iPort/AI (#MIIC-202) RS-232 to I^2C Bus Host Adapter or iPort/AFM (#MIIC-203) RS-232 to I^2C Bus Host Adapter (**See Note**)

> **NOTE:** The this LabVIEW library will work with the iPort/AFM when operating in it's default mode.
> The iPort/AFM, when operating in default mode acts like an iPort/AI.
>
> See the iPort/AI (#MIIC-202) User's Guide for adapter installation instructions.

Software Installation

Insert the distribution disk in a drive. Click Start|Run|A:SETUP.EXE, where x is the drive with the distribution disk. Follow the instructions on screen.

The installation creates two folders on the install system, one for the iPort/AI VIs and one for the LabVIEW sample applications.

LabVIEW iPort VIs (Virtual Instruments)

The LabVIEW iPort/AI VIs provide a graphical interface for configuring and controlling of the MCC iPort/AI RS-232 to I^2C Bus Host Adapter. The VIs can be placed on a LabVIEW block diagram and wired up to other system components to add I^2C Bus capabilities to a LabVIEW application.

The LabVIEW iPort VI interface and sample LabVIEW application programs that use this interface can be found in later sections of this guide.

Part 3

Application Development

Application Development

This section provides a basic introduction to application development, and a few basic projects to get you started.

Overview

The MCC LabVIEW iPort/AI VIs provided icons and palette entries for the LabVIEW toolboxes. You can simply add a VI to your project by right-clicking on the VI's icon, placing the icon on your block diagram, and using the wiring tool to connect the icon with other application components.

A LabVIEW Quick How-To

This section provides a basic introduction to adding iPort/AI VIs to a LabVIEW block diagram, wiring them up, and sending an I^2C Bus Master Transmit message. This introductory application uses four iPort/AI VIs in performing an I^2C Bus Master Transmit operation. The steps to perform this operation include:

1. Open a communications link with the iPort/AI host adapter.
2. Set the I^2C Bus slave address for the destination device on the bus.
3. Master Transmit a message on the I^2C Bus to the slave device.
4. Close the iPort/AI communication link.

This simplified application operates in open-loop mode. It sends commands to the iPort/AI host adapter, but does not read or process iPort/AI responses as would normally be required in a more complete application. A more robust implementation is presented in subsequent LabVIEW applications included in later sections of this guide. The LabVIEW source for this and all the projects included with the iPort/AI Library are installed into the LabVIEW project folder during installation. You can access these projects from the LabVIEW Tools menu.

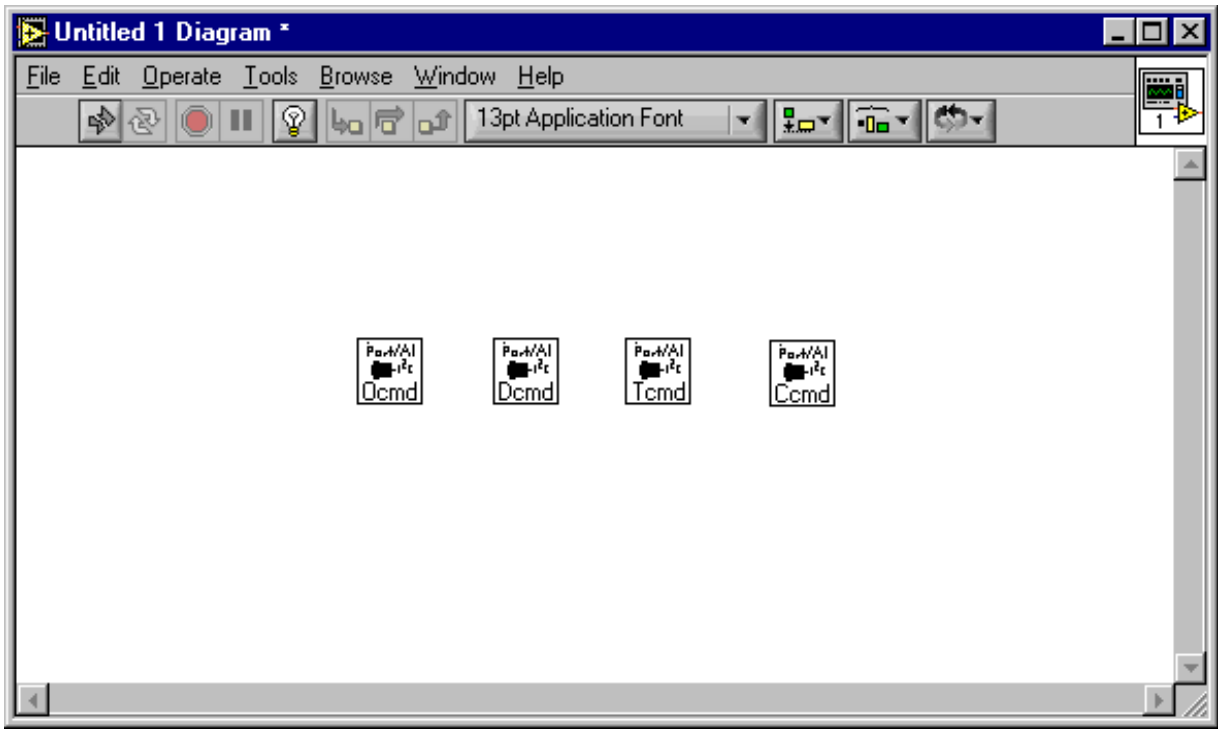
The following assumes you have installed the LabVIEW iPort/AI Library into the LabVIEW instr.lib folder on the host computer.

1. Start LabVIEW and create a new VI.

2. Click on the block diagram window and right click on the diagram to pop-up the LabVIEW Functions palette. From the LabVIEW Functions palette, choose Instrument I/O, Instrument Drivers, MCC iPort/AI I^2C Bus Host Adapter. Then left-click on the iPort/AI Open (Ocmd) VI to select it.

3. Place the Open VI icon on the block diagram. The iPort Open VI sets which RS-232 serial communication port is connected to the iPort host adapter, and other optional parameters. It also establishes a communication link with the adapter.

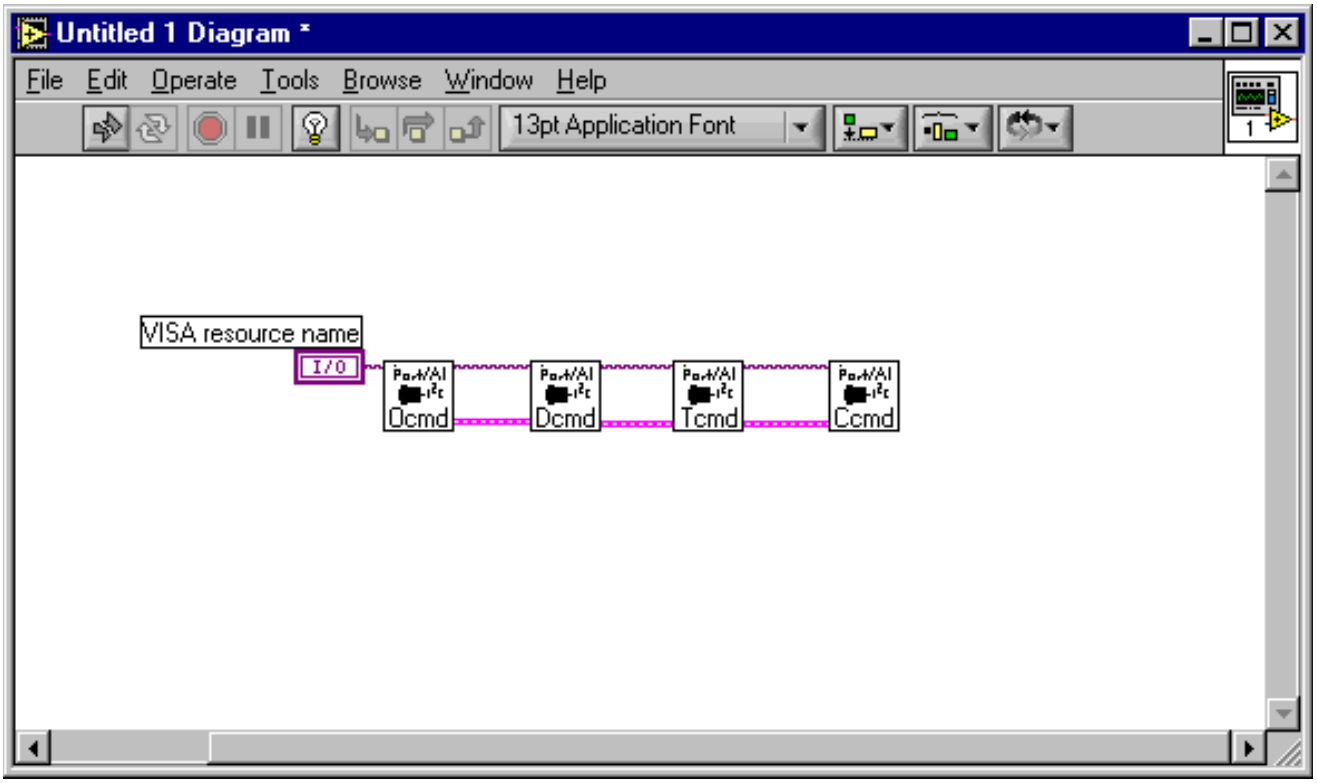
4. Repeat steps 2 and 3 for the iPort Set Destination Address (Dcmd), Master Transmit (Tcmd), and Close (Ccmd) VIs. When you are done, your block diagram should look like this:



5. All iPort VIs have built-in context help that shows VI input and output connector usage and VI capabilities. This context help can assist you in wiring the VIs to other block diagram components. To display the context help, click Help|Show Context Help on the LabVIEW menu bar, and hold the mouse cursor over the VI icon.

6. 
Now we need to start connecting block diagram components. Click on the wiring tool in the LabVIEW Tools palette. Then, using the wiring tool, right-click on the VISA Resource Name terminal on the Open VI to activate the pop-up menu. Select Create|Control to have LabVIEW add a VISA Resource Name control. This front panel control is used to specify the RS-232 serial port connected to the iPort host adapter.

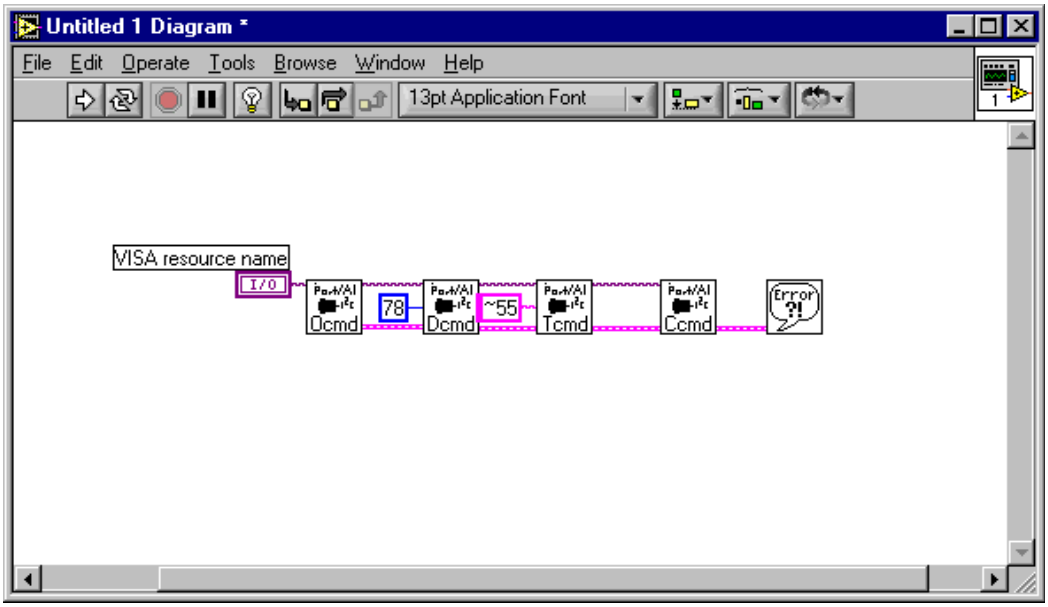
7. Again using the wiring tool, connect the Duplicate VISA Resource Name and Error output terminals on the Open VI to the VISA Resource Name and Error input terminals on the Set Destination Address VI. Do the same wiring for the Master Transmit and Close VIs. Now, your block diagram should look like this:



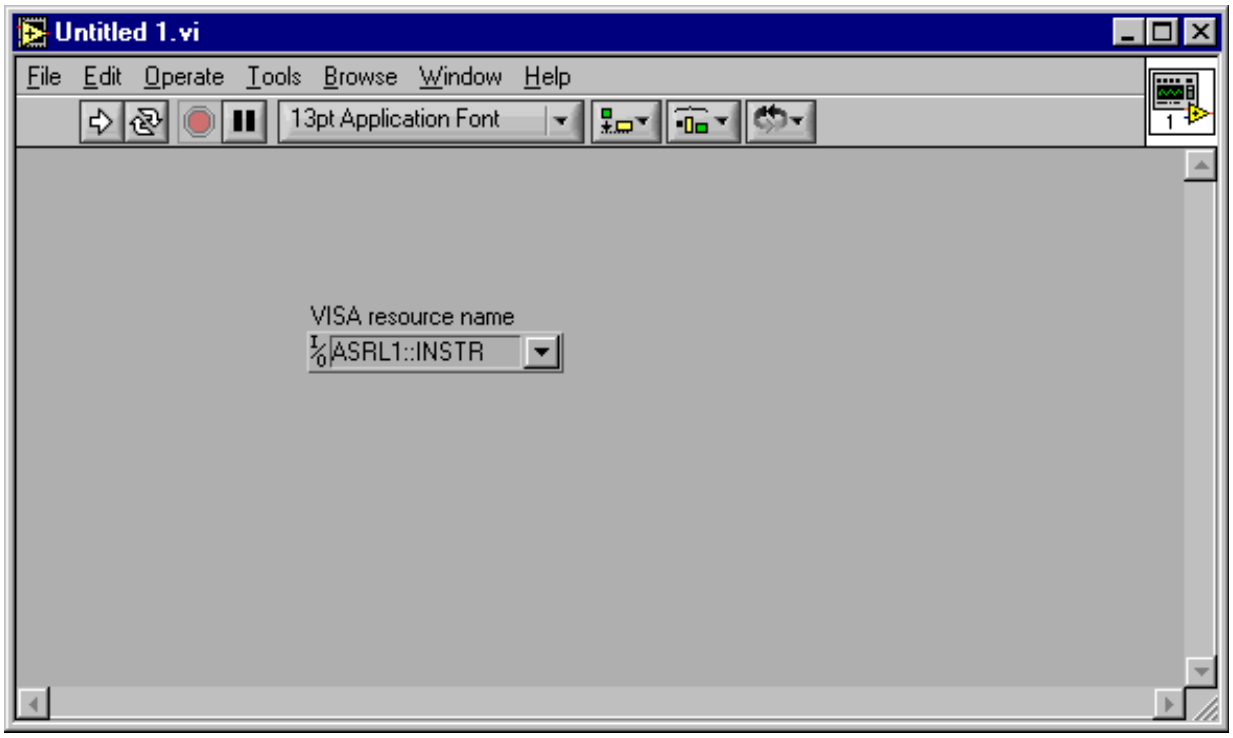
8. Now we need to set the destination slave address where we will send our I²C Bus message. To do this, right-click on the Set Destination Address VI Destination Slave Address terminal to activate the pop-up menu. Select Create|Constant to have LabVIEW add a constant parameter to the block diagram. The default value of the constant is “0". Type in the value 78. This is the default address (0x4E) of the 8-bit I/O device on the MCC I2C Bus Prototype Board (#IP-101). (Note: If you are not using the Prototype Board, you may want to enter the address of one of the I²C Bus devices on your target system.)

9. Now we need to setup the data for the Master Transmit message. To do this, right-click on the Master Transmit VI I²C Message String terminal to activate the pop-up menu. Select Create|Constant and type in “~55", a string we call a Hex-equivalent string that represents the single-byte hexadecimal value 0x55.

10. Now, to add a minimal amount of error handling, lets add an error dialog box to the block diagram to report any errors. This will help us solve problems that might occur even in this introductory project. Right-click on the block diagram to pop-up the Function pallet. Select Time&Dialog|Simple Error Handler.vi, and place its icon on the diagram to the right of the Close VI icon. Wire the Error Out terminal on the Close VI to the Error In terminal on the Error Handler VI. At this point your block diagram should look like this:



11. Now the only thing remaining to do is to tell LabVIEW which RS-232 serial communication port is connected to the iPort host adapter. To do this, click on the front panel and select the Edit Text tool in the LabVIEW Tools pallet. Use the Edit Text tool to enter “ASRLx::INSTR” in the VISA Resource Name control, where “x” is “1" for Port 1, “2" for Port 2, an so on. Save the project. When you are done, the front panel should look like this:



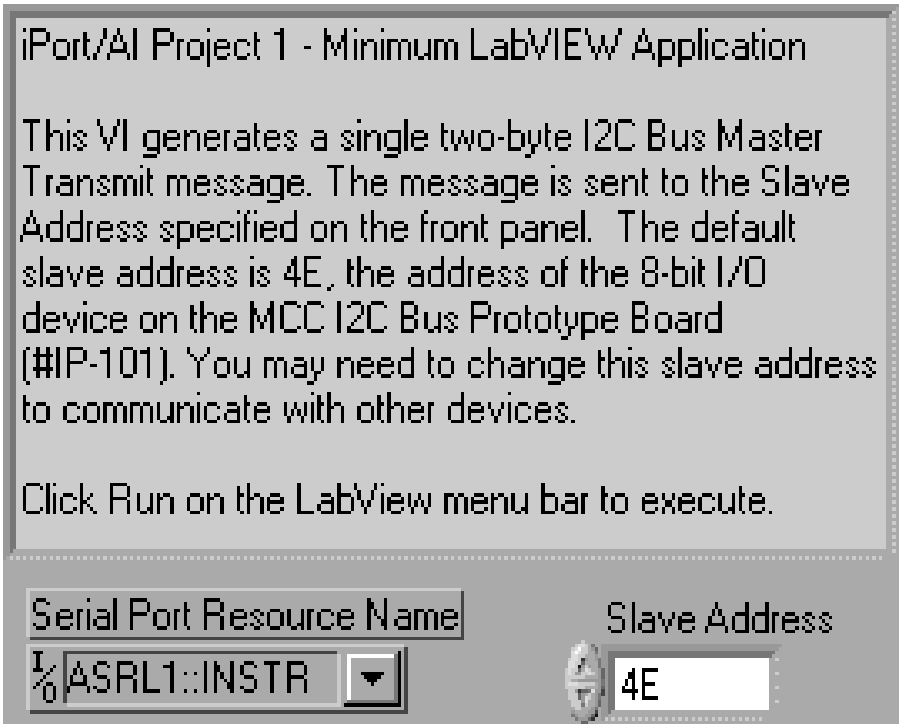
12. Now its time to test our new VI. To do this you will need to have connected the iPort host adapter to the RS-232 serial port, and connected the adapter to a target system containing the I^2C Bus slave device that will receive our Master Transmit message. This target system can be one of your own design, an existing device, or one you purchase. To help in this area, MCC offers its I^2C Bus Prototype Board (#IP-101) that contains two I^2C Bus slave devices, an 8-bit I/O device, and a memory device. To test the VI, click on the block diagram, and turn on Highlight Execution on the LabVIEW button bar. Then, click the Run button on the LabVIEW button bar. If all goes well, you should be able to follow execution through the VI, send the Master Transmit message to the selected slave device, and have the VI terminate without displaying the Error Dialog.

More Example Programs

NOTE: The sample project VIs included with the iPort VI Library communicate with the default I^2C Bus slave address 0x4E. This is the default address of the 8-bit I/O device on the MCC I^2C Bus Prototype Board (#IP-101). You may need to change this slave address to communicate with devices on your target system.

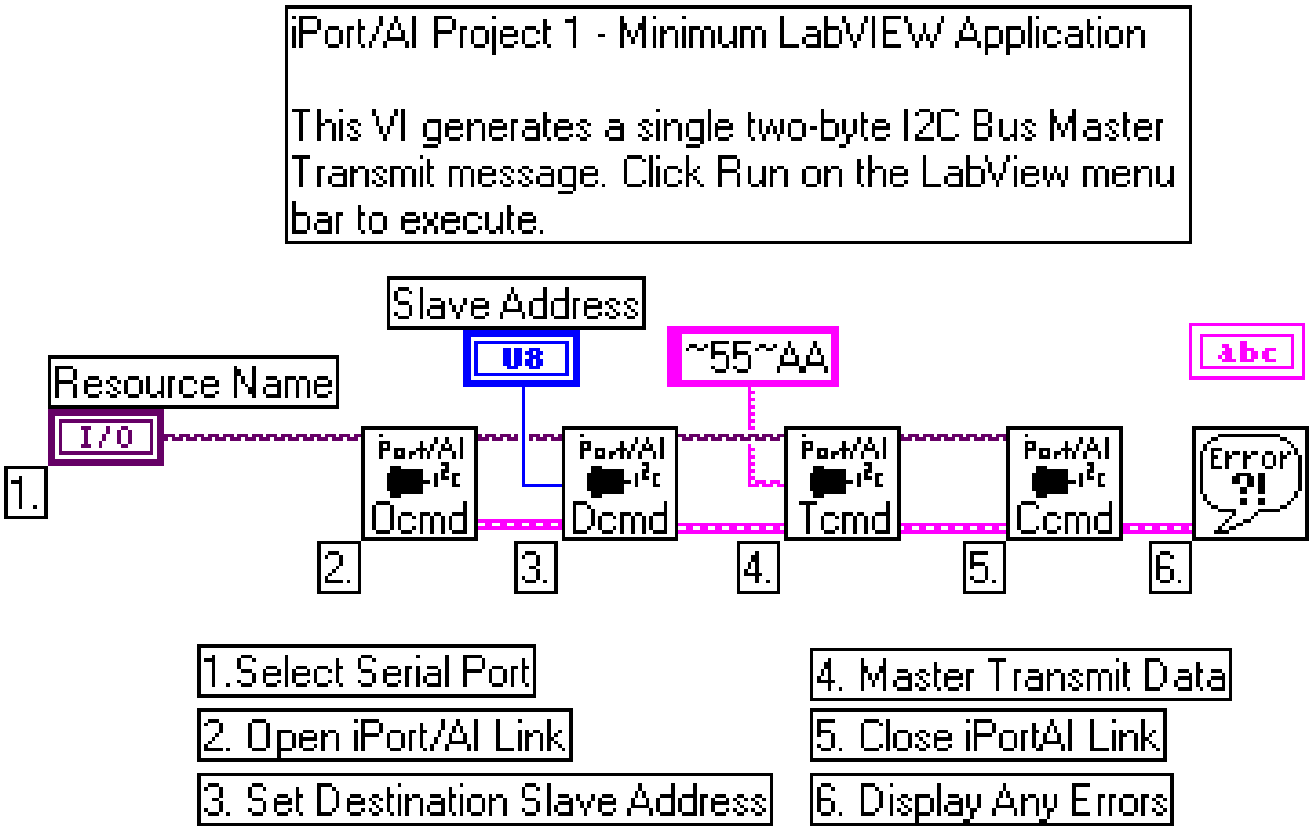
Project 1, A Minimum LabVIEW Application

Project 1, like the previous quick example, also operates in open-loop mode. But here we have added the ability to change the I^2C Bus destination slave address to the front panel.



Front Panel

This VI generates a single two-byte I²C Bus Master Transmit message. The message is sent to the Slave Address specified on the front panel. The default slave address is 0x4E, the address of the 8-bit I/O device on the MCC I²C Bus Prototype Board (#IP-101). You may need to change this slave address to communicate with other devices.

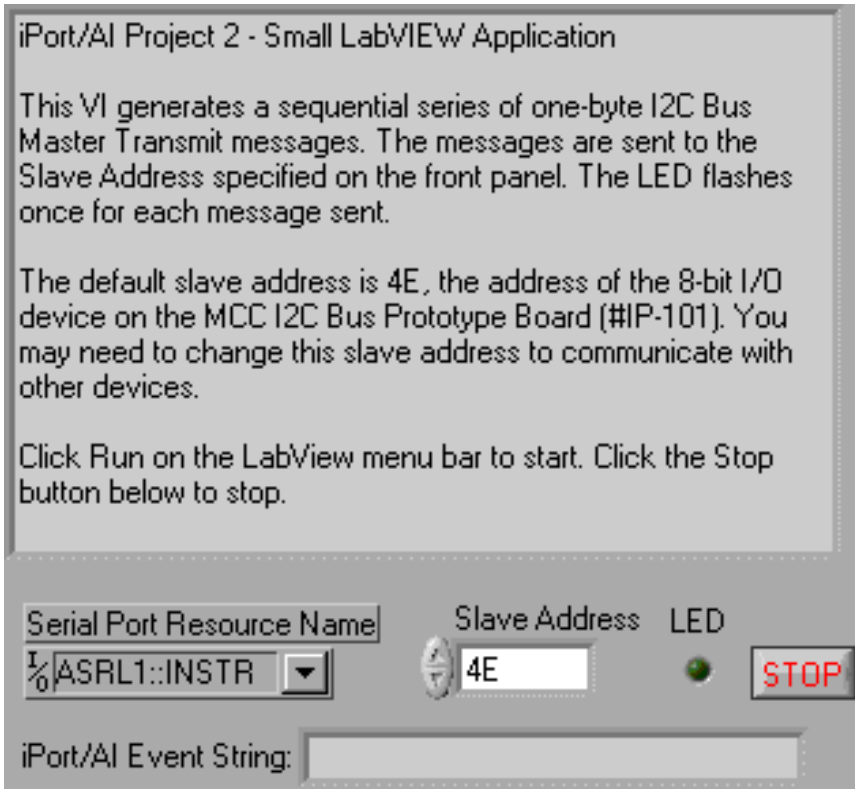


Project 2, A Small LabVIEW Application

Unlike the previous example, this example operates in closed-loop mode. Here, once the Master Transmit command is sent to the iPort/AI, the application enters an Event Loop that monitors and processes responses from the iPort/AI Adapter.

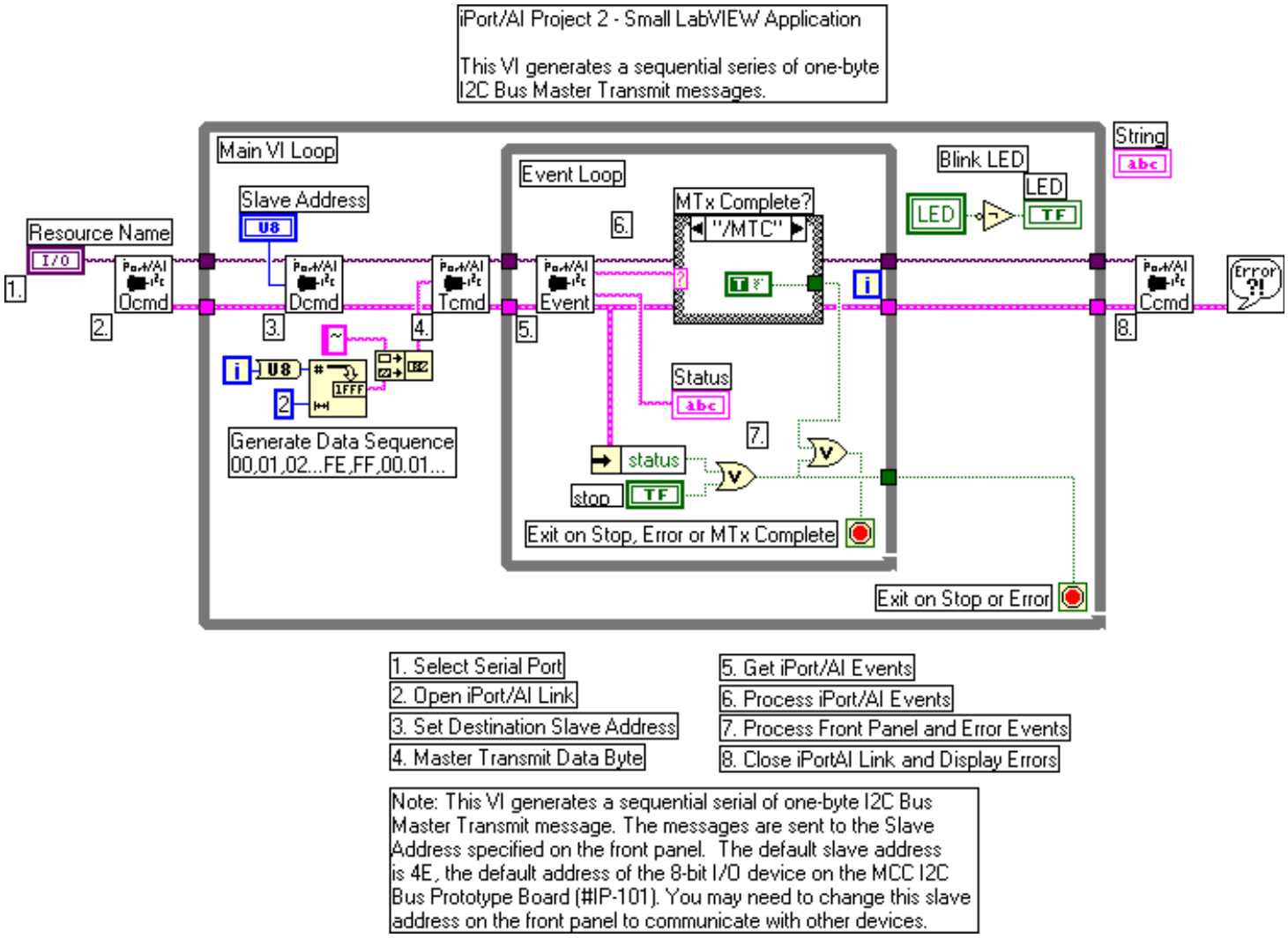
The Event Loop calls the iPort/AI Event Handler VI. This VI monitors the RS-232 communications port for messages from the iPort/AI host adapter, and returns an Event Code string that can be used to control a LabVIEW case structure, and an Event String that includes the Event Code string plus any data received by the host adapter.

The Event Loop also includes a case structure. This case structure has a diagram for each iPort/AI Event Code string, and takes appropriate actions when Event Codes are received.



Front Panel

This VI generates a continuous series of one-byte I²C Bus Master Transmit messages. The messages are sent to the Slave Address specified on the front panel. The LED flashes once for each message sent. The default slave address is 0x4E, the address of the 8-bit I/O device on the MCC I²C Bus Prototype Board (#IP-101). You may need to change this slave address to communicate with other devices.

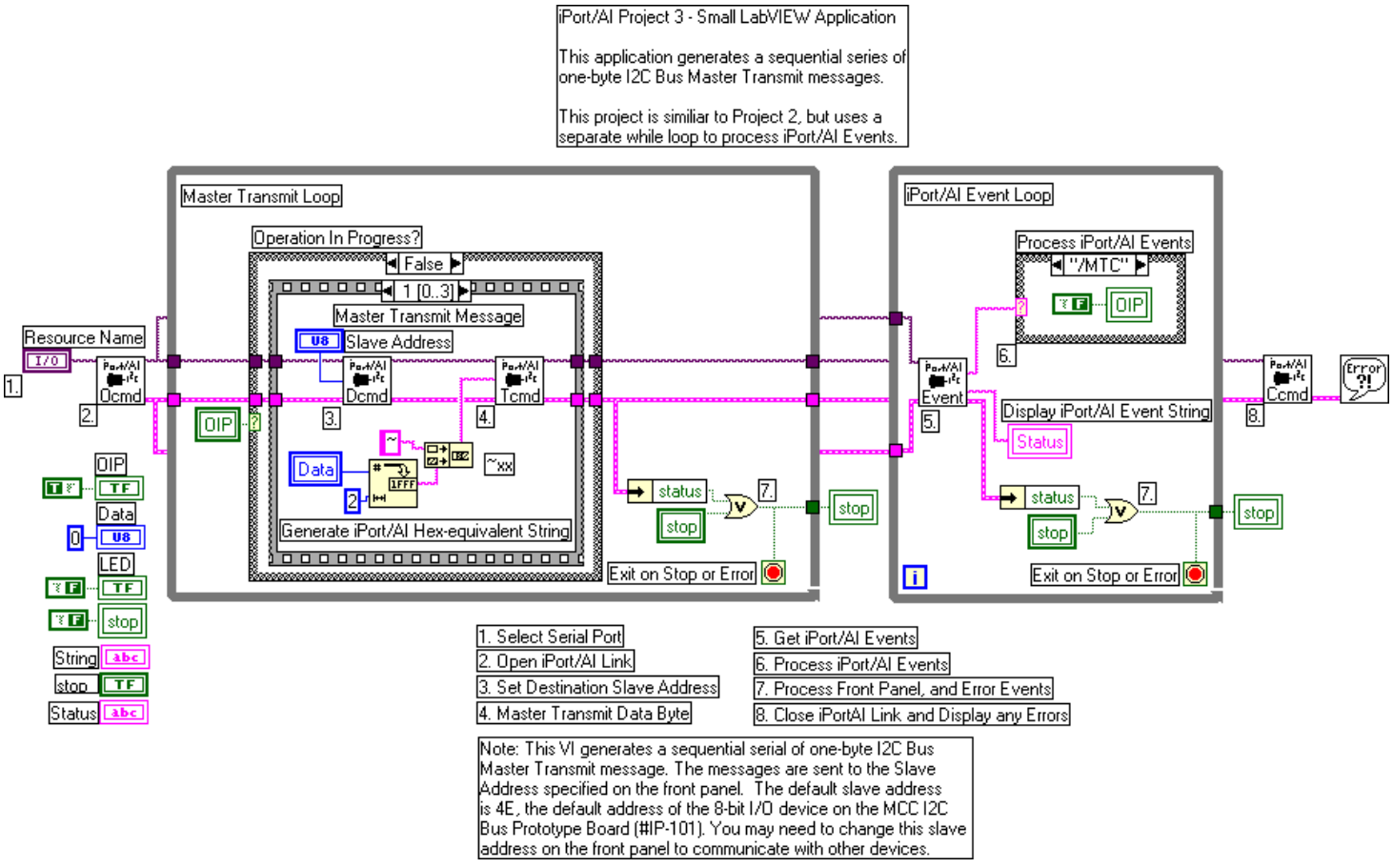


Project 3, A Different Way

This project is similar to Project 2 above, but uses separate while loops to process Master Transmit and iPort/AI response events.

Front Panel

This VI generates a continuous series of one-byte I^{2}C Bus Master Transmit messages. The messages are sent to the Slave Address specified on the front panel. The LED flashes once for each message sent. This project is similar to Project 2, but it uses a separate parallel loop to process iPort/AI events. The default slave address is 0x4E, the address of the 8-bit I/O device on the MCC I^{2}C Bus Prototype Board (#IP-101). You may need to change this slave address to communicate with other devices.



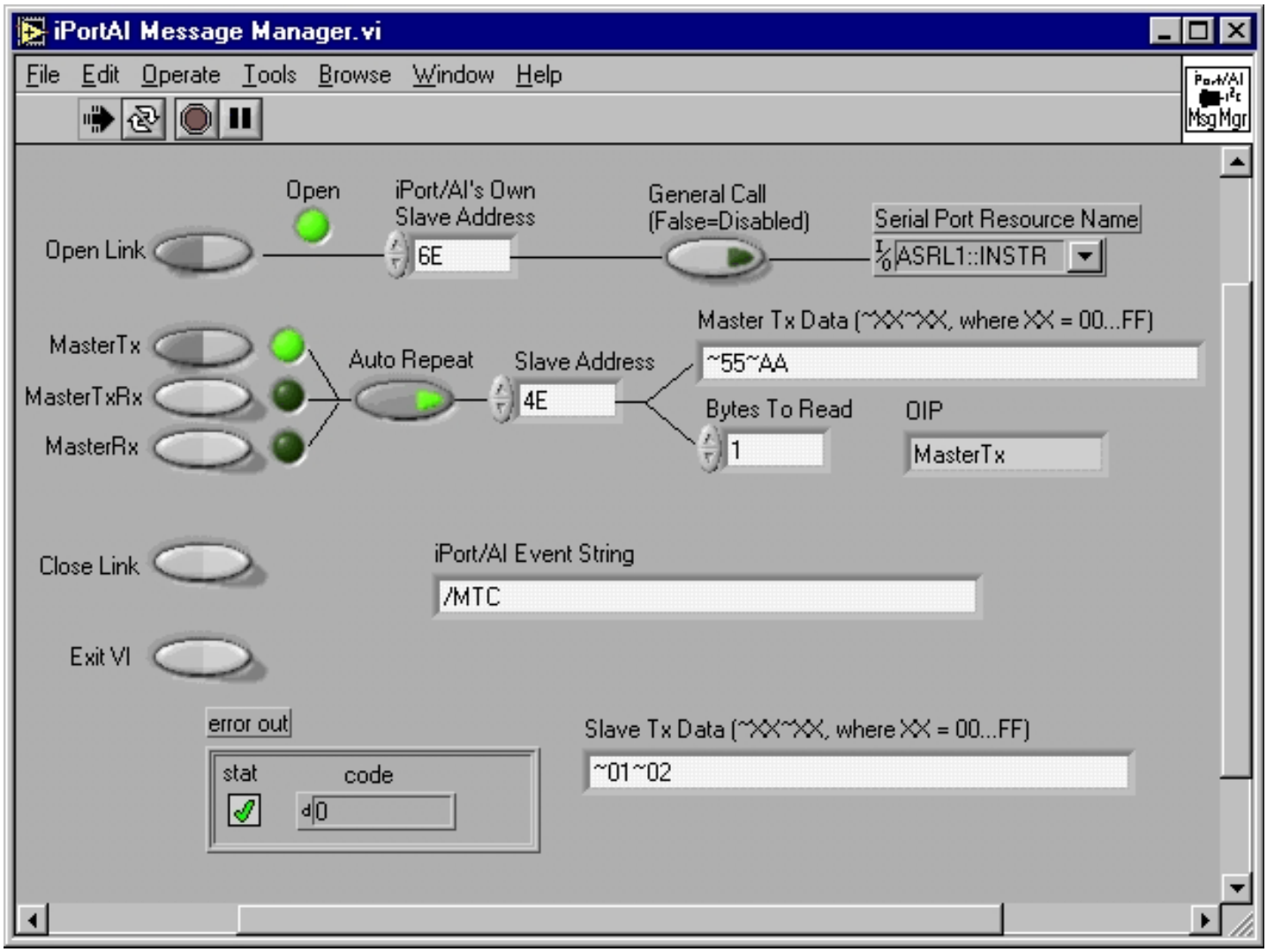
LabVIEW Message Manager Project

iPort/AI Message Manager VI is a full-featured LabVIEW VI that supports all four I^2C Bus message modes, including:

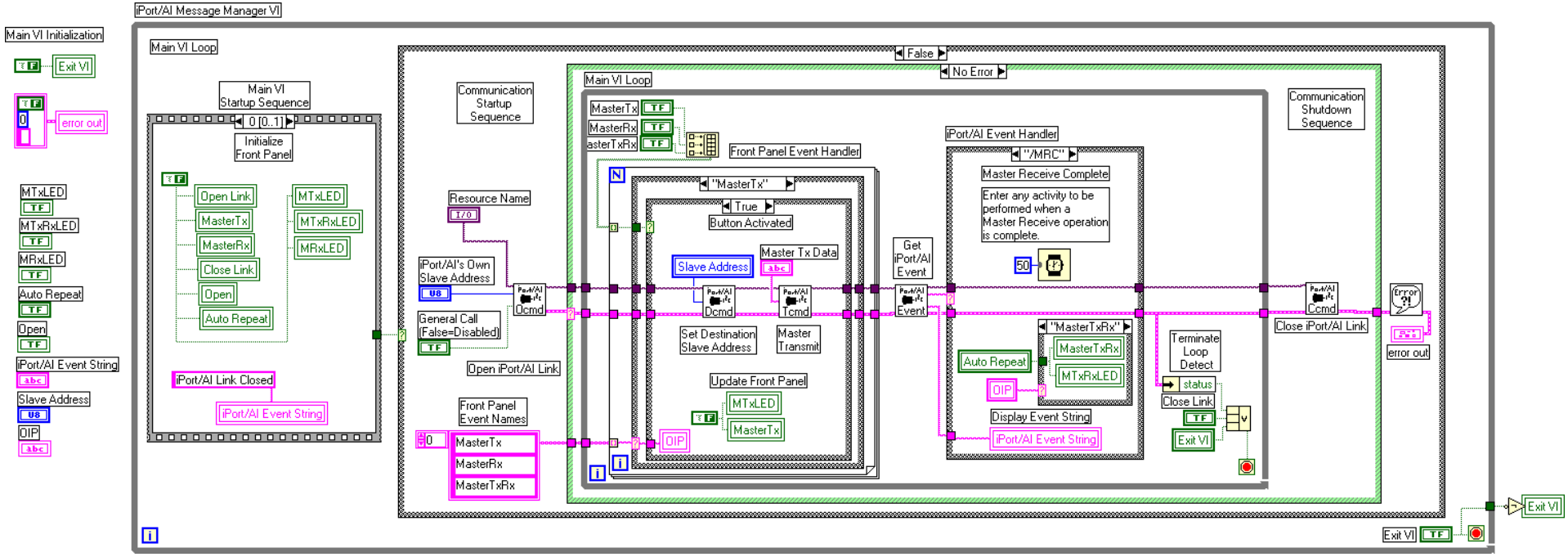
1. Master Transmit
2. Master Receive
3. Slave Transmit
4. Slave Receive

Also support are other I^2C Bus features, including:

1. General Call
2. Master Transmit/Receive



Message Manager Front Panel



Message Manager Block Diagram

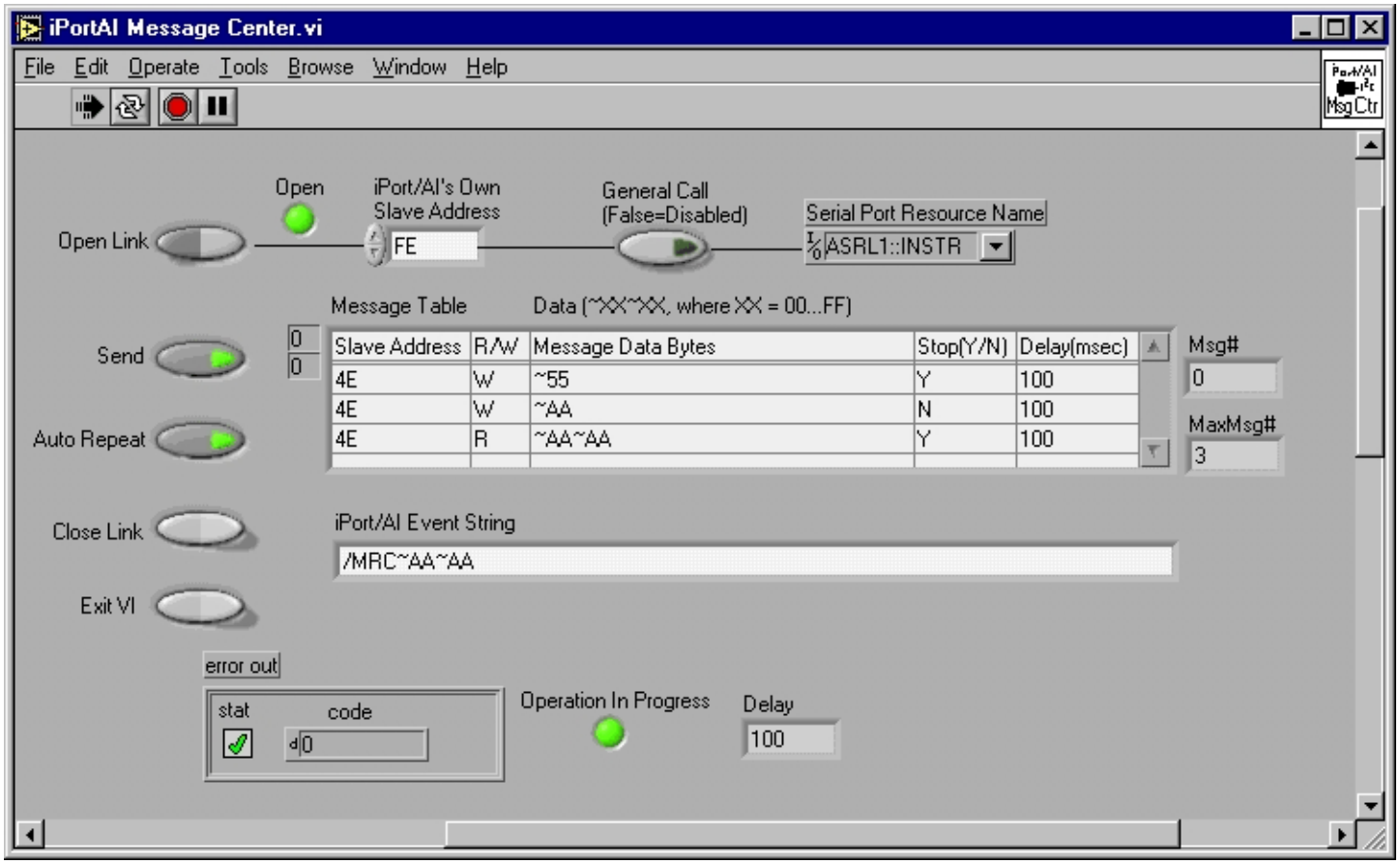
As shown in it's block diagram, the Message Manager VI first executes a start-up sequence, then enters its main loop. The main loop consists of a Front Panel Handler, and an iPort/AI Event Handler.

The Front Panel Handler detects and processes operator interactions with the front panel.

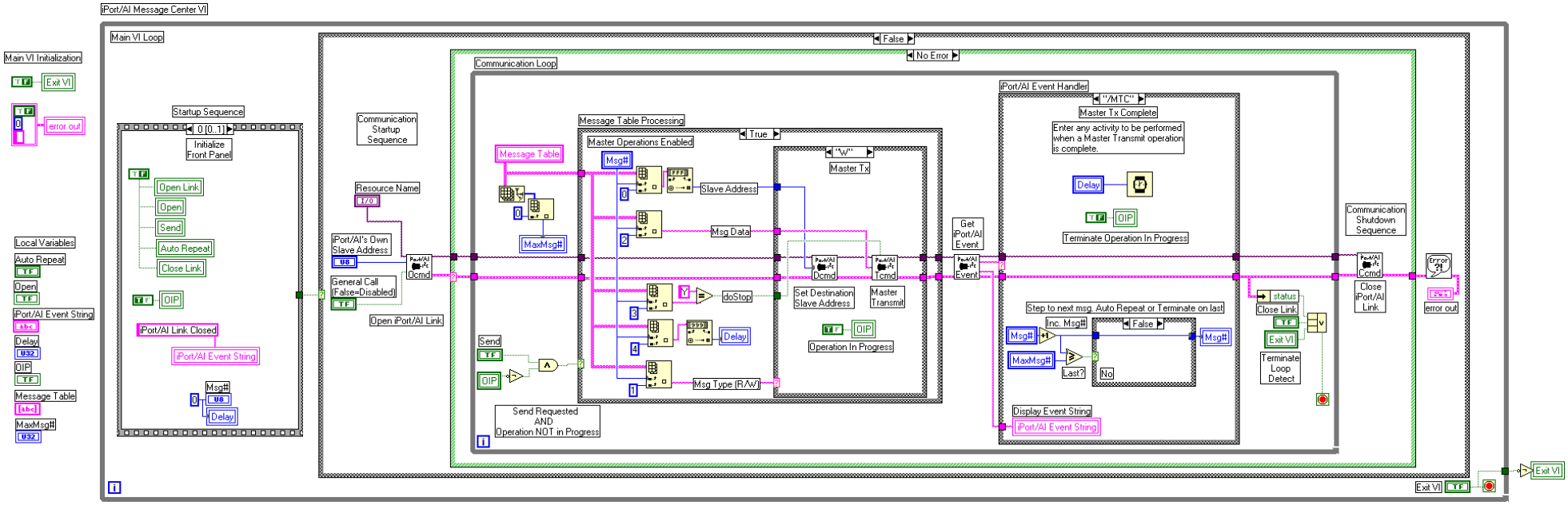
The iPort/AI Event Handler detects and processes responses from the iPort/AI host adapter.

LabVIEW Message Center Project

iPort/AI Message Center VI is a full-featured LabVIEW VI that supports I^2C Bus Master Transmit and Receive operations. One or more messages can be entered into a spreadsheet front panel control, indicating the slave address, message direction (Read or Write), message data, message stop control, and an optional time delay. Messages in the spreadsheet may be sent once, or repeated.



Message Center Front Panel



Message Center Block Diagram

The Message Center VI is similar to the previous Message Manager VI, except the Front-Panel Handler has been replaced with a Message Table Processor.

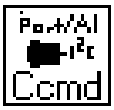
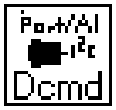
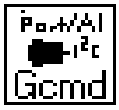
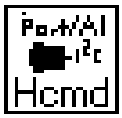
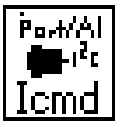
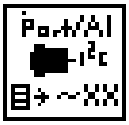
The Message Table Processor handles operator interactions with the front panel, and sequences through a series of messages to be sent across the I^2C Bus.

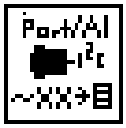
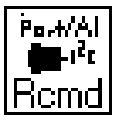
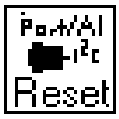
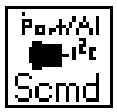
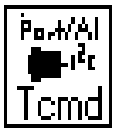
Part 4

LabVIEW iPort/AI Library Reference

This section provides a description of the LabVIEW iPort/AI library command VI's.

LabVIEW iPort/AI Library VI's

Command	Description
	**Close I²C Connection (Ccmd) VI** Causes the adapter to disconnect from the I²C Bus.
	**Set Destination Slave Address (Dcmd) VI** Sets the I²C Bus destination slave address for subsequent Master Transmit or Master Receive messages sent by the adapter.
	**General Call Enable (Gcmd) VI** Enables or disables the adapter's addressed slave response to the General Call (0x00) address.
	**Hex Only Display (Hcmd) VI** Controls how the adapter will send master or slave receive message data to its host computer.
	**Set Own Slave Address (Icmd) VI** Sets the adapter's own slave address.
	**Byte Array to Hex-equivalent String.vi** Convert a single-dimension byte array into a compatible Hex-equivalent string.

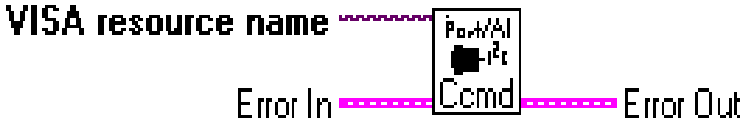
	**EventHandler V01.10 VI** Monitors the host adapter serial link for responses.
	**String To Byte Array.vi** Convert a Hex-equivalent string into a one-dimensional byte array.
	**Open Connection (Ocmd) VI** Establishes the link between the host computer and the adapter.
	**Master Read Message (Rcmd) VI** Read the specified number of data bytes from the currently selected Destination slave address.
	**Reset VI** Causes the adapter to re-boot and revert to its default state.
	**Slave Transmit Message (Scmd) VI** Causes the adapter to write specific data bytes to the requesting I^2C Bus Master Receiver device.
	**Master Transmit Message (Tcmd) VI** Write the specified data bytes to the currently selected Destination slave address.

LabVIEW iPort/AI Library Reference

Close I²C Connection (Ccmd) VI

The I²C Connection (Ccmd) VI sends a /C command to the host adapter and closes the RS-232 serial communication link. A /C command causes the adapter to disconnect from the I²C Bus. The adapter's normal response to the /C command is "/CCC", Close Connection Complete.

Connector Pane

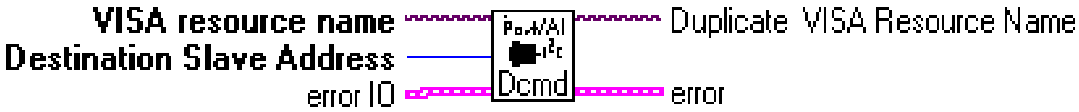


	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
	**Error In** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
	**Error Out** contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Set Destination Slave Address (Dcmd) VI

The Set Destination Slave Address (Dcmd) VI sends a /D command to the iPort/AI host adapter. The /D command sets the I^2C Bus destination slave address for subsequent Master Transmit or Master Receive messages sent by the iPort/AI. The /D command accepts even numbered slave addresses in the range of 0x00 to 0xFE. The selected slave address remains in effect until changed by another /D command, or the adapter is reset. The default slave address is 0x00. The adapter's normal response to the /D command is "*", which is translated by the adapter's Event VI to "/RDY".

Connector Pane



I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
U8	**Destination Slave Address** specifies the slave address for subsequent Master Transmit or Master Receive messages sent by the adapter. Slave addresses may be even numbered in the range of 0x00 to 0xFE.
	error IO describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
I/O	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	error out contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

General Call Enable (Gcmd) VI

The Call Enable (Gcmd) VI sends a /G command to the host adapter. The /G command enables or disables the adapter's addressed slave response to the general call (0x00) address. The adapter's normal response is "*", which is translated by the Event VI to "/RDY".

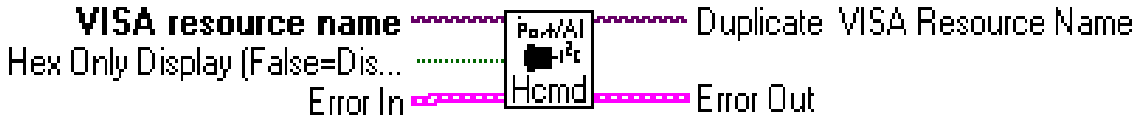
Connector Pane

I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
TF	**General Call** enable specifies that the adapter will respond to the I^2C Bus General Call address (0x00).
	Error In describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
I/O	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	Error Out contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces

Hex Only Display (Hcmd) VI

The Hex Only Display (Hcmd) VI sends a /H command to the host adapter. The /H command controls how the adapter will send master or slave receive message data to its host computer. When disabled, received I^2C Bus message data bytes representing ASCII printable characters are sent as their ASCII printable character. Non-ASCII printable data bytes are always sent in Hex (~00...~FF) form. When Hex Only Display enabled, all received message data is sent in Hex (~00...~FF) form. The adapter's normal response is "*", which is translated by the Event VI to "/RDY".

Connector Pane

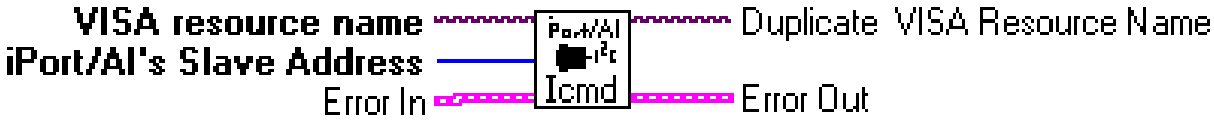


I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
TF	**Hex Only Display** controls the output of received message data in ASCII printable and Hex, or Hex only format.
	Error In describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
I/O	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	Error Out contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Set iPort/AI's Own Slave Address (Icmd) VI

The Set Own Slave Address (Icmd) VI sends a /I command to the host adapter. The /I command sets the adapter's own slave address to the specified Slave Address. The adapter responds to slave transmit and slave receive messages sent to this address. The adapter's normal response to the /I command is "*", which is translated by the Event VI to "/RDY".

Connector Pane



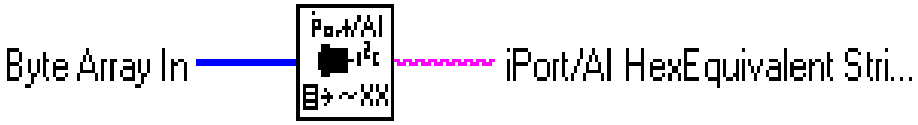
I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
U8	**iPort/AI's Slave Address** specifies the slave address the adapter will respond to in slave transmit and receive operations. The Slave Addresses can be an even numbered in the range of 0x02 to 0xFE.
	Error In describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
I/O	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	Error Out contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Byte Array to Hex-equivalent String.vi

The Byte Array to Hex-equivalent String VI is a helper VI. It can be used to convert a single-dimension byte array into a compatible Hex-equivalent string. Hex-equivalent strings are used by Master Transmit, Master TxRx, and Slave Transmit VIs to represent one or more bytes (U8) of any value (00...FF hexadecimal or 0...255 decimal) and still maintain the ASCII character interface with the adapter.

A Hex-equivalent string contains sets of three ASCII characters for each data byte. This three ASCII character set uses the form "~xx", where the "~" character is a Hex-equivalent marker, and the "xx" represents a string of two ASCII-Hexadecimal characters that represent an 8-bit unsigned number in the range of 00 to FF hexadecimal.

Connector Pane



U8	**Byte Array In** a single-dimension byte array to be converted to an adapter compatible Hex-equivalent string.
abc	**iPort/AI HexEquivalent String Out** contains a compatible Hex-equivalent string. Hex-equivalent strings are used by Master Transmit, Master TxRx, and Slave Transmit VIs to represent one or more bytes (U8) of any value (00...FF hexadecimal or 0...255 decimal) and still maintain the ASCII character interface with the adapter. A Hex-equivalent string contains sets of three ASCII characters for each data byte. This three ASCII character set uses the form "~xx", where the "~" character is a Hex-equivalent marker, and the "xx" represents a string of two ASCII-Hexadecimal characters that represent an 8-bit unsigned number in the range of 00 to FF hexadecimal.

EventHandler V01.10 VI

The Event Handler VI monitors the host adapter serial link for responses. With each Event Handler call, any available receive characters are assembled into a string until the input stream is exhausted, or a terminating character (CR, or *) is received and a minimum (4 character) response is present. The Event String Out may contain:

a. The complete response string received from the adapter.

b. An "/NDA" if no input stream data is available.

c. A null string ("") if no complete response is present.

d. An Event Handler error string .

The Event Code Out contains the first 4 characters on the Event String, and can be used as input to a case structure to identify the specific adapter event.

Connector Pane

	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
	**Error In** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.

	**Event Code Out** is a null string, or a 4 character string representing an handler event. An "/NDA" indicates no data is available in the serial port input stream. A null string indicates the handler is processing a response, but the response is not complete. Other strings include event handler errors, adapter response codes, or "/RDY" which the event handler generates when it receives "*" ready response. This string is suitable for input to a LabVIEW case structure.
	**Event String Out** contains the complete adapter response string. For a Master Read operation, this string contains "/MRCxxx". For a Slave Receive operation, this string contains "/SRCxxx" standard slave addressing, or "/GRCxxx" for general call slave addressing. In all three cases, "xxx" represents printable ASCII or Hex-equivalent (~00...~FF) data bytes.
	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.

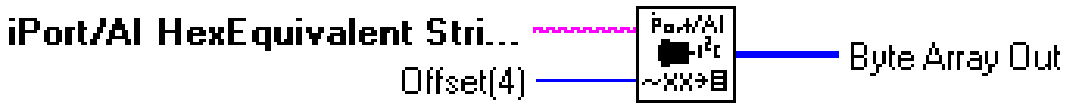
Hex-equivalent String To Byte Array.vi

The Hex-equivalent String to Byte Array VI is a helper VI. It can be used to convert a Hex-equivalent string into a one-dimensional byte array ready for additional processing. Hex-equivalent strings are return by the Event Handler VI in response to Master Receive, Master TxRx, and Slave Receive messages. A Hex-equivalent string represent one or more bytes (U8) of any value (00...FF hexadecimal or 0...255 decimal) and still maintain the ASCII character interface with the adapter.

To assist in processing the Event String Out terminal on the Event Handler VI, the Hex-equivalent String to Byte Array VI provides a default offset of 4 to step past the leading four character Event Code present in the Event String Out.

A Hex-equivalent string contains sets of three ASCII characters for each data byte. This three ASCII character set uses the form "~xx", where the "~" character is a Hex-equivalent marker, and the "xx" represents a string of two ASCII-Hexadecimal characters that represent an 8-bit unsigned number in the range of 00 to FF hexadecimal.

Connector Pane



U8	**Offset (4)** to the first Hex-equivalent string set. The default value (4) steps past the leading four character Event Code present in the Event String Out.
abc	**HexEquivalent String Out** contains a compatible Hex-equivalent string to be converted to a single-dimension byte array. Hex-equivalent strings are used in the Event String Out connector of Master Receive Complete (/MRC), Slave Receive Complete (/SRC) and General Call Receive (/GRC) Events to represent one or more bytes (U8) of any value (00...FF hexadecimal or 0...255 decimal) and still maintain the ASCII character interface with the adapter. A Hex-equivalent string contains sets of three ASCII characters for each

	data byte. This three ASCII character set uses the form "~xx", where the "~" character is a Hex-equivalent marker, and the "xx" represents a string of two ASCII-Hexadecimal characters that represent an 8-bit unsigned number in the range of 00 to FF hexadecimal.
	**Byte Array In** a single-dimension byte array representing the data in a compatible Hex-equivalent string.

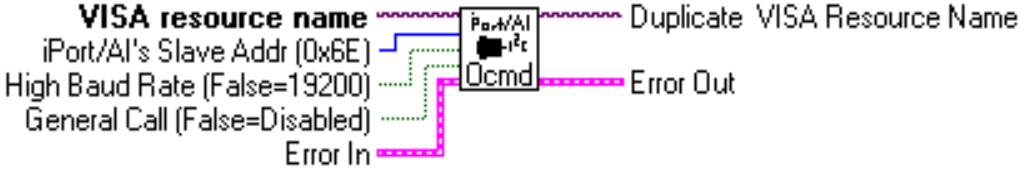
Open Connection (Ocmd) VI

The Open Connection (Ocmd) VI establishes the link between the host computer and the adapter. The following steps are performed:

1. Sets serial port communication parameters.
2. Issues an adapter Reset command.
3. Initializes the adapter's own slave address and general call (address 0x00) enable.
4. Sends a /O command to the host adapter.

The /O command activates the adapter as an active master or slave device on the I^2C Bus. The normal response to the /O command is "*", which is translated by the Event VI to "/RDY".

Connector Pane



I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
U8	**iPort/AI's Slave Address** specifies the slave address the adapter will respond to in slave transmit and receive operations. The Slave Addresses can be an even numbered in the range of 0x02 to 0xFE.
TF	**High Baud Rate** enables 115,000 baud communication. Available only for adapters supporting this rate.
TF	**General Call** enable specifies that the adapter will respond to the I^2C Bus General Call address (0x00).
	Error In describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error

	out. The error in cluster contains the following parameters.
	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	**Error Out** contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Master Read Message (Rcmd) VI

The Master Read Message (Rcmd) VI sends a /R command to the host adapter. The /R command causes the adapter to read the specified number of data bytes from the currently selected Destination slave address, with or without generating a message terminating I^2C Bus Stop.

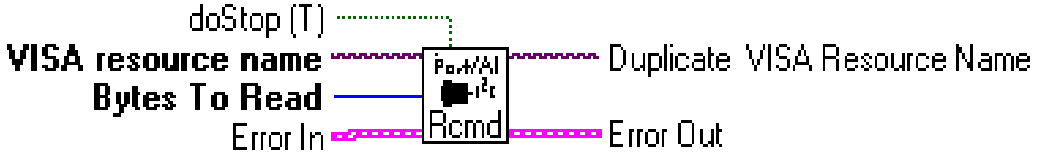
The Destination slave address is the address specified by the last Set Destination Slave Address command received by the adapter.

The specified number on bytes to read can be in the range of 0 to 32767, with a byte count of zero indicating a Variable Length message read, where the first byte received from the slave device indicates the number of additional trailing bytes to read. The adapter automatically reads the first byte and all additional bytes from the slave.

The normal response to the /R command is "/MRC" followed by the data read from the slave. All message bytes, including the Length byte are included in the response. Other possible responses are possible depending on the adapter status and I^2C Bus activity. See the Event VI, and the adapter's User's Guide, for a complete list of responses.

Received text is a representation of the data of the data bytes within the Master Receive message. The format of this data is controlled by the current value of the Hex Only Display setting.

Connector Pane



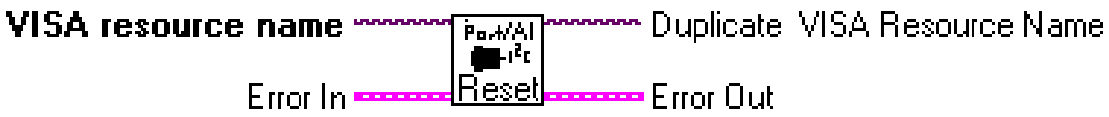
I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
U16	**Bytes To Read** specifies the number on bytes to read. This can be in the range of 0 to 32767, with a byte count of zero indicating a Variable Length message read, where the first byte received from the slave device indicates the number of additional trailing bytes to read. The adapter

	automatically reads the first byte and all additional bytes from the slave.
	**doStop** specifies if the adapter should generate a message terminating I^2C Bus Stop.
	**Error In** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	**Error Out** contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Reset VI

The VI sends a reset sequence (Ctrl/R, Ctrl/R, Ctrl/R) to the host adapter. A reset sequence causes the adapter to re-boot and revert to its default state. The normal response is "*", which is translated by the Event VI to "/RDY".

Connector Pane



	VISA resource name specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
	Error In describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
	Duplicate VISA Resource Name is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	Error Out contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Slave Transmit Message (Scmd) VI

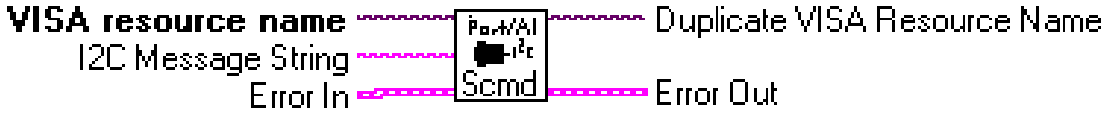
The Slave Transmit Message (Scmd) VI sends a /S command to the host adapter. The /S command should be issued to the adapter in response to a Slave Transmit Request "/STR". The /S command causes the adapter to write the specified data bytes to the requesting I²C Bus Master Receiver device.

NOTE: Upon receiving a Slave Transmit request from a Master Receiver device on the I²C Bus, the adapter outputs a Slave Transmit Request "/STR" to its host computer, and initiates an I²C Bus Clock Stretch (SCL line Low) until a /S command is received from the host. While clock stretching, no other messages can be transmitted on the I²C Bus.

The Slave Transmit Message VI accepts one or more printable ASCII or Hex-equivalent (~00...~FF) data bytes. The special characters tilde (~) and Carriage Return (CR) have special meaning to the adapter and must be sent in their Hex-equivalent form (~ = ~7E, CR = ~0D).

The normal response to the /S command is Slave Transmit Complete "/STC", although other responses are possible depending on adapter's status and I²C Bus activity. See the Event VI, and the adapter's User's Guide, for a complete list of responses.

Connector Pane



I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
abc	**I²C Message String** specifies the message data the adapter transmits to the requesting I²C Bus Master Receiver device. The Slave Transmit

	Message VI accepts one or more printable ASCII or Hex-equivalent (~00...~FF) data bytes. The special characters tilde (~) and Carriage Return (CR) have special meaning to the adapter and must be sent in their Hex-equivalent form (~ = ~7E, CR = ~0D). Example: To send the three bytes 00,01, and 02, the message string should be "~00~01~02".
	**Error In** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	**Error Out** contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Master Transmit Message (Tcmd) VI

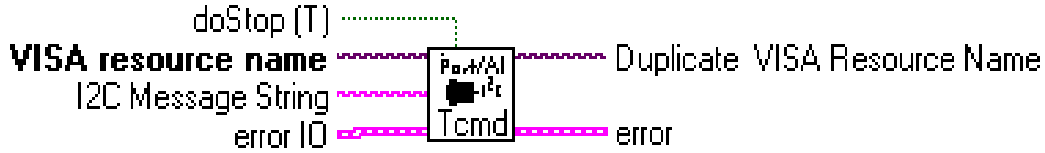
The iPort/AI Master Transmit Message (Tcmd) VI sends a /T command to the iPort/AI host adapter. The /T command causes the iPort/AI to write the specified data bytes to the currently selected Destination slave address, with or without generating a message terminating I^2C Bus Stop.

The Destination slave address is the address specified by the last Set Destination Slave Address command received by the iPort/AI.

The Master Transmit Message VI accepts zero or more printable ASCII or Hex-equivalent (~00...~FF) data bytes. The special characters tilde (~) and Carriage Return (CR) have special meaning to the iPort/AI and must be sent in their Hex-equivalent form (~ = ~7E, CR = ~0D).

The iPort/AI normal response to the /T command is "/MTC", although other responses are possible depending on iPort/AI status and I^2C Bus activity. See the iPort/AI Event VI, and the iPort/AI User's Guide, for a complete list of responses.

Connector Pane



I/O	**VISA resource name** specifies the resource that will be opened. This control also specifies the session and class. Refer to VISA Resource Name Control for more information.
abc	**I^2C Message String** specifies the message data the iPort/AI transmit to the currently selected slave address. The Master Transmit Message VI accepts zero or more printable ASCII or Hex-equivalent (~00...~FF) data bytes. The special characters tilde (~) and Carriage Return (CR) have special meaning to the iPort/AI and must be sent in their Hex-equivalent form (~ = ~7E, CR = ~0D). Example: To send the three bytes 00,01, and 02, the message string should be

	"~00~01~02".
	**doStop** specifies if the iPort/AI should generate a message terminating I^2C Bus Stop.
	**Error In** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters.
	**Duplicate VISA Resource Name** is a copy of the VISA resource name that is passed out of the VISA functions. Refer to VISA Resource Name Control for more information.
	**Error Out** contains error information. If error in indicates an error, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Micro Computer Control (MCC)

Developer License Agreement

****** NOTICE TO USER: THIS IS A CONTRACT. BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.******

This Developer License Agreement accompanies an MCC Software product and related explanatory materials ("Software"). The term "Software" also shall include any upgrades, modified versions or updates of the Software licensed to you by MCC.

Please read this Agreement carefully.

Upon your acceptance of this Agreement, MCC grants to you a nonexclusive license to use the Software, provided that you agree to the following:

1. Use of the Software.

You may install the Software on a hard disk or other storage device; install and use the Software on a file server for use on a network for the purposes of (i) permanent installation onto hard disks or other storage devices or (ii) use of the Software over such network; and make backup copies of the Software.

2. Distribution of Software.

You may make and distribute unlimited copies of the Software, including copies for commercial distribution, as long as:

a. The Software is not the sole or major component of the distribution.

b. Each copy that you make and distribute contains this Agreement.

c. Each copy that you make and distribute contains the same copyright and other

proprietary notices pertaining to this Software that appear in the Software.

If you download the Software from the Internet or similar on-line source, you must include the MCC copyright notice for the Software with any on-line distribution and on any media you distribute that includes the Software.

3. Ownership of Software.

This Software is owned by MCC or its suppliers and is protected by copyright law and international copyright treaty. Therefore, you must treat this Software like any other copyrighted material (e.g., a book), except that you may make and distribute copies of the Software as defined in "Distribution of Software", above.

4. Export Restrictions

This Software is subject to U.S. Commerce Department export restrictions, and is intended for use in the country into which MCC distributed it (or in the EEC, if distributed into the EEC).

5. Termination

Violation of any of the above provisions automatically terminates this license. Upon termination of this license, you agree to stop any distribution of the Software, and destroy all copies of the Software in your possession.

Life Support Applications

MCC Products are not designed for use in life support appliances, devices, or systems where the malfunction of a MCC Product can reasonably be expected to result in a personal injury.

High Risk Activities

The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring

fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). MCC and its suppliers specifically disclaim any express or implied warranty of
fitness for High Risk Activities.

Limited Warranty

MCC warrants, as the sole warranty, that any disks on which the Software is furnished will be free of defects in materials and workmanship under normal use and conditions for a period of thirty (30) days from the date received. No distributor, dealer, or any other entity or person is authorized to expand or alter this Agreement. MCC does not warrant that the functions contained in the Software will be uninterrupted or error-free. Except as stated above in this paragraph, the Software is provided as is without warranty of any kind either expressed or implied, included but not limited to the implied warranties of merchantability and fitness for a particular purpose. You assume entire risk as it applies to the quality and performance of the Software. Should the Software prove defective, the Purchaser (and not MCC, authorized MCC distributors, or dealers) assume the entire cost of all necessary servicing, repair or correction.

Limitation of Remedies and Damages

MCC's entire liability and remedy will be the replacement of any disks not meeting MCC "Limited Warranty" explained above.

In no event will MCC be liable for any damages direct, indirect, incidental, or consequential, including damages for lost profits, lost savings, or other incidental or consequential damage arising out of the use or inability to use such Software, even if MCC has been advised of the possibility of such damages or for any claim by any other party. In no event will MCC's liability of damages to the you or any other person ever exceed the amount of the license fee you paid to use the Software

regardless of the form of the claim.

U.S. GOVERNMENT RESTRICTED RIGHTS

The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraphs(c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs(c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Micro Computer Control Corporation, 17 Model Avenue, Hopewell, New Jersey 08525.

GENERAL PROVISIONS

This license may only be modified in writing signed by you and an authorized officer of MCC. If any provision of this statement is found void or unenforceable, the remainder will remain valid and enforceable according to its terms. If any remedy provided is determined to have failed for its essential purpose, all limitations of liability and exclusions of damages set forth in the Limited Warranty shall remain in effect.

This Agreement is governed by the laws of the State of New Jersey USA (except USA federal law governs copyrights and registered trademarks.) If an provision of this Agreement is deemed invalid by any court having jurisdiction, that particular provision will be deemed deleted and will not affect the validity of any other provision of this Agreement. MCC reserves all rights not specifically granted in this statement.