Command Interface

for

MCC

ASCII Interface I^2C Bus Adapters

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.

WARNING - Life Support Applications: MCC products are not designed for use in life support appliances, devices, or systems where the malfunction of the product can reasonably be expected to result in a personal injury.

WARNING - Radio Frequency Emissions: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules, and the CE General Emissions Standard EN55032 and General Immunity Standard EN55035. These limits are designed to provide reasonable protection against interference when the equipment is operated in a ***commercial environment***. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. Operation of this equipment in a ***residential area*** is likely to cause interference, in which case the user will be required to correct the interference at his own expense.

WARNING - Electrostatic Discharge (ESD) Precautions: Any damage caused by Electrostatic Discharge (ESD) through inadequate earth grounding is NOT covered under the warranty of this product. See the “Electrostatic (ESD) Precautions” section of this guide for more information.

Electronic Waste Notice - This product must NOT be thrown into general waste, but should be collected separately and properly recycled under local regulations.

Created in the United States of America
12-OCT-2018

Table of Contents

Introduction

Micro Computer Control Corporation provides three methods for creating custom application software for ASCII Interface I^2C Bus Adapters:

MS.NET Class Library (Recommended)

The MS.NET Class Library provides a comprehensive set of tools for the creation of robust I^2C Bus applications. Included are Constructors, Methods, Properties, Events, Enumerations, and SampleCode for Visual Basic.NET, Visual C#, Visual C++, Visual J#, and LabVIEW.

For more information see:

www.mcc-us.com/dotNET/ASCIIiPortDotNetClassLibrary.htm

LabVIEW VI Library

The LabVIEW VI Library provides a complete set of low-level, mid-level, and high-level Virtual Instruments (Vis) for the LabVIEW developer. Included are VIs for establishing a connection to the Adapter, performing I^2C Bus Master and Slave operations, and Sample LabVIEW applications.

For more information see:

www.mcc-us.com/LabVIEW_Library/LabVIEW_Library.htm

ASCII Command Interface

The ASCII Command Interface provides a direct low-level ASCII command application program interface to the I^2C Bus Adapter. ASCII commands can be accessed from a terminal emulation program running on the host computer, or from an application program using host computer operating system serial port functions.

This Programmer's Reference describes the ASCII Command Interface, and includes a Quick Start, ASCII Command Table, ASCII Command and Response descriptions, and Sample Code.

Programmer's Quick Start

Creating a custom Adapter program is easier if you know what to expect. Follow these steps to manually control the Adapter from your computer's keyboard and screen using a terminal emulation program.

1. Install the Adapter as directed in the "Hardware Set-Up" section of the Adapter's User's Guide.

2. Use a terminal emulator program to start communicating with the I^2C adapter. Remember to select the correct Com Port (i.e. see Windows Device Manager) and set the communication parameters to 19200 Baud, 8 Data Bits, No Parity, and 1 Stop Bit.

3. Enter **//[CR]** to get an Adapter Status Report. Note: All Adapter commands are terminated with a Carriage Return ([CR]) character. On most terminal emulators, press the Enter key.

4. Enter **/F0[CR]** (XON/XOFF) or **/F1[CR]** (RTS/CTS) to set Adapter's communications Flow Control to match your terminal.

5. Enter **/Ixx[CR]** (xx = 02…FE even) to set Adapter's Own I^2C Slave Address.

6. Enter **/O[CR]** to Open the Adapter Connection. The Adapter does not need to be connected to an I^2C Bus to open a connection.

7. Enter **/Dxx[CR]** (xx = 00…FE even) to select a Destination I^2C Slave Address

8. Enter **/Ttext[CR]** (text = ASCII or Hex-Equivalent ~00…~FF) to Master Transmit a message to the current Destination I^2C Slave device

9. Enter **/Rn[CR]** (n = 0…32767) to Master Read a message from the current Destination I^2C Slave device.

ASCII Command Interface

Note: [CR] = Carriage Return Code or Enter Key.
Syntax: [Select], (Optional), xx = [00..FE], n = [0..32767]

Adapter Commands

Command	Description	Supported Adapters
Ctrl/R,Ctrl/R,Ctrl/R	**Adapter Reset** This command resets the Adapter to its default state.	All
//[CR]	**Status Display** Display Adapter status information.	All
/B[0|1|2][CR]	**Serial Communication Baud Rate Control** Set the Adapter's serial communication baud rate (0 = 19.2K, 1 = 57.6K, 2 = 115.2K Baud).	All except iPort/AI which operates at a fixed rate of 19.2K baud. i2cStick, iPort/USB 2, and i2cStick 2 internally re-maps all three baud rates to 115.2K.
/C[CR]	**Close I^2C Connection** Disconnect Adapter from the I^2C Bus.	All
/Dxx[CR]	**Set Destination I^2C Slave Address** Set the Adapter's destination I^2C Slave Address for subsequent Master Transmit or Receive operations.	All
/E[0|1][CR]	**Echo/Prompt Control [0 = Off, 1 = On]** Enable/Disable data entry echo and prompts.	All
/F[0|1][CR]	**Flow Control [0 = XON/XOFF, 1 = RTS/CTS]** Select serial communication handshaking protocol.	All
/G[0|1][CR]	**I^2C General Call Control [0 = Disabled, 1 = Enabled]** Enables/Disables Adapter response to I^2C Bus General Call (00) messages.	All
/H[0|1][CR]	**Hex Only Display Control [0 = Disabled, 1 = Enabled]** Controls display format of received message data.	All
/Ixx[CR]	**Set Adapter's Own I^2C Slave Address** Sets Adapter's own I^2C Slave Address. Adapter will respond to I^2C Bus messages sent to this address.	All

/K[0|1|2|3][CR]	**I²C Bus Clock Rate Control** Set I²C Bus Clock Rate Control (0=23, 1=86, 2=100, 3=400 KHz)	All except iPort/AI which operates at a fixed rate of 100 KHz.
/M[CR]	**Command Menu Display** Displays Adapter's Command Menu	All
/N([0|1|A|R])[CR]	**iNterrupt Signal Monitor/Control/Status** Sets Monitor/Control/Status of INT line [0 = Disable, 1 = Enable, A = Assert, R = Release, CR=Status]. INT signal not supported on i2cStick and i2cStick 2.	All except iPort/AI, i2cStick, and i2cStick. INT signal not present.
/O[CR]	**Open I²C Connection** Activates Adapter as an I²C device attached to the bus.	All
/(*)Rnnnn[CR]	**Master Read Message** Read the specified number of data bytes from the current Destination I²C Slave device. * = No Stop for Repeated Start.	All. iPort/AI V2.00+firmware required for * support.
/Stext[CR]	**Slave Transmit Message** Write the specified data bytes to a requesting I²C Master Receiver device.	All
/(*)Ttext[CR]	**Master Transmit Message** Master Transmit the specified data bytes to the current Destination I²C Slave device. * = No Stop for Repeated Start.	All. iPort/AI V2.00+firmware required for * support.
/Un[CR]	**I²C Bus Time-oUt** Set I²C Bus Time-oUt in msec (0=Disable)	All except iPort/AI which operates with a fixed time-out of 1 second.
/V[CR]	**Display Adapter Firmware Version** (Major XX.XX Minor)	All except iPort/AI. iPort/AFM requires firmware V2.0+.
/X[CR]	**eXtended Commands** (See Description Below)	All except iPort/AI. iPort/AFM requires firmware V2.0+.
/(*)Y[CR]	**Display Tx bYte Count** Display number of data bytes last sent to slave device. * = Also display last received Acknowledgment bit from slave device.	All except iPort/AI. iPort/AFM requires firmware V2.0+. All except iPort/AFM, iPort/USB, and iPort/LAN for * support.

Adapter Responses		
Response	**Description**	**Supported Adapters**
“*”	Adapter Ready	All
/BC0/1/2[CR]	Baud Rate Change Complete	All except iPort/AI.
/CCC[CR]	Close Connection Complete	All
/NSA[CR]	INT Signal Asserted (Low)	All except iPort/AI, i2cStick, and i2cStick 2.
/NSR[CR]	INT Signal Released (High)	All except iPort/AI, i2cStick, and i2cStick 2.
/OCC[CR]	Open Connection Complete	All
/MTC[CR]	Master Transmit Complete	All
/MRCtext[CR]	Master Read Complete	All
/STR[CR]	Slave Transmit Request	All
/STC[CR]	Slave Transmit Complete	All
/VCCXX.XX[CR]	Firmware Verson (Major.Minor)	All except iPort/AI. iPort/AFM requires firmware V2.0+.
/XCC[CR]	See eXtended Command Section	All except iPort/AI. iPort/AFM requires firmware V2.0+. All except iPort/AFM, iPort/USB, and iPort/LAN for * support.
/SNA[CR]	Slave Not Acknowledging	All
/I81[CR]	Adapter Busy	All
/I83[CR]	Arbitration Loss	All
/I84[CR]	I^2C Bus Error Detected	All
/I85[CR]	I^2C Bus Time-out Detected	All
/I88[CR]	Adapter Connection Closed	All
/I89[CR]	Invalid Command Argument	All
/I8A[CR]	Slave Tx Request Not Active	All
/I8F[CR]	Invalid Adapter Command	All
/I90[CR]	Adapter Rx Buffer Overflow	All

Synchronous Interface Events

Synchronous Events are those Adapter interface activities initiated by the Host computer.

Adapter Reset

Reset Adapter to its default state.

The Reset Command consists of three (3) sequential Ctrl/R characters. Ctrl/R is the character code Decimal 18 and Hexadecimal 0x12. When using a terminal emulator program, you can generate a Ctrl/R by holding down the Ctrl key and pressing the R key.

To guarantee an Adapter reset to the default baud rate, the Reset Command should be sent at all three possible baud rates, with a minimum 10 msec delay after each command.

Note: It is recommended that the Host computer turn off all serial port flow control before sending the Reset Command to override any flow control from the I^2C adapter that could block the transmission. Host computer flow control should be enabled once the Reset is complete.

Command: Ctrl/R,Ctrl/R,Ctrl/R ‘Adapter Reset
Response. * ‘Ready
Default Setting: Adapter Link Closed, Baud Rate 19.2K

Status Display

Display Adapter status.

Command: //[CR] 'Status Display
Response (Device Specific):
Adapter I^2C Host Adapter Vxx.xx
Copyright © xxxx, Micro Computer Control Corp.
Visit our Web Site at: http://www.mcc-us.com

Serial Communications Baud Rate (115.2kHz)
Destination I^2C Slave Address (xxH)
Echo/Prompt (Disabled)
Flow Control (XON/XOFF)
Hex Only Display (Enabled)

I²C Connection (Closed)
General Call (Enabled)
iPort's own Slave Address (xxH)
I²C Bus Clock Rate (100kHz)
iNterrupt Signal (Released)
I²C Bus Time-oUt (10000 msec)

Serial Communications Baud Rate

This command sets the serial communications baud rate.
(0=19.2k, 1=57.6k, 2= 115.2k)

Command: /B[0|1|2][CR]	'Set Serial Com Baud Rate
Response 1: /BC0[CR]	'Baud Change Complete
Response 2: /BC1[CR]	'Baud Change Complete
Response 3: /BC2[CR]	'Baud Change Complete
Response 3: /I89[CR]	'Invalid Command Argument

Default Setting: /B0[CR]

Adapter response is sent a previous baud rate. Host computer should change its baud rate after receiving the Baud Change Complete response.

Note: Supported by all adapters except the iPort/AI which operates at a fixed 19.2K baud rate. i2cStick, iPort/USB 2, and i2cStick 2 internally re-maps 19.2K baud and 57.6K baud to 115.2K baud. No Baud Rate command required.

Close I²C Connection

Disconnect Adapter from the I²C Bus.

Command: /C[CR]	'Close I²C Connection
Response: /CCC[CR]	'Close Connection Complete
Default Setting:	'Closed

Note: Supported by all adapters.

Set Destination I²C Slave Address

Set the destination I²C Slave Address (Hex-Even 00,02...FE) for all subsequent Master Transmit or Receive operations.

Command: /Dxx[CR] 'Set Destination I^2C Slave Address
Response 1: * 'Adapter Ready
Response 2: /I89[CR] 'Invalid Command Argument
Default Setting: 00

Adapter automatically transmits Slave Address LS bit 0 for Write, or 1 for Read operations.

Note: Supported by all adapters.

Echo/Prompt Control

This command enables or disables data entry echo and prompts used as feedback to manual operations from a computer terminal.

Command: /E[0|1][CR] 'Echo/Prompt Control [0 = Off, 1 = On]
Response: * 'Adapter Ready
Default Setting: Off

Note: Supported by all adapters.

Serial Communications Flow Control

Select the serial communication handshaking protocol to be used in communicating with the Host computer.

The Adapter implements either XON/XOFF (by default) or RTS/CTS flow control protocols. Flow control is used by the Adapter to limit character flow to and from the Host computer to avoid overflowing internal communication buffers and lost data.

The XON/XOFF protocol inserts characters directly into the ASCII data stream. XON (Hexadecimal 0x11) is used to enable the flow of data. XOFF (Hexadecimal 0x13) is used to stop the flow of data.

The RTS/CTS protocol uses two additional wires in the cable connecting communicating devices. The RTS wire is an output signal. It indicates that the device generating the signal has buffer space available, and can receive data. The CTS wire is an input signal. It indicates that the other device has buffer space available, and can receive more data.

In general, XON/XOFF requires a minimal three-wire connection, Ground, Transmit Data, and Receive Data. This protocol does insert control characters into the stream of data, and may not be appropriate for all Host systems. If supported, these control characters are normally automatically stripped out of the data stream by Host communication driver software, and are not visible at the application program level.

The RTS/CTS protocol requires a serial port, cabling, and Host communication driver software that supports the additional control signals.

Command: /F[0|1][CR] Flow Control [0 = XON/XOFF, 1 = RTS/CTS]
Response: * 'Adapter Ready
Default Setting: XON/XOFF

Note: Supported by all adatpers.

I^2C General Call Control

Enables or disables Adapter response to I^2C Bus General Call (Address x00) messages.

Command: /G[0|1][CR] 'I^2C General Call [0 = Disabled, 1 = Enabled]
Response: * 'Adapter Ready
Default Setting: Enabled

Note: Supported by all adapters.

Hex Only Display Control

Controls Hex Only (~00...~FF) output of Master or Slave received data.

When enabled, all received I^2C message data bytes are displayed in Hex (~00…~FF) format. When disabled, received I^2C message data bytes representing ASCII printable characters (x20...x7F) are displayed as their ASCII printable character. Non-ASCII printable data bytes are always displayed in Hex (~00…~FF) form.

Command: /H[0|1][CR] 'Hex Only Display [0 = Disabled, 1 = Enabled]
Response: * 'Adapter Ready
Default Setting: Enabled

Note: Supported by all adapters.

Set Adapters's Own I²C Slave Address

Sets Adapter's own I²C Slave Address (Hex 2...FE). Subsequent I²C messages to this address will cause Adapter to become an active Slave device on the bus.

Command: /Ixx[CR] 'Set Adapter's Own I²C Slave Address
Response 1: * 'Adapter Ready
Response 2: /I89[CR] 'Invalid Command Argument
Default Setting: 6E

Note: Supported by all adapters.

I²C Bus Clock Rate Control

Set the I²C Bus master clock rate. (0=23, 1=86, 2=100, 3=400 KHz)

Command: /K[0|1|2|3][CR] 'Set Adapter's Clock Rate (Device Specific)
Response 1: * 'Adapter Ready
Default Setting: /K2[CR]

The Adapter's I²C Bus master clock rate is controlled by the frequency of the oscillator used in the adapter. The oscillator frequency has been selected to give accurate serial baud rates, as the serial baud rate must exactly match the rate used by the host computer. Actual master I²C clock rates are close to, but not faster than, the stated rates. Slave I²C clock rates are driven by the external master device, with possible clock-stretching as required to store or retrieve message data.

Note: Supported by all adapters except the iPort/AI that operates at a fixed rate of 100 KHz.

Command Menu Display

Display Adapter's command menu.

Command: /M[CR] 'Command Menu Display
Response (Device Specific):
Adapter Command Menu Syntax: [Select], (Optional), xx=[00..FE], n=[1..32767]

//	Status Display
/B[0|1|2]	Serial Baud Rate Control (0=19.2, 1=57.6, 2=115.2KHz)
/C	Close I^2C Connection
/Dxx	Set Destination I^2C Slave Address
/E[0|1]	Echo/Prompt Control (0=Disable, 1=Enable)
/F[0|1]	Flow Control (0=XON/XOFF, 1=RTS/CTS)
/G[0|1]	General Call Control (0=Disable, 1=Enable)
/H[0|1]	Hex Only Display Control (0=Disable, 1=Enable)
/Ixx	Set Adapter's Own I^2C Slave Address
/K[0|1|2|3]	I^2C Bus Clock Rate Control (0=23, 1=86, 2=100, 3=400 KHz)
/M	Menu Display
/N([0|1|A|R])	iNterrupt Signal Monitor/Control/Status (0=Disable, 1=Enable / A=Assert, R=Release / <CR>=Status)
/O	Open I^2C Connection
/(*)Rn	Master Rx Message *=No Stop
/S(text)	Slave Tx Message
/(*)T(text)	Master Tx Message *=No Stop
/Un	Set I^2C Bus Time-oUt in msec 0=Disable)
/V	Display Firmware Version (Major XX.XX Minor)
/X[...]...	Extended Cmds (See Prompt or User's Guide)
/(*)Y	Display Tx bYte Count *= with last received Ack bit

Note: Supported by all adapters.

Interrupt Signal Control/Status

The INT signal allows the Adapter to participate in INT master and/or slave communications.

	Control	
Command:	/N0[CR]	Disable INT Signal Monitor *
	/N1[CR]	Enable INT Signal Monitor *
	/NA[CR]	Assert (Low) INT Signal *
	/NR[CR]	Release (Hi-Z) INT Signal *
	Status	
Command:	/N[CR]	Status Request
Response:	/NSA	INT Signal Asserted (Low)
	/NSR	INT Signal Released (High)

Response: * 'Adapter Ready

Default Setting: Monitor Disabled, INT Signal Released

Note: Supported by all adapters except the iPort/AI, i2cStick, and i2cStick 2 which do not support the INT signal.

Open I²C Connection

Activates Adapter as an active device on the I²C Bus.

Command: /O[CR] 'Open I²C Connection
Response: /OCC[CR] 'Open Connection Complete
Default Setting: Closed

Note: Supported by all adapters.

Master Read Message

This command causes Adapter to read the specified number of data bytes from the currently selected Destination I²C Slave Address with or without generating an I²C Stop condition after the last byte is received.

Enter Byte Count (Decimal 0...32767) then Press Enter, or ESCape to Cancel. A Byte Count of Zero (0) represents a Variable Length message, where the first byte read from the I²C Slave device indicates the number of additional trailing bytes that are available to read. The Adapter automatically reads the first byte, then the additional bytes as specified by the first byte. All message bytes including the Length byte are returned to the Host computer.

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

If the slave device acknowledges its I²C Slave Address, the specified number of bytes are read. The Adapter acknowledges all bytes read except the last. If not disabled, the message is then terminated with an I²C Stop condition.

Sending Master Receive messages with No Stop allows the Master to retain exclusive control of the I²C Bus until it finally sends a Stop. During this time, the Master can send additional (Repeated Start) Master Transmit or Master Receive messages to the same or other I²C Slave devices.

Command: /(*)Rnnnn[CR] 'Master Read Message (* = No Stop)

Response 1: /MRCtext[CR] 'Master Read Complete
Response 2: /SNA[CR] 'Slave Not Acknowledging
Response 3: /I81[CR] 'Adapter is Busy, Command Ignored
Response 4: /I83[CR] ' I^2C Arbitration Loss Detected
Response 5: /I88[CR] 'Connection Not Open
Response 6: /I89[CR] 'Invalid Command Argument
Default Setting: None

Note: Supported by all. iPort/AI V2.00+firmware required for * support.

Slave Transmit Message

This command should be issued to Adapter in response to a Slave Transmit Request (/STR). This command causes Adapter to write the specified data bytes to the requesting I^2C Master Receiver device.

Enter Message Bytes (1 or more Printable ASCII or Hex-equivalent ~00..~FF), then Press Enter, or ESCape to Cancel.

Note 1: Upon receiving a Slave Transmit request from a Master Receiver device on the I^2C Bus, the Adapter outputs a Slave Transmit Request to its Host computer, and initiates an I^2C Clock Stretch (SCL Low) until a Slave Transmit command is received from the Host computer. While clock stretching, no other messages can be transmitted on the I^2C Bus.

Note 2: The tilde (~) character and the Carriage Return (CR) characters are used as special marker characters within all Adapter transmitted text messages. These characters may not be used within the text of a message, but must be replaced by the following "Hex equivalent" characters:

Tilde replaced by "~7E"
Carriage Return replaced by "~0D"

Adapter automatically translates "Hex equivalent" characters to their single-byte value for transmission across the I^2C Bus.

All entered data bytes are transmitted to the requesting Master Receiver device. Slave Transmit stops upon receiving the first negative acknowledgment (Nack)

from the Master Receiver.

Command: /Stext[CR] 'Slave Transmit Message
Response 1: /STC[CR] 'Slave Transmit Complete
Response 2: /I88[CR] 'Connection Not Open
Response 3: /I8A[CR] ‘Slave Transmit Request Not Active, Cmd Ignored
Default Setting: None

Examples:

/Sabcd1234[CR] ‘ASCII Printable characters "abcd1234"
/S~00~01~02[CR] ‘Binary data bytes 00, 01,02
/Sab~7Ecd[CR] ‘Tilde embedded in ASCII Printable characters
/S12~0D24[CR] ‘Carriage Return embedded in ASCII Printable characters

Note: Supported by all adapters.

Master Transmit Message

Write the specified data bytes to the currently selected Destination I^2C Slave Address with or without generating an I^2C Stop condition after the last byte is transmitted.

Enter Message Bytes (0 or more Printable ASCII or Hex-equivalent ~00..~FF), then Press Enter, ESCape to Cancel.

Note: The tilde (~) character and the Carriage Return (CR) characters are used as special marker characters within all Adapter transmitted text messages. These characters may not be used within the text of a message, but must be replaced by the following "Hex-equivalent" characters:

Tilde replaced by "~7E"
Carriage Return replaced by "~0D"

Adapter automatically translates "Hex equivalent" characters to their single-byte value for transmission across the I^2C Bus.

All entered data bytes are transmitted to the Destination I^2C Slave Receiver device. Master Transmit stops upon receiving the first negative acknowledgment (Nack)

from the Slave Receiver. If not disabled, the message is then terminated with an I²C Stop condition.

Sending Master Transmit messages with No Stop allows the Master to retain exclusive control of the I²C Bus until it finally sends a Stop. During this time, the Master can send additional (Repeated Start) Master Transmit or Master Receive messages to the same or other I²C Slave devices.

Note: See the Display Tx bYte Count command (/Y) for additional information on the last completed Master Transmit message.

Command: /(*)Ttext[CR]	'Master Transmit Message (* = No Stop)
Response 1: /MTC[CR]	'Master Transmit Complete
Response 2: /SNA[CR]	'Slave Not Acknowledging
Response 3: /I81[CR]	'Adapter is Busy, Command Ignored
Response 4: /I83[CR]	' I²C Arbitration Loss Detected
Response 5: /I88[CR]	'Connection Not Open
Default Setting:	None

Examples:

/Tabcd1234[CR]	‘ASCII Printable characters "abcd1234"
/T~00~01~02[CR]	‘Binary data bytes 00, 01,02
/*T~00~01~02[CR]	‘Binary data bytes 00, 01,02 with No Stop
/Tab~7Ecd[CR]	‘Tilde embedded in ASCII Printable characters
/T12~0D24[CR]	‘Carriage Return embedded in ASCII Printable characters

Note: Supported by all. iPort/AI V2.00+firmware required for * support.

Set I²C Bus Time-oUt in msec

Set I²C Bus time-out in milliseconds.

The Adapter reports a bus time-out if no inter-byte bus activity for the specified time occurs within an I²C Bus message.

Command: /Unnnnn[CR]	'I²C Bus time-oUt (nnnn = 0 (disable)...32000 msec)
Response: *	'Adapter Ready
Default Setting:	10000 msec (10 seconds)

Note: Supported by all adapters except iPort/AI which operates with a fixed time-out of 1 second.

Display Firmware Version

Display Adapter firmware version

Command: /V[CR] 'Firmware Version
Response: /VCCXX.XX[CR] '(Major XX.XX Minor)

Note: Supported by all adapters except iPort/AI. iPort/AFM requires firmware V2.0+.

eXtended Commands

The eXtended commands are used to generate "out-of-spec" signaling. eXtended commands cannot use the adapter's I^2C hardware to control the SCL and SDA lines, as the I^2C hardware only generates I^2C compatible signals. The eXtended commands use firmware to "bit-bang" the SCL and SDA lines. This firmware cannot operate as fast as the hardware, and it can be interrupted at any time by adapter internal interrupts.

The eXtended commands run directly off the command characters as they are received on the serial link. Speed of execution of eXtended commands is controlled by the serial link communication rate, the execution speed of the firmware, delays caused by execution interruptions that may occur while a command is executing, and I^2C Bus clock-stretching by external slave devices.

The following commands manipulate the I^2C Clock (SCL) and data (SDA) lines.

Command: /X[S|~xx|R|r|P|0|1|?|D|d|C|c|L|A| |"]..., then Press Enter or ESCape
Enter /X followed by zero or more sub-commands, then [CR]

Response: /XCC(see commands below)[CR]

High Level Sub-Commands:

S = Send Start
~xx = Send Byte (xx = 00...FF)(response = A or N)
R = Read Byte with Ack (response = ~xx)

r = Read Byte with Nak (response = ~xx)
P = Send Stop

Mid Level Sub-Commands:

0 = Send 0 Bit
1 = Send 1 Bit
? = Read Bit (response = 0 or 1)

Low Level Sub-Commands:

D = Set SDA High
d = Set SDA Low
C = Set SCL High
c = Set SCL Low
L = Read SCL (response = 0 or 1)
A = Read SDA (response = 0 or 1)

Miscellaneous Sub-Commands:

space = no action
"comment" = no action

Examples:

Master transmit three bytes to slave address 0x4e using high level, mid level, and low level sub-commands.

High Level Command:/X S ~4e ~01 ~02 ~03 P [CR]
High Level Response: /XCCAAAA[CR]

Mid Level Command:/X S 01001110 ? 00000001 ? 00000010 ? 00000011 ? P [CR]
Mid Level Response: /XCC0000[CR]

Low Level Command:/X dc dCcDCcdCcdCcDCcDCcDCcdCc DCAc dCcdCcdCcdCcdCcdCcdCcDCc DCAcdCcdCcdCcdCcdCcdCcDCcdCc DCAc dCcdCcdCcdCcdCcdCcdCcDCc DCAc dCD[CR]
Low Level Response: /XCC0000[CR]

Master read three bytes from slave address 0x4F. First two bytes are acknowledged

by master.

Command: /X S ~4f Rrr P [CR]
Response: /XCCA~xx~xx~xx[CR] ‘(xx = 00...FF)

Master transmit a Write WCR command to a Xicor X9241 at slave address 0x50. WCR data is 0x00.

Command: /X S ~50 ~a0 ~00 P [CR]
Response: /XCCAAA[CR]

Master transmit a Write WCR command to a Xicor X9241 at slave address 0x50. WCR data is 0x3f.

Command: /X S ~50 ~a0 ~3f P [CR]
Response: /XCCAAA[CR]

Issue a Read WCR command to a Xicor X9241 at slave address 0x50.

Command: /X S ~50 ~90 ~R P [CR]
Response: /XCCAA~xx[CR] ‘(xx = 00...FF)

Issue an Increment Wiper command to a Xicor X9241 at slave address 0x50.

Command: /X S ~50 ~20 1 P [CR]
Response: /XCCAA[CR]

Issue a Decrement Wiper command to a Xicor X9241 at slave address 0x50.

Command: /X S ~50 ~20 0 P [CR]
Response: /XCCAA[CR]

Note: Supported by all adapters except iPort/AI. iPort/AFM requires firmware V2.0+.

Display Tx bYte Count

Returns the number of bytes received by the slave device in the last master transmit message, with an option to received the state of the last received Acknowledgment

bit.

Note: The byte count and last received acknowledgment bit state can be used for SMBus Packet Error Control (PEC) error detection. See SMBus v1.1+ specifications for details.

Command: /(*)Y[CR]	‘Tx bYte Count (* = with last received Ack bit)
Response: /TBCn[CR]	'n =00000...32767
Response: /TBCn(A|N)[CR]	'n =00000...32767, A = ACK, N = NACK

Note: Supported by all adapters except iPort/AI. iPort/AFM requires firmware V2.0+. * supported by all except iPort/AFM, iPort/USB, and iPort/LAN.

Asynchronous Interface Events

Asynchronous Events are those Adapter interface activities initiated by the Adapter in response to activities on the I^2C Bus.

Slave Transmit Request

This event is caused by the reception of an I^2C Bus Slave Transmit message directed at the current Adapter's own Slave address.

Prompt: /STR[CR] 'Slave Transmit Request
Command: /Stext[CR] 'Slave Transmit Text

The normal Host computer response is to send a Slave Transmit (/Stext[CR]) command.

Note: Upon receiving a Slave Transmit request from a Master Receiver device on the I^2C Bus, Adapter outputs a Slave Transmit Request to its Host computer, and initiates an I^2C Clock Stretch (SCL Low) until a Slave Transmit Text command is received from the Host computer. While clock stretching, no other messages can be transmitted on the I^2C Bus.

Slave Receive Complete

This event is caused by the reception of an I^2C Bus Slave Receive message directed at the current Adapter's own Slave address.

The received text is a representation of the data bytes within the Slave Receive message. The format of this data is controlled by the current setting of the Hex Only Display Control.

Prompt: /SRCtext[CR] 'Slave Receive Complete
Command: None Required

General Call Receive Complete

This event is caused by the reception of an I^2C Bus Slave Receive message directed at the I^2C General Call Address (00), when Adapter's General Call recognition is enabled.

The received text is a representation of the data bytes within the Slave Receive message. The format of this data is controlled by the current setting of the Hex Only Display Control.

Prompt: /GRCtext[CR] ‘General Call Receive Complete
Command: None Required

Adapter Ready
Prompt: * ‘Adapter Ready

Cause: Adapter is ready for the next Host command.

Slave Not Acknowledging

Prompt: /SNA[CR] ‘Slave Not Acknowledging

Cause: There is no response (I^2C Slave Address Acknowledgment) during a Master Transmit or Receive operation from an I^2C Slave device at the current Destination I^2C Address. Check for wrong Destination Slave Address (/D) or slave device not active.

Adapter Busy

Prompt: /I81[CR] ‘Adapter Busy

Cause: The host computer attempted a Master operation while Adapter was busy. The host computer should wait for any previously issued command to complete, process any pending slave events, and retry the last command.

I^2C Bus Arbitration Loss

Prompt: /I83[CR] ‘I^2C Arbitration Loss Detected

Cause: Adapter lost I^2C Bus Arbitration to another bus master device while Master Transmitting or Master Receiving an I^2C message. Host should process any active slave events and repeat the last command.

I^2C Bus Error Detected

Prompt: /I84[CR] ‘I^2C Bus Error Detected

Cause: Adapter has detected an invalid signal condition on the I^2C Bus. The host computer should retry the last command or issue an Adapter Reset command.

I^2C Bus Time-out Detected

Prompt: /I85[CR] ‘I^2C Bus Time-out Detected

Cause: Adapter issues this response when it detects a byte transfer delay greater than the specified I^2C Bus Time-oUt period. No corrective action is taken by the Adapter regarding I^2C Bus activity. No host computer response is required, but this event can be used to detect possible bus problems.

Adapter Connection Closed

Prompt: /I88[CR] ‘Adapter Connection is Closed.

Cause: The host computer is attempting to perform an I^2C Bus message operation while the Adapter Connection is Closed. The host computer should issue an Open I^2C Connection command before attempting to perform I^2C Bus message operations.

Invalid Command Argument

Prompt: /I89[CR] ‘Invalid Command Argument Detected

Cause: This event normally indicates the value of a host command argument was out of range. The host should reissue command with correct arguments.

Slave Transmit Request Not Active

Prompt: /I8A[CR] ‘Slave Transmit Request Not Active

Cause: This event indicates the host attempted to issue a Slave Transmit Text command when no Slave Transmit Request was present.

Invalid Command

Prompt: /I8F[CR] ‘Invalid Command

Cause: This event normally indicates that an invalid command was issued by the

host. The host should reissue the correct command.

Adapter Receive Buffer Overflow

Prompt: /I90[CR] ‘Adapter Serial Receive Buffer Overflow

Cause: This event normally indicates that data sent to the Adapter via the serial port has been lost. Check the host computer’s Serial Port Flow Control (XON/XOFF, or Hardware) to make sure it matches current Adapter Flow Control. Also, check if the host computer’s FIFO buffers in its 16550 UART are enabled. If so, reduce or disable Transmit Data Buffering. On Windows-based host computers, see the Device Manager, COM port, Advanced Settings. You may need to power down the host computer for any FIFO change to take effect.

Example Code

The following examples are written in MS Visual Basic for Windows using the serial communications control (MSCOMM.VBX). It can be used as a guide in implementing Adapter interface programs in other programming languages and operating environments.

Note: Sample code is also available online at: www.mcc-us.com

Adapter Reset

```
Comm1.Output = Chr$(18)        'Ctrl/R
Comm1.Output = Chr$(18)        'Ctrl/R
Comm1.Output = Chr$(18)        'Ctrl/R
```

Adapter Initialization

```
Comm1.Output = "/f0"           'Set Adapter XON/XOFF Flow Control
Comm1.Output = Chr$(13)

Comm1.Output = "/i70"          'Set Adapter's Own Slave Address (0x70)
Comm1.Output = Chr$(13)

Comm1.Output = "/d4e"          'Set Destination Slave Address (0x4E)
Comm1.Output = Chr$(13)

Comm1.Output = "/o"            'Open I²C Connection
Comm1.Output = Chr$(13)
```

Master Transmit Message

```
Comm1.Output = "/T~00~01" 'Send Master Tx Command (0x00, 0x01)
Comm1.Output = Chr$(13)    'Terminate Command
```

Master Receive Message

```
Comm1.Output = "/R10"          'Send Master Rx Command (10 Bytes)
Comm1.Output = Chr$(13)        'Terminate Command
```

Communication Event Processing

```
Static Sub Comm1_OnComm ()
Static LineBuf$

While Comm1.InBufferCount
    Msg$ = Comm1.Input ' Get Comm input character
    CharIn$ = Msg$

    If Msg$ = Chr$(13) Then Msg$ = ""     ' Remove CR
    If Msg$ = Chr$(10) Then Msg$ = ""     ' Remove LF
    If Msg$ = "*" Then                    ' if Adapter Ready
        Msg$ = "****"                     ‘ Substitute Token
        CharIn$ = Chr$(13)                ‘ Terminate Line
    End If

    LineBuf$ = LineBuf$ + Msg$          'Add new text to line buffer

    If CharIn$ = Chr$(13) Then          ' if Carriage Return detected
        iPortResp$ = Left$(LineBuf$, 4)      'Isolate Response Code

        ' Test for Adapter Synchronous Interface Events

        If (StrComp(iPortResp$, "/OCC") = 0) Then
            ' Open Connection Complete Processing
            TextBox.Text = "/OCC Open Connection Complete"

        ElseIf (StrComp(iPortResp$, "/MTC") = 0) Then
            ' Master Transmit Complete Processing
            TextBox.Text = "/MTC Master Tx Complete"

        ElseIf (StrComp(iPortResp$, "/MRC") = 0) Then
            ' Master Rx Complete Processing
            TextBox.Text = LineBuf$   'Update Display

        ElseIf (StrComp(iPortResp$, "/STC") = 0) Then
            ' Slave Tx Complete Processing
            TextBox.Text = "/STC Slave Tx Complete"

        ElseIf (StrComp(iPortResp$, "/CCC") = 0) Then
            ' Close Connection Complete Processing
```

```
    TextBox.Text = "/CCC Close Connection Complete "

ElseIf (StrComp(iPortResp$, "/BC0") = 0) Then
    ' Adapter Baud Change 0 {19.2K}
    TextBox.Text = "Adapter Baud Change 0 {19.2K} "

ElseIf (StrComp(iPortResp$, "/BC1") = 0) Then
    ' Adapter Baud Change 1 {57.6K}
    TextBox.Text = "Adapter Baud Change 1 {57.6K} "

ElseIf (StrComp(iPortResp$, "/BC2") = 0) Then
    ' Adapter Baud Change 2 {115.2K}
    TextBox.Text = "Adapter Baud Change 0 {115.2K} "

' Test for Adapter Asynchronous Interface Events

ElseIf (StrComp(iPortResp$, "/SRC") = 0) Then
    ' Slave Rx Complete Processing
    TextBox.Text = LineBuf$       'Update Display

ElseIf (StrComp(iPortResp$, "/GRC") = 0) Then
    ' General Call Rx Complete Processing
    TextBox.Text = LineBuf$       'Update Display

ElseIf (StrComp(iPortResp$, "/STR") = 0) Then
    ' Slave Tx Request Processing
    Comm1.Output = "/S~00~01" 'Send Slave Tx Msg (0x00, 0x01)
    Comm1.Output = Chr$(13)     'Terminate Command
    TextBox.Text = LineBuf$       'Update Display

' Test for Adapter Response Messages

ElseIf (StrComp(iPortResp$, "****") = 0) Then
    TextBox.Text = "* Adapter Ready" 'Update Display

ElseIf (StrComp(iPortResp$, "/SNA") = 0) Then
    TextBox.Text = "/SNA Slave Not Acknowledging"

ElseIf (StrComp(iPortResp$, "/I81") = 0) Then
    TextBox.Text = "/I81 Adapter Busy" 'Update Display
```

```
        ElseIf (StrComp(iPortResp$, "/I83") = 0) Then
           TextBox.Text = "/I83 Arbitration Loss" 'Update Display

        ElseIf (StrComp(iPortResp$, "/I84") = 0) Then
           TextBox.Text = "/I84 I2C Bus Error Detected"

        ElseIf (StrComp(iPortResp$, "/I85") = 0) Then
           TextBox.Text = "/I85 I2C Bus Time-out Detected"

        ElseIf (StrComp(iPortResp$, "/I88") = 0) Then
           TextBox.Text = "/I88 Adapter Connection Closed"

        ElseIf (StrComp(iPortResp$, "/I89") = 0) Then
           TextBox.Text = "/I89 Invalid Command Argument"

        ElseIf (StrComp(iPortResp$, "/I8A") = 0) Then
           TextBox.Text = "/I8A Slave Tx Request Not Active"

        ElseIf (StrComp(iPortResp$, "/I8F") = 0) Then
           TextBox.Text = "/I8F Invalid Adapter Command"

        ElseIf (StrComp(iPortResp$, "/I90 = 0) Then
           TextBox.Text = "/I90 Adapter Rx Buffer Overflow”

        Else
           TextBox.Text = LineBuf$   'Other Update Display
        End If
     LineBuf$ = ""
     End If
Wend
End Sub
```

Revision Report

This section defines revisions and changes made to the Adapter interface:

Revision: 1.00

1. Initial Release

Additional Information

For additional information on the I^2C Bus, please refer to the following:

“What is I^2C?”
www.mcc-us.com/I2CBusTechnicalOverview.pdf

“Frequently Asked Questions (FAQ)”
www.mcc-us.com/faq.htm

"The I^2C and How to Use It"
www.mcc-us.com/i2chowto.htm