

VT-142x

Host API Library Software Reference Manual

For the VT-142x Family of PMC DSP Cards

Document Number: 560-002-039

Version 1.3
January, 2005



The Evergreen Group Inc.

IMPORTANT NOTICE

The Evergreen Group Inc. (The Evergreen Group) reserves the right to make changes to their products or to discontinue any product or service without notice. We advise customers to obtain the latest version of relevant information to verify, before placing orders, that information they may be relying on is current and complete. We sell all products subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

The Evergreen Group warrants performance of its products to the specifications applicable at the time of sale in accordance with our standard warranty. Testing and other quality control techniques are utilized to the extent The Evergreen Group deems necessary to support this warranty. We do not necessarily perform specific testing of all parameters of each device.

CERTAIN APPLICATIONS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE (“CRITICAL APPLICATIONS”). THE EVERGREEN GROUP PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. THE CUSTOMER UNDERSTANDS THAT INCLUSION OF THE EVERGREEN GROUP’S PRODUCTS IN SUCH APPLICATIONS IS FULLY AT THE CUSTOMER’S RISK.

In order to minimize risks associated with the customer’s applications, the customer must provide adequate design and operating safeguards to minimize inherent or procedural hazards. The Evergreen Group assumes no liability for applications assistance or customer product design. The Evergreen Group does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of The Evergreen Group covering or relating to any combination, machine, or process in which such products or services might be or are used. The publication of information regarding any third party products or services by The Evergreen Group does not constitute approval, warranty or endorsement thereof by The Evergreen Group.

Windows and Windows NT are registered trademarks of Microsoft Corporation.

TI and Code Composer Studio are trademarks of Texas Instruments Corporation.

Other product and company names herein may be the trademarks of their respective owners.

This document is derived from an original created by Valley Technologies, and is used under license.

Contents

1	Introduction	1
2	VT-1420 Board Setup/Configuration and General Routines	2
2.1	VT1420ConfigBoard	2
2.2	VT1420GetBoardConfig	2
2.3	VT1420GetCardPtr.....	3
2.4	VT1420GetConfigFlag	3
2.5	VT1420Init	4
2.6	VT1420SetDebug	4
2.7	VT1420Terminate	4
3	Bridge Routines	5
3.1	VT1420ReadBridgeConfig.....	5
3.2	VT1420WriteBridgeConfig	5
3.3	VT1420ReadBridgeMem.....	6
3.4	VT1420WriteBridgeMem	7
4	Bridge SRAM Routines	8
4.1	VT1420EraseSRAM.....	8
4.2	VT1420ReadSRAM	8
4.3	VT1420WriteSRAM.....	9
4.4	VT1420SaveSRAMToFile	9
4.5	VT1420WriteSRAMFromFile	10
5	DSP Routines.....	11
5.1	VT1420DualDSPReset.....	11
5.2	VT1420Execute.....	11
5.3	VT1420SetDSP	11
5.4	VT1420SetLED.....	12
5.5	VT1420ReadDSPConfig	12
5.6	VT1420WriteDSPConfig	13
5.7	VT1420ReadDSPMem.....	13
5.8	VT1420WriteDSPMem.....	14
5.9	VT1420WarmReset.....	14
6	DSP RAM Routines	15
6.1	VT1420FillMem.....	15
6.2	VT1420GetSDRAMSize.....	16
6.3	VT1420ReadMemToFile	16
6.4	VT1420WriteMemFromFile	17
6.5	VT1420WriteMemFromDotHexFile	17
6.6	VT1420AddressWriteTest.....	18
6.7	VT1420WalkOnes.....	18
6.8	VT1420RAMTest.....	18
7	DSP Flash Memory Routines	19
7.1	VT1420ResetFlash	19
7.2	VT1420EraseFlash	19
7.3	VT1420EraseFlashSector.....	19

7.4	VT1420GetFlashSize	20
7.5	VT1420GetFlashType.....	20
7.6	VT1420FlashTest	20
7.7	VT1420ReadFlash.....	21
7.8	VT1420ReadFlashToFile	21
7.9	VT1420WriteFlash.....	22
7.10	VT1420WriteFlashFromFile	22
7.11	VT1420WriteFlashFromDotHexFile	23
Appendix A – Related Documentation		24
Applicable documents		24
CUSTOMER SUPPORT.....		25
Technical Assistance.....		25
World Wide Web.....		25
Warranty Service.....		25
Life Support Policy.....		25
Office Location:		26
The Evergreen Group.....		26
Internet:		26

1 Introduction

This document provides the necessary software routine descriptions for the VT142x family of PMC Host API library functions. The VT1420 Host API library is “**vt1420api.lib**”. The required header file for this library is “**vt1420api.h**”.

This document contains a table for each public routine in the VT1420x Host API library. The table contains the following sections:

1. **Prototype** – This section contains the prototype for the routine.
2. **Description** – This section provides a brief description of the routine.
3. **Arguments** – This section provides a list of arguments for the routine and where appropriate valid values for these arguments.
4. **Output** – This section describes any extra output to expect from the routine. Not all routines contain an Output section.
5. **Return** – This section describes the return values for the routine.

2 VT-1420 Board Setup/Configuration and General Routines

2.1 VT1420ConfigBoard

Prototype	int VT1420ConfigBoard (CardPtr cardptr);
Description	This routine configures the 21555 bridge, DSPA (and DSPB if Dual DSP). This includes the PCI, EMIF and LED interfaces.
Arguments	CardPtr cardptr - Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

2.2 VT1420GetBoardConfig

Prototype	int VT1420GetBoardConfig (CardPtr cardptr, ConfigPtr *pconfigptr, BOOLEAN fGetRevID);
Description	This determines the VT142x board configuration and prints the configuration values to the screen. The configuration values include DSP type, DSP speed, number of DSPs, and memory sizes.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT1420xDSP board. ConfigPtr pconfigptr- Pointer to a ConfigPtr structure containing board configuration values. BOOLEAN fGetRevID – If true RevId is printed.
Output	Prints out the VT142x board configuration values.
Return	Integer value - Non Zero return value is an error.

2.3 VT1420GetCardPtr

Prototype	int VT1420GetCardPtr (int nBoardNum, CardPtr *cardptr);
Description	This routine populates the CardPtr structure for the VT142x board. This routine must be called to populate the CardPtr structure for each VT1420xboard before board accesses can be made. After the CardPtr structure is populated it is used in all of the other VT142x API calls.
Arguments	int nBoardNum – Number of the board to access. CardPtr cardptr – Return value containing all of the information required to access the VT142x.
Return	Integer value - Non Zero return value is an error.

2.4 VT1420GetConfigFlag

Prototype	int VT1420GetConfigFlag (CardPtr cardptr, int nRegValue , int *pnReturn);
Description	This routine uses the nRegValue to determine which configuration value to read, either DSP speed or DSP type, from the selected DSP. pnReturn value will contain the requested configuration value.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. int nRegValue - A value of 0 requests the DSP speed value. A value of 1 requests the DSP type. int pnReturn – Contains the obtained requested value: nRegValue = 0, return value of 0 = 400 MHZ nRegValue = 0, return value of 1 = 600 MHZ nRegValue = 1, return value of 0 = 6416DSP nRegValue = 1, return value of 1 = 6415DSP
Return	Integer value - Non Zero return value is an error.

2.5 VT1420Init

Prototype	int VT1420Init (int *pnNumBoards);
Description	This routine enumerates the bus and probes for VT142x cards. It sets the pnNumBoards variable equal to the number of VT142x boards that were found.
Arguments	int pnNumBoards – Returns the number of VT142x boards that were found.
Return	Integer value - Non Zero return value is an error.

2.6 VT1420SetDebug

Prototype	void VT1420SetDebug (BOOLEAN fDebugValue);
Description	This routine sets the debug output level.
Arguments	BOOLEAN fDebugValue – TRUE = debug on FALSE = debug off
Return	No return value

2.7 VT1420Terminate

Prototype	int VT1420Terminate ();
Description	This routine releases all VT1420xCardPtrs. It should be called at the end off all applications. Note: After this call is made VT1420Init must be called in order to talk to the VT142x again.
Arguments	none
Return	Integer value - Non Zero return value is an error.

3 Bridge Routines

3.1 VT1420ReadBridgeConfig

Prototype	int VT1420ReadBridgeConfig (CardPtr cardptr, uint32 nFullAddr, uint32 *pnVal);
Description	This routine reads the bridge PCI configuration value located at nFullAddr and loads the value into the pnVal variable.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 nFullAddr – Address in PCI Configuration space to read. Valid byte offsets for PCI Configuration space range from 0x00 to 0xFF. uint32 pnVal – Configuration value located at nFullAddr.
Return	Integer value - Non Zero return value is an error.

3.2 VT1420WriteBridgeConfig

Prototype	int VT1420WriteBridgeConfig (CardPtr cardptr, uint32 nFullAddr, uint32 nVal);
Description	This routine writes the bridge PCI configuration value nVal to the location nFullAddr.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 nFullAddr – Address for byte offset in PCI Configuration space in which to write. Valid byte offsets for PCI Configuration space range from 0x00 to 0xFF. uint32 nVal – Value to write.
Return	Integer value - Non Zero return value is an error.

3.3 VT1420ReadBridgeMem

Prototype	int VT1420ReadBridgeMem (CardPtr cardptr, uint32 nFullAddr, uint32 *pnVal, int nBar, int nBytes);
Description	This routine reads the value located at nFullAddr and loads that value into the pnVal variable.
Arguments	<p>CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.</p> <p>uint32 nFullAddr – Address offset from base address of selected nBar to read. Address offset values begin at 0x0 and range up to but not exceeding the BAR size indicated in nBar below.</p> <p>uint32 pnVal – Value located at nFullAddr.</p> <p>int nBar – Memory BAR to read. Valid values are: 0: Bridge CSR Memory (4KB, non-prefetchable) 1: Bridge I/O Space (256B) 2: DSP I/O Space (16B) 3: DSP Memory (4MB, prefetchable) 4: DSP Register Space (8MB, non-prefetchable)</p> <p>int nBytes – Number of bytes to read. Valid values are 1, 2, or 4.</p>
Return	Integer value - Non Zero return value is an error.

3.4 VT1420WriteBridgeMem

Prototype	int VT1420WriteBridgeMem (CardPtr cardptr, uint32 nFullAddr, uint32 nVal, int nBar, int nBytes);
Description	This routine writes the value nVal to the location nFullAddr.
Arguments	<p>CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.</p> <p>uint32 nFullAddr –Address offset from base address of selected nBar to write to. Address offset values begin at 0x0 and range up to but not exceeding the BAR size indicated in nBar below.</p> <p>uint32 nVal – Value to write.</p> <p>Int nBar – Memory BAR to write to. Valid values are: 0: Bridge CSR Memory (4KB, non-prefetchable) 1: Bridge I/O Space (256B) 2: DSP I/O Space (16B) 3: DSP Memory (4MB, prefetchable) 4: DSP Register Space (8MB, non-prefetchable)</p> <p>int nBytes – Number of bytes to write. Valid values are 1, 2, or 4.</p>
Return	Integer value - Non Zero return value is an error.

4 Bridge SROM Routines

4.1 VT1420EraseSROM

Prototype	int VT1420EraseSROM (CardPtr cardptr);
Description	This routine erases the SROM.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

4.2 VT1420ReadSROM

Prototype	int VT1420ReadSROM (CardPtr cardptr, uint8 *pcBytes int nNumBytes);
Description	This routine reads the SROM starting at the base address. The amount of data to be read is defined by the nNumBytes variable. The values read from the SROM are returned in the variable pcBytes.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint8 pcBytes – Return value containing the values from the SROM. int nNumBytes – Number of bytes to read.
Return	Integer value - Non Zero return value is an error.

4.3 VT1420WriteSROM

Prototype	int VT1420WriteSROM (CardPtr cardptr, uint8 *pcBytes, int nNumBytes);
Description	This routine programs the SROM with the values contained in the variable pcBytes. The amount of data to write is defined by the variable nNumBytes.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint8 *pcBytes – Contains the data to write to the SROM. int nNumBytes – Defines the amount of data to write to the SROM.
Return	Integer value - Non Zero return value is an error.

4.4 VT1420SaveSROMToFile

Prototype	int VT1420SaveSROMToFile (CardPtr cardptr, int nNumBytes, char *sFilename);
Description	This routine reads the values from the SROM and writes them to the file specified by the variable sFilename.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. int nNumBytes – Number of bytes to read. char sFilename – Directory path and file name of the output file to which memory data is written.
Output	File containing memory values.
Return	Integer value - Non Zero return value is an error.

4.5 VT1420WriteSROMFromFile

Prototype	int VT1420WriteSROMFromFile (CardPtr cardptr, char *sFilename);
Description	This routine programs the SROM with the values contained in the input file. The variable sFilename defines the input file.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name for the input data file. An example file is included with the VT-1420 installation.
Return	Integer value - Non Zero return value is an error.

5.8 VT1420WriteDSPMem

Prototype	int VT1420WriteDSPMem (CardPtr cardptr, uint32 nFullAddr, uint32 nVal, int nBytes);
Description	This routine writes the value nVal to the location nFullAddr. The value is written to the DSP processor that was specified using the <i>VT1420SetDSP</i> routine. This routine can access any address in the DSP's memory space and it does so by first properly setting the DSPP page register of the DSP.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 nFullAddr – Address to write to. uint32 nVal – Value to write. int nBytes – Number of bytes to write. Valid values are 1, 2, or 4.
Return	Integer value - Non Zero return value is an error.

5.9 VT1420WarmReset

Prototype	int VT1420WarmReset (CardPtr cardptr);
Description	This routine sets the DSP internal reset register. This causes the selected DSP to perform an internal reset. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

6 DSP RAM Routines

Prior to using all DSP RAM routines, the proper DSP must be selected by calling the routine *VT1420SetDSP*. DSP RAM includes internal RAM and SDRAM on the EMIF A. In addition, the *VT1420ReadMemToFile* routine works with the Flash on the EMIF B. Note: the EMIFs must be configured prior to accessing the SDRAM and Flash (see *VT1420ConfigBoard*).

6.1 VT1420FillMem

Prototype	int VT1420FillMem (CardPtr cardptr, uint32 nStart, uint32 nNumWords, uint32 nVal);
Description	This routine fills the RAM memory of the selected DSP starting at nStart. The value to be written is defined by the nVal variable. The amount of data to be written is defined by the nNumWords variable. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. This routine can access any address in the DSP's memory space and it does so by properly setting the DSPP page register of the DSP. The Flash memory cannot be written to using this function.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 nStart – Starting address for the memory fill. uint32 nNumWords – Number of words to write. uint32 nVal – Value to write.
Return	Integer value - Non Zero return value is an error.

6.2 VT1420GetSDRAMSize

Prototype	int VT1420GetSDRAMSize (CardPtr cardptr, uint32 *pnSize, int *pnBanks);
Description	This routine determines the amount of SDRAM that is with the selected DSP of the VT142x. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. The size of the SDRAM is loaded into the variable pnSize and the number of banks of SDRAM is loaded into the variable pnBanks.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 pnSize – Return value of memory size. int pnBanks – Return value for number of memory banks.
Return	Integer value - Non Zero return value is an error.

6.3 VT1420ReadMemToFile

Prototype	int VT1420ReadMemToFile (CardPtr cardptrchar *sFilename, uint32 nStart, uint32 nNumWords);
Description	This routine reads the number of words specified by the variable nNumWords from memory of the selected DSP starting at the address specified by the variable nStart and writes the data to a file specified by the variable sFilename. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. This routine can access any address in the DSP's memory space and it does so by properly setting the DSPP page register of the DSP.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name of the output file to which memory data is written. uint32 nStart – Starting address to read the memory from uint32 nNumWords – Number of words to read.
Return	Integer value - Non Zero return value is an error.

6.4 VT1420WriteMemFromFile

Prototype	int VT1420WriteMemFromFile (CardPtr cardptr, char *sFilename, uint32 nStartAddr);
Description	This routine fills the memory of the selected DSP with the values located in the file specified by the variable sFilename. The starting address is defined by the variable nStartAddr. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. This routine can access any address in the DSP's memory space and it does so by properly setting the DSPP page register of the DSP.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name for the input data file. An example file with the proper format can be obtained by using <i>VT1420ReadMemToFile</i> . uint32 nStartAddr – Starting address.
Return	Integer value - Non Zero return value is an error.

6.5 VT1420WriteMemFromDotHexFile

Prototype	int VT1420WriteMemFromDotHexFile (CardPtr cardptr, char *sFilename)
Description	This routine loads a HEX formatted executable into memory of the selected DSP from a file. The input file is specified by the variable sFilename. The DSP to load the executable into is determined by the <i>VT1420SetDSP</i> routine. This is an application load routine and is intended for internal RAM and SDRAM.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name for the input data file. The file of proper format is generated by a combination of CodeComposer tools, which generates *.out file, and <i>hex6x</i> utility, which converts *.out to *.hex, both from TI. See the <i>out2hex.bat</i> and the .out and corresponding .hex example files in the VT-1420 installation.
Return	Integer value - Non Zero return value is an error.

6.6 VT1420AddressWriteTest

Prototype	Int VT1420AddressWriteTest (CardPtr cardptr, uint32 nStartAddr, uint32 nSize);
Description	This routine performs a SDRAM memory test. This test is an address to data test. The memory address is used as the value to be written.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 nStartAddr – Starting address for the memory test. uint32 nSize – Amount of memory to be tested.
Return	Integer value - Non Zero return value is an error.

6.7 VT1420WalkOnes

Prototype	int VT1420WalkOnes (CardPtr cardptr, uint32 nStartAddr);
Description	This routine performs a walking ones memory test on the SDRAM starting at the address specified by the nStartAddr variable.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 nStartAddr – Starting address for the memory test.
Return	Integer value - Non Zero return value is an error.

6.8 VT1420RAMTest

Prototype	int VT1420RAMTest (CardPtr cardptr);
Description	This routine determines the SDRAM size and number of banks of the selected DSP, and then performs a <i>VT1420WalkOnes</i> routine followed by a <i>VT1420AddressWriteTest</i> routine for the entire SDRAM.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

7 DSP Flash Memory Routines

Prior to using all DSP Flash routines, the proper DSP must be selected by calling the routine *VT1420SetDSP*. Note: EMIF B must be configured prior to accessing the Flash (see *VT1420ConfigBoard*).

7.1 VT1420ResetFlash

Prototype	int VT1420ResetFlash (CardPtr cardptr);
Description	This routine resets the Flash memory of the selected DSP and puts the Flash controller into a known state. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

7.2 VT1420EraseFlash

Prototype	int VT1420EraseFlash (CardPtr cardptr);
Description	This routine erases the Flash of the selected DSP. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

7.3 VT1420EraseFlashSector

Prototype	int VT1420EraseFlashSector (CardPtr cardptr, int nSect);
Description	This routine erases the specified sector of the Flash memory of a selected DSP. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. int nSect – Sector to erase. Valid values are 1 through 18. Refer the VT-1420 User's Manual for information on Flash sectors.
Return	Integer value - Non Zero return value is an error.

7.4 VT1420GetFlashSize

Prototype	int VT1420GetFlashSize (CardPtr cardptr, int *pnSize);
Description	This routine returns the size of the Flash memory of the selected DSP. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. The pnSize variable is used to return the value.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. int pnSize – Return value containing the size of the Flash memory. Note: For Flash sizes greater than 1MB, the upper addresses are accessed via paging as described in the VT-1420 User's Manual.
Return	Integer value - Non Zero return value is an error.

7.5 VT1420GetFlashType

Prototype	int VT1420GetFlashType (CardPtr cardptr, uint32 *pnManID, uint32 *pnDevID);
Description	This routine returns the Device ID and Manufacturer's ID of the selected DSP's Flash memory. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint32 pnManID – Return value for the Manufacturer's ID. uint32 pnDevID – Return value for the Device ID.
Return	Integer value - Non Zero return value is an error.

7.6 VT1420FlashTest

Prototype	int VT1420FlashTest (CardPtr cardptr);
Description	This routine performs a read and write memory test on the Flash memory of the selected DSP. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. Note: During this test the Flash memory is erased and any data previously in the Flash memory will be lost.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board.
Return	Integer value - Non Zero return value is an error.

7.7 VT1420ReadFlash

Prototype	int VT1420ReadFlash (CardPtr cardptr, uint8 *pcBytes, int nNumToRead);
Description	This routine reads the values from the Flash Memory of the selected DSP and loads them into the variable pcBytes. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. The amount of data to read is defined by the nNumToRead variable.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint8 pcBytes – Return value containing the values read from the Flash memory. int nNumToRead – Defines the amount of data to read from the Flash memory.
Return	Integer value - Non Zero return value is an error.

7.8 VT1420ReadFlashToFile

Prototype	int VT1420ReadFlashToFile (CardPtr cardptr, char *sFilename);
Description	This routine reads the values from the Flash memory of the selected DSP and writes them to the output file defined by the variable sFilename. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name of the output file to which memory data is written.
Output	File containing memory values.
Return	Integer value - Non Zero return value is an error.

7.9 VT1420WriteFlash

Prototype	int VT1420WriteFlash (CardPtr cardptr, uint8 *pcBytes int nNumBytes, uint32 nStartAddr);
Description	This routine programs the Flash memory of the selected DSP with the values contained in the variable pcBytes. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. The amount of data to write is defined by the variable nNumBytes. The variable nStartAddr defines the starting address.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. uint8 *pcBytes – Contains the data to write to the Flash memory. int nNumBytes – Defines the amount of data to write to the Flash memory. uint32 nStartAddr – Starting address.
Return	Integer value - Non Zero return value is an error.

7.10VT1420WriteFlashFromFile

Prototype	VT1420WriteFlashFromFile (CardPtr cardptr, char *sFilename, uint32 nStartAddr)
Description	This routine fills the Flash memory of the selected DSP with the values contained in the input file. The selected DSP processor was specified using the <i>VT1420SetDSP</i> routine. The variable sFilename defines the input file. The starting address is defined by the variable nStartAddr.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name for the input data file. An example file with the proper format can be obtained by using <i>VT1420ReadFlashToFile</i> . uint32 nStartAddr – Starting address.
Return	Integer value - Non Zero return value is an error.

7.11 VT1420WriteFlashFromDotHexFile

Prototype	int VT1420WriteFlashFromDotHexFile (CardPtr cardptr, char *sFilename);
Description	This routine loads a HEX formatted executable into the Flash memory of a selected DSP from a file. The DSP to load the executable into is determined by the <i>VT1420SetDSP</i> routine. The input file is specified by the variable sFilename.
Arguments	CardPtr cardptr – Pointer to the base address of the selected VT142x DSP board. char sFilename – Directory path and file name for the input data file. The file of proper format is generated by a combination of CodeComposer tools, which generates *.out file, and <i>hex6x</i> utility, which converts *.out to *.hex, both from TI. See the <i>out2hex.bat</i> and the .out and corresponding .hex example files in the VT-1420 installation.
Return	Integer value - Non Zero return value is an error.

Appendix A – Related Documentation

Applicable documents

The publications listed below are included with the installation distribution, and may be referenced in this document..

Title	Document Number	Revision
VT-1420/1423 Data Sheet	VTI 590-002-039	Rev A
VT-1425/1426 Data Sheet		Rev A
VT-142x DSP PMC Module User's Manual	VTI 560-000-039	2.1 or greater
VT-1420 Command Line Interface (CLI) User's Manual	VTI 560-003-039	1.2 or greater

Appendix B – List of Acronyms, Abbreviations and Initializations

Address to Data – Memory Test that uses the memory address as data to write to that memory location.

API – Application Programming Interface

DSP – Digital Signal Processor

Flash Memory – Non-volatile programmable memory

LED – Light Emitting Diode

PCI – Peripheral Component Interconnect

PMC – PCI Mezzanine Card

RAM – Random Access Memory

SDRAM – Synchronous Dynamic Random Access Memory

SRM – Software Reference Manual

SRAM – Serial Read Only Memory

Walking Ones – Memory test that writes a data pattern of shifting ones and zeros across the memories.

CUSTOMER SUPPORT

Technical Assistance

The Evergreen Group's technical support staff is available to assist you with questions that you may have. Please contact us using one of the methods shown below.

Phone: (610) 871-1955 8AM to 5PM EST, Monday through Friday

Fax: (610) 680-3305

Email: support@evergreengrp.com

Website: <http://www.evergreengrp.com>

World Wide Web

The Evergreen Group maintains an active site on the World Wide Web. The site contains current information about the company and locations of sales offices, new and existing products, contacts for sales, service and technical support information. You can also send email to The Evergreen Group using the web site. Requests for sales, service, and technical support information will receive a prompt response.

Warranty Service

The API Host Library is included as is with no warranty or no implied warranty.

Life Support Policy

The Evergreen Group's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of The Evergreen Group. Refer to following for definitions of critical components and life support devices.

1. A critical component is any component of a life support device or system whose failure to perform can be expected to cause the failure of the life support device or system, affect its safety, or limit its effectiveness.
2. Life support devices or systems are devices or systems which support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.



Office Location:

The Evergreen Group Inc.

4460 Bachman Drive
Schnecksville, PA 18078

Tel: 610.871.1955
Fax 610.680.3305

Internet:

Web Site: <http://www.evergreengrp.com>
<http://www.pmcmodules.com>
E-Mail: info@evergreengrp.com