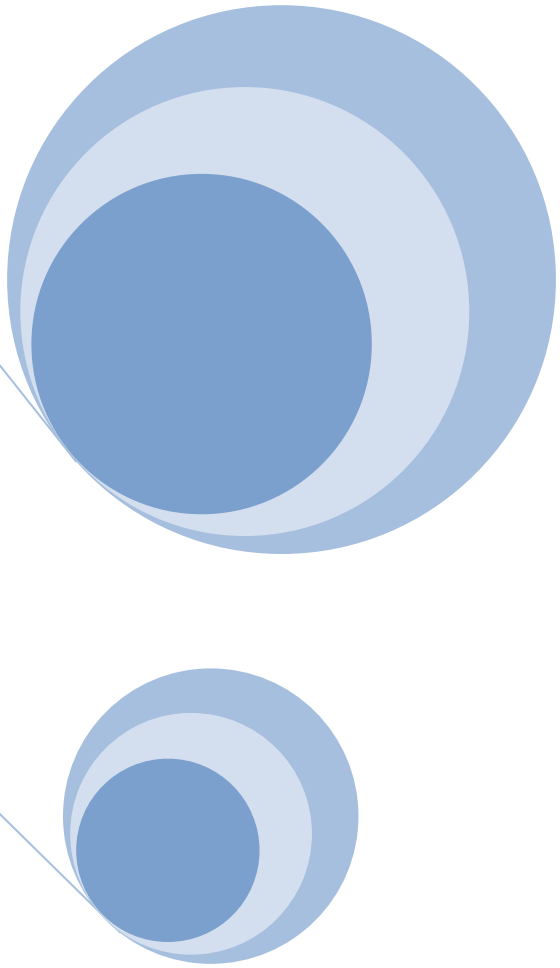


VisiSonics Digital Microphone Array (VDAM) System

This document provides instructions on use of the VisiSonics Digital Microphone Array System and associated software. License terms are also provided.

The VDAM system uses software and technology licensed from the University of Maryland by VisiSonics.

Version 1.6

June 28, 2016

Contents

1. List of Components Supplied

1 Set of VisiSonics Chainable Microphones (number will vary by purchase: 8, 16, 32, 64, etc)

1 Buffer board per 16 microphones

1 Controller Unit (FPGA)

1 VisiSonics Microphone Hub

1 USB 2.0-B to USB 2.0-A Cable

1 VDAM Power Supply

All Inter-Microphone cabling

2. Notices and Warnings

This device and software are supplied under license given at the end of this document. Use of the device is subject to agreement with the terms of this license.

The VDAM system is a delicate piece of electronics. It should not be shaken, dropped; or subject to stress, jerky motion, or extreme temperatures. The device storage and operating temperature is from 5 ^{o}C to 40 ^{o}C. Failure to handle with care or to maintain the device voids all warranties. The device contains no user serviceable components.

Should the device fail to function under normal operational conditions, VisiSonics will cover repairs at its cost (less shipping) during the warranty period, subject to condition that the device has been maintained properly. Repairs will be covered in subsequent years only if an appropriate service agreement is in place. Otherwise, repairs are covered under time and materials terms; contact VisiSonics for further details.

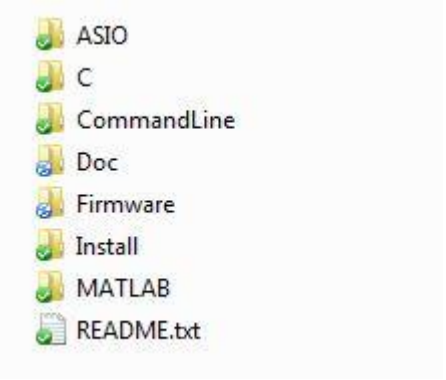
Warranties cover the microphones and hub only; they do not extend to the cables, or power supply. Some of these items are covered by their respective manufacturers' warranties.

3. Quick Start Guide and User Manual

The VisiSonics Digital Microphone Array (VDAM) consists of a number of microphones connected in parallel chains of 16 microphones with a small buffer board every 8 microphones. Each microphone is connected to its nearest neighbor by two cables. The 20 pin white flat flex cable transmits data. The black co-axial cable provides the sampling clock. The chains of 16 microphones are connected to the controller unit that controls the data collection. The VDAM system interfaces with a laptop computer through a standard USB 2.0 connection. Power is provided to the system through the VisiSonics' hub.

3.1 Downloading and Installing VDAM SDK and Driver Software

1. **Do not plug the microphone array hub board into the computer until the following software install procedure has been completed.**
2. With your purchase, VisiSonics shared a DropBox folder with the SDK and installer for the VDAM microphone array.
3. Do not run the Installer directly from DropBox
4. Copy the file VDAM_SDK.zip to your hard drive in a drive to which you have write permission, e.g. C:\Users\Yourname.
5. Unzip the Zip file in to the same directory
6. This will create a folder called "VDAM_SDK"
7. Open this folder
8. You should see the following 6 files in the folder (right)
9. Shut down all other programs running on your computer
10. Make sure you have administrator privileges on the computer. If you do not, please contact someone who can give you these privileges
11. From within the Install folder run FP-4.0.6-Setup-Win-x64.exe to install the driver.
12. Reboot the computer.
13. The command line executable for capturing data from the array can be found within the CommandLine folder.



3.2 Installing the Matlab toolbox

1. Open Matlab
2. Change the Matlab working directory to the MATLAB folder within the VDAM_SDK.
3. At the Matlab command prompt type "install.m". This will register the path and the documentation with Matlab.
4. Type "docsearch VDAM" at the command prompt to view the detailed toolbox documentation from within the Matlab help browser. Alternatively type doc and select the VDAM toolbox from the list of installed toolboxes .
    a. Note: If you are using Matlab 2013 or newer you may need to use the command
    `>>doc -classic`
    In order to see the supplemental VDAM documentation.

3.3 Wiring connections

The picture shows how the array will arrive with the exception that the hub board will not be connected to the chains. The foam can serve to hold the array for initial tests.

After removing the microphones from the packaging before your first run DOUBLE CHECK that the cables are still seated properly. Failure to do so can damage your array.

3.4 Hub Connections

Connect the hub board to the chains as shown to the right. The sliver side of the ribbon connector must be connected facing up. **Failure to ensure proper connection could damage the array.** Connect each of the chains to each of the ports on the hub board as shown.

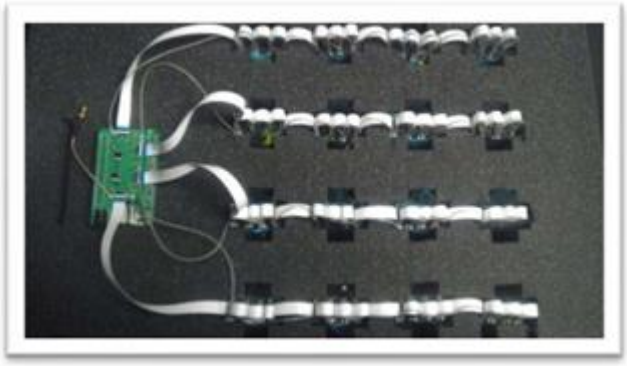


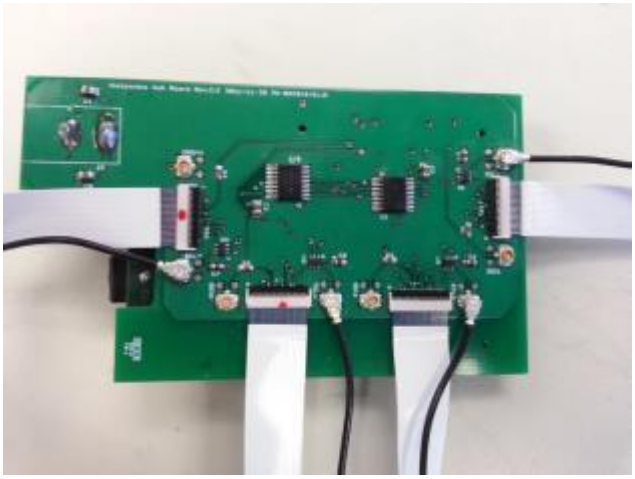
Figure 1 - Correct connection of the 4 chains to the hub board. Notice that the Silver sides of the ribbon connectors are face up.

Each chain will have one 20 pin flat flex and one co-axial to connect. The co-axial will connect to the micro co-axial connector to the right of it data port.

Flat Flex will connect to "J1-ToHub" side of first microphone in chain

3.5 Microphone Connections

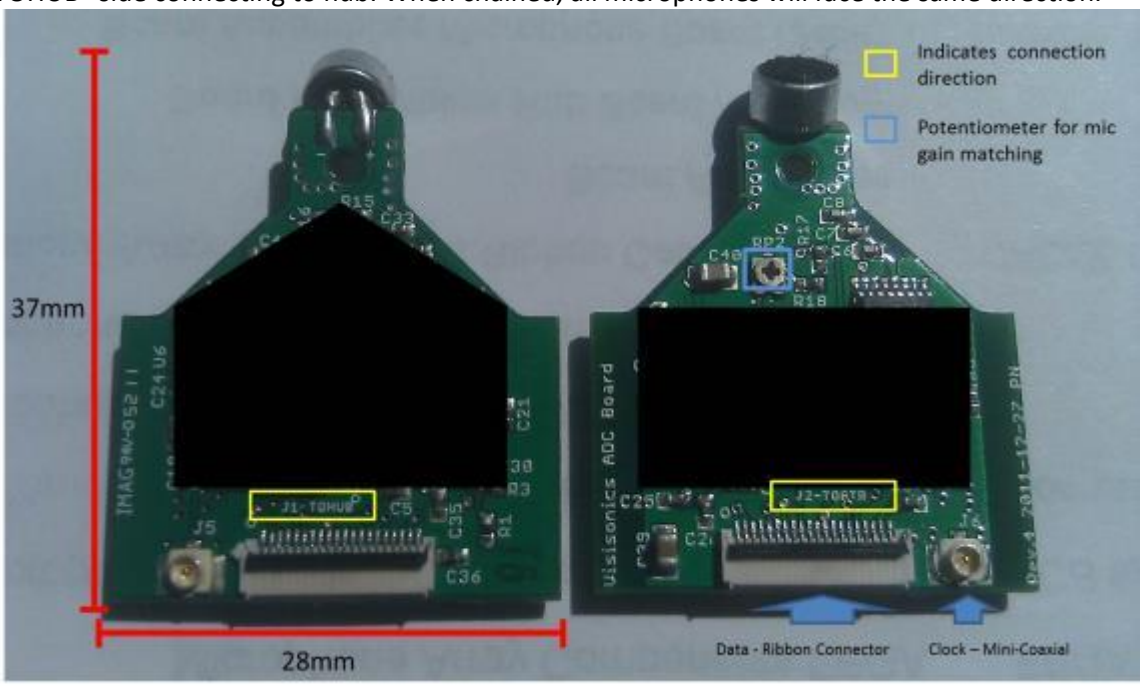
There are four connectors on the hub for the four microphone chains of up to 16. If you have greater than 8 microphones in a chain you will want to insert a buffer board in the middle of the chain to ensure good data transfer.



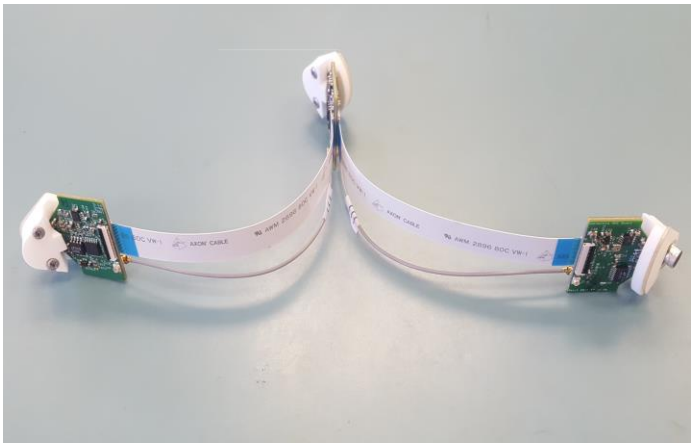
The ribbon cable connectors have a black locking lever on the back of the connector. To insert the ribbon cable, ensure that the black lever is pointing up. Insert the cable and press the lever down to lock. The silver side should be up. Take care to insert the ribbon cables straight to avoid damaging the cable and potential shorting during operation.

Connecting microphone to microphone, cables will link "J2-TOSTR" to "J1-TOHUB" connectors. Ensure that "J2-TOSTR" connects down the chain away from hub. First Microphone closest to hub will have "J1-

TOHUB" side connecting to hub. When chained, all microphones will face the same direction.



The microphones in the chain should never be connected or disconnected when the hub board is powered or connected to USB. Make sure that both the power cable and the USB cable are disconnected before removing or adding microphones to the hub or a chain.



Once the chains have all been connected to the hub board and the software has been successfully installed you can connect the USB cable and power cable to the hub board. You should receive a message that it is installing drivers for the xem3010 FPGA unit. Once this has completed you should be able to capture data from the array using the command line interface described below.

Please keep in mind that you are receiving board level internal array components. Standard ESD precautions should be taken when handling the circuit boards.

3.6 Capturing data from the array (Command Line)

1. To capture data from the array open a command line that has administrative privileges.
2. To do this go to the Windows Icon (usually on bottom left),
3. Go to All Programs
4. Go to Accessories
5. Right click on Command Prompt, and choose "Run as administrator"
6. Change directory "`CD`" to a directory where you wish to place the recorded wave files.
7. Type the following at the command line: `Stream4x16flow.exe`
8. It should print the following:
   a. `***Use: stream4X16 <bit file name> <output size. in Mb> < output file name>`
9. Each command line argument
   a. <bit file name> can be one of the following 3 bit files with absolute path
      i. "`C:\Windows\Starlink_v205.bit`" : use this for the lowest gain setting of the array
      ii. "`C:\Windows\Starlink_v206.bit`" : use this for the middle gain setting of the array
      iii. "`C:\Windows\Starlink_v207.bit`" : use this for the highest gain setting of the array
   b. <output size, in Mb> is the amount of data you want to record from the array in mega bytes
      i. This quantity has a minimum value of 32, and must be a multiple of 8 megabytes. The sampling rate of the array is 44.1K at 16bits of resolution and 64 channels.
      ii. The equation for the duration of the recorded file is as follows:

$$Duration(seconds) = \frac{< sizeMB > * 2^{20}}{(44100 * 2 * 64)}$$

   c. <output file name> is the prefix for the saved wave files
      i. The wave files will be stored as follows
         1. <prefix> -chain1-ch00.wav
         2. <prefix> -chain1-ch02.wav, etc…
10. Sixty-four (64) files spanning from chain1-ch00.wav to chain4-ch15.wav will be created. If files with the same prefix already exist in the folder they will be over-written.

Example:

To capture 128MB of data on the lowest gain setting from the array and store it in files with the prefix "testData" use the following command line in a folder of your choice.

"`stream4x16flow.exe c:\windows\starlink_v205.bit 128 testData`"

3.7 Capturing data from the array (Matlab)

We provide an object oriented interface to the array to help you build real-time applications in the

Matlab environment. In this section we will outline the use of this interface to capture data directly into the Matlab workspace. Full documentation can be found within the Matlab help browser by typing docsearch VDAM. Here we outline the key objects.

3.7.1 Important functions.

a. `micArray = Microphone_Array();`
   i. `This function creates a microphone array object in the matlab workspace. The user settable properties of this object are as follows`
      1. gainSetting = 1, this property can be 1, 2, or 3, corresponding to -15dB, 0dB, and +15dB gain settings in the array
      2. n_channels: 64, this indicates how many microphones you have connected in the current array
      3. block_size= 16384, this sets the amount of data per channel that will be returned each time the array is queried.
      4. bgbuf_size: 134217728, this variable indicates the internal array size where streaming data will be buffered. This is a circular buffer that will wrap around and overwrite data if the user does not access the data quickly enough using getDataBlock(). It is the application developer's responsibility to read data frames sufficiently quickly so that no data is lost. At the default value and assuming a 64 element microphone array approximately 50 seconds of data will be internally buffered before the write operation wraps.

b. `micArray.init_array()`, this function initializes the array based on the properties set.
c. `micArray.start()`, this function starts the streaming of data from the array to the internal big buffer.
d. `Data = micArray.getDataBlock()`, this function returns data from each channel organized sequentially. The size of data must be <1,n_channels*block_size>
e. `micArray.stop()`, this function stops the streaming of data into the internal buffer
f. `delete(micArray)`, destroy the object and clean resources.

3.7.2 Example Code

We provide several example scripts to demonstrate how to use this interface. Full documentation of each of these examples can be found in the Matlab help browser by typing "docsearch VDAM".

1) LevelCheck(gainSetting)
   a. This example script simply presents a sound level meter as a bar. The input parameter can be either 1,2, or 3 to set which gain setting should be used.
2) record_array_to_wav.m
   a. this is a simple example of how to record the array data to wav.

3) AudioImageExample.m
    a. This outlines a spherical acoustic image example of a sound source localizer based on the classic GCC-PHAT algorithm.
4) offline_16ch_planar_DNS_beam.m
    a. This script demonstrates an .an offline beamformer for a 16 element planar array.
5) online_16ch_planar_DNS_beam.m
    a. Example code for a realtime beamformer for a 16 element planar array
6) offline_8ch_linear_DNS_beam.m
    a. example code for an offline 8 channel linear array beamformer
7) online_8ch_linear_DNS_beam.m
    a. example code for an real-time 8 channel linear array beamformer

3.8 Capturing data from the array c api (dll)

We also provide a dll for direct access in programming languages such as C, C++. This section outlines the important functions in the dll.

```
// ------------------------------------------------------------
// _vs_aif_get_version -- initialize the library
// Input:  None
// Output: major / minor versions and build date (YYYYMMDD)
// Return: Error code
// ------------------------------------------------------------
int _vs_aif_get_version(int *_major_version,
    int *_minor_version, int *_build_date);
// ------------------------------------------------------------


// ------------------------------------------------------------
// _vs_aif_acquisition_open -- open the acquisition device
// Input:  _reserved1:     reserved, should be zero
//         _channel_count: number of channels
//                         (currently only 64 supported)
//         _block_size:    number of samples (per channel)
//                         in one data transfer block
//         _buffer_size:   size of background data buffer
//         _cfg_fname:     device configuration file name
//         _reserved2:     reserved, should be zero
// Output: _handle:        a handle to the open device
// Return: Error code
// ------------------------------------------------------------
int _vs_aif_acquisition_open(int *_handle,
    int _reserved1, int _channel_count, int _block_size,
    int _buffer_size, char *_cfg_fname, int _reserved2);
// ------------------------------------------------------------

// ------------------------------------------------------------
// _vs_aif_acquisition_start -- start the acquisition process
// Input:  _handle: a handle to open device
//                  (obtained from _vs_acquisition_open)
```

```
// Output: None
// Return: Error code
// ------------------------------------------------------------
int _vs_aif_acquisition_start(int _handle);
// ------------------------------------------------------------


// ------------------------------------------------------------
// _vs_aif_retrieve_data_block -- obtain acquired data
// Input:  _handle: a handle to open device
//                  (obtained from _vs_acquisition_open)
//         _data:   a pointer to the data buffer;
//                  the buffer should be allocated by caller;
//                  the buffer size should be at least
//                  _channel_count * _block_size * sizeof(int)
// Output: _data:   buffer filled with deinterleaved samples;
//                  i.e. _block_size samples of channel 0,
//                  then _block_size samples of channel 1,
//                  and so on until channel 63. Transfers
//                  are always done in _block_size units.
// Return: Error code
// ------------------------------------------------------------
// Note:   This function is non-blocking. If you call it and
//         there is not enough data yet to fully populate the
//         buffer, the function returns _VS_AIF_DATA_NOT_READY
//         error code. This is not a fatal error; it simply
//         means "try again later".
// ------------------------------------------------------------
int _vs_aif_retrieve_data_block(int _handle, int *_data);
// ------------------------------------------------------------


// ------------------------------------------------------------
// _vs_aif_acquisition_stop -- terminate the acquisition process
// Input:  _handle: a handle to open device
//                  (obtained from _vs_acquisition_open)
// Output: None
// Return: Error code
// ------------------------------------------------------------
int _vs_aif_acquisition_stop(int _handle);
// ------------------------------------------------------------


// ------------------------------------------------------------
// _vs_aif_acquisition_close -- close the acquisition device
// Input:  _handle: a handle to open device
//                  (obtained from _vs_acquisition_open)
// Output: None
// Return: Error code
// ------------------------------------------------------------
int _vs_aif_acquisition_close(int _handle);
// ------------------------------------------------------------
```

4. VisiSonics Technical Support Contact Information

VisiSonics Support Team prides itself on providing quality products and customer support. Should you have any questions or concerns on the set up or operation of the VRAP camera after reading this user guide please feel free to contact VisiSonics. Thank you for your purchase.

VisiSonics Corporation
Technical Support
387 Technology Dr. Suite 3107d
College Park, MD 20742
301-405-8907
info@visisonics.com

5. Copyrights and Intellectual Property

6. VisiSonics Corporation End User License Agreement

PLEASE READ CAREFULLY: THE USE OF THIS SOFTWARE IS SUBJECT TO THE SOFTWARE LICENSE TERMS OF VisiSonics Corporation, INC. BY YOUR USE OF THE SOFTWARE INCLUDED WITH THIS PRODUCT YOU AGREE TO THE LICENSE TERMS.

VisiSonics Corporation Software License Agreement

(Single User Application)

In return for acquiring a license to use the VisiSonics software, the related documentation, you agree to the following terms and conditions:

1. License. This Agreement grants you, the Licensee, a license to:

(a) Use the Software on a single computer system, which is not intended for use by more than five (5) users. The Software may be "in use" on only the computer for which it was initially installed. (Different license terms and fees are applicable for networked or multiple user applications.) As a specific condition of this license, you agree to use the Software in compliance with all applicable laws, including copyright laws, and that you will not copy, transmit, perform or distribute any audio or other content using the Software without obtaining all necessary licenses or permissions from the owner of the content.

2. Restrictions. You may not distribute copies of the Software to others (including distribution on the Internet) or electronically transfer the Software from one computer to another. You are not authorized to use any licensed codecs in the Software or the Software itself in real time broadcasting (terrestrial, satellite, cable or other media) or broadcasting via the Internet or other networks such as, but not limited to, intranets, etc. or in pay-audio or audio-on-demand applications. You may not post or otherwise make available the Software, or any portion thereof, in any form, on the Internet. You may not use the Software in a computer service business, including in time-sharing applications. The Software contains trade secrets and, in order to protect them, you may not decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form. YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, LOAN, RESELL FOR PROFIT, DISTRIBUTE, NETWORK OR CREATE DERIVATIVE WORKS BASED UPON THE

SOFTWARE OR ANY PART THEREOF.

3. Ownership of Software. As Licensee, you own the media upon which the software is recorded or fixed, but VisiSonics Corporation and its licensors retain title and ownership of the Software recorded on the original media and all subsequent copies of the Software, regardless of the form or media in which or on which the original and other copies may exist. This license is not a sale of the Software or any copy.

4. Confidentiality. You agree to maintain the Software in confidence and that you will not disclose the Software to any third party without the express written consent of VisiSonics Corporation. You further agree to take all reasonable precautions to preclude access of unauthorized persons to the Software.

5. Term. This license is effective until January 1, 2045, unless terminated earlier. You may terminate the license at any time by destroying the Software (including the related documentation) together with all copies or modifications in any form. VisiSonics Corporation will have the right to terminate our license immediately if you fail to comply with any term or condition of this Agreement. Upon any termination, including termination by you, you must destroy the Software (including the related documentation), together with all copies or modifications in any form.

6. Limited Warranty. VisiSonics Corporation and its Licensors warrant only that the Software will perform substantially in accordance with the accompanying written documentation for a period of thirty (30) days from the date of your Software purchase receipt. VISISONICS CORPORATION AND IT LICENSORS DO NOT AND CANNOT WARRANT THE RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. THE FOREGOING STATES THE SOLE AND EXCLUSIVE REMEDIES VisiSonics Corporation OR ITS LICENSORS WILL PROVIDE FOR BREACH OF WARRANTY. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, VisiSonics Corporation MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED, AS TO NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights, which vary from state to state.

The entire liability of VisiSonics Corporation and its licensors, and your exclusive remedy for a breach of this warranty, shall be: (a) the replacement of the Software not meeting the above limited warranty which is returned to VisiSonics Corporation; or (b) if VisiSonics Corporation or its distributor is unable to deliver replacement Software, you may

terminate this Agreement by returning the Software and your money will be refunded.

7. Limitation of Liability. IN NO EVENT WILL VISISONICS CORPORATION OR ITS LICENSORS BE LIABLE TO YOU FOR ANY INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR LOSS OF DATA, EVEN IF VISISONICS CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Some states do not allow the exclusion or limitation of special, incidental, or consequential damages, so the above limitation or exclusion may not apply to you.

8. Export. You acknowledge that the laws and regulations of the United States and other countries may restrict the export and re-export of the Software. You agree that you will not export or re-export the Software or documentation in any form in violation of applicable United States and foreign law.

9. Government Restricted Rights. The Software is subject to restricted rights as follows. If the Software is acquired under the terms of a GSA contract: use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract. If the Software is acquired under the terms of a DOD or civilian agency contract, use, duplication or disclosure by the Government is subject to the restrictions of this Agreement in accordance with 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors and 49 C.F.R. 227.7202-1 of the DOD FAR Supplement and its successors

10. General. You acknowledge that you have read this Agreement, understand it, and that by using the Software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between VisiSonics Corporation and you, and supersedes any proposal or prior agreement, oral or written, and any other communication between VisiSonics Corporation and you relating to the subject matter of this Agreement. No additional or any different terms will be enforceable against VisiSonics Corporation unless VisiSonics Corporation gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of VisiSonics Corporation. You assume full responsibility for the use of the Software and agree to use the Software legally and responsibly. This Agreement shall be governed by Maryland law, except as to copyright matters, which are covered by Federal law. This Agreement is deemed entered into at College Park, Maryland by both parties. Should any provision of this Agreement be declared unenforceable in any jurisdiction, then such provision shall be deemed severable from this Agreement and shall not affect the remainder hereof. All rights in the Software not specifically granted in this Agreement are reserved by VisiSonics Corporation and its licensors.

Should you have any questions concerning this Agreement, you may contact VisiSonics Corporation by writing to:

VisiSonics Corporation, Inc.
Legal Department
387 Technology Drive; Suite 3107a
College Park, MD 20742