DIGITAL VOICE SYSTEMS, INC.

The Speech Compression Specialists

USB-3000™/ USB-3003™

USB Vocoder Device

Version 1.7

June, 2015

User's Manual

USB-3000™/ USB-3003™
Vocoder Device

User's Manual

Version 1.7
June, 2015

Digital Voice Systems, Inc.
234 Littleton Road
Westford, MA 01886

(The most up to date version of the manual is always available at www.dvsinc.com)

USB-3000™
END USER License Agreement

*** Important Read Carefully ***

1. Preliminary Statements and Definitions

1.1 This nonexclusive end user product license agreement is a legal agreement between the customer (the END USER) and Digital Voice Systems, Inc. (DVSI) covering the terms and conditions under which DVSI's proprietary content (that may consist of and is not limited to software, hardware, documentation and other material) is licensed to the END USER as part of this PRODUCT.

a) The PRODUCT shall mean the Hardware, Software, Documentation and other materials that were provided by DVSI, either directly or indirectly through distributors or agents, to END USER as part of a sale, delivery or other transaction.

b) Hardware can be in the form of Integrated Circuits (such as Digital signal Processors) Circuit boards and electronics enclosed in a chassis. DVSI's AMBE-3000™ Vocoder Chip is an example of an Integrated Circuit.

c) Software can be in form of computer code, firmware masked into an IC or stored or embedded into ROM or RAM or Flash memory, or software stored on any media (such as CD-ROM, floppy disk, hard drive, solid-state memory or the Internet)

d) Documentation means written or electronic information, including user manuals, technical documents, training materials, specifications or diagrams, that pertain to or are delivered with the PRODUCT in any manner (including in print, on CD-ROM, or on-line).

1.2 DVSI has developed a number of voice coding methods and algorithms (the "Technology") which include DVSI's Advanced Multi-Band Excitation ("AMBE®") , AMBE+™, and AMBE+2™ voice coders. The Technology codes speech at low bit rates and may include error correction, echo cancellation and other auxiliary functions.

1.3 "DVSI Voice Compression Software" shall mean the voice coding Software that implements or embodies the Technology and is embedded into or otherwise provided with the PRODUCT.

1.4 "DVSI Voice Codec" shall mean the DVSI Voice Compression Software, any PRODUCT Hardware into which the DVSI Voice Compression Software is embedded or executed and any associated Documentation.

1.5 DVSI represents that it owns certain "Proprietary Rights" in the PRODUCT including patent rights, copyrights, trademarks and trade secrets. These rights include one or more of the following US Patents #5,630,011; #5,649,050; #5,701,390; #5,715,365; #5,754,974; #5,826,222; #5,870,405; #6,161,089; #6,199,037; #6,912,495; #7,634,399; #7,957,963; #7,970,606; #8,036,886; #8,200,497; #8,315,860 and #8,359,197; and under other US and foreign patents and patents pending. AMBE®, AMBE+™ and AMBE+2™ are trademarks of Digital Voice Systems, Inc.

1.6 "END USER" shall mean the person and/or organization to whom the DVSI Vocoder Product (software or hardware) was delivered or provided to as specified in the purchase order or other documentation. In the event that the END USER transfers his rights under this license to a third party as specified in Section 3.0, then this third party shall become an "END USER".

1.7 DVSI reserves the right to make modifications and other changes to its products and services at any time and to discontinue any product or service without notice.

2. License Granted

2.1 Subject to the conditions herein and upon initial use of the DVSI Product, DVSI hereby grants to END USER a non-exclusive, limited license to use the DVSI Voice Compression Software and Technology within the PRODUCT. No license is granted for any use of the DVSI Voice Compression Software or Technology on any other device or Hardware or in any manner other than within the original unmodified PRODUCT purchased from DVSI. No license is granted to copy or modify the DVSI Voice Compression Software or the PRODUCT either in whole or in part.

2.2 No license, right or interest in any trademark, trade name or service mark of DVSI is granted under this Agreement. END USER acknowledges that the PRODUCT may contain trade secrets of DVSI, including but not limited to the specific design, and associated interface information.

2.3 END USER shall not copy, extract, reverse engineer, disassemble, de-compile or otherwise reduce the DVSI Voice Compression Software to human-readable form. END USER shall not alter, duplicate, make copies of, create derivative works from, distribute, disclose, provide or otherwise make available to others, the DVSI Voice Compression Software and Technology and/or trade secrets contained within the PRODUCT in any form to any third party without the prior written consent of DVSI. The END USER shall implement reasonable security measures to protect such trade secrets.

2.4 This is a license, not a transfer of title, to the DVSI Voice Compression Software, Technology and Documentation, and DVSI retains ownership and title to all copies.

3. Transfer of License

3.1 The END USER shall have the right to transfer the rights under this Agreement to a third party by either (i) providing the third party with a copy of this Agreement or (ii) providing the third party with an agreement written by the END USER (hereinafter "END USER Agreement") so long as the END USER Agreement is approved in writing by DVSI prior to transfer of the PRODUCT. The END USER Agreement shall contain comparable provisions to those contained herein for protecting the Proprietary Information from disclosure by such third party. Third parties shall agree to accept all the terms and conditions under either Agreement or the END USER Agreement.

4. Term and Termination

4.1 This Agreement is effective upon initial delivery of the PRODUCT and shall remain in effect until terminated in accordance with this agreement.

4.2 This Agreement shall terminate automatically without notice from DVSI if END USER fails to comply with any of the material terms and conditions herein. END USER may terminate this Agreement at any time upon written notice to DVSI certifying that END USER has complied with the provisions of Section 3.

4.3 Upon termination of this Agreement for any reason, END USER shall: (i) return the PRODUCT and documentation purchased or acquired, or in Licensee's possession, to DVSI; (ii) have no further rights to any DVSI Software or the Technology without a separate written license from DVSI; (iii) discontinue all use of the PRODUCT;

All confidentiality obligations of Customer and all limitations of liability and disclaimers and restrictions of warranty shall survive termination of this Agreement. In addition, the provisions of the sections titled "U.S. Government End User Purchasers" and "General Terms Applicable to the Limited Warranty Statement and End User License" shall survive termination of this Agreement.

5. Payments

5.1 In consideration of the materials delivered as part of the Product, and in consideration of the license granted by DVSI for the PRODUCT, and in consideration of DVSI's performance of its obligations hereunder, the END USER agrees to pay to DVSI the fees as specified in DVSI's invoice. Payments of fees shall be received by DVSI prior to shipment of the PRODUCT.

6. Proprietary Notices

6.1 END USER shall maintain and not remove any copyright or proprietary notice on or in the PRODUCT.

6.2 Reproduction of non-proprietary information found in DVSI Users Manuals or data sheets is permissible only if the END USER reproduces without alteration, and includes all copyright and other proprietary notices, all associated warranties, conditions and limitations on all copies, in any form.

7. Proprietary Information

7.1 The parties agree that the PRODUCT shall be considered Proprietary Information.

7.2 Except as otherwise provided in this Agreement, END USER shall not use, disclose, make, or have made any copies of the Proprietary Information, in whole or in part, without the prior written consent of DVSI.

8. Limited Warranty

8.1 DVSI warrants the PRODUCT to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery. The date of delivery is set forth on the packaging material in which the Product is shipped. This limited warranty extends only to the Customer who is the original purchaser. If the PRODUCT is found to be defective and the condition is reported to DVSI, within the warranty period, DVSI may, at its option, repair, replace, or refund of the purchase price of the PRODUCT. DVSI may require return of the PRODUCT as a condition to the remedy.

Restrictions. This warranty does not apply if the Product (a) has been altered, (b) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by DVSI, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident;

8.2 Except as stated in Section 8.1, the PRODUCT is provided "as is" without warranty of any kind. DVSI does not warrant, guarantee or make any representations regarding the use, or the results of the use, of the PRODUCT with respect to its correctness, accuracy, reliability, speech quality or otherwise. The entire risk as to the results and performance of the PRODUCT is assumed by the END USER. After expiration of the warranty period, END USER, and not DVSI or its employees, assumes the entire cost of any servicing, repair, replacement, or correction of the PRODUCT.

8.3 DVSI represents that, to the best of its knowledge, it has the right to enter into this Agreement and to grant a license to use the PRODUCT to END USER.

8.4 Except as specifically set forth in this Section 8, DVSI makes no express or implied warranties including, without limitation, the warranties of merchantability or fitness for a particular purpose or arising from a course of dealing, usage or trade practice, with respect to the PRODUCT. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to END USER. No oral or written information or advice given by DVSI or its employees shall create a warranty or in any way increase the scope of this warranty and END USER may not rely on any such information or advice. The limited warranties under this Section 8 give END USER specific legal rights, and END USER may have other rights, which vary from state to state.

9. Limitation of Liability

The END USER agrees that the limitations of liability and disclaimers set forth herein will apply regardless of whether the END USER has accepted the product or service delivered by DVSI.

9.1 In no event shall DVSI be liable for any special, incidental, indirect or consequential damages resulting from the use or performance of the PRODUCT whether based on an action in contract, or for applications assistance, or product support, or tort (including negligence) or otherwise (including, without limitation, damages for loss of business revenue, profits, business interruption, and loss of business information or lost or damaged data), even if DVSI or any DVSI representative has been advised of the possibility of such damages.

9.2 Because some states or jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to END USER.

9.3 DVSI's maximum liability for damages arising under this Agreement shall be limited to 20% (twenty percent) of the fees paid by END USER for the particular PRODUCT that gave rise to the claim or that is the subject matter of, or is directly related to, the cause of action.

10. Taxes

10.1 All payments required under Section 4 or otherwise under this Agreement are exclusive of taxes and END USER agrees to bear and be responsible for the payment of all such taxes (except for taxes based upon DVSI's income) including, but not limited to, all sales, use, rental receipt, personal property or other taxes which may be levied or assessed in connection with this Agreement.

11. Export

11.1 United States export laws and regulations prohibit the exportation of certain products or technical data received from DVSI under this Agreement to certain countries except under a special validated license. Some of the restricted countries include: Libya, Cuba, North Korea, Iraq, Serbia, Taliban in Afghanistan, Sudan, Burma, and Iran. The END USER hereby gives its assurance to DVSI that it will not knowingly, unless prior authorization is obtained from the appropriate U.S. export authority, export or re-export, directly or indirectly to any of the restricted countries any products or technical data received from DVSI under this Agreement in violation of said United States Export Laws and Regulations. DVSI neither represents that a license is not required nor that, if required, it will be issued by the U.S. Department of Commerce. Licensee shall assume complete and sole responsibility for obtaining any licenses required for export purposes.

12. Governing Law

12.1 This Agreement is made under and shall be governed by and construed in accordance with the laws of the Commonwealth of Massachusetts, (USA), except that body of law governing conflicts of law. If any provision of this Agreement shall be held unenforceable by a court of competent jurisdiction, that provision shall be enforced to the maximum extent permissible, and the remaining provisions of this Agreement shall remain in full force and effect. This Agreement has been written in the English language, and the parties agree that the English version will govern.

Special Handling Instructions

To avoid damage from the accumulation of a static charge, industry standard electrostatic discharge precautions and procedures must be employed during handling and installation the USB-3000™.

Read Instructions and User's Manual – All of the safe handling and operating instructions should be read before integration of the USB-3000™ begins. Failure to exercise reasonable care and to follow all instructions and heed all warnings may result in injury to property or to individuals.

Retain Instructions - The handling and operating instructions should be retained for future reference.

Follow Instructions - All operating and use instructions should be followed.

Storage

To insure maximum shelf life in long term storage, USB-3000™ should be kept in an a static shield, moisture controlled package at <40°C and <90% Relative Humidity

Installation

Ventilation - The USB-3000™ unit should be situated so that its location or position does not interfere with proper ventilation and air circulation.

Heat - The USB-3000™ unit should be situated away from devices that could act as a heat source.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Overview

This manual provides information about the USB-3000™ and the USB-3003™. These two devices are identical except for the fact that the USB-3003™ can support three channels at one time. For simplicity, most of the document will refer to the device as the USB-300x™. The USB-3003™ will only be called out in Sections that pertain to details that involve three channels.

The USB-300X™ contains Digital Voice Systems' proprietary and patented Advanced Multi-Band Excitation AMBE® voice compression technology. The USB-300x™ provides the flexibility to operate at virtually any data rate from 2000 bps to 9600 bps. This variety of speech and/or FEC rates permits vocoder optimization within system requirements that leads to excellent voice quality with superior robustness to bit errors and acoustic background noise. This data rate flexibility makes the USB-300x™ a cost efficient design and development tool for high performance, low bandwidth voice communication applications. In addition, the USB-300x™ includes a number of advanced features such as Voice Activity Detection (VAD), adaptive Comfort Noise Insertion (CNI) and support for DTMF tones.

The AMBE® voice compression technology has been thoroughly evaluated and tested under various conditions. It has been implemented and field proven by a variety of manufacturers around the world. The true value of the AMBE® vocoder is in providing a 2-3x improvement in channel capacity while maintaining a high performance level. The AMBE® vocoder technology has been proven in critical applications such as digital mobile radio, satellite communication systems and in other wireless communication devices. The success of this vocoder technology has resulted in it being chosen by many mobile radio manufacturers including APCO Project 25 in North America and DMR and dPMR in Europe.

1.2. USB-3000™ Features

- ◊ Instant access to DVSI's AMBE® Vocoder Technology via USB connection.
- ◊ Encode and decode files to/from a PC through the USB interface.
- ◊ Can be designed into a real-time full-duplex communication system.
- ◊ Virtual Com port design allows for flexible implementation into a variety of configurations.
- ◊ The USB-300x™ offers high quality speech compression and FEC data rates that can be set from 2000 bps to 9600 bps.
- ◊ Control of the AMBE-3000™ Vocoder Chip capabilities such as Soft decision FEC, Voice Activity Detection (VAD), adaptive Comfort Noise Insertion (CNI) and DTMF tones.
- ◊ The USB device comes with sample control software and reference documentation.
- ◊ Compatible with Window XP and Windows 7.
- ◊ Create a multi-channel system with additional USB-3000™ devices.

Figure 1 USB-3000™ and the USB-3003™ P25

1.3. What's included with the USB-3000™

The USB-300x™ is a complete hardware package. The device comes with a CD that contains documentation and operation software. Documentation includes a USB-300x™ User's manual and an AMBE-3000™ User's Manual. The software on the CD is an executable program that can be run to encode or decode files. The source code for the program is provided so that users can use it as an example to create their own programs. The CD also contains sample speech and compressed speech files as described in Section 2.

1.4. Product Description

The USB-300x™ is ideal for encoding and decoding speech using a PC platform. Simply connect the USB-300x™ to a Windows based PC's USB interface and begin to configure vocoder rate and options then encode and decode files or process real time speech. The USB-300x™ can play a key role in the development of communication systems, including push-to-talk land mobile radio, satellite and wireless telephony. The USB-300x™ can be used to create multi-channel systems by connecting more than one USB-300x™ to a single PC.

The USB-300x™ incorporates a USB to serial UART Integrated Circuit Device manufactured by FTDI (P/N FT232R). This allows designers to utilize FTDI's off-the-shelf drivers (compatible with several operating systems) for application customization and flexibility. The USB-300x™ is configured to use the USB drivers offered by FTDI. Refer to section 2.4 USB-300x™ USB Driver Description for more details or visit the FTDI website at http://www.ftdichip.com/FTDrivers.htm for more information.

The USB-300x™ is available in five models allowing it to meet a variety of system requirements.

Interoperable DVSI Vocoders

Versions	USB-3000™ / USB-3003™	USB-3000™ P25 / USB-3003™ P25	USB-3000™ SAT
AMBE-2000™ Chip Rates	YES	YES	YES
AMBE-3000™ Chip Rates	YES	YES	YES
MotoTRBO	YES	YES	YES
DMR	YES	YES	YES
dPMR	YES	YES	YES
D-STAR	YES	YES	YES
NXDN	YES	YES	YES
APCO Project 25 Half Rate	YES	YES	YES
APCO Project 25 Full Rate	NO	YES	NO
TerreStar	NO	NO	YES
MexSat	NO	NO	YES
GlobalStar	NO	NO	YES

1.4.1. USB-3000™ and the USB-3003™

The USB-300x™ implements DVSI's patented AMBE+2™ Voice Compression Algorithm. It can operate at virtually any data rate from 2.0 to 9.6 kbps. The USB-300x™ has all the rates of the AMBE-3000™ Vocoder

chip including MotoTRBO, DMR, dPMR and NXDN. The USB-3003™ is identical to DVSI's standard USB-3000™ except that it supports three simultaneous channels.

1.4.2. USB-3000™ P25 and USB-3003™ P25 Versions

The USB-3000™ P25 includes DVSI's patented Enhanced Dual-Rate Vocoder technology that is fully interoperable with APCO Project 25 standard (TIA-102BABA). The USB-3000™ P25 provides quick and easy access to the APCO Project 25 Phase 1, 7200 bps "full-rate" vocoder plus the second-generation Phase 2, 3600 bps "half-rate" vocoder. Both vocoder rates can be set to run with or without Forward Error Correction. In addition to the APCO Project 25 rates, the USB-3000™ P25 also supports all of the 61 built in rates that are available on the standard version USB-3000™.

The USB-3000™ P25 is identical to DVSI's USB-3000™ except that it includes two additional custom rates that are not supported by the standard USB-3000™. Because of this difference, the USB-3000™ P25 reports a different product name and version string than the standard USB-3000™.

The USB-3003™ P25 is the same as the USB-3000™ P25 except it provides three channels of APCO Project 25 vocoder.

1.4.3. USB-3000™ SAT

The USB-3000™-SAT is identical to DVSI's USB-3000™ except that it supports two additional custom rates that are not supported by the standard USB-3000™. One rate is 2450 bps and is fully interoperable with the TerreStar satellite terrestrial mobile broadband network. The second rate is 4000 bps and is fully interoperable with the TerreStar / GlobalStar mobile satellite voice networks. Because of this difference, the USB-3000™ SAT reports a different product name and version string than the standard USB-3000. For complete details regarding the USB-3000™ SAT refer to the AMBE-3000™ User's Manual as well as, AMBE-3000™ SAT Version Vocoder Chip Description document.

2. Installation

2.1. PC Requirements

The USB-3000™ has been tested and run on PCs running Microsoft Windows XP and Windows 7 operating systems. The USB-3000™ uses the USB 2.0 connection for all communication, system setup, and file I/O.

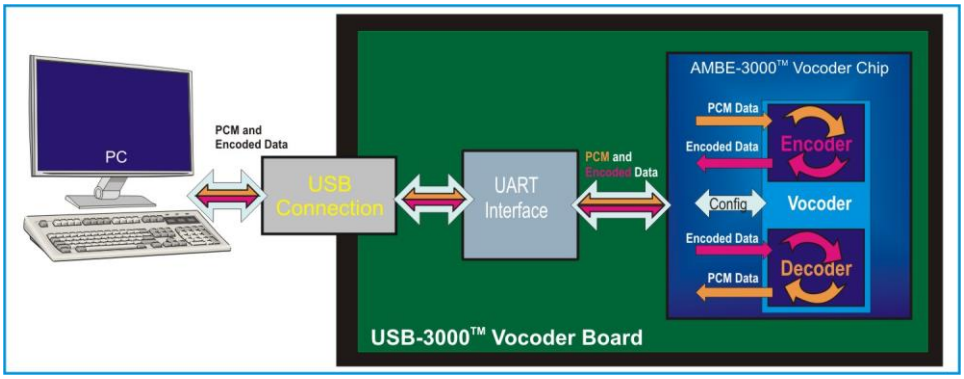

Figure 2 USB-300x™ Block Diagram

2.2. Installation Steps

Installation of the USB-300x™ USB consists of the following steps:

- Step 1. Copy the contents of the USB-300x™ Software CD to the PC as detailed in the next section
- Step 2. Connect the USB-300x™ to the PC
- Step 3. Install the drivers
- Step 4. Verify USB-300x™ Operation

2.3. Copying the USB-300x™ Software disk to the PC

Before installing the USB-300x™, it is recommended to copy the contents of the USB-300x™ CD to the PC. The USB-300x™ CD contains documentation, software and test vector files that are helpful in installation, operation and testing.

Windows XP and Windows 7 Operating System

- Step 1 Create a folder named **`C:\usb3000`** on the PC.
- Step 2 Copy the entire contents of the **`\usb3000`** directory from the CD provided with the USB-300x™ into this folder.
- Step 3 Go to the C:\usb3000 directory and unzip tv.zip file to C:\usb3000\tv. This compressed data file contains test vectors that may be used for vocoder testing.
- Step 4 Before continuing review all of the documentation in the C:\usb3000\Docs directory.

The USB-300x™ CD is setup with the following folder structure. For a description of the contents in the folders, refer to Section 3.3.

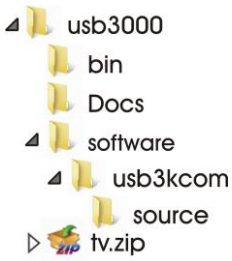

Figure 3 USB-300x™ - CD Folder structure

2.4. USB-300x™ USB Driver Description

The USB-300x™ incorporates a USB to serial UART Integrated Circuit Device manufactured by Future Technology Devices International Ltd. (FTDI) (P/N FT232R). This provides easy installation of the USB-300x™ allow users the flexibility of customization of an application. The download and installation procedures for FTDI's Generic VCP drivers and D2XX drivers are available on FTDI's website (http://www.ftdichip.com/FTDrivers.htm).

2.5. USB Driver Installation

To begin using the USB-300x™, connect it to an available USB port on the computer and install the required drivers. These drivers set up the USB-300x™ to communicate on the PC's serial COM port. The USB-300x™ has drivers available for both Windows-32 bit and Windows-64 bit operating systems.

Each USB-300x™ has a unique serial ID number. This allows more than one USB-300x™ to be connected to a PC at one time. However, the drivers must be installed the first time a new USB-300x™ is connected to the PC. Once the driver is installed for, a specific USB-300x™ Windows will automatically re-load the driver each time it is re-connected.

The USB-300x™ requires that two drivers be installed. The USB-300x™ uses standard drivers from FTDI. These drivers get automatically installed by Windows when a new USB device is connected to the PC.

2.5.1. Driver installation procedure for Windows 7.

Connect the USB-300x™ to an available USB port on the PC. Windows 7 indicates it is beginning to install the driver.

Figure 4 Installing driver software indicator

Windows 7 begins driver installation. Windows 7 then finishes installing the driver for the COM port and displays the software for this device has been successfully installed.

Click "X" to close the balloon.

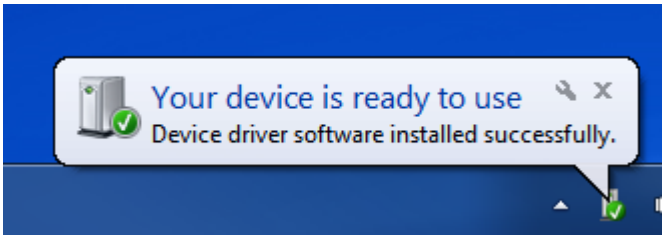

Figure 5 USB-300x™ drivers are installed and ready to use

After drivers are installed, it is important to determine which COM port has been assigned to the device. To do this use Windows Device Manager. To open Windows Device Manager Click "Start", click "Run", and then type "devmgmt.msc" (without the quotation marks). Alternatively, open the Device Manager (located in "Control Panel\System" then select the "Hardware" tab and click "Device Manger") and select "View > Devices by Type", the USB-300x USB device should appear under Ports (COM & LPT)" as USB Serial Port (COMXX) where "XX" is the port number of the USB interface.

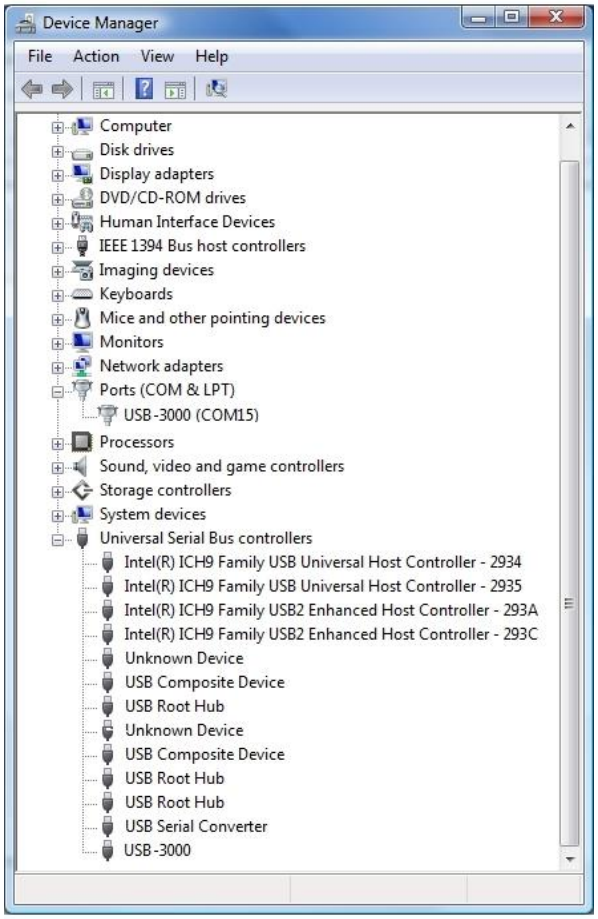

Figure 6 Windows Device Manager showing the USB-300x USB on COM 15

Note: Write down the Com Port that is being used for the USB Driver, this value will be required to run the USB-300x USB control program. In Figure 6 the COM port is shown as COM15.

NOTE: If the Device is shown with a yellow exclamation point then the USB driver is not completely installed. To fix this, uninstall the Device, disconnect the USB-3000™ from the PC's USB port and then reconnect the USB-3000™ to the PC's USB port and allow MS Windows to find new hardware. Then follow the instructions to reinstall the device.

2.6. USB COM rate settings

The serial interface supports asynchronous communication of both speech data and channel data using the standard non-return-to-zero (NRZ) format. The UART data interface uses a packet structure that is detailed in the AMBE-3000™ Vocoder Chip User's Manual.

Each serial word transmitted or received uses 8 data bits, no parity bits, and one stop bit. The serial port operates at baud rates from 28800 up to 460,800 baud. See Table 1 UART Baud Rates for available rates.

COM Port Baud Rate (baud)	Switch Position 1 (Not Used)	Switch Position 2	Switch Position 3	Switch Position 4
28,800	OFF	ON	ON	ON
57,600	OFF	OFF	ON	ON
115,200	OFF	ON	OFF	ON
230,400	OFF	OFF	OFF	ON
460,800	OFF	ON	ON	OFF
921,600*	OFF	OFF	ON	OFF

Table 1 UART Baud Rates

*Note: The 921,600 baud rate will not work with the USB-3000™

For maximum throughput DVSI strongly recommends using the default COM Port Baud rate settings as follows:

USB-3000™ = 460,800 baud
USB-3003™ = 921,600 baud

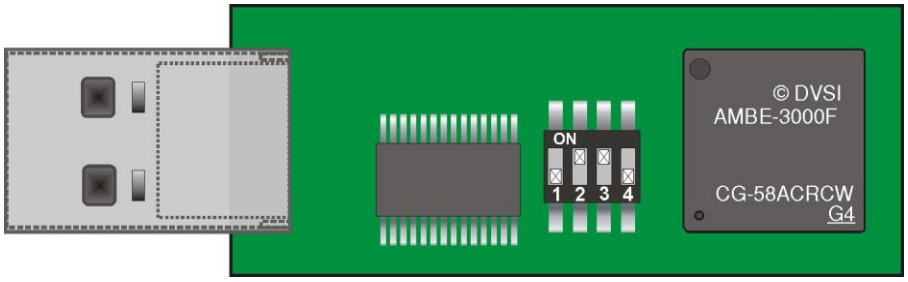

Figure 7 USB-3000™ Default Switch settings (460,800 Baud)

With the USB-3000™ oriented as shown in Figure 7 USB-3000™ Default Switch settings (460,800 Baud) - OFF is in the down Position and ON is in the UP position

The USB-3003™ default switch settings are OFF, OFF, ON, OFF as shown in Table 1 UART Baud Rates

2.7. Verifying USB-300x™ Com Port Settings

After the drivers are installed, it is good practice to verify the COM port settings. To do this,

Open up Device Manager
Select Ports (COM& LPT)
Right click on USB-3000 Serial Port and select Properties
The Properties window is shown in Figure 8 USB-300x Serial Port Properties Window

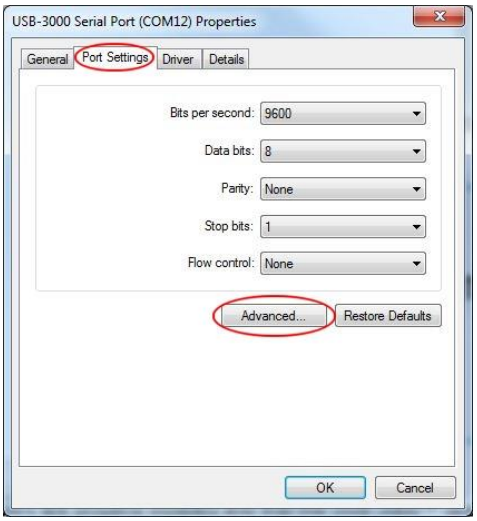

Figure 8 USB-300x Serial Port Properties Window

Select Port Settings TAB
Select Advanced Button
The window shown in Figure 9 COM Port Advanced Settings appears
Verify the setting in this window.

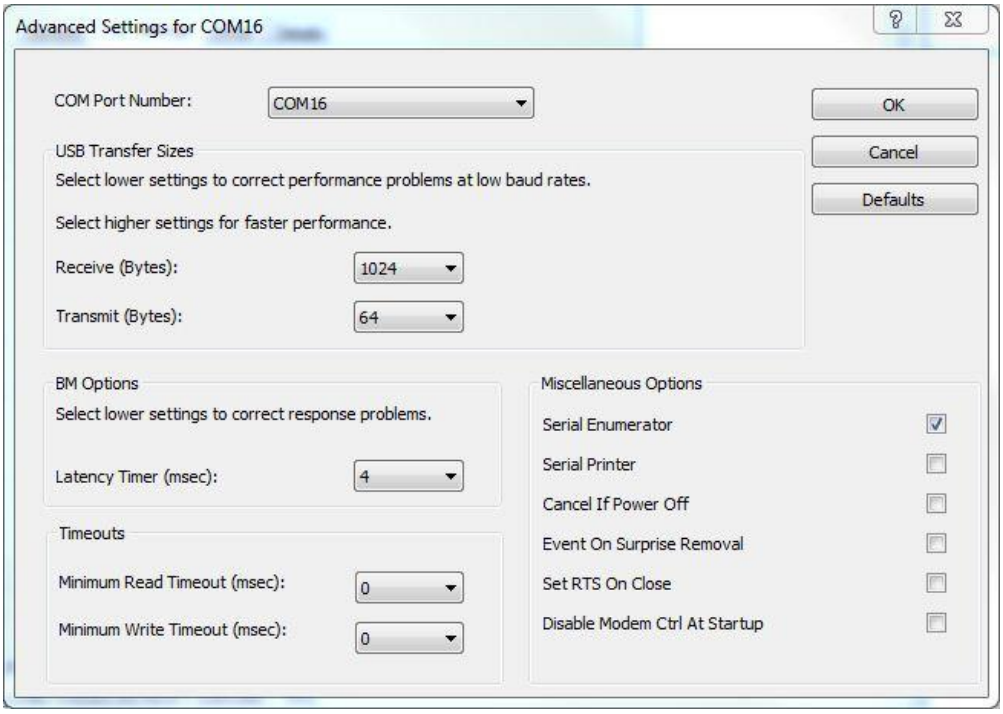

Figure 9 COM Port Advanced Settings

2.8. Verifying USB-300x™ Installation of drivers

A simple test to verify the drivers are properly installed and that the USB-300x™ operates is to use the **`usb3kverify.bat`** batch or the **`usb3kversion.bat`** files that are included in the **`usb3000/bin`** directory. Before either bat file can run the USB-300™ USB must be set up with the correct COM port and Baud rate. In

addition, for the **usb3kverify.bat** file the test vectors (**tv.zip** file) must be unzipped to the /usb3000/tv directory. To set up the USB-300x™ run **usb3ksetup.bat** (see section bin folder contents) as follows:

usb3ksetup.bat [no options]

the response will be

```
Enter COM Port:
```

This is the port the USB-300x™ is using (to find what port it is on see section USB COM rate settings). As an example to set the com port to com number 16 enter the com port as follows:

```
COM16
```

the response will be

```
Enter Baud Rate:
```

This is the rate the USB-300x™ with communicate over the com port see section USB COM rate settings. The default factory settings are set to a value of 460800 baud, so enter the rate as follows:

```
460800
```

the response will send you back to the command line.

Check to be sure the tv.zip file is unzipped to the **/usb3000/tv** directory. (See Section tv.zip file contents)
Now run **usb3kverify.bat** or **usb3kversion.bat** (See Section bin folder contents)

3. Operation

3.1. Overview

The USB-300x™ is setup, controlled and operated through the PC's USB interface. The USB interface is the physical link that establishes a communication connection where the PC sends data packets (speech, channel, or control) directly to the vocoder and gets either compressed speech, PCM packets or Response packets in return.

There are two control executable programs that are available to use with the USB-300x™. Both programs are run from a Command Prompt window using command line instructions. The usb3k_client32.exe program is used in conjunction with the dvsiserver32.exe program. Also included, is the **`usb3kcom.exe`**. This is an older software, provided for legacy system users but can also be used with newer models.

3.2. USB-300x™ Default Settings

The USB-300x™ allows users to configure features of the AMBE-3000™ Vocoder Chip. When the USB-300x™ is plugged into the USB port USB-300x™ is setup to the as follows:

Discontinuous Transmission Enable (DTX) = Disabled
Echo Canceller (EC) = Disabled
Skew Control = Disabled
Noise Suppression = Enabled
Companding = Disabled

Echo Suppressor = Disabled

RateIndex = 0 (2400 bps with 0 bps FEC)
Parity Bit = Enabled

For description of these special functions, see Section 4.4 of the AMBE-3000™ Vocoder Chip User's Manual.

These are the initial settings of the USB-300x™ and are set whenever the device is plugged into a PC or reset through a power cycle.

3.3. USB-300x™ Software CD Description

3.3.1. bin folder contents


bin

The bin folder contains various executables that can be used to run the AMBE-3000™ vocoder and to help in testing various files and formats. All of the executables and batch file are run in a DOS command prompt window.

`readme.txt`
This file describes the command line syntax for each of the .exe and .bat contained in the \bin directory.

`usb3ksetup.bat`

This batch file is used to specify the COM port and baud-rate for the USB-300x connection. The user is required to enter the COM port and then the baud rate. The batch file sets up environment variables on the PC using the user entered information. It also adds the \bin directory to the path on the PC. The command line syntax is as follows:

```
usb3ksetup.bat [no options]
```

the response will be

```
Enter COM Port:
```

This is the port the USB-300x™ is using to find what port it is on see section USB COM rate settings. As an example to set the com port to com number 16 enter the com port as follows:

```
COM16
```

the response will be

```
Enter Baud Rate:
```

This is the rate the USB-300x™ with communicate over the com port see section USB COM rate settings. As an example to set the baud rate to the default value of 460800 baud, enter the rate as follows:

```
460800
```

the response will send you back to the command line.

`usb3kverify.bat`

The usb3kverify.bat runs a set of tests used by DVSI to determine that the board is working correctly prior to shipping. This batch file can be used to verify that the hardware has been set up correctly. Before running this batch file be sure to have unzipped the tv.zip file to the /usb3000/tv folder.

```
usb3kverify.bat [no options]
```

`usb3kversion.bat`

The usb3kversion.bat prints to screen the version information of the AMBE-3000™ vocoder chip in the USB-300x™. This batch file can be also used to verify that the hardware has been set up correctly.

```
usb3kversion.bat [no options]
```

the response for the USB-300x™ version will be similar to the following:

```
C: \usb3000\bin>usb3kcom port COM16 460800 -version
   ******************************************************
   DVSI CONFIDENTIAL PROPRIETARY

   DVSI AMBE/AMBE+/AMBE+2 Speech Coder
   USB-3000 USB3KCOM Version 1.0.0, September 17, 2010


   (C) Copyright, Digital Voice Systems, Inc., 2010
   All Rights Reserved

   Notice: This software is protected by US and foreign
   patents (including US #5,226,084, #5,247,579
   #5,491,772, #5,517,511) and patents pending.
   Any use of this software requires a separate
   written license from DVSI.

   AMBE, AMBE+, and AMBE+2 are a trademarks of
   Digital Voice Systems, Inc.
   ******************************************************
   Product=<AMBE3000F>,Version=<V120.E100.XXXX.C106.G514.R0
   08.A0030608.C0020208>
```

Note that the batch file, `usb3kversion.bat` reports the product name and version string for the USB-3000™.

The response for the USB-300x™ version will include something similar to the following as the last line printed to screen when this command line is used:

```
C: \usb3000\bin>usb3kcom port COM16 460800 -version

Product=<AMBE3000R>,Version=<V130.E100.XXXX.C106.G514.R10.A0030608.C0020208>
```

The response for the USB-3003™ versions will include the something similar to following as the last line printed to screen when this command line is used:

```
C: \usb3000\bin>usb3kcom port COM16 460800 -version

Product=<AMBE3003F>,Version=<V130.E100.XXXX.C106.G514.R10.A0030608.C0020208>
```

The response for the USB-3000™ P25 version will include something similar to the following as the last line printed to screen when this command line is used:

```
C: \usb3000\bin>usb3kcom port COM16 460800 -version

Product=<AMBE3000P25F>,Version=<V130.E100.XXXX.C106.G514.R10.A0030608.C0020208>
```

The response for the USB-3003™ P25 version will include something similar to the following as the last line printed to screen when this command line is used:

```
C: \usb3000\bin>usb3kcom port COM16 460800 -version

Product=<AMBE3003P25F>,Version=<V130.E100.XXXX.C106.G514.R10.A0030608.C0020208>
```

The response for the USB-3000™ SAT versions will include something similar to the following as the last line printed to screen when this command line is used:

```
C: \usb3000\bin>usb3kcom port COM16 460800 -version

Product=<AMBE3000SATF>,Version=<V130.E100.XXXX.C106.G514.R10.A0030608.C0020208>
```

`a2lin.exe`

The a2lin.exe will convert an 8-bit a-law audio file <file_in> to a 16 bit linear pcm file <file_out>. The command line syntax is as follows:

```
a2lin.exe <file_in> <file_out>
```

`lin2a.exe`

The lin2a.exe will convert a 16 bit linear PCM file <file_in> to an 8-bit a-law audio file <file_out>. The command line syntax is as follows:

```
lin2a.exe <file_in> <file_out>
```

`mu2lin.exe`

The mu2lin.exe will convert an 8-bit u-law audio file <file_in> to a 16 bit linear pcm file <file_out>. The command line syntax is as follows:

```
mu2lin.exe <file_in> <file_out>
```

`lin2mu.exe`

The lin2mu.exe will convert a 16 bit linear PCM file <file_in> to an 8-bit u-law audio file <file_out>. The command line syntax is as follows:

```
lin2mu.exe <file_in> <file_out>
```

`compare.bat`

The compare.bat will compare two files to each other. This is a great tool to verify two files are bit exact. The command line syntax is as follows:

```
compare.bat <file1> <file2>
```

if the files are the same, the response will be a blank line

if the files are different, the response will be

```
Files are Different
```

`usb3kpacket.bat`

usb3kpacket.bat runs extensive packet mode tests. It processes all the test vectors included on the CD, for every rate (0-61). Each file is processed using linear PCM samples plus A-law and u-law companding. Each file is processed with DTX disabled and again with DTX enabled.

The command line syntax is as follows:

```
usb3kpacket.bat [no options]
```

`usb3kpacketsub.bat`

The usb3kpacketsub.bat is a batch file that is called when usb3kpacket.bat is run. This bat file is not intended to run independently.

`usb3kvectors.bat`

The usb3kvectors.bat is a batch file that is called when usb3kpacket.bat is run. This bat file is not intended to run independently.

`usb3krates.bat`

The usb3krates.bat is a batch file that is called when usb3kpacket.bat is run. This bat file is not intended to run independently.

`usb3kerr.bat`

The usb3kerr.bat file runs packet mode tests with bit errors for both hard-decision and soft-decision. It decodes all the bit error test vectors included on the CD, every rate (0-61).

The command line syntax is as follows:

```
usb3kerr.bat [no options]
```

`usb3kerrsub.bat`

The usb3kerrsub.bat is a batch file that is called when usb3kerr.bat is run. This bat file is not intended to run independently.

`usb3k_client32.exe and dvsiserver32.exe`

(For all USB-3000™ models)
The usb3k_client32.exe and dvsiserver32.exe are the recommended programs to be used to process the test vectors. The command line syntax is described in section 3.7.

`usb3kcom.exe`
The usb3kcom does communicate with both the USB-3000™ and the USB-3003™ but DVSI strongly suggests using the client / server model as described above.

For USB-3000™ P25 and USB-3003-P25 models only
`p25.bat [port]`
The `p25.bat file` runs the usb3kcom.exe program with custom rates words in the command line in order to perform P25 mode tests. "port" is the desired COM port, (for example, COM6). This is supported by the USB-3000™ P25 and USB-3003-P25 versions only.

NOTE: Test vectors are different for each of the USB3000™ versions.

3.3.2. Docs folder contents

The Docs folder contains the up to date manuals for the USB-3000™ and the AMBE-3000™ Vocoder Chip.

`USB-3000™ User's Manual (USB_3000_Manual.pdf)`
`AMBE-3000™ Vocoder Chip User's Manual (AMBE-3000_Manual.pdf)`

3.3.3. Software folder contents

The Source folder contains all of the source code required to build the USB-300x™ executable files. Review of this code can be beneficial in writing customized programs the fit specific needs.

3.3.4. tv.zip file contents

NOTE: Test vectors are different for each of the USB-300x™ versions.

The tv.zip file needs to be unzipped into a directory under usb3000 called tv. To do this use the following steps:

- Step 1 Copy the zip file tv.zip from the CD provided with the USB-3000™ into to the usb3000 directory.
- Step 2 Go to the C:\usb3000 directory and unzip tv.zip file to C:\usb3000\tv. This compressed data file contains test vectors that are used by the bat files in the /usb300/bin directory and may be used for vocoder testing.

The **`tv`** directory contains 9 **Original speech files** and 62 directories. The 62 directories contain **encoded files** and **processed** (encoded/decoded) **files** of the nine original speech files. The name of each directory includes the rate index number used to run the files that are found in the directory.

For example,
directory **r39** contains data files that were run using the rate index 39, which is a data rate of 3600bps.

where as,
directory **r42** contains data files that were run using the rate index 42, which is a data rate of 4800bps.

Original Speech files
The original speech files are found in the tv directory only. They are in different formats and have a **.pcm** extension in the file name, as shown below.
<*file name*>.pcm -- (pcm format audio file 16 bit audio file sampled at 8kHz.)
<*file name*>.pcma -- (a-law format audio file 8 bit audio file sampled at 8kHz.)
<*file name*>.pcmu -- (μ-law format audio file 8 bit audio file sampled at 8kHz.)

Encoded Files
Encoded Files are files that have been encoded using the USB-3000™. They are located in each of **r<rate_index>** subdirectories of the tv directory. Encoded files are indicated with a **.bit** extension in the file name as shown below
<*file name*>.bit -- (encoded data file from a pcm audio format file.)
<*file name*>.bita -- (encoded data file from an a-law audio format file.)
<*file name*>.bitu -- (encoded data file from a μ-law audio format file.)

The encoder data rate is indicated by the **rate index value** that is part of the directory name. Rate index values are referenced in the Table 18 Standard Rate Table for AMBE-3000™ Vocoder Chip **(by index number)**.

For example,
the file **clean.bit** in directory **r39** is a compressed data file encoded at rate index 39, which is a data rate of 3600bps.
where as,
the file **clean.bit** in directory **r42** is a compressed data file encoded at rate index 42, which is a data rate of 4800bps.

Processed files (Encoded / Decoded)
Processed files are files that have been Encoded / Decoded through the USB-3000™. They are located in each of **r<rate_index>** subdirectories of the tv directory. Processed files are indicated with a **.pcm** extension in the file name as shown below
<*file name*>.pcm -- (encoded/decoded data file from a pcm audio format file.)
<*file name*>.pcma -- (encoded/decoded data file from an a-law audio format file.)
<*file name*>.pcmu -- (encoded/decoded data file from a μ-law audio format file.)

The encoder/decoder data rate is indicated by the **rate index value** that is part of the directory name. Rate index values are referenced in the Table 18 Standard Rate Table for AMBE-3000™ Vocoder Chip (by index number).

For example,
The file **clean.pcm** in directory **r39** is a pcm file that was encoded/decoded at rate index 39, which is a data rate of 3600bps.
where as,
the file **clean.pcm** in directory **r42** is a pcm file that was encoded/decoded at rate index 42, which is a data rate of 4800bps.

3.4. usb3k_client and dvsiserver32.exe Program Description

The client/server (usb3k_client)/(dvsiserver32.exe) programs provide the best approach to interfacing with the USB-3000™ and USB-3003™. The use of the usb3k_client and dvsiserver32.exe programs eliminate the need for the COM port and baud rate to be included as part of the command line options. These programs help streamline the work whether it is dealing with a single channel or multiple channels at once.

The DVSI server program (dvsiserver32.exe) is used to establish a link between the PC and connected USB devices. This program must be running before starting the usb3k_client32.exe program. To run dvsiserver32.exe, double click on the executable. When the program starts, it will poll the USB ports to find any connected USB devices each one will then be listed in the window opened by the program. For information about each of the connected device just hover the mouse over the device serial number in the window.

Dvsiserver32.exe effectively translates the D2XX driver interface of the USB into a message pipe. To communicate with any device, a client program simply opens a message pipe to the device and sends packets through the message pipe. The server can establish a connection to many devices and can expose these devices to clients upon request. The server takes care of all the hardware details such as setting up the baud rate, flow control, timeouts, etc.

The usb3k_client32.exe program is written entirely in C to facilitate easy modification by customers. The software is compact, yet includes all necessary code to arrange the channel packet protocol and communicate with the USB through the dvsiserver.exe executable program. The source code for the usbclient.exe program is provided as an example to assist in the creation custom programs. The USB can be set up and controlled using a command prompt window and the command line interface program "usb3k_client32.exe.

3.5. Running the usb3k_client32.exe and dvsiserver32.exe Programs

Before starting either program an USB device must be connected to the PC. The dvsiserver.exe program (dvsiserver.exe) must be started before the usb3k_client32.exe program by simply clicking on the executable. See Figure 10 dvsiserver.exe program. After the dvsiserver program has started the usb3k_client32.exe program can be run by opening a command prompt window and changing to the directory (C:/usb3000/bin) that has the usb3k_client32.exe program file in it and typing in the desired command. Alternatively the directory C:/usb3000/bin may be added to your "PATH" environment variable.

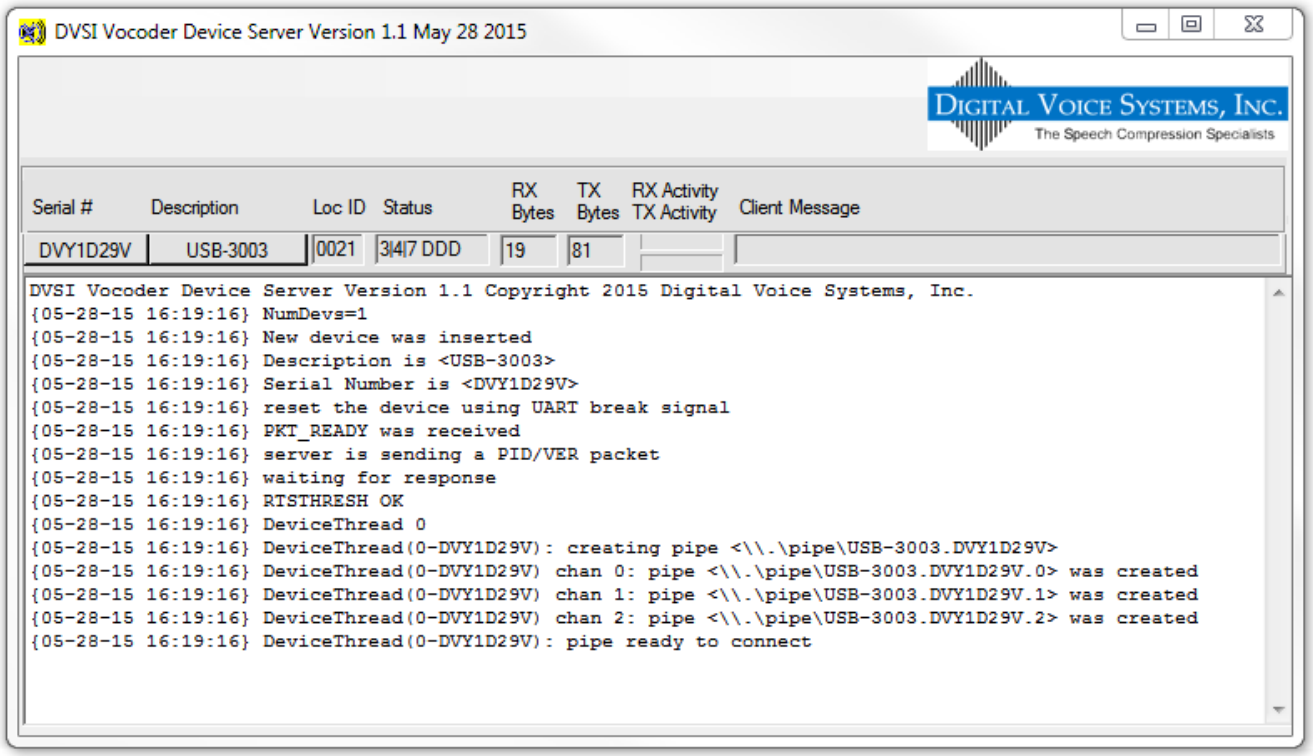

Figure 10 dvsiserver.exe program

3.6. usb3k_client32.exe Command line Structure

usb3k_client32.exe supports the following generic command line formats.

usb3k_client32.exe -enc [options] [input file name] [output file name]

usb3k_client32.exe -dec [options] [input file name] [output file name]

usb3k_client32.exe –version

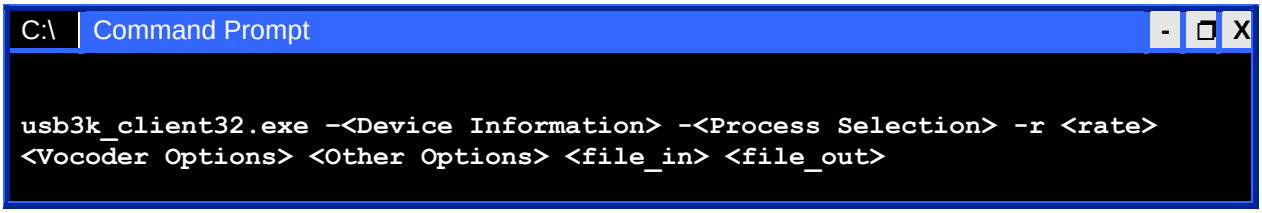

Figure 11 Command line structure

where:

-< Device Information>
indicates which device the command is controlling. Device information options are shown in Table 2 Device Information Options. This is useful in cases where multiple USB-300x™s are attached to the computer, but it is desired to address a particular USB-300x™. Note that the values for the device "Description" (-d); "Location ID" (-l); "Serial Number" (-n) and the "Product ID Name" (-p) for all connected USB-300x™s are displayed within the dvsiserver32.exe window. If this option is not specified, then usb3k_client addresses any available USB-300x™.

Option	Description	Example
-d [description]	Device description ID stored in the FTDI chip.	`-d USB-3000`
-l [Loc ID]	Device location ID is dependent upon which USB port the device is connected to.	`-l 0022`
-n [serial number]	Device Serial Number stored in the FTDI chip.	`-n FTXYH4MG`
-p [prodid name]	Product ID Name stored in the AMBE300x chip.	`-p AMBE3003F`

Table 2 Device Information Options

-< Process Selection >
sets the USB-3000™ to encode or decode or both:

Option	Description
-enc	with this selection, the USB-3000 USB encodes speech packets from the <file_in>. The USB-3000™ will then respond back to the PC with Channel packets where the channel data is stripped from the packet and written to <file_out>. When complete, the USB-3000™ is ready to process more packets.
-dec	with this selection, the USB-3000™ decodes channel packet data from the file <file_in>. The data in the <file_in> is expected to be in hard decision (8 bits per byte) format. Refer to the AMBE-3000™ Vocoder Chip User's Manual Section 6.9 CHAND field description. The speech packets are then sent back to the PC where the Speech data is extracted and written to the <file_out>. When complete, the USB-3000™ is ready to process more packets.
-dec -sdbits 1	Specify hard-decision decoding. The channel data input to the decoder uses hard-decision format, where each byte contains 8 hard-decision bits.
-dec -sdbits 4 or -decsd	Specify soft-decision decoding. The channel data input to the decoder uses soft-decision format, where each byte contains 2 4-bit soft-decision bits.
-decsd	with this selection, the USB-3000™ decodes channel packet data from the file <file_in>. The data in the <file_in> is decoded. The data in the <file_in> is expected to be in 4-bit soft decision (2 bits per byte) format. Refer to the AMBE-3000™ Vocoder Chip User's Manual Section 6.9 CHAND4 field description. The speech packets are then sent back to the PC where the Speech data is extracted and written to the <file_out>. When complete, the USB-300x™ is ready to process more packets.
-encdec	with this selection, the USB-300x™ encodes speech packets from the <file_in> then decodes and sends them back to the PC where the Speech data is extracted and written to the <file_out>. When complete, the USB-300x™ is ready to process more packets.
-alaw	Specify a-law companding. When used in conjunction with –enc, the speech data in the input speech file are assumed to contain a-law compressed samples. When used in conjunction with –dec, the speech data in the output speech file will be a-law compressed. Used with packet mode only.
-ulaw	When used in conjunction with –enc, the speech data in the input speech file are assumed to contain u-law compressed samples. When used in conjunction with –dec, the speech data in the output speech file will be u-law compressed. Used IN packet mode only.

-r <rate>
Specifies the vocoder bit rate. This option is used in conjunction with –enc or –dec. By default the AMBE-3000™ vocoder rate index selected is index 32 (3600 bps voice + 2800 bps FEC = 6400 bps total). –r R allows any other rate index to be selected. See AMBE-3000™ User's Manual for more information about available rate indices. Additional custom rates may be selected by specifying six 16-bit rate words using the format shown.

Option	Description
-r <rate Index No.>	Where the rate index is between 0and 63 as shown in Table 18 Standard Rate Table for AMBE-3000™ Vocoder Chip **(by index number)**
-r <Custom Rate words>	Where custom rate words are used in the format (0xNNNN 0xNNNN 0xNNNN 0xNNNN 0xNNNN 0xNNNN) as described Table 17 Custom Rate Control Words

-< Vocoder Options >
are the vocoder settings as follows:

Option	Description
-tone Idx Amp	Force transmission or generation of a tone. When used in conjunction with –enc, the encoder is forced to transmit channel data representing a tone with the specified tone index and amplitude. Note that not all vocoder rates support in-band tone signaling for all tone types (see AMBE-3000™ User's Manual). The quality is lower for tones that are not supported via in-band signaling. When used in conjunction with –dec, the decoder is forced to synthesize the specified tone. In-band signaling is not required for this feature; therefore, all tones are fully supported by all vocoder rates.
-dtx N	Enable Discontinuous Transmission (DTX) simulation. When DTX is enabled in conjunction with –enc the bits for certain silence frames are replaced with all ones. Replacing the channel data with all ones is used to simulate frames that are not transmitted. N controls how frequently silence frames are "transmitted". N = 0, indicates that silence frames are always transmitted, N = 1, indicates that every other silence frame is "transmitted", N =2, indicates that every third silence frame is "transmitted". N = 9999, indicates that every 10000th silence frame is "transmitted". N = 10000, is a special case where silence frames are never "transmitted". When N ≥ 0 is enabled in conjunction with –dec, the DTX simulation is enabled at the decode side. Before decoding a frame of bits, first a check is made to see if the frame of bits is all ones. If the frame is all ones then the decoder sets the CNI flag, indicating that no channel data was received such that the decoder should generate comfort noise. If the frame does not contain all ones, then it is passed to the decoder for normal decoding and the CNI flag is not set. Note that the DTX simulation is implemented in the microcontroller software.
-ns <state>	This option is used in conjunction with –enc. where the noise suppression is either turned off (state=0) or on (state=1) (default=on)

<file_in> <file_out>
This part of the command line is where that path to the files that are to be processed. All input files (.pcm files in the test vectors) to be encoded must be 16-bit linear PCM data sampled at 8kHz. All input files to be decoded must be previously encoded (.bit files in the test vectors) at a known rate.

All encoder output files will be encoded at the rate specified in the command line. Where as, all decoded files will be in the 16-bit linear PCM data sampled at 8kHz format.

- < Other Options >:
We have introduced usb3k_client32.exe using a few examples, but usb3k_client32.exe supports options that are more advanced. This section will provide more details about the usb3k_client32.exe command line options.

Option	Description
-q	is an optional switch that may be use to prevent the DVSI Copyright Notice from being printed to screen after the command has been processed.
-cmp [compare file name]	To validate that the file was decoded correctly simply compare the created file **dvsi36tst.pcm** with the similar file **dvsi36.pcm** included on the USB-300x™ CD.
-version	This should be specified without any other options. It will send a packet to the AMBE-3000™ to query its product ID and version strings. The resulting strings are displayed.

Table 3 Additional usb3k_client32.exe options

Vocoder Options Format
-r <"rate index#" or "custom rate words">
where the "rate index#" sets the bit rate of the AMBE-3000™ encoder. Alternatively, if "custom rate words" are used to set the rate – the format must be six words in hexadecimal format separated by a space as shown in the following example:
-r 0x0460 0x0986 0x0000 0x0000 0x0000 0x5660

3.6.1. Example Encode USB-300x™ File

-enc is used to encode a file from the PC, the (.pcm suffix) file input to the USB-300x™ via USB interface must be 16-bit linear PCM data sampled at 8kHz. The file from the PC will be encoded by the AMBE-3000™ and then sent back to PC and saved as the (.bit) file as named in the command line.

This example uses the USB-300x™ to encode the speech file, ../tv/dvsi.pcm (example speech data provided with the USB-300x™), using USB-300x™ rate index 40 (4000 bps voice rate with no FEC). The compressed speech data is written to a file named "**dvsi40tst.bit**". The baud rate used for communication between the PC and the USB-3000™ is 230400. The USB-3000™ is operated in packet mode for this command. Note that for real time operation in packet mode, a baud rate of at least 172800 is required.

*Note: Before running usb3k_client32.exe be sure that the dvsiserver32.exe program is running.

Figure 12 Encode File command line example

where

-r40 is the rate index of the bit rate the file is to be encoded at 4000 bps.

-enc encodes the PCM file and saves it to a file

dvsi.pcm is the name of the PCM file to be encoded

dvsi40tst.bit is the name of the encoded file to be saved

To validate that the file was encoded correctly simply compare the created file **dvsi40tst.bit** with the similar file **dvsi40.bit** included on the USB-3000™ CD.
In the USB-300x™ directory use the following DOS command.

Figure 13 Command line to check bit exact

3.6.1. Example Decode USB-300x™ File

-dec is used to decode a previously encoded (.bit) file from the PC, When the USB-300x™ receives the encoded (.bit) file over the USB interface it is processed by the AMBE-3000™ Vocoder Chip's decoder. The synthesized digital speech data is sent back to the PC via the USB interface and saved as the (.pcm suffix) file named in the command line.

This example uses the USB-3000™ to decode the channel file, dvsitst.bit, which was produced in example 1. Note that the rate index selected must match the rate index used in example 1 (40 in this case). The decoded speech data is written to a file named "dvsi40tst.pcm". The baud rate used for communication is 230400. The AMBE-3000™ is operated in packet mode for this command. Note that for real time operation in packet mode, a baud rate of at least 172800 is required.

```
usb3k_client32.exe-r 40 -dec dvsi40tst.bit dvsi40tst.pcm
```

Figure 14 Decode File Command Line Example

where

-r 40 is the rate index of the bit rate the file is to be decoded at 4000bps.

-dec decodes the previously encoded file and saves it to a file

dvsi40tst.bit is the name of the encoded file to be decoded

dvsi40tst.pcm is the name of the decoded file to be saved

To validate that the file was decoded correctly simply compare the created file **dvsi40tst.pcm** with the similar file **dvsi40.pcm** included on the USB-3000™ CD.
In the USB-3000™ directory use the following DOS command.

Figure 15 Command line to check bit exact

3.6.1. Example Encode USB-3000™ P25 File

APCO Project 25 full-rate
To configure the USB-3000™ P25 to run APCO Project 25 full-rate with FEC (7200 bps), the **usb3kcom.exe** program command line must include the following custom rate words (see example Figure 16 Encode File -- APCO P25 Full-Rate with FEC)

"-r 0x0558 0x086b 0x1030 0x0000 0x0000 0x0190"

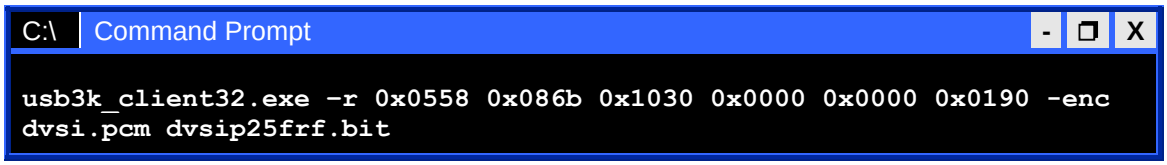

Figure 16 Encode File -- APCO P25 Full-Rate with FEC

To configure the USB-3000™ P25 to run APCO Project 25 full-rate with no FEC (4400 bps), the **usb3k_client32.exe** program command line must include the following custom rate words (see example Figure 17 Encode File -- APCO P25 Full-Rate without FEC)

"-r 0x0558 0x086b 0x0000 0x0000 0x0000 0x0158"

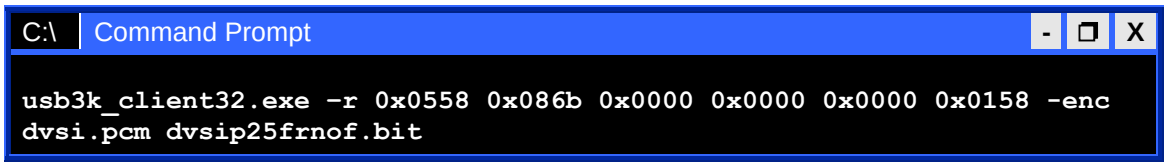

Figure 17 Encode File -- APCO P25 Full-Rate without FEC

Note:	The batch file, p25.bat, provides an example of using these custom rate words on the usb3kcom command line.

3.6.1. Example Encode USB-3000™ SAT File

TerreStar 2450 bps rate
To configure the USB-3000™ SAT to run the 2450 bps TerreStar vocoder the usb3kcom.exe program command line must include either

the Rate index control word `62`
(see example Figure 18 Encode File Rate Index – TerreStar 2450 bps)

Figure 18 Encode File Rate Index – TerreStar 2450 bps

or
custom rate words "`-r 0x0631 0x0754 0x0000 0x0000 0x0000 0x00331`"
(see example Figure 19 Encode File - Rate Control Words – TerreStar 2450 bps

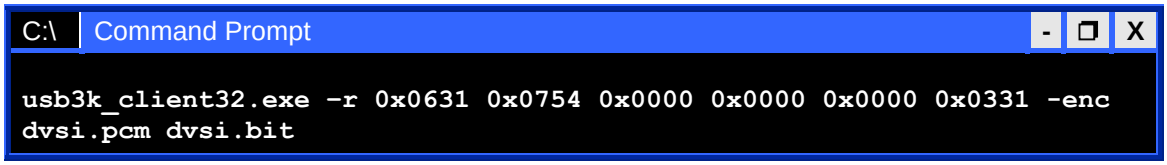

Figure 19 Encode File - Rate Control Words – TerreStar 2450 bps

TerreStar / GlobalStar 4000 bps rate
To configure the USB-3000™ SAT to run the 4000 bps TerreStar / GlobalStar vocoder the usb3kcom.exe program command line must include either the

Rate index control word `63`
(see example Figure 20 Encode File Rate Index – TerreStar / GlobalStar 4000 bps)

```
usb3k_client32.exe –r 63 -enc dvsi.pcm dvsi.bit
```

Figure 20 Encode File Rate Index – TerreStar / GlobalStar 4000 bps

or
custom rate words "`-r 0x064F 0x0986 0x0000 0x0000 0x0000 0x0350`"
(see example Figure 21 Encode File Rate Control Words– TerreStar / GlobalStar 4000 bps)

```
C:\ Command Prompt
usb3k_client32.exe –r 0x064F 0x0986 0x0000 0x0000 0x0000 0x0350 -enc
dvsi.pcm dvsi.bit
```

Figure 21 Encode File Rate Control Words– TerreStar / GlobalStar 4000 bps

3.7. usb3kcom.exe Program Description (obsolete)

The usb3kcom does communicate with both the USB-3000™ and the USB-3003™ but DVSI strongly suggests using the client / server model as described in 3.4. The usb3kcom.exe program is written entirely in C and C++ to facilitate easy modification by customers. The software is compact, yet includes all necessary code to communicate with the USB-3000™, provide interrupts for timing and arranges the channel packet protocol. The source code for the program is provided as an example to assist in the creation custom programs. This software demonstrates useful features of the AMBE-3000™ Vocoder Chip and is a good reference to use as a starting point for more complex designs tailored to specific needs.

The **`usb3kcom.exe`** program allows users to encode a PCM file and save it to the PC, or decode an encoded file and save it to the PC, or do both.

The USB-3000™ is set-up, and controlled from a PC. In order for the USB-3000™ to work with the PC drivers must be installed. See section USB Driver Installation for how to install the USB drivers. Once the connection between the PC and the USB-3000™ is established the USB-3000™ operations software should be installed on the PC.
The **`usb3kcom.exe`** program is supports both the single channel USB-3000™ and the three channel USB-3003™.

3.8. Running the usb3kcom Program

The usb3kcom.exe program is an executable file that requires no installation other than copying it to the \usb3000\bin folder. The USB-3000™ kit includes a CD that has the usb3kcom.exe PC executable program on it. This software should already have been copied from the CD to a directory located on a C-drive named C:\ usb3000 before the drivers were installed. If the CD was not copied onto the PC at that time, go back and follow the steps as detailed in Section 2.3.

The usb3kcom.exe program is used for all versions of the USB-3000™ and USB-3003™. The USB-3000™ and USB-3003™ are factory set to start-up ready to accept commands or to encode/decode packets via the USB interface. All control of the USB-3000™ is performed using a command prompt window and a command line interface. The file **`usb3kcom.exe`** is the control program that is run for board set-up and operation. To run the

program, open a command prompt window and change to the directory (`C:\usb3000\bin`) that has the `usb3kcom.exe` program file in it and type in the desired command.

<PC Communications Port Settings>

Specifies which COM Port the USB-3000™ interface is using. See Section USB Driver Installation for how to determine the COM Port Value. This value is always followed by the Baud rate value and is used in every command line.

PC communications port settings format

-port COM<#> <baud rate>

Note: The word COM must be in all Capital letters and there is no space between the word and the value.

<Vocoder Options>

The USB-3000™ provides the ability to set the data rate and noise suppression of the AMBE-3000™ Vocoder chip.

3.1. Running usb3kcom.exe Program with the USB-3003™

When using the usb3kcom.exe program file with the USB-3003™ it is necessary to include the **<Process Selection>** and **<file_in> <file_out>** information for each of the three channels. For easy configuration of these parameters, the usb3kcom.exe program file is able to call an ASCII text file that can contain the pertinent information of which channels and files to process.

For the USB-3003™ multi-channel device, the command line MUST be as follows:

Figure 22 Command line structure

where:

<file_name> is the name of a text file that contains the **<Process Selection>** (port, channel and rate) and **<file_in> <file_out>** information.

Example ASCII text file that will use com port `COM10` at `921600` baud to encode and decode three files (`infile0 infile1, infile2`) to three output files (`outfile0, outfile1, outfile2`) at index rates `31, 60, 44` respectively.

```
#This is a comment
-port COM10 921600 -encdec
-r 31 infile0 outfile0
-r 60 infile1 outfile1
-r 44 infile2 outfile2
#end
```

Lines that start with `#` are comments, but the file needs to end with `#end`

The first line in the file that is not a comment is expected to contain the COM port parameters baud rate and "Main options" just like the single channel program. The next three lines are "Secondary Options" and relate to each one of the three channels. If only two channels are to be called, only enter two "Secondary Options" lines. If only one channel is to be used, then only one line "Secondary Options" line of information needs to be entered. By only entering one secondary option line, this method of calling a text file may also be used for a single channel USB-3000™

4. Packet Interface

4.1. Overview

The USB-300x™ packet interface is ideal for situations where the **usb3kcom.exe** program is not feasible. The packets are used when communicating with the USB-300x™ to configure the AMBE-3000™ vocoder chip, poll vocoder status information, as well as, the ability to transfer speech data samples to the encoder or from the decoder.

The packet interface used by the USB-300x™ is identical to the one used in DVSI's AMBE-3000™ Vocoder Chip. The AMBE-3000™ Vocoder Chip always uses a packet format for the compressed voice data bits and for the chip configuration/control. The Packets are designed such that they can be as small as possible.

Every packet includes a HEADER that consists of a START byte for identification of the beginning of the packet, LENGTH data to indicate how many bytes are in the packet and a TYPE byte that specifies what to do with the packet. Packets are processed in a first-in-first-out manner.

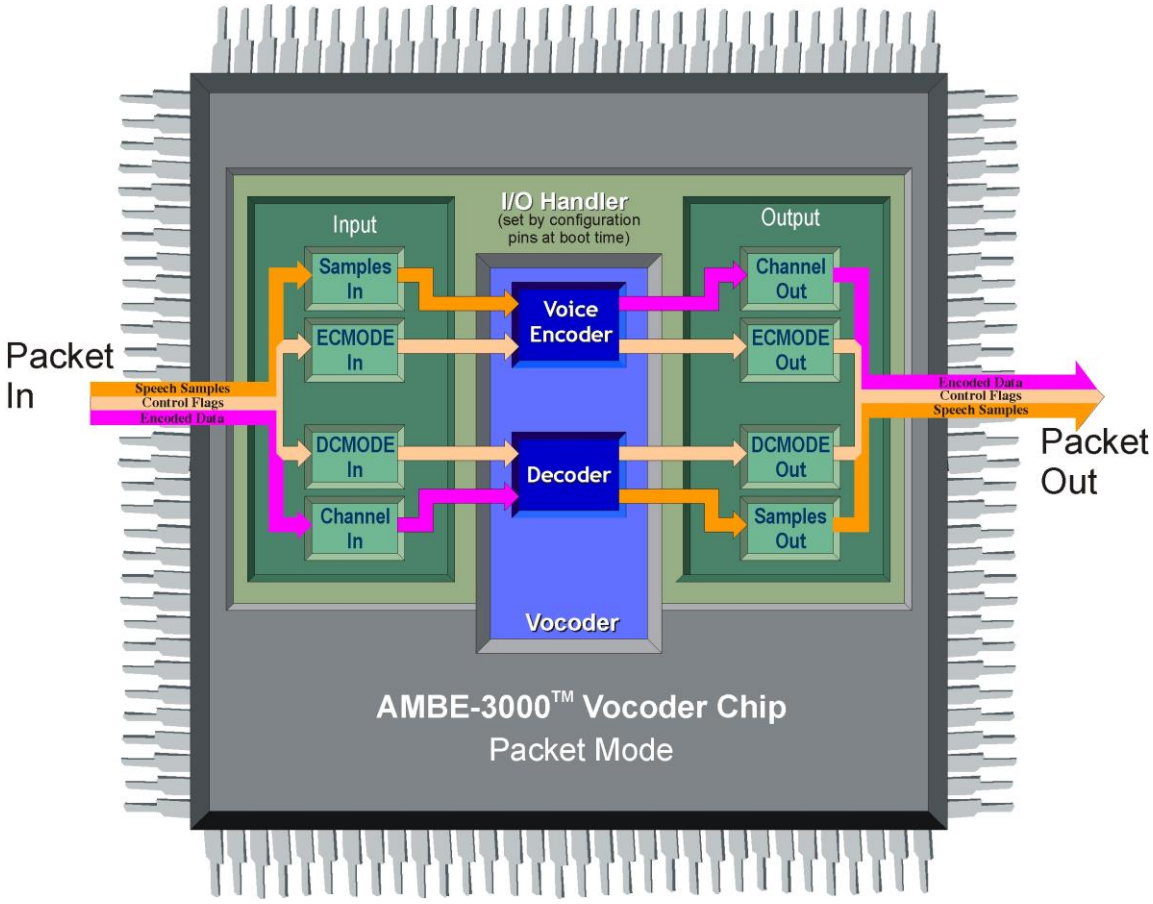

Figure 23 AMBE-3000™ Packet Mode Block Diagram

4.2. Packet Structure

The USB-300x™ transmits / receives data packets via the USB to the onboard FTDI USB chip. Then through the FTDI chip's UART Serial interface, packets are sent/received to/from the AMBE-3000™ vocoder chip. The packet is the same structure as described in the respective AMBE-3000™ Vocoder chip User's Manual and AMBE-3003™ Vocoder chip User's Manual. It is recommended that the user read the correct User's Manual for the single channel (AMBE-3000™ Vocoder chip) or three channel (AMBE-3003™) chip to become familiar with the packet structure.

As the USB-300x™ receives packets; it processes the packets and sends response packets as soon as the data is ready. The USB-300x™ sends response packets in the same order that the packets are received. The AMBE-3000™ Vocoder Chip maintains a FIFO for received packets and a separate FIFO for packets that are awaiting transmission. The FIFOs are each large enough to accommodate up to two speech packets and two channel packets. The USB-300x™ can continue to transmit/receive packets while it is still processing prior packets.

When the USB-300x™ receives a speech packet, it takes the speech samples from the packet, encodes them and sends back a channel packet.

When the USB-300x™ receives a channel packet, it takes the channel data from the packet, decodes the channel data, and sends back a speech packet.

When the USB-300x™ receives a configuration control packet, it makes the requested configuration changes and sends back a configuration response packet.

The following are two simple examples of a configuration / control packet that can be used to gain a good understanding of how the packet interface works.

4.3. Example Product ID Packet

When a Product ID packet is sent to the USB-300x™, it will respond with a string that contains the product identification of the internal AMBE-3000™ Vocoder chip.

Following is an example configuration / control packet (hexadecimal) for input to the USB-300x™:

Packet Format with Parity Field					
Control / Configuration Packet For PRODUCT ID					
Packet Header			Fields	Parity	
START_BYTE	**LENGTH**	**TYPE**	**Control Packet Field ID**	PKT_PARITY	PARITY_BYTE
(1 byte)	(2 bytes)	(1 byte)	(1 byte)	(1 byte)	(1 byte)
61	**0003**	**00**	**30**	**2F**	**1C**

Table 4 PKT_PRODID Field

Product ID Packet Description
The first byte (0x61) is the packet header byte. The next two bytes (0x0003) specify the total length of the packet fields (3 bytes in this example). Note that the total packet length including the header, length, and type is 7 bytes. The next byte (0x00) specifies that the packet type is a control packet. The next byte (0x30) is the field identifier for a indicating it is a PKT_PRODID request. The next two bytes (0x2F1C) show that parity fields are enabled. The first byte of the parity field is the parity field identifier and is always equal to 0x2f. The second byte of the parity field is the parity byte. It is obtained by "Exclusive-oring" every byte in the packet, except for the START_BYTE and the PARITY_BYTE, together. If parity fields are enabled, the AMBE-3000™ Vocoder Chip checks the parity byte for all received packets and discards any packet that has an incorrect parity byte.

The USB-300x™ will respond with the following configuration / control packet (hexadecimal).

Packet Format with Parity Field						
Control / Configuration **RESPONSE** Packet For PRODUCT ID						
Packet Header			Fields		Parity	
START_BYTE	**LENGTH**	**TYPE**	**Control Packet Field ID**	**Response Field Data**	**PKT_ PARITY**	**PARITY_ BYTE**
(1 byte)	(2 bytes)	(1 byte)	(1 byte)	(10 bytes)	(1 byte)	(1 byte)
61	**00 0D**	**00**	**30**	**41 4D 42 45 33 30 30 30 46 00**	**2F**	**5C**

Table 5 PKT_PRODID Response Field

The first byte (0x61) is the packet header byte. The next two bytes (0x000D) specify the total length of the packet fields (13 bytes in this example). Note that the total packet length including the header, length and type is 17 bytes. The next byte (0x00) specifies that the packet type is a control packet. The next byte (0x30) is the field identifier for a indicating it is a PKT_PRODID response. The next 10 bytes is a null-terminated string that contains the product identification in hex (AMBE3000F. in ASCII). The last two bytes (0x2F1C) show that parity fields are enabled (0x2f) and the parity field value (0x5C).

4.4. Example PKT_VERSTRING Packet

The following is a configuration / control packet (hexadecimal) PKT_VERSTRING field. This field will cause the USB-300x™ to respond with a string that contains the product version number of the internal AMBE-3000™ Vocoder chip.

Packet Format with Parity Field					
Control / Configuration Packet For PRODUCT VERSION STRING					
Packet Header			Fields	Parity	
START_BYTE	**LENGTH**	**TYPE**	**Control Packet Field ID**	PKT_PARITY	PARITY_BYTE
(1 byte)	(2 bytes)	(1 byte)	(1 byte)	(1 byte)	(1 byte)
61	**0003**	**00**	**31**	**2F**	**1D**

Table 6 PKT_VERSTRING Field

This is the similar to the prior example except that it is requesting the product version number of the embedded AMBE-3000™ Vocoder Chip. The first byte (0x61) is the packet header byte. The next two bytes (0x0003) specify that the length of the packet (excluding the header, length, and type bytes) is 3 bytes. The next byte (0x00) specifies that the packet type is a control packet. The next byte (0x31) is the field identifier for a PKT_VERSTRING field. The last two bytes (0x2F1C) show that parity fields are enabled (0x2f) and the parity field value (0x1D).

From the above example, the following is the response packet from the AMBE-3000™ Vocoder Chip:

Packet Format with Parity Field						
Control / Configuration **RESPONSE** Packet For PRODUCT VERSION STRING						
Packet Header			Fields		Parity	
START_BYTE	**LENGTH**	**TYPE**	**Control Packet Field ID**	**Response Field Data**	**PKT_ PARITY**	**PARITY_ BYTE**
(1 byte)	(2 bytes)	(1 byte)	(1 byte)	(48 bytes)	(1 byte)	(1 byte)
61	00 33	00	31	56 31 32 30 2E 45 31 30 30 2E 58 58 58 58 2E 43 31 30 36 2E 47 35 31 34 2E 52 30 30 37 2E 41 30 30 33 30 36 30 38 2E 43 30 30 32 30 32 30 38 00	2F	73

Table 7 PKT_VERSTRING Response Field

Again, the first byte (0x61) is the packet header byte. The next two bytes (0x0033) specify the total length of the packet fields (51 bytes in this example). Note that the total packet length including the header, length, and type is 55 bytes. The next byte (0x00) specifies that the packet type is a control packet. The next byte (0x31) is the field identifier for a indicating it is a PKT_VERSTRING response. The next 48 bytes is a null-terminated string that contains the product version in hex (`V100.E100.XXXX.C106.G514.R007.A0030608.C0020208` in ASCII). The last two bytes (0x2F73) show that parity fields are enabled (0x2f) and the parity field value (0x73).

4.5. USB-300x™ P25 Packet Interface

The packet interface used by the USB-300x™ P25 is the same as to the one used in the USB-300x™, except that the PKT_RATEP field is expanded to support two additional custom rates. To run the APCO Project 25 rates the following packet fields can be used.

APCO Project 25 full-rate (PKT_RATEP field)
To select the APCO full-rate vocoder via packet interface the PKT_RATEP field (13 bytes total) is used

Example of a PKT_RATEP field with the APCO Project 25 full-rate with FEC (7200 bps)

Field Identifier	RCW 0	RCW 1	RCW 2	RCW 3	RCW 4	RCW 5
0x0A	0x0558	0x086b	0x1030	0x0000	0x0000	0x0190

Table 8 PKT_RATEP Field to select full-rate with FEC

Example of a PKT_RATEP field with the APCO Project 25 full-rate with No FEC (4400 bps)

Field Identifier	RCW 0	RCW 1	RCW 2	RCW 3	RCW 4	RCW 5
0x0A	0x0558	0x086b	0x0000	0x0000	0x0000	0x0158

Table 9 PKT_RATEP Field to select full-rate with No FEC

APCO Project 25 half-rate (PKT_RATET field)
To select the APCO half-rate vocoder via packet interface the **PKT_RATET** field (2 bytes total) is used.

Example of a PKT_RATET field with the APCO Project 25 half-rate with FEC (3600 bps) Rate Index 33

PKT_RATET Field	
Field Identifier	Control Field Data
1 Byte	1 Byte
0x09	0x21

Table 10 PKT_RATET Field to select Rate Index Value 33

Example of a PKT_RATET field with the APCO Project 25 half-rate with No FEC (2450 bps) Rate Index 34

PKT_RATET Field	
Field Identifier	Control Field Data
1 Byte	1 Byte
0x09	0x22

Table 11 PKT_RATET Field to select Rate Index Value 34

4.6. USB-300x™ SAT Packet Interface

The packet interface used by the USB-300x™ SAT is the same as to the one used in the USB-300x™, except that the PKT_RATET and PKT_RATEP fields are expanded to support two additional rates (Rate 62 and Rate 63) . To run the **TerreStar / GlobalStar** rates the following packet fields can be used.

TerreStar 2450 bps Rate Index Value (PKT_RATET Field)
(2 bytes total)

PKT_RATET Field – TerreStar 2450 bps	
Field Identifier	Control Field Data
1 Byte	1 Byte
0x09	0x3E

Table 12 PKT_RATET Field for TerreStar 2450bps (Rate Index Value 62)

TerreStar / GlobalStar 4000 bps Rate Index Value (PKT_RATET Field)
(2 bytes total)

PKT_RATET Field – TerreStar / GlobalStar 4000 bps	
Field Identifier	Control Field Data
1 Byte	1 Byte
0x09	0x3f

Table 13 PKT_RATET Field for TerreStar / GlobalStar 4000 bps (Rate Index Value 63)

TerreStar 2450 bps Custom Rate Words (PKT_RATEP Field)
(13 bytes total)

PKT_RATEP Field – TerreStar (2450 bps)						
Field Identifier	Control Fields Data					
1 Byte	Rate Control Words (6 Words)					
0x0A	0x0631	0x0754	0x0000	0x0000	0x0000	0x0331

Table 14 PKT_RATEP Field for TerreStar 2450bps (Rate Control Words)

TerreStar / GlobalStar 4000 bps Custom Rate Words (PKT_RATEP Field)
(13 bytes total)

PKT_RATEP Field – TerreStar / GlobalStar (4000 bps)						
Field Identifier	Control Fields Data					
1 Byte	Rate Control Words (6 Words)					
0x0A	0x064F	0x00986	0x0000	0x0000	0x0000	0x0350

Table 15 PKT_RATEP Field for TerreStar / GlobalStar 4000 bps (Rate Control Words)

5. Hardware Specifications

5.1. Overview

This section contains hardware Specifications of the USB-300x™.

NOTE: All specifications subject to change.

5.2. Board Connection

USB Serial Port	
Type	Serial
Connector	USB Type A (male plug)

USB Pin Out	
Pin Number	Name
1	USB_5v
2	D-
3	D+
4	ID
Shield	Connected to Ground

5.3. Mechanical

Mechanical	
Weight	<1 oz.
Size including connector (L x W x H)	2.5 X 1.0 X 0.5 inches

6. Appendix

6.1. Rate Tables

Total Rate (bps)	Speech Rate (bps)	FEC Rate (bps)	Rate Index
2000	2000	0	31
2250	2250	0	36
2400	2400	0	0
	2350	50	5
	2400	0	37
2450	Rate Index 34 is interoperable with APCO Project 25 half-rate without FEC		
	2450	0	34
	Rate Index 62 is interoperable with TerreStar satellite terrestrial mobile network		
	2450	0	62
2700	2450	250	47
3000	3000	0	38
3400	2250	1150	35
3600	3600	0	1
	3350	250	11
	3600	0	16
	Rate Index 33 is interoperable with APCO Project 25 half-rate with FEC (3600 bps)		
	2450	1150	33
	3600	0	39
	3350	250	48
4000	4000	0	15
	3750	250	14
	4000	0	17
	2400	1600	22
	4000	0	40
	3750	250	49
	2600	1400	55
	Rate Index 63 is interoperable with **TerreStar / GlobalStar** mobile satellite network		
	4000	0	63
4400	4400	0	41
	2450	1950	51
4800	4800	0	3
	4550	250	7
	3600	1200	2
	3100	1700	8
	4800	0	18
	4000	800	24
	3600	1200	23
	2400	2400	25

	4800	0	42
	4550	250	50
	2450	2350	52
	3600	1200	56
	4000	800	57
6000	2450	3550	53
6400	4150	2250	10
	6400	0	19
	4000	2400	26
	3600	2800	32
	6400	0	43
	4000	2400	58
7200	4400	2800	9
	4400	2800	27
	7200	0	44
	2450	4750	54
	4400	2800	59
8000	7750	250	12
	4650	3350	13
	8000	0	20
	4000	4000	28
	8000	0	45
	4000	4000	60
9600	9600	0	4
	4850	4750	6
	9600	0	21
	3600	6000	30
	2400	7200	29
	9600	0	46
	3600	6000	61
Table Key			
AMBE-1000™ Rates (AMBE™ Vocoder)			
AMBE-2000™ Rates (AMBE+™ Vocoder)			
AMBE-3000™ Rates (AMBE+2™ Vocoder)			

Table 16 Standard Rate Table for AMBE-3000™ (by rate)

Total Rate (bps)	Speech Rate (bps)	FEC Rate (bps)	RCW 0	RCW 1	RCW 2	RCW 3	RCW 4	RCW 5
This rate is interoperable with APCO Project 25 full-rate with FEC (7200 bps)								
7200	4400	2800	**0x0558**	**0x086b**	**0x1030**	**0x0000**	**0x0000**	**0x0190**
This rate is interoperable with APCO Project 25 full-rate without FEC (4400 bps)								
4400	4400	0	**0x0558**	**0x086b**	**0x0000**	**0x0000**	**0x0000**	**0x0158**
This rate is interoperable with D-STAR								
3600	2400	1200	**0x0130**	**0x0763**	**0x4000**	**0x0000**	**0x0000**	**0x0048**
This rate is interoperable with **TerreStar** satellite terrestrial mobile broadband network								
2450	2450	0	**0x0631**	**0x0754**	**0x0000**	**0x0000**	**0x0000**	**0x0331**
This rate is interoperable with **TerreStar** / **GlobalStar** mobile satellite voice networks								
4000	4000	0	**0x064F**	**0x0986**	**0x0000**	**0x0000**	**0x0000**	**0x0350**

Table 17 Custom Rate Control Words

Vocoder Rates by Index Number				
AMBE-1000™ Rates				
Rate Index #	Total Rate		Speech Rate	FEC Rate
0	2400		2400	0
1	3600		3600	0
2	4800		3600	1200
3	4800		4800	0
4	9600		9600	0
5	2400		2350	50
6	9600		4850	4750
7	4800		4550	250
8	4800		3100	1700
9	7200		4400	2800
10	6400		4150	2250
11	3600		3350	250
12	8000		7750	250
13	8000		4650	3350
14	4000		3750	250
15	4000		4000	0
AMBE-2000™ Rates				
Rate Index #	Total Rate		Speech Rate	FEC Rate
16	3600		3600	0
17	4000		4000	0
18	4800		4800	0
19	6400		6400	0
20	8000		8000	0
21	9600		9600	0
22	4000		2400	1600
23	4800		3600	1200
24	4800		4000	800
25	4800		2400	2400
26	6400		4000	2400
27	7200		4400	2800
28	8000		4000	4000
29	9600		2400	7200
30	9600		3600	6000
31	2000		2000	0
AMBE-3000™ Rates				
Rate Index #	Total Rate		Speech Rate	FEC Rate
32	6400		3600	2800
Rate 33 is interoperable with APCO Project 25 half-rate with FEC (3600 bps)				
33	3600		2450	1150
Rate 34 is interoperable with APCO Project 25 half-rate without FEC (2450 bps)				
34	2450		2450	0
35	3400		2250	1150
36	2250		2250	0
37	2400		2400	0
38	3000		3000	0
39	3600		3600	0
40	4000		4000	0

41	4400		4400	0
42	4800		4800	0
43	6400		6400	0
44	7200		7200	0
45	8000		8000	0
46	9600		9600	0
47	2700		2450	250
48	3600		3350	250
49	4000		3750	250
50	4800		4550	250
51	4400		2450	1950
52	4800		2450	2350
53	6000		2450	3550
54	7200		2450	4750
55	4000		2600	1400
56	4800		3600	1200
57	4800		4000	800
58	6400		4000	2400
59	7200		4400	2800
60	8000		4000	4000
61	9600		3600	6000
Rate Index 62 is interoperable with **TerreStar** satellite terrestrial mobile network				
62	2450		2450	0
Rate Index 63 is interoperable with **TerreStar / GlobalStar** mobile satellite network				
63	4000		4000	0

Table 18 Standard Rate Table for AMBE-3000™ Vocoder Chip (by index number)

6.2. File Formats

The usb3kcom.exe program uses three types of files for storing input and/or output data transferred to/from the USB-300x™. The three (3) file formats are as follows:

1. **PCM File Type**. A PCM file is a binary file that contains 16-bit PCM speech samples sampled at 8 kHz. The file does not contain any header information. It contains only speech data. The data may be input to the encoder or output from the decoder. Each speech sample occupies two successive bytes in the file. The first byte contains the least significant 8-bits of the PCM sample and the second byte contains the most significant 8-bits of the PCM sample. To illustrate this assume that the following 16-bit PCM samples are stored in a PCM file:

   0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080,
   0x0100, 0x0200, 0x0400, 0x0800, 0x1000, 0x2000, 0x4000, 0x8000

   The order in which the bytes are read from the file is as follows:

   0x01, 0x00, 0x02, 0x00, 0x04, 0x00, 0x08, 0x00,
   0x10, 0x00, 0x20, 0x00, 0x40, 0x00, 0x80, 0x00,
   0x00, 0x01, 0x00, 0x02, 0x00, 0x04, 0x00, 0x08,
   0x00, 0x10, 0x00, 0x20, 0x00, 0x40, 0x00, 0x80.

2. **Hard-Decision Bit File Type**. A hard-decision bit file contains compressed speech data output by the encoder. The bit file can be used as input to the decoder. The data is packed using 8 bits per byte. For hard-decision, each bit must be 0 or 1. If the 16 bits

   `a, b, c, d, e f, g, h, i, j, k, l, m, n, o, p`

   are the first 16 bits stored in a hard-decision bit file. Then the first two bytes of the file will be binary `abcdefgh` and `ijklmnop`.

3. **Soft-Decision Bit File Type**. A soft-decision bit file contains compressed speech data output by the encoder that has then been converted to 4-bit soft-decision format. Soft-decision format is not output directly by the encoder, but it can be input directly to the decoder when soft-decision decoding is specified. The data is packed using two soft-decision bits per byte. Each soft decision bit must be a 4-bit value in the range from 0x0 to 0xF. A binary "0" is represented as 0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, or 0x7, with 0x0 being the most confident "0" and 0x7 being the least confident "0". A binary "1" is represented as 0xF, 0xE, 0xD, 0xC, 0xB, 0xA, 0x9, or 0x8, with 0xF being the most confident "1" and 0x8 being the least confident "1". If a soft-decision bit file is derived directly from a hard-decision bit file, then each bit will have maximum confidence and will be equal to either 0x0 for "0" or 0xF for "1". If the 16 bits

   `a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p`

   are the first 16 bits stored in a hard-decision bit file. Then the first eight bytes of the converted soft-decision format file will be binary

   `aaaabbbb ccccdddd eeeeffff gggghhhh iiiijjjj kkkkllll mmmmnnnn oooopppp.`
   If each of the 16 4-bit soft-decision bits are denoted as `A, B, C, D, E, F, G, H, I, J, K`, and `L`, then the first eight bytes of the file could be expressed as hex `AB, CD, EF, GH, IJ, KL, MN`, and `OP`. `A` through `P` are each 4-bit soft-decision bits in the range from 0x0 to 0xF. If the transmission is not ideal, then the values for each 4-bit soft-decision bit will vary between 0x0 and 0xF depending upon the confidence of each received bit.

6.3. Additional Reference Material

AMBE-3000™ vocoder chip User's Manual
http://www.dvsinc.com/literature.htm

AMBE-3003™ vocoder chip User's Manual
http://www.dvsinc.com/literature.htm

6.4. How to check the USB-300x™ Version

On earlier versions of the USB-300x™ (units shipped prior to October 2014), DVSI had modified the generic FTDI drivers so that the USB-300x™ uses a virtual COM port on the PC to provide access to the USB-300x™. These USB-300x™ versions required two drivers to be installed, the Virtual COM Port (VCP) driver and the

USB Serial Port driver. The application software provided with the USB-300x™ required the installation of these two drivers.

Without installing any drivers the USB-300x™ can be checked to see if, it is either an earlier version (DVSI drivers) or a new (FTDI drivers) version. It is not necessary to install any driver to check the version of the USB-300x. To check the version type, simply connect the USB-300x™ to a PC's USB connector and follow these steps:

1. When connecting the USB-300x™ to a PC for the first time the "Found New Hardware Window" will appear. Click CANCEL because the drivers do not need to be installed at this time. Continue on to Step 2.

2. Now that the USB-300x™ is connected to the USB interface, open Windows Device Manager. To open Windows Device Manager Click "Start", click "Run", and then type "devmgmt.msc" (without the quotation marks). Alternatively, open the Device Manager (located in "Control Panel\System" then select the "Hardware" tab and click "Device Manger") and select "View > Devices by Type"

3. Look under "Ports (COM & LPT)" for USB Serial Port (COMXX). Right click on this and select properties.

4. A new window will open, and then select the Details tab. In the Details window, select Hardware IDs under the Property pulldown menu. The value should appear as
   `FTDIBUS\COMPORT&VID_0403&PID_8F50` where 8F50 indicates it is an earlier version with a DVSI modified virtual COM port
   or
   `FTDIBUS\COMPORT&VID_0403&PID_6001` where 6001 indicates it uses FTDI's drivers

Each version of the USB-300x™ will have a different product description string as shown in Table 19 USB-300™ Product description string

Product name	FTDI Description string	DVSI Product Name String
USB-3000™	USB-3000	AMBE3000R
USB-3003™	USB-3003	AMBE3003F
USB-3000™ P25	USB-3000	AMBE3000P25F
USB-3003™ P25	USB-3003	AMBE3003P25F
USB-3000™ SAT	USB-3000	AMBE3000SATF

Table 19 USB-300™ Product description string

To view the FTDI Description string follow steps 1 through step 5 in the Reprogramming a USB-3000™ Shipped Prior to October 2014 procedure in section 6.5. Once Step 5 is completed, the description can be seen in the "Property" / "Value" window as shown in Figure 31 Verify device is programmed correctly.

To view the DVSI Product Name string run the bat file named usb3kversion.bat as described in Section 3.3 USB-300x™ Software CD Description. Alternatively, a ProductID packets can be sent to the USB-300x™ as described in section 4.3 Example Product ID Packet.

6.5. Reprogramming a USB-3000™ Shipped Prior to October 2014

If you already have the older version of the USB-3000™, the product ID (PID) must be reprogrammed before it can operate with the FTDI drivers. To reprogram the PID in a USB-3000™ in order to operate with the FTDI drivers the user must follow the procedure below.

Step 1. Install the DVSI drivers provided with the USB-3000™. Refer to the USB-3000™ User Manual for the driver installation procedure.

Step 2. Connect the USB-3000™ to the PC.

Step 3. Download a utility program (FT_PROG.exe) from the FTDI website
http://www.ftdichip.com/resources/utilities.htm

> Note:
>
> The FT_PROG requires Microsoft .NET Framework 2.0 to be installed on your system in order to run. If this is not already installed it can be obtained from Microsoft's website:
> http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en
>
> After downloading, follow the installation wizard for.NET 2.0.

Step 4. Run the programming utility by double clicking on the FT_PROG.exe icon. The FT Program will open in an idle mode with the following screen.

> Note
> For more detailed information regarding the FT_PROG.exe refer to the following document that can be found on FTDI's website
> http://www.ftdichip.com/
>
> Future Technology Devices International Ltd.
> Application Note AN_124
> User Guide For FTDI FT_PROG Utility

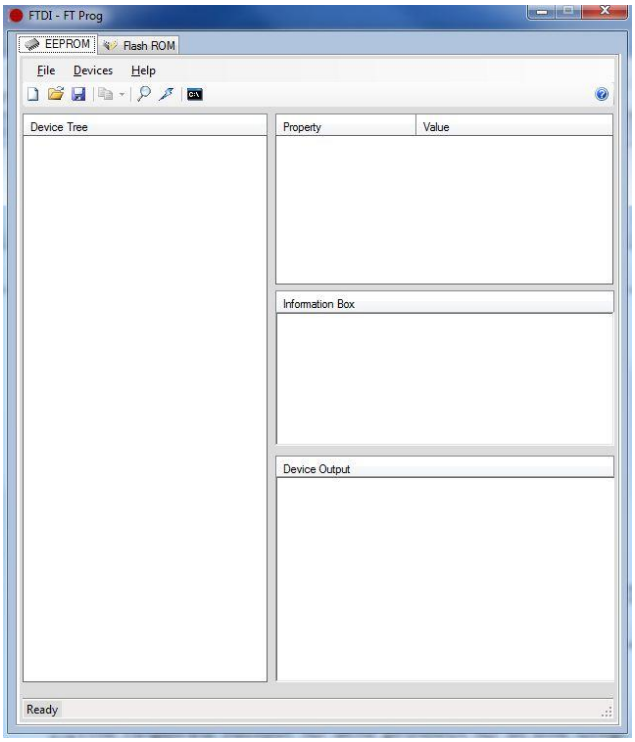

Figure 24 FTDI Program Initial Screen

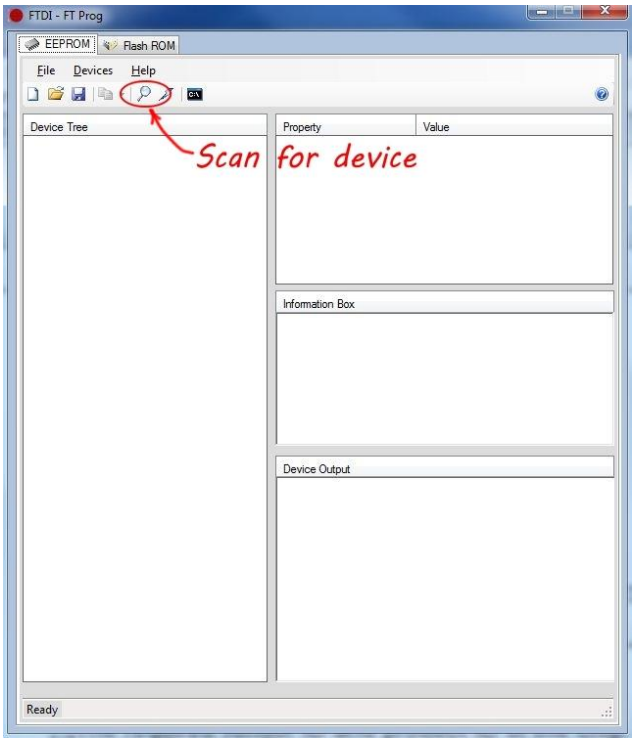

Figure 25 Scan for devices

Step 5. Scan the USB bus for USB-3000™ devices, by clicking on the "Scan and Parse" magnifying glass button on the toolbar (see Figure 25 Scan for device). Alternatively, select "Scan and Parse" from the "Devices"

menu. Upon successful completion of the operation, any connected USB-3000™ devices will be displayed within the "Device Tree" window, as shown below.

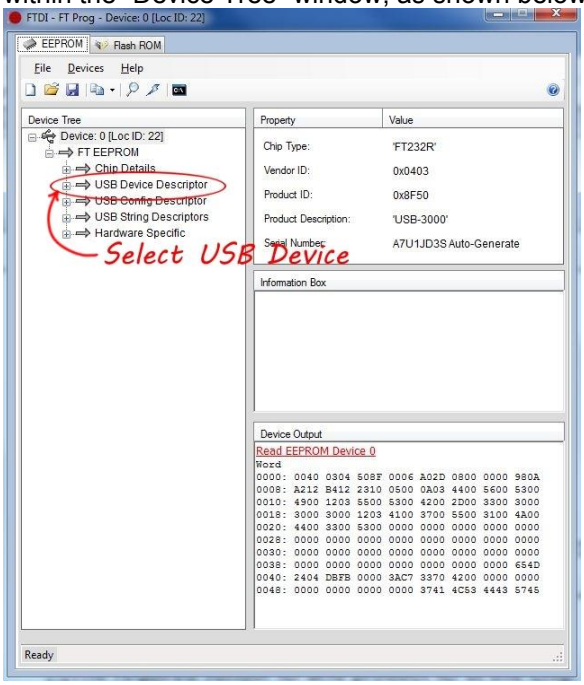

Figure 26 Select USB device

Step 6. To change the Product ID from DVSI's PID to the FTDI's default PID select the USB Device Descriptor in the left window panel. Expand it by clicking on the "+" sign. Figure 26 Select USB device then in the right side window panel click on the Custom VID/PID pull down menu and select FTDI Default. Figure 27 Select FTDI default

The value for the FTDI default product ID is 0x6001

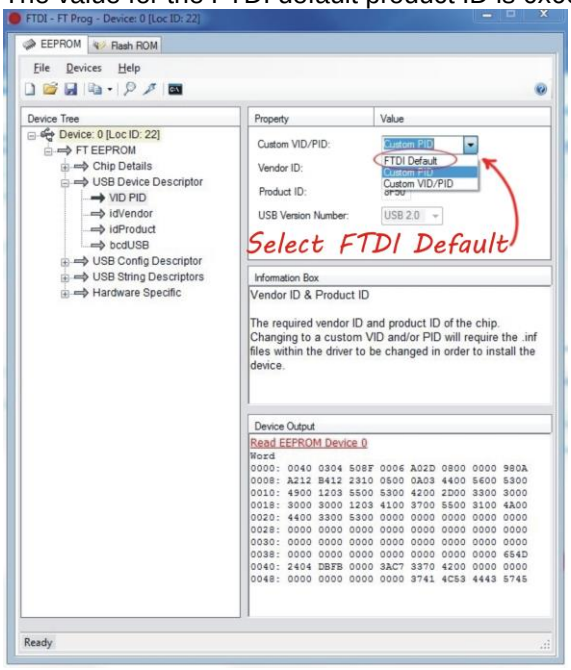

Figure 27 Select FTDI default setting

Step 7. To set the Product ID value onto the USB-3000™ select the program icon (the lightning bolt in the toolbar). See Figure 28 Program default settings to device

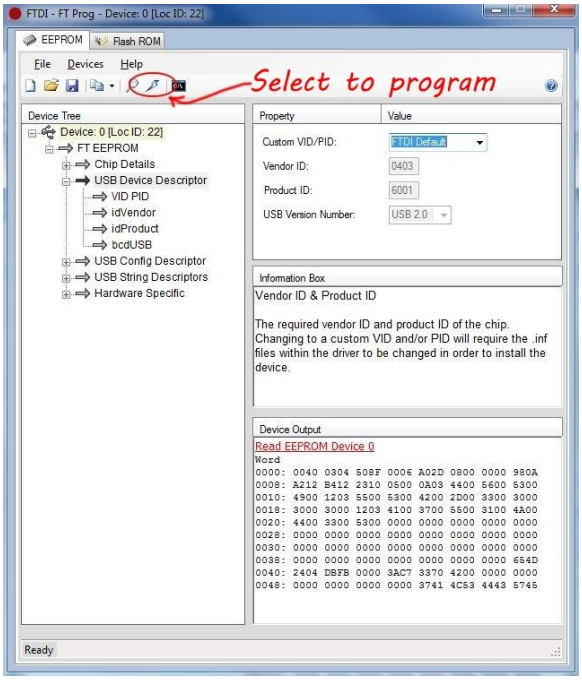

Figure 28 Program default settings to device

The "Program Devices" window [as below Figure 29 Review settings and proceed to program] will appear. From this interface, select the USB-3000™ device from the "Device List" on the left of the window.

This FTDI program utility window allows the PID on multiple devices to be reprogrammed all at once just by selecting multiple devices from the "Device List" on the left hand side of the window.

To program the selected devices press "Program" on the bottom right of the window.

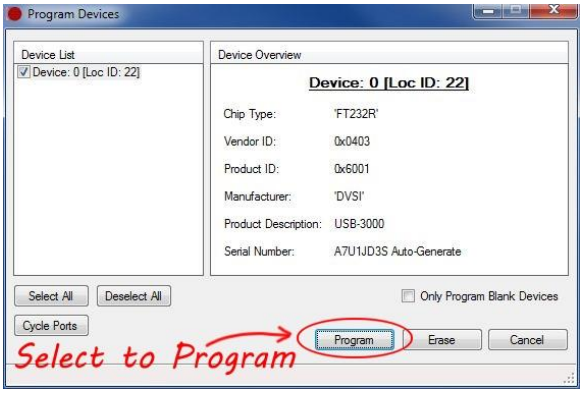

Figure 29 Review settings and proceed to program

Step 8. Verify the USB-3000™ now has the default FTDI Product ID; by right clicking on the USB Device Descriptor and Click on select the "Cycle Port" button. See Figure 30 Cycle Port This will to re-enumerate the USB-3000™ device after the EEPROM has been reprogrammed. This is useful because the device only reads the EEPROM when it is enumerated on USB, so it forces the device to use the new EEPROM contents.

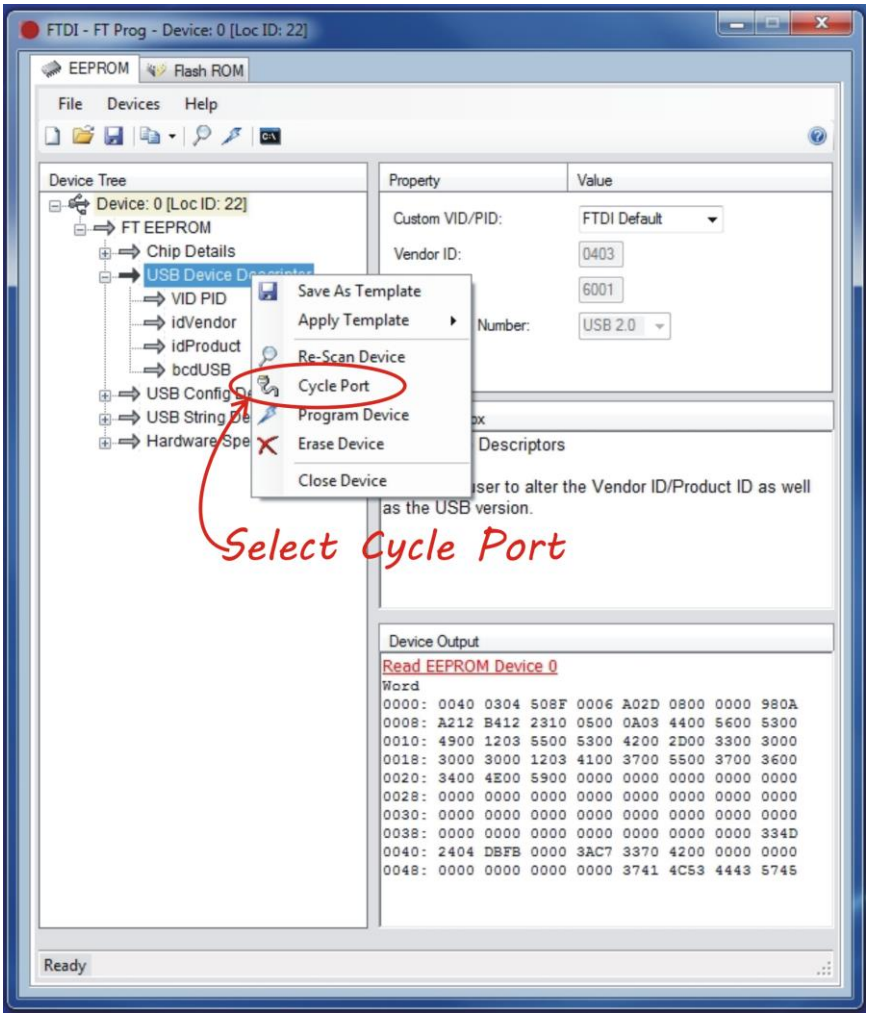

Figure 30 Cycle Port

Visually check the right side panel in the window now indicates the USB-3000™ Product ID is 0x6001. See Figure 31 Verify device is programmed correctly.

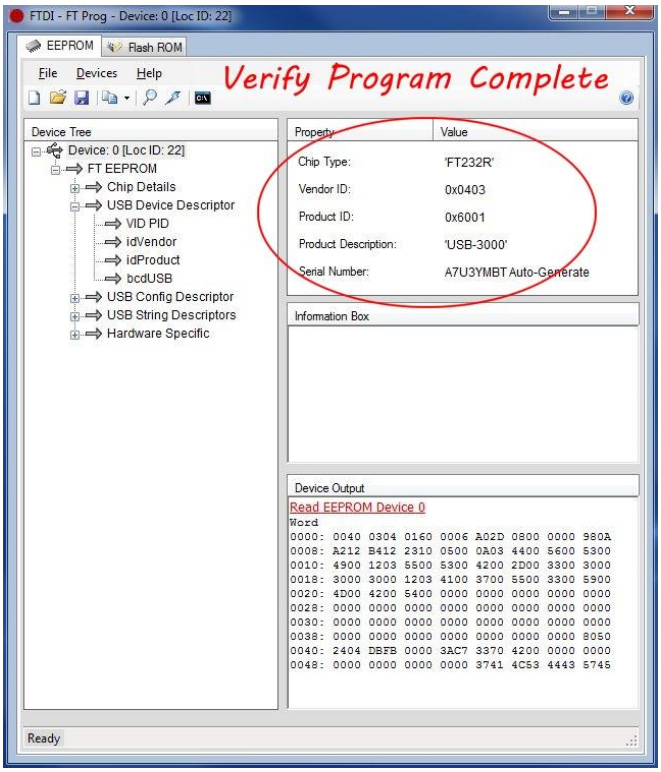

Figure 31 Verify device is programmed correctly

Step 9. Follow the instructions on the FTDI website for installing the latest version of the FTDIdrivers (as required to run your application).

http://www.ftdichip.com/Support/Knowledgebase/index.html

6.5.1. Reverting back to DVSI factory settings

To change the USB-3000™ back to the original settings as it was delivered from DVSI follow steps 1 through 5 as described herein. Then follow this Step 6.

Revert Step 6: To change the Product ID from FTDI's default PID to DVSI's PID select the USB Device Descriptor in the left window panel. Expand it by clicking on the "+" sign. Figure 26 Select USB device, then in the right side window panel click on the FTDI Default pull down menu and select Custom PID. See Figure 32 Reprogram DVSI Product ID.

Then enter the value for the DVSI product ID as **0x8F50**

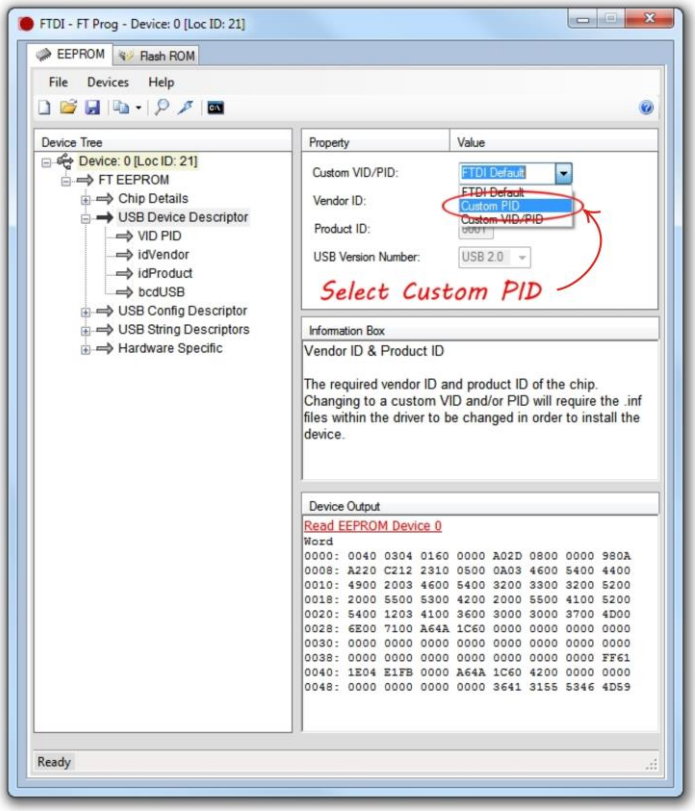

Figure 32 Reprogram DVSI Product ID

7. Support

7.1. DVSI Contact Information

If you have, questions about the USB-3000™ please contact:

Digital Voice Systems, Inc.
234 Littleton Road
Westford, MA 01886 USA

Phone: (978) 392-0002
Fax: (978) 392-8866

Email: info@dvsinc.com
Web: www.dvsinc.com

Support engineers are available
Monday through Friday, 9:00 AM to 5:00 PM eastern time and can be contacted by:
Phone: (978) 392-0002
Fax: (978) 392-8866
Email: info@dvsinc.com
World Wide Web: http://www.dvsinc.com

7.2. Table of Revisions

History of Revisions			
Revision Number	Date of Revision	Description	Page
1.0	Oct.2010	Initial Release	
1.1	Nov. 2010	Revised Section 2.3 Copying the USB-300x™ Software disk to the PC	4
1.2	March 2011	Edited D-Star Control words Table 17 Custom Rate Control Words	36
1.3	March 2011	Added the availability of Windows-64 bit drivers.	
1.4	September 2011	Added section 2.4 USB-300x™ USB Driver Description	5
1.5	October 2011	Added description of the USB-3000™ P25 and USB-3000™ P25-OEM	various
1.6	February 2012	Added section 6.4 How to check the USB-300x™	40
		Added description of the TerreStar / GlobalStar mobile satellite network rates	various
1.6a	May 2012	Edited 8650 to 8F50 typo in Section 6.4	40
1.7	September 2014	Removed references to OEM version now only shipping one version that uses FTDI drivers	various
		Added information about the USB-3003™	

NOTES