Application Report
SPRA931 – August 2003

Audio Echo on the DM642 EVM

Video and Imaging Systems

ABSTRACT

The software demonstrates an audio loopback example on the DM642 EVM, with programmable echo added to the input signal. The demonstration creates and primes the audio I/O streams, initializes the echo buffer, enters a loop reading audio samples, then processes the samples and writes them back into the audio codec.

The demonstration:

- Computes the read pointer and adds a sample on each audio frame
- Reads, clamps, writes, and copies each sample
- Wraps the pointer back to the start of the buffer when the end of the echo buffer is reached

Contents

Software Architecture **1**
System Requirements/Configuration **3**
Software Requirements 3
Hardware Requirements 3
Hardware Setup **4**
Demonstration Execution **4**
Controlling Echo Parameters 5
Example Code Build Procedure **5**
Build Procedure 5
Known Limitations 6

Figures

Figure 1. Simple Process Flow **2**
Figure 2. Hardware Setup **4**
Figure 3. Directory Structure **5**

Software Architecture

This example shows an audio loopback task, collecting stereo audio samples at 48K samples/second, adding a variable delay and attenuation echo to each sample, and playing back the new samples. Figure 1 shows a simple flow diagram of this process.

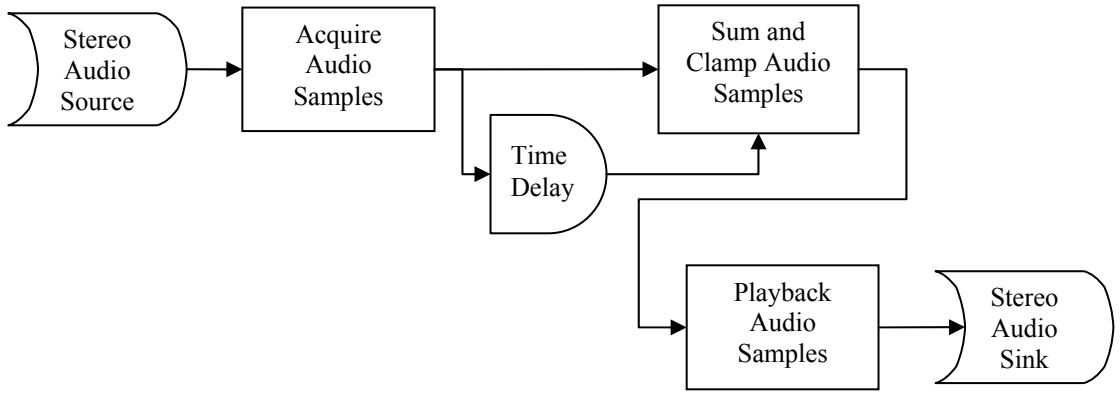

Figure 1. Simple Process Flow

The program consists of a single task, tskAudioDemo(), that performs the following functions:

- Creates the audio I/O streams
- Primes the audio I/O streams
- Initializes the echo buffer: this buffer keeps track of the most recent 1 second worth of audio samples
- Enters an endless loop reading audio samples, processing them, and writing them back out again to the audio codec

Each audio frame read/written contains 10 ms worth of stereo audio samples at 16 bits per channel/sample. The function copyWithEcho() performs the following functions on each audio frame:

- Computes the read pointer for the echo samples in the echo buffer
- For every sample in the input buffer:
  - Reads the input sample and add an attenuated echo sample to it
  - Clamps the resulting sample to a short (16-bit) value
  - Writes the sample with echo to the output buffer
  - Copies the original input sample to the echo buffer for future use
- When the end of the echo buffer is reached, wraps the write pointer back to the start of the buffer

NOTE: Since the echo buffer is an integral number of audio frames, you only need to check for write pointer wrap-around at the end of the copy. This is not true for the echo pointer, which could wrap within an input frame.

System Requirements/Configuration

Software Requirements

- Microsoft Windows NT (SP6), 2000 (SP1 and SP2), or XP
- Code Composer Studio™ Integrated Development Environment (IDE) version 2.20.18 or later
- Device Driver Kit (DDK) version 1.1 or later

Hardware Requirements

- 233-MHz or higher Pentium-compatible CPU (500-MHz or higher Pentium II CPU or equivalent is recommended)
- DM642 EVM
- XDS 510 or 560 emulator
- Stereo audio source
- Stereo amplifier or amplified speakers

Hardware Setup

To run the example, the hardware must be set up as shown:

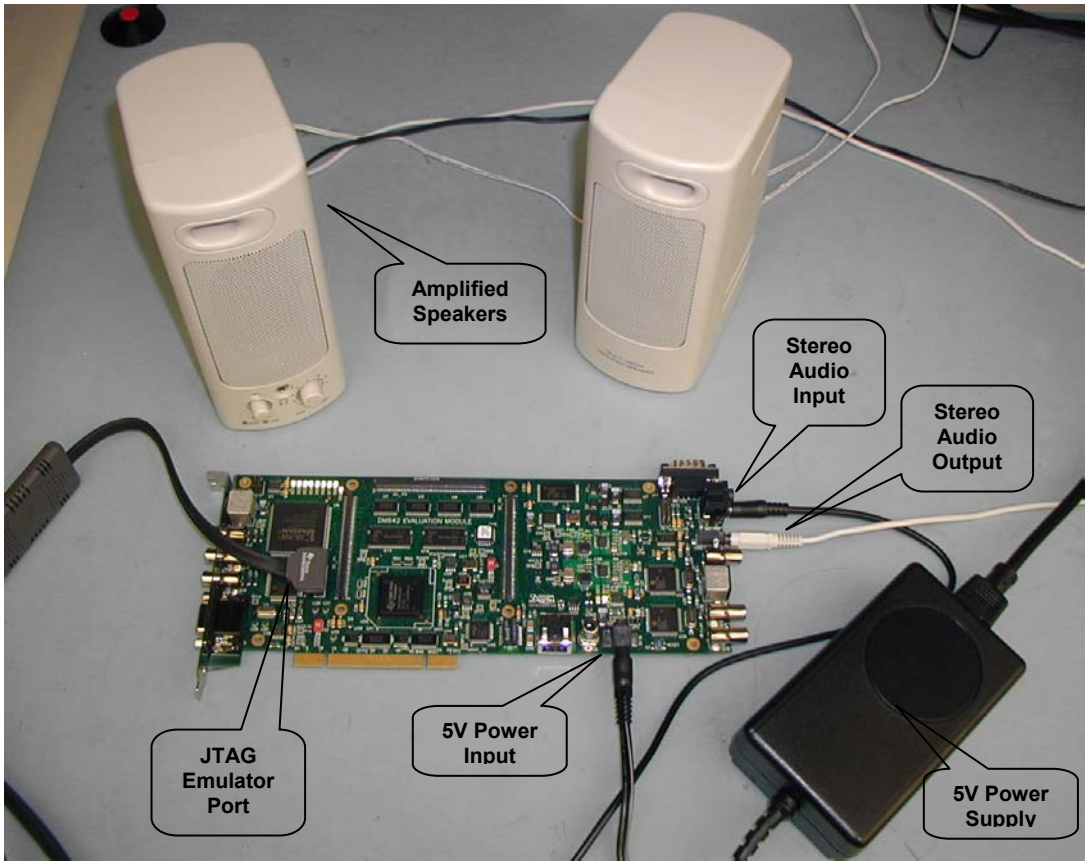

Figure 2. Hardware Setup

- The DM642 EVM must be connected to its 5-V power supply.
- The 5-V power supply must be connected to an appropriate power outlet.
- The XDS510 or 560 emulator must be connected to the JTAG connector to download the example code to the board and control it from Code Composer Studio™ IDE.
- The stereo audio source must be connected to the LINE IN (bottom of J13) connector.
- The speakers must be connected to the LINE OUT connector (J14).

Demonstration Execution

To run the demonstration, make sure that the hardware has been set up as explained in the previous section. Then, follow these steps:

1. Power-up the DM642 EVM board—the board goes through its self test and lights the orange LED at DS9, as well as LEDs DS1 through DS8.

2. Start Code Composer Studio™ IDE on your PC.
3. Reset the DM642 board using GEL -> Resets -> Reset_BreakPts_and_EMIF.
4. Load project `EVMDM642_echo` from the example directory `examples\audio\echo`.
5. Execute the program by pressing F5 or selecting Debug -> Run.
6. You should now hear the audio input signal with echo added to it.

Controlling Echo Parameters

Make sure that the executable has been built with full symbolic debug enabled (option –g):

1. Open a watch window (View -> Watch Window), and select the Watch 1 tab.
2. Add the variables `delayTime` and `echoAtt` to the watch window.
3. You can control the delay time by setting the value of the `delayTime` variable. Select a value between 1 and 999 (milliseconds).
4. You can control the delay sample attenuation by setting the value of the `echoAtt` variable, which contains the attenuation value * 256 —legal values are in the range 0..256, corresponding to a mix value of 0.0 to 1.0.

Example Code Build Procedure

The directory structure is as shown:

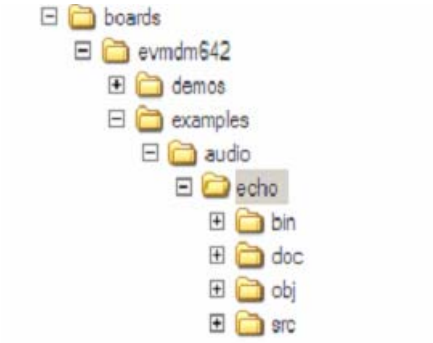

Figure 3. Directory Structure

Build Procedure

Follow these steps:

1. Start Code Composer Studio™ IDE on your PC.
2. Open the evmdm642_echo project (evmdm642_echo.pjt) from the examples\audio\echo folder.
3. Select Project -> Build or Project -> Rebuild All to rebuild the project.

4. Load the executable evmdm642_echo.out from the bin directory.
5. Run using the <F5> key or select Debug -> Run.

Known Limitations

Keep the variable `delayTime` in the range 0..999 and `echoAtt` in the range 0..256. Values outside this range can cause unpredictable results.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265