G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs User’s Guide
Note the following details of the code protection feature on Microchip devices:

• Microchip products meet the specification contained in their particular Microchip Data Sheet.

• Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.

• There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.

• Microchip is willing to work with the customer who is concerned about the integrity of their code.

• Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.
# Table of Contents

Preface ................................................................................................................................. 5

Chapter 1. Introduction
   1.1 Library Overview ........................................................................................................ 11
   1.2 Features .................................................................................................................... 13
   1.3 System Requirements ............................................................................................... 13

Chapter 2. Installation
   2.1 Installation Procedure .............................................................................................. 15
   2.2 Library Files ............................................................................................................. 15

Chapter 3. Quick Start Demonstration
   3.1 Quick Start Demonstration for dsPIC33F Device Family ........................................ 21
   3.2 Quick Start Demonstration for dsPIC33E Device Family ........................................ 30

Chapter 4. Application Programming Interface (API)
   4.1 Adding the Library to the Application ....................................................................... 37
   4.2 Using the Library ..................................................................................................... 37
   4.3 Resource Requirements ........................................................................................... 38
   4.4 Library Interface ..................................................................................................... 39
   4.5 Application Tips ...................................................................................................... 43

Chapter 5. PC Encoding Utility
   5.1 Microphone ............................................................................................................ 45
   5.2 WAVE ( .wav ) File ................................................................................................. 47
   5.3 Output Files ............................................................................................................ 50

Index .................................................................................................................................. 51

Worldwide Sales and Service .............................................................................................. 52
Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a “DS” number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is “DSXXXXXA”, where “XXXXX” is the document number and “A” is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE online help. Select the Help menu, and then Topics to open a list of available online help files.

INTRODUCTION

This preface contains general information that will be useful to know before using the G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs. Items discussed in this chapter include:

• Document Layout
• Conventions Used in this Guide
• Recommended Reading
• The Microchip Web Site
• Development Systems Customer Change Notification Service
• Customer Support
• Document Revision History

DOCUMENT LAYOUT

This document describes how to use the G.711 Speech Encoding/Decoding Library for 16-Bit MCUs and DSCs as a development tool to emulate and debug firmware on a target board. This user’s guide is composed of the following chapters:

• Chapter 1. “Introduction” – This chapter provides a brief overview of the G.711 Speech Encoding/Decoding Library.
• Chapter 2. “Installation” – This chapter provides detailed information needed to install the G.711 Speech Encoding/Decoding Library on a PC.
• Chapter 3. “Quick Start Demonstration” – This chapter describes the demonstrations available for the G.711 Speech Encoding/Decoding Library.
• Chapter 4. “Application Programming Interface (API)” – This chapter outlines how the Application Programming Interface (API) functions provided in the G.711 Speech Encoding/Decoding Library can be included in your application software through the API.
CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

<table>
<thead>
<tr>
<th>DOCUMENTATION CONVENTIONS</th>
<th>Description</th>
<th>Represents</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arial font:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Italic characters</td>
<td>Referenced books</td>
<td><strong>MPLAB® IDE User’s Guide</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Emphasized text</td>
<td>...is the only compiler...</td>
<td></td>
</tr>
<tr>
<td>Initial caps</td>
<td>A window</td>
<td>the Output window</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A dialog</td>
<td>the Settings dialog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A menu selection</td>
<td>select Enable Programmer</td>
<td></td>
</tr>
<tr>
<td>Quotes</td>
<td>A field name in a window or dialog</td>
<td>“Save project before build”</td>
<td></td>
</tr>
<tr>
<td>Underlined, italic text with right angle bracket</td>
<td>A menu path</td>
<td><strong>File&gt;Save</strong></td>
<td></td>
</tr>
<tr>
<td>Bold characters</td>
<td>A dialog button</td>
<td>Click <strong>OK</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A tab</td>
<td>Click the <strong>Power</strong> tab</td>
<td></td>
</tr>
<tr>
<td>Text in angle brackets &lt; &gt;</td>
<td>A key on the keyboard</td>
<td>Press &lt;Enter&gt;, &lt;F1&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>Courier New font:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plain Courier New</td>
<td>Sample source code</td>
<td>#define START</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Filenames</td>
<td>autoexec.bat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>File paths</td>
<td>C:\mcc18\h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keywords</td>
<td>_asm, _endasm, static</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command-line options</td>
<td>-Opa+, -Opa-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit values</td>
<td>0, 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constants (in source code)</td>
<td>0xFF, ‘A’</td>
<td></td>
</tr>
<tr>
<td>Italic Courier New</td>
<td>A variable argument</td>
<td>file.o, where file can be any valid filename</td>
<td></td>
</tr>
<tr>
<td>Square brackets [ ]</td>
<td>Optional arguments</td>
<td>mcc18 [options] file [options]</td>
<td></td>
</tr>
<tr>
<td>Curly brackets and pipe character: {}</td>
<td>Choice of mutually exclusive arguments; an OR selection</td>
<td>errorlevel {0</td>
<td>1}</td>
</tr>
<tr>
<td>Ellipses...</td>
<td>Replaces repeated text</td>
<td>var_name [ , var_name...]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Represents code supplied by user</td>
<td>void main (void) { ... }</td>
<td></td>
</tr>
</tbody>
</table>
WARRANTY REGISTRATION

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in the Warranty Registration Card entitles users to receive new product updates. Interim software releases are available at the Microchip web site.

RECOMMENDED READING

This user’s guide describes how to use the G.711 Speech Encoding/Decoding Library for 16-Bit MCUs and DSCs. The following Microchip documents are available from the Microchip web site (www.microchip.com), and are recommended as supplemental reference resources.

dsPIC30F Family Reference Manual (DS70046)

Refer to this document for detailed information on dsPIC30F device operation. This reference manual explains the operation of the dsPIC30F Digital Signal Controller (DSC) family architecture and peripheral modules but does not cover the specifics of each device. Refer to the appropriate device data sheet for device-specific information.

dsPIC33F/PIC24H Family Reference Manual Sections

Refer to these documents for detailed information on dsPIC33F/PIC24H device operation. These reference manual sections explain the operation of the dsPIC33F/PIC24H DSC family architecture and peripheral modules, but do not cover the specifics of each device. Refer to the specific device data sheet for device-specific information.

PIC24F Family Reference Manual Sections

Refer to these documents for detailed information on PIC24F device operation. These reference manual sections explain the operation of the PIC24F MCU family architecture and peripheral modules, but do not cover the specifics of each device. Refer to the specific device data sheet for device-specific information.

dsPIC33E/PIC24E Family Reference Manual Sections

Refer to these documents for detailed information on dsPIC33E/PIC24E device operation. These reference manual sections explain the operation of the dsPIC33E/PIC24E DSC family architecture and peripheral modules, but do not cover the specifics of each device. Refer to the specific device data sheet for device-specific information.

16-bit MCU and DSC Programmer’s Reference Manual (DS70157)

This manual is a software developer’s reference for the 16-bit Microcontrollers (MCUs) and Digital Signal Controllers (DSC). It describes the instruction set in detail and also provides general information to assist in developing software for 16-bit MCUs and DSCs.

MPLAB® Assembler, Linker and Utilities for PIC24 MCUs and dsPIC® DSCs User’s Guide (DS51317)

MPLAB Assembler for PIC24 MCUs and dsPIC® DSCs (formerly MPLAB ASM30) produces relocatable machine code from symbolic assembly language for the dsPIC DSC and PIC24 MCU device families. The assembler is a Windows console application that provides a platform for developing assembly language code. The assembler is a port of the GNU assembler from the Free Software Foundation (www.fsf.org).
MPLAB® C Compiler for PIC24 MCUs and dsPIC® DSCs User’s Guide (DS51284)

This document describes the features of the optimizing C compiler, including how it works with the assembler and linker. The assembler and linker are discussed in detail in the “MPLAB® Assembler, Linker and Utilities for PIC24 MCUs and dsPIC® DSCs User’s Guide” (DS51317).

Readme Files

For the latest information on using other tools, read the tool-specific Readme files in the Readme subdirectory of the MPLAB IDE installation directory. The Readme files contain updated information and known issues that may not be included in this user’s guide.

THE MICROCHIP WEB SITE

Microchip provides online support through our web site at: http://www.microchip.com. This web site makes files and information easily available to customers. Accessible by most Internet browsers, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listings
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listings of seminars and events; and listings of Microchip sales offices, distributors and factory representatives

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com, click **Customer Change Notification** and follow the registration instructions.

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the MPLAB® C compiler; MPASM™ and MPLAB 16-bit assemblers; MPLINK™ and MPLAB 16-bit object linkers; and MPLIB™ and MPLAB 16-bit object librarians.
- **Emulators** – The latest information on the Microchip MPLAB REAL ICE™ In-Circuit Emulator.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 3.
- **MPLAB IDE** – The latest information on Microchip MPLAB IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB SIM simulator, MPLAB IDE Project Manager and general editing and debugging features.
- **Programmers** – The latest information on Microchip programmers. These include the MPLAB PM3 device programmer and the PICkit™ 3 development programmers.
CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at:
http://www.microchip.com/support

DOCUMENT REVISION HISTORY

Revision A (July 2011)

This is the initial release of the G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs User’s Guide.
Chapter 1. Introduction

This chapter introduces the G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs. This library provides functionality to compress a speech signal, which is useful in applications that have limited memory or communication resources. This library supports the dsPIC33F and dsPIC33E Digital Signal Controller (DSC) device families, as well as the PIC24 Microcontroller (MCU) device families.

This user’s guide provides information that you can use to incorporate the G.711 Speech Encoding/Decoding Library into your embedded solution.

The following topics are covered in this chapter:

- Library Overview
- Features
- System Requirements

1.1 LIBRARY OVERVIEW

The G.711 Speech Encoding/Decoding Library reduces the number of bytes required to represent a speech frame. This reduction, or compression of speech data, is specified by the compression ratio. Figure 1-1 shows a typical application of the G.711 Speech Encoding/Decoding Library.

The communication terminal shown in Figure 1-1 uses the G.711 Speech Encoder to perform speech compression. The communication terminal can then store the compressed data in Flash program memory and transmit the compressed data over a communication link to a remote terminal. The communication terminal decompresses the compressed speech data that is received from the remote terminal by using the G.711 Speech Decoder, and outputs the decompressed speech signal to a local speaker. By using the G.711 Speech Encoding/Decoding Library in such an application, the amount of memory required to store speech data and the communication bandwidth requirement is reduced significantly.
The G.711 Speech Encoding/Decoding Library operates at a sampling rate of 8 kHz, and is suitable for the following applications:

- Voice recording and playback
- Voice over Internet Protocol (VoIP)
- Communication
- Automated announcement systems
- Intercom
- Walkie-talkie
- Any application using message playback

The G.711 Speech Encoding/Decoding Library is based on the ITU-T G.711 recommendation, which uses the basic A-law and μ-law companding techniques. The library is appropriate for both half-duplex and full-duplex systems. The library is written in the Assembly language.

The G.711 Speech Encoding/Decoding Library operates at a sampling rate of 8 kHz and supports a 64 kbps output data rate.

The library Application Programming Interface (API) provides encode and decode functions for both A-law and μ-law companding techniques. The application can use the encoder and decoder independent of each other, or use them together. Multiple instances of the encoder and decoder can also be instantiated.

The G.711 Speech Encoding/Decoding Library also contains a PC-based Encoder Utility (PCEU), which allows speech to be recorded on a PC (through the PC’s microphone port), and converted to G.711 encoded files. These files can then be included into the user application as a part of the application code, thereby allowing encoded speech segments to be stored in the device Flash program memory, or in data RAM. The PCEU can also encode prerecorded WAVE (.wav) files and can accept multiple files as input.
1.2 FEATURES

The G.711 Speech Encoding/Decoding Library has the following features:

- Simple user interface – only one library source file and one header file
- All functions are called from a C application program
- Full compliance with the Microchip C30 Compiler, Assembler and Linker
- Highly optimized assembly code that can be executed on 16-bit MCUs and DSCs
- Compact and concise API for easier integration with application
- Compression ratio of 2:1
- Library package contains source file
- G.711 PCEU for recording and encoding speech through a PC
- Audio bandwidth: 0 kHz to 4 kHz at a sampling rate of 8 kHz
- Can be integrated with Microchip’s Noise Suppression, Acoustic Echo Cancellation and Line Echo Cancellation Libraries
- Library includes:
  - Sample demonstration applications with complete source code
  - User’s guide
  - HTML help files for PCEU

1.3 SYSTEM REQUIREMENTS

The G.711 Speech Encoding/Decoding Library package installation requires a PC-compatible system with these attributes:

- 1 GHz or higher processor
- HTML browser
- 16 MB RAM (minimum)
- 40 MB available hard drive space
- Sound card
Chapter 2. Installation

This chapter describes the various files in the G.711 Speech Encoding/Decoding Library, and includes instructions for installing the library on your laptop or PC for use with programming tools.

The following topics are covered in this chapter:

• Installation Procedure
• Library Files

2.1 INSTALLATION PROCEDURE

1. Double-click G711 setup.exe. The License Agreement screen appears.
2. Review the License Agreement and click I Agree to continue. The Installation Destination dialog appears.
3. Specify the location (i.e., a directory) where the library should be installed, and then click Install.
4. Click Close to close the dialog. This completes the installation.

The installation creates the folder, G.711 v2.0, which contains the files described in 2.2 "Library Files".

2.2 LIBRARY FILES

The G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs CD or zip archive file creates a directory, G.711 v2.0. The directory contains these folders:

• demos
• docs
• h
• PCEU
• Library Source
2.2.1 demos Folder

The demos folder contains application code examples, which demonstrate the use of library in different application scenarios.

The demos folder contains the following sub-folders:

- Communication
- Playback
- RecordPlay

2.2.1.1 Communication FOLDER

The Communication folder contains files, which demonstrate the use of the G.711 Speech Encoding/Decoding Library in a communication setup that consists of two development boards communicating speech data over a serial link. Table 2-1 describes the files in this folder.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsPIC33E G711 Communication Demo 1.hex</td>
<td>dsPIC33E demonstration hexadecimal file for board 1.</td>
</tr>
<tr>
<td>dsPIC33E G711 Communication Demo 2.hex</td>
<td>dsPIC33E demonstration hexadecimal file for board 2.</td>
</tr>
<tr>
<td>dsPIC33E G711 Communication Demo.mcp</td>
<td>dsPIC33E demonstration MPLAB Project file.</td>
</tr>
<tr>
<td>dsPIC33F G711 Communication Demo 1.hex</td>
<td>dsPIC33F demonstration hexadecimal file for board 1.</td>
</tr>
<tr>
<td>dsPIC33F G711 Communication Demo 2.hex</td>
<td>dsPIC33F demonstration hexadecimal file for board 2.</td>
</tr>
<tr>
<td>dsPIC33F G711 Communication Demo.mcp</td>
<td>dsPIC33F demonstration MPLAB Project file.</td>
</tr>
<tr>
<td>cleanup.bat</td>
<td>A batch file script for cleaning up intermediate build files.</td>
</tr>
<tr>
<td>h\Explorer16.h</td>
<td>C header file for Explorer 16 Development Board routines.</td>
</tr>
<tr>
<td>h\MEB.h</td>
<td>C header file for Multimedia Expansion Board (MEB) routines.</td>
</tr>
<tr>
<td>h\G711.h</td>
<td>C header file defining the interface to the G.711 Library.</td>
</tr>
<tr>
<td>h\UART2Drv.h</td>
<td>C header file defining the interface to the UART driver.</td>
</tr>
<tr>
<td>h\WM8510CodecDrv.h</td>
<td>C header file defining the interface to the WM8510 codec driver.</td>
</tr>
<tr>
<td>h\WM8731CodecDrv.h</td>
<td>C header file defining the interface to the WM8731 codec driver.</td>
</tr>
<tr>
<td>src\Explorer16.c</td>
<td>C source file containing routines for the Explorer 16 Development Board.</td>
</tr>
<tr>
<td>src\MEB.c</td>
<td>C source file containing routines for the MEB.</td>
</tr>
<tr>
<td>src\main.c</td>
<td>C source file containing the main speech processing routine.</td>
</tr>
<tr>
<td>src\UART2Drv.c</td>
<td>C source file containing the driver routines for UART2.</td>
</tr>
<tr>
<td>src\WM8731CodecDrv.c</td>
<td>C source file containing the driver routines for WM8731 codec.</td>
</tr>
<tr>
<td>src\WM8510CodecDrv.c</td>
<td>C source file containing the driver routines for WM8510 codec.</td>
</tr>
</tbody>
</table>
2.2.1.2  Playback FOLDER

The Playback folder contains demonstration files that can be used to show the use of G.711 Speech Encoding/Decoding Library, to implement a playback-only system. The demonstration will playback encoded speech data stored in Flash program memory. Table 2-2 describes the files in this folder.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsPIC33E G711 Playback Demo.hex</td>
<td>dsPIC33E demonstration hexadecimal file.</td>
</tr>
<tr>
<td>dsPIC33F G711 Playback Demo.hex</td>
<td>dsPIC33F demonstration hexadecimal file.</td>
</tr>
<tr>
<td>dsPIC33E G711 Playback Demo.mcp</td>
<td>dsPIC33E MPLAB Project file.</td>
</tr>
<tr>
<td>dsPIC33F G711 Playback Demo.mcp</td>
<td>dsPIC33F MPLAB Project file.</td>
</tr>
<tr>
<td>cleanup.bat</td>
<td>A batch file script for cleaning up intermediate build files.</td>
</tr>
<tr>
<td>h\Explorer16.h</td>
<td>C header file for Explorer 16 Development Board routines.</td>
</tr>
<tr>
<td>h\MEB.h</td>
<td>C header file for MEB routines.</td>
</tr>
<tr>
<td>h\G711.h</td>
<td>C header file defining the interface to the G.711 Library.</td>
</tr>
<tr>
<td>h\PgmMemory.h</td>
<td>C header file defining the interface to routines to read Flash program memory.</td>
</tr>
<tr>
<td>h\WM8731CodecDrv.h</td>
<td>C header file defining the interface to the WM8731 codec driver.</td>
</tr>
<tr>
<td>h\WM8510CodecDrv.h</td>
<td>C header file defining the interface to the WM8510 codec driver.</td>
</tr>
<tr>
<td>src\Explorer16.c</td>
<td>C source file containing routines for the Explorer 16 Development Board.</td>
</tr>
<tr>
<td>src\MEB.c</td>
<td>C source file containing routines for the MEB.</td>
</tr>
<tr>
<td>src\main.c</td>
<td>C source file containing the main speech processing routine.</td>
</tr>
<tr>
<td>src\SpeechSegment.s</td>
<td>Encoded Speech Segment file.</td>
</tr>
<tr>
<td>src\PgmMemory.s</td>
<td>Assembly source code containing the routines to read from Flash program memory.</td>
</tr>
<tr>
<td>src\WM8731CodecDrv.c</td>
<td>C source file containing the driver routines for the WM8731 codec.</td>
</tr>
<tr>
<td>src\WM8510CodecDrv.c</td>
<td>C source file containing the driver routines for the WM8510 codec.</td>
</tr>
</tbody>
</table>
2.2.1.3 RecordPlay FOLDER

The RecordPlay demonstration shows the use of the G.711 Speech Encoding/Decoding Library, to implement a voice recorder system. The demonstration will encode speech data, store it in external serial Flash memory, and then playback this data when requested. Table 2-3 describes the files in this folder.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsPIC33E G711 RecordPlay Demo.hex</td>
<td>dsPIC33E demonstration hexadecimal file.</td>
</tr>
<tr>
<td>dsPIC33F G711 RecordPlay Demo.hex</td>
<td>dsPIC33F demonstration hexadecimal file.</td>
</tr>
<tr>
<td>dsPIC33E G711 RecordPlay Demo.mcp</td>
<td>dsPIC33E demonstration MPLAB Project file.</td>
</tr>
<tr>
<td>dsPIC33F G711 RecordPlay Demo.mcp</td>
<td>dsPIC33F demonstration MPLAB Project file.</td>
</tr>
<tr>
<td>cleanup.bat</td>
<td>A batch file script for cleaning up intermediate build files.</td>
</tr>
<tr>
<td>h\Explorer16.h</td>
<td>C header file for Explorer 16 Development Board routines.</td>
</tr>
<tr>
<td>h\MEB.h</td>
<td>C header file for MEB.</td>
</tr>
<tr>
<td>h\G711.h</td>
<td>C header file defining the interface to the G.711 Library.</td>
</tr>
<tr>
<td>h\SST25VF040BDrv.h</td>
<td>C header file defining the interface to routines for SST25VF040B serial Flash memory.</td>
</tr>
<tr>
<td>h\SST25VF016BDrv.h</td>
<td>C header file defining the interface to routines for SST25VF016B serial Flash memory.</td>
</tr>
<tr>
<td>h\WM8510CodecDrv.h</td>
<td>C header file defining the interface to the WM8510 codec driver.</td>
</tr>
<tr>
<td>h\WM8731CodecDrv.h</td>
<td>C header file defining the interface to the WM8731 codec driver.</td>
</tr>
<tr>
<td>src\Explorer16.c</td>
<td>C source file containing routines for the Explorer 16 Development Board.</td>
</tr>
<tr>
<td>src\MEB.c</td>
<td>C source file containing routines for the MEB.</td>
</tr>
<tr>
<td>src\main.c</td>
<td>C source file containing the main speech processing routine.</td>
</tr>
<tr>
<td>src\SpeechSegment.s</td>
<td>Encoded Speech Segment file.</td>
</tr>
<tr>
<td>src\SST25VF040BDrv.c</td>
<td>C source file containing the routines for SST25VF040B serial Flash memory.</td>
</tr>
<tr>
<td>src\SST25VF016BDrv.c</td>
<td>C source file containing the routines for SST25VF016B serial Flash memory.</td>
</tr>
<tr>
<td>src\WM8510CodecDrv.c</td>
<td>C source file containing the driver routines for the WM8510 codec.</td>
</tr>
<tr>
<td>src\WM8731CodecDrv.c</td>
<td>C source file containing the driver routines for the WM8731 codec.</td>
</tr>
</tbody>
</table>
2.2.2  **docs Folder**

The **docs** folder contains the user's guide for the G.711 Speech Encoding/Decoding Library. To view the document, double-click the file name. The user's guide can also be downloaded from the Microchip web site ([www.microchip.com](http://www.microchip.com)).

2.2.3  **h Folder**

The **h** folder contains the C header files, which define the Application Programming Interface (API). **Table 2-4** describes the files in this folder.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711.h</td>
<td>Include file that contains the interface to the G.711 Speech Encoding/Decoding Library. This file must be included in the application to use any of the four modes.</td>
</tr>
</tbody>
</table>

2.2.4  **Library Source Folder**

The **Library Source** folder contains the source file, **G711.s**, for the G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs. The same source file can be used for either dsPIC DSC or PIC24 MCU.

2.2.5  **PCEU Folder**

The **PCEU** folder contains files required by G.711 PC Encoding Utility (PCEU). **Table 2-5** describes the files in this folder.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsPICSpeechRecord.exe</td>
<td>PCEU executable file. Double-click this file to start the G.711 PCEU utility.</td>
</tr>
<tr>
<td>SpeechRecord_G711.dll</td>
<td>DLL files required by the PCEU.</td>
</tr>
<tr>
<td>SpeechRecord_G726.dll</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3. Quick Start Demonstration

This chapter describes the quick start demonstration code examples for the dsPIC33F and dsPIC33E device families, which are part of the G.711 Speech Encoding/Decoding Library package.

3.1 QUICK START DEMONSTRATION FOR dsPIC33F DEVICE FAMILY

The following demonstration examples are covered in this section:

- Communication Demonstration
- Playback Demonstration
- Record Play Demonstration

3.1.1 Communication Demonstration

The Communication demonstration shows how the G.711 Speech Encoding/Decoding Library can be used to reduce the required bandwidth in a full-duplex communication type of application running on a dsPIC33F device.

3.1.1.1 DEMONSTRATION SUMMARY

The Communication code example requires the use of two Explorer 16 Development Boards with two Audio PICtail™ Plus Daughter Boards (not included with the software package), which are set up as shown in Figure 3-1.

**FIGURE 3-1: dsPIC33F G.711 COMMUNICATION DEMONSTRATION SETUP**

Headsets are connected to the Audio PICtail Plus Daughter Boards. When a user speaks into the headset connected to board 1, the WM8510 codec on the Audio PICtail Plus Daughter Board will sample and convert the data, and provide it to the dsPIC DSC device through the DCI module. The dsPIC DSC device will compress the speech signal using the G.711 encoder, and then transmit the compressed speech frame data through the UART2 module and the RS-232 transceiver to board 2.
The dsPIC DSC device on board 2 receives the encoded frame through the on-board RS-232 transceiver and the device’s UART2 module. The dsPIC DSC device decodes the received frame using the G.711 decoder, and then plays out the signal on the headset through its DCI module and the on-board WM8510 codec.

For a speech signal that is sampled at 8 kHz at 16-bit resolution, the resulting data rate is 128 kbps. To communicate this data to a remote terminal over a communication link, the minimum required communication link bandwidth would be 128 kbps. By encoding the speech signal, the minimum communication link bandwidth would be 64 kbps. This results in a bandwidth reduction of 2x. The code example invokes either the G711Lin2Alaw() or G711Lin2Ulaw() function from the G.711 Speech Encoding/Decoding Library to encode and speech data, and uses either the G711ALaw2Lin() or G711Ulaw2Lin() function to decode the speech data.

3.1.1.2 DEMONSTRATION SETUP

The demonstration application is intended to run on the Explorer 16 Development Board with an Audio PICtail Plus Daughter Board (not included with the software package).

Use the procedure outlined in the following section to set up the demonstration.

3.1.1.2.1 Configure Explorer 16 Development Board and Audio PICtail Plus Daughter Boards

Before applying power, you need to configure the boards:

1. On the Audio PICtail Plus Daughter Board 1, set jumper J4 (O/P SEL) to the codec position and set jumper J8 (LN/MIC) to the microphone position, as shown in Figure 3-2.

2. Insert the Audio PICtail Plus Daughter Board 1 into the Explorer 16 Development Board 1 PICtail Plus socket J5.
3. Connect an output device, such as headphones, to socket J10 on the Audio PICtail Plus Daughter Board 1.
5. Repeat steps 1 though 4 for the second Audio PICtail Plus Daughter Board and Explorer 16 Development Board.
6. Connect one end of the DB9M-DB9M Null Modem Adapter to P1 on Explorer 16 Development Board 1. Then, connect one end of the RS-232 cable to the Null Modem Adapter.
7. Connect the other end of the RS-232 cable to P1 on the Explorer 16 Development Board 2.
8. Once the setup is complete, apply power to the Explorer 16 Development Boards. The setup would be similar to the one shown in Figure 3-3.

FIGURE 3-3: SETUP FOR dsPIC33F G.711 COMMUNICATION DEMONSTRATION
3.1.1.2.2 Programming the dsPIC DSC Device

Use this process to load the G.711 communication demonstration into the dsPIC DSC device on the Explorer 16 Development Board.

1. On your PC, launch MPLAB IDE and open the dsPIC33F G711 Communication Demo.mcp project located in the Communication folder. For more information on using MPLAB IDE, refer to the “MPLAB® IDE, Simulator, Editor User's Guide” (DS51025).

2. Select either File > Import > dsPIC33F G711 Communication Demo 1.hex to import the project hexadecimal file for board 1, or select File > Import > dsPIC33F G711 Communication Demo 2.hex to import the project hexadecimal file for board 2.

3. Select Programmer > Connect to link the MPLAB ICD 3 to the dsPIC DSC target device. The Output window confirms that the MPLAB ICD 3 is ready.

4. Select Programmer > Program. The Output window displays the download process and indicates that the programming has succeeded.

**Note:** The MPLAB REAL ICE™ can be used instead of MPLAB ICD 3.

3.1.1.3 Demonstration Procedure

After the demonstration application has been programmed into both the devices, the application is ready to run.

Use the following procedure to run the demonstration:

1. Press **Reset** on board 1. LED D3 on the board will light up indicating that the board is waiting for synchronization to be complete.

2. Press **Reset** on board 2. Now, LED D4 on both of the boards will light up indicating that the boards are synchronized and communicating encoded speech data frames.

3. Speak into the microphone connected to board 1. This speech will be heard in the output device, such as headphones, connected to board 2 and vice-versa.

3.1.1.4 Demonstration Code Description

The demonstration code runs on a dsPIC DSC device, using the Primary Oscillator as the clock source with the PLL set for 40 MIPS operation.

The file, **main.c**, contains the main function for the demonstration application. This main function allocates all of the variables and arrays in data memory that are needed by the UART and WM8510 codec driver.

The main function calls the **WM8510Init()** function to initialize the DCI module, the I^2^C™ module, the WM8510 codec and the DCI interrupt. The WM8510 codec module acts as a Master, and drives the serial clock and frame synchronization lines. The DCI module acts as a Slave. The DCI module is set for the multi-channel Frame Sync Operating mode, with 16-bit data words and 16 data words or time slots per frame, of which only one transmit slot and one receive slot are used in this demonstration. The dsPIC DSC device will send the control information to the WM8510 codec through the dsPIC DSC device’s I^2^C module. Subsequently, the **WM8510Start()** function is used to enable the DCI module and I^2^C module. The **WM8510SampleRate8KConfig()** function configures the WM8510 codec for a sampling rate of 8 kHz.

The **UART2Init()** function configures the UART2 module on the dsPIC DSC for operation at the baud rate specified in UART2Drv.h. The driver provides a blocking interface through the **UART2Read()** and **UART2Write()** functions. The main processing uses the **UART_ioctl()** function with a UART2_RX_NDATA command to know when a frame of data has been received and when frame transmission has been completed.
The `BoardSynchronize()` function is called to ensure that the two boards are synchronized. The UART driver is checked for a full frame of encoded data. If a frame of data has been received, the `G711Alaw2Lin()` or `G711Ulaw2Lin()` function is called to decode this frame. The decoded speech frame is written to the WM8510 codec driver. The codec driver is polled for a frame of raw speech samples. If the frame is ready, the `G711Lin2Alaw()` or `G711Lin2Ulaw()` function is called. The encoded frame is written to the UART2 driver for transmission.

### Note:

The selection between A-law (default) and µ-law encoding is done at compile time, by uncommenting one of the constants, `ALAW` or `ULAW`, in the `main.c` file. This applies to all of the demonstrations described in this section.

### 3.1.2 Playback Demonstration

The Playback demonstration shows how the G.711 Speech Encoding/Decoding Library can be used in a playback type of application where encoded frames are read from Flash program memory, decoded by the G.711 decoder, and then played out. By storing the encoded speech frames in Flash program memory, the application can minimize external memory usage.

#### 3.1.2.1 DEMONSTRATION SUMMARY

The Playback code example requires the use of one Explorer 16 Development Board with one Audio PICtail Plus Daughter Board (not included with the software package). A headset is connected to the Audio PICtail Plus Daughter Board. The demonstration code reads G.711 encoded audio data, which are stored in Flash program memory. The encoded audio data is then decoded using the G.711 decoder. The decoded speech data is written to the WM8510 codec for playback on the output device, such as headphones.

The encoded speech data is obtained using the G.711 PCEU. The resulting `.s` file generated by the PCEU is included into the project and compiled. The encoded speech data now becomes a part of Flash program memory. The code example invokes the `G711Alaw2Lin()` or `G711Ulaw2Lin()` function from the G.711 Speech Encoding/Decoding Library to decode speech data.

#### 3.1.2.2 DEMONSTRATION SETUP

The demonstration application is intended to run on the Explorer 16 Development Board with an Audio PICtail Plus Daughter Board (not included with the software package).

Use the procedure outlined in the following section to set up the demonstration.
3.1.2.2.1 Configure Explorer 16 Development Board and Audio PICtail Plus Daughter Board

Before applying power, you need to configure the boards:
1. On the Audio PICtail Plus Daughter Board, set jumper J4 (O/P SEL) to the codec position and set jumper J8 (LN/ MIC) to the microphone position.
3. Connect the output device, such as headphones, to socket J10 on the Audio PICtail Plus Daughter Board.
4. Once the setup is complete, apply power to the Explorer 16 Development Board.

3.1.2.2.2 Programming the dsPIC DSC Device

Use this process to load the G.711 Playback demonstration into the dsPIC DSC device on the Explorer 16 Development Board:
1. On your PC, launch MPLAB IDE and open the dsPIC33F G711 Playback Demo.mcp project file located in the Playback folder. For more information on using MPLAB IDE, refer to the “MPLAB® IDE, Simulator, Editor User’s Guide” (DS51025).
2. Select File > Import > dsPIC33F G711 Playback Demo.hex to import the project hexadecimal file.
3. Select Programmer > Connect to link the MPLAB ICD 3 to the dsPIC DSC target device. The Output window confirms that the MPLAB ICD 3 is ready.
4. Select Programmer > Program. The Output window displays the download process and indicates that the programming has succeeded.
5. Connect MPLAB ICD 3 to the Explorer 16 Development Board.
6. Program the dsPIC DSC device on the board.
7. When the program is loaded, disconnect MPLAB ICD 3 from the board (remove the telephone cable from the MPLAB ICD 3 connector).

**Note:** The MPLAB REAL ICE can be used instead of MPLAB ICD 3.

3.1.2.3 DEMONSTRATION PROCEDURE

After the demonstration application has been programmed into both the devices, the application is ready to run.

Use the following procedure to run the demonstration:
1. Press Reset on board 1. LED D3 on the board will light up indicating that the board is waiting for synchronization to be complete.
2. Press Reset on board 2. Now, LED D4 on both the boards will light up indicating that the boards are synchronized and communicating encoded speech data frames.
3. Listen to the decoded speech sample being played back repeatedly on the output device, such as headphones.
3.1.2.4 DEMONSTRATION CODE DESCRIPTION

The demonstration code runs on a dsPIC DSC device, using the Primary Oscillator as the clock source with the PLL set for 40 MIPS operation.

The file, main.c, contains the main function for the demonstration application. This main function allocates all of the variables and arrays in data memory that are needed by the WM8510 codec driver.

The main function calls the WM8510Init() function to initialize the DCI module, the I2C module, the WM8510 codec and the DCI interrupt.

The WM8510 codec acts as a Master and drives the serial clock and frame synchronization lines. The DCI module acts as a Slave. The DCI module is set for the multi-channel Frame Sync Operating mode, with 16-bit data words and 16 data words or time slots per frame, of which only one transmit slot and one receive slot are used in this demonstration. The dsPIC DSC device will send the control information to WM8510 codec through the dsPIC DSC device’s I2C module. Subsequently, the WM8510Start() function is used to enable the DCI module and the I2C module. The WM8510SampleRate8KConfig() configures the WM8510 codec for a sampling rate of 8 kHz.

The PgmMemOpen() function opens a handle to Flash program memory for reading. The main loop reads one frame from the Flash program memory using the PgmMemRead() and passes it to the G711Alaw2Lin() or G711Ulaw2Lin() function for decoding. The decoded speech frame is written to the WM8510 codec driver for playback.

3.1.3 Record Play Demonstration

The G.711 Record Play demonstration shows the use of the G.711 Speech Encoding/Decoding Library in a voice recorder type of application. The demonstration emphasizes the reduction in memory requirement for storing speech data. This demonstration is an example of a half-duplex system.

3.1.4 Demonstration Summary

The Record Play demonstration code example requires the use of one Explorer 16 Development Board with one Audio PICtail Plus Daughter Board (not included with the software package). A headset is connected to the Audio PICtail Plus Daughter Board.

In Record mode, the WM8510 codec on the Audio PICtail Plus Daughter Board will sample and convert the speech signal captured by the microphone and provide it to the dsPIC DSC device through the DCI module. The dsPIC DSC device will compress the speech signal using the G.711 encoder, pack the encoded data, and then store it in serial Flash memory available on the Audio PICtail Plus Daughter Board. In Playback mode, the application reads the packed encoded speech data stored in serial Flash memory, unpacks data, and then decodes this data using the G.711 decoder. The decoded speech data is written to the WM8510 codec for playback on the output device, such as headphones.

For a speech signal that is sampled at 8 kHz at 16-bit resolution, the resulting data rate is 128 kbps. To store one minute of raw speech data in memory would require 960 KB of memory. By encoding the speech signal using G.711 at 64 kbps bit rate, the memory required to store one minute of speech is 480 KB.

The code example invokes the encoder and decoder functions from the G.711 Speech Encoding/Decoding Library to encode and decode speech data.
3.1.4.1 DEMONSTRATION SETUP

The demonstration application is intended to run on the Explorer 16 Development Board with an Audio PICtail Plus Daughter Board (not included with the software package).

Use the procedure outlined in the following section to set up the demonstration.

3.1.4.1.1 Configure Explorer 16 Development Board and Audio PICtail Plus Daughter Board

Before applying power, you need to configure the boards:

1. On the Audio PICtail Plus Daughter Board, set jumper J4 (O/P SEL) to the codec position and set jumper J8 (LN/MIC) to the microphone position.
3. Connect the output device, such as headphones, to socket J10 on the Audio PICtail Plus Daughter Board.
4. Connect a microphone to socket J1 on the Audio PICtail Plus Daughter Board.
5. Once the setup is complete, apply power to the Explorer 16 Development Board.

3.1.4.1.2 Programming the dsPIC DSC Device

Use this process to load the G.711 Playback demonstration into the dsPIC DSC device on the Explorer 16 Development Board.

1. On your PC, launch MPLAB IDE and open the dsPIC33F G711 Record Play Demo.mcp project located in the RecordPlay folder. For more information on using MPLAB IDE, refer to the “MPLAB® IDE, Simulator, Editor User's Guide” (DS51025).
2. Select File > Import > dsPIC33F G711 Record Play Demo.hex to import the project hexadecimal file.
3. Select Programmer > Connect to link the MPLAB ICD 3 to the dsPIC DSC target device. The Output window confirms that the MPLAB ICD 3 is ready.
4. Select Programmer > Program. The Output window displays the download process and indicates that the programming has succeeded.
5. Connect MPLAB ICD 3 to the Explorer 16 Development Board.
6. Program the dsPIC DSC device on the board.
7. When the program is loaded, disconnect MPLAB ICD 3 from the board (remove the telephone cable from the MPLAB ICD 3 connector).

**Note:** The MPLAB REAL ICE can be used instead of MPLAB ICD 3.

3.1.4.2 DEMONSTRATION PROCEDURE

After the demonstration application has been programmed into the device, the application is ready to run.

Use the following procedure to run the demonstration:

1. Press Reset on the board. LED D3 on the board will light up indicating that the application is running.
2. Press switch S3 to start the erase and record process. LED D4 will light up indicating the serial Flash memory is being erased. After the erase is done, LED D4 turns off and LED D5 will switch on indicating that the recording is in progress. The microphone signal will be captured and stored in serial Flash memory.
3. Press switch S6. This will stop the Record mode and start the Playback mode. LED D5 turns off and LED D6 lights up.
4. Press switch S3 to start recording again.
3.1.4.3 DEMONSTRATION CODE DESCRIPTION

The demonstration code runs on a dsPIC DSC device, using the Primary Oscillator as the clock source with the PLL set for 40 MIPS operation.

The file, main.c, contains the main function for the demonstration application. This main function allocates all of the variables and arrays in data memory that are needed by the WM8510 codec driver and the SST25VF040B serial Flash memory driver.

The main function also calls the WM8510Init() function to initialize the DCI module, the I\(^2\)C module, the WM8510 codec and the DCI interrupt. The WM8510 codec module acts as a Master and drives the serial clock and frame synchronization lines. The DCI module acts as a Slave. The DCI module is set for the multi-channel Frame Sync Operating mode, with 16-bit data words and 16 data words or time slots per frame, of which only one transmit slot and one receive slot are used in this demonstration. The dsPIC DSC device will send the control information to WM8510 codec through the dsPIC DSC device’s I\(^2\)C module. Subsequently, the WM8510Start() function is used to enable the DCI module and the I\(^2\)C module. The WM8510SampleRate8KConfig() configures the WM8510 codec for a sampling rate of 8 kHz.

The SST25VF040BInit() function is called to initialize the serial Flash memory. The SST25VF040BStart() function enables the dsPIC DSC device’s SPI™ module, which communicates with the serial Flash memory.

The main function polls the switches to determine if Record mode or Playback mode is selected. If Record mode is selected, the SST25VF040BIOCtl() function is called with the chip erase command. When the chip erase is complete, the current speech frame is encoded using the G711Lin2Alaw() or G711Lin2Ulaw() function and written to the Flash program memory using the SST25VF040BWrite() function. The speech frame is also provided to the WM8510 codec driver for output to the headphones. If Playback mode is selected, the serial Flash memory is read using the SST25VF040BRead() function, decoded using the G711ALaw2Lin() or G711Ulaw2Lin() function, and written to the WM8510 codec driver for output to the headphones.
3.2 QUICK START DEMONSTRATION FOR dsPIC33E DEVICE FAMILY

The following topics are covered in this section:

- Communication Demonstration
- Playback Demonstration
- Record Play Demonstration

3.2.1 Communication Demonstration

The Communication demonstration shows how the G.711 Speech Encoding/Decoding Library can be used to reduce the required bandwidth in a full-duplex communication type of application running on a dsPIC33E device.

3.2.1.1 DEMONSTRATION SUMMARY

The Communication code example requires the use of two Multimedia Expansion Boards (MEBs) with two dsPIC33E USB Starter Kits (not included with the software package), which are set up as shown in Figure 3-4.

FIGURE 3-4: dsPIC33E G.711 COMMUNICATION DEMONSTRATION

SETUP

Headsets are connected to the MEBs. When a user speaks into the headset connected to board 1, the WM8731 codec on the MEB will sample and convert the data, and provide it to the dsPIC DSC device through the DCI module. The dsPIC DSC device will compress the speech signal using the G.711 encoder, and then transmit the compressed speech frame data through the UART2 module and I/O expansion connector to board 2.

The dsPIC DSC device on board 2 receives the encoded frame through the on-board I/O expansion connector and the device’s UART2 module. The dsPIC DSC device decodes the received frame using the G.711 decoder, and then plays out the signal on the headset through its DCI module and the on-board WM8731 codec.

For a speech signal that is sampled at 8 kHz at 16-bit resolution, the resulting data rate is 128 kbps. To communicate this data to a remote terminal over a communication link, the minimum required communication link bandwidth would be 128 kbps. By encoding the speech signal, the minimum communication link bandwidth would be 64 kbps. This results in a bandwidth reduction of 2x.
Quick Start Demonstration

The code example invokes either the `G711Lin2Alaw()` or `G711Lin2Ulaw()` function (for A-law) from the G.711 Speech Encoding/Decoding Library to encode speech data, and uses either the `G711Alaw2Lin()` or `G711Ulaw2Lin()` function to decode speech data.

### 3.2.1.2 DEMONSTRATION SETUP

Use the procedure outlined in the following section to set up the demonstration.

#### 3.2.1.2.1 Configure MEBs and dsPIC33E USB Starter Kits

Before applying power, you need to configure the boards:

1. Insert a dsPIC33E USB Starter Kit into the starter kit connector on MEB 1.
2. Connect the dsPIC33E USB Starter Kit to a PC using the USB A-to-mini-B cable provided with the Starter Kit.
3. Connect the output device, such as headphones, to MEB 1.
4. Connect a microphone to MEB 1.
5. Repeat steps 1 though 4 for the second dsPIC33E USB Starter Kit and MEB.
8. Once the setup is complete, the setup would be similar to the one shown in Figure 3-4.

#### 3.2.1.2.2 Programming the dsPIC DSC Device

Use this process to load the G.711 communication demonstration into the dsPIC DSC device on the dsPIC33E USB Starter Kit.

1. On your PC, launch MPLAB IDE and open the `dsPIC33E G711 Communication Demo.mcp` project located in the Communication folder. For more information on using MPLAB IDE, refer to the “MPLAB® IDE, Simulator, Editor User's Guide” (DS51025).
2. Select either File > Import > dsPIC33E G711 Communication Demo 1.hex to import the project hexadecimal file for board 1, or select File > Import > dsPIC33E G711 Communication Demo 2.hex to import the project hexadecimal file for board 2.
3. Select Programmer > Starter Kit on Board as the programmer, and then select Connect to link to the dsPIC DSC target device. The Output window confirms that the target device is ready.
4. Select Programmer > Program. The Output window displays the download process and indicates that the programming has succeeded.

```
Note: After programming, unplug and reconnect the USB cable to the starter kit, to ensure that the WM8731 audio codec can be reconfigured.
```

5. Repeat steps 2 through 4 for the second board.
3.2.1.3 DEMONSTRATION PROCEDURE

After the demonstration application has been programmed into both the devices, the application is ready to run.

Use the following procedure to run the demonstration:

1. Using the MPLAB IDE, reset and run board 1. LED D1 on the board will light up indicating that the board is waiting for synchronization to complete.
2. Using the MPLAB IDE, reset and run board 2. Now, LED D2 on both the boards will light up indicating that the boards are synchronized and communicating encoded speech data frames.
3. Speak into the microphone connected to board 1. This speech will be heard in the output device, such as headphones, connected to board 2 and vice-versa.

**Note:** If only one PC is available for running the demonstration, then program board 2, disconnect the USB cable from board after programming, and then use an external 9V power supply to reset and run board 2 (after the program in board 1 is already running).

3.2.1.4 DEMONSTRATION CODE DESCRIPTION

The demonstration code runs on a dsPIC DSC device, using the Primary Oscillator as the clock source with the PLL set for 40 MIPS operation.

The file, main.c, contains the main function for the demonstration application. This main function allocates all of the variables and arrays in data memory that are needed by the UART and WM8731 codec driver.

The main function calls the WM8731Init() function to initialize the DCI module, the I²C module, the WM8731 codec and the DCI interrupt. This configures the WM8731 codec for a sampling rate of 8 kHz. The WM8731 codec module acts as a Master and drives the serial clock and frame synchronization lines. The DCI module acts as a Slave. The DCI module is set for the multi-channel Frame Sync Operating mode, with 16-bit data words and two data words or time slots per frame (that is, only one transmit slot and one receive slot are used in this demonstration). The dsPIC DSC device will send the control information to the WM8731 codec through the dsPIC DSC device’s I²C module. Subsequently, the WM8731Start() function is used to enable the DCI module and I²C module.

The UART2Init() function configures the UART2 module on the dsPIC DSC for operation at the baud rate specified in UART2Drv.h. The driver provides a blocking interface through the UART2Read() and UART2Write() functions. The main processing uses the UART_ioctl() function with a UART2_RX_NDATA command to know when a frame of data has been received and when frame transmission has been completed.

The BoardSynchronize() function is called to ensure that the two boards are synchronized. The UART driver is checked for a full frame of encoded data. If a frame of data has been received, the G711Alaw2Lin() or G711Ulaw2Lin() function is called to decode this frame. The decoded speech frame is written to the WM8731 codec driver. The codec driver is polled for a frame of raw speech samples. If the frame is ready, the G711Lin2Alaw() or G711Lin2Ulaw() function is called. The encoded frame is written to the UART2 driver for transmission.

**Note:** The selection between A-law (default) and µ-law encoding is done at compile time, by uncommenting one of the constants, ALAW or ULAW, in the main.c file. This applies to all of the demonstrations described in this section.
3.2.2 Playback Demonstration

The Playback demonstration shows how the G.711 Speech Encoding/Decoding Library can be used in a playback type of application where encoded frames are read from Flash program memory, decoded by the G.711 decoder, and then output. By storing the encoded speech frames in Flash program memory, the application can minimize external memory usage.

3.2.2.1 DEMONSTRATION SUMMARY

The Playback code example requires the use of one MEB with one dsPIC33E USB Starter Kit (not included with the software package). A headset is connected to the MEB. The demonstration code reads G.711 encoded audio data, which are stored in Flash program memory. The encoded audio data is then decoded using the G.711 decoder. The decoded speech data is written to the WM8731 codec for playback on the output device, such as headphones.

The encoded speech data is obtained using the G.711 PCEU. The resulting .s file generated by the PCEU is included into the project and compiled. The encoded speech data now becomes a part of Flash program memory. The code example invokes the G711Decode() function from the G.711 Speech Encoding/Decoding Library to decode speech data.

3.2.2.2 DEMONSTRATION SETUP

The demonstration application is intended to run on the MEB with a dsPIC33E USB Starter Kit (not included with the software package).

Use the procedure outlined in the following section to set up the demonstration.

3.2.2.2.1 Configure MEB and dsPIC33E USB Starter Kit

Before applying power, you need to configure the boards:

1. Insert a dsPIC33E USB Starter Kit into the starter kit connector on the MEB.
2. Connect the dsPIC33E USB Starter Kit to a PC using the USB A-to-mini-B cable provided with the dsPIC33E USB Starter Kit.
3. Connect an output device, such as headphones, to the MEB.

3.2.2.2.2 Programming the dsPIC DSC Device

Use this process to load the G.711 Playback demonstration into the dsPIC DSC device on the dsPIC33E USB Starter Kit:

1. On your PC, launch MPLAB IDE and open the dsPIC33E G711 Playback Demo.mcp project file located in the Playback folder. For more information on using MPLAB IDE, refer to the "MPLAB® IDE, Simulator, Editor User's Guide" (DS51025).
2. Select File > Import > dsPIC33E G711 Playback Demo.hex to import the project hexadecimal file.
3. Select Programmer > Starter Kit on Board as the programmer, and then select Programmer > Connect to link to the dsPIC DSC target device. The Output window confirms that the target device is ready.
4. Select Programmer > Program. The Output window displays the download process and indicates that the programming has succeeded.

**Note:** If the USB cable is already plugged in, then unplug and reconnect the USB cable to the dsPIC33E USB Starter Kit, to ensure that the WM8731 audio codec can be reconfigured.
3.2.2.3 DEMONSTRATION PROCEDURE

After the demonstration application has been programmed into the device, the application is ready to run.

Use the following procedure to run the demonstration:

1. Using the MPLAB IDE, reset and run the board.
2. Listen to the speech sample being played back repeatedly on the output device, such as headphones.

3.2.2.4 DEMONSTRATION CODE DESCRIPTION

The demonstration code runs on a dsPIC DSC device, using the Primary Oscillator as the clock source with the PLL set for 40 MIPS operation.

The file, main.c, contains the main function for the demonstration application. This main function allocates all of the variables and arrays in data memory that are needed by the WM8731 codec driver.

The main function calls the WM8731Init() function to initialize the DCI module, the I2C module, the WM8731 codec and the DCI interrupt. This configures the WM8731 codec for a sampling rate of 8 kHz. The DCI module acts as a Master, and drives the serial clock and frame synchronization lines. The DCI module acts as a Slave. The DCI module is set for the multi-channel Frame Sync Operating mode, with 16-bit data words and two data words or time slots per frame (only one transmit slot and one receive slot are used in this demonstration). The dsPIC DSC device will send the control information to WM8731 codec through the dsPIC DSC device’s I2C module. Subsequently, the WM8731Start() function is used to enable the DCI module and I2C module.

The PgmMemOpen() function opens a handle to Flash program memory for reading. The main loop reads one frame from the Flash program memory using the PgmMemRead() and passes it to the G711Alaw2Lin() or G711Ulaw2Lin() function for decoding. The decoded speech frame is written to the WM8731 codec driver for playback.

3.2.3 Record Play Demonstration

The G.711 Record Play demonstration shows the use of the G.711 Speech Encoding/Decoding Library for 16-bit MCUs and DSCs in a voice recorder type of application. The demonstration emphasizes the reduction in memory requirement for storing speech data. This demonstration is an example of a half-duplex system.

3.2.3.1 DEMONSTRATION SUMMARY

The Record Play demonstration code example requires the use of one MEB with one dsPIC33E USB Starter Kit (not included with the software package). A headset is connected to the MEB.

In Record mode, the WM8731 codec on the MEB will sample and convert the speech signal captured by the microphone, and provide it to the dsPIC DSC device through the DCI module. The dsPIC DSC device will compress the speech signal using the G.711 encoder, pack the encoded data, and then store it in the serial Flash memory available on the MEB. In Playback mode, the application reads the packed encoded speech data stored in serial Flash memory, unpacks the data, and then decodes this data using the G.711 decoder. The decoded speech data is written to the WM8731 codec for playback on the headphones.
For a speech signal that is sampled at 8 kHz at 16-bit resolution, the resulting data rate is 128 kbps. To store one minute of raw speech data in memory would require 960 KB of memory. By encoding the speech signal, the memory required to store one minute of speech is 480 KB. The code example invokes the encoder and decoder functions from the G.711 Speech Encoding/Decoding Library to encode and decode speech data.

3.2.3.2 DEMONSTRATION SETUP

The demonstration application is intended to run on the MEB with a dsPIC33E USB Starter Kit (not included with the software package).

Use the procedure outlined in the following section to set up the demonstration.

3.2.3.2.1 Configure MEB and dsPIC33E USB Starter Kit

Before applying power, you need to configure the boards:

1. Insert a dsPIC33E USB Starter Kit into the starter kit connector on the MEB.
2. Connect the dsPIC33E USB Starter Kit to a PC using the USB A-to-mini-B cable provided with the dsPIC33E USB Starter Kit.
3. Connect an output device, such as headphones, to the MEB.

3.2.3.2.2 Programming the dsPIC DSC Device

Use this process to load the G.711 Playback demonstration into the dsPIC DSC device on the Explorer 16 Development Board.

1. On your PC, launch MPLAB IDE and open the dsPIC33E G711 Playback Demo.mcp project file located in the Playback folder. For more information on using MPLAB IDE, “MPLAB® IDE, Simulator, Editor User’s Guide” (DS51025).
2. Select File > Import > dsPIC33E G711 Playback Demo.hex to import the project hexadecimal file.
3. Select Programmer > Starter Kit on Board as the programmer, and then select Programmer > Connect to link to the dsPIC DSC target device. The Output window confirms that the target device is ready.
4. Select Programmer > Program. The Output window displays the download process and indicates that the programming has succeeded.

Note: If the USB cable is already plugged in, then unplug and reconnect the USB cable to the dsPIC33E USB Starter Kit, to ensure that the WM8731 audio codec can be reconfigured.

3.2.3.3 DEMONSTRATION PROCEDURE

After the demonstration application has been programmed into the device, the application is ready to run.

Use the following procedure to run the demonstration:

1. Using the MPLAB IDE, reset and run the board.
2. Press switch S1 for a few seconds, to start the erase and record process. Release the switch when LED D1 lights up indicating the serial Flash memory is being erased. After the erase is done, LED D1 turns off and LED D2 will switch on indicating that recording is in progress. The microphone signal will be captured and stored in serial Flash memory.
3. Press switch S1 again for a few seconds. This will stop the Record mode and start the Playback mode. Release the switch when LED D2 turns off and LED D3 lights up.
4. To start recording again, press switch S1 again, as described in step 2.
3.2.3.4 DEMONSTRATION CODE DESCRIPTION

The demonstration code runs on a dsPIC DSC device, using the Primary Oscillator as the clock source with the PLL set for 40 MIPS operation.

The file, main.c, contains the main function for the demonstration application. This main function allocates all the variables and arrays in data memory that are needed by the WM8731 codec driver and the SST25VF016B serial Flash memory driver.

The main function calls the WM8731Init() function to initialize the DCI module, the I²C module, the WM8731 codec and the DCI interrupt. This configures the WM8731 codec for a sampling rate of 8 kHz. The WM8731 codec module acts as a Master and drives the serial clock and frame synchronization lines. The DCI module acts as a Slave. The DCI module is set for the multi-channel Frame Sync Operating mode, with 16-bit data words and two data words or time slots per frame (only one transmit slot and one receive slot are used in this demonstration).

The dsPIC DSC device will send the control information to WM8731 codec through the dsPIC DSC device’s I²C module. Subsequently, the WM8510Start() function is used to enable the DCI module and I²C module.

The SST25VF016BInit() function is called to initialize the serial Flash memory. The SST25VF016BStart() function enables the dsPIC DSC device’s SPI module, which communicates with the serial Flash memory.

The main function polls the switches to determine if Record mode or Playback mode is selected. If Record mode is selected, the SST25VF016BIOCtl() function is called with the chip erase command. When the chip erase is complete, the current speech frame is encoded using the G711Lin2Alaw() or G711Lin2Ulaw() functions and written to Flash program memory using the SST25VF016BWrite() function. The speech frame is also provided to the WM8731 codec driver for output to the headphones. If Playback mode is selected, the serial Flash memory is read using the SST25VF016BRead() function, decoded using the G711Alaw2Lin() or G711Ulaw2Lin() function, and then written to the WM8731 codec driver for output to the headphones.
Chapter 4. Application Programming Interface (API)

This chapter describes the Application Programming Interface (API) functions that are available in the G.711 Speech Encoding/Decoding Library.

The following topics are covered in this chapter:
- Adding the Library to the Application
- Using the Library
- Resource Requirements
- Library Interface
- Application Tips

4.1 ADDING THE LIBRARY TO THE APPLICATION

To use the G.711 Speech Encoding/Decoding Library in an application, the library source file, G711.s (located in the Library Source folder of your library installation), must be added to the application project workspace.

To use the library functions, include the G711.h file in the application source code. This file can be copied from the h folder (located in the library installation directory) to the application project folder.

4.2 USING THE LIBRARY

The G.711 Speech Encoding/Decoding Library has been designed to be usable in a re-entrant environment. This feature enables the algorithm to process multiple channels of audio. Since the encoder and the decoder algorithms operate independent of each other, an application does not have to call a decoder or encoder if it is not needed.

There is no state memory or scratch memory required for using this library, and there is no library initialization function. The user simply needs to call the appropriate encoder/decoder function (depending on the choice of encoder, A-law or μ-law), and specify the frame size in the function call.

Note: A speech sample encoded using A-law encoding cannot be decoded using μ-law and vice-versa.
4.3 RESOURCE REQUIREMENTS

The resource requirements for running the G.711 Speech Encoding/Decoding Library are provided in the following sections and tables.

### TABLE 4-1: PROGRAM MEMORY USAGE

<table>
<thead>
<tr>
<th>Resource</th>
<th>Size (bytes)</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-law Encoder</td>
<td>123</td>
<td>.text</td>
</tr>
<tr>
<td>A-law Decoder</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>µ-law Encoder</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>µ-law Decoder</td>
<td>91</td>
<td></td>
</tr>
</tbody>
</table>

The Data memory usage is limited to the memory required for input and output buffers. The size of these buffers is user definable.

### TABLE 4-2: DYNAMIC MEMORY

<table>
<thead>
<tr>
<th>Section</th>
<th>Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heap</td>
<td>0</td>
</tr>
<tr>
<td>Stack</td>
<td>&lt; 60</td>
</tr>
</tbody>
</table>

### TABLE 4-3: COMPUTATIONAL SPEED

<table>
<thead>
<tr>
<th>Function</th>
<th>MIPS</th>
<th>Typical Call Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-law Encoder</td>
<td>0.25</td>
<td>Once every frame</td>
</tr>
<tr>
<td>A-law Decoder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>µ-law Encoder</td>
<td>0.25</td>
<td>Once every frame</td>
</tr>
<tr>
<td>µ-law Decoder</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.4 LIBRARY INTERFACE

This section describes the G.711 Speech Encoding/Decoding Library interface. The following topics are described in this section:

- Encoder Functions
- Decoder Functions

4.4.1 Encoder Functions

Table 4-4 lists the functions related to the G.711 Encoder.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711Lin2Alaw,</td>
<td>The G711Lin2Alaw routine encodes one input frame using A-law compression. The G711Lin2Ulau...</td>
</tr>
<tr>
<td>G711Lin2Ulau</td>
<td>routine encodes one input frame using μ-law compression.</td>
</tr>
</tbody>
</table>
G711Lin2Alaw, G711Lin2Ulaw

Description
This routine encodes one input frame. Irrespective of whether A-law or μ-law is used, each output sample requires one byte. Therefore, the output array should be a signed character array of the same size as the input array.

Include

G711.h

Prototypes

```c
void G711Lin2Alaw (int * input, char * output, int size);
void G711Lin2Ulaw (int * input, char * output, int size);
```

Preconditions
None.

Arguments

inputFrame input array containing samples to be encoded. The samples should be signed integer values. The size of the array should match the encoder frame size specified in the function call.

outputFrame output array where the encoded samples will be stored. Each byte of output array will store one encoded sample.

size processing frame size (minimum of 1).

Return Value
None.

Remarks
None.

Code Example

```c
#define G711_FRAME_SIZE 80
int input[G711_FRAME_SIZE];
char output[G711_FRAME_SIZE];

G711Lin2Alaw (input, output, G711_FRAME_SIZE);
```
4.4.2 Decoder Functions

Table 4-5 lists the functions related to the G.711 decoder.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711Ulaw2Lin</td>
<td>The G711Ulaw2Lin routine decodes one input frame using µ-law decompression.</td>
</tr>
</tbody>
</table>

G711Alaw2Lin, G711Ulaw2Lin

Description
This routine decodes one input frame. The input array should be a signed character array of the same size as the output array.

Include
G711.h

Prototype
void G711Alaw2lin (char * inputFrame, int * outputFrame, int * size);
void G711Ulaw2lin (char * inputFrame, int * outputFrame, int * size);

Preconditions
None.

Arguments
inputFrame  input array containing samples to be decoded. The encoded samples should be arranged such that one element of the array holds one encoded sample.
outputFrame output array where the decoded samples will be stored. The decoded value is a signed integer value.
size processing frame size (minimum value of 1).

Return Value
None.

Remarks
None.

Code Example
#define G711_FRAME_SIZE 80

int output[G711_FRAME_SIZE];
char input[G711_FRAME_SIZE];

G711Alaw2lin(input, output, G711_FRAME_SIZE);
4.5 APPLICATION TIPS

The following are a few tips to consider when using the G.711 Speech Encoding/Decoding Library.

1. The optimum input signal levels for testing audio systems are generally considered to be between -10 dBm0 and -30 dBm0. If the digital input speech levels have peaks that are up to three-fourth of full range, good use is being made of the available precision; levels higher than this carry a risk of amplitude clipping.

2. The choice between A-law and µ-law encoding in telephony applications is often determined by regional and international telecommunication standards.
Chapter 5. PC Encoding Utility

The G.711 Speech Encoding/Decoding Library package includes a PC Encoding Utility (PCEU) called the dsPIC33F Speech Encoding Utility, which allows users to encode a microphone signal or a WAVE (.wav) file and generate a file containing G.711 encoded samples. This generated file can then be readily included in the user application to store the G.11 encoded samples either in the device Flash program memory or RAM. This is suitable for Playback type of applications where a prerecorded message is played back on the output device. The dsPIC33F Speech Encoding Utility is available in the PCEU folder of the library installation directory.

The following sections provide the steps to be followed, to encode a microphone signal connected to the PC sound card and to encode a .wav file.

- Microphone
- WAVE (.wav) File
- Output Files

**Note:** The dsPIC33F Speech Encoding Utility can be used to generate the encoded speech for either dsPIC33F or dsPIC33E.

### 5.1 MICROPHONE

This section provides the steps to be followed to use the dsPIC33F Speech Encoding Utility, to encode a signal from the microphone connected to the PC sound card.

1. Connect a compatible microphone to the microphone input of the PC sound card.
   Use any of the available system utilities to ensure nominal microphone gain.
2. Double-click `dsPICSpeechRecord.exe` in the PCEU folder of the library installation directory to start the utility.

**FIGURE 5-1: dsPIC33F SPEECH ENCODER UTILITY**
3. Select Input > Mic to select the microphone connected to the PC sound card.

**FIGURE 5-2: INPUT SOURCE SELECTION**

4. Select Input > G711 to select the G.711 encoder, to encode the microphone signal.

5. Select either Target Memory > Program Memory to store the encoded speech in Flash program memory, or select Target Memory > RAM to store the encoded speech in RAM. The generated encoded samples can be either stored in device RAM or in Flash program memory. Depending on the selected option, the PCEU either generates a .c file for RAM, or generates a .s file for Flash program memory.

**FIGURE 5-3: TARGET MEMORY SELECTION**

6. Select Output > Array name to specify the name of the encoded samples array in the generated file.

**FIGURE 5-4: ARRAY NAME SELECTION**

7. Select Output > Filename to specify the name of the generated file and the path where the generated file will be stored.

**FIGURE 5-5: FILE NAME SELECTION**
8. Select **Options > Law** to specify the G.711 encoding scheme (A-law or µ-law) to be used while generating the encoded samples.

9. Click **Record**. The recording of the microphone signal starts and the user can talk into the microphone. The signals will be encoded using the G.711 encoder. The timer on the PCEU interface will start counting the length of the speech segment.

10. Click **Stop** to stop the recording. The encoding process is complete and the generated files are available for inclusion into a dsPIC DSC application. Refer to **5.3 “Output Files”** for more information on the generated file.

### 5.2 WAVE (.wav) FILE

This section provides the steps to use the dsPIC33F Speech Encoding Utility, to encode a .wav file. Note that the .wav file should be in the 16-bit at 8 kHz format.

1. Double-click dsPICSpeechRecord.exe in the PCEU folder of the library installation directory to start the utility.

**FIGURE 5-6: dsPIC33F SPEECH ENCODER UTILITY**

![Current Encoder Settings](image)

- **Input:** Online Encoding
- **Output File:** Speech.g71
- **Output Array:** G711_data
- **Target Memory:** Program Memory
- **LAW:** A-Law

2. Select **Input > Speech File** to select a .wav file as the source input for the utility.

**FIGURE 5-7: INPUT SOURCE SELECTION**

![Input Source Selection](image)

3. Select **Input > G.711** to select the G.711 encoder, to encode the microphone signal.
4. Select either Target Memory > Program Memory to store the encoded speech in Flash program memory, or select Target Memory > RAM to store the encoded speech in RAM. The generated encoded samples can be either stored in device RAM or in Flash program memory. Depending on the selected option, the PCEU either generates a .c file for RAM, or a .s file for Flash program memory.

**FIGURE 5-8:** TARGET MEMORY SELECTION

5. Select Output > Array name to specify the name of the encoded samples array in the generated file.

**FIGURE 5-9:** ARRAY NAME SELECTION

6. Select Output > Filename to specify the name of the generated file and the path where the generated file will be stored.

**FIGURE 5-10:** FILE NAME SELECTION

7. Select Options > Law to specify the G.711 encoding scheme (A-law or µ-law) to be used while generating the encoded samples.

8. Click Encode. The Open file for encoding dialog appears.

9. Select the .wav file to be encoded, and then click Open.
10. The PCEU encodes the file and the generated files are available for inclusion into a dsPIC DSC application. Refer to 5.3 “Output Files” for more information on the generated file.
5.3 OUTPUT FILES

The dsPIC33F Speech Encoding Utility generates the following files:

TABLE 5-1: FILES GENERATED BY SPEECH ENCODING UTILITY

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.g71</td>
<td>File containing G.711 encoded samples.</td>
</tr>
<tr>
<td>.s</td>
<td>This file contains the encoded samples that are generated, if the target memory is Flash program memory. This file can be included directly into the application MPLAB project.</td>
</tr>
<tr>
<td>.c</td>
<td>This file contains the encoded samples that are generated, if the target memory is Data RAM. This file can be included directly into the application MPLAB project.</td>
</tr>
<tr>
<td>.raw</td>
<td>Contains raw audio samples.</td>
</tr>
</tbody>
</table>

5.3.1 Encoded Samples Stored in Flash Program Memory

If the Flash program memory is selected as the target memory for the encoded samples, the PCEU generates a .s file.

By default, the encoded samples array is called G711_data. The encoded samples are stored in 24-bit Flash program memory in the Little Endian byte order (first byte at lowest memory). Each encoded sample is allocated with one byte. Therefore, each Flash program memory word accommodates three encoded samples. Note that since all the three bytes of the Flash program memory are used for storing the encoded samples, the application program must use the table read instructions (as opposed to Program Space Visibility (PSV) technique) to read the Flash program memory.

5.3.2 Encoded Samples stored in Data RAM

If RAM is selected as the target memory for the encoded samples, the PCEU generates a .c file which contains an array of the encoded samples.

By default, the encoded sample array is called G711_data. The type of the array is a signed integer. Each array element stores two encoded samples.
# Index

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-Law and µ-law Companding Techniques</td>
<td>12</td>
</tr>
<tr>
<td>API</td>
<td></td>
</tr>
<tr>
<td>Decoder Functions</td>
<td></td>
</tr>
<tr>
<td>G711Alaw2Lin</td>
<td>42</td>
</tr>
<tr>
<td>G711Ulaw2Lin</td>
<td>42</td>
</tr>
<tr>
<td>Encoder Functions</td>
<td></td>
</tr>
<tr>
<td>G711Lin2Alaw</td>
<td>40</td>
</tr>
<tr>
<td>G711Lin2Ulaw</td>
<td>40</td>
</tr>
<tr>
<td>Resource Requirements</td>
<td>38</td>
</tr>
<tr>
<td>Application Programming Interface (API)</td>
<td>37</td>
</tr>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>API</td>
<td></td>
</tr>
<tr>
<td>Encoder Functions</td>
<td></td>
</tr>
<tr>
<td>G711Lin2Alaw</td>
<td>40</td>
</tr>
<tr>
<td>G711Lin2Ulaw</td>
<td>40</td>
</tr>
<tr>
<td>Resource Requirements</td>
<td>38</td>
</tr>
<tr>
<td>Application Programming Interface (API)</td>
<td>37</td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Conventions</td>
<td>6</td>
</tr>
<tr>
<td>Customer Notification Service</td>
<td>8</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
</tr>
<tr>
<td>Conventions</td>
<td>6</td>
</tr>
<tr>
<td>Layout</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>Document Layout</td>
<td>5</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
</tr>
<tr>
<td>Conventions</td>
<td>6</td>
</tr>
<tr>
<td>Layout</td>
<td>5</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Installation Files</td>
<td></td>
</tr>
<tr>
<td>Communication Folder</td>
<td>16</td>
</tr>
<tr>
<td>demos Folder</td>
<td>16</td>
</tr>
<tr>
<td>h Folder</td>
<td>19</td>
</tr>
<tr>
<td>Library Source Folder</td>
<td>19</td>
</tr>
<tr>
<td>Playback Folder</td>
<td>17</td>
</tr>
<tr>
<td>L</td>
<td></td>
</tr>
<tr>
<td>Library Files</td>
<td>15</td>
</tr>
<tr>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Microchip Internet Web Site</td>
<td>8</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Optional Hardware</td>
<td></td>
</tr>
<tr>
<td>Audio PICtail Plus Development Board</td>
<td>21</td>
</tr>
<tr>
<td>dsPIC33E USB Starter Kit</td>
<td>30</td>
</tr>
<tr>
<td>Explorer 16 Development Board</td>
<td>21</td>
</tr>
<tr>
<td>Multimedia Expansion Board</td>
<td>30</td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>PCEU</td>
<td>19</td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Reading, Recommended</td>
<td>7</td>
</tr>
<tr>
<td>Readme</td>
<td>8</td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Speech Signal Compression</td>
<td>11</td>
</tr>
<tr>
<td>System Requirements</td>
<td>13</td>
</tr>
<tr>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Warranty Registration</td>
<td>7</td>
</tr>
</tbody>
</table>
# Worldwide Sales and Service

**AMERICAS**

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: http://www.microchip.com/support
Web Address: www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Boston**
Westborough, MA
Tel: 774-447-0087
Fax: 774-447-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Cleveland**
Independence, OH
Tel: 216-447-0046
Fax: 216-447-0043

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Indianapolis**
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**Santa Clara**
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

**Toronto**
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

**ASIA/PACIFIC**

**Asia Pacific Office**
Suits 3707-14, 37th Floor
Tower 6, The Gateway Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

**China - Chongqing**
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

**China - Hangzhou**
Tel: 86-571-2819-3180
Fax: 86-571-2819-3189

**China - Hong Kong SAR**
Tel: 852-2401-1200
Fax: 852-2401-3431

**China - Nanjing**
Tel: 86-25-8473-2640
Fax: 86-25-8473-2647

**China - Qingdao**
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

**China - Wuhan**
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

**China - Xian**
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

**ASIA/PACIFIC**

**India - Bangalore**
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

**India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

**India - Pune**
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

**Japan - Yokohama**
Tel: 81-45-471-6166
Fax: 81-45-471-6122

**Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or 82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

**Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

**Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Hsin Chu**
Tel: 886-3-6578-300
Fax: 886-3-6578-370

**Taiwan - Kaohsiung**
Tel: 886-7-213-7830
Fax: 886-7-330-9305

**Taiwan - Taipei**
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

**Thailand - Bangkok**
Tel: 66-2-694-1341
Fax: 66-2-694-1350

**EUROPE**

**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**UK - Wokingham**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

05/02/11