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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, Keeloq, Keeloq logo, MPLAB, PIC, PICmicro, PICSTART, rPIC and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, Hi-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MIWI, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, PICtail logo, REAL ICE, rFLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WipeLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, Keeloq® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
# Table of Contents

**Chapter 1. Introduction**
- 1.1 AGC Library Overview .................................................................................. 11
- 1.2 Features ....................................................................................................... 12
- 1.3 Host System Requirements .......................................................................... 12

**Chapter 2. Installation**
- 2.1 Installation Procedure ................................................................................... 13
- 2.2 AGC Library Files ......................................................................................... 15

**Chapter 3. Quick Start Demonstration**
- 3.1 Demonstration Summary .............................................................................. 17
- 3.2 Demonstration Setup .................................................................................... 18
- 3.3 Demonstration Procedure ............................................................................ 22
- 3.4 Demonstration Code Description .................................................................. 23

**Chapter 4. Application Programming Interface (API)**
- 4.1 Adding the AGC Library to an Application .................................................... 25
- 4.2 AGC Operation and Parameters .................................................................. 27
- 4.3 External Gain Control ................................................................................... 28
- 4.4 Using the AGC Library .................................................................................. 28
- 4.5 Resource Requirements ............................................................................... 30
- 4.6 AGC Library API Functions .......................................................................... 31
- 4.7 Application Tips ............................................................................................ 60
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 on-line help. Select the Help menu, and then Topics to open a list of available on-line help files.

INTRODUCTION

This preface contains general information that will be useful to know before using the dsPIC® DSC Automatic Gain Control (AGC) library. Items discussed in this chapter include:

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

DOCUMENT LAYOUT

This user’s guide describes how to use the AGC library. The document is organized as follows:

• Chapter 1. “Introduction” – This chapter introduces the AGC library and provides a brief overview of noise suppression and the library features. It also outlines requirements for a host PC.
• Chapter 2. “Installation” – This chapter provides detailed information needed to install the AGC library demonstration on a PC.
• Chapter 3. “Quick Start Demonstration” – This chapter provides a hands-on demonstration of automatic gain control in a working application.
• Chapter 4. “Application Programming Interface (API)” – This chapter outlines how the API functions provided in the AGC library can be included in your application software via the Application Programming Interface.
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>Arial font:</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>...is the <em>only</em> compiler...</td>
</tr>
<tr>
<td></td>
<td>Emphasized text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial caps</td>
<td>A window</td>
<td>the <em>Output window</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A dialog</td>
<td>the <em>Settings dialog</em></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 <em>OK</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A tab</td>
<td>Click the <em>Power</em> tab</td>
<td></td>
</tr>
<tr>
<td>N'Rnnnn</td>
<td>A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.</td>
<td>4'b0010, 2'hF1</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>Courier New font:</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><strong>autoexec.bat</strong></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</td>
<td>0xFF, ‘A’</td>
<td></td>
</tr>
<tr>
<td>Italic Courier New</td>
<td>A variable argument</td>
<td><em>file.o</em>, where <em>file</em> can be any valid filename</td>
<td></td>
</tr>
<tr>
<td>Square brackets []</td>
<td>Optional arguments</td>
<td>mcc18 [options] <em>file</em> [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><em>var_name</em>, <em>var_name</em>...</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 AGC library. Other useful documents include:

dsPIC30F Family Reference Manual (DS70046)

Consult this document for detailed information on dsPIC30F device operation. This reference manual explains the operation of the dsPIC30F MCU 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.

dsPIC30F Family Overview (DS70043)

This document provides an overview of the functionality of the dsPIC® product family. It helps determine how the dsPIC30F 16-bit Digital Signal Controller family fits a specific product application. This document is a supplement to the dsPIC30F Family Reference Manual.

dsPIC33F Family Reference Manual Sections

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


This manual is a software developer’s reference for the dsPIC30F and dsPIC33F 16-bit MCU families of devices. It describes the instruction set in detail and also provides general information to assist in developing software for the dsPIC30F and dsPIC33F MCU families.

MPLAB® ASM30, MPLAB® LINK30 and Utilities User’s Guide (DS51317)

This document details Microchip Technology’s language tools for dsPIC devices based on GNU technology. The language tools discussed are:

• MPLAB ASM30 Assembler
• MPLAB LINK30 Linker
• MPLAB LIB30 Archiver/Librarian
• Other Utilities
MPLAB C30 C Compiler User's Guide (DS51284)

This document details the use of Microchip’s MPLAB C30 C Compiler for dsPIC devices to develop an application. MPLAB C30 is a GNU-based language tool, based on source code from the Free Software Foundation (FSF). For more information about the FSF, see www.fsf.org.

Other GNU language tools available from Microchip are:

• MPLAB ASM30 Assembler
• MPLAB LINK30 Linker
• MPLAB LIB30 Librarian/Archiver

MPLAB® IDE Simulator, Editor User’s Guide (DS51025)

Consult this document for more information pertaining to the installation and implementation of the MPLAB Integrated Development Environment (IDE) Software.

To obtain any of these documents, visit the Microchip web site at www.microchip.com.
THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, 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 listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, 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 www.microchip.com, click on Customer Change Notification and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the MPLAB C18 and MPLAB C30 C compilers; MPASM™ and MPLAB ASM30 assemblers; MPLINK™ and MPLAB LINK30 object linkers; and MPLIB™ and MPLAB LIB30 object librarians.
- **Emulators** – The latest information on Microchip in-circuit emulators. This includes the MPLAB ICE 2000, MPLAB ICE 4000, and MPLAB REAL ICE™.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 2.
- **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 and PRO MATE® II device programmers and the PICSTART® Plus and PICkit® 1 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
- Development Systems Information Line

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://support.microchip.com.
DOCUMENT REVISION HISTORY

Revision A (August 2009)

This is the initial released revision of this document.
Chapter 1. Introduction

The dsPIC® DSC Automatic Gain Control (AGC) Library provides an algorithm to automatically control the level of a signal. This chapter provides an overview of the library.

Topics covered include:
• AGC Library Overview
• Features
• Host System Requirements

1.1 AGC LIBRARY OVERVIEW

The AGC library is useful in speech and audio applications where distance between the system user and the microphone varies. The main function of the AGC algorithm is to estimate the short-term peak amplitude of input speech and apply a gain factor so that it is brought up to a desired level that has been set by the user. If no speech is detected, the gain will gradually 'leak' back down to a preset default level. It is assumed that the system of which the AGC is a part has been correctly set up for normal use by someone with close proximity to the microphone. Therefore, the AGC library maintains the input signal level to the subsequent processing blocks in the signal processing chain. The rate at which the gain changes are applied to the input signal can be controlled. Figure 1-1 shows an example of how the AGC can be used.

FIGURE 1-1: AGC IN AN APPLICATION

While the AGC library is implemented entirely in software, it can also be used to control the gain of an external codec and a software hook is provided for this. The library will also flag amplitude clipping on the input.
1.2 FEATURES

The AGC library features:

- Simple user interface – only one library file and one header file
- All functions called from a C application program
- Full compliance with the Microchip C30 Compiler, Assembler and Linker
- Highly optimized assembly code that uses the DSP instructions and advanced addressing modes
- Comprehensive API provides parametric control of the AGC engine
- Customized for speech applications
- Input signal clip detection
- Hooks to control gain of external codec
- Gain attack, release and leakage rate controls
- Audio bandwidth: 8 – 48 kHz sampling rate
- Library available for download from the Microchip web site, which also includes:
  - A copy of this user’s guide
  - Sample wave files
  - Sample demonstration application
  - Windows Help file for the AGC Library Application Programming Interface (API)

1.3 HOST SYSTEM REQUIREMENTS

The AGC Library requires a PC-compatible host system with these minimum characteristics:

- Intel Pentium® class or higher processor, or equivalent
- HTML browser
- 16 MB RAM
- 40 MB available hard drive space
- Microsoft® Windows® 98, Windows 2000, or Windows XP
Chapter 2. Installation

This chapter describes the installation procedure for the AGC library. To use the library, you must install it on your laptop or desktop PC. After installation, the library files can then be included into the target application. Topics covered include:

- Installation Procedure
- AGC Library Files

2.1 INSTALLATION PROCEDURE

The AGC library is available on a CD or as an archive file, which can be downloaded from the Microchip web site. In either case, installation of the library requires execution of the setup.exe file.

To install the library follow these steps:

1. Run the setup.exe file. This will start the installation and a license agreement dialog box will appear.

   **FIGURE 2-1: LIBRARY LICENSE AGREEMENT**

   ![License Agreement Dialog Box]

2. Review the license agreement and then click I Agree to continue the installation.
3. When prompted, specify the destination folder for the library files and folders.

**FIGURE 2-2: SPECIFY INSTALLATION FOLDER**

![Specify Installation Folder](image)

4. The installer will install the library files into the specified folder.

**FIGURE 2-3: INSTALLATION IN PROGRESS**

![Installation in Progress](image)

5. Click **Close** to close the dialog box.

6. In the next dialog box, click **Yes** to view the help file; otherwise, click **No**. This completes the AGC Library installation.

As part of the installation process, the folder, **AGC**, is created. Refer to **Section 2.2 “AGC Library Files”** for more information on these files.
2.2 AGC LIBRARY FILES

The AGC library installer creates a directory named AGC. This directory contains these six folders.

- demo
- docs
- h
- lib
- wavefiles

2.2.1 demo folder

This folder contains files that are required by the AGC library quick start demonstration, and contains these three subfolders.

- h
- libs
- src

Table 2-1 provides a description of these subfolders.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>demo\AGC demo.hex</td>
<td>Demonstration hex file.</td>
</tr>
<tr>
<td>demo\AGC demo.mcp</td>
<td>Demonstration MPLAB® project file.</td>
</tr>
<tr>
<td>demo\AGC demo.mcw</td>
<td>Demonstration MPLAB workspace.</td>
</tr>
<tr>
<td>demo\cleanup.bat</td>
<td>Batch file script to clean intermediate build files.</td>
</tr>
<tr>
<td>demo\h\dsPICDEM1_1Plus.h</td>
<td>C header file containing dsPICDEM™ 1.1 Plus Development Board routines.</td>
</tr>
<tr>
<td>demo\h\lcd.h</td>
<td>C header file defining the interface to the LCD driver.</td>
</tr>
<tr>
<td>demo\h\agc_api.h</td>
<td>C header file defining the interface to the AGC library.</td>
</tr>
<tr>
<td>demo\h\Si3000Drv.h</td>
<td>C header file defining the interface to the Si3000 Codec driver.</td>
</tr>
<tr>
<td>demo\h\libs\agclibv1_0.a</td>
<td>AGC library archive file.</td>
</tr>
<tr>
<td>demo\src\dsPICDEM1_1Plus.c</td>
<td>C source file containing the routines for the dsPICDEM 1.1 Plus Development Board.</td>
</tr>
<tr>
<td>demo\src\lcd.s</td>
<td>Assembly source code for communicating with the LCD controller.</td>
</tr>
<tr>
<td>demo\src\main.c</td>
<td>C source file containing the main speech processing routine.</td>
</tr>
<tr>
<td>demo\src\lcd_strings.c</td>
<td>C source file for LCD Display.</td>
</tr>
<tr>
<td>demo\src\Si3000Drv.c</td>
<td>C source file containing the implementation of the Si3000 Codec driver.</td>
</tr>
</tbody>
</table>
2.2.2 docs Folder

This folder contains a PDF of the AGC library user's guide. To view this document, Acrobat® Reader® is required. The user's guide can also be downloaded from the Microchip web site (www.microchip.com).

This folder also contains a Windows Help file for the AGC library Application Programming Interface (API) as listed in Table 2-2.

<table>
<thead>
<tr>
<th>Table 2-2: HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>AGC Help.chm</td>
</tr>
</tbody>
</table>

2.2.3 h Folder

This folder contains an include file for the AGC library as listed in Table 2-3.

<table>
<thead>
<tr>
<th>Table 2-3: INCLUDE FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>agc_api.h</td>
</tr>
</tbody>
</table>

2.2.4 lib Folder

This folder contains a library archive file for the AGC library as listed in Table 2-4.

<table>
<thead>
<tr>
<th>Table 2-4: LIBRARY FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>agclibv1_0.a</td>
</tr>
</tbody>
</table>

2.2.5 wavefiles Folder

This folder contains two WAVE files, which can be used with the quick start demonstration. The WAVE files can be played back on the PC using a media player with the repeat function ON to make the WAVE file run continuously. A description of these files is provided in Table 2-5.

<table>
<thead>
<tr>
<th>Table 2-5: WAVEFILES</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>microchip_sas_fem_8K.wav</td>
</tr>
<tr>
<td>microchip_sas_male_8K.wav</td>
</tr>
</tbody>
</table>
Chapter 3. Quick Start Demonstration

This chapter describes the AGC library quick start demonstration. Topics covered include:
- Demonstration Summary
- Demonstration Setup
- Demonstration Procedure
- Demonstration Code Description

3.1 DEMONSTRATION SUMMARY

A demonstration application program included with the AGC library demonstrates the library’s functionality. In the demonstration setup (illustrated in Figure 3-1), the dsPICDEM™ 1.1 Plus Development Board is configured to receive an 8 kHz audio signal from the PC through its microphone input port. The AGC library algorithm will process the signal and will output the signal through the speaker output port. The on-board Si3000 codec is used as the microphone and speaker interface.

A PC is used to drive sample audio signals through an audio cable from the PC’s speaker out port to J16 (MIC IN) on the dsPICDEM 1.1 Plus Development Board. A headset or speaker is connected to the J17 (SPKR OUT) on the dsPICDEM 1.1 Plus Development Board. Changing the level (within a significant range) of the audio signal from the PC will not change the output level of the signal heard on the headset/speaker. This is due to the AGC library controlling the level of the signal.

FIGURE 3-1: SETUP FOR AGC LIBRARY DEMO
You can use the WAVE files provided with the demonstration (in the wavefiles folder of the installation directory) as 8 kHz sampled speech signals, or you can provide your own signals. The input signal is captured by the on-board Si3000 voice band codec and the Data Converter Interface (DCI) module on the dsPIC® DSC device. The dsPIC DSC device then plays out the gain-controlled signal through the device’s DCI module and the on-board Si3000 codec.

When started, the program initializes with automatic gain control turned off, indicated by LED1 switched off and OFF displayed on the LCD. With automatic gain control off, the level of the signal heard in the headset changes if the input level is changed from the PC.

Automatic gain control is enabled by pressing SW1. LED1 is now switched on and ON is display on the LCD. The speech signal heard on the headset is now gain controlled. Changing the level of the input signal will not change the level of the output. If the input signal level is below a peak level threshold, the gain is set to unity. If the input signal level is high and above the clip threshold, LED2 switches on.

Turn on the repeat function in the Media Player on your PC to allow the WAVE file to run continuously. Then, change the level in the media player to evaluate the performance of the AGC library.

Note: Some media players insert a break before each repeat of the WAVE file. If you want to avoid this, sound editor programs usually provide for continuous looping. A good, freeware sound editor is Audacity, available from http://audacity.sourceforge.net/.

3.2 DEMONSTRATION SETUP

The demonstration application is run on a dsPICDEM™ 1.1 Plus Development Board (not included with the software license). Use the procedures outlined in the following sections to set up the demonstration.

3.2.1 Configure dsPICDEM™ 1.1 Plus Development Board

Before applying power, configure the board:

1. Set jumper J9 (adjacent to the oscillator socket) to the SLAVE position. This setting allows the on-board Si3000 codec chip to function as a serial clock Slave.
2. Connect the audio cable between the Speaker Out port on the PC and the ‘MIC IN’ (J16) jack on the dsPICDEM 1.1 Plus development board.
3. Connect the headset or speaker to the ‘SPKR OUT’ (J17) jack.
4. Connect the MPLAB ICD 2 between the PC (USB cable) and dsPICDEM 1.1 Plus development board (RJ-11 phone cable).
5. Connect the 9V power supply to power-up the dsPICDEM 1.1 Plus development board.
3.2.2 Program the dsPIC® DSC Device

Use this process to load the AGC library demonstration into the dsPIC DSC device on the dsPICDEM 1.1 Plus Development Board.

1. On your PC, launch MPLAB IDE and open the AGC demo.mcp project located in the demo folder (see Figure 3-3). For more information on using MPLAB IDE, refer to the "MPLAB IDE User’s Guide" (DS51025).
2. Import the project hex file (File>Import>AGC demo.hex).
3. From the Programmer menu, select Connect to link to the MPLAB ICD 2. The Output window shows that the MPLAB ICD 2 is ready.
4. Connect the MPLAB ICD 2 to the dsPICDEM 1.1 Plus Development Board.
5. Program the dsPIC DSC device on the board. From the Programmer menu, select Program. The Output window displays the download process and indicates that the programming has succeeded (see Figure 3-4).
6. When the program is loaded, disconnect the MPLAB ICD 2 from the board.
FIGURE 3-4: DOWNLOAD STATUS

Connecting to MPLAB ICD 2
... Connected
Setting Ydd source to target
Target Device dsPIC33FJ256GP710 found, revision = Rev 0x3002
... Reading ICD Product ID
Running ICD Self Test
... Passed
MPLAB ICD 2 Ready
Programming Target...
... Validating configuration fields
ICDWarn0046: Because clock switching is enabled, MPLAB ICD 2 requires the user to cycle target power after a period of time before continuing.
  Erasing Part
... Programming Program Memory (0x00 - 0x3FF)
Verifying...
... Program Memory
  Verify Succeeded
... Programming Configuration Bits
  Config Memory
Verifying configuration memory...
... Programming succeeded
01-Jun-2009, 14:27:27

MPLAB ICD 2 Ready
3.3 DEMONSTRATION PROCEDURE

With the demonstration application programmed into the device, the demonstration is ready to run. You can use the provided WAVE files, which are located in the wavefiles folder of the installation directory input speech signals, or you can provide your own signals. The input signal is sampled through the on-board Si3000 voice band codec and the Data Converter Interface (DCI) module of the dsPIC DSC device. The dsPIC DSC device then plays out the processed (gain controlled) signal through the device's DCI module and the on-board Si3000 codec.

The demonstration application relays the state of operation via the LEDs and the LCD. While the application is loading and initializing the on-chip and off-chip peripherals, a boot screen appears, which then switches automatically to the run-time screen, as shown in Figure 3-5.

FIGURE 3-5: DEMONSTRATION RUN-TIME LCD SCREEN

AGC Demo
OnOff: SW1 LED2: Clip
AGC is ON

* CLIPPED

The run-time screen displays the following:

1. The name of the demonstration.
2. SW1 is used to switch AGC on and off. LED2 serves as a CLIP indicator.
3. The current state of the algorithm.
4. A VU meter showing the input level. The band shows an acceptable input range.
   The word CLIPPED is displayed when the input signal is too large.

When started, the program initializes with AGC turned OFF, indicated by LED1 turned off and OFF displayed on the LCD. With the AGC off, the level of the signal heard on the headset changes with level change in the input signal. This can be observed by changing the input level.

The AGC is enabled by pressing SW1. LED1 is now switched on and ON is displayed on the LCD. The level of the signal in the headset stays fairly constant as the AGC attempts to maintain the level of the input signal at the set level. This can be observed by changing the input level.

If the input signal amplitude is large and above the clip threshold, then LED2 will switch ON till the clipping condition persists.
3.4 DEMONSTRATION CODE DESCRIPTION

The demonstration code runs on a dsPIC33F device, using the primary oscillator as the clock source with the PLL configured for 40 MIPS operation.

The file, main.c, contains the main function for the demonstration application. This file allocates all of the variables and arrays in data memory that are needed for DCI data buffering, as well as the blocks of data memory that need to be allocated for the AGC library functions.

The main function calls AGC_init() from the AGC library, which initializes the AGC algorithm to its default state.

The main function also calls SI3000_open() to initialize the DCI module, the Si3000 codec, and the DCI interrupt. The DCI module acts as a Master and drives the serial clock and frame synchronization lines. The Si3000 codec 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. Only one transmit slot and one receive slot are used in this demonstration.

The SI3000_open() function also initializes the Si3000 codec. The codec is reset, by connecting the RF6 pin of the dsPIC DSC device to the Reset pin of the Si3000, holding the port bit RF6 low for 100 cycles and then bringing it high. The codec is configured for a sample rate of 8 kHz. The MIC Gain is set to 10 dB and the Receive Gain is set to 0 dB. Both speakers are set to active and the Transmit Gain is set to 0 dB. The Analog Attenuation parameter is set to 0 dB. After initializing all of the Si3000 control registers, a delay is introduced for calibration of the Si3000 to occur. Finally, the DCI interrupt is enabled.

The codec driver is polled for a full frame of data. When the codec driver indicates a full frame of data is available, the contents of the codec data buffers are copied into the sin array and the AGC_apply() function from the AGC library is called with sin as the input data frame. The sin data buffer, which is also the output of the AGC_apply() function (after it has been executed), is played out on the speaker output of the headset.

The output on the LCD is made possible by initialization of the SPI module in the InitSPI function, and LCD driver functions and LCD string definitions present in the lcd.s and lcd_strings.c files, respectively.

The AGC library in the demonstration is configured to default settings. The target amplitude is 50% of Full scale. The clip threshold is set at 97%. The AGC headroom is set at 75%.

To toggle the AGC on and off, switch SW1 is polled. In the main loop, the value of agcEnable is read and passed to AGC_apply as the enable flag. If agcEnable is '0', the AGC Library is still called, but the input/output buffer is not changed. This enables the AGC Library to track the level changes in the signal. The AGC_detectInputClip() function is called after the AGC_apply() function to detect if the input signal has crossed the clip level. A clip condition will switch on LED2.
NOTES:
Chapter 4. Application Programming Interface (API)

This chapter describes in detail the Application Programming Interface (API) available for the AGC library. Topics covered include:

• Adding the AGC Library to an Application
• AGC Operation and Parameters
• External Gain Control
• Using the AGC Library
• Resource Requirements
• AGC Library API Functions
• Application Tips

4.1 ADDING THE AGC LIBRARY TO AN APPLICATION

To use the AGC library in an application, the library archive must be added to the application project workspace and the file agc_api.h must be included in the application code.

Use the following procedure to add the library to the application.

1. In the application MPLAB workspace, right click Library Files in the project window and select Add Files.

   ![FIGURE 4-1:)

2. Browse to the location of the agclibv1_0.a file (available in the lib folder within the installation directory).

3. Select the file and click Open.
4. The library is now added to the application.

To use the library functions, include the file `agc_api.h` in the application source code. This file can be copied from the `h` folder (located in the installation directory) to the application folder.
4.2 AGC OPERATION AND PARAMETERS

This section explains the operation of the AGC algorithm. Parameters that will be required while using the library are also explained.

4.2.1 AGC Operation

The AGC algorithm operates by monitoring the maximum amplitude of a 10 millisecond speech frame. If the frame is classified as being a part of a speech burst, its level is compared with the desired amplitude, which is called the amplitude target. If the amplitude of the frame needs to be increased, the overall amplitude of the signal is increased at a specified rate, which is called the attack rate. If the amplitude of the frame needs to be decreased, then the overall amplitude of the signal is decreased at a specified rate, which is called the release rate.

If the amplitude is above a level called headroom, this implies that the amplitude is nearing full scale and needs to be decreased at a more rapid rate, which is called the recovery rate. If the frame is classified as being a part of a silence frame (not a part of a speech burst), the signal level is decreased by a rate called leakage factor rate. The leakage feature is optional.

4.2.2 AGC Algorithm Parameters

The operation of the AGC algorithm is controlled by changing its parameters. A description of all the parameters is provided here.

- **Amplitude Target:** This parameter defines the target amplitude that the input signal level is compared against in order to compute the gain. The AGC will try to maintain average output amplitude at this level. This parameter is set using the \texttt{AGC\_setAmplitudeTarget()} function.

- **Input Clip Threshold:** This parameter identifies the amplitude above which the input signal is defined to be saturated or clipped. The AGC algorithm provides an indication of this condition. The clipping condition by itself does not affect the operation of the AGC algorithm. This parameter is set by using the \texttt{AGC\_setInputClipThreshold()} function. The \texttt{AGC\_detectInputClip()} function can be used to determine whether the input signal has clipped.

- **Peak Detect Ratio:** This ratio is used by the AGC algorithm to determine whether the current frame is to be treated as a part of the speech burst or as a silence. The lowest amplitude in the AGC History Buffer determines the silence level. If the silence frame and the ratio of the maximum amplitude of the current frame is smaller than the silence level by the Peak Detect factor, the frame is treated as if it were speech. Using a low value will make the algorithm less sensitive and slow to respond to signal changes from silence to speech or during inter-syllable durations. Choosing too high a value will cause the AGC algorithm to respond too quickly, giving all syllables a uniformly high stress (as if the person speaking is shouting). This parameter is set by using the \texttt{AGC\_setPeakDetect()} function.

- **Maximum Gain:** This parameter places an upper limit on the gain that the AGC algorithm can apply to the input signal. This parameter is set by using the \texttt{AGC\_setMaxGain()} function.

- **Minimum Gain:** This parameter places a lower limit on the gain that the AGC algorithm can apply to the input. If the input to the AGC is above the headroom level, then Minimum gain level will be applied to it. This parameter is set by using the \texttt{AGC\_setMinGain()} function.

- **Headroom:** The AGC algorithm can be directed to reduce the gain if the predicted output climbs above a critical level. This level is specified by the headroom. The headroom is always more than the Amplitude Target but less than Input Clip Threshold. The headroom is set by using the \texttt{AGC\_setHeadroom()} function.
• **Attack Rate:** The attack rate determines the rate at which the gain is increased if the current input amplitude is less than the set amplitude target. The attack rate is set using the `AGC_setAttack()` function.

• **Release Rate:** The release rate determines the rate at which the gain is decreased when the current output amplitude is greater than the set amplitude target. The release rate is set using the `AGC_setRelease()` function.

• **Leakage Rate:** The leakage rate determines the rate at which the gain is decreased if the current frame is detected as a silence (or close to silence) frame. That is, if the maximum amplitude is less than the current silence level by the peak detect factor, the gain is decreased at leakage rate. In an extended silence, the gain will gradually decrease to Minimum Gain. If it is desired to maintain a constant gain level during silences, this parameter should be set to zero. The leakage rate is set using the `AGC_setLeakageFactor()` function.

• **Recovery Rate:** The recovery rate determines the rate at which the gain is decreased if the current output amplitude is greater than the headroom level. Typically, if the signal is above the headroom level, this indicates the signal is headed towards a clip condition. The recovery rate is set using the `AGC_setRecovery()` function.

### 4.3 EXTERNAL GAIN CONTROL

The AGC Library provides hooks to the application to control the gain of an external codec or a Programmable Gain Amplifier. This is useful in applications that feature such components. The `AGC_getDesiredGain()` function provides the gain (in dB) required to maintain the signal level at the set Amplitude Target. This gain can then be provided to an external codec or a Programmable Gain Amplifier, which can then be used to control the signal level.

### 4.4 USING THE AGC LIBRARY

The AGC library can process many independent channels of audio with each channel having its own settings and parameters. The application must include the `agclibv1_0.a` and `agc_api.h` files in the application. The minimum coding steps required to use the AGC library are listed here:

1. **Select the sampling rate:** Specify the sampling rate of the input signal by specifying the value of `AGC_PROC_FRAME` macro in `agc_api.h`. The following coding steps need to be coded in the application. Refer to Example 4-1 for the actual code.

2. **Allocate memory for the AGC State memory:** This is an integer array for size `AGC_XSTATE_MEM_SIZE_INT`. Every audio channel must have its own state memory.

3. **Initialize the AGC State memory:** Use the `AGC_init()` function to initialize the AGC state memory for every audio channel.

4. **Set the amplitude target:** Set the amplitude target for AGC output for each channel by using the `AGC_setAmplitudeTarget()` function. Note that if this step is skipped, the AGC will set the amplitude target to default (i.e., 50%).

5. **Apply AGC to the audio channel:** Apply AGC to the audio channel by calling the `AGC_apply()` function. Example 4-1 provides the code for calling this function.
EXAMPLE 4-1: CALLING THE AGC_apply() FUNCTION

```c
#include "agc_api.h"

int agcStateX1[AGC_XSTATE_MEM_SIZE_INT]; /* Step 2 */
int agcStateX2[AGC_XSTATE_MEM_SIZE_INT]; /* Step 2 */

int main(void)
{
    AGC_init(agcStateX1); /* Step 3 */
    AGC_init(agcStateX2); /* Step 3 */
    AGC_setAmplitudeTarget(agcStateX1,Q15(0.65f)); /* Step 4 */
    AGC_setAmplitudeTarget(agcStateX2,Q15(0.60f)); /* Step 4 */
    .
    .
    while(1)
    {
        AGC_apply(agcStateX1,input1,AGC_TRUE,AGC_PROC_FRAME); /* Step 5 */
        AGC_apply(agcStateX2,input2,AGC_TRUE,AGC_PROC_FRAME); /* Step 5 */
    }
}
```

Some of the AGC functions require input parameters to be specified as Q15 fractional value in hexadecimal format. To simply usage, the library provides a Q15() macro, which converts a floating point number to a Q15 fractional value in hexadecimal format.
4.5 RESOURCE REQUIREMENTS

The AGC Library requires the following resources while running on the dsPIC DSC device.

4.5.1 Program Memory Usage

<table>
<thead>
<tr>
<th>Size (bytes)</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code in Program Memory</td>
<td>1140 .libagc</td>
</tr>
<tr>
<td>Tables in Program Memory</td>
<td>1611 .const</td>
</tr>
<tr>
<td>Total Program Memory</td>
<td>2751</td>
</tr>
</tbody>
</table>

4.5.2 Data Memory Usage (8 kHz sampling rate)

<table>
<thead>
<tr>
<th>Size (bytes)</th>
<th>Alignment</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>agc_state_mem_x</td>
<td>292</td>
<td>X data memory</td>
</tr>
<tr>
<td>sin</td>
<td>160</td>
<td>X data memory</td>
</tr>
<tr>
<td>Total Data Memory</td>
<td>1284</td>
<td></td>
</tr>
</tbody>
</table>

The size of agc_state_mem_x depends on the lookback AGC_AMP_HIST, and the size of sin (Send in), assuming the default input block length of 10 ms, is proportional to the sampling rate. Sampling rates up to 48 kHz are supported.

4.5.3 Estimated Dynamic Memory Usage

<table>
<thead>
<tr>
<th>Size (bytes)</th>
<th>Alignment</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heap</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stack</td>
<td>&lt; 320</td>
<td></td>
</tr>
</tbody>
</table>

4.5.4 Computational Speed

<table>
<thead>
<tr>
<th>Function</th>
<th>MIPS</th>
<th>Typical Call Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGC_init()</td>
<td>&lt; 0.5</td>
<td>Once</td>
</tr>
<tr>
<td>AGC_apply()</td>
<td>0.33 (8 kHz) to 1.98 (48 kHz)</td>
<td>10 ms</td>
</tr>
<tr>
<td>All other functions</td>
<td>Minimal</td>
<td>As required</td>
</tr>
</tbody>
</table>

4.5.5 Data Format

sin can be 10-, 12-, or 16-bit linear PCM data. The AGC algorithm automatically adjusts for the data format used.
4.6 AGC LIBRARY API FUNCTIONS

This section provides detailed information for the following API functions:

- AGC_init
- AGC_apply
- AGC_detectInputClip
- AGC_getDesiredGain
- AGC_setInputClipThreshold
- AGC_getInputClipThreshold
- AGC_setPeakDetect
- AGC_getPeakDetect
- AGC_setMaxGain
- AGC_getMaxGain
- AGC_setMinGain
- AGC_getMinGain
- AGC_setAmplitudeTarget
- AGC_getAmplitudeTarget
- AGC_setHeadroom
- AGC_getHeadroom
- AGC_setAttack
- AGC_getAttack
- AGC_setRelease
- AGC_getRelease
- AGC_setLeakageFactor
- AGC_getLeakageFactor
- AGC_setRecovery
- AGC_getRecovery
- AGC_TRUE
- AGC_FALSE
- AGC_SAMPLE_RATE
- AGC_PROC_FRAME
- AGC_AMP_HIST
- AGC_XSTATE_MEM_SIZE_INT
- AGC_CLIP_THRES_DEFAULT
- AGC_PEAK_DETECT_DEFAULT
- AGC_MAX_GAIN_DEFAULT
- AGC_MIN_GAIN_DEFAULT
- AGC_INIT_GAIN_DEFAULT
- AGC_AMPLITUDE_TARGET_DEFAULT
- AGC_HEADROOM_DEFAULT
- AGC_ATTACK_DEFAULT
- AGC_RELEASE_DEFAULT
- AGC_LEAKAGE_FACTOR_DEFAULT
- AGC_RECOVERY_DEFAULT
AGC_init

Description
Initializes the AGC algorithm.

Include
agc_api.h

Prototype
void AGCinit(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
None.

Remarks
None.

Code Example
int agc_state_mem_x [AGC_XSTATE_MEM_SIZE_INT] _XBSS(2);
...
AGC_init(agc_state_mem_x);
AGC_apply

Description:
This function applies AGC to the current frame of data. It also calculates the desired AGC gain (in dB) which is available for use by suitable hardware (if software gain application is disabled, the program continues to run in the background). It also updates the amplitude clip detect flag.

Include
agc_api.h

Prototype
void AGC_apply(int* ptrStateX, int* Sin, int enable, int agcProcSize);

Arguments
ptrStateX   A pointer to the X memory for this instance of AGC.
Sin        A pointer to the input/output buffer of size AGC_PROC_FRAME.
Enable     A flag to indicate if AGC is required for this buffer
           (AGC_TRUE/AGC_FALSE).
agcProcSize The length of the input/output buffer Sin.

Return Value
None.

Remarks
The AGC algorithm is process-in-place meaning that the output is passed back in the input buffer. Setting Enable to AGC_FALSE returns an unprocessed buffer of data, but the AGC algorithm still runs in the background and all state variables are updated.

Code Example
int agc_state_mem_x [AGCG_XSTATE_MEM_SIZE_INT] _XBSS(2);
int Sin [AGC_PROC_FRAME] _XBSS(2);
...
AGC_init(agc_state_mem_x);
...
AGC_apply(agc_state_mem_x, Sin, AGC_TRUE, AGC_PROC_FRAME);
AGC_detectInputClip

Description
Returns a flag to indicate whether input audio is close to clipping.

Include
agc_api.h

Prototype
int AGC_detectInputClip(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Values
AGC_TRUE The current input frame is above the clip threshold.
AGC_FALSE The current input frame is below the clip threshold.

Remarks
Clip threshold parameter can be selected by the user (see function
AGC_setInputClipThreshold).

Code Example
int clipDetected;
...
clipDetected = AGC_detectInputClip(ptrStateX);
AGC_getDesiredGain

Description
Returns the gain that an external codec could apply to the signal to achieve the target amplitude.

Include
agc_api.h

Prototype
int AGC_getDesiredGain(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit positive integer representing gain in dB.

Remarks
This output is available whether or not AGC is actually applied: for instance, it could be used as input to an external gain control (e.g., on the audio codec).

Code Example
int gainDesired;
...gainDesired = AGC_getDesiredGain(ptrStateX);
AGC_setInputClipThreshold

Description
Sets the input clip threshold.

Include
agc_api.h

Prototype
void AGC_setInputClipThreshold(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the threshold as a fraction.

Return Value
None.

Remarks
None.

Code Example
AGC_setInputClipThreshold(ptrStateX, Q15(0.97f));
This sets the threshold to 97% of full scale.
**AGC_getInputClipThreshold**

**Description**
This function will get the input clip threshold for the specified instance of AGC.

**Include:**
agc_api.h

**Prototype**

```c
int AGC_getInputClipThreshold(int* ptrStateX, int value);
```

**Arguments**

- **ptrStateX**
  A pointer to the X memory for this instance of AGC.

**Return Value**
A 16-bit positive integer representing the parameter as a fraction.

**Remarks**
None.

**Code Example**

```c
int clipLevel;
clipLevel = AGC_getInputClipThreshold(ptrStateX);
```
AGC_setPeakDetect

Description
Sets the amplitude ratio for detecting input frame as a speech frame. Presence of speech frame inhibits AGC gain from leaking to initial gain.

Include
agc_api.h

Prototype
void AGC_setPeakDetect(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
None.

Code Example
AGC_setPeakDetect(ptrStateX, Q15(0.25F));
This sets the peak detect ratio to 25%.
AGC_getPeakDetect

Description
Returns the current value of the amplitude ratio for detecting input peak amplitudes.

Include
agc_api.h

 Prototype
int AGC_getPeakDetect(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit positive integer representing the parameter as a fraction.

Remarks
None.

Code Example
int peakRatio;
...
peakRatio = AGC_getPeakDetect(ptrStateX);
AGC_setMaxGain

Description
Sets the maximum permitted value that the AGC can apply to the signal.

Include
agc_api.h

Prototype
void AGC_setMaxGain(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing a gain level in dB.

Return Value
None.

Remarks
None.

Code Example
AGC_setMaxGain(ptrStateX, 32);
This limits the output gain to no more than 32 dB.
AGC_getMaxGain

Description
Returns the maximum gain calculated that the AGC will apply to the signal.

Include
agc_api.h

Prototype
int AGC_getMaxGain(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
16-bit integer representing the maximum gain in dB.

Remarks
None.

Code Example
int gainLevel;

gainLevel = AGC_getMaxGain(ptrStateX);
AGC_setMinGain

Description
Sets the minimum gain that the AGC will apply to the signal.

Include
agc_api.h

Prototype
void AGC_setMinGain(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing a gain level in dB.

Return Value
None.

Remarks
It is expected that this value will be unity in most instances, but a higher value may be required for a particular setup.

Code Example
AGC_setMinGain(ptrStateX, 2);
This prevents the output gain from falling below 2 dB.
AGC_getMinGain

Description
Returns the minimum gain that the AGC will apply to the signal.

Include
agc_api.h

Prototype
int AGC_getMinGain(int* ptrStateX);

Arguments
ptrStateX        A pointer to the X memory for this instance of AGC.

Return Value
16-bit integer representing the minimum gain in dB.

Remarks
None.

Code Example
int gainLevel;

gainLevel = AGC_getMinGain(ptrStateX);
AGC_setAmplitudeTarget

Description
Sets the target peak amplitude for the output speech, as a fraction of full scale.

Include
agc_api.h

Prototype
void AGC_setAmplitudeTarget(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
The target amplitude is specified as Q15 number. Alternatively the Q15 macro could be used to convert a float value to a Q15 number.

Code Example
AGC_setAmplitudeTarget(ptrStateX, Q15(0.65f));
This sets target level as 65% of full scale.
AGC_getAmplitudeTarget

Description
Returns the current value of target amplitude.

Include
agc_api.h

Prototype
int AGC_getAmplitudeTarget(int* ptrStateX);

Arguments
ptrStateX       A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit value representing the parameter as a fraction.

Remarks
None.

Code Example
int ampLevel;

ampLevel = AGC_getAmplitudeTarget(ptrStateX);
AGC_setHeadroom

Description
Sets the AGC headroom as a fraction of full scale.

Include
agc_api.h

Prototype
void AGC_setHeadroom(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
It is expected that it will always be greater than the value chosen for AGC amplitude target.

Code Example
AGC_setHeadroom((ptrStateX, Q15(0.75f));
This sets target level as 75% of full scale.
AGC_getHeadroom

Description
Returns the current value of AGC Headroom level.

Include
agc_api.h

Prototype
int AGC_getHeadroom(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit value representing the parameter as a fraction.

Remarks
None.

Code Example
int headRoomValue;

headRoomValue = AGC_getHeadroom(ptrStateX);
AGC_setAttack

Description
Sets the AGC attack smoothing factor (factor for increasing gain) as a fraction.

Include
agc_api.h

Prototype
void AGC_setAttack(int* ptrStateX, int value);

Arguments
ptrStateX  A pointer to the X memory for this instance of AGC.
value      A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
None.

Code Example
AGC_setAttack(ptrStateX, Q15(0.05f));
This sets the attack rate at 5%.
AGC_getAttack

Description
Returns the current value of the AGC Attack smoothing factor.

Include
agc_api.h

Prototype
int AGC_getAttack(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit positive integer representing the parameter as a fraction.

Remarks
None.

Code Example
int attackValue;

attackValue = AGC_getAttack(ptrStateX);
AGC_setRelease

Description
Sets the AGC release smoothing factor (for decreasing gain).

Include
agc_api.h

Prototype
void AGC_setRelease(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
None.

Code Example
AGC_setRelease(ptrStateX, Q15(0.02f));
This sets the release smoothing factor to 2%.
AGC_getRelease

Description
Returns the current value of AGC Release Smoothing factor.

Include
agc_api.h

Prototype
int AGC_getRelease(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit positive integer value representing the parameter as a fraction.

Remarks
None.

Code Example
int releaseValue;

releaseValue = AGC_getRelease(ptrStateX);
AGC_setLeakageFactor

Description
Sets the AGC leakage smoothing factor (factor for decreasing gain when speech is not present).

Include
agc_api.h

Prototype
void AGC_setLeakageFactor(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
None.

Code Example
AGC_setLeakageFactor(ptrStateX, Q15(0.001f));
This sets the leakage factor to 0.1%.
AGC_getLeakageFactor

**Description**
Returns the current value of the AGC Leakage Smoothing Factor.

**Include**
agc_api.h

**Prototype**
```c
int AGC_getLeakageFactor(int* ptrStateX);
```

**Arguments**
- **ptrStateX** A pointer to the X memory for this instance of AGC.

**Return Value**
A 16-bit positive integer representing the parameter as a fraction.

**Remarks**
None.

**Code Example**
```c
int leakValue;

leakValue = AGC_getRelease(ptrStateX);
```
AGC_setRecovery

Description
Sets the AGC Recover smoothing factor (rate at which the signal is decreased after climbing above clip threshold).

Include
agc_api.h

Prototype
void AGC_setRecovery(int* ptrStateX, int value);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.
value A 16-bit positive integer representing the parameter as a fraction.

Return Value
None.

Remarks
None.

Code Example
AGC_setRecovery(ptrStateX, Q15(0.1f));
Sets the recovery rate to 10%.
AGC_getRecovery

Description
Returns the current value of AGC Recovery smoothing factor.

Include
agc_api.h

Prototype
int AGC_getRecovery(int* ptrStateX);

Arguments
ptrStateX A pointer to the X memory for this instance of AGC.

Return Value
A 16-bit positive integer representing the parameter as a fraction.

Remarks
None.

Code Example
int recoValue;
recoValue = AGC_getRecovery(ptrStateX);
### AGC_TRUE

**Description**
Used to indicate true to the AGC algorithm.

**Value**
1

### AGC_FALSE

**Description**
Used to indicate false to the AGC algorithm.

**Value**
0

### AGC_SAMPLE_RATE

**Description**
Defines the sample rate at which the AGC operates. This value can be changed to suit the application requirements.

**Value**
8000 (default)

### AGC_PROC_FRAME

**Description**
Length of input/output audio buffer (shared with other audio functions).

**Value**
80 at 8 kHz sampling rate; in general, equivalent to 10 ms.

### AGC_AMP_HIST

**Description**
Number frames look-back for the AGC algorithm. Since this determines memory size it may not be changed at run-time.

**Value**
128
<table>
<thead>
<tr>
<th><strong>AGC_XSTATE_MEM_SIZE_INT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Size in integers of the memory location required for the X-State memory.</td>
</tr>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>(2\cdot AGC_AMP_HIST + 42)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>AGC_CLIP_THRES_DEFAULT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Threshold to detect input amplitude clipping (16-bit integer interpreted as Q15 fractional).</td>
</tr>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Q15(0.97f)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>AGC_PEAK_DETECT_DEFAULT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Amplitude ratio used to detect possible speech (16-bit integer interpreted as Q15 fractional).</td>
</tr>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Q15(0.3f)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>AGC_MAX_GAIN_DEFAULT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Maximum permitted amplification in dB applied by the AGC. It is a positive integer.</td>
</tr>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>45</td>
</tr>
</tbody>
</table>
### AGC_MIN_GAIN_DEFAULT

**Description**
Minimum permitted amplification in dB applied by the AGC. It must not be negative.

**Value**
0

### AGC_INIT_GAIN_DEFAULT

**Description**
Initial value of gain applied by AGC in dB: if it is known roughly how much gain is needed, this will speed the initial convergence of the AGC.

**Value**
0

### AGC_AMPLITUDE_TARGET_DEFAULT

**Description**
Target amplitude for the AGC, as a fraction of full scale (16-bit integer interpreted as Q15 fractional). This is understood to be an average of speech peak levels – actual amplitudes may be much higher.

**Value**
Q15(0.50f)

### AGC_HEADROOM_DEFAULT

**Description**
Amplitude threshold to prevent clipping, as a fraction of full scale (16-bit integer interpreted as Q15 fractional). Note that it must leave some room for overshoot.

**Value**
Q15(0.75f)
AGC_ATTACK_DEFAULT

Description
Smoothing ratio when an increase in gain is demanded (16-bit integer interpreted as Q15 fractional).

Value
Q15(0.02f)

AGC_RELEASE_DEFAULT

Description
Smoothing ratio when a decrease in gain is demanded (16-bit integer interpreted as Q15 fractional).

Value
Q15(0.001f)

AGC_LEAKAGE_FACTOR_DEFAULT

Description
Smoothing ratio to return gain to default level when speech is not detected (16-bit integer interpreted as Q15 fractional). Set to zero to disable the leakage feature.

Value
Q15(0.0001f)

AGC_RECOVERY_DEFAULT

Description
Smoothing ratio to respond to sudden increase in input speech level (16-bit integer interpreted as Q15 fractional).

Value
Q15(0.1f)
4.7  APPLICATION TIPS

The default values of the AGC parameters configure the AGC library to operate in a typical voice application. The following tips provide information on ways to optimize the AGC algorithm in a given application:

1. The optimal value for the Headroom is 75% of the full scale range. The optimal value for the Amplitude Target is 65% of the full scale range. Setting up the parameters in this way ensures good use of the available precision. The difference between the Amplitude Target and the Headroom accommodates for overshoot.

2. Setting the Peak Detect parameter to too high a value will make the AGC algorithm overly sensitive, giving all syllables a uniformly high stress. Too low a value will make the AGC slow to respond. A value of 0.3 will keep the speech sounding natural.

3. The AGC algorithm is designed for speech signals and uses the statistical nature of speech signals as a basis for its operation. The algorithm will not process input signals where syllables are exaggerated or where the input signal is more musical in nature.

4. In typical applications, the Minimum Gain is set to 0 dB (or unity gain). It can be set to a non-zero positive value to provide a default gain to signal.

5. The Clip Detection function should be used to check whether the input signal is too large. This feature can prove useful as a protection mechanism for subsequent signal processing stages or for protecting output devices.

6. The Amplitude target could be set to a value between 25% to 85% of full scale. A low value should be used if the input is expected to be spiky.

7. For deterministic operation of the AGC algorithm, some guidelines need to be followed:
   a) Amplitude Target < Headroom < 95% of Full Scale.
   b) 0.0 < Attack rate < Recovery Rate.
   c) 0.0 < Leakage Rate < Release rate < Recovery Rate.
   d) Attack rate < Recovery Rate < 0.5.
Index

A
Adding the AGC Library to an Application.............. 25
AGC Algorithm Parameters................................... 27
  Amplitude Target ........................................... 27
  Attack Rate ................................................. 28
  Headroom ....................................................... 27
  Input Clip Threshold ....................................... 27
  Leakage Rate ................................................... 28
  Maximum Gain ................................................. 27
  Minimum Gain ................................................... 27
  Peak Detect Ratio ............................................ 28
  Recovery Rate ............................................... 28
  Release Rate ...................................................... 28
AGC Library
  Application Example ....................................... 11
  Features ......................................................... 12
  Overview ......................................................... 11
  Requirements .................................................... 12
AGC Library Demonstration Setup ...................... 17, 18
AGC Library Files
demo Folder ..................................................... 15
docs Folder ....................................................... 16
AGC Operation .................................................... 27
API ................................................................. 5
API Functions
  AGC_AMP_HIST ............................................... 56
  AGC_AMPLITUDE_TARGET_DEFAULT ..................... 58
  AGC_apply ....................................................... 33
  AGC_ATTACK_DEFAULT ..................................... 59
  AGC_CLIP_THRESH_DEFAULT ............................... 57
  AGC_detectInputClip ....................................... 34
  AGC_FALSE ....................................................... 56
  AGC_getAmplitudeTarget ................................... 45
  AGC_getAttack ............................................... 49
  AGC_getDesiredGain ....................................... 35
  AGC_getHeadroom .......................................... 47
  AGC_getInputClipThreshold ............................... 37
  AGC_getLeakageFactor ..................................... 53
  AGC_getMaxGain .............................................. 41
  AGC_getMinGain .............................................. 43
  AGC_getPeakDetect ........................................ 39
  AGC_getRecovery .......................................... 55
  AGC_getRelease ............................................ 51
  AGC_HEADROOM_DEFAULT ................................ 58
  AGC_init ....................................................... 32
  AGC_INIT_GAIN_DEFAULT ................................ 58
  AGC_LEAKAGE_FACTOR_DEFAULT ......................... 59
  AGC_MAX_GAIN_DEFAULT ................................... 57
  AGC_MIN_GAIN_DEFAULT ................................... 58
  AGC_PEAK_DETECT_DEFAULT .............................. 57
  AGC_PROC_FRAME .......................................... 56
AGC_RECOVERY_DEFAULT ...................................... 59
AGC_RELEASE_DEFAULT ...................................... 59
AGC_SAMPLE_RATE ............................................ 56
AGC_setAmplitudeTarget .................................. 44
AGC_setAttack ............................................... 48
AGC_setHeadroom .......................................... 46
AGC_setInputClipThreshold ............................... 36
AGC_setLeakageFactor ..................................... 52
AGC_setMaxGain .............................................. 40
AGC_setMinGain .............................................. 42
AGC_setPeakDetect .......................................... 38
AGC_setRecovery ............................................ 54
AGC_setRelease ............................................. 50
AGC_TRUE ......................................................... 56
AGC_XSTATE_MEM_SIZE_INT ................................ 57
Application Programming Interface (API) .............. 25
Application Tips .............................................. 60
C
Configuring the dsPICDEM™ 1.1 Plus Development Board ................................................................. 18
Customer Notification Service ............................ 9
Customer Support ............................................. 9
D
Demonstration
  Board Configuration ....................................... 18
  Demonstration Code ....................................... 23
  Demonstration Procedure .................................. 22
  Development Board Setup ................................. 19
Documentation
  Conventions ..................................................... 6
  Layout ............................................................ 5
E
External Gain Control ....................................... 28
F
Free Software Foundation .................................. 8
G
GNU Language Tools ....................................... 8
I
Installing the AGC Library ................................ 13
Internet Address ............................................. 9
M
Microchip Internet Web Site ............................... 9
MPLAB IDE User's Guide ..................................... 8
P
Programming the dsPIC® DSC Device .................... 19
# Index

**R**
- Reading, Recommended ........................................... 7
- Resource Requirements .......................................................... 30
  - Computational Speed .................................................. 30
  - Data Format ...................................................................... 30
  - Data Memory Usage ....................................................... 30
  - Estimated Dynamic Memory Usage ................................. 30
  - Program Memory Usage .................................................. 30

**U**
- Using the AGC Library ...................................................... 28
  - Allocating Memory for the AGC State Memory .................. 28
  - Applying AGC to the Audio Channel ............................... 28
  - Initializing the AGC State Memory .................................. 28
  - Selecting the Sampling Rate ......................................... 28
  - Setting the Amplitude Target .................................... 28

**W**
- Warranty Registration ................................................... 7
- WWW Address .................................................................... 9
# Worldwide Sales and Service

<table>
<thead>
<tr>
<th>AMERICAS</th>
<th>ASIA/PACIFIC</th>
<th>ASIA/PACIFIC</th>
<th>EUROPE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corporate Office</strong>&lt;br&gt;2355 West Chandler Blvd.&lt;br&gt;Chandler, AZ 85224-6199&lt;br&gt;Tel: 480-792-7200&lt;br&gt;Fax: 480-792-7277&lt;br&gt;Technical Support:&lt;br&gt;<a href="http://support.microchip.com">http://support.microchip.com</a>&lt;br&gt;Web Address:&lt;br&gt;www.microchip.com</td>
<td><strong>Asia Pacific Office</strong>&lt;br&gt;Suites 3707-14, 37th Floor&lt;br&gt;Tower 6, The Gateway Harbour City, Kowloon Hong Kong&lt;br&gt;Tel: 852-2401-1200&lt;br&gt;Fax: 852-2401-3431</td>
<td><strong>India - Bangalore</strong>&lt;br&gt;Tel: 91-80-3090-4444&lt;br&gt;Fax: 91-80-3090-4080</td>
<td><strong>Austria - Wels</strong>&lt;br&gt;Tel: 43-7242-2244-39&lt;br&gt;Fax: 43-7242-2244-393</td>
</tr>
<tr>
<td><strong>Atlanta</strong>&lt;br&gt;Duluth, GA&lt;br&gt;Tel: 678-957-9614&lt;br&gt;Fax: 678-957-1455</td>
<td><strong>China - Beijing</strong>&lt;br&gt;Tel: 86-10-8528-2100&lt;br&gt;Fax: 86-10-8528-2104</td>
<td><strong>India - New Delhi</strong>&lt;br&gt;Tel: 91-11-4160-8631&lt;br&gt;Fax: 91-11-4160-8632</td>
<td><strong>Denmark - Copenhagen</strong>&lt;br&gt;Tel: 45-4450-2828&lt;br&gt;Fax: 45-4485-2829</td>
</tr>
<tr>
<td><strong>Boston</strong>&lt;br&gt;Westborough, MA&lt;br&gt;Tel: 774-760-0087&lt;br&gt;Fax: 774-760-0088</td>
<td><strong>China - Chengdu</strong>&lt;br&gt;Tel: 86-28-8665-5511&lt;br&gt;Fax: 86-28-8665-7889</td>
<td><strong>India - Pune</strong>&lt;br&gt;Tel: 91-20-2566-1512&lt;br&gt;Fax: 91-20-2566-1513</td>
<td><strong>France - Paris</strong>&lt;br&gt;Tel: 33-1-69-53-63-20&lt;br&gt;Fax: 33-1-69-30-90-79</td>
</tr>
<tr>
<td><strong>Chicago</strong>&lt;br&gt;Itasca, IL&lt;br&gt;Tel: 630-285-0071&lt;br&gt;Fax: 630-285-0075</td>
<td><strong>China - Hong Kong SAR</strong>&lt;br&gt;Tel: 852-2401-1200&lt;br&gt;Fax: 852-2401-3431</td>
<td><strong>Japan - Yokohama</strong>&lt;br&gt;Tel: 81-45-471-6166&lt;br&gt;Fax: 81-45-471-6122</td>
<td><strong>Germany - Munich</strong>&lt;br&gt;Tel: 49-89-627-144-0&lt;br&gt;Fax: 49-89-627-144-44</td>
</tr>
<tr>
<td><strong>Cleveland</strong>&lt;br&gt;Independence, OH&lt;br&gt;Tel: 216-447-0064&lt;br&gt;Fax: 216-447-0043</td>
<td><strong>China - Nanjing</strong>&lt;br&gt;Tel: 86-25-8473-2460&lt;br&gt;Fax: 86-25-8473-2470</td>
<td><strong>Korea - Daegu</strong>&lt;br&gt;Tel: 82-53-744-4301&lt;br&gt;Fax: 82-53-744-4302</td>
<td><strong>Italy - Milan</strong>&lt;br&gt;Tel: 39-0331-742611&lt;br&gt;Fax: 39-0331-466781</td>
</tr>
<tr>
<td><strong>Dallas</strong>&lt;br&gt;Addison, TX&lt;br&gt;Tel: 972-818-7423&lt;br&gt;Fax: 972-818-2924</td>
<td><strong>China - Qingdao</strong>&lt;br&gt;Tel: 86-532-8502-7355&lt;br&gt;Fax: 86-532-8502-7205</td>
<td><strong>Korea - Seoul</strong>&lt;br&gt;Tel: 82-2-554-7200&lt;br&gt;Fax: 82-2-558-5932 or 82-2-558-5934</td>
<td><strong>Netherlands - Drunen</strong>&lt;br&gt;Tel: 31-416-690399&lt;br&gt;Fax: 31-416-690340</td>
</tr>
<tr>
<td><strong>Detroit</strong>&lt;br&gt;Farmington Hills, MI&lt;br&gt;Tel: 248-538-2250&lt;br&gt;Fax: 248-538-2260</td>
<td><strong>China - Shanghai</strong>&lt;br&gt;Tel: 86-21-5407-5533&lt;br&gt;Fax: 86-21-5407-5066</td>
<td><strong>Malaysia - Kuala Lumpur</strong>&lt;br&gt;Tel: 60-3-6201-9857&lt;br&gt;Fax: 60-3-6201-9859</td>
<td><strong>Spain - Madrid</strong>&lt;br&gt;Tel: 34-91-708-08-90&lt;br&gt;Fax: 34-91-708-08-91</td>
</tr>
<tr>
<td><strong>Kokomo</strong>&lt;br&gt;Kokomo, IN&lt;br&gt;Tel: 765-864-8360&lt;br&gt;Fax: 765-864-8337</td>
<td><strong>China - Shenyang</strong>&lt;br&gt;Tel: 86-24-2334-2829&lt;br&gt;Fax: 86-24-2334-2393</td>
<td><strong>Malaysia - Penang</strong>&lt;br&gt;Tel: 60-4-227-8870&lt;br&gt;Fax: 60-4-227-4068</td>
<td><strong>UK - Wokingham</strong>&lt;br&gt;Tel: 44-118-921-5869&lt;br&gt;Fax: 44-118-921-5820</td>
</tr>
<tr>
<td><strong>Los Angeles</strong>&lt;br&gt;Mission Viejo, CA&lt;br&gt;Tel: 949-462-9523&lt;br&gt;Fax: 949-462-9608</td>
<td><strong>China - Shenzhen</strong>&lt;br&gt;Tel: 86-755-8203-2660&lt;br&gt;Fax: 86-755-8203-1760</td>
<td><strong>Philippines - Manila</strong>&lt;br&gt;Tel: 63-2-634-9065&lt;br&gt;Fax: 63-2-634-9069</td>
<td></td>
</tr>
<tr>
<td><strong>Santa Clara</strong>&lt;br&gt;Santa Clara, CA&lt;br&gt;Tel: 408-961-6444&lt;br&gt;Fax: 408-961-6445</td>
<td><strong>China - Wuhan</strong>&lt;br&gt;Tel: 86-27-5980-5300&lt;br&gt;Fax: 86-27-5980-5118</td>
<td><strong>Singapore</strong>&lt;br&gt;Tel: 65-6334-8870&lt;br&gt;Fax: 65-6334-8850</td>
<td></td>
</tr>
<tr>
<td><strong>Toronto</strong>&lt;br&gt;Mississauga, Ontario, Canada&lt;br&gt;Tel: 905-673-0699&lt;br&gt;Fax: 905-673-6509</td>
<td><strong>China - Xiamen</strong>&lt;br&gt;Tel: 86-592-2388138&lt;br&gt;Fax: 86-592-2388130</td>
<td><strong>Taiwan - Hsin Chu</strong>&lt;br&gt;Tel: 886-3-6578-300&lt;br&gt;Fax: 886-3-6578-370</td>
<td></td>
</tr>
<tr>
<td><strong>China - Xian</strong>&lt;br&gt;Tel: 86-29-8833-7252&lt;br&gt;Fax: 86-29-8833-7256</td>
<td><strong>Taiwan - Kaohsiung</strong>&lt;br&gt;Tel: 886-7-536-4818&lt;br&gt;Fax: 886-7-536-4803</td>
<td><strong>Taiwan - Taipe</strong>&lt;br&gt;Tel: 886-2-2500-6610&lt;br&gt;Fax: 886-2-2508-0102</td>
<td></td>
</tr>
<tr>
<td><strong>China - Zhuhai</strong>&lt;br&gt;Tel: 86-756-3210040&lt;br&gt;Fax: 86-756-3210049</td>
<td><strong>Taiwan - Taipei</strong>&lt;br&gt;Tel: 86-2-694-1351&lt;br&gt;Fax: 66-2-694-1350</td>
<td><strong>Thailand - Bangkok</strong>&lt;br&gt;Tel: 66-2-694-1351</td>
<td></td>
</tr>
</tbody>
</table>