



dsPIC30F
Acoustic Echo Cancellation
Library 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.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor 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, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance 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.

© 2004, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==**

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, 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

Preface	1
Chapter 1. Introduction	
1.1 Introduction	7
1.2 Highlights	7
1.3 Acoustic Echo Cancellation Overview	7
1.4 Features	8
1.5 Host System Requirements	9
1.6 Devices Supported	9
1.7 Accessory Kit	9
Chapter 2. Installation	
2.1 Introduction	11
2.2 Highlights	11
2.3 Installation Procedure	11
2.4 Library Archive	12
2.5 Include Files	12
2.6 Demo Files	13
2.7 User's Guide	13
Chapter 3. Application Programming Interface	
3.1 Introduction	15
3.2 Highlights	15
3.3 Acoustic Echo Cancellation Library Functions	15
3.4 'C' Data Types Used for Function Arguments	15
3.5 Function Prototypes and Arguments	16
Chapter 4. Resource Requirements	
4.1 Introduction	21
4.2 Highlights	21
4.3 Program Memory Requirements	21
4.4 Data Memory Requirements	22
4.5 Computational Speed Requirements	22
Chapter 5. Acoustic Echo Cancellation Algorithm	
5.1 Introduction	23
5.2 Highlights	23
5.3 AEC Algorithm Overview	23
5.4 NLMS Algorithm	25
5.5 Double Talk Detection	28

dsPIC30F Acoustic Echo Cancellation Library User's Guide

5.6 Voice Activation Detection	28
5.7 Non-Linear Processor	34
5.8 Howling Control (Optional)	34
Chapter 6. AEC Demonstration	
6.1 Introduction	37
6.2 Highlights	37
6.3 Demonstration Summary	37
6.4 Demonstration Setup	39
6.5 Demonstration Procedure	41
6.6 Demo Code Description	41
Index	43
Worldwide Sales and Service	44



dsPIC30F ACOUSTIC ECHO CANCELLATION LIBRARY USER'S GUIDE

Preface

INTRODUCTION

This user's guide supports the dsPIC30F Acoustic Echo Cancellation Library. This manual provides information you need to incorporate acoustic echo cancellation capability into your embedded solution.

HIGHLIGHTS

Items discussed in this chapter are:

- About This Guide
- Warranty Registration
- Recommended Reading
- The Microchip Web Site
- Development Systems Customer Notification Service
- Customer Support

ABOUT THIS GUIDE

This User's Guide describes how to use the dsPIC30F Acoustic Echo Cancellation Library. The document is organized as follows:

- **Chapter 1: Introduction** – This chapter introduces the dsPIC30F Acoustic Echo Cancellation Library (Chapter Name) and provides a brief overview of acoustic echo cancellation. It also outlines requirements for a host PC, dsPIC30F device resource requirements and specific dsPIC30F devices that support the AEC library. Finally, it provides information on the optional AEC Accessory Kit.
- **Chapter 2: Installation** – This chapter provides instructions for installing the library files and describes the contents of the source files, include files, demo files and archive files.
- **Chapter 3: Application Programming Interface** – This chapter provides detailed information needed to interface the AEC library to a user application on a dsPIC30F device.
- **Chapter 4: Resource Requirements** – This chapter describes the resources required to support the AEC Library in a dsPIC30F embedded application.
- **Chapter 5: Acoustic Echo Cancellation Algorithm** – This chapter describes the acoustic echo cancellation algorithms contained in the AEC Library.
- **Chapter 6: AEC Demonstration** – This chapter provides a hands-on demonstration of acoustic echo cancellation in a working application.

dsPIC30F Acoustic Echo Cancellation Library User's Guide

Conventions Used in This Guide

This User's Guide uses the following documentation conventions:

DOCUMENTATION CONVENTION

Description	Represents	Examples
Code (Courier font):		
Plain characters	Sample code Filenames and paths	<code>#define START</code> <code>c:\autoexec.bat</code>
Angle brackets: < >	Variables	<label>, <exp>
Square brackets []	Optional arguments	<code>pic30-as [main.s]</code>
Curly brackets and pipe character: { }	Choice of mutually exclusive arguments; an OR selection	<code>errorlevel {0 1}</code>
Lower case characters in quotes	Type of data	"filename"
Ellipses...	Used to imply (but not show) additional text that is not relevant to the example	<code>list</code> <code>["list_option... ,</code> <code>"list_option"]</code>
0xnnn	A hexadecimal number where n is a hexadecimal digit	<code>0xFFFF, 0x007A</code>
Italic characters	A variable argument; it can be either a type of data (in lower case characters) or a specific example (in upper case characters)	<code>char isascii (char,</code> <code>ch);</code>
Interface (Arial font):		
Underlined, italic text with right arrow	A menu selection from the menu bar	<i>File > Save</i>
Bold characters	A window or dialog button to click	OK, Cancel
Characters in angle brackets < >	A key on the keyboard	<Tab>, <Ctrl-C>
Documents (Arial font):		
Italic characters	Referenced books	<i>MPLAB IDE User's Guide</i>

Documentation Updates

All documentation becomes dated, and this user's guide is no exception. Since the dsPIC30F Acoustic Echo Cancellation Library and other Microchip tools are constantly evolving to meet customer needs, some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site to obtain the latest documentation available.

Documentation Numbering Conventions

Documents are numbered with a "DS" number. The 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 = The document number.

A = The revision level of the document.

WARRANTY REGISTRATION

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in your Warranty Registration Card entitles you 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 dsPIC30F Acoustic Echo Cancellation 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 Data Sheet, Motor Control and Power Conversion Family (DS70082)

Consult this document for detailed information on the dsPIC30F Motor Control and Power Conversion devices. Reference information found in this data sheet includes:

- Device memory map
- Device pinout and packaging details
- Device electrical specifications
- List of peripherals included on the device

dsPIC30F Data Sheet, General Purpose and Sensor Families (DS70083)

Consult this document for detailed information on the dsPIC30F Sensor and General Purpose devices. Reference information found in this data sheet includes:

- Device memory map
- Device pinout and packaging details
- Device electrical specifications
- List of peripherals included on the device

dsPIC30F5011, dsPIC30F5013 Data Sheet, High Performance Digital Signal Controllers (DS70116)

This data sheet contains specific information for the dsPIC30F5011/5013 Digital Signal Controller (DSC) devices.

dsPIC30F6011, dsPIC30F6012, dsPIC30F6013, dsPIC30F6014 Data Sheet, High Performance Digital Signal Controllers (DS70117)

This data sheet contains specific information for the dsPIC30F6011/6012/6013/6014 Digital Signal Controller (DSC) devices.

dsPIC30F Programmer's Reference Manual (DS70030)

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

dsPIC30F Acoustic Echo Cancellation Library User's Guide

dsPIC® High Performance 16-bit Digital Signal Controller Family Overview (DS70043)

This document provides an overview of the functionality of the dsPIC product family. Its purpose is to help you determine how the dsPIC 16-bit Digital Signal Controller Family fits your specific product application. This document is a supplement to the dsPIC30F Family Reference Manual.

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

This document helps you use 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)

A guide to using the dsPIC DSC C compiler. MPLAB LINK30 is used with this tool.

dsPIC™ Language Tools Libraries (DS51456)

DSP, dsPIC peripheral and standard (including math) libraries for use with dsPIC language tools.

GNU HTML Documentation

This documentation is provided on the language tool CD-ROM. It describes the standard GNU development tools, upon which these tools are based.

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.

Microchip Web Site

Our web site (www.microchip.com) contains a wealth of documentation. Individual data sheets, application notes, tutorials and user's guides are all available for easy download. All documentation is in Adobe® Acrobat® (pdf) format.

THE MICROCHIP WEB SITE

Microchip provides online support on the Microchip World Wide Web (WWW) site. The website is used by Microchip as a means to make files and information easily available to customers. To view the site, you must have access to the Internet and a web browser, such as, Netscape® Navigator or Microsoft® Internet Explorer. The Microchip web site is available at: www.microchip.com.

The web site provides a variety of services. Users may download files for the latest development tools, data sheets, application notes, user's guides, articles, and sample programs. A variety of information specific to the business of Microchip is also available, including listings of Microchip sales offices, distributors and factory representatives.

Technical Support

- Frequently Asked Questions (FAQ)
- Online Discussion Groups – Conferences for products, Development Systems, technical information and more
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip products

Developer's Toolbox

- Design Tips
- Device Errata

Other Available Information

- Latest Microchip Press Releases
- Listing of Seminars and Events
- Job Postings

DEVELOPMENT SYSTEMS CUSTOMER NOTIFICATION SERVICE

Microchip started the customer notification service to help our customers stay current on Microchip products with the least amount of effort. Once you subscribe, you will receive E-mail notification whenever we change, update, revise or have errata related to your specified product family or development tool of interest.

Go to the Microchip web site at (www.microchip.com) and click on Customer Change Notification. Follow the instructions to register.

The Development Systems product group categories are:

- Compilers
- Emulators
- In-Circuit Debuggers
- MPLAB
- Programmers

Here is a description of these categories:

Compilers – The latest information on Microchip C compilers and other language tools. These include the MPLAB C17, MPLAB C18 and MPLAB C30 C compilers; MPASM™ and MPLAB ASM30 assemblers; MPLINK™ and MPLAB LINK30 object linkers; MPLIB™ and MPLAB LIB30 object librarians.

Emulators – The latest information on Microchip in-circuit emulators. This includes the MPLAB ICE 2000 and MPLAB ICE 4000.

In-Circuit Debuggers – The latest information on Microchip in-circuit debuggers. These include the MPLAB ICD and MPLAB ICD 2.

MPLAB – 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 and MPLAB SIM30 simulators, MPLAB IDE Project Manager and general editing and debugging features.

Programmers – The latest information on Microchip device programmers. These include the PRO MATE® II device programmer and PICSTART® Plus development programmer.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Corporate Applications Engineer (CAE)
- Hotline

Customers should call their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. See the sales offices and locations listed on the back of this publication.

Corporate Applications Engineers (CAEs) may be contacted at (480) 792-7627.

In addition, there is a Systems Information and Upgrade Line. This line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.

The Hotline Numbers are:

1-800-755-2345 for U.S. and most of Canada.

1-480-792-7302 for the rest of the world.

Chapter 1. Introduction

1.1 INTRODUCTION

This chapter introduces the dsPIC30F Acoustic Echo Cancellation Library. This library provides a function to eliminate echo generated in the acoustic path between a speaker and a microphone. This manual provides information needed to incorporate acoustic echo cancellation capability into an embedded solution.

1.2 HIGHLIGHTS

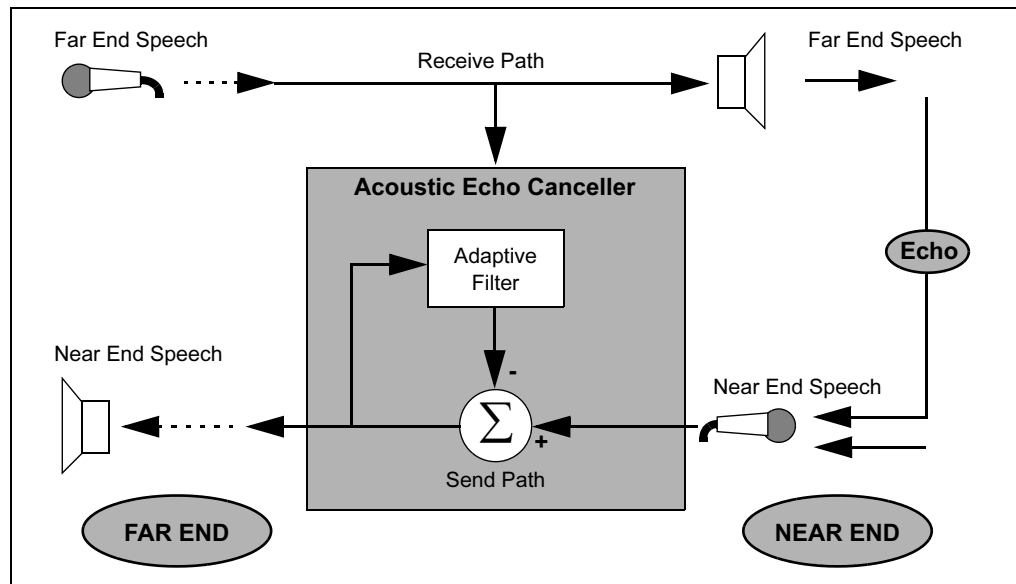
This chapter includes information on these topics:

- Acoustic Echo Cancellation Overview
- Features
- Host System Requirements
- Devices Supported
- Accessory Kit

1.3 ACOUSTIC ECHO CANCELLATION OVERVIEW

Acoustic echo cancellation eliminates echoes generated in the acoustic path between a speaker and a microphone, as illustrated in Figure 1-1. Acoustic echo cancellation is useful for speech and telephony applications in which a microphone is located near a speaker and is susceptible to output from the speaker. Such acoustic echoes result in a perceptible and distracting echo effect at the far end.

FIGURE 1-1: ACOUSTIC ECHO CANCELLATION



dsPIC30F Acoustic Echo Cancellation Library User's Guide

Acoustic echo cancellation is especially suitable for applications such as:

- Hands-free cell phone kits
- Speakerphones
- Intercoms
- Teleconferencing systems

For hands-free phones intended to be used in compact environments such as a car cabin, this library is fully compliant with the G.167 Standard for Acoustic Echo Cancellation.

The Acoustic Echo Cancellation (AEC) Library is written entirely in assembly language and is highly optimized to make extensive use of the dsPIC30F DSP instruction set and advanced addressing modes. The algorithm avoids data overflow. The AEC Library provides an `AcousticEchoCancellerInit` function for initializing the various data structures required by the algorithm and an `AcousticEchoCanceller` function to remove the echo component from a 10 ms block of sampled 16-bit speech data. You can easily call both functions through a well-documented Application Programmer's Interface (API).

The `AcousticEchoCanceller` function is primarily a Time Domain algorithm. The received far-end speech samples (typically received across a communication channel such as a telephone line) are filtered using an adaptive Finite Impulse Response (FIR) filter. The coefficients of this filter are adapted using the Normalized Least Mean Square (NLMS) algorithm, such that the filter closely models the acoustic path between the near end speaker and the near end microphone (i.e., the path traversed by the echo). Voice Activity Detection (VAD) and Double Talk Detection (DTD) algorithms are used to avoid updating the filter coefficients when there is no far end speech and also when there is simultaneous speech from both ends of the communication link (double talk). As a consequence, the algorithm functions correctly even in the presence of full-duplex communication. A Non-Linear Processor (NLP) algorithm is used to eliminate residual echo.

1.4 FEATURES

Key feature of the dsPIC30F Acoustic Echo Cancellation Library include:

- Employs only two user functions: `AcousticEchoCancellerInit` and `AcousticEchoCanceller`. Both functions can be called from a C or assembly application program
- Full compliance with the Microchip dsPIC30F C30 Compiler, Assembler and Linker
- Simple user interface – only one library file and one header file
- Highly optimized assembly code that uses DSP instructions and advanced addressing modes
- Echo cancellation for 16, 32 or 64 ms echo delays or “tail lengths” (configurable)
- Fully tested for compliance with G.167 specifications for in-car applications
- Audio Bandwidth: 0 to 4 kHz at 8 kHz sampling rate
- Convergence Rate: Up to 43 dB/sec., typically greater than 30 dB/sec.
- Echo Cancellation: Up to 50 dB, typically > 40 dB
- Can be used together with the Noise Suppression (NS) Library, since the same processing block size (10 ms) is used
- Library User's Guide is provided to help the user understand and use the library
- Demo application source code is provided with the library
- Accessory Kit available for purchase (see **Section 1.7 “Accessory Kit”**).

1.5 HOST SYSTEM REQUIREMENTS

The AEC Library requires a PC-compatible system with these attributes:

- Intel Pentium class or higher processor, or equivalent
- 16 MB RAM (minimum)
- 40 MB (minimum) available hard drive space
- Microsoft Windows[®] 98, Windows 2000 or Windows XP

1.6 DEVICES SUPPORTED

The AEC Library is supported by these dsPIC30F devices:

- dsPIC30F6014
- dsPIC30F6012
- dsPIC30F5013 (for a maximum of 32 ms echo delay)
- dsPIC30F5011 (for a maximum of 16 ms echo delay)

1.7 ACCESSORY KIT

An optional Acoustic Accessory Kit (Part # AC300030) includes these items:

- 6 ft. Audio Cable (1/8" stereo)
- Headset
- Two 14.7456 MHz oscillators
- Microphone
- Speaker
- 6 ft. DB9 M/F RS-232 cable
- DB9M-DB9M Null Modem Adaptor

dsPIC30F Acoustic Echo Cancellation Library User's Guide

NOTES:

Chapter 2. Installation

2.1 INTRODUCTION

This section describes the various files in the AEC library and includes instructions for installing the library on your laptop or PC for use with dsPIC30F programming tools.

2.2 HIGHLIGHTS

This chapter includes information on these topics:

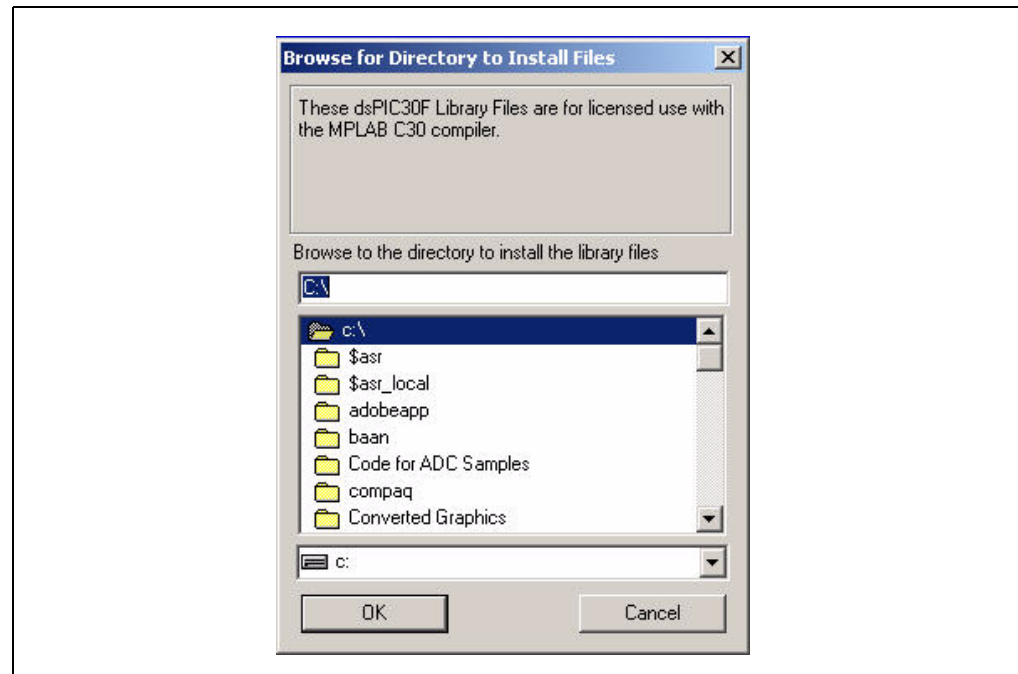
- Installation Procedure
- Library Archive
- Include Files
- User's Guide

2.3 INSTALLATION PROCEDURE

The dsPIC30F AEC Library is packaged on a CD. It can also be purchase and downloaded from the Microchip web site. To install the library follow these steps:

1. Insert the library CD into the appropriate drive and start the setup procedure. Run `aec.exe` from the download location. The installation screen displays, as shown in Figure 2-1.

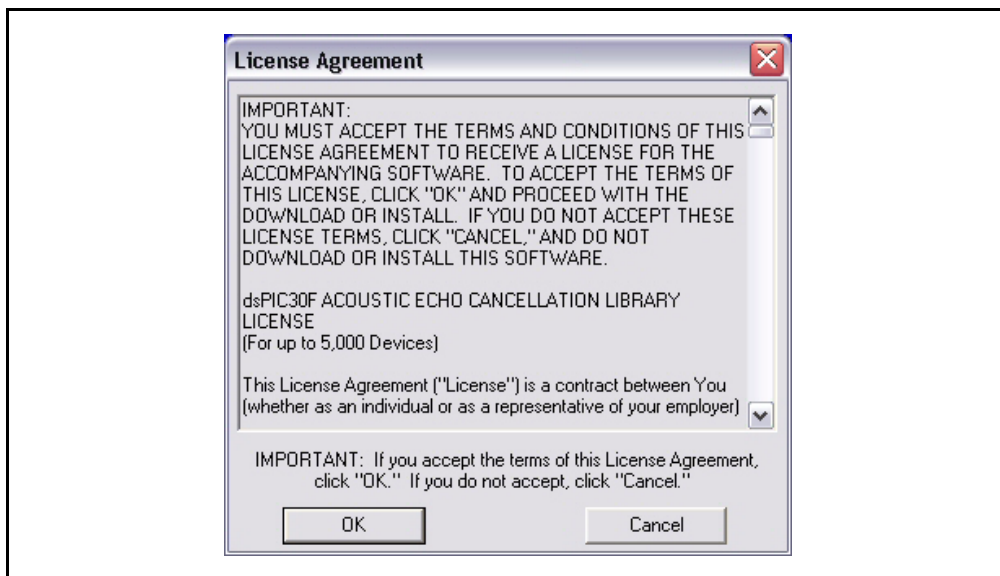
FIGURE 2-1: INSTALLATION SCREEN



dsPIC30F Acoustic Echo Cancellation Library User's Guide

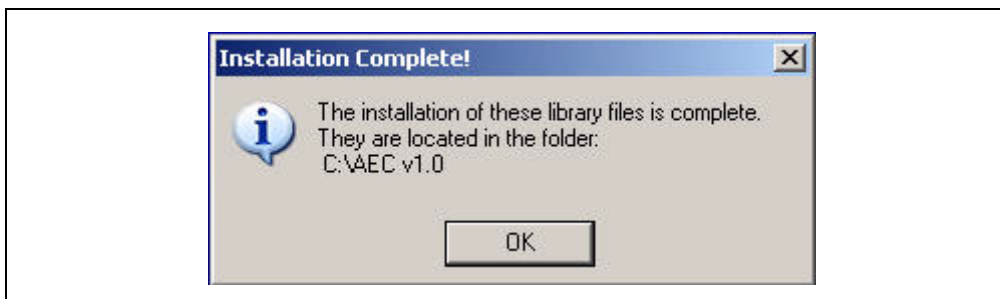
2. Decide where to install the files (the default location is the root of your hard drive), then click **OK** to continue.
3. Review the License Agreement (Figure 2-2), then click OK to continue.

FIGURE 2-2: PROGRAM LICENSE AGREEMENT



4. When the Installation Complete message displays (Figure 2-3), click **OK**.

FIGURE 2-3: INSTALLATION COMPLETE MESSAGE



2.4 LIBRARY ARCHIVE

The Library Archive (`libaec.a`) is a pre-compiled object file that contains all the Acoustic Echo Cancellation Library functions. Upon installing the library, the `libaec.a` file is located in the `AEC v1.0\lib` folder on your hard drive.

2.5 INCLUDE FILES

The C header file for the AEC Library is `aec.h`. This file contains various constant parameters used by C application programs that use the AEC algorithm, as well as the C function prototypes of the two user-callable functions `AcousticEchoCancellerInit` and `AcousticEchoCanceller`. The maximum echo tail length to be cancelled is specified in this file.

The assembly header file for the AEC Library is `aec.inc`. This file contains various constant parameters used by dsPIC30F assembly language application programs that incorporate the AEC algorithm. The maximum echo tail length to be cancelled is specified in this file.

After installation, both the include files are located in the `AEC v1.0\inc` folder.

2.6 DEMO FILES

The source, include, MPLAB project and workspace files, as well as a ready-to-use.hex file (echodemo.hex) and.cof (echodemo.cof) file, are provided in the AEC v1.0\demo folder. This folder also contains a local copy of the libaec.a and aec.h files, which are used by the demo to avoid any project tool path dependencies.

2.7 USER'S GUIDE

The AEC Library User's Guide is located in the AEC v1.0\doc folder and is called AEC Library User's Guide.pdf. This document can also be downloaded from the [Acoustic Echo Cancellation Library](#) product page on the Microchip web site (www.microchip.com).

dsPIC30F Acoustic Echo Cancellation Library User's Guide

NOTES:

Chapter 3. Application Programming Interface

3.1 INTRODUCTION

This section outlines how the functions provided in the AEC Library can be used by user application software, via an Application Programming Interface.

3.2 HIGHLIGHTS

This chapter includes information on these topics:

- Acoustic Echo Cancellation Library Functions
- 'C' Data Types Used for Function Arguments
- Function Prototypes and Arguments

3.3 ACOUSTIC ECHO CANCELLATION LIBRARY FUNCTIONS

The AEC Library consists of the two functions listed below, which can be called from a user application. The user program(s) that utilizes these functions can be written either in 'C' or dsPIC30F assembly language.

TABLE 3-1: AEC LIBRARY FUNCTIONS

Function Name	Purpose	Usage
AcousticEchoCancellerInit	To initialize the data structures used by this function.	Called by the user only once, before the first usage of the function.
AcousticEchoCanceller	To cancel out the acoustic echo component from a 10 ms block of signal samples.	Called by the user once for every 10 ms block of signal samples.

3.4 'C' DATA TYPES USED FOR FUNCTION ARGUMENTS

Two basic data types (corresponding to standard data types defined by the 'C' programming language) are used for various parameters or arguments passed to the library functions. These data types, along with the alias symbols defined for them in the "aec.h" include file, are listed below:

TABLE 3-2: ARGUMENT DATA TYPES

Symbol	Standard Data Type
Word16	Short int
Word32	Long int

3.5 FUNCTION PROTOTYPES AND ARGUMENTS

The prototypes, along with a description of the arguments used by each function, are given below. The 'C' function prototypes are defined in the header file `aec.h`. For each function, the actual name of the library function, as would be used when it is called from a user application written in assembly language, is also listed.

3.5.1 AEC Initialization Function

Function prototype, when the function is invoked from 'C'

```
void AcousticEchoCancellerInit (Word32 *echo_mem,  
    Word16 order, Word32 *vad_mem,  
    Word32 *scratch_ptr, Word16 howling_control );
```

Note: These five arguments must be successively assigned to W0, W1, W2, W3 and W4, respectively, before the function can be called.

Function call, when the function is invoked from assembly language

```
CALL _AcousticEchoCancellerInit
```

Note: The `RCALL` instruction may be used instead of `CALL`, when the conditions for usage of the `RCALL` instruction are satisfied

Function Arguments

Description:

<code>echo_mem</code>	<p>Pointer to a block of static variables in RAM utilized by the AEC algorithm. This memory block needs to be allocated by the user application. Its length is defined in the <code>aec.h</code> header file (used if the calling application routine is written in C) and the <code>aec.inc</code> include file (used if the calling application routine is written in assembly). It is dependent on the maximum echo tail length for which the echo needs to be eliminated.</p> <p>In order to configure the appropriate echo tail length, the <code>echo_delay</code> constant in the <code>aec.h</code> or <code>aec.inc</code> file must be modified by the user.</p> <p>The <code>echo_mem</code> memory block must have a size equal to $((8 * \text{echo_delay}) + 47)$ long integers or $((32 * \text{echo_delay}) + 188)$ bytes.</p> <p>This memory block must be allocated in X data memory.</p>
<code>order</code>	<p>Order of AEC adaptive filter, as given by the constant <code>ORDER</code> defined in <code>aec.h</code> and <code>aec.inc</code>.</p> <p>For a signal sampling rate of 8 kHz, $\text{ORDER} = 8 * \text{echo_delay}$.</p> <p>The user must pass the constant <code>ORDER</code> as the function argument.</p>
<code>vad_mem</code>	<p>Pointer to a 296 byte (74 long integers) block of static variables in RAM used by the Voice Activity Detection algorithm (described in the next section). This memory block must be allocated by the user application in X data memory.</p>
<code>scratch_mem</code>	<p>Pointer to a block of RAM utilized by the AEC algorithm. This memory block must be allocated by the user application in X data memory, and must have a size equal to: $(\text{ORDER} + (3 * \text{FRAME})) / 2$ long integers, or $2 * (\text{ORDER} + (3 * \text{FRAME}))$ bytes, where <code>FRAME</code> = 80 (a constant defined in <code>aec.h</code> and <code>aec.inc</code>), which represents the number of signal samples in each 10 ms processing frame.</p>
<code>Howling_control</code>	<p>This argument determines if the optional Howling Control feature will be enabled. If the value passed is '1', then Howling Control is enabled; if it is '0', then Howling Control is Disabled.</p> <p>Note: For 64 ms echo tail length, this argument must be '0', i.e., Howling Control is not supported for 64 ms echo tail length.</p>
Return Value:	None

3.5.2 AEC Frame Processing Function

Function prototype, when the function is invoked from 'C'

```
void AcousticEchoCanceller (Word32 *echo_mem, Word32 *vad_mem,  
    Word16 *rin_in, Word16 *sin_in, Word16 *sout,  
    Word16 inhibit_flag, Word16 *yscrptr);
```

Note: These seven arguments must be successively assigned to W0, W1, W2, W3, W4, W5 and W6, respectively, before the function can be called.

Function call, when the function is invoked from assembly language

```
CALL _AcousticEchoCanceller
```

Note: The RCALL instruction may be used instead of CALL, when the conditions for usage of the RCALL instruction are satisfied.

Function Arguments

Input Arguments:

<code>echo_mem</code>	<p>Pointer to a block of static variables in RAM utilized by the AEC algorithm. This memory block needs to be allocated by the user application. Its length is defined in the <code>aec.h</code> header file (used if the calling application routine is written in C) and the <code>aec.inc</code> include file (used if the calling application routine is written in assembly). It is dependent on the maximum echo tail length for which the echo needs to be eliminated.</p> <p>In order to configure the appropriate echo tail length, the <code>echo_delay</code> constant in the <code>aec.h</code> or <code>aec.inc</code> file must be modified by the user.</p> <p>The <code>echo_mem</code> memory block must have a size equal to: $((8 * \text{echo_delay}) + 47)$ long integers or $((32 * \text{echo_delay}) + 188)$ bytes.</p> <p>This memory block must be allocated in X data memory.</p>
<code>vad_mem</code>	<p>Pointer to a 296-byte (74 long integers) block of static variables in RAM used by the Voice Activity Detection algorithm (described in the next section). This memory block must be allocated by the user application in X data memory.</p>
<code>rin_in</code>	<p>Pointer to the RAM buffer containing the current processing frame of 80 samples of the Receive-In (RIN) signal.</p>
<code>sin_in</code>	<p>Pointer to the RAM buffer containing the current processing frame of 80 samples of the Send-In (SIN) signal.</p>
<code>inhibit_flag</code>	<p>A flag variable which determines if adaptation of filter coefficients will be performed for the current processing frame. If the argument value passed is '1', adaptation is inhibited; if the value passed is '0', filter adaptation is performed. This parameter is primarily useful for application debugging purposes; during normal operation this argument should be '1'.</p>
<code>yscrptr</code>	<p>Pointer to a block of RAM utilized by the AEC algorithm. This memory block must be allocated by the user application in Y data memory and must have a size equal to $(\text{ORDER} + \text{FRAME})$ words or $(2 * (\text{ORDER} + \text{FRAME}))$ bytes, where <code>FRAME</code> = 80.</p>

Output Argument:

<code>sout</code>	<p>Pointer to the RAM buffer containing the current processing frame of 80 samples of the Send-Out (SOUT) signal.</p>
-------------------	---

Return Value: None

dsPIC30F Acoustic Echo Cancellation Library User's Guide

NOTES:

Chapter 4. Resource Requirements

4.1 INTRODUCTION

This chapter describes the resources required to support the AEC Library.

4.2 HIGHLIGHTS

This chapter includes information on these topics:

- Program Memory Requirements
- Data Memory Requirements
- Computational Speed Requirements

4.3 PROGRAM MEMORY REQUIREMENTS

All lookup tables used by the library functions are located in program memory. lists the program memory requirements.

TABLE 4-1: PROGRAM MEMORY REQUIREMENTS

Function	Memory Required
Code in Program Memory	5180 bytes
Tables in Program Memory	860 bytes

dsPIC30F Acoustic Echo Cancellation Library User's Guide

4.4 DATA MEMORY REQUIREMENTS

The various blocks of data memory that the user application is required to allocate are described in the API description. The number of bytes of RAM used by each of these memory blocks is listed in Table 4-2 for different echo tail lengths (16, 32 and 64 ms). The memory block pointer names used in the API section have been used in this table.

TABLE 4-2: DATA MEMORY REQUIREMENTS

16 ms echo tail length:		
ECHO_MEM	700 bytes	(X data memory)
VAD_MEM	296 bytes	(X data memory)
SCRATCH_MEM	736 bytes	(X data memory)
YSCRPTR	416 bytes	(Y data memory)
RIN_IN	160 bytes	
SIN_IN	160 bytes	
SOUT	160 bytes	
Total RAM Usage	2628 bytes	
32 ms echo tail length:		
ECHO_MEM	916 bytes	(X data memory)
VAD_MEM	296 bytes	(X data memory)
SCRATCH_MEM	992 bytes	(X data memory)
YSCRPTR	672 bytes	(Y data memory)
RIN_IN	160 bytes	
SIN_IN	160 bytes	
SOUT	160 bytes	
Total RAM Usage	3356 bytes	
64 ms echo tail length:		
ECHO_MEM	2236 bytes	(X data memory)
VAD_MEM	296 bytes	(X data memory)
SCRATCH_MEM	1504 bytes	(X data memory)
YSCRPTR	1184 bytes	(Y data memory)
RIN_IN	160 bytes	
SIN_IN	160 bytes	
SOUT	160 bytes	
Total RAM Usage	5700 bytes	

4.5 COMPUTATIONAL SPEED REQUIREMENTS

Million Instructions Per Second (MIPS) required on a dsPIC30F device are listed in Table 4-3 for different echo tail lengths (without Howling Control enabled).

TABLE 4-3: COMPUTATIONAL SPEED REQUIREMENTS

Echo Tail Length	MIPS
16 ms	7.5
32 ms	10.5
64 ms	16.5

Chapter 5. Acoustic Echo Cancellation Algorithm

5.1 INTRODUCTION

This chapter describes the acoustic echo cancellation algorithms contained in the AEC Library.

5.2 HIGHLIGHTS

This chapter includes information on these topics:

- AEC Algorithm Overview
- NLMS Algorithm
- Double Talk Detection
- Non-Linear Processor
- Howling Control (Optional)

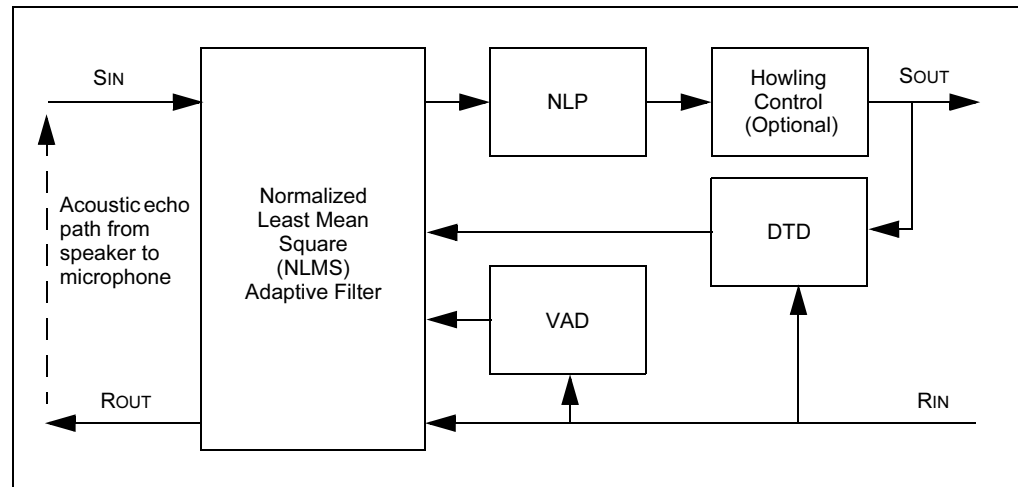
5.3 AEC ALGORITHM OVERVIEW

The Acoustic Echo Cancellation (AEC) algorithm can be divided broadly into these functions:

- Normalized Least Mean Square (NLMS) Adaptation Algorithm
- Double Talk Detector (DTD)
- Voice Activity Detector (VAD)
- Non-Linear Processor (NLP)
- Howling Control

A conceptual block diagram illustrating the operation of the AEC algorithm is shown in Figure 5-1.

FIGURE 5-1: CONCEPTUAL BLOCK DIAGRAM OF AEC ALGORITHM



dsPIC30F Acoustic Echo Cancellation Library User's Guide

A typical AEC system involves four signals:

- Far-end speech receive input (RIN),
- Near-end speech send output (SOUT),
- Far-end speech output (ROUT), usually sent to a local speaker
- Near-end speech input (SIN), usually received from a local microphone.

In systems in which the near-end speaker and microphone do not have sufficient acoustic separation, the SIN signal not only contains the microphone input (presumably spoken by a talker at the near-end), but also an undesirable echo generated by the acoustic path from the speaker to the microphone. This signal is then transmitted to the far-end through the communication channel (wired or wireless), with the result that the listener at the other end hears a perceptible echo of his/her own speech. Traditionally, this problem had to be avoided by allowing only one person to talk at any given time (i.e., by not allowing "double talk").

An Acoustic Echo Cancellation algorithm consists of an adaptive filter and various associated control functions, which not only eliminate the acoustic echo but also enable double talk (i.e., full-duplex operation). This algorithm operates at the same communicating node at which the echo was generated. The control functions used in the AEC algorithm in conjunction with the Normalized LMS adaptive filter are Double Talk Detector, Voice Activity Detector, Non-Linear Processor and Howling Control.

Normalized Least Mean Squares (NLMS) is the fundamental adaptation algorithm used for estimating and cancelling out the acoustic echo. However, the filter adaptation should only be performed when a far-end speech signal is received through the communication channel. This decision is taken by the Voice Activity Detector (VAD) block.

Moreover, when the two talkers on both sides (RIN and SIN) speak at the same time, it is known as a double-talk condition. During double talk, the SIN signal acts as uncorrelated noise and may cause the adaptive filter to diverge from its desired state, thereby reducing the efficacy of the AEC algorithm. This is prevented by the Double-Talk Detection (DTD) block.

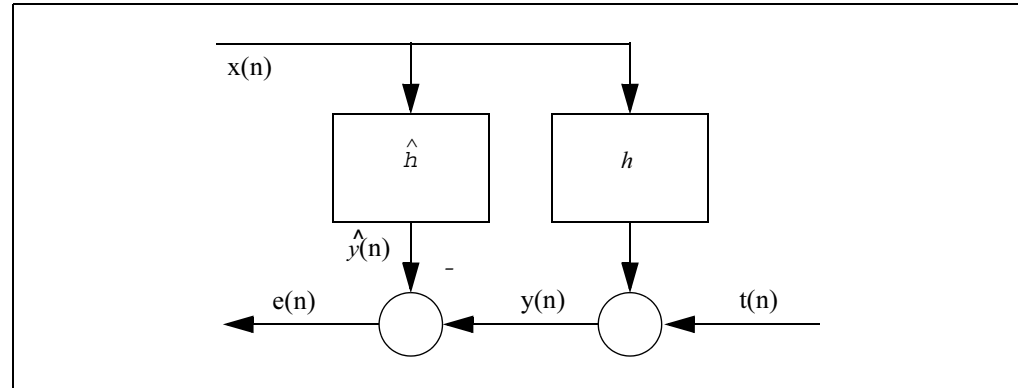
A Non-Linear Processor (NLP) block is required to remove any residual echo, because the adaptive filter in itself can not model non-linearities in the echo path.

Acoustic Echo Cancellation Algorithm

5.4 NLMS ALGORITHM

Figure 3.2 illustrates the fundamental principle of Acoustic Echo Cancellation. The far-end signal $x(n)$ traverses the acoustic echo path between the speaker and the microphone (with its impulse response represented here as h), thereby generating an echo. The signal $y(n)$ is the sum of near-end speech $t(n)$ and the echo. The signal $y(n)$ is the near-end signal.

FIGURE 5-2: GENERIC ECHO CANCELLER



The AEC algorithm filters the far-end speech $x(n)$ using a continuously-adapting estimate (with its impulse response represented as \hat{h}) of the echo path. It then subtracts the output of this estimated echo path from the 'output' (the near-end speech input) of the actual echo path, thereby effectively eliminating the echo from the resultant signal which can then be transmitted over the communication channel.

5.4.1 Bandpass Filters

Bandpass filters are used to remove unwanted low frequency and high frequency components of the SIN and RIN signals. The cutoff frequencies for both the bandpass filters are 100 Hz and 3.2 kHz.

The bandpass filtering used for this signal pre-processing operation is performed using a cascade of a 2nd order High Pass Filter (HPF) and a 2nd order Low Pass Filter (LPF), with the resultant filter transfer function being Equation 5-1:

EQUATION 5-1:

$$S(z) = S_1 \cdot S_2$$

The transfer function for the HPF is given by Equation 5-2

EQUATION 5-2:

$$S_1(z) = \frac{0.94597 - (1.89195z^{-1} + 0.94597z^{-2})}{1 + 1.88903z^{-1} - 0.89487z^{-2}}$$

The transfer function for the LPF is given by Equation 5-3

EQUATION 5-3:

$$S_2(z) = \frac{0.63894 + 1.27789z^{-1} + 0.63894z^{-2}}{1 - (1.14298z^{-1} - 0.41280z^{-2})}$$

5.4.2 Decorrelation Filters

To avoid degradation of the performance of the NLMS algorithm due to the strong correlations of speech signals, each of the speech inputs are “pre-whitened” by applying a Decorrelation Filter, before passing it to the adaptive filter.

A Decorrelation Filter is a prediction error filter with its coefficients matched to the correlation properties of the speech signal. Where speech signals are concerned, this serves to increase the convergence rate of the coefficients of the adaptive filter.

A 1st order Decorrelation Filter is used to decorrelate the input signal:

$$y(n) = x(n) - (\alpha) \cdot x(n-1)$$

An inverse de-correlation filter of the 1st order decorrelation filter is used at the other end to recover the signal.

5.4.3 Adaptive Filter

Adaptive step-size control is used to improve the filter performance for speech and to better deal with noisy environments. The step-size μ can be calculated for every 10 ms (80-sample) frame based on the residual echo energy from the previous frame.

$$\text{step_size} = (1 / (1 / \text{step_factor} + \text{prev_sout_egy}))$$

where:

step_size = Variable step size used for adaptation of NLMS filter,

prev_sout_egy = Energy contained in previous frame of residual echo,

step_factor = Constant term corresponding to a step size of 0.732421875.

The purpose of the prev_sout_egy term is to decrease the step size during double talk and noisy conditions when the energy of the frame temporarily increases. Therefore, using adaptive step-size control serves to improve the robustness of the adaptive filter.

The impulse response estimate is updated as shown in Equation 5-4.

EQUATION 5-4:

$$\hat{h}_k(n+1) = \hat{h}_k(n) + \left[\frac{\mu \cdot e(n)}{N \cdot (E_x(n) + \delta)} \right] \cdot x(n-k)$$

where:

N is the length of the NLMS adaptive filter (512, 256 or 128 taps in this case),

e(n) is the difference between the near-end signal y(n) and the predicted signal $\hat{y}(n)$, given by Equation 5-5 and Equation 5-6,

μ is referred to as the adaptive filter loop gain or step-size parameter ($0 < \mu < 2$ for NLMS),

δ is a regularization parameter that prevents division by zero, and

$E_x(n)$ is the average far-end energy, given by Equation 5-7.

EQUATION 5-5:

$$\hat{y}(n) = \sum_{k=0}^{N-1} \hat{h}_k(n) x(n-k)$$

EQUATION 5-6:

$$e(n) = y(n) - \hat{y}(n)$$

Acoustic Echo Cancellation Algorithm

EQUATION 5-7:

$$E_x(n) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} x^2(n-k)$$

μ should be any value between 0 and 2, so that stability of the NLMS algorithm is ensured. However, a suitable value should be used such that the convergence rate of the adaptation algorithm is fast, divergence rate in the case of double talk is slow and the steady state (residual) error is minimum. The effective value of μ used by the algorithm depends on the average far-end energy.

The predicted signal $\hat{y}(n)$ is obtained by filtering the signal $x(n)$ with a filter having a transfer function. Thereafter, the error $e(n)$ is calculated. The average far-end energy $E_x(n)$ is calculated as described by Equation 5-7.

To optimize the calculations for NLMS filter coefficient updates, the AEC algorithm uses the previously calculated error $e(n-1)$, and also uses the previous input signal vector $x(n-1)$.

Thus, Equation 5-4 is transformed into:

EQUATION 5-8:

$$\hat{h}_k(n+1) = \hat{h}_k(n) + \left[\frac{\mu \cdot e(n-1)}{N \cdot E_x(n)} \right] \cdot x(n-1-k)$$

where $E_x(n)$ is defined by $E_x(n) = m2^{-\text{exp}}$ (m is the mantissa and exp is the exponent of the floating point value $E_x(n)$).

Therefore, Equation 5-8 can also be written as Equation 5-9.

EQUATION 5-9:

$$\hat{h}_k(n+1) = \hat{h}_k(n) + \left[\frac{\mu \cdot e(n-1) \cdot 2^{-\text{SF}} \cdot 2^{-p} \cdot 2^{\text{exp}}}{m} \right] \cdot x(n-1-k)$$

Where:

$$\text{SF} = \log_2(N)$$

$$p = 1 \quad \text{for } E_x(n) > 0x00008000$$

$$p = 2 \quad \text{for } 0x00008000 < E_x(n) < 0x00000800$$

$$\text{Effective Adaptation Step Size is } \mu_p = \mu 2^{-p}$$

Thus, the step size is automatically scaled to obtain a low step size when the far-end energy $E_x(n)$ is low. In general, low far-end energy indicates the absence of far-end speech. When there is no far end speech, the step size gets reduced to prevent divergence of the filter. The constants 0x00008000 and 0x800 represent thresholds for far-end energy, based on which the adaptation step size is altered.

Equation 5-9 is used for implementing the adaptation of the NLMS adaptive filter.

dsPIC30F Acoustic Echo Cancellation Library User's Guide

The parameters associated with a specified maximum echo delay 'td' in ms, for a sampling rate Fs, are shown in Table 5-1:

TABLE 5-1: ADAPTIVE FILTER PARAMETERS

Constant Parameters	Description	Value
N	Number of taps of the filter	$t_d * F_s$
SF	Associated with the implementation of Equation 5-9	$\log_2(N)$
δ	Associated with the implementation of Equation 5-4	N

5.5 DOUBLE TALK DETECTION

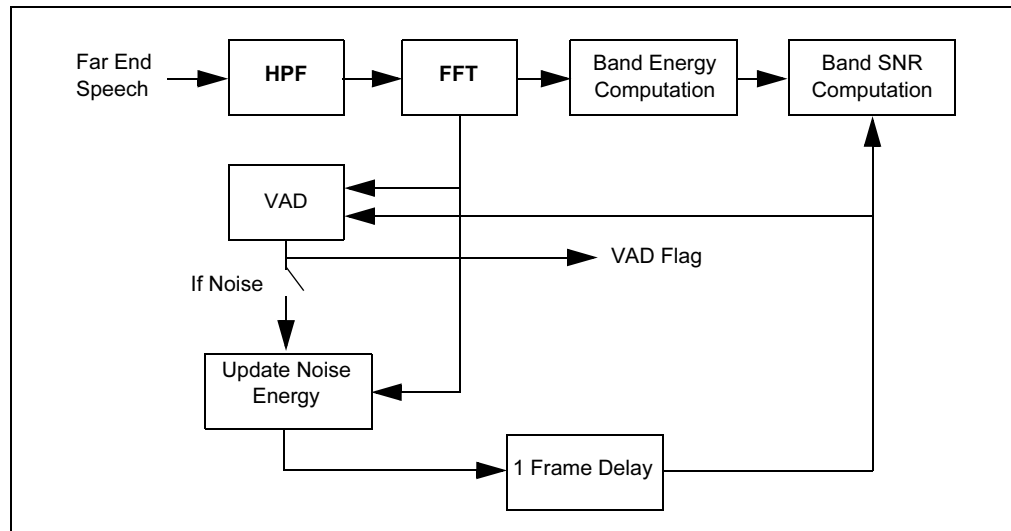
Double Talk is the condition that occurs as a result of two talkers on both sides (RIN and SIN) talking at the same time. During double talk, the signal SIN acts like uncorrelated noise and may cause the coefficients of the NLMS adaptive filter to diverge, thereby failing to effectively cancel the acoustic echo. To prevent such a condition, a Double Talk Detector (DTD) is used to inhibit adaptation of the filter during periods of simultaneous far-end and near-end speech. In this algorithm, an energy-based double talk detector is used, in which double talk is detected when Average Energy of SOUT > Average Energy of RIN

A hang-over time of one 10 ms frame is specified so that if double talk is detected the adaptation is inhibited for one frame beyond the detected end of the double talk condition.

5.6 VOICE ACTIVATION DETECTION

Voice Activity Detection (VAD) is used to detect the presence of far-end speech. VAD uses the background noise energy estimate and energy of the current RIN frame to detect speech. A block diagram of the VAD algorithm is shown in Figure 5-3.

FIGURE 5-3: VOICE ACTIVATION DETECTOR



Acoustic Echo Cancellation Algorithm

5.6.1 High Pass Filter

A High Pass Filter with a cut-off frequency of 80 Hz is used to remove DC and other low frequency components of the far-end speech signal. The output of the HPF is denoted as $s'(n)$.

The transfer function of the LPF is described by Equation 5-10.

EQUATION 5-10:

$$H(z) = 0.5 \frac{0.92727435 - 1.8544941z^{-1} + 0.92727435z^{-2}}{1 - 1.9059465z^{-1} + 0.9114024z^{-2}}$$

5.6.2 Fast Fourier Transform

The input samples are pre-emphasized using Equation 5-11.

EQUATION 5-11:

$$d(n) = s'(n) + \xi(s'(n-1))$$

Where: $\xi = -0.8$.

Pre-emphasized samples are buffered for 10 ms and overlapped with 3 ms of signal samples from the previous frame, thus yielding 104 samples. The resulting buffer is zero-padded with 24 zeros, thereby forming a 128-sample buffer. This buffer is windowed using a smoothed trapezoidal window, as defined in Table 5-2.

TABLE 5-2: BUFFER WINDOW COEFFICIENTS

Window Coefficient	For
$\sin^2(\pi(n+0.5)/2D)$	$0 \leq n < D$
1	$D \leq n < L$
$\sin^2(\pi(n-L+D+0.5)/2D)$	$L \leq n < D+L$
0	$D+L \leq n < M$

Where:

D is the number of overlapped samples from previous frame = 24

L is the number of samples per frame = 80

M is the FFT length = 128

A 64-point Complex Fast Fourier Transform (FFT) is performed on the 128-point real buffer, and some post-processing of the 64-point Complex FFT output is performed in order to obtain a 128-point Real FFT output vector. A detailed description of the post-processing method is given below.

Since the FFT algorithm can handle complex-valued input sequences, we can exploit this capability in the computation of the DFT of two real-valued sequences.

Suppose $x_1(n)$ and $x_2(n)$ are two real-valued sequences of length N, and $x(n)$ is a complex-valued sequence defined as Equation 5-12.

EQUATION 5-12:

$$x(n) = x_1(n) + jx_2(n)$$

For $0 \leq n \leq N-1$

dsPIC30F Acoustic Echo Cancellation Library User's Guide

Discrete Fourier Transform (DFT) is a linear operation, and hence the DFT of $x(n)$ may be expressed as Equation 5-13.

EQUATION 5-13:

$$X(k) = X_1(k) + jX_2(k)$$

The sequences $x_1(n)$ and $x_2(n)$ can be expressed in terms of $x(n)$ as in Equation 5-14 and Equation 5-15, respectively.

EQUATION 5-14:

$$x_1(n) = (x(n) + x^*(n))/2$$

EQUATION 5-15:

$$x_2(n) = (x(n) - x^*(n))/2$$

Where $x^*(n)$ is the conjugate of $x(n)$.

Hence, the DFTs of $x_1(n)$ and $x_2(n)$ are expressed as Equation 5-16 and Equation 5-17.

EQUATION 5-16:

$$X_1(n) = \{ \text{DFT}(x(n)) + \text{DFT}(x^*(n)) \} / 2$$

EQUATION 5-17:

$$X_2(n) = \{ \text{DFT}(x(n)) - \text{DFT}(x^*(n)) \} / 2j$$

Since the DFT of $x^*(n)$ is $X^*(N-k)$, the above equations become Equation 5-18 and Equation 5-19, respectively.

EQUATION 5-18:

$$X_1(K) = \{ x(K) + x^*(N-k) \} / 2$$

EQUATION 5-19:

$$X_2(k) = \{ X(k) - X^*(N-k) \} / 2j$$

Thus, by performing a single DFT on a complex-valued sequence $x(n)$, we have obtained the DFT of two sequences, with only a small amount of additional computation involved in computing $X_1(k)$ and $X_2(k)$ from $X(k)$.

Acoustic Echo Cancellation Algorithm

Suppose $g(n)$ is a real valued sequence of $2N$ points. Let us divide or decimate $g(n)$ into 2 N point sequences, as given below:

$$\begin{aligned}x_1(n) &= g(2n) \\x_2(n) &= g(2n+1)\end{aligned}$$

Thus we have subdivided the $2N$ -point real sequence into two N -point real sequences, on which we can apply the method described above.

Let $x(n)$ be the N -point complex-valued sequence:

$$x(n) = x_1(n) + jx_2(n)$$

Using the above results, we have:

$$X_1(k) = \{X(k) + X^*(N-k)\} / 2$$

$$X_2(k) = \{X(k) - X^*(N-k)\} / 2j$$

Finally, we must express the $2N$ point DFT in terms of two N point DFTs, $X_1(k)$ and $X_2(k)$. To accomplish this, we employ a method similar to the decimation in-time FFT algorithm, as shown in Equation 5-20.

EQUATION 5-20:

$$G(k) = \sum_{n=0}^{n=N-1} g(2n) W_{2N}^{2nk} + \sum_{n=0}^{n=N-1} g(2n+1) W_{2N}^{2n+1k}$$

EQUATION 5-21:

$$= \sum_{n=0}^{n=N-1} x_1(n) W_N^{nk} + W_{2N}^k \sum_{n=0}^{n=N-1} x_2(n) W_N^{nk}$$

Consequently:

EQUATION 5-22:

$$G(k) = X_1(k) + W_{2N}^k X_2(k)$$

EQUATION 5-23:

$$G(k+N) = X_1(k) - W_{2N}^k X_2(k)$$

Where $k = 0, 1, \dots, N-1$

Since the sequence is real, its FFT is an even function. Therefore, out of the 128 complex elements in the output, we only need to compute the first 64 elements.

5.6.3 Band Energy Computation

The frequency spectrum of the far-end speech signal (obtained using FFTR as described above) is divided into 16 non-uniform bands. The energy in each band is computed using leaky integration. The spectrum of the signal is divided as shown in Table 5-3:

TABLE 5-3: BAND ENERGY SIGNAL SPECTRUM

Band	Start Frequency (FFT Index) FL	End Frequency (FFT Index) FH
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	16
8	17	19
9	20	22
10	23	26
11	27	30
12	31	35
13	36	41
14	42	48
15	49	55
16	56	63

Each band energy is independently computed as Equation 5-24:

EQUATION 5-24:

$$E_{ch}(m,i) = \max \left\{ E_{\min}, \alpha_{ch} E_{ch}(m-1,i) + (1 - \alpha_{ch}) \cdot \frac{\sum_{k=f_L(i)}^{k=f_H(i)} |G(k)|^2}{f_H(i) - f_L(i) + 1} \right\}$$

Where:

- $\alpha_{ch} = 0$ for first frame and 0.55 for all other frames,
- $E_{\min} = 0.0625$, $G(k)$ is the FFT of the input signal, and
- m denotes the current Rin frame.

Acoustic Echo Cancellation Algorithm

5.6.4 Band SNR Computation

For each of the frequency bands, the Signal-to-Noise Ratio (SNR) is computed using:

- the signal band energy computed in the Band Energy Computation stage, and
- the noise band energy estimate (which is updated during noise only frames).

For each frequency band, the SNR thus obtained is used to compute the corresponding scale factor.

If $E_n(m,i)$ is the noise energy estimate for frame m , then the Scale Factor for band i is given by:

EQUATION 5-25:

$$\sigma(i) = \max\left\{0, \text{round}\left(10\log\left(\frac{E_{ch}(m,i)}{E_n(m,i)}\right)\right)\right\}$$

5.6.5 VAD and Noise Band Energy Computation

The band energies of the current R_{in} frame and the noise band energies are compared, in order to classify the current frame as either a 'noise frame' or a 'speech frame'. A hangover of six frames is applied for transition from speech frames to noise frames. If the current frame is declared as a noise frame by the VAD function, then the noise band energy is updated.

For the first 12 frames, the noise band energy is updated continuously using band energy irrespective of whether these frames are noise only frames or noisy speech frames.

The following procedure is used to determine if the current frame is a noise-only frame.

- A count is set to zero. Each signal band energy is compared with the corresponding noise band energy.
- If the signal band energy is greater than 1.5 times noise band energy, the count is incremented.
- If the count is greater than 5 for the current frame (i.e., if 5 of the 16 band energies are more than 1.5 times the corresponding noise band energies) the current frame is declared as a speech frame.

Otherwise, the current frame is determined to be a noise frame, and the Noise band energy is updated using the leaky integration method given by Equation 5-26.

EQUATION 5-26:

$$E_n(m+1,i) = \max\{E_{min}, (\alpha_n E_n(m,i)) + (1 - \epsilon_n) E_{cn}(m,i)\} \quad 0 \geq i > N_c$$

5.7 NON-LINEAR PROCESSOR

The AEC algorithm by itself may not be capable of adequately modeling echo paths that generate significant levels of non-linear distortion. This necessitates the usage of a Non-Linear Processor (NLP).

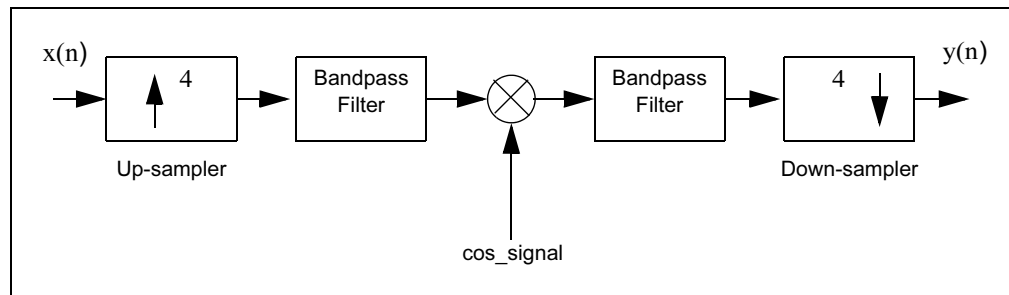
The function of the NLP is to substantially suppress the residual echo level which remains at the output of the NLMS adaptive filter, so that a very low returned echo level can be achieved even if the echo path is non-linear. The NLP is located in the send path between the output of the NLMS filter and the Sout port of the system. The NLP basically attenuates low-level signals (which are assumed to be residual echo) by 18 dB, and passes high-level signals (which are assumed to be desirable near-end speech).

A hang-over count of three frames (30 ms) is applied from the time a double talk condition is detected.

5.8 HOWLING CONTROL (OPTIONAL)

Howling is a typical problem in full duplex communication. It builds up due to the acoustic feedback path. One way to reduce howling is to shift the frequency of the signal that is picked up by the microphone by 10 Hz to 20 Hz, before it is sent out over a communication channel. This shift is usually not perceived as unnatural by the human ear. The shifted signal appears at the destination loudspeakers and travels back to the originator shifted by another 10 Hz to 20 Hz. The signal travels many times through this acoustic path and is quickly shifted out of the passband, thus reducing the problem of unpleasant feedback. A block diagram of the Howling Control function is shown below.

FIGURE 5-4: HOWLING CONTROL ALGORITHM



The up-sampler fills in three zeros between the adjacent signal samples $x(n)$ and $x(n+1)$.

The bandpass filter operates at four times the sampling frequency of input, retaining the components of the spectrum that are supposed to be frequency shifted by 'fh' Hz. The actual shift is performed by multiplication with an appropriate cosine signal. The bandpass filter removes some artifacts that appear due to the shifting operation and the down sampler takes every fourth sample of the band pass result to yield the output signal at the original sampling rate.

The cutoff frequencies of the 1st bandpass filter are 8.1 kHz and 11.3 kHz, and the cosine signal used for the shifting operation is a $(8000 - fh)$ Hz signal. The bandpass filter used here is a 10th order IIR filter. This 10th order filter is implemented as a cascade of five 2nd order filters.

The cutoff frequencies of the 2nd bandpass filter are 100 Hz and 3.3 kHz. The bandpass filter used here is a 4th order IIR filter, which is implemented as a cascade of two 2nd order filters.

Acoustic Echo Cancellation Algorithm

The transfer function of a general Nth order IIR filter is given by Equation 5-27.

EQUATION 5-27:

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{1 - \sum_{k=1}^{N-1} a_k z^{-k}}$$

Where: N = order of the IIR filter
 b_k = zeros of the IIR filter (b coefficients)
 a_k = poles of the IIR filter (a coefficients)

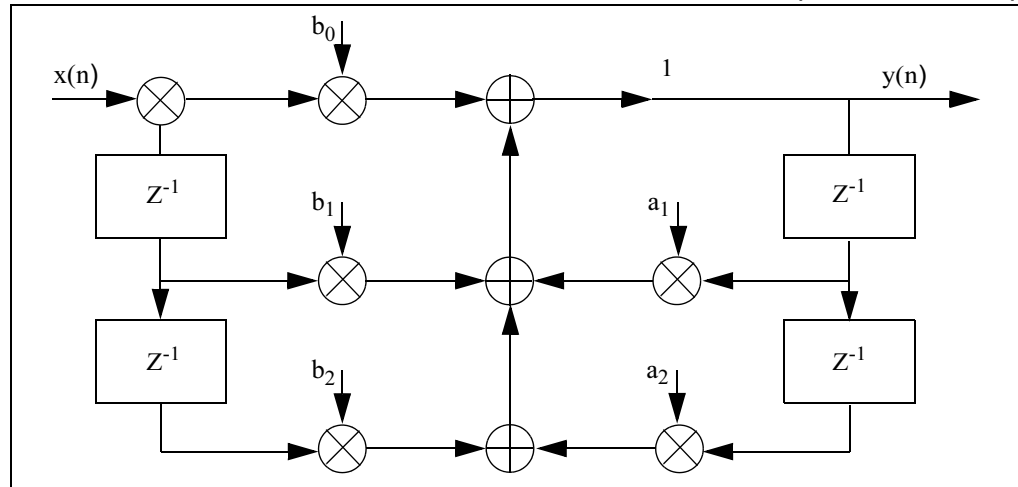
The transfer function of a 2nd order IIR filter is given by Equation 5-28.

EQUATION 5-28:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - (a_1 z^{-1} + a_2 z^{-2})}$$

The signal flow graph of a second order IIR filter, implemented in DIRECT FORM 1 is shown in Figure 5-5.

FIGURE 5-5: SIGNAL FLOW OF 2ND ORDER IIR FILTER (DIRECT FORM 1)



The transfer function G(z) of the first bandpass filter (BPF_1) can be expressed as Equation 5-29.

EQUATION 5-29:

$$G(z) = H_1 \cdot H_2 \cdot H_3 \cdot H_4 \cdot H_5$$

The transfer function of the 1st stage of BPF_1 is given by Equation 5-30.

EQUATION 5-30:

$$H_1(z) = \frac{1 - 0.00035z^{-1} - 0.0245184z^{-2}}{1 - 0.5200z^{-1} - 0.5095z^{-2}}$$

dsPIC30F Acoustic Echo Cancellation Library User's Guide

The transfer function of the 2nd stage of BPF_1 is given by Equation 5-31.

EQUATION 5-31:

$$H_2(z) = \frac{0.18286 - 0.3660z + 0.18314z^{-2}}{1 - 0.23306z^{-1} - 0.57366z^{-2}}$$

The transfer function of the 3rd stage of BPF_1 is given by Equation 5-32.

EQUATION 5-32:

$$H_3(z) = \frac{0.14946 - 0.29883z^{-1} + 0.14937z^{-2}}{1 - 0.83368z^{-1} - 0.61961z^{-2}}$$

The transfer function of the 4th stage of BPF_1 is given by Equation 5-33.

EQUATION 5-33:

$$H_4(z) = \frac{0.38245 - 0.76503z^{-1} + 0.38257z^{-2}}{1 - 0.06182z^{-1} - 0.81573z^{-2}}$$

The transfer function of the 5th stage of BPF_1 is given by Equation 5-34.

EQUATION 5-34:

$$H_5(z) = \frac{0.50020 + z^{-1} + 0.49980z^{-2}}{1 - 1.09887z^{-1} - 0.849007z^{-2}}$$

Similarly, the second bandpass filter (BFF_2) can be expressed as: Equation 5-35.

EQUATION 5-35:

$$T(z) = T_1 \cdot T_2$$

The transfer function of the 1st stage of BPF_2 is given by Equation 5-36.

EQUATION 5-36:

$$T_1(z) = \frac{0.068460 + 0.13692z^{-1} + 0.06846z^{-2}}{1 + 1.1500138z^{-1} - 0.42438z^{-2}}$$

The transfer function of the 2nd stage of BPF_2 is given by Equation 5-37.

EQUATION 5-37:

$$T_2(z) = \frac{0.98531 - 1.9706z^{-1} + 0.06846z^{-2}}{1 + 1.1500138z^{-1} - 0.42438z^{-2}}$$

Chapter 6. AEC Demonstration

6.1 INTRODUCTION

This chapter provides a hands-on demonstration of acoustic echo cancellation in a working application.

6.2 HIGHLIGHTS

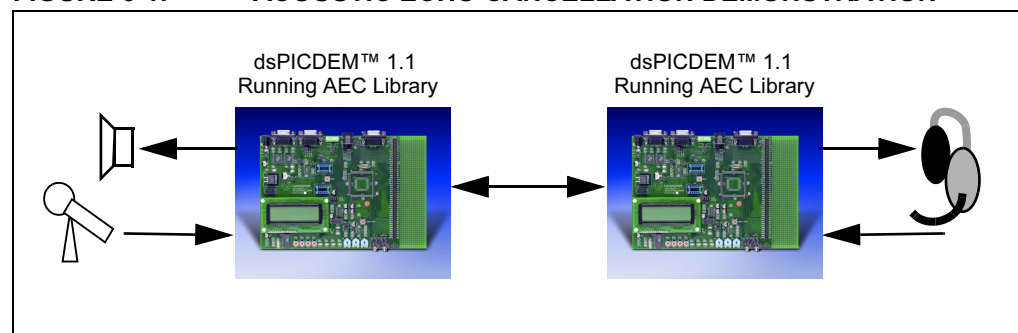
This chapter includes information on these topics:

- Demonstration Summary
- Demonstration Setup
- Demonstration Procedure
- Demo Code Description

6.3 DEMONSTRATION SUMMARY

To demonstrate the functionality of the AEC Library, a sample application emulating two speaker phones engaged in voice communication is provided with the library. This software requires the use of two dsPICDEM 1.1 boards (not included with the software license). The two dsPICDEM 1.1 development boards are configured as typical communication nodes using the components included in the optional Accessory Kit, as shown in Figure 6-1. The RS-232 cable with a null modem adapter serves as the communication channel.

FIGURE 6-1: ACOUSTIC ECHO CANCELLATION DEMONSTRATION



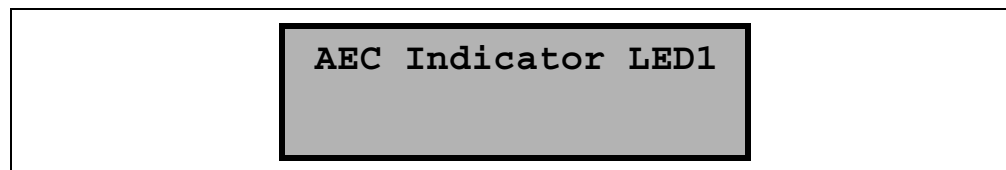
dsPIC30F Acoustic Echo Cancellation Library User's Guide

A speaker and a microphone are connected to dsPICDEM 1.1 Board #1 and located in proximity to each other. A headset is connected to Board #2. When a person talks into the headset connected to Board #2, the speech signal is sampled through the on-board Si3000 voice band codec and the Data Converter Interface (DCI) module of the dsPIC30F device. The dsPIC30F device then compresses the signal using μ -Law compression and transmits the compressed speech signal through its UART1 module and the on-board RS-232 transceiver to Board #1.

The dsPIC30F device on Board #1 receives the signal through the on-board RS-232 transceiver and the device's UART1 module and decompresses the signal using μ -Law decompression. The dsPIC30F device then plays out the signal on the speaker through its DCI module and on-board Si3000 codec. Due to the proximity of the speaker to the microphone, the sound from the speaker enters the microphone and is sampled by the dsPIC30F device through the codec. The device then compresses the microphone input signal and transmits it to Board #2 via the RS-232 interface. If a person is talking into the microphone connected to Board #1, the signal transmitted to Board #2 is a combination of the near-end speech and the undesirable acoustic echo of the far-end speech. This combination of speech and echo can be heard on the headset connected to Board #2.

Initially, the LCD on Board #1 displays the status shown in Figure 6-2. LED1, located beneath and to the right of the LCD, is off. Pressing switch SW1 on Board #1 invokes the `AcousticEchoCanceller` function from the AEC Library on Board #1. At this point the acoustic echo generated between the speaker and the microphone is eliminated. The headset on Board #2 now hears only the speech signal from the Board #1 microphone. The AEC algorithm, now activated, has cancelled out the echo.

FIGURE 6-2: PROGRAM STATUS DISPLAY



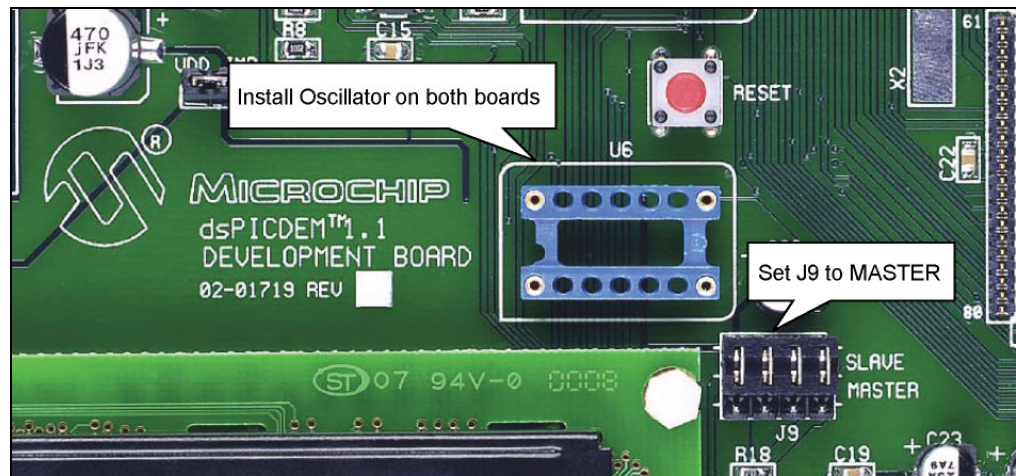
6.4 DEMONSTRATION SETUP

Follow these steps to set up the demonstration.

6.4.1 Configure dsPICDEM 1.1 Boards

1. Insert a 14.7456 MHz oscillator into oscillator socket U6 on both dsPICDEM 1.1 boards, as shown in Figure 6-3. You can use the two oscillators included in the Accessory Kit or supply your own.
2. Set the jumper marked J9 to the 'Master' position on both boards. This jumper allows the on-board Si3000 codec to function as a serial clock Master using the 14.7456 MHz oscillator.
3. Apply power to both boards.

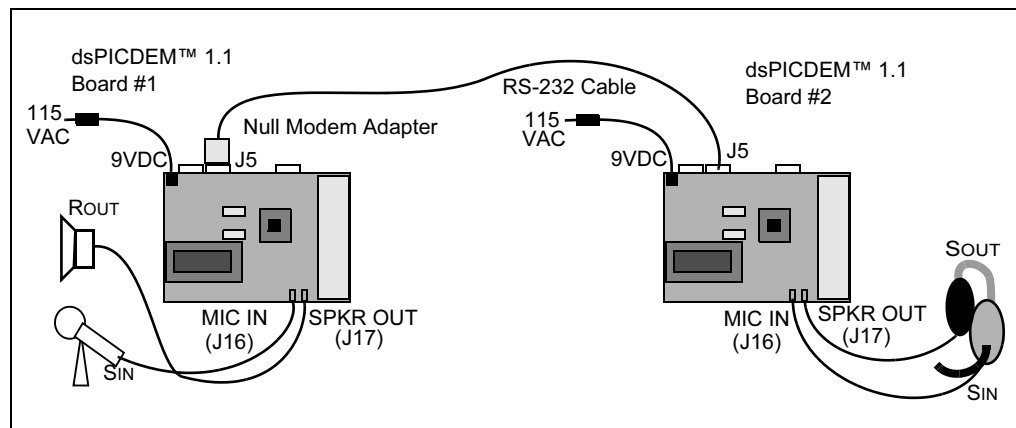
FIGURE 6-3: DEMO BOARD SETUP



6.4.2 Set up Demo

After both boards are properly configured, attach the speakers and microphones and interconnect the boards, as shown in Figure 6-4. These instructions assume you are using components from the optional Accessory Kit. You can use equivalent devices if you choose.

FIGURE 6-4: CONNECT dsPICDEM 1.1 BOARDS



dsPIC30F Acoustic Echo Cancellation Library User's Guide

1. Connect the fold-up speaker to the SPKR OUT jack (J17) on Board #1.
2. Connect the lapel microphone to the MIC IN jack (J16) on Board #1. Be sure the the microphone is turned on and is situated close enough to the speaker to generate feedback into the microphone.
3. Connect the headset microphone to the MIC IN (J16) jack on Board #2.
4. Connect the headset speaker to the SPKR OUT (J17) jack on Board #2.
5. Connect one end of the DB9M-DB9M Null Modem Adapter to PORT B (J5) on Board #1. Then connect one end of the RS-232 cable to the Null Modem Adapter.
6. Connect the other end of a 6-ft DB9 M/F RS-232 cable to the 'PORT B' (J5) port on Board #2.

6.4.3 Program dsPIC30F Device

After the boards are configured and interconnected, you must build a project in MPLAB IDE to program the demonstration software into to the dsPIC30F devices on the dsPICDEM 1.1 boards. Follow these steps:

1. Open MPLAB IDE. Open the `echodemo.mcw` workspace or `echodemo.mcp` project located in the `Demo` folder.
2. Select *Project>Build All* to build the project.

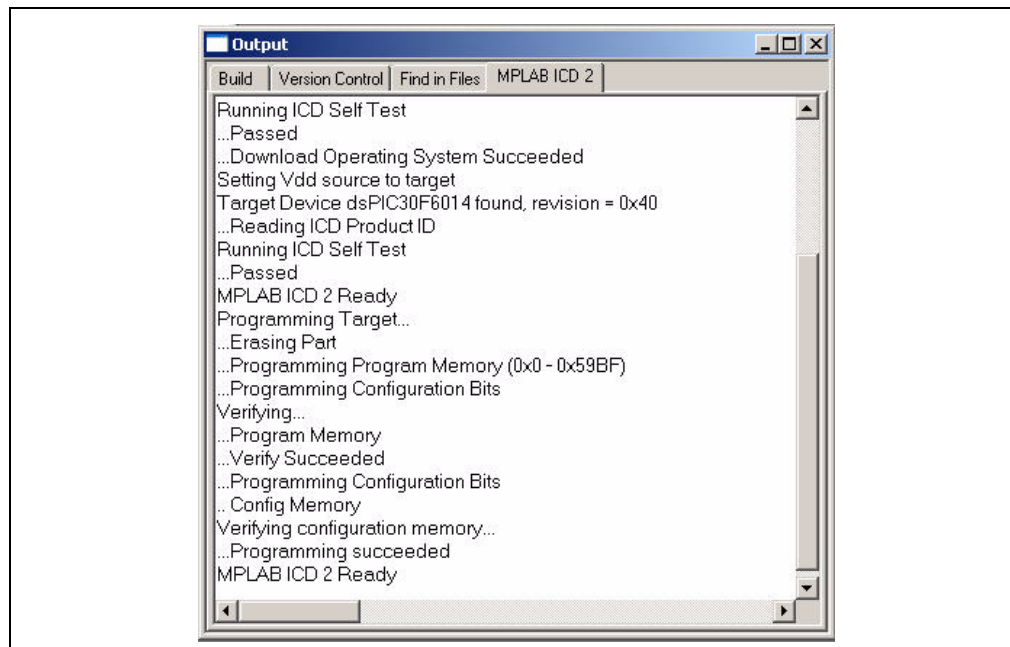
Alternatively, open MPLAB IDE, select dsPIC30F6014 as the device, select ICD2 as the Programmer, and import the `echodemo.hex` file from the `Demo` folder. For more information on using MPLAB IDE, refer to *dsPIC™ Language Tools Getting Started (DS70094B)*.

3. Connect the ICD 2 to dsPICDEM 1.1 Board #1. Program the dsPIC30F6014 device on the board (*Programmer>Program*).

The project output window displays programming status, as shown in

4. Connect ICD2 to dsPICDEM 1.1 Board #2 and program the dsPIC30F6014 device.
5. Disconnect the ICD2 from both boards. Make sure that the two boards are not located too close to each other, in order to avoid any acoustic coupling between the two boards.

FIGURE 6-5: PROGRAMMING STATUS IN OUTPUT WINDOW



6.5 DEMONSTRATION PROCEDURE

After the demo application has been programmed into both devices, the demo is ready to run. Follow these steps to run the demo:

1. Press the Reset button on both dsPICDEM 1.1 boards. The same code should now be running on both boards.
2. Put on the headset connected to dsPICDEM 1.1 board #2 and listen for a beep. This beep indicates that the codec has been initialized.
3. Start talking on the microphone input of the headset. On the speaker output of the headset, you should be able to hear an echo of your own speech.
4. If you wish, have someone simultaneously talk into the microphone connected to dsPICDEM 1.1 Board #1. In this case you would hear the other person's speech as well as an echo of your own speech.
5. Press the switch marked 'SW1' on dsPICDEM 1.1 Board #1. Observe that the LED1 on Board #1 turns on, indicating that the AEC algorithm is active.
6. Again, speak into the microphone of the headset. You should no longer hear the echo of your own speech.
7. To observe echo cancellation during double-talk, let a person simultaneously talk into the microphone connected to dsPICDEM 1.1 Board #1. You will only hear the other person's speech, free of the echo of your own speech.

Alternatively, instead of talking into the headset you can drive an audio .wav file from the speaker output of a computer to the microphone input of dsPICDEM 1.1 Board #2 through a 6-ft stereo audio cable. You can use the audio cable included in the Accessory Kit for this procedure.

To experiment with different values of maximum echo tail length, simply change the `echo_delay` constant defined in the `aec.h` include file, rebuild `echodemo.mcp`, re-program the two devices, and re-run the demo application.

6.6 DEMO CODE DESCRIPTION

A brief description of the demo code is given below, with special focus on the initialization of key peripherals such as DCI and UART. It can be executed for maximum echo tail length of 64-ms tail length (with 64 ms being the default setting of the `echo_delay` constant). The same demo code runs on two dsPIC30F6014 devices, both using the Primary Oscillator as the clock source, with the XT w/PLL 16x Primary Oscillator mode.

`echodemo.c` contains the main function for the demo application. `echodemo.c` allocates all the variables and arrays in data memory that are needed for DCI and UART data buffering, as well as the blocks of data memory that need to be allocated for the AEC Library functions. `indatabuffer` and `outdatabuffer` are UART input and output buffers. `indataRIndex`, `indataWIndex`, `outdataRIndex` and `outdataWIndex` are Read and Write indices for the `indatabuffer` and `outdatabuffer` arrays. `CodecRxBuffer` and `CodecTxBuffer` are the receive and transmit buffers for codec data.

The main function calls the `AcousticEchoCancellorInit` function from the AEC Library, which initializes the AEC algorithm.

The main function also calls the `Init_DCI()` function to initialize the DCI module, the Si3000 codec and DCI interrupt. The DCI module acts as a Slave, and the Si3000 codec acts as a Master and drives the serial clock and frame synchronization lines. 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 demo.

dsPIC30F Acoustic Echo Cancellation Library User's Guide

Subsequently, the `Init_si3000()` function is used to initialize the Si3000 codec. The codec is reset by connecting the RF6 pin of the dsPIC30F device to the Reset pin of the Si3000, holding RF6 low for 100 cycles and then bringing it high. The codec is configured for a sample rate of 8 kHz. Two 14.7456 MHz oscillators are used for driving the Si3000 codec on both the boards. The MIC Gain and Receive Gain are set to 0 dB. Both the 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 the Si3000 control registers, a delay is introduced for calibration to occur. Finally, the DCI interrupt is enabled.

UART initialization and data processing is performed by the `init_uart` function. The UART module is configured to generate an interrupt for every byte transmitted or received. The UART module is run at a baud rate of 115200 bps, with an 8-bit, no-parity, 1-Stop-bit data format (8-N-1). In the UART Transmit and Receive Interrupt Service Routines, the corresponding interrupt flag is cleared, data is either written to U1TXREG, or read from U1RXREG and saved in a circular buffer. Note that the UART code also allows the user to use an RS-422 cable (instead of an RS-232 cable and Null Modem Adapter), if a greater degree of acoustic isolation is required between the two boards.

The `intr_count` variable is incremented every time a DCI interrupt occurs. When the count is 80, the contents of the codec data buffers are copied into the `rinbuff` and `sinbuff` arrays, and the `AcousticEchoCanceller` function from the AEC Library is called. `rinbuff` contains the far-end input data to be played out on the speaker, and `sinbuff` contains the near-end microphone input. The buffering delay is compensated by using an appropriate offset. The `sout` data buffer, which is the output of the `AcousticEchoCanceller` function, is compressed using μ -law compression and transmitted using the UART1 module to the other end of the communication link.

Display of the AEC On/Off status on the LED 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. To allow operation of the SW1 switch, the INTx pins are initialized in the `Init_INTpin` function, and the ISR for the external interrupt pin INT0 is located in the `INTx_Interrupt.s` file.

Whenever SW1 is pressed, the AEC algorithm is enabled and LED1 is turned on. A flag `aecflag` is toggled in the ISR `__INT1Interrupt`. In the main loop, the value of this flag is tested; if the flag is set, the AEC output is played out to the speaker output of the headset. Otherwise the microphone input signal from the headset is directly played out on the speaker output of the headset.



dsPIC30F ACOUSTIC ECHO CANCELLATION LIBRARY USER'S GUIDE

Index

A		H	
Accessory Kit	9	High Pass Filter	29
AcousticEchoCanceller Function	8	Host System Requirements	9
AcousticEchoCancellerInit Function.....	8	I	
Adaptive Filter	26	Include Files	12
AEC Initialization Function	16	Installation Procedure	11
Algorithm		Internet Address	4
Double Talk Detection (DTD).....	8	L	
Normalized Least Mean Square (NLMS).....	8	Library Archive	12
Voice Activity Detection (VAD).....	8	Library Functions.....	15
Algorithm Overview	23	M	
Application Programming Interface.....	15	Microchip Web Site	4
Argument Data Types	15	MPLAB IDE User's Guide	4
B		N	
Bandpass Filters	25	NLMS	8
C		Normalized Least Mean Square Algorithm	8
Computational Speed Requirements	22	O	
Customer Notification Service.....	5	Overview	
Customer Support.....	6	Acoustic Echo Cancellation	7
D		P	
Data Memory Requirements	22	Program Memory Requirements	21
Decorrelation Filters	26	R	
Demo Files	13	Recommended Reading	3
Documentation		T	
Updates	2	Typical Applications	8
Double Talk Detection Algorithm.....	8	V	
DTD.....	8	VAD	8
E		Voice Activity Detection Algorithm	8
Echo Tail Length	22	W	
F		Warranty Registration	3
FIR Filter	8	WWW Address	4
G			
Generic Echo Canceller	25		



MICROCHIP

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: 480-792-7627
Web Address: <http://www.microchip.com>

Atlanta

3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago

333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

San Jose

1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

Toronto

6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia

Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing

Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu

Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou

Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR

Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai

Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen

Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde

Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao

Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India

Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaughnessy Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

Japan

Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471-6166 Fax: 81-45-471-6122

Korea

168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore

200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan

Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan

Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria

Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark

Regus Business Centre
Lautrup hoy 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France

Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - 1er Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany

Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy

Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands

P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom

505 Eskdale Road
Wokingham Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/17/04

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Microchip:](#)

[AC300030](#)