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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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’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 Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLog, microD, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, 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, IGSP, 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.
# Table of Contents

## Preface

Chapter 1. Introduction

- 1.1 Introduction ................................................................. 7
- 1.2 System Requirements ..................................................... 7
- 1.3 Software Components ..................................................... 8
- 1.4 Functional Features ....................................................... 9
- 1.5 References .................................................................. 10

Chapter 2. Installation

- 2.1 Introduction ................................................................. 11
- 2.2 System Requirements ..................................................... 11
- 2.3 Installation Procedure ................................................... 11
- 2.4 Uninstall Procedure ....................................................... 14

Chapter 3. Overview

- 3.1 Introduction ................................................................. 15
- 3.2 Overview of the dsPIC30F Speech Recognition Library .............. 15
- 3.3 Speech Recognition Process Flow .................................... 16
- 3.4 Word Library ............................................................... 18

Chapter 4. Application Program Interface

- 4.1 Introduction ................................................................. 19
- 4.2 Requirements .............................................................. 19
- 4.3 Data Structures ............................................................ 21
- 4.4 Library Functions ......................................................... 24
- 4.5 SR_Initialization( ) Function ......................................... 25
- 4.6 DCIInterrupt( ) Function ............................................... 28
- 4.7 SR_Recognizer( ) Function ............................................. 29
- 4.8 T4Interrupt( ) Function .................................................. 33
- 4.9 SR_STATUSbits Interaction .............................................. 34
- 4.10 Using External Memory ............................................... 36
- 4.11 Required dsPIC30F System Resources ............................. 37
- 4.12 Recognition Accuracy .................................................. 38

Chapter 5. Incorporating Speech Recognition

- 5.1 Introduction ................................................................. 39
- 5.2 Incorporating Speech Recognition in an Application .................. 39
- 5.3 Building the Speech Recognition Library Into Your Code ............. 47
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 chapter introduces the dsPIC30F Speech Recognition Library User’s Guide. The Speech Recognition Library enables developers to incorporate a voice-controlled user interface in their embedded solution. 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
This document describes how to use the dsPIC30F Speech Recognition Library as a voice controlled user interface to embedded solutions running on the dsPIC30F Digital Signal Controller. The manual layout is as follows:

- **Chapter 1: Introduction** – This chapter introduces the dsPIC30F Speech Recognition Library and defines the minimum system support requirements. It also describes the software library content and functionality. The chapter concludes with a list of information resources on speech recognition.

- **Chapter 2: Installation** – This chapter provides instructions for installing the dsPIC30F Speech Recognition Library.

- **Chapter 3: Overview** – This chapter provides a high-level explanation of the role of the Speech Recognition Library as a component of the user’s application.

- **Chapter 4: Application Program Interface** – This chapter provides detailed information for interfacing the dsPIC30F Speech Recognition Library with the embedded user application.

- **Chapter 5: Incorporating Speech Recognition** – This chapter identifies the timing aspects of integrating the dsPIC30F Speech Recognition Library with the user’s application.

- **Chapter 6: Demo Application** – This chapter provides instructions for setting up and running a sample speech recognition application.

- **Appendix A: Master Word Library** – This appendix lists the vocabulary that can be used for verbal control of the user's application.

- **Appendix B: External Memory Board** – This appendix provides the design for an adaptor board that allows external memory to be connected to the dsPIC30F to allow portions of the Speech Recognition Library to operate from off-chip memory.
CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

<table>
<thead>
<tr>
<th>DOCUMENTATION CONVENTIONS</th>
<th>Description</th>
<th>Represents</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arial font:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Italic characters</td>
<td>Referenced books</td>
<td><strong>MPLAB® IDE User’s Guide</strong></td>
<td>...is the only compiler...</td>
</tr>
<tr>
<td>Emphasized text</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial caps</td>
<td>A window</td>
<td>the Output window</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A dialog</td>
<td>the Settings dialog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A menu selection</td>
<td>select Enable Programmer</td>
<td></td>
</tr>
<tr>
<td>Quotes</td>
<td>A field name in a window or dialog</td>
<td>“Save project before build”</td>
<td></td>
</tr>
<tr>
<td>Underlined, italic text with right angle bracket</td>
<td>A menu path</td>
<td><strong>File&gt;Save</strong></td>
<td></td>
</tr>
<tr>
<td>Bold characters</td>
<td>A dialog button</td>
<td>Click <strong>OK</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A tab</td>
<td>Click the <strong>Power</strong> tab</td>
<td></td>
</tr>
<tr>
<td>&quot;bnnnn&quot;</td>
<td>A binary number where n is a digit</td>
<td>b00100, ‘b10</td>
<td></td>
</tr>
<tr>
<td>Text in angle brackets &lt; &gt;</td>
<td>A key on the keyboard</td>
<td>Press &lt;Enter&gt;, &lt;F1&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>Courier font:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plain Courier</td>
<td>Sample source code</td>
<td>#define START</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Filenames</td>
<td>autoexec.bat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>File paths</td>
<td>c:\mcc18\h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keywords</td>
<td>_asm, _endasm, static</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command-line options</td>
<td>-Opa+, -Opa-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit values</td>
<td>0, 1</td>
<td></td>
</tr>
<tr>
<td>Italic Courier</td>
<td>A variable argument</td>
<td>file.o, where file can be any valid filename</td>
<td></td>
</tr>
<tr>
<td>&quot;0xnnn&quot;</td>
<td>A hexadecimal number where n is a hexadecimal digit</td>
<td>0xFFFF, 0x007A</td>
<td></td>
</tr>
<tr>
<td>Square brackets [ ]</td>
<td>Optional arguments</td>
<td>mcc18 [options] file [options]</td>
<td></td>
</tr>
<tr>
<td>Curly brackets and pipe character: {</td>
<td>Choice of mutually exclusive arguments; an OR selection</td>
<td>errorlevel {0</td>
<td>1}</td>
</tr>
<tr>
<td>Ellipses...</td>
<td>Represents repeated text</td>
<td>var_name [, var_name...]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Represents code supplied by user</td>
<td>void main (void) { ... }</td>
<td></td>
</tr>
</tbody>
</table>

WARRANTY REGISTRATION

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

This user's guide describes how to use the dsPIC30F Speech Recognition Library. The following Microchip documents are available and recommended as supplemental reference resources.

Readme for dsPIC30F Speech Recognition Library
For the latest information on using the dsPIC30F Speech Recognition Library, read the "Readme for dsPIC30F Speech Recognition Library.txt" file (an ASCII text file) in the doc folder of the installation directory. The Readme file contains update information and known issues that may not be included in this user's guide.

dsPIC30F Speech Recognition Word Library Builder Manual (DS70137)
Use this manual in conjunction with the dsPIC30F Speech Recognition Library User's Guide for creating application word libraries.

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 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 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

This document helps you use Microchip’s MPLAB C30 C compiler for dsPIC devices to develop your 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.

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, contact the nearest Microchip sales location (see back page) or 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 (FAQ), 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 when 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 C17, 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 and MPLAB ICE 4000.

• **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 and MPLAB SIM30 simulators, 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 development programmer.
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

In addition, there is a Development Systems Information Line which lists the latest versions of Microchip's development systems software products. This line also provides information on how customers can receive currently available upgrade kits.

The Development Systems Information Line numbers are:

1-800-755-2345 – United States and most of Canada
1-480-792-7302 – Other International Locations
Chapter 1. Introduction

1.1 INTRODUCTION

The dsPIC30F Speech Recognition Library is a software development tool that allows the incorporation of speech activation into product applications running on the Microchip dsPIC30F family of 16-bit digital signal controllers. Implemented as a C- or assembly-callable library, the dsPIC30F Speech Recognition Library helps create the voice-controlled user interface. Items discussed in this chapter are:

• System Requirements
• Software Components
• Functional Features
• References

1.2 SYSTEM REQUIREMENTS

The Speech Recognition Library requires the following minimum dsPIC30F resources:

• 2.8 Kbytes on-chip data RAM (Data Memory)
• 3.5 Kwords on-chip Flash memory (Program Memory) for library code
• 8 Kbytes Flash memory (on-chip or external) for Vector Codebook (VCB) table
• 1.5 Kbytes Flash memory (on-chip or external) for Hidden Markov Model (HMM) tables (one per word in library)
• 8 MIPS of processing capability if VCB and HMM-data are completely stored in Program memory (PM)
• Data Converter Interface (DCI) for interfacing with Silicon Labs Si-3000 audio codec for providing an input speech signal from connected microphone (12 KHz sampling frequency)

Additionally, an external microphone with a uniform frequency response not lower than 6 KHz is required.
1.3 SOFTWARE COMPONENTS

The Speech Recognition Library is distributed on CD with the files shown in Figure 1-1:

FIGURE 1-1: SPEECH RECOGNITION LIBRARY CD CONTENTS

The files in the gray area are used with the Speech Recognition Library. These files are installed by running SRLinst.exe (see Section 2.3 “Installation Procedure”).

Note: The Speech Recognition Library installation defaults to the PIC30 tools directory, which allows the files to work properly with the PIC30 tools.
Table 1-1 describes how these Speech Recognition Library files are used.

**TABLE 1-1: SPEECH RECOGNITION LIBRARY FILE FUNCTIONS**

<table>
<thead>
<tr>
<th>Folder</th>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>lib</td>
<td>SR_Lib.a</td>
<td>Speech Recognition Library (SRL) application file</td>
</tr>
<tr>
<td>support/h</td>
<td>SR_Lib.h</td>
<td>Header support file that defines function prototypes and data interface structure needed for integrating SRL with the user application.</td>
</tr>
<tr>
<td>examples/</td>
<td>ASR_Demo</td>
<td>Sample application files used to demonstrate the use of the Speech Recognition Library (see Section 6.4 “Demonstration Application”)</td>
</tr>
<tr>
<td>ASR_Demo</td>
<td>ASR_Demo.hex</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SR_demo.c</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SR_dem_Utils.s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Uninst.isu</td>
<td>Uninstall file for Speech Recognition Library.</td>
</tr>
<tr>
<td>docs</td>
<td>70140a.pdf</td>
<td>User’s Guide</td>
</tr>
<tr>
<td></td>
<td>Readme.txt</td>
<td>Readme file</td>
</tr>
</tbody>
</table>

In addition to the dsPIC30F Speech Recognition Library files, the CD also contains the installation file for the Speech Recognition Word Library Builder, which is the PC utility used to create the necessary supplemental files that are needed to use the dsPIC30F Speech Recognition Library with an application. After installing this program on a PC, these supplemental support files must be created for the custom library (called ‘filename’):

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[filename]_int.s</td>
<td>Data allocated to internal memory</td>
</tr>
<tr>
<td>[filename]_ext.s</td>
<td>Data allocated to external memory</td>
</tr>
<tr>
<td>[filename].h</td>
<td>Contains a set of constants for each word in the word library</td>
</tr>
</tbody>
</table>

See Section 5.2 “Incorporating Speech Recognition in an Application” for details on integrating the speech recognition files into the application code.

**1.4 FUNCTIONAL FEATURES**

The dsPIC30F Speech Recognition Library provides state-of-the-art, word-based speech recognition that uses a predetermined vocabulary. Functionality includes:

- Speaker-independent recognition of isolated words
- Recognition time of 0.5 sec. or less
- Hidden Markov Model (HMM) based recognition algorithm
- Optional activation by user-specified keyword
- Support for a Word Library of 1 to 100 words
- Efficient recognition in noisy conditions:
  - white noise
  - office noise
  - passenger compartment noise
  - any mixture with SNR not worse than 15 dB
- U.S. English language support
- Small memory footprint
- Supports placement of HMM data in both on-chip and off-chip (external) memory
- Optional system self-test using a predefined keyword
1.5 REFERENCES

The following references are suggested for background information on the use of speech recognition.


Chapter 2.  Installation

2.1 INTRODUCTION

This chapter provides instructions for installing the dsPIC30F Speech Recognition Library on your PC, where it can be used with the MPLAB IDE to incorporate speech recognition in an application program running on a dsPIC30F Digital Signal Controller. Items discussed in this chapter are:

- System Requirements
- Installation Procedure
- Uninstall Procedure

2.2 SYSTEM REQUIREMENTS

The dsPIC30F Speech Recognition Library requires the following resources:

- Disk Space: 1.7 MB minimum free space

2.3 INSTALLATION PROCEDURE

The dsPIC30F Speech Recognition Library is packaged on a CD. To install the library follow these steps:

1. Insert the CD into the appropriate drive. When the Speech Recognition Library installation screen displays, as shown in Figure 2-1, click on SR Library Setup to start the setup procedure.

FIGURE 2-1: SPEECH RECOGNITION TOOLS INSTALLATION SCREEN
2. Click **Next** on the Welcome dialog (Figure 2-2) to begin installation.

**FIGURE 2-2: INSTALLATION WELCOME SCREEN**

![Installation Welcome Screen](image)

Click **Next** to begin the installation process.

3. Review and accept the License Information (see Figure 2-3).

**FIGURE 2-3: LICENSING INFORMATION DIALOG**

![License Agreement Dialog](image)

Review license agreement and click **Yes** to accept.
4. Specify where to install the library files and the folder for the program icons as shown in Figure 2-4.

**FIGURE 2-4: FILE LOCATION AND SETUP DIALOGS**

By default the Speech Recognition Library files are installed in:

```
"PATH"\PIC30_Tools\lib
```

where "PATH" represents the system folder where the MPLAB C30 compiler is installed.

If the MPLAB C30 compiler is not installed yet, the files will be placed in directory:

```
C:\Program Files\MPLAB IDE\SR
```
Installation progress displays briefly as the files are installed. When the files are installed the Setup Complete dialog displays, as shown in Figure 2-5.

**FIGURE 2-5: SETUP COMPLETE DIALOG**

5. Click **Finish**.

### 2.4 UNINSTALL PROCEDURE

To uninstall the Speech Recognition Library:
1. From Start select **Settings > Control Panel**.
2. From the Control Panel window, select **Add/Remove Programs**.
3. Select **Microchip Speech Recognition Library**.
4. Click **Change/Remove**.
5. Follow the Windows uninstall procedures.
Chapter 3. Overview

3.1 INTRODUCTION

The dsPIC30F Speech Recognition Library allows the incorporation of speech recognition in an embedded application program running on a dsPIC30F device. A predefined list of words controls the application with only a modest amount of RAM and program memory. The word list is created with the dsPIC30F Speech Recognition Library Word Library Builder. The Word Library Builder is a PC-based program that lets users select and implement the user interface vocabulary. The Word Library Builder provides the supplemental files needed by the user application to define the allowable words, the Library Keyword, the word recognition features for the Vector Codebook and the Hidden Markov Model (HMM) data files for each library word. For more information on the Word Library Builder, see the dsPIC30F Speech Recognition Word Library Builder User’s Guide (DS70137). Items discussed in this chapter are:

- Overview of the dsPIC30F Speech Recognition Library
- Speech Recognition Process Flow
- Word Library

3.2 OVERVIEW OF THE dsPIC30F SPEECH RECOGNITION LIBRARY

The dsPIC30F Speech Recognition Library provides an audio interface to a user’s application program, allowing the user to control the application by uttering discrete words that are contained in a predefined word library. The words chosen for the library are specifically relevant to the interaction between the application program and the user. Upon recognition of a word, the application program takes an appropriate action, as shown in Figure 3-1.

FIGURE 3-1: OVERVIEW OF SPEECH RECOGNITION

[Diagram showing audio interface, dsPIC30F, Application Board, HMM Library, Vector Codebook, Keyword, and Controlled Device]
The dsPIC30F Speech Recognition Library uses a recognition algorithm based on discrete Hidden Markov Model (HMM) of words (one HMM model for each word in an application word library). A word spoken through a microphone connected to the dsPIC30F application board is analyzed on a frame-by-frame basis using RASTA-PLP-algorithm and quantized into feature vectors of sound characteristics against a vector codebook. The quantized feature vectors are then examined to determine what word HMM model they most closely match.

The dsPIC30F Speech Recognition Library operates in both clear (free of noise) and noisy conditions. Background interference can include white noise, office noise and passenger compartment noise components in any mixture. Total signal-to-noise ratio (SNR) should be no less than 15 dB. A signal level above the noise threshold is presumed to be an incoming word.

The dsPIC30F Speech Recognition Library can operate with a word library of up to 100 words. The word library is built around a keyword that is readily interpreted. Depending on the operating mode used, this keyword can be used to self-test the library and to trigger a recognition session.

Successful recognition requires the words to be separated by a pause of at least one-half second but less than some specified period (normally programmed for five seconds). After a pause that times out, a new recognition session must be started.

Optionally, the operating mode can be set to disable self-testing and/or keyword activation. When keyword activation is disabled, there is no timeout. Words must only be separated by at least 500 milliseconds.

3.3 SPEECH RECOGNITION PROCESS FLOW

Figure 3-2 is a simplified flow diagram of the dsPIC30F Speech Recognition Library operation. When the application board is powered up, the library is initialized, which also establishes the operating mode. The operating mode determines if a self-test is run and if the keyword is used to activate a recognition session. The self-test processes a sample utterance (stored in memory) of the library keyword. If the self-test fails, the diagnostic routine returns an error code indicating that appropriate remedial action must be taken. For example, it may be necessary to verify the build of the application. If the self-test is successful, the library estimates a baseline noise level and enters either the Keyword Search mode or the Listen mode, depending on whether keyword activation is enabled.

If keyword activation is enabled, the library remains in the Keyword Search mode until the user articulates the keyword. In this mode, the keyword must be recognized before any further speech recognition can take place. This functionality is important in applications where user security is an issue. For example, a user might be required to articulate a specific word such as “security” followed by a series of numbers to gain access to something. Once the keyword is recognized, the library enters the Listen mode. In this mode the library must receive a new word within the specified Listen Window timeout period (normally about 5 seconds). If the timeout period elapses without a new word, and keyword activation is enabled, the library reverts to the Keyword Search mode. The user must then rearticulate the keyword to start a new recognition session.

If keyword activation is not enabled, the library immediately enters the Listen mode. In this mode, the library attempts to recognize any incoming word. Upon successful recognition of the first word, the library notifies the user application and listens for the next word.
FIGURE 3-2: OVERVIEW FLOW DIAGRAM

Apply Power
  Initialize Operating Mode
    Self-Test Enabled?
      Yes
        Run Self-Test
          Self-Test OK?
            Yes
              Keyword Activation Enabled?
                Yes
                  Keyword Search Mode
                    Yes
                      Keyword Processed?
                        Yes
                          Pass Keyword to Application
                        No
                          Listen Mode
                            Listen Window Timeout?
                              Yes
                                Correct Problem
                              No
                                New Word?
                                  Yes
                                    “Unrecognized” Response
                                  No
                                    Word in Library?
                                      Yes
                                        Pass Word to Application
                                      No
                                        “Unrecognized” Response
                            No
                              “Unrecognized” Response
                        No
                          Listen Window Timeout?
                            Yes
                              Correct Problem
                            No
                              “Unrecognized” Response
              No
                No
                  Correct Problem
                Yes
                  Correct Problem
      No
        Self-Test Enabled?
          Yes
            Pass Keyword to Application
          No
            Run Self-Test
              Self-Test OK?
                Yes
                  Keyword Activation Enabled?
                    Yes
                      Keyword Search Mode
                        Yes
                          Keyword Processed?
                            Yes
                              Pass Keyword to Application
                            No
                              Listen Mode
                                Listen Window Timeout?
                                  Yes
                                    Correct Problem
                                  No
                                    New Word?
                                      Yes
                                        “Unrecognized” Response
                                      No
                                        Word in Library?
                                          Yes
                                            Pass Word to Application
                                          No
                                            “Unrecognized” Response
                                No
                                  “Unrecognized” Response
                            No
                              “Unrecognized” Response
                        No
                          Listen Window Timeout?
                            Yes
                              Correct Problem
                            No
                              “Unrecognized” Response
3.4 WORD LIBRARY

The list of words identified by the dsPIC30F Speech Recognition Library is called the word library. The words in the word library are user selectable, but they must come from the list of words that form the Master Word Library. The Master Word Library consists of 100 common words that may be used to add command sequences for an application. For example, a security panel application may use a library similar to that shown in Table 3-1.

### TABLE 3-1: REPRESENTATIVE WORD LIBRARY

<table>
<thead>
<tr>
<th>Library Word</th>
<th>Application Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>security</td>
<td>Keyword – activates recognition PIN entry</td>
</tr>
<tr>
<td>on</td>
<td>Enable security or camera</td>
</tr>
<tr>
<td>off</td>
<td>Disables security</td>
</tr>
<tr>
<td>camera</td>
<td>Select camera (for on or off)</td>
</tr>
<tr>
<td>zone</td>
<td>Select zone number</td>
</tr>
<tr>
<td>zero</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>one</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>two</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>three</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>four</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>five</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>six</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>seven</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>eight</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
<tr>
<td>nine</td>
<td>Digit for security PIN code, camera or zone selection</td>
</tr>
</tbody>
</table>

The Master Word Library is shown in Appendix A.

The word library is set up to contain only words that are relevant to an embedded user application. Select the unique words needed for an application using the dsPIC30F Speech Recognition Library Word Library Builder. Word library building is a required step to ensure that speech recognition is effective. The building process generates several data files specific to the word library. These files must be linked into the application program. The files generated by word library building are defined in Chapter 5. “Incorporating Speech Recognition”.

In all word libraries, one word must be designated as the application “keyword”. The keyword is a special word that is used for the optional self-test and as the activation word when the optional keyword activation feature is enabled. With keyword activation, the speech recognition library listens only for the keyword until it recognizes the keyword. Then it listens for other words in the word library.

Keyword activation is a safety measure that can be used to prevent the user application from accidentally responding to words when it should not. To enable keyword activation, the function `SR_Initialization()` must be called with the `SR_mode` parameter set to ‘0’ or ‘1’ (see Section 4.5.8 “SR_mode Parameter (Int)”.)
Chapter 4. Application Program Interface

4.1 INTRODUCTION

The dsPIC30F Speech Recognition Library integrates with the user application running on the dsPIC30F device to provide voice control of the application. This chapter provides information needed to interface the dsPIC30F Speech Recognition Library with an application. Items discussed in this chapter are:

- Requirements
- Data Structures
- Library Functions
- SR_Initialization( ) Function
- DCIInterrupt( ) Function
- SR_Recognizer( ) Function
- T4Interrupt( ) Function
- Using External Memory
- Required dsPIC30F System Resources
- Recognition Accuracy

4.2 REQUIREMENTS

4.2.1 System Frequency Requirements

The dsPIC30F Speech Recognition Library requires that speech be sampled at a fixed rate of 12.0 kHz. Since sampling is performed by an external codec which interfaces with the dsPIC30F via its Digital Converter Interface (DCI) module, a limited number of system frequencies can be used by your application. To accommodate the 12.0 kHz sampling rate, the dsPIC30F processor can execute only at multiples of 6.144 MHz that are at least 12.288 MHz. This means that allowable execution speeds for the Speech Recognition Library are 12.288 MHz, 18.432 MHz and 24.576 MHz. To accommodate these system frequencies, operate the dsPIC30F using only the clock and PLL combinations shown in Table 4-1. These clock/PLL combinations are also compatible for common UART communication rates.

**TABLE 4-1: CLOCK AND PLL COMBINATIONS FOR 12 kHz SAMPLING RATE**

<table>
<thead>
<tr>
<th>Processor Frequency (MIPS)</th>
<th>Clock Frequency (MHz)</th>
<th>PLL Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.288</td>
<td>6.144</td>
<td>x8</td>
</tr>
<tr>
<td>18.432</td>
<td>4.608</td>
<td>x16</td>
</tr>
<tr>
<td>18.432</td>
<td>9.216</td>
<td>x8</td>
</tr>
<tr>
<td>24.576</td>
<td>6.144</td>
<td>x16</td>
</tr>
</tbody>
</table>
4.2.2 Memory Requirements:

The dsPIC30F Speech Recognition Library requires the allocation of memory as shown in Table 4-2:

**TABLE 4-2: SPEECH RECOGNITION LIBRARY MEMORY REQUIREMENTS**

<table>
<thead>
<tr>
<th>Memory Amount</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 Kwords</td>
<td>Library code (program memory)</td>
</tr>
<tr>
<td>8 Kbytes</td>
<td>Vector codebook (program or external memory)*</td>
</tr>
<tr>
<td>1.5 Kbytes each</td>
<td>Hidden Markov Model (per word) (program or external memory)</td>
</tr>
<tr>
<td>15-20 Kbytes</td>
<td>Self-test keyword utterances (program or external memory)</td>
</tr>
<tr>
<td>1.90 Kbytes</td>
<td>X RAM</td>
</tr>
<tr>
<td>0.83 Kbytes</td>
<td>Y RAM</td>
</tr>
</tbody>
</table>

* Although the vector codebook may be stored in external memory, it is highly recommended that this table reside in program memory.

4.2.3 Performance Requirements

There are two main computational components of the speech recognition library: feature vector analysis and word recognition. Feature vector analysis is performed in real-time while speech is being sampled and requires the most processing demand. After a word ending has been detected, speech sampling is temporarily halted and word recognition begins. Although word recognition is not considered a real-time activity, it still requires significant processing capability. The specific computational power required by the Speech Recognition Library depends on how the memory requirements are implemented. Table 4-3 shows the expected performance of the dsPIC30F relative to memory allocation.

**TABLE 4-3: dsPIC30F PERFORMANCE CAPABILITIES FOR SPEECH RECOGNITION LIBRARY**

<table>
<thead>
<tr>
<th>Performance</th>
<th>Memory Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 MIPS</td>
<td>VCB and HMM data are stored in program memory</td>
</tr>
<tr>
<td>9 MIPS</td>
<td>VCB stored in program memory while HMM data are completely or partially stored in external memory</td>
</tr>
<tr>
<td>13 MIPS</td>
<td>Both VCB and HMM data are stored in external memory*</td>
</tr>
</tbody>
</table>

* Assumes external memory on 16-bit parallel port.

4.2.4 External Hardware Resources

The dsPIC30F Speech Recognition Library requires the external hardware listed in Table 4-4.

**TABLE 4-4: EXTERNAL HARDWARE REQUIREMENTS**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio-codec</td>
<td>Si3000 (Silicon Laboratory)</td>
</tr>
<tr>
<td>External Memory (optional)</td>
<td>128K x 16-bit Am29F200BT-55EC (AMD) memory module</td>
</tr>
<tr>
<td>External Microphone</td>
<td>Linear frequency response up to 6 kHz (minimum)</td>
</tr>
</tbody>
</table>

4.2.5 Software Requirements:

- MPLAB IDE V6.53 or higher
- MPLAB C30 Compiler V1.20 or higher
4.3 DATA STRUCTURES

The dsPIC30F Speech Recognition Library interfaces with an application using two data structures. One provides status information and control capability of the recognition process. The other provides information about the acoustical environment.

4.3.1 SR_STATUSbits Structure

The primary interface for the user program is the SR_STATUSbits structure defined in SR_LIB.H (see Figure 4-1). This structure is used by SR_Recognizer() to complete the recognition tasks, to enable the user to obtain the state of the library and to correctly and safely disable the recognition process when required. A good understanding of the SR_STATUSbits structure is required for integrating the dsPIC30F Speech Recognition Library with an application.

**FIGURE 4-1: SR_STATUSbits STRUCTURE DEFINITION:**

```
// SR status structure
struct {
  unsigned buffer :1; // filling up sample buffer flag
  unsigned state :1; // recognizer state flag
  unsigned pause :1; // pause between words flag
  unsigned enable :1; // SR enable flag (0 disable, 1 enable)
  unsigned clipping :1; // input signal clipping flag
  unsigned idle :1; // waiting for filling up input buffer flag
  unsigned block :1; // frame analysis block flag
} SR_STATUSbits __attribute__((__near__));
```

To access the data structure, which consists of the seven status flags shown, use the C syntax:

```
SR_STATUSbits.[field_name]
```

where [field_name] is one of the flag names (Buffer, State, Pause, Enable, Clipping, Idle or Block).

Enable and Block are the only flags that can be written to. All other flags are read only.

4.3.1.1 BUFFER FLAG (Read Only)

The Buffer flag is controlled by the DCI interrupt handler to indicate the status of the input buffer. The input buffer accumulates input speech samples from the DCI interrupt. This flag has the following values:

1. Input buffer is full of speech samples that are ready for feature vector analysis. When this flag becomes ‘1’, control must be returned to the Speech Recognition Library within 330 microseconds
2. Input buffer is being filled with speech samples from the incoming word. Contents of the previous frame are being analyzed in the work buffer to establish a feature vector

After the input buffer is filled with one sound frame (84 DCI samples of the incoming word), the Buffer flag is set to ‘1’. After 330 microseconds, the DCI interrupt handler resets the Buffer flag to ‘0’ and begins writing the next set of samples into the input buffer.
4.3.1.2 **STATE FLAG** (Read Only)

The **State** flag indicates the current operating state of the Speech Recognition Library. This flag has the following values:

1. Speech Recognition Library is waiting for or is processing the keyword.
2. Speech Recognition Library is waiting for or is processing any word in the word library.

4.3.1.3 **PAUSE FLAG** (Read Only)

The **Pause** flag indicates if the dsPIC30F Speech Recognition Library is between identifying adjacent words. This flag has the following values:

1. Pause detected between words. In this state, the application program has up to a 100 msec window in which it can safely interrupt the Speech Recognition Library. See Section 5.2.2 “Word Recognition” for more information.
2. Incoming signal differs from the noise baseline. In this state, the user should only interrupt the Speech Recognition Library as described in Section 5.2.1 “Feature Vector Analysis”.

4.3.1.4 **ENABLE FLAG** (Read/Write)

The **Enable** flag allows the user to disable speech recognition and divert all of the dsPIC30F device resources to the user application program. This flag has the following values:

1. Speech Recognition Library is enabled (default value after **SR_Initilization()** is called).
2. Speech Recognition Library is disabled.

**Note:** To avoid any loss of data, the Speech Recognition Library should be disabled only when **Pause flag** = 1. The library can be re-enabled without calling **SR_Initilization()** again by setting the **Enable** flag to '1'.

4.3.1.5 **CLIPPING FLAG** (Read Only)

The **Clipping** flag indicates that the speech signal has exceeded an admissible loudness level for the current word. This fact must be considered as a warning that the recognition result may be incorrect. The **Clipping** flag has the following values:

1. Input speech signal has exceeded the admissible loudness level.
2. Input speech signal has not been clipped.

The **Clipping** flag is cleared at the end of the word recognition process.
4.3.1.6 **IDLE FLAG (Read Only)**

The Idle flag indicates the current frame operating state. This flag has the following values:

1. Analysis of the current frame is completed. The library is waiting for another 7 msec frame to process.
2. Analysis of the current frame is in process.

This flag is reset automatically when the input signal buffer is filled and its contents has been copied to the work buffer. It is set to '1' when the work buffer has been processed. After it is set to '1' the user application can take ownership of the CPU until the Buffer flag is set to '1'.

4.3.1.7 **BLOCK FLAG (Read/Write)**

The Block flag is used to set how `SR_Recognizer()` processes a frame of data.

1. `SR_Recognizer()` will analyze the current frame of data in a single function call.
2. `SR_Recognizer()` will analyze the current frame of data using multiple function calls.

By default, `SR_Recognizer()` processes a frame of data using multiple function calls (see Section 5.2.1 “Feature Vector Analysis”). This allows the application to perform other real-time tasks (like operate an RTOS) while still performing speech recognition. If the application program does not require processor CPU time while the frame is being analyzed, the Block flag can be set to '1', allowing `SR_Recognizer()` to process the frame slightly faster.

| Note 1: | The Block flag is cleared by `SR_Initialization()` |
| Note 2: | The Block flag can be set or reset at any time, but a new value of this flag becomes valid only when the next speech frame analysis starts. |

4.3.2 **SR_SNRLlevel Structure**

A secondary interface for the user program is the `SR_SNRLlevel` structure defined in `SR_LIB.H` (see Figure 4-2). This structure provides recent signal and noise level information.

**FIGURE 4-2: **`SR_SNRLlevel` STRUCTURE DEFINITION

```c
// Noise, Speech level and SNR structure
struct {
    unsigned noise;     // background noise energy level
    unsigned speech;    // current speech energy level
    unsigned snr;       // current Signal-to-Noise ratio
} SR_SNRLlevel __attribute__((__near__));
```

The `noise` variable defines the background noise level in decibel (dB), where the `speech` variable defines the most recent processed speech signal level in dB. The Speech-to-Noise Ratio variable (`snr`) reflects the calculated SNR for the recognized word (also in dB). The SNR level is determined as the difference between the speech level and the noise level.

To retrieve the values of these variables, use the C syntax:

```c
SR_SNRLlevel.[field_name]
```

where `[field_name]` is one of the variable names: noise, speech or snr.

This structure is read only and can be used to provide feedback to users of the system (e.g., “speak louder” or “speak softer”).
4.4 LIBRARY FUNCTIONS

The dsPIC30F Speech Recognition Library functions are implemented in one archive file (SR_LIB.A) and one header file (SR_LIB.H). The functions in the library adhere to the Microchip C30 compiler function-calling convention. The library consists of two callable functions and two interrupt service routines.

**TABLE 4-5: dsPIC30F SPEECH RECOGNITION LIBRARY FUNCTIONS**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR_Initialization()</td>
<td>Initializes speech recognition and sets the operating mode. This function must be called before calling SR_Recognizer().</td>
</tr>
<tr>
<td>SR_Recognizer()</td>
<td>Performs the speech recognition process.</td>
</tr>
<tr>
<td>_DCIInterrupt()</td>
<td>Moves accumulated speech samples to the input buffer. Occurs every 1/3000 seconds.</td>
</tr>
<tr>
<td>_T4Interrupt()</td>
<td>Measures the listen window from the end of the last spoken word and triggers a return to keyword search mode (if that feature is used).</td>
</tr>
</tbody>
</table>

The user interface is provided by the header file (SR_LIB.H). This file includes the function prototypes for SR_Initialization() and SR_Recognizer(), as shown in Figure 4-3.

**FIGURE 4-3: FUNCTION PROTOTYPES OF SR_LIB.H FILE**

```c
// Initialization procedure
int SR_Initialization (unsigned int fclock, int pll_setting,
                        int timeout, int mic_gain, int SR_mode);

// Parameters:
// fclock      - frequency of an external quartz resonator in kHZ (16-bit integer)
// pll_setting - preset PLL mode of dsPIC chip (16-bit integer)
// timeout     - desired timeout period in seconds (16-bit integer)
// mic_gain    - microphone gain level in Silicon Labs Si3000 codec (16-bit integer)
// SR_mode     - mode of SR operation (16-bit integer)

// Returns:
//    0   - if self-test was completed successfully
//    1   - otherwise.

// Recognition procedure
int SR_Recognizer (void);

// Parameters: none

// Returns:
//    i   the index of recognized word defined in library header file
//        if word recognition was successfully completed
//    0   if recognition not completed
```
4.5 **SR_Initialization() FUNCTION**

4.5.1 **Overview**

The **SR_Initialization()** function performs the following library tasks:

- Initializes the global variables and arrays of the Speech Recognition Library.
- Initializes the DCI module for multi-channel mode with dsPIC30F as master.
- Initializes the Si3000 audio codec.
- Initializes the 16-bit pause timer implemented on Timer4.
- Executes the self-test, if self-test is enabled. Self-test mode is enabled by setting the **SR_mode** parameter to ‘0’ or ‘2’ (see Section 4.5.8 “**SR_mode Parameter (Int)**”).

4.5.2 **Return Value**

The values returned by the **SR_Initialization()** function are:

- 0  Self-test completed successfully or not executed
- 1  Self-test unsuccessful

4.5.3 **Parameters**

The **SR_Initialization()** function has five parameters, as shown in Table 4-6.

### TABLE 4-6: **SR_INITIALIZATION( ) FUNCTION PARAMETERS:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Size</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>fclock</td>
<td>int</td>
<td>Frequency of external clock oscillator in kHz</td>
</tr>
<tr>
<td>pll_setting</td>
<td>int</td>
<td>Clock oscillator PLL setting</td>
</tr>
<tr>
<td>timeout</td>
<td>int</td>
<td>Desired timeout period in seconds</td>
</tr>
<tr>
<td>mic_gain</td>
<td>int</td>
<td>Microphone gain level in Si3000 codec</td>
</tr>
<tr>
<td>SR_mode</td>
<td>int</td>
<td>Sets operating mode (see Section 4.5.8 “<strong>SR_mode Parameter (Int)</strong>”)</td>
</tr>
</tbody>
</table>

4.5.4 **fclock Parameter (Int)**

The **fclock** parameter is set to the frequency of the dsPIC external clock oscillator. This parameter is set in kHz. Together with the next parameter of **SR_Initialization()** function, **pll_setting, fclock** provides the correct sampling rate of the DCI module of dsPIC30F and Si3000 codec. The admissible pairs of these parameters that provide the required sampling frequency are shown in the Table 4-1.

4.5.5 **pll_setting Parameter (Int)**

The **pll_setting** parameter is set to the PLL clock setting of the dsPIC30F. Together with the first parameter of **SR_Initialization()** function, **fclock, pll_setting** provides the correct sampling rate of the Si3000 codec. The admissible pairs of these parameters that provide the required sampling frequency are shown in the Table 4-1.

4.5.6 **timeout Parameter (Int)**

The **timeout** parameter is set to the desired length (in seconds) of the Listen Window. The Listen Window timer is used only when keyword activation is enabled (**SR_Mode** = 0 or 1). It controls the length of time the library will recognize any word after the last word has been recognized successfully (see Section 3.3 “Speech Recognition Process Flow”).
The Listen Window is implemented using Timer 4 with a software counter in the Timer 4 Interrupt Service Routine. The maximum allowed time out is based on processor speed, as shown in Table 4-7.

**TABLE 4-7: POSSIBLE COMBINATIONS OF dsPIC30F PARAMETERS PROVIDING 12 kHz SAMPLING FREQUENCY IN AUDIO CODEC Si3000**

<table>
<thead>
<tr>
<th>Processor Speed</th>
<th>Maximum Time Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.288</td>
<td>21 sec</td>
</tr>
<tr>
<td>18.432</td>
<td>14 sec</td>
</tr>
<tr>
<td>24.576</td>
<td>10 sec</td>
</tr>
</tbody>
</table>

4.5.7 **mic_gain Parameter (Int)**

The `mic_gain` parameter defines a level (in dB) of microphone gain in the Si3000 audio codec. It takes a value of 0, 10, 20 or 30, which represents the microphone gain in decibels. This level depends on the sensitivity of the microphone used. The more sensitive the microphone, the lower the gain setting needed. The best value for this parameter depends on the microphone sensitivity and should be determined by experimentation.

4.5.8 **SR_mode Parameter (Int)**

The `SR_mode` parameter defines the operation mode of the dsPIC30F Speech Recognition Library. The library supports four modes of operation:

- **0 Self-test enabled, keyword activation enabled.** In this mode, self-test runs immediately after `SR_Initialization()` is called. The keyword must be processed before further speech recognition is activated in Listen mode. Keyword recognition is also required after each time-out pause between spoken words.

- **1 Self-test disabled, keyword activation enabled.** In this mode, the self-test is not run after `SR_Initialization()` is called. The keyword must be processed before further speech recognition is activated in Listen mode. Keyword recognition is also required after each time-out pause between spoken words.

- **2 Self-test enabled, keyword activation disabled.** In this mode, self-test runs immediately after `SR_Initialization()` is called. Speech recognition is activated in Listen mode, waiting to recognize any word in the word library.

- **3 Self-test disabled, keyword activation disabled.** In this mode, the self-test is not run after `SR_Initialization()` is called. Speech recognition is activated to Listen mode, waiting to recognize any word in the word library.

**Note:** In all operating modes the user can hear the audio signal supplied by the microphone using the SPKR OUT audio jack. In modes 2 or 3, the `timeout` parameter is not used and the user regains the Timer4 resource.
4.5.9 Resources

Table 4-8 lists the system resources used by the `SR_Initialization()` function.

<table>
<thead>
<tr>
<th>RESOURCE</th>
<th>USAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0-W7</td>
<td>Used, not restored</td>
</tr>
<tr>
<td>W8-W13</td>
<td>Used, restored</td>
</tr>
<tr>
<td>ACCA</td>
<td>Used, not restored</td>
</tr>
<tr>
<td>ACCB</td>
<td>Used, not restored</td>
</tr>
<tr>
<td>CORCON</td>
<td>Used, restored</td>
</tr>
<tr>
<td>TBLPAG</td>
<td>Used, restored</td>
</tr>
<tr>
<td>DO</td>
<td>Used, two (2) levels</td>
</tr>
<tr>
<td>REPEAT</td>
<td>Used</td>
</tr>
</tbody>
</table>

4.5.10 Self-Test

A self-test function is provided to check the operational capability of the dsPICDEM™ 1.1 GPD board (dsPIC30F and Si3000 audio codec). The self test is enabled when `SR_Mode` is set to ‘0’ or ‘2’. It is based on the following sequence of operations:

- The dsPIC30F reads out stored samples of a keyword utterance and transmits them through the Data Converter Interface (DCI) module to the Digital-to-Analog Converter (DAC) of the Si3000.
- The Si3000 codec is placed in loop-back mode and returns the received samples through its ADC back to the dsPIC30F.
- The dsPIC30F Speech Recognition Library then processes the stored keyword as it is received via the DCI.

Successful recognition proves the operational capability of the dsPICDEM 1.1 development board with the installed dsPIC30F Speech Recognition Library application. If the self-test fails, check the connections of the dsPIC30F device and Si3000 codec. Also check the build of the application to verify that the stored keyword data is properly built into the application.
4.6 DCIInterrupt() FUNCTION

4.6.1 Overview

The DCI interrupt handler is used to receive speech data from the Si3000 codec and buffer the data into a frame of 84 samples for processing by SR_Recognizer(). The DCI interrupt handler also provides audio feedback for the system. All data received by the dsPIC30F is sent back to the Si3000 codec for playback.

4.6.2 Return Value

None

4.6.3 Parameters

None

4.6.4 Resources

Table 4-9 lists the system resources used by the DCIInterrupt() function.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registers</td>
<td></td>
</tr>
<tr>
<td>W0-W3</td>
<td>Used, restored</td>
</tr>
<tr>
<td>Instructions</td>
<td></td>
</tr>
<tr>
<td>DO instructions</td>
<td>Not used</td>
</tr>
<tr>
<td>REPEAT instructions</td>
<td>Not used</td>
</tr>
</tbody>
</table>

4.6.5 Detailed Operation

The dsPIC30F DCI converts the voice input data from the audio codec into discrete digital samples. The DCI interrupt handler captures the samples and loads them into the input buffer, four samples at a time, until 84 samples are received. The 84 sample buffer represents one 7 msec segment of the word (one sound frame). The period of this interrupt is approximately 330 microseconds.

When the input buffer is full, the DCI Interrupt handler sets the SR_STATUSbits buffer flag to '1', which tells SR_Recognizer() that the input buffer must be copied to the work buffer and processed. On the very next DCI Interrupt after setting the buffer flag to '1', the buffer flag is set to '0' and the input buffer is overwritten with new DCI samples.

The DCI interrupt handler is critical for correct operation of an application. The DCI interrupt handler takes less than 50 cycles to execute. If this interrupt is not serviced every 1/3000 second, input speech samples will be lost and word recognition will be compromised. SR_Initialization() configures the DCI interrupt with the default priority level 4. This priority level is not changed by the library, so the DCI interrupt priority level can be modified after SR_Initialization() is called. It is recommended that the dsPIC30F be configured to run with nested interrupts with the DCI interrupt having highest priority. Table 4-10 shows how long it takes to execute the DCI interrupt handler at the possible operating frequencies.

<table>
<thead>
<tr>
<th>Performance (P)</th>
<th>DCI Interrupt Handler Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.288 MIPS</td>
<td>4.1 µsec</td>
</tr>
<tr>
<td>18.432 MIPS</td>
<td>2.7 µsec</td>
</tr>
<tr>
<td>24.576 MIPS</td>
<td>2.0 µsec</td>
</tr>
</tbody>
</table>
4.7  

**SR_Recognizer( ) FUNCTION**

### 4.7.1  Overview

The `SR_Recognizer()` function performs the actual step-by-step analysis of the incoming sound frames and matching of the resulting frame vectors to the HMM library. When the input buffer is full, its contents are transferred to the work buffer, where that sound frame is analyzed. Each sound frame analysis results in a feature vector based on a “best-fit” comparison of vector codebook properties. This process repeats until the end of word is detected. At that point the quantized feature vectors of the analyzed frames are mapped to each HMM word model to find the closest word match.

If keyword activation is enabled, `SR_Recognizer()` only maps the incoming word to the HMM of the designated keyword until a match is recognized. Once the keyword is recognized, `SR_Recognizer()` then attempts to match each subsequent incoming word to all HMM words in the library until the Listen Window, set by the `SR_Initialization()` function, expires.

If keyword activation is disabled, `SR_Recognizer()` immediately tries to recognize all incoming words and ignores the Listen Window set in `SR_Initialization()`.

### 4.7.2  Return Value

`SR_Recognizer()` returns an integer representing an enumerated value to the application program.

The word library is defined in the library header file (`library_name.h`) and contains an enumeration declaration of all words in the library, as shown in Figure 4-4. If recognition is successful, the return value is the enumerated response that corresponds to the matched word in the library. For example, if the feature vectors of the incoming word most closely match the word “camera”, the enumerated response is `SRL_CAMERA`, which returns the integer ‘3’. If recognition is unsuccessful, the return value is `SRL_UNKNOWN`. If recognition is not complete, the return value is `SRL_NONE`.

The enumeration definitions begin with `SRL_NONE` and advance sequentially to `SRL_NUM_WORDS + 1`, which is always `SRL_UNKNOWN`. In between are the library words specific to the application program.
Table 4-11 illustrates the return values associated with the word library example shown in Figure 4-4.

**TABLE 4-11: RETURN VALUES FOR EXAMPLE LIBRARY**

<table>
<thead>
<tr>
<th>Enumerated Value</th>
<th>Return Value</th>
<th>Recognition Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRL_NONE</td>
<td>0</td>
<td>This value indicates there is no result yet (recognition isn't completed or the Speech Recognition Library is not initialized)</td>
</tr>
<tr>
<td>SRL_SECURITY</td>
<td>1</td>
<td>Represents recognized word 1</td>
</tr>
<tr>
<td>SRL_ARM</td>
<td>2</td>
<td>Represents recognized word 2</td>
</tr>
<tr>
<td>SRL_CAMERA</td>
<td>3</td>
<td>Represents recognized word 3</td>
</tr>
<tr>
<td>SRL_DISARM</td>
<td>4</td>
<td>Represents recognized word 4</td>
</tr>
<tr>
<td>SRL_EIGHT</td>
<td>5</td>
<td>Represents recognized word 5</td>
</tr>
<tr>
<td>SRL_FIVE</td>
<td>6</td>
<td>Represents recognized word 6</td>
</tr>
<tr>
<td>SRL_FOUR</td>
<td>7</td>
<td>Represents recognized word 7</td>
</tr>
<tr>
<td>SRL_NINE</td>
<td>8</td>
<td>Represents recognized word 8</td>
</tr>
<tr>
<td>SRL_ONE</td>
<td>9</td>
<td>Represents recognized word 9</td>
</tr>
<tr>
<td>SRL_SEVEN</td>
<td>10</td>
<td>Represents recognized word 10</td>
</tr>
<tr>
<td>SRL_SIX</td>
<td>11</td>
<td>Represents recognized word 11</td>
</tr>
<tr>
<td>SRL_THREE</td>
<td>12</td>
<td>Represents recognized word 12</td>
</tr>
<tr>
<td>SRL_TWO</td>
<td>13</td>
<td>Represents recognized word 13</td>
</tr>
<tr>
<td>SRL_ZERO</td>
<td>14</td>
<td>Represents recognized word 14</td>
</tr>
<tr>
<td>SRL_ZONE</td>
<td>15</td>
<td>Represents recognized word 15</td>
</tr>
<tr>
<td>SRL_UNKNOWN</td>
<td>16</td>
<td>Indicates the word was not recognized as belonging to the word library</td>
</tr>
</tbody>
</table>
FIGURE 4-4: EXAMPLE HEADER FILE

// Word Library Builder Version 1.00
//
// Word Library Vocabulary Header file
//
// Creation date: 10/19/04 16:40:05
//
// Word Library file: C:\temp\srl_manual\example.swl
// Language: English
// Comments: Created by Word Library Builder
//
// Word Library: example
// Noise profile: standard
// SNR level: 15 dB
// white: 100%
// car: 0%
// office: 0%
// Keyword: security
// number of words: 15

#define SRL_NUM_WORDS 15 // WL size

enum {
    SRL_NONE, // not ready yet
    SRL_SECURITY, // word 1 (keyword)
    SRL_ARM, // word 2
    SRL_CAMERA, // word 3
    SRL_DISARM, // word 4
    SRL_EIGHT, // word 5
    SRL_FIVE, // word 6
    SRL_FOUR, // word 7
    SRL_NINE, // word 8
    SRL_ONE, // word 9
    SRL_SEVEN, // word 10
    SRL_SIX, // word 11
    SRL_THREE, // word 12
    SRL_TWO, // word 13
    SRL_ZERO, // word 14
    SRL_ZONE, // word 15
    SRL_UNKNOWN, // unknown word
};
4.7.3 Parameters

This function has no parameters.

4.7.4 Resources

Table 4-12 lists the system resources used by the SR_Recognizer() function.

**TABLE 4-12: SR_RECOGNIZER() RESOURCE USAGE**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0-W7</td>
<td>Used, not restored</td>
</tr>
<tr>
<td>W8-W13</td>
<td>Used, restored</td>
</tr>
<tr>
<td>ACCA</td>
<td>Used, not restored</td>
</tr>
<tr>
<td>ACCB</td>
<td>Used, not restored</td>
</tr>
<tr>
<td>CORCON</td>
<td>Used, restored</td>
</tr>
<tr>
<td>TBLPAG</td>
<td>Used, restored</td>
</tr>
</tbody>
</table>

**Instructions**

<table>
<thead>
<tr>
<th></th>
<th>Used, two (2) levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO instructions</td>
<td>Used</td>
</tr>
<tr>
<td>REPEAT instructions</td>
<td>Used</td>
</tr>
</tbody>
</table>

4.7.5 Detailed Operation

The SR_Recognizer() function waits until DCIInterrupt() collects 84 speech samples and then performs five main tasks identified as:

<table>
<thead>
<tr>
<th>Task</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Copy the current word frame from the input buffer to the work buffer.</td>
</tr>
<tr>
<td>B</td>
<td>Analyze the current speech signal to detect word ending.</td>
</tr>
<tr>
<td>C</td>
<td>Calculate the feature vector for the current frame in the work buffer.</td>
</tr>
<tr>
<td>D</td>
<td>Quantize the feature vector using the vector codebook.</td>
</tr>
<tr>
<td>E</td>
<td>Search Hidden Markov Model for word match using the Viterbi algorithm.</td>
</tr>
</tbody>
</table>

Tasks A-D are time critical and must be completed before the next input buffer is filled. These tasks require 8 MIPS to successfully implement real-time calculations if all table data (VCB, HMM) are placed in the dsPIC30F program memory. If VCB is placed in the dsPIC30F program memory and all or part of HMM data are placed in an external memory, these four tasks require up to 9 MIPS. If all table data (VCB, HMM) are placed in an external memory, these four tasks require up to 13 MIPS. To preclude insufficient processing capability, it is strongly recommended that the vector codebook be stored in dsPIC30F program memory.

During the first four tasks the Pause flag is set to '0' (Pause = 0). While the next word frame is filling the input buffer, the contents of the work buffer are used to calculate a feature vector, which is matched against the vector codebook. The resulting feature vector is then collected into a feature-vector array, and the Idle flag is set to '1'. Tasks A-D are repeated until a word ending is detected by the library.

Task E is performed in background mode with sampling halted. During Task E, the Pause flag is set to '1' (Pause = 1), and word recognition is performed. The quantized feature vectors are compared to each HMM word model in the library. The SR_Recognizer() function is called successively until the feature-vector array has been processed against every word in the library.
Based on the Viterbi algorithm, each attempted match results in a probability rating (i.e., how closely the feature-vector array matches the current model word). The value SRL_None is returned after each comparison to allow the recognition process to continue. After the feature-vector array has been matched to the last word in the word library, the word represented by the highest probability of a match is returned and passed to the application program as a discrete value (e.g., SRL_ONE, SRL_SIX, etc.), based on the enumeration declaration in the word library header file. If none of the comparisons produces a satisfactory probability of a match, an “unknown word” response (SRL_UNKNOWN) is returned.

Note: To accommodate a flexible interface with user application code, SR_Recognizer() processes Tasks A-E in multiple stages. Refer to Section 5.2 “Incorporating Speech Recognition in an Application” for more information.

4.8 T4Interrupt() FUNCTION

4.8.1 Overview

The Timer4 interrupt handler implements the Listen Window used when keyword activation is enabled (SR_mode = 0 or 1).

4.8.2 Return Value

None

4.8.3 Parameters

None

4.8.4 Resources

Table 4-13 lists the resources used by the T4Interrupt() function.

TABLE 4-13: T4Interrupt() RESOURCE USAGE

<table>
<thead>
<tr>
<th>Resource</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registers</td>
<td></td>
</tr>
<tr>
<td>W0,W1</td>
<td>Used, restored</td>
</tr>
<tr>
<td>RCOUNT</td>
<td>Used, restored</td>
</tr>
<tr>
<td>Instructions</td>
<td></td>
</tr>
<tr>
<td>DO instructions</td>
<td>Not used</td>
</tr>
<tr>
<td>REPEAT instructions</td>
<td>Used</td>
</tr>
</tbody>
</table>

4.8.5 Detailed Operation

When keyword activation is enabled, the Timer4 interrupt handler will run at 16 times the period of the timeout specified in SR_Initialization(). The 16-bit timer is used with a 1:256 prescaler. The interrupt handler uses a software counter to produce the desirable timeout value. When the desired timeout value expires, the interrupt sets the SR_STATUSbits State flag to '1' indicating that the user must utter the keyword to activate recognition.

SR_Initialization() configures the Timer4 interrupt with the default priority level '4'. This priority level is not changed by the library, so you may modify the Timer4 interrupt priority level after SR_Initialization() is called. It is recommended that the Timer4 interrupt be given a low interrupt priority.
4.9 SR_STATUSBITS INTERACTION

With the exception of the Enable and Block flags, the SR_STATUSbits structure is read-only and controlled by the dsPIC30F Speech Recognition Library. Any attempt to change the read-only flags can lead to errors in the speech recognition results. The Enable and Block flags are readable and writable. You can change them as required by your application needs.

Figure 4-5 is an illustration showing the timing relationship between the different speech recognition status flags in the SR_STATUSbits structure.

**FIGURE 4-5: SPEECH RECOGNITION LIBRARY TIMING RELATIONSHIPS**

When SR_Initialization() is called, the Speech Recognition Library is initialized to the specified operating mode. The operating mode determines if a self-test is run and whether the library uses keyword activation. A baseline noise level is established and the State flag is set, putting the Speech Recognition Library into Listen mode. To trigger a recognition session, the user speaks into the microphone. If keyword activation is enabled, the State flag is set to '1', and the first word spoken must be the keyword to satisfy the user application. When the keyword is recognized, the State flag is set to '0', allowing any word in the library to be recognized. Recognition processing continues until a time-out occurs.

The dsPIC30F Data Converter Interface (DCI) converts the voice input data from the audio codec into discrete digital samples. The DCI interrupt handler captures the samples and loads them into the input buffer, four samples at a time, until the buffer capacity (84 samples) is reached. Each interrupt cycle takes roughly 330 microseconds. The 84-sample buffer represents one 7-msec segment of the word (one sound frame).

When the input buffer is full, the DCI Interrupt handler sets the Buffer flag to '1'. When SR_Recognizer() next runs, the input buffer is copied to the work buffer and frame processing begins. During this time, the Speech Recognition Library has the greatest processing demands as it performs the feature vector analysis on the contents of the work buffer. Meanwhile, the Buffer flag is reset (Buffer = 0) by the next DCI interrupt, and the input buffer is overwritten by the new DCI samples. When SR_Recognizer() completes analyzing the current frame, the Idle flag is set to '1'. A closeup of the Buffer and Idle flags is shown in Figure 4-6.
When the end of a word is detected, the word recognition process begins. During this time, frame vectors are processed against the Hidden Markov Model words to determine which library word should be selected. During this processing, the Pause flag is set to ‘1’, allowing the user to interrupt the library for an extended period of time (see Section 5.2 “Incorporating Speech Recognition in an Application”).
4.10 USING EXTERNAL MEMORY

The dsPIC30F Speech Recognition Library provides an external memory interface, which allows even memory-constrained applications to use voice control. The External Memory Interface uses an Am29F200B (AMD) 128K x 16-bit Flash memory device. The library is written to accommodate the use of external memory for storing any (or all) of the following data tables:

- Vector codebook
- Hidden Markov Models (all the models, or a portion of the models)
- Sample keyword utterance

When external memory is used, the Speech Recognition Library automatically accesses the external memory as required, instead of accessing the data in internal program memory. The location of these data tables is set in the library source file provided by the dsPIC30F Speech Recognition Word Library Builder. When the Word Library Builder generates the support files for your custom word library, you will have the opportunity to request the use of external memory.

The use of external memory results in the need for increased processing performance as well as the use of general purpose I/O pins. Since accessing external memory is slower than accessing internal memory, more processing time is required by the Speech Recognition library. As shown in Table 4-3, when the HMM words are stored externally, real-time performance requirements increase by 1 MIPs. When the vector codebook is stored externally, real-time processing requirements increase by 4 MIPs.

Microchip has a design for a small, external memory board that is compatible with the dsPICDEM 1.1 General Purpose Development Board. This external memory board plugs directly into the 100-pin header of the dsPICDEM 1.1 GPD board. The design for this board is shown in Appendix B. The pins required for using the external memory board are shown in Table 4-14.

TABLE 4-14: PINS USED BY SPEECH RECOGNITION LIBRARY

<table>
<thead>
<tr>
<th>dsPIC30F Pin</th>
<th>Application Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA6</td>
<td>Control RY/BY pin of the external memory</td>
</tr>
<tr>
<td>RA7</td>
<td>Control WE pin of the external memory</td>
</tr>
<tr>
<td>RC13</td>
<td>Control LE pin of the control circuitry</td>
</tr>
<tr>
<td>RD0-RD15</td>
<td>Transmit Address to the external memory</td>
</tr>
<tr>
<td></td>
<td>Receive Data from the external memory</td>
</tr>
<tr>
<td>RF7</td>
<td>Control CE pin of the external memory</td>
</tr>
<tr>
<td>RF8</td>
<td>Control OE pin of the external memory</td>
</tr>
</tbody>
</table>
4.11 REQUIRED dsPIC30F SYSTEM RESOURCES

4.11.1 Special Function Registers

The dsPIC30F Speech Recognition Library works with the following special function registers:

- For common calculations:
  ACCA, ACCB, CORCON, SR, TBLPAG, W0-W13
- For configuring and controlling the DCI module:
  DCICON1, DCICON2, DCICON3, DCISTAT, IEC2, IFS2, RSCON, RXBUF0, RXBUF1, RXBUF2, RXBUF3, TRISF, TRISG, TSCON, TXBUF0, TXBUF1, TXBUF2, TXBUF3
- For configuring and controlling Timer4:
  IEC1, IFS1, PR4, T4CON, TMR4
- For looping:
  DCOUNT, DOSTART, DOEND, RCOUNT
- For interfacing to an external memory (optional):
  PORTA, PORTC, PORTD, PORTF, TRISA, TRISC, TRISD, TRISF.

The Speech Recognition Library application uses the following CORCON setting:

\[ \text{CORCON} = 0x0002. \]

The dsPIC30F Speech Recognition Library has a mechanism that saves and restores the contents of the CORCON and TBLPAG registers. The user must save and restore the ACCA, ACCB, CORCON, SR, TBLPAG and W0-W13 registers if \( \text{SR}_\text{Recognizer}() \) is disrupted. Additionally, all DO and REPEAT registers must be saved and restored if these instructions are used by functions that disrupt \( \text{SR}_\text{Recognizer}() \) operation.

4.11.2 Peripherals

The Speech Recognition Library uses the dsPIC30F peripherals listed in Table 4-15.

**TABLE 4-15: PERIPHERALS USED BY SPEECH RECOGNITION LIBRARY**

<table>
<thead>
<tr>
<th>dsPIC30F Peripheral</th>
<th>Application Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCI module</td>
<td>Samples the input sound signal through the Si3000 audio codec. The DCI module is programmed as the Master controller providing a sampling rate of 12 KHz.</td>
</tr>
<tr>
<td>Timer4 (16-bit mode)</td>
<td>Implements the Listen Window timer (only required for SR_Mode 2 and 3.</td>
</tr>
<tr>
<td>Ports A, C, D and F</td>
<td>For accessing an external memory where a part of table data for Speech Recognition Library application may be placed, Speech Recognition Library uses Port RD and some pins of Ports RA, RC, RF (see Table 4-14).</td>
</tr>
</tbody>
</table>
4.12 RECOGNITION ACCURACY

The dsPIC30F Speech Recognition Library has been developed to yield the highest possible accuracy. However, recognition accuracy depends on a number of factors:

- Correct pronunciation of words by the speaker
- Speaker’s voice
- Ambient noise levels
- Size of word library
- Similarity of words in library

For example, recognition accuracy could diminish for the following reasons:

- The signal level is too low or too high.
- The signal is too noisy (SNR < 15 dB).
- Words in the library sound alike.

Consult the dsPIC30F Speech Recognition Word Library Builder User’s Guide (DS70137) for assistance in creating a word library that will achieve optimal recognition accuracy.
Chapter 5. Incorporating Speech Recognition

5.1 INTRODUCTION

This chapter provides information to help you understand how speech recognition is performed and what you must do to integrate speech recognition with your application. A basic understanding of the library’s functionality is important to ensure real-time processing, which must occur every 7 milliseconds. Items discussed in this chapter are:

- Incorporating Speech Recognition in an Application
- Building the Speech Recognition Library Into Your Code

5.2 INCORPORATING SPEECH RECOGNITION IN AN APPLICATION

When incorporating speech recognition in a dsPIC30F user application program, the user must consider the timing implications for the two main speech processing tasks: feature vector analysis and word recognition.

5.2.1 Feature Vector Analysis

Feature vector analysis is performed when the library is searching for the keyword or a new word (shown as the Keyword Search Mode and Listen Mode in Figure 3-2). Tasks A-D described in Section 4.7.5 “Detailed Operation” comprise this process. During this time, input speech frames are analyzed in real-time. A word utterance is received as a series of sound frames. Each sound frame consists of 84 digital samples (84 16-bit words) received by the DCI and passed to the Input buffer, and subsequently the Work buffer. Each sound frame must be analyzed before the next sound frame fills the Input buffer.

Feature vector analysis includes processing the sound frame to compute a feature vector for that sound frame. This process must be completed within 7 milliseconds for each sound frame to avoid losing any data in the next input buffer. This analysis step is resource intensive. For convenience, the processing is sliced into stages when the Block flag is set to '0'. When Block = 0, 9-19 stages, depending on the complexity of the data frame, are required to complete the analysis. Each stage consumes no more than 3500 instruction cycles for its execution and represents one call to SR_Recognizer(). This process is illustrated in Figure 5-1.
Staged processing allows the user application to periodically execute during feature vector analysis. If staged processing is not required, then one call to SR_Recognizer() will process the input buffer in approximately 56000 cycles. Table 5-1 shows the maximum time required to run SR_Recognizer() for feature vector analysis. The time in this table also shows the period during which you can safely execute application code without formally interrupting the feature vector analysis. Times are shown for Block flag = '0' and '1'.

Vector analysis is performed on the Work buffer contents. Analysis can be partitioned into stages of up to 3500 instruction cycles with each stage giving the user application an opportunity for processing.

SR_Initialization()
TABLE 5-1: FEATURE VECTOR ANALYSIS TIMING

<table>
<thead>
<tr>
<th>Performance (P)</th>
<th>Maximum Processing Time for SR_Recognizer()$^1$</th>
<th>Block=0$^2$</th>
<th>Block=1$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.288 MIPS</td>
<td></td>
<td>285 µsec</td>
<td>4.56 msec</td>
</tr>
<tr>
<td>18.432 MIPS</td>
<td></td>
<td>214 µsec</td>
<td>3.42 msec</td>
</tr>
<tr>
<td>24.576 MIPS</td>
<td></td>
<td>142 µsec</td>
<td>2.28 msec</td>
</tr>
</tbody>
</table>

Note 1: Assumes VCB is stored in internal (program) memory
2: Based on 3500 cycles
3: Based on 56000 cycles

During feature vector analysis, 8 MIPS of processing power (or approximately 56000 cycles) are required by the speech recognition program when the vector codebook resides in program memory. When the vector codebook is stored in external memory, 13 MIPS are required. This means that estimated total time available to execute user application code every 7 millisecond frame is shown in Table 5-2.

TABLE 5-2: ESTIMATED TIME AVAILABLE FOR USER APPLICATION

<table>
<thead>
<tr>
<th>Performance (P)</th>
<th>User Time Available (TA) Every 7-msec Frame</th>
<th>VCB in Program Memory</th>
<th>VCB in External Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.288 MIPS</td>
<td>2.44 msec</td>
<td>Not possible</td>
<td></td>
</tr>
<tr>
<td>18.432 MIPS</td>
<td>3.58 msec</td>
<td>2.06 msec</td>
<td></td>
</tr>
<tr>
<td>24.576 MIPS</td>
<td>4.72 msec</td>
<td>3.30 msec</td>
<td></td>
</tr>
</tbody>
</table>

5.2.2 Word Recognition

The second main speech processing task is interpretation of the word after the end of word is detected. During word recognition, the array of feature vectors is matched against all Hidden Markov Model words in the library. While this step is performed, an input signal is not being received or analyzed, so this is not considered a real time activity.

The number of stages depends on the number of words in your word library. During the recognition phase, each call to SR_Recognizer() processes one HMM word model, which takes a maximum of 12,000 instruction cycles to execute. For example, a 10-word library requires ten calls to SR_Recognizer().

Note: The Block flag does not affect word recognition processing.
Table 5-3 shows the maximum time required to run `SR_Recognizer()` for word recognition. The times in this table also show the period with which you can safely execute application code without formally interrupting the word recognition task.

### TABLE 5-3: WORD RECOGNITION TIMING

<table>
<thead>
<tr>
<th>Performance (P)</th>
<th>Maximum Processing Time for <code>SR_Recognizer()</code>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.288 MIPS</td>
<td>977 µsec</td>
</tr>
<tr>
<td>18.432 MIPS</td>
<td>651 µsec</td>
</tr>
<tr>
<td>24.576 MIPS</td>
<td>488 µsec</td>
</tr>
</tbody>
</table>

* Based on 12000 cycles

During word recognition, you can run application code for the total time determined by this equation:

$$TA = (100 - (12N/P)) \text{ msec}$$

where:

- $TA$ = time available
- $N$ = number of words in library
- $P$ = processor performance in MIPS

If your application code exceeds this time, there is a chance that the next word spoken by the user will be misinterpreted.

**Note:** During word recognition, the `Pause` flag is set to '1', which allows you to interrupt the recognition process for the time determined by this equation.
5.2.3 Library Integration

The simplest method of integrating the Speech Recognition Library is to use a
while(1) loop, with the initialization of the library occurring outside of the while(1)
loop. Figure 5-3 illustrates sample code for this type of solution. In this example, the
application waits until a valid response is returned from SR_Recognizer(), and then
it processes the recognized word. Until a word is recognized by the library, no action is
taken by the program.

A more complex application may not have the luxury of sitting idle, waiting for a word
to be recognized by the library. In these applications, other system processes must time
share with the Speech Recognition Library. This means that some external event, like
an interrupt, is used to signal when another application process must run. Since
SR_Recognizer() runs the longest during word recognition (as described in Section
5.2.2 “Word Recognition”), you can safely place calls to your application code in the
while(1) loop if the real-time constraints of your application are not less than the
times shown in Table 5-3. An example of such a call is shown in Figure 5-4. In this
example, a packet received over the UART is processed. The packet is processed only
when the global flag new_packet is not '0'. This flag is set by the UART interrupt
handler, after a complete packet is received and ready to be processed.
FIGURE 5-3: EXAMPLE OF SIMPLE SPEECH RECOGNITION CODE

//-------------------------------------------------------------
// dsPIC30F Speech Recognition Example Program
//-------------------------------------------------------------
#include "eval.h"
#include "SR_Lib.h"

//-------------------------------------------------------------
// Initialization Settings
//-------------------------------------------------------------
#define fclock 6144 // clock oscillator frequency
#define pll_setting 8 // PLL multiplier mode
#define timeout 5 // timeout period in seconds
#define mic_gain 20 // microphone gain in dB
#define SR_mode 1 // no self-test, use keyword activation

// Command Processing Function
extern void ProcessCommand (int word);

int main(void)
{
  int word;

  // Initialize speech recognizer
  SR_Initialization(fclock, pll_setting, timeout, mic_gain, SR_mode);

  // SR Main Working Cycle
  while (1)
  {
    // recognize words spoken by the user
    word = SR_Recognizer();
    switch (word) {
      case SRL_NONE:
        // recognition in process - doesn't require action
        break;
      case SRL_CONTROL ... SRL_TWO:
        // valid word recognized - process it
        ProcessCommand(word);
        break;
      case SRL_UNKNOWN:
        default:
        // unknown response - doesn't require action
        break;
    } /* end switch(word) */
  } /* end while(1) */
} /* end main() */
Incorporating Speech Recognition

FIGURE 5-4: CALLING APPLICATION CODE FROM THE WHILE(1) LOOP

```c
if (new_packet)
{
    /* process the packet */
    ProcessPacket();
    new_packet = 0;
}
```

When your application code is called from the while(1) loop, the real-time operational requirements of the Speech Recognition Library must still be realized. Namely, you must ensure that enough processing time is provided to the Speech Recognition Library to perform the feature vector analysis (Table 5-2) and word recognition (Table 5-3).

If the real-time constraints of your application are faster than 977 microseconds, you do not have the luxury of servicing your application’s demands within the while(1) loop. Instead, your application code should be executed directly from the handler that interrupts SR_Recognizer(). For instance, if the application must start to transmit a packet within 20 µsec of a timer interrupt, the timer interrupt handler should directly initiate the packet transmission.

It is important to remember that there are gaps in speech recognition processing during which you can use the processor for your application program. You should consider this capability when you architect your application. As described in Section 5.2.1 “Feature Vector Analysis”, several milliseconds of processing time are freed during feature vector analysis. You can also take advantage of the pause between words to perform other application tasks.

An additional important consideration related to library integration is the use of interrupts. During feature vector analysis, approximately every 330 µsec the DCI interrupt handler must be processed. If this interrupt is not allowed to run as required, speech data will be lost and the recognition accuracy will be jeopardized. It is recommended that you set the interrupt priority of your application's interrupts such that the DCI Interrupt has the highest priority. The DCI interrupt is discussed in detail in Section 4.6 “DCIInterrupt( ) Function”.

5.2.4 Application Design Tips

5.2.4.1 KEYWORD ACTIVATION

To make a robust application, it is recommended that you use the keyword activation feature. To use keyword activation you call SR_Initialization() using either SR_Mode 0 or 1. In these modes, the recognizer is less likely to act on extraneous noises that may be incorrectly identified as a valid word. When the library is enabled, it is constantly sampling microphone data, and it is possible that a sharp external noise adequately above the noise threshold could be recognized as a word in the library. The keyword activation feature will minimize the chance of this happening when your system is idle.
5.2.4.2 SOFTWARE MENUS

Even though there is only one keyword, it is possible to add software menus to your application using a state machine. In Figure 5-5 illustrates a state machine for a simple application that uses a seven-word library:

**TABLE 5-4: WORD LIBRARY FOR STATE MACHINE**

<table>
<thead>
<tr>
<th>Library Word</th>
<th>Control Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>control (keyword)</td>
<td>Enables the voice controlled user-interface</td>
</tr>
<tr>
<td>fan</td>
<td>Turns fan on</td>
</tr>
<tr>
<td>high</td>
<td>Switches fan speed to high</td>
</tr>
<tr>
<td>light</td>
<td>Toggles light on or off</td>
</tr>
<tr>
<td>low</td>
<td>Switches fan speed to low</td>
</tr>
<tr>
<td>off</td>
<td>Turns fan off</td>
</tr>
<tr>
<td>medium</td>
<td>Switches fan speed to medium</td>
</tr>
</tbody>
</table>

**FIGURE 5-5: RECOGNITION STATES**

In this application a light and fan are voice activated. The light is toggled on and off. The fan is turned on, and when it is on it can be switched to high, medium or low speed or switched off. When the application is powered up, it enters STATE 0 (Keyword Search mode), where it waits for the keyword to be uttered.

When the keyword ("control") is recognized, the application enters STATE 1 (LISTEN mode) and listens for another incoming word. A new word must be processed within the timeout period or the application reverts to Keyword Search mode (STATE 0). In STATE 1, when the word "fan" is recognized, the fan is turned ON and control passes to STATE 2. When the word "light" is recognized, the lighted is toggled on or off and control passes to STATE 3.

STATE 2 responds only to the words "high", "medium", "low" and "off". Any other word detected during STATE 2 is ignored. STATE 3 doesn’t respond to any word. When either of these states time out, control reverts to STATE 0, which requires the keyword to be restated by the user. Each state prevents any false recognition by the library from affecting the application.
5.3 BUILDING THE SPEECH RECOGNITION LIBRARY INTO YOUR CODE

5.3.1 Speech Recognition Support Files

The dsPIC30F Speech Recognition Library requires several custom support files that are unique to your own word library. These files are required for building with the Speech Recognition library and are created with the dsPIC30F Speech Recognition Word Library Builder. The list of support files is shown in Table 5-5. Note that the base file name of these files correspond to the name of the word library.

<table>
<thead>
<tr>
<th>File name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>[filename]_int.s</td>
<td>Data files for the Vector Codebook, Hidden Markov Models and the keyword sample that are allocated to internal memory.</td>
</tr>
<tr>
<td>[filename]_ext.s</td>
<td>Data files for the Vector Codebook, Hidden Markov Models and the keyword sample that are allocated to external memory.</td>
</tr>
<tr>
<td>[filename].h</td>
<td>Header file that contains a set of constants for each word in the word library. This file is used to interpret the value returned by the SR_Recognizer() function in the dsPIC30F Speech Recognition Library.</td>
</tr>
<tr>
<td>[filename]_voc.s</td>
<td>Data file that contains the ASCII string of each word in the library. This file is used with the dsPIC30F Speech Recognition Library demonstration program to display the actual library word on the LCD.</td>
</tr>
</tbody>
</table>

5.3.2 Building the Speech Recognition Library

In addition to the support files identified above, the library archive file, SR_LIB.A, and library header file, SR_LIB.H, are required. When using MPLAB, these files must be added to your project to build with the Speech Recognition library.

<table>
<thead>
<tr>
<th>File name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>sr_lib.a</td>
<td>Object file for the Speech Recognition Library, residing in the directory where Speech Recognition Library is installed.</td>
</tr>
<tr>
<td>sr_lib.h</td>
<td>Header file for the Speech Recognition Library, residing in the directory where Speech Recognition Library is installed.</td>
</tr>
</tbody>
</table>

To integrate the Speech Recognition Library into your user application program, follow these steps:

1. Copy the files from Table 5-5 to your project folder.
2. Include SR_Lib.a in your project in MPLAB IDE.
3. Add [filename]_int.s and [filename]_ext.s as source files to the project in MPLAB IDE.
4. Add sr_lib.h and [filename].h files as include files to your project in MPLAB IDE.
5. Make a call to SR_Initialization() from your application. Here you must consider oscillator frequency, PLL settings, desired time-out value, microphone sensitivity, and operating mode. See Section 4.5 “SR_Initialization() Function” for details.
6. Build in calls to SR_Recognizer() into your application using the guidelines in Section 5.2 “Incorporating Speech Recognition in an Application”.
7. Build the application using the MPLAB IDE.
Chapter 6. Demo Application

6.1 INTRODUCTION

This section describes the Speech Recognition Library demonstration. Items discussed in this chapter are:

• Demo Description
• Demo Setup
• Demonstration Application

6.2 DEMO DESCRIPTION

The Speech Recognition Library demonstration is a sample application that runs on the dsPICDEM 1.1 Development Board. The application provides speaker independent recognition of 10 library words, which are listed in Table 6-1. Recognition results are displayed on the LCD that is part of the dsPICDEM 1.1 board.

<table>
<thead>
<tr>
<th>Word 1</th>
<th>alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word 2</td>
<td>control – keyword</td>
</tr>
<tr>
<td>Word 3</td>
<td>light</td>
</tr>
<tr>
<td>Word 4</td>
<td>off</td>
</tr>
<tr>
<td>Word 5</td>
<td>one</td>
</tr>
<tr>
<td>Word 6</td>
<td>security</td>
</tr>
<tr>
<td>Word 7</td>
<td>start</td>
</tr>
<tr>
<td>Word 8</td>
<td>temperature</td>
</tr>
<tr>
<td>Word 9</td>
<td>three</td>
</tr>
<tr>
<td>Word 10</td>
<td>two</td>
</tr>
</tbody>
</table>

The demo application uses the self-test function. When power is applied to the board, the program runs automatically. First it performs the self test, then it waits for the keyword to be spoken before recognizing any other word.
6.3 DEMO SETUP

Follow these steps to set up the demonstration. For best performance, Microchip recommends using the Plantronic .Audio 40 headset, which comes with Accessory Kit AC300031.

6.3.1 Configure dsPICDEM 1.1 Board

1. Insert a 6.144 MHz oscillator into oscillator socket U5 on the dsPICDEM 1.1 board, as shown in Figure 6-3. The oscillator included in the dsPICDEM 1.1 Accessory Kit can be used or supply your own.

2. Set the jumper marked J9 to the ‘Slave’ position. This jumper allows the on-board Si3000 codec to function as slave device to the dsPIC30F.

3. Apply power to the board.
6.3.2 Program dsPIC30F Device

After the board is configured a project must be built in MPLAB IDE to program the demo application into the dsPIC30F device on the dsPICDEM 1.1 board. Use these steps:

1. Open MPLAB IDE. Open the ASR_Demo.mcp project located in the Demo folder.
2. Select Project>Build All to build the project.
3. Connect the MPLAB ICD 2 to the dsPICDEM 1.1 Board (see Figure 6-2). Program the dsPIC30F6014 device on the board (Programmer>Program).

FIGURE 6-2: MPLAB ICD 2 SETUP

4. When the project output window shows that programming is complete, as shown in Figure 6-3, disconnect the MPLAB ICD 2.

FIGURE 6-3: PROGRAMMING STATUS WINDOW
6.3.3 Set up Demo

After the dsPIC30F6014 is programmed, attach the headset, as shown in Figure 6-4. If a different microphone is being used, the mic_gain parameters in SR_Initialization() may need to be changed.

FIGURE 6-4: DEMO SET UP

dsPICDEM™ 1.1 Development Board running Speech Recognition Library demo application

115 VAC → 9VDC → MIC IN (J16) → SPKR OUT (J17) → LCD displays status of Speech Recognition
6.4 DEMONSTRATION APPLICATION

After the demo application has been programmed it is ready to run. Follow these steps:

1. The headset should be put on and the microphone adjusted to within 1-2 inches of the user’s mouth.

2. The Reset button on the dsPICDEM 1.1 board should be pressed. The LCD will indicate that the program is running, as shown in Figure 6-5.

FIGURE 6-5: DEMO STARTUP DISPLAY

![Demo Startup Display]

The LCD indicates that the program is self-testing, and then shows that the self-test completed successfully, as shown in Figure 6-6. When the self-test runs, “control” will be heard playing back through the headset.

FIGURE 6-6: SELF TEST COMPLETION MESSAGE

![Self Test Completion Message]

Note: If the self test fails, contact Microchip for resolution.

3. The user should speak the keyword (“Control”) into the microphone.

   The demo application uses optional keyword activation. The keyword must be recognized before any other words can be used. See Section 4.5.8 “SR_mode Parameter (Int)” for additional information.

   When the keyword is recognized the LCD displays a message similar to Figure 6-7.

FIGURE 6-7: SUCCESSFUL WORD RECOGNITION MESSAGE

![Successful Word Recognition Message]
In addition to displaying the recognized word, the program provides feedback to the speaker. The second row indicates the loudness level, which allows the speaker to either speak louder or softer to achieve optimum recognition accuracy. Variants on the loudness level are:

<table>
<thead>
<tr>
<th>Loudness Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Corresponds to a good manner of speaking.</td>
</tr>
<tr>
<td>Too HIGH</td>
<td>Indicates that the speaker uttered the last word too loudly, or the microphone is too close to the speaker for the preset microphone attenuation in the audio codec, causing the sound level to exceed the input dynamic range.</td>
</tr>
<tr>
<td>Too LOW</td>
<td>Indicates that the speaker has uttered last word in too softly, or the microphone is too far from the speaker for the preset microphone attenuation in the audio codec, causing the sound level to be lower than the admissible threshold.</td>
</tr>
</tbody>
</table>

The third row displays the Signal-to-Noise Ratio level (SNR) for the last spoken word. The fourth row displays the measured Speech_level (SL) and Noise_level (NL). These level values are defined as:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech_level</td>
<td>Level (dB) of the active speech signal between detected start-points</td>
</tr>
<tr>
<td>Noise_level</td>
<td>Level (dB) of the current background noise</td>
</tr>
<tr>
<td>SNR_level</td>
<td>Calculated ratio of speech level to noise level</td>
</tr>
</tbody>
</table>

See Section 4.3.2 “SR_SNRLevel Structure” for details.

4. Speak any word in the demo word library and observe the result.
5. Repeat step 3 for each word. If a word is not recognized, try speaking louder or softer, or more distinctly.
6. After exercising the library, allow the program to sit idle for several seconds (more than five seconds).

The LCD display requests the keyword to initiate a new recognition session, as shown in Figure 6-8.

**FIGURE 6-8: PAUSE MODE MESSAGE**

```
0001 CONTROL
Loudness: Normal
SNR = 15.4 dB
Speak a Keyword!
```
Appendix A. Master Word Library

A.1 INTRODUCTION

The master word library for the dsPIC30F Speech Recognition Library is a discrete list of 100 words from which you can choose a vocabulary for your user application. Table A-1 lists the full possible vocabulary for the library.

<table>
<thead>
<tr>
<th>TABLE A-1: MASTER WORD LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>alarm</td>
</tr>
<tr>
<td>answer</td>
</tr>
<tr>
<td>arm</td>
</tr>
<tr>
<td>back</td>
</tr>
<tr>
<td>balance</td>
</tr>
<tr>
<td>bass</td>
</tr>
<tr>
<td>brake</td>
</tr>
<tr>
<td>call</td>
</tr>
<tr>
<td>camera</td>
</tr>
<tr>
<td>cancel</td>
</tr>
<tr>
<td>cd</td>
</tr>
<tr>
<td>channel</td>
</tr>
<tr>
<td>clear</td>
</tr>
<tr>
<td>close</td>
</tr>
<tr>
<td>computer</td>
</tr>
<tr>
<td>control</td>
</tr>
<tr>
<td>cook</td>
</tr>
<tr>
<td>delete</td>
</tr>
<tr>
<td>dial</td>
</tr>
<tr>
<td>disarm</td>
</tr>
<tr>
<td>door</td>
</tr>
<tr>
<td>down</td>
</tr>
<tr>
<td>dvd</td>
</tr>
<tr>
<td>eight</td>
</tr>
<tr>
<td>eject</td>
</tr>
<tr>
<td>e-mail</td>
</tr>
<tr>
<td>fan</td>
</tr>
<tr>
<td>five</td>
</tr>
<tr>
<td>forward</td>
</tr>
<tr>
<td>four</td>
</tr>
<tr>
<td>front</td>
</tr>
<tr>
<td>go</td>
</tr>
<tr>
<td>halt</td>
</tr>
<tr>
<td>hangup</td>
</tr>
</tbody>
</table>
Appendix B. External Memory Board

B.1 INTRODUCTION

This appendix provides a design for an external memory board that allows portions of the Speech Recognition Library to operate from off-chip memory.
FIGURE B-1: SCHEMATIC DIAGRAM
# Index

## A
- Accuracy .................................................. 38
- AMD Memory Module ................................. 20
- Audio Codec ............................................. 20

## B
- Background Interference............................... 16
- Baseline Noise Level ................................... 34
- Buffer Flag .............................................. 21, 23

## C
- Calling Function
  - SR_Initilization() .................................. 24
  - SR_Recognizer() ...................................... 24
- Customer Notification Service ..................... 5
- Customer Support ...................................... 6

## D
- Data Converter Interface .............................. 34
- DCI Interrupt Handler .................................. 34
- Documentation
  - Conventions ........................................ 3

## E
- Enable Flag .............................................. 22
- External Memory Module ............................. 20

## F
- Feature Vector Analysis ............................... 34, 39
- Feature-Vector Array .................................. 32
- Free Software Foundation ............................ 4

## G
- GNU Language Tools .................................... 4

## H
- Header File ............................................. 24
- Hidden Markov Model .................................. 35, 41
- HMM Data File .......................................... 9

## I
- Initialization Parameter
  - SR_mode .................................................. 26
- Initialization Resources .............................. 27
- Input Buffer ......................................... 34, 39
- Installation
  - Procedure ........................................... 11
  - System Requirements ............................ 11
- Internet Address ....................................... 5

## K
- Keyword
  - Activation ............................................. 16, 34
  - Search Mode ......................................... 16
  - Keyword Search Mode .............................. 39

## L
- Listen Mode ............................................. 16, 34, 39

## M
- Master Word List ....................................... 55
- Memory Requirements .................................. 20
- Microchip Web Site ................................... 5
- Microphone ............................................. 20

## O
- Office Noise ............................................. 16

## P
- Passenger Compartment Noise .................... 16
- Pause ..................................................... 16
- Pause Flag ............................................. 22, 32
- Performance Requirements ......................... 20

## R
- Reading, Recommended ............................... 4
- Readme .................................................. 4
- Required Resources
  - Special Function Registers ....................... 37
- Return Value
  - SR_Initilization() Function ...................... 25
  - SR_Recognizer() Function ......................... 29

## S
- Self-test .................................................. 16
- Self-test Mode ........................................ 27
- Signal-to-Noise Ratio ................................ 16
- Software Requirements ............................. 20
- Sound Frame ........................................ 21, 29, 34, 39
- Special Function Registers ......................... 37
- Speech Recognition
  - Operating Mode ..................................... 16, 34
  - Overview ............................................. 15
  - References ........................................... 10
  - Self Test ............................................. 9
  - Self-test ............................................. 16
  - Timeout Period .................................... 16
  - Speech Sampling Rate ............................ 19
- SR_Initilization() Function
  - Overview ............................................. 25
  - Parameters ......................................... 25
  - Return Value ....................................... 25
<table>
<thead>
<tr>
<th>Location</th>
<th>Contact Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AMERICAS</strong></td>
<td></td>
</tr>
<tr>
<td>Corporate Office</td>
<td>2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: <a href="http://support.microchip.com">http://support.microchip.com</a> Web Address: <a href="http://www.microchip.com">www.microchip.com</a></td>
</tr>
<tr>
<td>Atlanta</td>
<td>Alpharetta, GA Tel: 770-640-0034 Fax: 770-640-0307</td>
</tr>
<tr>
<td>Boston</td>
<td>Westford, MA Tel: 978-692-3848 Fax: 978-692-3821</td>
</tr>
<tr>
<td>Chicago</td>
<td>Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075</td>
</tr>
<tr>
<td>Dallas</td>
<td>Addison, TX Tel: 972-818-7423 Fax: 972-818-2924</td>
</tr>
<tr>
<td>Detroit</td>
<td>Farmington Hills, MI Tel: 248-538-2250 Fax: 248-538-2260</td>
</tr>
<tr>
<td>Kokomo</td>
<td>Kokomo, IN Tel: 765-864-8360 Fax: 765-864-8387</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608</td>
</tr>
<tr>
<td>San Jose</td>
<td>Mountain View, CA Tel: 650-215-1444 Fax: 650-961-0286</td>
</tr>
<tr>
<td>Toronto</td>
<td>Mississauga, Ontario, Canada Tel: 905-673-0699 Fax: 905-673-6509</td>
</tr>
<tr>
<td><strong>ASIA/PACIFIC</strong></td>
<td></td>
</tr>
<tr>
<td>Australia - Sydney</td>
<td>Tel: 61-2-9868-6733 Fax: 61-2-9868-6755</td>
</tr>
<tr>
<td>China - Beijing</td>
<td>Tel: 86-10-8528-2100 Fax: 86-10-8528-2104</td>
</tr>
<tr>
<td>China - Chengdu</td>
<td>Tel: 86-28-8676-6200 Fax: 86-28-8676-6599</td>
</tr>
<tr>
<td>China - Fuzhou</td>
<td>Tel: 86-591-8750-3506 Fax: 86-591-8750-3521</td>
</tr>
<tr>
<td>China - Hong Kong SAR</td>
<td>Tel: 852-2401-1200 Fax: 852-2401-3431</td>
</tr>
<tr>
<td>China - Shanghai</td>
<td>Tel: 86-21-5407-5533 Fax: 86-21-5407-5066</td>
</tr>
<tr>
<td>China - Shenyang</td>
<td>Tel: 86-24-2334-2829 Fax: 86-24-2334-2393</td>
</tr>
<tr>
<td>China - Shenzhen</td>
<td>Tel: 86-755-8203-2660 Fax: 86-755-8203-1760</td>
</tr>
<tr>
<td>China - Shunde</td>
<td>Tel: 86-757-2839-5507 Fax: 86-757-2839-5571</td>
</tr>
<tr>
<td>China - Qingdao</td>
<td>Tel: 86-532-502-7355 Fax: 86-532-502-7205</td>
</tr>
<tr>
<td>India - Bangalore</td>
<td>Tel: 91-80-2229-0061 Fax: 91-80-2229-0062</td>
</tr>
<tr>
<td>India - New Delhi</td>
<td>Tel: 91-11-5160-8631 Fax: 91-11-5160-8632</td>
</tr>
<tr>
<td>Japan - Kanagawa</td>
<td>Tel: 81-45-471-6166 Fax: 81-45-471-6122</td>
</tr>
<tr>
<td>Korea - Seoul</td>
<td>Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934</td>
</tr>
<tr>
<td>Singapore</td>
<td>Tel: 65-6334-8870 Fax: 65-6334-8850</td>
</tr>
<tr>
<td>Taiwan - Kaohsiung</td>
<td>Tel: 886-7-536-4818 Fax: 886-7-536-4803</td>
</tr>
<tr>
<td>Taiwan - Taipei</td>
<td>Tel: 886-2-2500-6610 Fax: 886-2-2508-0102</td>
</tr>
<tr>
<td>Taiwan - Hsinchu</td>
<td>Tel: 886-3-572-9526 Fax: 886-3-572-6459</td>
</tr>
<tr>
<td><strong>EUROPE</strong></td>
<td></td>
</tr>
<tr>
<td>Austria - Weis</td>
<td>Tel: 43-7242-2244-399 Fax: 43-7242-2244-393</td>
</tr>
<tr>
<td>Denmark - Ballerup</td>
<td>Tel: 45-4450-2828 Fax: 45-4485-2829</td>
</tr>
<tr>
<td>France - Massy</td>
<td>Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79</td>
</tr>
<tr>
<td>Germany - Ismaning</td>
<td>Tel: 49-89-627-144-0 Fax: 49-89-627-144-44</td>
</tr>
<tr>
<td>Italy - Milan</td>
<td>Tel: 39-0331-742611 Fax: 39-0331-466781</td>
</tr>
<tr>
<td>Netherlands - Drunen</td>
<td>Tel: 31-416-690399 Fax: 31-416-690340</td>
</tr>
<tr>
<td>England - Berkshire</td>
<td>Tel: 44-118-921-5869 Fax: 44-118-921-5820</td>
</tr>
</tbody>
</table>