Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Embedded SDK (Software Development Kit)

G.711 Log-PCM Library

SDK117/D
Rev. 2, 07/19/2002

© Motorola, Inc., 2002. All rights reserved.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Contents

About This Document

Chapter 1
Introduction

Chapter 2
Directory Structure

Chapter 3
G.711 Library Interfaces

Chapter 4
Building the G.711 Library

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

List of Tables

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

List of Figures

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

List of Examples

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

About This Document

This manual describes the Log-PCM, (G.711), algorithm for use with Motorola's Embedded Software Development Kit, (SDK).

Audience

This document targets software developers implementing linear PCM to log-PCM functions, and vice versa, within software applications.

Organization

This manual is arranged in the following sections:

- **Chapter 1, Introduction**—provides a brief overview of this document
- **Chapter 2, Directory Structure**—provides a description of the required core directories
- **Chapter 3, G.711 Library Interfaces**—describes all of the G.711 Library functions
- **Chapter 4, Building the G.711 Library**—tells how to execute the system library project build
- **Chapter 5, Linking Applications with the G.711 Library**—describes organization of the G.711 Library
- **Chapter 6, G.711 Applications**—describes the use of G.711 Library through test/demo applications
- **Chapter 7, License**—provides the license required to use this product

Suggested Reading

We recommend that you have a copy of the following references:

- *DSP56800 Family Manual*, DSP56800FM/AD
- *DSP56824 User's Manual*, DSP56824UM/AD
- *Inside CodeWarrior: Core Tools*, Metrowerks Corp.

For More Information On This Product,
Go to: www.freescale.com

Conventions

This document uses the following notational conventions:

Typeface, Symbol or Term	Meaning	Examples
`Courier Monospaced Type`	Code examples	`//Process command for line flash`
Italic	Directory names, project names, calls, functions, statements, procedures, routines, arguments, file names, applications, variables, directives, code snippets in text	...and contains these core directories: *applications* contains applications software... ...CodeWarrior project, *3des.mcp* is... ...the *pConfig* argument.... ...defined in the C header file, *aec.h*....
Bold	Reference sources, paths, emphasis	...refer to the **Targeting DSP56F80x Platform** manual.... ...see: **C:\Program Files\Motorola\Embedded SDK\help\tutorials**
Blue Text	Linkable on-line	...refer to Chapter 7, License....
Number	Any number is considered a positive value, unless preceded by a minus symbol to signify a negative value	3V -10 DES^{-1}
ALL CAPITAL LETTERS	# defines/ defined constants	# define INCLUDE_STACK_CHECK
Brackets [...]	Function keys	...by pressing function key [F7]
Quotation marks, “...”	Returned messages	...the message, “Test Passed” is displayed.... ...if unsuccessful for any reason, it will return “NULL”...

Definitions, Acronyms, and Abbreviations

The following list defines the acronyms and abbreviations used in this document. As this template develops, this list will be generated from the document. As we develop more group resources, these acronyms will be easily defined from a common acronym dictionary. Please note that while the acronyms are in solid caps, terms in the definition should be initial capped ONLY IF they are trademarked names or proper nouns.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

DSP	Digital Signal Processor or Digital Signal Processing
FFT	Fast Fourier Transforms
FIR	Finite Impulse Response
I/O	Input/Output
IDE	Integrated Development Environment
IIR	Infinite Impulse Response
LSB	Least Significant Bit
MAC	Multiply/Accumulate
MIPS	Million Instructions Per Second
MSB	Most Significant Bit
OnCE™	On-Chip Emulation
OMR	Operating Mode Register
PC	Program Counter
PCM	Pulse Code Modulation
SDK	Software Development Kit
SP	Stack Pointer
SPI	Serial Peripheral Interface
SR	Status Register
SRC	Source

References

The following sources were used to produce this book:

1. *DSP56800 Family Manual*, DSP56800FM/AD
2. *DSP56824 User's Manual*, DSP56824UM/AD
3. *Embedded SDK Programmer's Guide*
4. ITU-T Recommendation G.711 (log-PCM)

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 1
Introduction

Welcome to Motorola's Family of Digital Signal Processors, (DSPs). This document describes the G.711 PCM Encoding and Decoding (log-PCM) Library, which is a part of Motorola's comprehensive Embedded Software Development Kit (SDK) for its DSPs. In this document, you will find all the information required to use and maintain the G.711 log-PCM Library interface and algorithms.

Motorola provides these algorithms to you for use on the Motorola Digital Signal Processors to expedite your application development and reduce the time it takes to bring your own products to market.

Motorola's G.711 log-PCM Library is licensed for your use on Motorola processors. Please refer to the standard Software License Agreement in **Chapter 7** for license terms and conditions; please consult with your Motorola representative for premium product licensing.

1.1 Quick Start

Motorola Embedded SDK is targeted to a large variety of hardware platforms. To take full advantage of a particular hardware platform, use **Quick Start** from the appropriate **Targeting Motorola DSP568xx Platform** documentation.

For example, the **Targeting Motorola DSP56824 Platform** manual provides more specific information and examples about this hardware architecture. If you are developing an application for a DSP56824EVM board or any other DSP56824 development system, refer to the **Targeting Motorola DSP56824 Platform** manual for **Quick Start** or other DSP56824-specific information.

1.2 Overview of G.711

The G.711-based Log-PCM conforms to the ITU-T Recommendation G.711, previously called "CCITT Recommendation".

The characteristics mentioned in the standard are recommended for encoding (A-law or µlaw) voice frequency signals to yield a bit rate of 64 Kbits/s at 8000Hz sampling rate.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

1.2.1 Background

The voice frequency signals are bandlimited to 4KHz, sampled at 8KHz and quantized with 16 bits per sample and correspond to the bit rate of 16 x 8K = 128Kbps. There are two methods to conserve bandwidth and allocate more bits to low-magnitude speech signals, (companding): A-law and µlaw, (µ255). Both companding methods work on 255 unequal quantization levels, which corresponds to 8 bits per sample. Thus the bit rate is 8 x 8K = 64kbit/s.

1.2.2 Features and Performance

The G.711 library is multichannel and re-entrant.

For details on Memory and MIPS for a particular DSP, refer to the **Libraries** chapter of the appropriate Targeting manual.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Chapter 2
Directory Structure

2.1 Required Core Directories

Figure 2-1 details required platform directories:

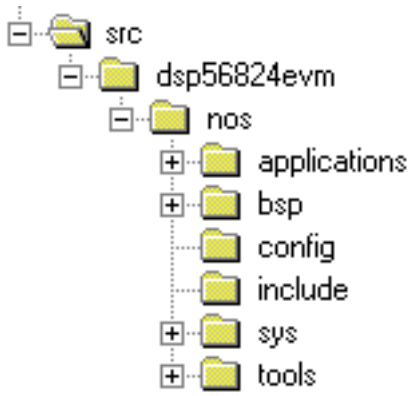

Figure 2-1. Core Directories

In this example, the DSP56824EVM has no operating system (nos) support. This platform contains the following core directories:

- ***applications*** contains applications software that can be exercised on this platform
- ***bsp*** contains board support package specific for this platform
- ***config*** contains default hardware/software configurations for this platform
- ***include*** contains SDK header files which define the Application Programming Interface
- ***sys*** contains required system components
- ***tools*** contains utilities used by system components

There are also optional directories that include domain-specific libraries.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

2.2 Optional (Domain-Specific) Directories

Figure 2-2 demonstrates how the G.711 algorithm is encapsulated in the domain-specific directories under the directory *telephony*.

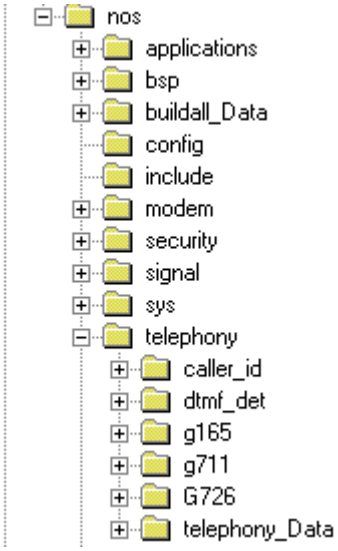

Figure 2-2. DSP56824 Directories

The *telephony* directory includes telephony-specific algorithms. Figure 2-3 shows the *g711* directory structure under the *telephony* directory.

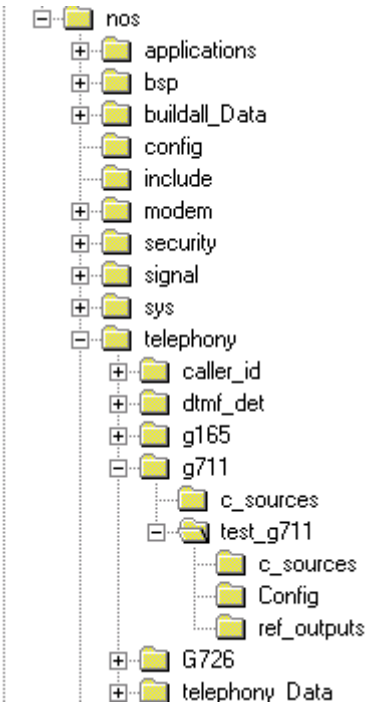

Figure 2-3. *telephony* Directory Structure

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

The *g711* directory under *telephony* includes these directories:

- ***c_sources*** includes only the process function APIs for the G.711
- ***test_g711*** includes C sources and configuration necessary for testing G.711 library modules
  - ***c_sources*** contains an example test code
  - ***Config*** contains configuration files *appconfig.c*, *appconfig.h* and *linker.cmd* specific to the G.711

The *applications* directory includes high-level software that exercises the *g711* library. **Figure 2-4** shows that the *applications* directory contains the *g711* application under the *telephony* sub-directory.

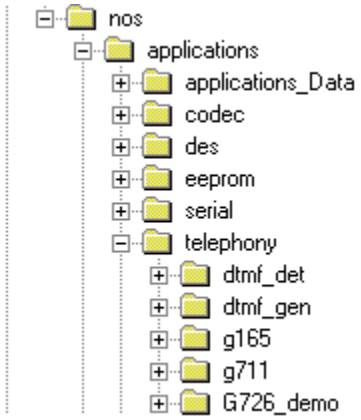

Figure 2-4. G.711 Application

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Chapter 3
G.711 Library Interfaces

3.1 G.711 Services

The G.711 library converts 16-bit PCM samples to A-law or µ-law 8-bit samples. The data to be supplied must be in 16-bit word, fixed-point (1.15) format, shown below:

i = information bit
s = sign

3.2 Interface

The C interface for G.711 library services is defined in the C header file *g711.h*, shown in **Code Example 3-1**.

Code Example 3-1. C Header File *g711.h*

```
/* File: g711.h */

#ifndef __G711_H
#define __G711_H

/*
   PCM Encoding (A-law / Mu-law Conversion)
*/

/***************************
 Foundational Include Files
***************************/

#include "port.h"
```


For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
/**************************
 Function Prototypes
**************************/

EXPORT Result g711_linear2alaw( Int16 *pPCM_values, unsigned char *pA_values,
                                UInt16 NumSamples);

EXPORT Result g711_alaw2linear(  unsigned char *pA_values, Int16 *pPCM_values,
                                UInt16 NumSamples);

EXPORT Result g711_linear2ulaw(  Int16 *pPCM_values, unsigned char *pU_values,
                                UInt16 NumSamples);

EXPORT Result g711_ulaw2linear(  unsigned char *pU_values, Int16 *pPCM_values,
                                UInt16 NumSamples);

EXPORT Result g711_ulaw2alaw(  unsigned char*pUval, unsigned char *pAval,
                                UInt16 NumSamples);

EXPORT Result g711_alaw2ulaw(  unsigned char*pAval, unsigned char *pUval,
                                UInt16 NumSamples);

#endif
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3 Specifications

The following pages describe the G.711 library functions.

Function arguments for each routine are described as *in*, *out*, or *inout*. An *in* argument means that the parameter value is an input only to the function. An *out* argument means that the parameter value is an output only from the function. An *inout* argument means that a parameter value is an input to the function, but the same parameter is also an output from the function.

Typically, *inout* parameters are input pointer variables in which the caller passes the address of a preallocated data structure to a function. The function stores its results within that data structure. The actual value of the *inout* pointer parameter is not changed.

Note: The G.711 (PCM Encoding) Library does not have *Create*, *Init*, *Control* and *Destroy* functions, as it is a front-end module which interfaces directly with the codec. Only input and output buffers are required as the part of the argument to the *log PCM Encoder* function call.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.1 *linear2alaw*

Call(s):

```
Result linear2alaw (Int16 *pPCM_values,
                    unsigned char *pA_values,
                    UInt16 NumSamples);
```

Required Header: *g711.h*

Arguments:

Table 3-1. *linear2alaw* Arguments

pPCM_values	*in*	Pointer to the user data signal 16-bit linear sample to be used by the *linear2alaw* algorithm
pA_values	*out*	Pointer to the output buffer of A-law samples
NumSamples	*in*	The number of data words to be processed

Description: The *linear2alaw* function will process linear samples and convert them to A-law encoded samples. Once the processing is complete, the result is returned to the user through a buffer, pointed to by *pA_values*. The user can call the *linear2alaw* function any number of times, as long as there is data.

Returns: Upon successful completion, *linear2alaw* will return “PASS”.

Special Considerations:

- In-place computation is allowed, i.e., input and output buffers could be identical

Code Example 3-2. Use of the *linear2alaw* Interface

```
#include “g711.h”

void test_g711 (void)
{
     Int16 pPCM_values[350];
     unsigned char pA_values[350];
     res = linear2alaw (pPCM_values, pA_values, 13);

     ...

     res = linear2alaw (pPCM_values, pA_values, 350);

     ...

}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.2 *alaw2linear*

Call(s):

```
Result alaw2linear (unsigned char *pA_values,
                    Int16 *pPCM_values,
                    UInt16 NumSamples);
```

Required Header: *g711.h*

Arguments:

Table 3-2. *alaw2linear* Arguments

pA_values	*in*	Pointer to the user data signal 16-bit linear sample to be used by the *alaw2linear* algorithm
pPCM_values	*out*	Pointer to the output buffer of PCM samples
NumSamples	*in*	The number of data bytes to be processed

Description: The *alaw2linear* function will process the A-law samples and convert them to PCM samples. Once the processing is complete , the result is returned to the user through a buffer, pointed to by *pPCM_values*. The user can call the *alaw2linear* function any number of times, as long as there is data.

Returns: Upon successful completion, *alaw2linear* will return “PASS”.

Special Considerations:

- In-place computation is allowed; i.e., input and output buffers could be identical

Code Example 3-3. Use of the *alaw2linear* Interface

```
#include “g711.h”

void test_g711 (void)
{

     Int16 pPCM_values[350];
     unsigned char pA_values[350];
     ...

     res = alaw2linear (pA_values, pPCM_values, 13);

     ...

     res = alaw2linear (pA_values, pPCM_values, 350);

     ...

}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.3 *linear2ulaw*

Call(s):

```
Result linear2ulaw (Int16 *pPCM_values,
                    unsigned char *pU_values,
                    UInt16 NumSamples);
```

Required Header: *g711.h*

Arguments:

Table 3-3. *linear2ulaw* Arguments

pPCM_values	*in*	Pointer to the user data signal 16-bit linear sample to be used by the *linear2ulaw* algorithm
pU_values	*out*	Pointer to the output buffer of μ-law samples
NumSamples	*in*	The number of data words to be processed

Description: The *linear2ulaw* function will process the linear samples and convert them to μ-law encoded samples. Once the processing is complete, the result is returned to the user through a buffer, pointed to by *pU_values*. The user can call the *linear2ulaw* function any number of times, as long as there is data.

Returns: Upon successful completion, *linear2ulaw* will return “PASS”.

Special Considerations:

- In-place computation is allowed; i.e., input and output buffers could be identical

Code Example 3-4. Use of *linear2ulaw* Interface

```
#include “g711.h”

void test_g711 (void)
{

     Int16 pPCM_values[350];
     unsigned char pU_values[350];
     ...

     res = linear2ulaw (pPCM_values, pU_values, 13);

     ...

     res = linear2ulaw (pPCM_values, pU_values, 350);

     ...

}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.4 *ulaw2linear*

Call(s):

```
Result ulaw2linear (unsigned char *pU_values,
                    Int16 *pPCM_values,
                    UInt16 NumSamples);
```

Required Header: *g711.h*

Arguments:

Table 3-4. *ulaw2linear* Arguments

pU_values	*in*	Pointer to the user data signal μ-law encoded samples to be used by the *ulaw2linear* algorithm
pPCM_values	*out*	Pointer to the output buffer of μ-law samples
NumSamples	*in*	The number of data words to be processed

Description: The *ulaw2linear* function will process the μ-law encoded samples and convert them to linear samples. Once the processing is complete, the result is returned to the user through a buffer, pointed to by *pPCM_values*. The user can call the *ulaw2linear* function any number of times, as long as there is data.

Returns: Upon successful completion, *ulaw2linear* will return “PASS”.

Special Considerations:

- In-place computation is allowed; i.e., input and output buffers could be identical

Code Example 3-5. Use of *ulaw2linear* Interface

```
#include “g711.h”

void test_g711 (void)
{

     Int16 pPCM_values[350];
     unsigned char pU_values[350];
     ...

     res = ulaw2linear (pU_values, pPCM_values, 13);

     ...

     res = ulaw2linear (pU_values, pPCM_values, 350);

     ...

}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.5 *alaw2ulaw*

Call(s):

```
Result alaw2ulaw (unsigned char *pAval,
                  unsigned char *pUval,
                  UInt16 NumSamples);
```

Required Header: *g711.h*

Arguments:

Table 3-5. *alaw2ulaw* Arguments

pAval	*in*	Pointer to the user data signal A-law encoded samples to be used by the *alaw2ulaw* algorithm
pUval	*out*	Pointer to the output buffer of μ-law samples
NumSamples	*in*	The number of data words to be processed

Description: The *alaw2ulaw* function will process A-law encoded samples and convert them to μ-law encoded samples. Once the processing is complete, the result is returned to the user through a buffer, pointed to by *pUval*. The user can call the *alaw2ulaw* function any number of times, as long as there is data.

Returns: Upon successful completion, *alaw2ulaw* will return “PASS”.

Special Considerations:

- In-place computation is allowed; i.e., input and output buffers could be identical

Code Example 3-6. Use of *alaw2ulaw* Interface

```
#include “g711.h”

void test_g711 (void)
{

     unsigned char pUval[350];
     unsigned char pAval[350];
     ...

     res = alaw2ulaw (pAval, pUval, 13);

     ...

     res = alaw2ulaw (pAval, pUval, 350);

     ...

}
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

3.3.6 *ulaw2alaw*

Call(s):

```
Result ulaw2alaw (unsigned char *pUval,
                  unsigned char *pAval,
                  UInt16 NumSamples);
```

Required Header: *g711.h*

Arguments:

Table 3-6. *ulaw2alaw* Arguments

pUval	*in*	Pointer to the user data signal μ-law encoded samples to be used by the *ulaw2alaw* algorithm
pAval	*out*	Pointer to the output buffer of A-law samples
NumSamples	*in*	The number of data words to be processed

Description: The *ulaw2alaw* function will process the μ-law encoded samples and convert them to A-law encoded samples. Once the processing is complete, the result is returned to the user through a buffer, pointed to by *pAval*. The user can call the *ulaw2alaw* function any number of times, as long as there is data.

Returns: Upon successful completion, *ulaw2alaw* will return “PASS”.

Special Considerations:

- In-place computation is allowed; i.e., input and output buffers could be identical

Code Example 3-7. Use of *ulaw2alaw* Interface

```
#include “g711.h”

void test_g711 (void)
{

     unsigned char pUval[350];
     unsigned char pAval[350];
     ...

     res = ulaw2alaw (pUval, pAval, 13);

     ...

     res = ulaw2alaw (pUval, pAval, 350);

     ...

}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 4
Building the G.711 Library

4.1 Building the G.711 Library

The G.711 library combines all of the components described in previous sections into one library: *g711.lib*. To build this library, a Metrowerks' CodeWarrior project, *g711.mcp*, is provided. This project and all the necessary components to build the G.711 library are located in the ...*nos**telephony**g711* directory of the SDK directory structure.

There are two methods to execute system library project build: dependency build and direct build.

4.1.1 Dependency Build

Dependency build is the easiest approach and requires no additional work on the user's part. If you add the G.711 library project, ***g711.mcp,*** to your application project as shown in **Figure 4-1**, the G.711 library will automatically build when the application is built.

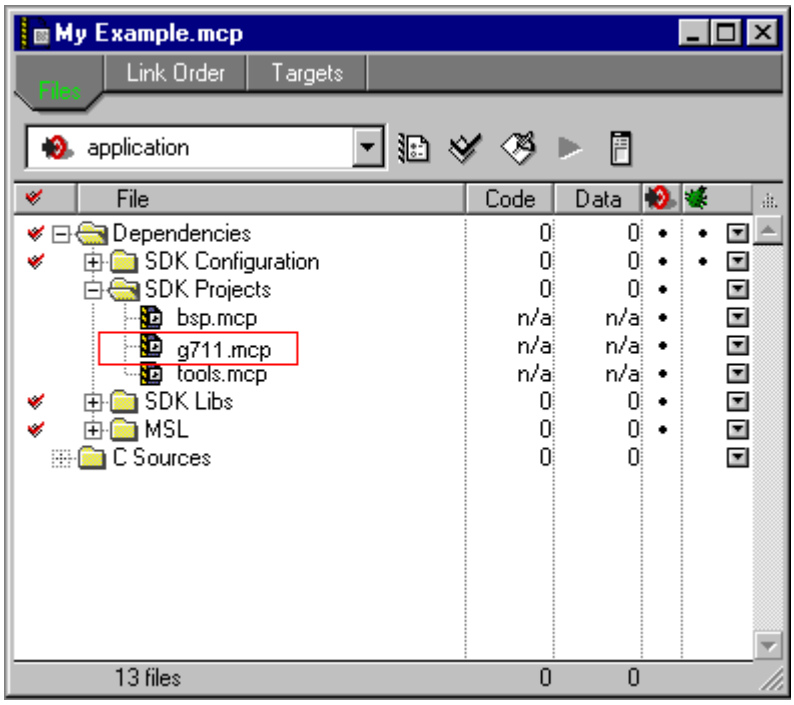

Figure 4-1. Dependency Build for G.711 Library

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

4.1.2 Direct Build

Direct build allows you to build a G.711 library independently of any other build. To do this:

Step 1. Open the *g711.mcp* project, as shown in **Figure 4-2**.

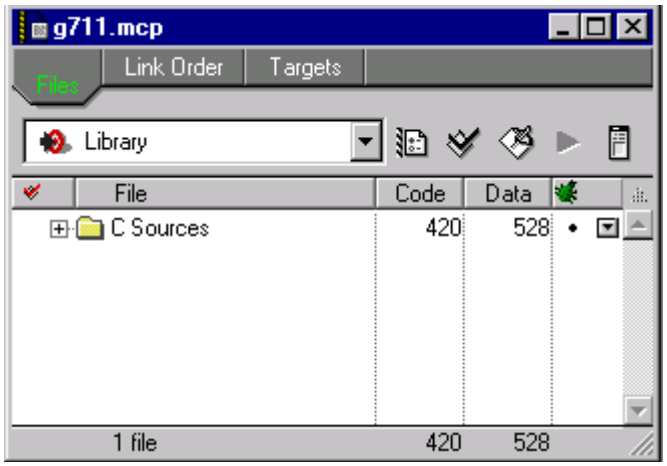

Figure 4-2. *g711.mcp* Project

Step 2. Execute the build by pressing function key [F7] or by choosing the *Make* command from the Project menu; see **Figure 4-3**.

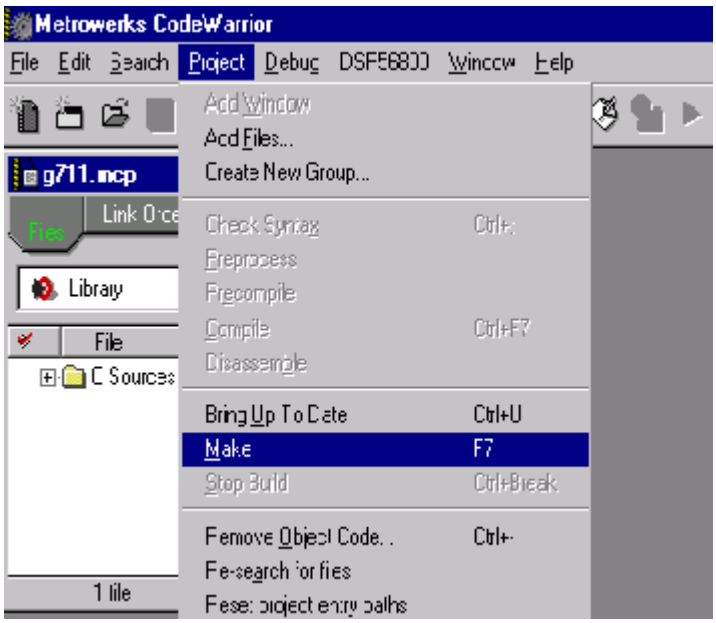

Figure 4-3. Execute Make

At this point, if the build is successful, a *g711.lib* library file is created in the ***...\nos\telephony\g711\Debug*** directory.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 5
Linking Applications with the G.711 Library

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

5.1 G.711 Library

The G.711 library does not contain *Create*, *Init*, *Destroy* or *Control* APIs, because it is a front-end module which directly interfaces with the codec. Only input and output buffers are required as a part of the functions argument. A possible application of the G.711 library could be as a front-end interface to the G.726 ADPCM Encoder.

5.1.1 Library Sections

Because the complete code in the G.711 library is in C language, there are no SECTIONS to be included in the *linker.cmd* for linking. The default file, *linker.cmd*, provided by Metrowerks' CodeWarrior Debug Application, should be used when writing a debug application only with *g711.mcp*.

Please see the *linker.cmd* example file in the *...telephony\g711\test_g711\Config* directory, found in the Software Development Kit, (SDK). A sample *linker.cmd* file is included in **Code Example 5-1** for reference.

Code Example 5-1. Sample *linker.cmd* File

```
# Linker.cmd file for DSP56824EVM External RAM
# using both internal and external data memory (EX = 0)
# and using external program memory (Mode = 3)

MEMORY {

      .pram   (RWX) : ORIGIN = 0x0000, LENGTH = 0xFF80  # ? external program memory
      .avail  (RW)  : ORIGIN = 0x0000, LENGTH = 0x0030  # available
      .cwregs (RW)  : ORIGIN = 0x0030, LENGTH = 0x0010  # C temp registrs in
                                 CodeWarrior
      .im1    (RW)  : ORIGIN = 0x0040, LENGTH = 0x07C0  # data 1
      .rom    (R)   : ORIGIN = 0x0800, LENGTH = 0x0800  # internal data ROM
      .im2    (RW)  : ORIGIN = 0x1000, LENGTH = 0x0600  # data 2
      .hole   (R)   : ORIGIN = 0x1600, LENGTH = 0x0A00  # hole
```


For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
        .data   (RW)  : ORIGIN = 0x2000, LENGTH = 0xC000  # data segment
        .em     (RW)  : ORIGIN = 0xE000, LENGTH = 0x1000  # data 3
        .stack  (RW)  : ORIGIN = 0xF000, LENGTH = 0x0F80  # stack
        .onchip1(RW)  : ORIGIN = 0xFF80, LENGTH = 0x0040  # on-chip peripheral
                                  registers
        .onchip2(RW)  : ORIGIN = 0xFFC0, LENGTH = 0x0040  # on-chip peripheral
                                  registers

}


FORCE_ACTIVE {FconfigInterruptVector}

SECTIONS {

                #
                # Data (X) Memory Layout
                #
                        _EX_BIT     = 0;

                        # Internal Memory Partitions (for mem.h partitions)
                        _NUM_IM_PARTITIONS = 2;  # .im1 and .im2

                        # External Memory Partition (for mem.h partitions)
                        _NUM_EM_PARTITIONS = 1;   # .em


                .main_application_code :
                {
                        # .text sections

                        #  config.c MUST be placed first, otherwise the Interrupt Vector
                        #  configInterruptVector will not be located at the correct
                              address, P:0x0000

                        config.c (.text)
                        * (.text)
                        * (rtlib.text)
                        * (fp_engine.text)
                        * (user.text)

                } > .pram


                .main_application_data :
                {
                        #
                        # Define variables for C initialization code
                        #
                        F_Xdata_start_addr_in_ROM = ADDR(.rom) + SIZEOF(.rom) / 2;
                        F_StackAddr               = ADDR(.stack);
                        F_StackEndAddr            = ADDR(.stack) + SIZEOF(.stack) / 2  - 1;

                        F_Xdata_start_addr_in_RAM = .;

                        #
```


For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
                # Memory layout data for SDK INCLUDE_MEMORY (mem.h) support
                #

                FmemEXbit = .;
                      WRITEH(_EX_BIT);
                FmemNumIMpartitions = .;
                      WRITEH(_NUM_IM_PARTITIONS);
                FmemNumEMpartitions = .;
                      WRITEH(_NUM_EM_PARTITIONS);
                FmemIMpartitionList = .;
                #     WRITEH(ADDR(.im1));
                #     WRITEH(SIZEOF(.im1) / 2);
                      WRITEH(ADDR(.im2));
                      WRITEH(SIZEOF(.im2) / 2);
                FmemEMpartitionList = .;
                      WRITEH(ADDR(.em));
                      WRITEH(SIZEOF(.em) /2);

                # .data sections

                * (.data)
                * (fp_state.data)
                * (rtlib.data)

                F_Xdata_ROMtoRAM_length = 0;

                F_bss_start_addr = .;
                _BSS_ADDR = .;

               * (rtlib.bss.lo)
                * (.bss)

                F_bss_length = . - _BSS_ADDR;  # Copy DATA


        } > .data

        FArchIO   = ADDR(.onchip2);

}
```

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 6
G.711 Applications

6.1 Test and Demo Applications

To verify the G.711 algorithm, test and demo applications have been developed. Refer to the **Targeting Motorola DSP568xx Platform** Manual for the DSP you are using to see if the test and demo applications are available for your target.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Chapter 7
License

7.1 Limited Use License Agreement

LIMITED USE LICENSE AGREEMENT

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THIS SOFTWARE. BY USING OR COPYING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT.

The software in either source code form ("Source") or object code form ("Object") (cumulatively hereinafter "Software") is provided under a license agreement ("Agreement") as described herein. Any use of the Software including copying, modifying, or installing the Software so that it is usable by or accessible by a central processing unit constitutes acceptance of the terms of the Agreement by the person or persons making such use or, if employed, the employer thereof ("Licensee") and if employed, the person(s) making such use hereby warrants that they have the authority of their employer to enter this license agreement,. If Licensee does not agree with and accept the terms of this Agreement, Licensee must return or destroy any media containing the Software or materials related thereto, and destroy all copies of the Software.

The Software is licensed to Licensee by Motorola Incorporated ("Motorola") for use under the terms of this Agreement. Motorola retains ownership of the Software. Motorola grants only the rights specifically granted in this Agreement and grants no other rights. Title to the Software, all copies thereof and all rights therein, including all rights in any intellectual property including patents, copyrights, and trade secrets applicable thereto, shall remain vested in Motorola.

For the Source, Motorola grants Licensee a personal, non-exclusive, non-assignable, revocable, royalty-free right to use, copy, and make derivatives of the Source solely in a development system environment in order to produce object code solely for operating on a Motorola semiconductor device having a central processing unit ("Derivative Object").

For the Object and Derivative Object, Motorola grants Licensee a personal, non-exclusive, non-assignable, revocable, royalty-free right to copy, use, and distribute the Object and the Derivative Object solely for operating on a Motorola semiconductor device having a central processing unit.

Licensee agrees to: (a) not use, modify, or copy the Software except as expressly provided herein, (b) not distribute, disclose, transfer, sell, assign, rent, lease, or otherwise make available the Software, any derivatives thereof, or this license to a third party except as expressly provided herein, (c) not remove obliterate, or otherwise defeat any copyright, trademark, patent or proprietary notices, related to the Software (d) not in any form export, re-export, resell, ship or divert or cause to be exported, re-exported, resold, shipped, or diverted, directly or indirectly, the Software or a direct product thereof to any country which the United States government or any agency thereof at the time of export or re-export requires an export license or other government approval without first obtaining such license or approval.

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS AND WITHOUT WARRANTY OF ANY KIND INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MOTOROLA BE LIABLE FOR

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ANY LIABILITY OR DAMAGES OF ANY KIND INCLUDING, WITHOUT LIMITATION, DIRECT OR INDIRECT OR INCIDENTAL OR CONSEQUENTIAL OR PUNITIVE DAMAGES OR LOST PROFITS OR LOSS OF USE ARISING FROM USE OF THE SOFTWARE OR THE PRODUCT REGARDLESS OF THE FORM OF ACTION OR THEORY OF LIABILITY (INCLUDING WITHOUT LIMITATION, ACTION IN CONTRACT, NEGLIGENCE, OR PRODUCT LIABILITY) EVEN IF MOTOROLA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THIS DISCLAIMER OF WARRANTY EXTENDS TO LICENSEE OR USERS OF PRODUCTS AND IS IN LIEU OF ALL WARRANTIES WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE.

Motorola does not represent or warrant that the Software is free of infringement of any third party patents, copyrights, trade secrets, or other intellectual property rights or that Motorola has the right to grant the licenses contained herein. Motorola does not represent or warrant that the Software is free of defect, or that it meets any particular requirements or need of the Licensee, or that it conforms to any documentation, or that it meets any standards.

Motorola shall not be responsible to maintain the Software, provide upgrades to the Software, or provide any field service of the Software. Motorola reserves the right to make changes to the Software without further notice to Licensee.

The Software is not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Software could create a situation where personal injury or death may occur. Should Licensee purchase or use the Software for any such unintended or unauthorized application, Licensee shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the Software.

The term of this Agreement is for as long as Licensee uses the Software for its intended purpose and is not in default of any provisions of this Agreement. Motorola may terminate this Agreement if Licensee is in default of any of the terms and conditions of this Agreement.

This Agreement shall be governed by and construed in accordance with the laws of the State of Arizona and can only be modified in a writing signed by both parties. Licensee agrees to jurisdiction and venue in the State of Arizona.

By using, modifying, installing, compiling, or copying the Software, Licensee acknowledges that this Agreement has been read and understood and agrees to be bound by its terms and conditions. Licensee agrees that this Agreement is the complete and exclusive statement of the agreement between Licensee and Motorola and supersedes any earlier proposal or prior arrangement, whether oral or written, and any other communications relative to the subject matter of this Agreement.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Index

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and the Stylized M Logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1–303–675–2140 or 1–800–441–2447

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3–20–1, Minami–Azabu. Minato–ku, Tokyo 106–8573 Japan. 81–3–3440–3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852–26668334

Technical Information Center: 1–800–521–6274

HOME PAGE: http://www.motorola.com/semiconductors/