
i.MX53 SDK

Windows Embedded Compact

7

Multimedia Framework

User's Guide

Document Number: 09-7523-USRGD-ZCH66

Rev. 0.4
05/2011



How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 010 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 that may be provided in Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2010. All rights reserved.

About This Book	v
Audience	v
Organization.....	v
Conventions	vi
References.....	vi
Revision History	vi
Chapter 1 SDK Installation	1
1.1 Installation the BSP.....	1
1.2 Install the SDK.....	2
1.2.1 Check the Installation.....	5
1.2.1.1 Check the SDK folder in BSP Code Tree.....	5
1.2.1.2 Check BSP modification performed by the Installer	6
1.2.1.3 Check workspace modification performed by the Installer	6
1.2.2 Remove an Existing Installed SDK	7
Chapter 2 Build OS Images	8
2.1 Working with Multimedia Components	8
2.1.1 Components in the catalog.....	8
2.1.1.1 Importing the Windows Embedded Compact 7 Catalog	8
2.1.2 Add/Remove components from Catalog.....	9
2.2 Build OS Images	9
2.2.1 Build image in platform builder.....	10
2.2.2 Build image in command line	10
2.3 Build multimedia components in sample workspace.....	10
2.3.1 Build all of the components simultaneously	10
2.3.2 Build individual components	10
Chapter 3 Using the Test Procedure	11
3.1 Audio Decoder Test	11
3.2 Video Decoder Test	11
3.3 Image Decoder Test	11
3.4 MP3 Encoder Test.....	12

3.5	WMA8 Encoder Test	12
3.6	PEQ Audio Post-processing Test.....	14
3.7	VPU Encoder Test	16
3.8	Digital Audio Recorder Test	17
3.9	Digital A/V Recorder to AVI Test.....	17
3.10	TVin Recorder Test.....	19
3.11	G.711 Encoder/Decoder Test.....	20
3.12	G.723.1 Encoder/Decoder Test.....	21
3.13	G.726 Encoder/Decoder Test.....	22
3.14	G.729AB Encoder/Decoder Test	23
3.15	NB_AMR Encoder/Decoder Test.....	24
3.16	WB_AMR Encoder/Decoder Test	25

About This Book

This guide explains how to install, build and execute Windows® Embedded Compact 7 OS images for the Babbage board, using the Freescale i.MX53 EVK Windows Embedded Compact 7 Multimedia Framework Software Development Kit (SDK).

Audience

This document is intended for software, hardware, and system engineers who are planning to use the product and for anyone who wants to understand more about the product.

Organization

This document contains the following chapters.

- Chapter 1 "[SDK Installation](#)" explains how to install/uninstall the SDK.
- Chapter 2 "[Build OS Images](#)" explains how to build Windows Embedded Compact 7 OS images using the BSP.
- Chapter 3 "[Test Procedure](#)" explains how to test multimedia component you installed.

Conventions

This document uses the following conventions:

- Courier* Is used to identify commands, explicit command parameters, code examples, expressions, data types, and directives.
- Italic* Is used for emphasis, to identify new terms, and for replaceable command parameters.

All source code examples are in C.

References

The following documents were referenced to build this document.

1. i.MX53 EVK Windows Embedded Compact 7 User's Guide

Revision History

The following table summarizes revisions to this manual since the previous release (Rev. 1.).

Table 1 Revision History

Doc. Version	Date / Author	Description / Location of Changes
1.0-D01	July-20-2010 / Jun Zhu	Update doc with FSL standard document template
1.0-D02	Sep-16-2010 / Fang Hui	Add speech test, image test, a/v recorder test
1.1	May-25-2011/Jun Zhu	Add TVin recorder test for MX53 ARD

Chapter 1

SDK Installation

The Freescale Multimedia Framework Software Develop Kit (SDK) is a collection of binary, code, and support files that you can use to create Windows Embedded Compact 7 OS images for the i.MX53 Babbage board. The SDK is distributed as a single archive EXE (.exe) file.

1.1 Installation the BSP

To install the BSP, use these steps:

1. Confirm that the BSP version for EVK.
2. Follow the steps in the Chapter 1 of *i.MX53 EVK Windows Embedded Compact 7 User's Guide*.

1.2 Install the SDK

To install the SDK, use these steps:

NOTE

Before installing the SDK, save any modified files and close the sample workspace, because the Installer will modify the file of the sample workspace project.

1. Run the installation execution package.

The Setup Wizard screen is displayed (Figure 1-1).



Figure 1-1 Setup Screen

2. Click **Next**.

The Choose Components Screen is displayed (Figure 1-2).

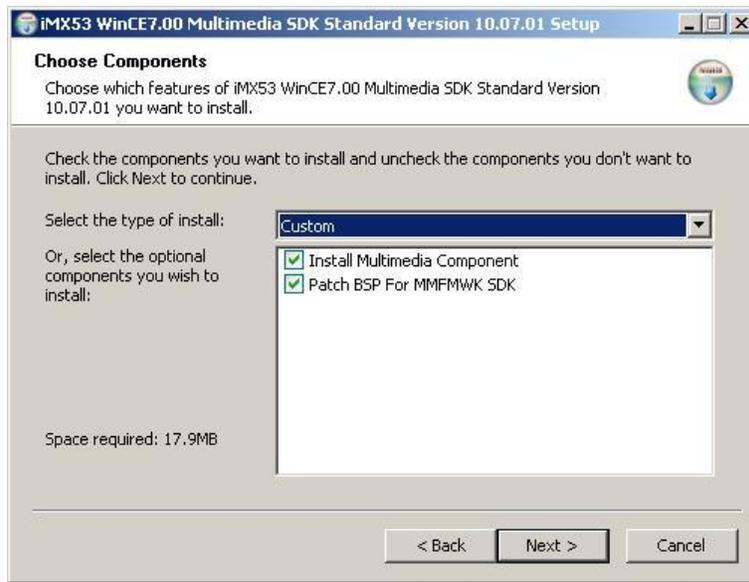


Figure 1-2 Component Choosing Screen

3. Click **Next**.

The installation location screen is displayed (Figure 1-3).

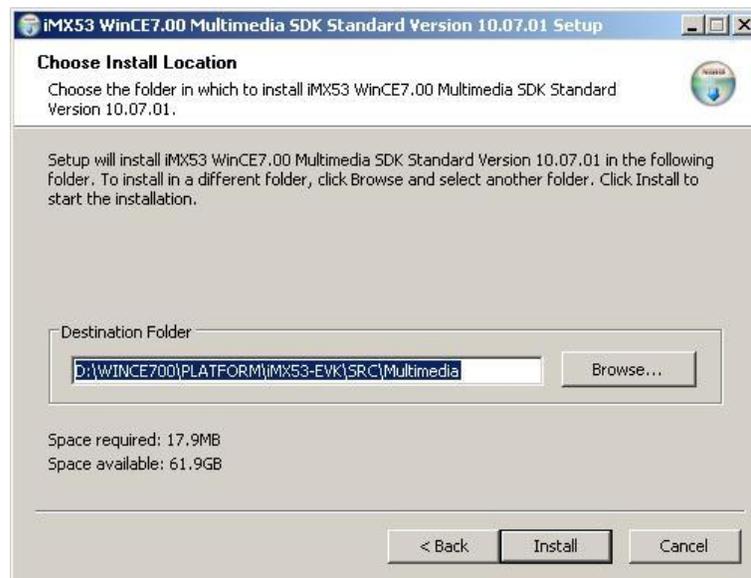


Figure 1-3 Install Location Choosing Screen

4. Under the Destination Folder, set the path of destination folder for SDK installation. By default, multimedia framework SDK will be installed at the following path:

`$(_WINCEROOT) \Platform \<platform_name> \src \Multimedia`

Where:

- `$(_WINCEROOT)` is the path of your Windows Embedded Compact root folder – “WINCE700”
 - `<platform_name>` is the name of the Babbage platform BSP directory – “iMX53-EVK” for i.MX53 Babbage Windows Embedded Compact 7 BSP
5. Continue with the installation wizard until the installation is complete.
 6. In the final installation window (Figure 1-4), you can select whether to do a clean build for the Windows Embedded Compact 7 OS image automatically.
 - If **Make Run-time Image (Nk.bin)** is selected, the entire image build procedure, which includes the CSP build, BSP build and SYSGEN will be implemented automatically.
 - During the automatic build, all of the FSL developed multimedia components are included in the OS Image by default.

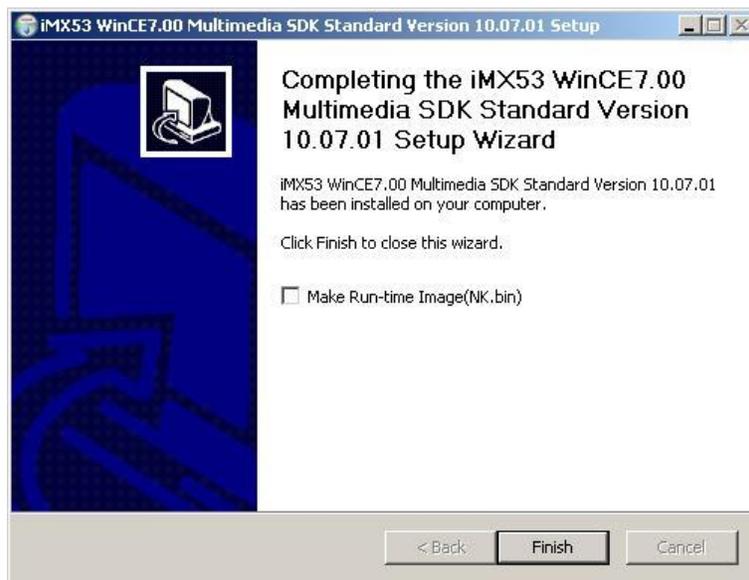


Figure 1-4 Setup Complete Screen

1.2.1 Check the Installation

This section explains how to ensure that the multimedia framework SDK was installed successfully. The installer copies the Multimedia SDK folder into BSP code tree:

- For Windows Embedded Compact 7, the folder is copied to
`\WINCE700\platform\<platform_name>\src`

1.2.1.1 Check the SDK folder in BSP Code Tree

The SDK code tree architecture is shown in Figure 1-5. The table that follows describes the folder content:

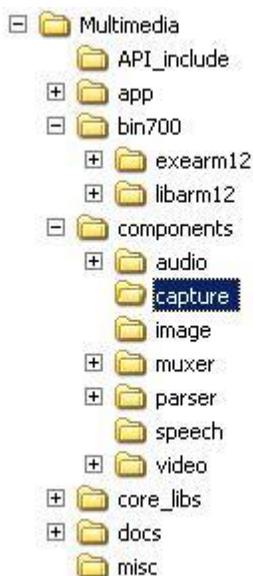


Figure 1-5 SDK Code Tree Screen

Table 2 SDK Code Tree Folder Content

Folder	Contents
API_Include	This folder contains the global header files of multimedia DirectShow DMO and Filters.
components	This folder, with its audio, video, parser and image subfolders, contains DirectShow DMO and Filters' source code of multimedia components.
core_libs	This folder contains header files and lib binaries of multimedia components
docs	This folder contains API document, Datasheet and release notes of all of the components
misc	This folder contains the *.reg and *.bib file required by platform builder
Bin700	This folder contains DLL and EXE binaries, which are built by the components' source code

NOTE

In i.MX53 SDK Windows Embedded Compact 7 multimedia framework, the Demo version may not contain any source code, core libraries and codec documents.

1.2.1.2 Check BSP modification performed by the Installer

The SDK installer modifies the BSP to support Freescale multimedia components.

To check the BSP modifications, use these steps:

1. In the **platform.reg** file under `$(_WINCEROOT)\platform\<platform>\files\`, check the following line was added:

```
#include "$(_TARGETPLATROOT)\src\Multimedia\misc\fs1mm_mx53.reg"
```

2. In the **platform.bib** file under `$(_WINCEROOT)\platform\<platform>\files\`, check the following line was added:

```
#include "$(_TARGETPLATROOT)\src\Multimedia\misc\fs1mm_mx53.bib"
```

1.2.1.3 Check workspace modification performed by the Installer

The SDK installer modifies the Platform Builder's workspace to support Freescale multimedia components.

To check the workspace, use these steps:

1. In Platform Builder IDE, select menu **Project -> iMX53-EVK-Mobility Properties -> Configuration Properties -> Environment**, check the variables were added, as indicated
 - Added – variable **TGTARM** has been set to **arm12**
 - Added – variable **BSP_WATERMARK** has been set to **1**
 - Added – variable **FSL_VIDEO_RENDER** has been set to **1**
2. In **Catalog Items** View, check that the following DirectShow system components of the OS Build were added or removed, as indicated
 - Added - the component **Core OS -> Windows Embedded Compact -> Graphics and Multimedia Technologies -> Media -> WMA and Mp3 Local Playback**
 - Added - the component **Core OS -> Windows Embedded Compact -> Graphics and Multimedia Technologies -> Media -> WMA and Mp3 Local Streaming**
 - Added - the component in **Core OS -> Windows Embedded Compact -> Graphics and Multimedia Technologies -> Media -> DirectShow-> DirectShow Core**
 - Added - the component in **Core OS -> Windows Embedded Compact -> Graphics and Multimedia Technologies -> Media -> DirectShow-> Local File Streamer**

-
- Added - the component **Core OS -> Windows Embedded Compact -> Graphics and Multimedia Technologies -> Media -> Audio Codecs and Renderers -> Wave/AIFF/au/snd File Parser**

1.2.2 Remove an Existing Installed SDK

This section explains how to remove an installation of the SDK from the Windows Embedded Compact 7 source code tree and Platform Builder development environment.

NOTE

Before uninstalling the SDK, save any modified files that you want to keep to a protected location, because uninstalling the SDK will remove all files that were populated by the Installer.

To remove the SDK, use these steps:

1. Close Platform Builder.
2. Click **Start -> Settings -> Control Panel -> Add or remove Programs**. The Add or remove Programs dialog is displayed.
3. Remove the SDK.
4. Manually remove the renaming SDK files and directories -
`$(_WINCEROOT) \Platform \<platform_name> \src \Multimedia`

Chapter 2

Build OS Images

After complete the SDK installation, you can use sample workspace to build a Windows Embedded Compact 7 OS Image based on installed SDK. And you also can add or remove the components to the image build.

2.1 Working with Multimedia Components

The multimedia components are located in the Windows Catalog. You import the catalog, and can then add the components to the OS Image build.

2.1.1 Components in the catalog

After the SDK has been installed, you can find the multimedia components in the window Catalog.

2.1.1.1 Importing the Windows Embedded Compact 7 Catalog

For Windows Embedded Compact 7, the multimedia SDK catalog is automatically imported when the sample workspace is opened.

To view the catalog, use these steps:

1. Click **View -> Other Windows -> Catalog items view**

The Catalog Items View dialog is displayed (Figure 2-1).

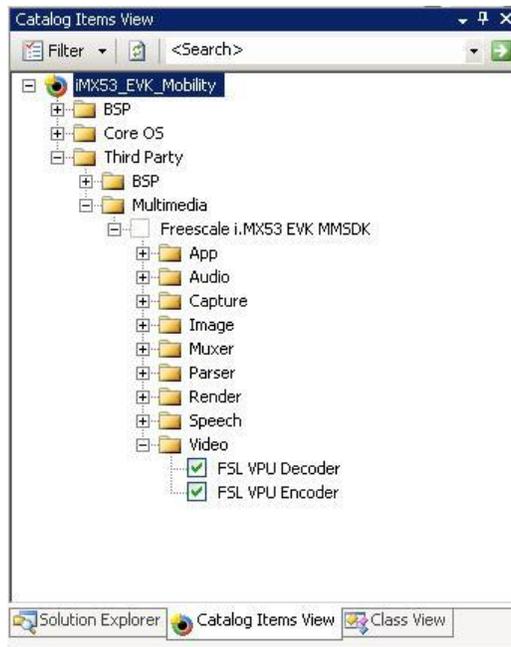


Figure 2-1 SDK Catalog View

2. To view the multimedia components, open **Third Party -> Multimedia -> Freescale i.MX53 EVK MMSDK**

If can not find these components in catalog items view, it need exit and restart the Platform Builder to make multimedia catalog imported completely.

2.1.2 Add/Remove components from Catalog

To add/remove components from the catalog, use these steps:

1. Click **View > Other Windows > Catalog Items View**.
2. To view the multimedia components, open **Third Party > Multimedia > Freescale i.MX53 EVK MMSDK**.
3. To add/remove a multimedia component, just select/unselect the associated check box.

2.2 Build OS Images

This section explains how to build Windows Embedded Compact 7 OS image included multimedia components in the sample workspace.

2.2.1 Build image in platform builder

For instructions for building OS images using Platform Builder, see the *i.MX53EVK Windows Embedded Compact 7 User's Guide*.

2.2.2 Build image in command line

The SDK installer provides the `fsl_mmfwk_build.bat` script, which builds the OS image. You can run this script during the last installation step or in command line after the installation.

To run the script in command line, follow these steps:

1. Open the command console on Desktop PC
2. Go to the `$(_WINCEROOT)\platform\ directory`
3. Run `fsl_mmfwk_build.bat`

2.3 Build multimedia components in sample workspace

This section explains how to build/rebuild installed multimedia components after the SDK upgrading or the modification on multimedia components' source code

2.3.1 Build all of the components simultaneously

To build the all of the installed components at once, use these steps:

1. In the project window, open the **Multimedia** project in the folder **Platform -> iMX53-EVK -> SRC**
2. Right click on the **Multimedia** project and select **Rebuild**.

This builds all of DLL and EXE binaries in **bin700** directories under the **Multimedia** folder in BSP code tree.

2.3.2 Build individual components

To build the individual component, use these steps:

1. In the project window, open the individual component subproject in the **Multimedia** project
2. Right click an individual component's sub-project and select **Rebuild**,

This builds the DLL and EXE binaries in **bin700** directories under the **Multimedia** folder in the BSP code tree.

Chapter 3

Using the Test Procedure

This chapter explains how to test Freescale multimedia components in the Windows Embedded Compact 7 OS, which image was built as described in chapter 2.

3.1 Audio Decoder Test

To perform these tests, use Windows Embedded Compact Music Player to playback the audio files.

3.2 Video Decoder Test

To perform these tests, use Windows Embedded Compact Video Player to playback the video files.

3.3 Image Decoder Test

To test image decoders, use test application that are built into the Windows Embedded Compact 7 OS image when the decoder component is added from Catalog during the image build procedure. Double-click the image file to display the image.

Supported image file extensions:

- *.bmp for BMP decoder
- *.gif for GIF decoder
- *.png for PNG decoder
- *.jpg for JPEG decoder

Image display is also supported by command line. To display an image, run the associated command:

Table 3 Command for Image Display

Image Type	Command
BMP	fsl_bmp_img_lib_test.exe <image_file_name>
GIF	fsl_gif_img_lib_test.exe <image_file_name>

PNG	fsl_png_img_lib_test.exe <image_file_name>
JPEG	fsl_jpeg_img_lib_test.exe <image_file_name>

3.4 MP3 Encoder Test

The test application for MP3 encoder is built into the Windows Embedded Compact 7 OS image, when the MP3 encoder component is added from Catalog during the image build procedure. The test application supports the WAV file input.

To execute the test bench, run the following at command line:

fsl_mp3_enc_dmo_test.exe <input file> <output file> <Encoder Configuration>

The parameter **<Encoder Configuration>** is optional, and describes as follows:

[-b <bps>] [-s <samplerate>]

Table 4 MP3 Encoder Configuration

Arguments	Description
-b <bps>	Bit rate for the encoded mp3 bit stream (default 128000), bit/second in Unit. The available values can be: 32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 320000
-s <samplerate>	Sample rate for the encoded mp3 bit stream (default 44100), sample/second in Unit The available values can be 32000, 44100, 48000

Refer to the following command line for information.

fsl_mp3_enc_dmo_test.exe pcm.wav output.mp3 -b 128000 -s 44100

NOTE

The current test application supports the WAV file input. The sample rate of an input WAV should be the same as “**-s <samplerate>**”. If the two rates are not identical, the test application stop the encode procedure and exit.

3.5 WMA8 Encoder Test

Test Application for WMA8 audio encoder is built into the Windows Embedded Compact 7 OS image, when the encoder component is added from Catalog during the image build procedure.

To execute the test, run the following at command line:

fsl_wma8_enc_test.exe <input file> <output file> <Encoder Configuration>

The parameters for Encoder Configuration are optional, and described as follows:

[<-a> <AUTHOR>] [<-b> <BITRATE>] [<-c> <COPYRIGHT>]

[<-d> <DISCRIPTION>] [<-T> <TITLE>]

The meaning for each parameter is as follow:

Table 5 WMA8 Encoder Configuration

Arguments	Description
<-a> <AUTHOR>	Specify an array of WCHARs that contains the author information in ASF header.
<-b> <BITRATE>	Bit rate for the encoded wma bit stream (default 70000). Any value ranging from 20k (bps) to 211.2k(bps).
<-c> <COPYRIGHT>	Specifies an array of WCHARs that contains the copyright information in ASF header
<-d> <DISCRIPTION>	Specifies an array of WCHARs that contains the description information in ASF header
<-r> <RATING>	Specifies an array of WCHARs that contains the rating information in ASF header
<-T> <TITLE>	Specifies an array of WCHARs that contains the title information in ASF header
<-h>	Get the usage of this test bench.

The following table (Table 6) describes the valid configuration combination for WMA8 Encoder.

Table 6 WMA8 Encoder's Valid Configuration Combination

Channels	Sample Rate (Hz)	Bit Rate (bps)
Mono	22050	22000, 16000, 22000
	32000	22000, 20000
	44100	52800, 48000, 35200, 32000
Stereo	22500	35200, 32000, 22000, 20000
	32000	52800, 48000, 44000, 40000, 35200
	44100	221200, 192000, 176000, 160000, 140800, 128000, 105600, 96000, 88000, 70400, 64000
	48000	211200, 192000, 176000, 160000, 140800, 128000

Refer to the following command line for information.

```
fsl_wma8_enc_test.exe pcm.wav output.wma -b 128000 -d "Freescale test music"
```

3.6 PEQ Audio Post-processing Test

The test applications for PEQ audio post-processing **fsl_peq_ppp_app_forwmp.exe**, will be built into the Windows Embedded Compact 7 OS image, when the post-processing component is added from Catalog during the image build procedure.

It is a graphic user interface for users to switch among PEQ profiles and control volume dynamically while playback the audio file with Windows Embedded Compact Media Player.

IMPORTANCE

Before using WMP to playback the audio files, make sure that test application for PEQ is running. Otherwise, PEQ audio post-processing will not function.

To use test application for PEQ testing, follow these steps:

1. To execute the test application, double-click the executable file or run the following at command line:

```
fsl_peq_ppp_app_forwmp.exe
```

Then the application window will pop up (Figure 3-1).

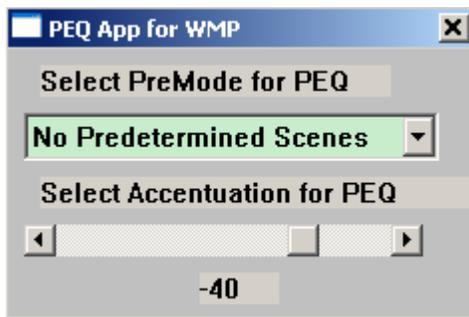


Figure 3-1 PEQ Setting

2. Use Windows Embedded Compact Media Player to playback an audio file. And keep the test application running when audio playback
3. During the playback periods, you can select different “**Premode**” dynamically through Profile List. Pre-defined profiles are defined as Table 7.

Table 7 PEQ Profiles

Index	Profile Name
0	Do not select predetermined scenes. (default)
1	Acoustic
2	Bass booster
3	Bass reducer
4	Classical
5	Dance
6	Deep
7	Electronic
8	hip hop
9	Jazz
10	Latin
11	Loudness
12	Lounge
13	Piano
14	Pop
15	R&B
16	Rock
17	small speakers
18	spoken word
19	treble booster
20	treble reducer
21	vocal booster
22	Flat

4. Use **Attenuation** scrollbar to control the volume.

NOTE

If the value for attenuation is set too high, the audio output would not be pleasing. It is strongly recommended that the value of attenuation is set to below -30.

3.7 VPU Encoder Test

The package contains the Test Bench for VPU Encoder filter, which supports encode the raw IYUV (YUV420p) format data file into H.264/MPEG4/H.263/MJPEG bit stream and store them into AVI container. So tester can check the encode result easily by playback the output AVI file.

Run the following command in “Command Prompt” to execute the test bench:

```
fsl_vpu_enc_dshow_testbench.exe <input file> <output file> <Encoder Configuration>
```

The parameters *<Encoder Configuration>* describes as Table 8.

Table 8 VPU Encoder Configuration

Arguments	Description
<width>	Frame width about the input YUV file
<height>	Frame height about the input YUV file
<bitrate in kbps>	Output bitrate, bit/second in Unit Up to 8 Mbps
<target frame rate>	Output frame rate, frame/second in Unit Up to 30 fps
<ENCODETYPE>	Output bitstream format. It should be one of the value below: <ul style="list-style-type: none">• H264• MPEG4• H263• MJPG
<key frame interval>	Key frame duration for output bitstream

NOTE

In order to keep the reasonable quality, MJPEG compression ratio should be smaller than 20. The compression ratio can be calculated by formula below:

$$compress_ratio = (width * height * frame\ rate * 12) / bit\ rate$$

Refer to the following command line for information.

```
fsl_vpu_enc_dshow_testbench.exe 480_320.yuv output.avi 480 320 800 30 H264 30
```

3.8 Digital Audio Recorder Test

Audio recording test bench uses MSFT's Audio Capture Filter to capture the PCM raw data from Audio Line-In Port and encodes them into MP3 by FSL's MP3 Encoder DMO.

NOTE

Digital Audio Recorder test application may not run successful, if the component "FSL MP3 Encoder" is not selected in the catalog before the WinCE OS Image building.

To execute the test bench, run the following at the command line:

```
fsl_dar_app.exe <Time in sec> <Output file> <Sample Rate> <BitRate>
```

The following table describes the arguments:

Table 9 Digital Audio Recorder Arguments

Arguments	Description
<Time in sec>	Duration for audio capture, second in Unit.
<Output file >	The name of output mp3 file
<Sample Rate >	Sample rate for Audio Line-In, sample/second in Unit The available values can be 32000, 44100, 48000
<BitRate>	Bit rate for the encoded mp3 bit stream, bit/second in Unit. The available values can be 32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 320000

Refer to the following command line for information.

```
fsl_dar_app.exe 10 output.mp3 44100 32000
```

3.9 Digital A/V Recorder to AVI Test

A/V Recorder for AVI Test bench uses MSFT's Audio Capture Filter and FSL Video Capture Filter to capture raw audio data from Line In and raw video data from Camera, then encode the audio and the video to one AVI file with mp3 audio format and h264, mp4v or h263 video format.

NOTE

A/V Recorder for AVI Test application may not run successful, if the component “FSL MP3 Encoder”, “FSL Video Capture”, FSL VPU Encoder”, “FSL AVI Muxer” and “FSL File Writer” are not selected in the catalog before the WinCE OS Image building.

To execute the test bench, run the following at the command line:

fsl_avi_mux_test.exe -av <Time in Sec> <Output File> <Sample Rate> <Audio BitRate> <Width> <Height> <Frame Rate> <Key Frame Rate> <Video BitRate> <Video Encoder Type>

The following table describes the arguments:

Table 10 Digital A/V Recorder Arguments

Arguments	Description
<Time in Sec>	Duration for audio/video capture, second in Unit.
<Output File >	The name of output avi file
<Sample Rate >	Sample rate for Audio Line-In, sample/second in Unit The available values can be 32000, 44100, 48000
<Audio BitRate>	Bit rate for the encoded mp3 bit stream, bit/second in Unit. The available values can be 32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 320000
<Width> <Height>	Width and height for camera capturing window. The available values can be: <ul style="list-style-type: none"> • Width = 160, Height = 120(QQVGA) • Width = 176, Height = 144(QCIF) • Width = 320, Height = 240(QVGA) • Width = 352, Height = 288(CIF) • Width = 640, Height = 480(VGA) • Width = 720, Height = 480(NTSC D1) • Width = 720, Height = 576(PAL D1)
<Frame Rate>	Frame rate for video camera capturing. The only available value should be 15
<Key Frame Rate>	Key frame rate for video encoder setting. The available value should be in the range [2, 256]
<Video BitRate>	Bit rate for the encoded video bit stream, kilo bits per second in Unit. The available value should be in the range [64, 4096]

<Video Encoder Type>	<p>Output video bitstream format.</p> <p>It should be one of the value below:</p> <ul style="list-style-type: none"> • 0 -- H264 • 1 -- MPEG4 • 2 -- H263 • 3 -- MJPG
-----------------------------------	---

Refer to the following command line for information:

```
fsl_avi_mux_test.exe -av 10 av_mp3_44100hz_32kbps_h264_qcif_512kbps.avi 44100 32000 176 144 15 20 512 0
```

3.10 TVin Recorder Test

TVin Recorder Test bench uses FSL Video Capture Filter to capture raw video data from TVin port, and then encode the video as h264, mp4v or h263 video format, and then save encoded stream into AVI file.

NOTE

TVin Recorder Test application may not run successful, if

1. TVin driver does not enabled in MX53 ARD BSP
2. The DirectShow components “FSL Video Capture”, FSL VPU Encoder”, “FSL AVI Muxer” and “FSL File Writer” are not selected in the catalog before the WinCE OS Image building.

To execute the test bench, run the following at the command line:

```
fsl_tv_in_recorder.exe <Time in Sec> <Output File> <Frame Rate> <Key Frame Rate> <Video BitRate> <Video Encoder Type>
```

The following table describes the arguments:

Table 11 TVin Recorder Arguments

Arguments	Description
<Time in Sec>	Duration for audio/video capture, second in Unit.
<Output File >	The name of output avi file
<Frame Rate>	<p>Frame rate for TVin recording.</p> <p>The only available value should be 25 or 30, depends on the TV input mode is PAL – 25; NTSC -- 30</p>

<Key Frame Rate>	Key frame rate for video encoder setting. The available value should be in the range [2, 256]
<Video BitRate>	Bit rate for the encoded video bit stream, kilo bits per second in Unit. The available value should be in the range [64, 4096]
<Video Encoder Type>	Output video bitstream format. It should be one of the value below: <ul style="list-style-type: none"> • 0 -- H264 • 1 -- MPEG4 • 2 -- H263 • 3 -- MJPG

Refer to the following command line for information:

fsl_tv_in_recorder.exe 10 tvin_record.avi 5 25 30 512 0

If TVin mode is PAL, record 10 seconds TV into H.264 512Kbps@25fps video with key frame rate 30fps.

3.11 G.711 Encoder/Decoder Test

G.711 encoder test application is built into the Windows Embedded Compact 7 OS image when the G.711 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g711_enc_test.exe <Options> <InFile> <OutFile>

The following table describes the arguments:

Table 12 G.711 Encoder Configuration

Arguments	Description
Options	<ul style="list-style-type: none"> • A A-law • M Mu-law • A2M convert A-law to Mu-law • M2A convert Mu-law to A-law
InFile	Name of the input file
OutFile	Name of the output file

G.711 decoder test application is built into the Windows Embedded Compact 7 OS image when the G.711 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g711_dec_test.exe <Options> <InFile> <OutFile>

The following table describes the arguments:

Table 13 G.711 Decoder Configuration

Arguments	Description
Options	<ul style="list-style-type: none"> • A A-law • M Mu-law
InFile	Name of the input file
OutFile	Name of the output file

3.12 G.723.1 Encoder/Decoder Test

G.723.1 encoder test application is built into the Windows Embedded Compact 7 OS image when the G.723.1 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g723_enc_test.exe [-Options] <InFile> <OutFile>

The following table describes the arguments:

Table 14 G.723.1 Encoder Configuration

Arguments	Description
Options	<ul style="list-style-type: none"> • -DTX for DTX test vector • -Noh Enable/Disable Hp Filter • -rate # bit-rate (in kbit/s): 63 for 6.3 kbit/s and 53 for 5.3 kbit/s • -?/-help print help message
InFile	Name of the input file
OutFile	Name of the output file

G.723.1 decoder test application is built into the Windows Embedded Compact 7 OS image when the G.723.1 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g723_dec_test.exe [-Options] <InFile> <OutFile>

The following table describes the arguments:

Table 15 G.723.1 Decoder Configuration

Arguments	Description
Options	<ul style="list-style-type: none"> • - Nop Enable/Disable Hp Filter • -?/-help print help message • -f <file> Use file as CRC file
InFile	Name of the input file
OutFile	Name of the output file

3.13 G.726 Encoder/Decoder Test

G.726 encoder test application is built into the Windows Embedded Compact 7 OS image when the G.726 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g726_enc_test.exe [-Options] <InFile> <OutFile>

The following table describes the arguments:

Table 16 G.726 Encoder Configuration

Arguments	Description
Options	<ul style="list-style-type: none"> -law <a u l > The letters A or a for G.711 A-law, letter u for G.711 m-law, or letter l for linear. Default is A-law. -rate # the bit-rate (in kbit/s): 40, 32, 24 or 16. Default is 32 kbit/s. -homing <InitFile> The file contains initialization (homing) sequence to drive the Encoder to a known initial state. Default is no init file and the Codec is in reset state. -?/-help print help message
InFile	Name of the input file
OutFile	Name of the output file

G.726 decoder test application is built into the Windows Embedded Compact 7 OS image when the G.726 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g726_dec_test.exe [-Options] <InFile> <OutFile>

The following table describes the arguments:

Table 17 G.726 Decoder Configuration

Arguments	Description
Options	
-law <a u l >	The letters A or a for G.711 A-law, letter u for G.711 m-law, or letter l for linear. Default is A-law.
-rate #	the bit-rate (in kbit/s): 40, 32, 24 or 16. Default is 32 kbit/s.
-homing <InitFile>	The file contains initialization (homing) sequence to drive the Encoder to a known initial state. Default is no init file and the Codec is in reset state.
-?/-help	print help message
InFile	Name of the input file
OutFile	Name of the output file

3.14 G.729AB Encoder/Decoder Test

G.729AB encoder test application is built into the Windows Embedded Compact 7 OS image when the G.729AB codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g729ab_enc_test.exe <InFile> <OutFile> <VAD_flag>

The following table describes the arguments:

Table 18 g.729ab Encoder Configuration

Arguments	Description
InFile	Name of the input file
OutFile	Name of the output file
VAD_flag	0: disable VAD, 1: enable VAD.

G.729AB decoder test application is built into the Windows Embedded Compact 7 OS image when the G.729AB codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

fsl_g729ab_dec_test.exe <InFile> <OutFile>

The following table describes the arguments:

Table 19 G.729ab Decoder Configuration

Arguments	Description
InFile	Name of the input file
OutFile	Name of the output file

3.15 NB_AMR Encoder/Decoder Test

NB_AMR encoder test application is built into the Windows Embedded Compact 7 OS image when the NB_AMR codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

```
fsl_nb_amr_enc_test.exe [-dtx|dtx2] [-bitstreamformat] [-numframetoencode] <amr_mode> <speech_file> <bitstream_file>
```

The following table describes the arguments:

Table 20 NB_AMR Encoder Configuration

Arguments	Description
-dtx	enables DTX or DTX2 mode
-bitstreamformat	File format, etsi, if1, if2 or mms, default is etsi
-numframetoencode	Number of frame to encode should be between 1 and 255
amr_mode	Specify the bit rate, it should be MR122, MR102, MR795, MR74, MR67, MR59, MR 515, MR475 or <code>-modefile = mode_file</code> (this will read AMR modes from text file)
speech_file	Name of the input file
bitstream_file	Name of the output file

Refer to the following command line for information.

```
fsl_nb_amr_enc_test.exe -dtx1 -if2 -numframe=1 MR122 \release\input\dtx1\DTX1.INP \release\output\if2\DTX1_MR122.COD
```

```
fsl_nb_amr_enc_test.exe -dtx1 -if2 -numframe=1 -modefile=\release\input\dtx1\allmodes.txt \release\input\dtx1\SPEECH.INP \release\output\if2\SPEECH_VAD1.COD
```

NB_AMR decoder test application is built into the Windows Embedded Compact 7 OS image when the NB_AMR codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

```
fsl_nb_amr_dec_test.exe [-bitstreamformat] [-numframetodecode] <encoded_file> <output_file>
```

The following table describes the arguments:

Table 21 NB_AMR Decoder Configuration

Arguments	Description
-bitstreamformat	Specify input file format, etsi, if1, if2 or mms, default is etsi
-numframetodecode	Number of frame to decode per time, default value is 1
encoded_file	Name of the input file
output_file	Name of the output file

Refer to the following command line for information.

```
fsl_nb_amr_dec_test.exe -if2 -numframe=1  

\release\input\if2_ref_vectors\no_dtx\t_mode\T21.COD \release\output\if2\T21.OUT
```

3.16 WB_AMR Encoder/Decoder Test

WB_AMR encoder test application is built into the Windows Embedded Compact 7 OS image when the WB_AMR codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

```
fsl_wb_amr_enc_test.exe [-dtx] [-bitstreamformat] < mode> <speech_file> <bitstream_file>
```

The following table describes the arguments:

Table 22 WB_AMR Encoder Configuration

Arguments	Description
-dtx	-dtx if DTX is ON, default is OFF
-bitstreamformat	include: -itu -mime -if1 -if2, default is -itu
mode	Specify the bit rate, it should be 0,1,2,3,4,5,6,7,8. "mode : (0) (1) (2) (3) (4) (5) (6) (7) (8) " "bitrate: 6.60 8.85 12.65 14.25 15.85 18.25 19.85 23.05 23.85kbit/s "
speech_file	Name of the input file
bitstream_file	Name of the output file

Refer to the following command line for information.

```
fsl_wb_amr_enc_test 0 release/wb_amr/input/T_inp/T00.inp
release/wb_amr/output/T00_660.cod
```

```
fsl_wb_amr_enc_test -dtx 0 release/wb_amr/input/DTX_test_vectors/T_inp/Dtx3.inp
release/wb_amr/output/Dtx3_660.cod
```

WB_AMR decoder test application is built into the Windows Embedded Compact 7 OS image when the WB_AMR codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

```
fsl_wb_amr_dec_test.exe [-bitstreamformat] <encoded_file> <output_file>
```

The following table describes the arguments:

Table 23 WB_AMR Decoder Configuration

Arguments	Description
-bitstreamformat	specify output file format include: -itu -mime -if1 -if2, default is -itu
encoded_file	Name of the input file
output_file	Name of the output file

Refer to the following command line for information.

```
fsl_wb_amr_dec_test.exe release/wb_amr/input/T_660/T00_660.cod
release/wb_amr/output/T00_660.out
```