SIGMATEL®

MIXED-SIGNAL SEMICONDUCTORS

Application Note

SMV PC Transcoding Software

Rev 4.1

TABLE OF CONTENTS

LIST OF FIGURES

1. SCOPE

This document is intended to provide customers with information regarding PC transcoding software development for Motion Video Format in the SigmaTel D-Major Software Development Kit. The document covers information about PC software architecture, development environment and SMV format library API and software customization.

2. PC SOFTWARE ARCHITECTURE

The software architecture is shown in the figure 1.

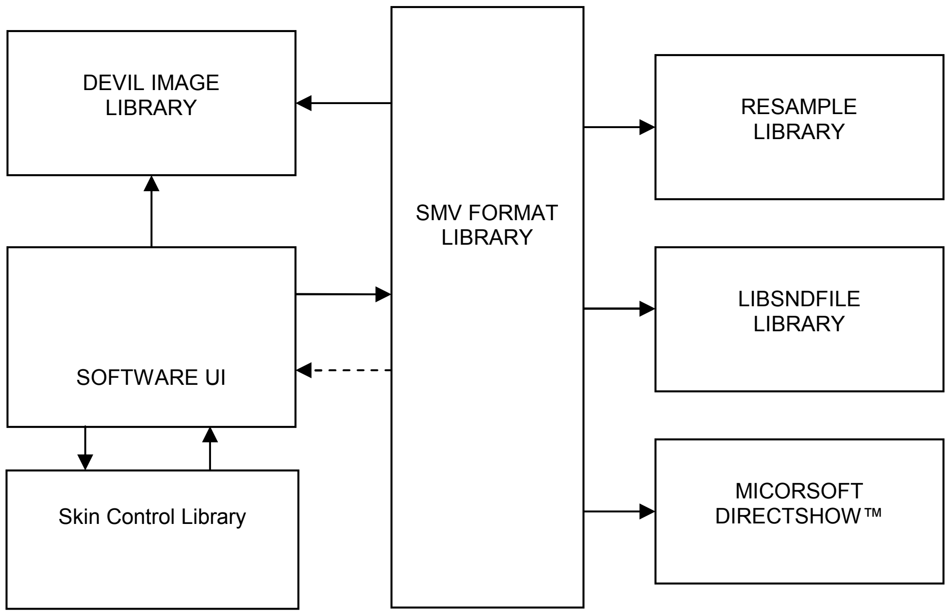


Figure 1: PC Software Architecture

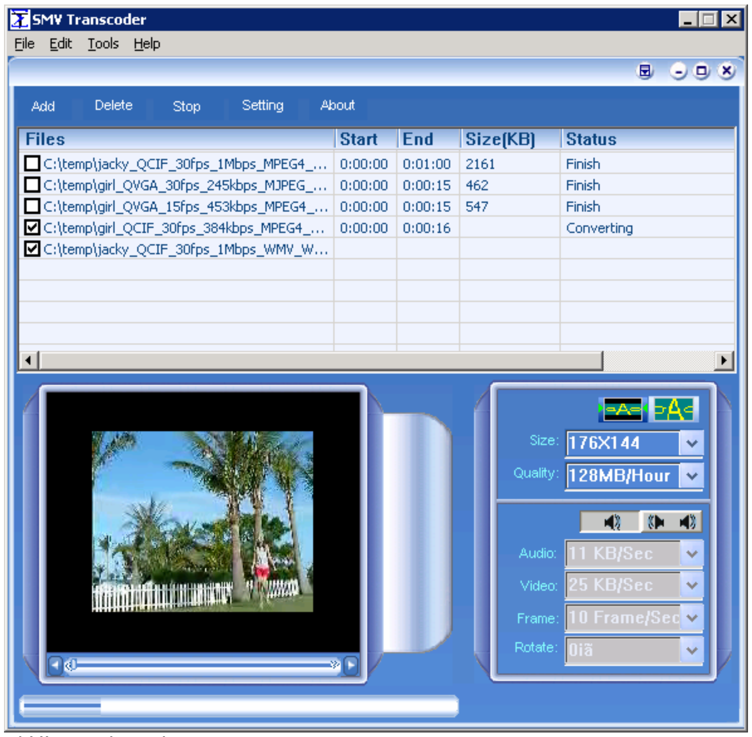
The software UI is an interface for user to specify the input video files, transcoding setting and video partial selection. In this App Note, UI sample code is provided for reference by using Microsoft Foundation Class library. Visual C++.NET 2003 is used to compile the project.

The SMV format library is used to preview and process the input video during transcoding. When an input video file is given, a filter graph in Microsoft DirectShow™ will be created for preview. The offset can be specified to render a particular frame. This feature is useful for video partial selection. Once the transcoding setting and partial selection are confirmed, SMV format library can be used to transcode the video file into SMV file. In the backend, SMV format library utilize libsndfile library, resample library, devil image library and Microsoft DirectShow™ library to achieve video preview and video transcoding.

During transcoding, there is a chance for UI application to retouch video graphic frame one by one. This feature allows further customization.

3. DEVELOPMENT ENVIRONMENT

The software package is designed to use Microsoft Visual C++.NET 2003 in compilation. The Visual C++.NET 2003 solution file is included. Since Microsoft DirectShow™ Library is called from SMV format library, it is not necessary to install Microsoft Platform SDK. There are two compilation configurations defined in the project: debug and release. Debug configuration is used for debugging purpose. The release configurations are used to generate the final executable file for PC transcoding software.



Files	Start	End	Size(KB)	Status
☐ C:\temp\jacky_QCIF_30fps_1Mbps_MPEG4_...	0:00:00	0:01:00	2161	Finish
☐ C:\temp\girl_QVGA_30fps_245kbps_MJPEG_...	0:00:00	0:00:15	462	Finish
☐ C:\temp\girl_QVGA_15fps_453kbps_MPEG4_...	0:00:00	0:00:15	547	Finish
☑ C:\temp\girl_QCIF_30fps_384kbps_MPEG4_...	0:00:00	0:00:16		Converting
☑ C:\temp\jacky_QCIF_30fps_1Mbps_WMV_W...				

Size: 176X144
Quality: 128MB/Hour
Audio: 11 KB/Sec
Video: 25 KB/Sec
Frame: 10 Frame/Sec
Rotate: 0iã

Figure 2: The outlook of compiled UI sample code

4. SMV FORMAT LIBRARY API

In the SMV format library API, there are seven APIs defined for video preview and video processing for SMV format file. The details of these API are described in the following:

LONG NewTranscoder(HWND hWnd, LPTSTR pszSrcFileName, LPTSTR pszDstFileName, LONGLONG pDuration);*

Input:

hWnd – window handle which is used to display the snapshot in range selection and transcoding process

pszSrcFileName – the full path location of the input video file

pszDstFileName – the full patch location of the output smv file. If a empty string is given, a preview windows will be created for video partial selection. Afterwards, Preview() can be used to render a particular frame by using the offset

pDuration – pointer of LONGLONG variable to store the total length of video time in terms of 100 nanoseconds

Output:

Status – Transcoder creation result. Here is the meaning of some value:

0 : success

0x80040265L: SMV_MESSAGE_NO_CODEC

0x8004025CL: SMV_MESSAGE_NO_AUDIO_CODEC

0x8004025DL: SMV_MESSAGE_NO_VIDEO_CODEC

others: result from DirectShow framework

Description:

This API should be the first API call when using SMV format library. The preview and transcoding mode will be decided according to the value of pszDstFileName. If a empty string is specified in the destination, preview mode is used. Otherwise, the transcoder is prepared for smv file generation.

BOOL CloseTranscoder(void);

Description:

This API should be the last API call after finish to use SMV format library.

HBITMAP Preview(LONGLONG offset, HDC dc, SIZE sz);

Input:

Offset – The video offset from the beginning. The unit is 100 nanoseconds

Dc – device context handle to display the snapshot of range selection

sz – the required resolution of the snapshot

Output:

hBitmap – bitmap handle which contains the snapshot of the current position. Application is responsible to free this handle when it is not used anymore.

Description:

SMV format library will capture the frame which has the specified offset from the beginning for preview. You can call this API as much as you can to provide smooth preview during partial selection.

BOOL Transcode(TRANSCODER_PARAM pParam);*

```
typedef struct
{
        HWND hWnd;
        double markA;
        double markB;
        int width;
        int height;
        bool bFullImage;
        int samplingfrquency;
        int nchannel;
        int framerate;
        int compressionlevel;
        int nRotation;
```

int nMirror;

int nPreviewInternal;

*void (*pxImageProcess)(ILint image);*

}

TRANSCODER_PARAM;

Input:

pParam – a parameter structure to specify detailed information for transcoding:

hWnd - The handle of the window which receive WM_UPDATE event for progress update in transcoding.

markA – the starting point of partial selection. It is the video offset from the beginning in the range of 0 to 1

markB – the ending point of partial selection. It is the video offset from the beginning in the range of 0 to 1

width – the width of target screen to playback the SMV file.

height – the height of target screen to playback the SMV file.

bFullImage – Option to specify to use either Full Image or Full Screen. TRUE represents Full Image while FALSE represents Full Screen.

samplingfrequency – the sampling frequency of audio data

nchannel – The number of audio channel to be used. monochrome (1) or stereo (2).

framerate – the target framerate of SMV file from 4 to 15

compressionlevel – average graphic data in kB per second

nRotation – specify the rotation options: 0 – no change, 1 – rotate left 90°, 2 – rotate 180° and 3 – rotate right 90°

nMirror – specify the mirror options: 0 – no change, 1 – horizontal flip and 2 – vertical flip

nPreviewInterval – specify the number of frame between two messages from the library, which contain the snapshot of the transcoding process.

bGraphicOptimizer – Enable / disable graphic optimizer

pxImageProcess – frame processing routine. Use NULL if no processing is required. In this function, the image index in Devil image library is given so that it is possible to process each frame data before output to smv file.

Description:

After the transcoder is created in transcoding mode by specifying the destination file location in the NewTranscoder(), the transcoding process can be started by this API. The argument markA and markB can be used to specify the video partial selection in term of video offset from the beginning. There is a argument to specify the option in full image and full screen. This option is used to control how to resize the input video during transcoding. If full image is used, the transcoder will resize the input video such that the whole video can be seen on the screen even if the target screen cannot be fully utilized. In contrast, If full screen is used, the transcoder will resize the input video such that the whole screen will be utilized even if some video part (either vertical or horizontal) will be cropped. Thus, the target screen size is not exactly equal to the resultant smv video size. Apart from resize operation, the video orientation can be changed to Landscape. When landscape orientation is enabled, the video graphic will be rotated 90 degree in anti-clockwise. The audio quality can be specified in sampling frequency and number of channel. The graphic quality can be specified by frame rate, average graphic data in kB per second. For frame rate, the transcoder will select the lowest frame rate between that in the input video file and the specified frame rate. For average graphic data per second, transcoder will find the suitable compression level to achieve such graphic data amount in each second. Thus, the compression level is different from second to second.

During the process, SMV format library will use the window handle to post WM_UPDATE message for progress report. The WPARAM value is the percentage of the progress to be done. At the same time, the The LPARAM value is the HBITMAP handle to obtain the snapshot of the transcoding process. Application is responsible to free this handle if it is not used anymore. The frequency to have snapshot is determined by the nPreviewInterval. The 100% should be generated at the end of the transcoding. Thus, this value is an indication of the completion. After that the transcoder will close automatically. Further transcoding requires to create a new transcoder by using NewTranscoder().

Also, the frame processing routine will be called for every frame before resize operation is performed. The image index of DevIL image library is given. Further image processing can be done through image library.

BOOL StopTranscode(void);

Description:

This API can be used to stop the transcoding process before it is finished. No output file will be generated.

5. SKIN UI ELEMENT LIBRARY

A self-developed skin UI element library is used to enable skin-able user interface in SMV transcoder. The library contains button, checkbox, combo box, text, list box, progress bar, slider, etc. In addition to the interface graphic, a SMV transcoder with professional interface can be constructed.

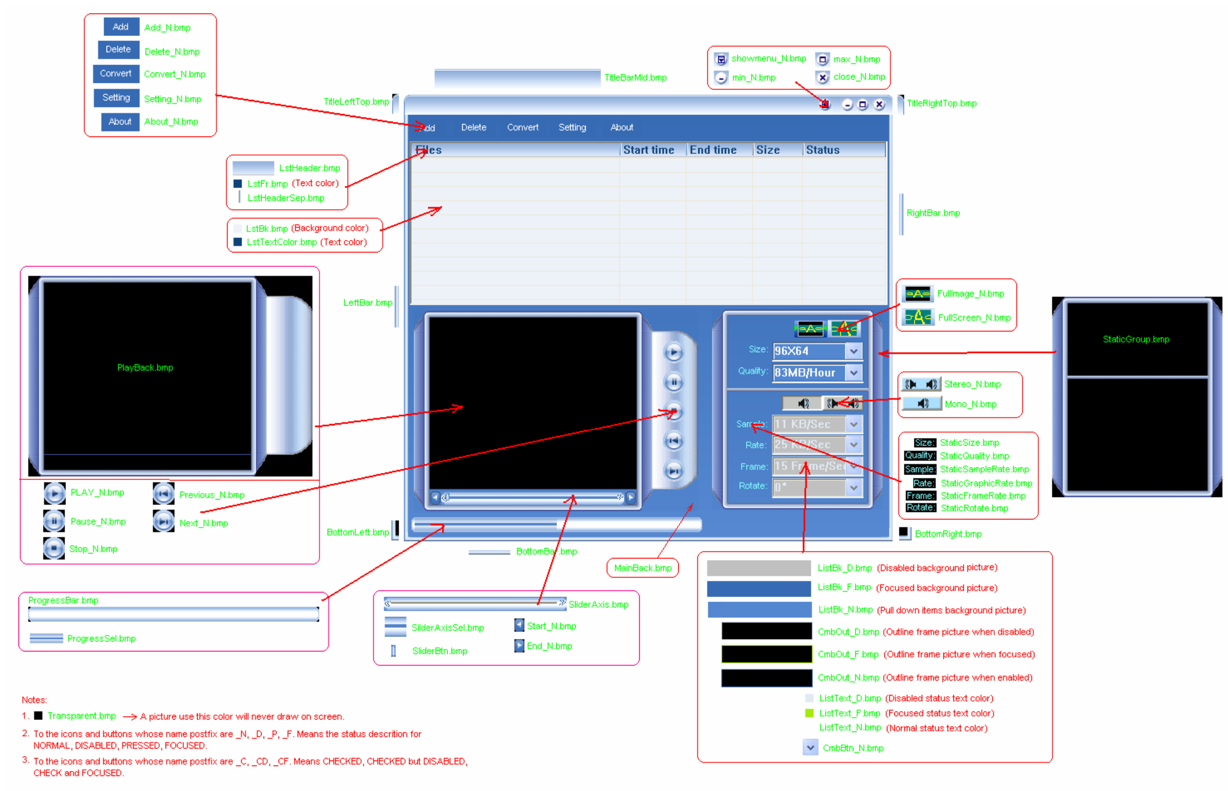
6. METHODOLOGY TO CHANGE THE SKIN

The skin of the application interface is defined by one configuration file and a number of bitmap files. The development kit in this app note contains two set of interface skin as a reference. In most cases, customer can modify the interface by editing the bitmap files as well as the configuration file. Each bitmap file represents the outlook of a particular UI element. The configuration file, skin.ini, is used to define the position of the UI element. In other words, there is no boundary for customer to define the application outlook without programming.

If customer wants to modify application logic of the SMV transcoder, it is required to develop their application based on the development kit and utilize the skin element library as well as smv format library. The default skin resource pack is not applicable. Customer needs to prepare a set of skin resource which match the new SMV transcoder.

In figure 3, the bitmap files in one reference skin is shown. The filename and the actual bitmap content are displayed with location pointing. Using this figure, it is easy for customer to prepare the bitmap files based on the customer's artwork design. In figure 4, the base point and size of SMV transcoder application is illustrated. This coordinate is used to define the position of UI element within the configuration file.

The SMV transcoder supports multiple interface skin for end-user selection. Maximum three different interface skin can be defined in default. The directory of the skin resource should be specified in the application configuration file, config.ini. If nothing is specified in the slot, that selection will be disabled.



Notes:

1. ■ Transparent.bmp → A picture use this color will never draw on screen.
2. To the icons and buttons whose name postfix are _N, _D, _P, _F. Means the status descrition for NORMAL, DISABLED, PRESSED, FOCUSED.
3. To the icons and buttons whose name postfix are _C, _CD, _CF. Means CHECKED, CHECKED but DISABLED, CHECK and FOCUSED.

Figure 3: Skin structure of a reference design

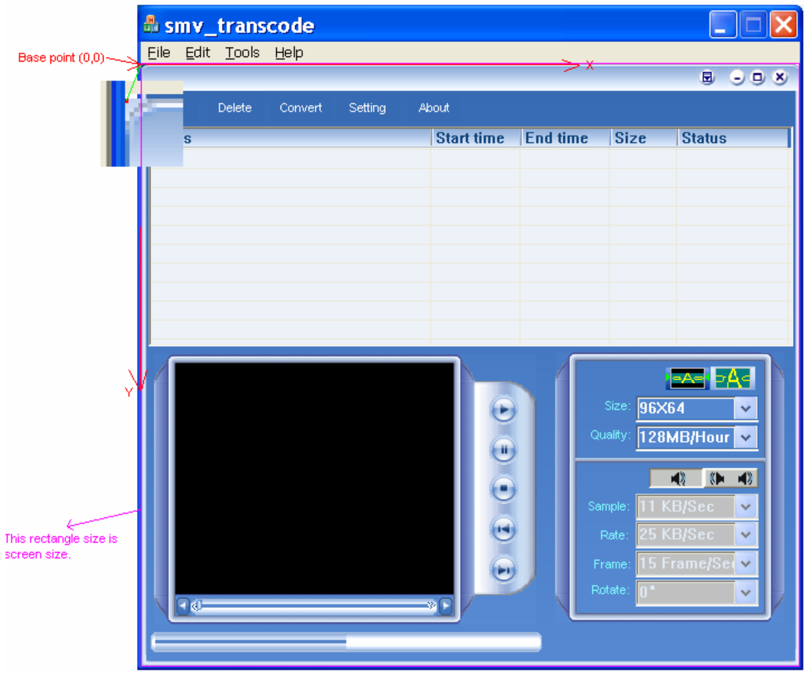


Figure 4: The base point and size of the application interface

7. SOFTWARE CUSTOMIZIATION

The software package provided by this App Note is targeted to those customers who want to develop their own PC transcoding software. The UI sample code in MFC library can be used as reference to develop a brand new user interface for the user to do the same actions:

- Specify the full path of the input video file
- Specify the full path of the output SMV file
- Transcoding parameters
  - Audio Quality
  - Graphic Quality
  - Video Orientation
  - Frame Rate
  - Dimension
  - Full Image or Full Screen
  - Partial Selection for Transcoding with preview
- Start Trancoding.
- Frame Image Processing
- Cancel Transcoding
- Exit the PC software

Apart from the mentioned actions, customers can develop other features such as media library management, SMV playback, advanced video selection, video graphic style, logo overlay, etc. These add-on features is out of scope in the App Note. Some features can be done in frame image processing. For example, a customer logo can be overlaid on each frame or some frames before passing to smv library to continue the transcoding process. Some features would be implemented from scratch without any help from smv library such as media library management.

The SMV format library APIs is provided to perform three critical features: frame preview in partial selection, SMV transcoding and frame image processing. The skin element library is provided for application interface construction.

8. ERROR HANDLING

During transcoding, SMV library will submit a message, WM_SHOWMESSAGE, to UI program for error notification. There are two parameters provided when message arrived in UI handling function. The first parameter define the message type: Error value of DirectX core, invalid graphic data size, codec detection error. In the error value of DirectX core, the second parameter is the actual value given by DirectX library. The message showed in UI can be customized in the development kit. For example, when codec detection error is received, codec package can be suggested to users according to the file format being transcoded.

9. HISTORY

Version 1.0

- Initial Version

Version 2.0

- Separate UI from the transcoding library. the UI code is provided as sample for software development
- Allow operation cancel during transcoding
- Allow full image or full screen conversion
- Show all supported media file in the source open dialog
- Add 128 x 128 video dimension option

Version 3.0

- SMV format version 2.0 is supported. Graphic compression algorithm is added
- Quality level is added to specify audio and video quality
- Video orientation can be specified
- Frame image processing using DevIL image library during transcoding
- Resolved defects
    - STMP00008710 – The transcoder is failed to process some video files in the second trial.
    - STMP0008711 – An incorrect frame rate is used in SMV file when there is no information to show the designed frame rate in the input video.

Version 3.2

- The error message displayed in smv library can be customized in UI example code
- Vertical and horizontal center alignments are used in graphic cropping when full screen option is enabled
- Unicode character set can be used in UI example code

Version 3.3

- Improve the transcoding speed by using linear scaling filter and enhanced JPEG quality searching algorithm
- Display total transcoding time at the end of the process

Version 3.4

- Improve the progress update of transcoding in different video formats
- Support Realplayer and QuickTime format (through Real alternative and QuickTime Alternative)
- Provide correct image index when calling to post-processing function
- Support 90°, 180° and 270° rotations. It is selectable in customized dialog box.
- Support horizontal and vertical flip in library level. No UI is added for this option
- Fixed white line problem in rotations
- Fixed frame rate estimation in Divx codec
- Fixed output filename generation in Realplayer file
- Fixed program crash when pressing "transcode" after no codec is detected

Version 3.5

- Remove format restriction for frame rate selection due to the new smv-specific JPEG decoder
- Add more options in dimensions, frame rate and compression level selection
- Default disable image processing in every frame
- Use better scaling filter in "High" quality level so that high image quality can be achieved.

Version 4.0

- A changeable skin interface is supported in SMV transcoder application
- Multiple video files manipulation
- Transcoding can be done in batch form with unique transcoding settings for each video file.
- In-place preview for range selection
- The SMV transcoding process is shown and the resolution of the snapshot is exactly the same as the transcoding setting.
- A fixed directory can be specified to output all SMV files
- Gerenal fix in smv format library to support different input video format.

Verion 4.1

- Match the transcoding parameters with the orientation setting in customized quality level
- Display correct preview during range selection according to the orientation setting.

10. OPEN SOURCE LIBRARY IN SMV TRANSCODER

Libsndfile

PC transcoding software uses an open source sound library, *libsndfile*, for ADPCM format conversion. The source code of *libsndfile* 1.0.11 and the license file of *libsndfile* are included for reference. The further information and the latest version of source code can be obtained in their official website:

http://www.mega-nerd.com/libsndfile/

Resample

PC transcoding software uses an open source sound library, libresample, for audio sampling frequency conversion. The source code of libresample 0.1.3 and the license file of libresmple are included for reference. The further information and the latest version of source code can be obtained in their official website:

http://www-ccrma.stanford.edu/~jos/resample/

DevIL

PC transcoding software uses an open source image library, DevIL, for frame graphic processing. The source code of DevIL 1.6.7 and the license file of DevIL are included for reference. The further information and the latest version of source code can be obtained in their official website:

http://www.imagelib.org and http://openil.sourceforge.net