

API Document

JPEG Decoder on ARM

Ver: 1.2

Release Date: 02-12-2005



Emuzed, Inc.

Proprietary & Confidential to Emuzed, Inc.

Revision History

Version	Date	Description of Change(s)	Author(s)
1.0	Jan 12, 2005	Initial draft	Durga Prasad.P
1.1	Feb 11, 2005	Modified create and decode image APIs.	Durga Prasad.P
1.2	Dec 02, 2005	Added Run time macros, revised color format macros.	Naresh Sharma

1 Introduction

The document describes the APIs for JPEG [1] image decoder. The decoder supports Baseline and Progressive JPEG coding modes.

2 API

This section describes the application programming interfaces for JPEG image decoder.

2.1 Parse JPEG Header

Function Name

```
int32 JDEmz_gParseJpegHeader(uint8 *configStream, uint32 configLength,  
                             tImageInfo_JDEmz *imageInfo)
```

Description

This function parses the JPEG image header. It returns information about the JPEG image resolution and color format.

Parameters

`configStream` :[IN] Buffer containing the configuration information of the image to be decoded.

`configLength` :[IN] Length of the configuration information.

`imageInfo` :[OUT] Image parameter information data structure. It contains the width and height of the image in pixels along with the image color format.

Return Value

E_SUCCESS

E_FAILURE

E_OUT_OF_MEMORY

E_CORRUPT

E_NOT_SUPPORTED

E_INSUFFICIENT_DATA

2.2 Create Decoder Instance

Function Name

```
int32 JDEmz_gCreateJpegDecoder(int32 **handle, uint8 *stream,  
                               uint32 streamLength, tRect_JDEmz *cropWindow, uint8 dctScaleFactor,  
                               tBaseVideoFrame_JDEmz *outBuffer)
```

Description

This function creates an instance of JPEG image decoder and parses the JPEG header. It allocates memory required for all the internal structures. The input / output buffers must be allocated by the user. The user also has to specify the cropWindow and dctScaleFactor.

Parameters

`handle` :[OUT] Handle to JPEG image decoder.

<code>stream</code>	: [IN] Buffer containing the entire image to be decoded.
<code>streamLength</code>	: [IN] Length of the input image buffer.
<code>cropWindow</code>	: [IN] Contains information about cropping co-ordinates and crop window dimensions with respect to the input image. When there is no cropping user has to set these values to zero.
<code>dctScaleFactor</code>	: [IN] Indicates the scale down factor in DCT domain. The supported values are 1, 2, 4 and 8 only. Any other value passed will return <code>E_ERROR_ARGUMENT</code> .
<code>outBuffer</code>	: [OUT] Decoded image data structure. The decoded image data can be in one of the following formats - YUV[4:4:4], YUV[4:2:0], YUV[4:2:2], RGB[4:4:4], MONOCHROME planar format. User has to allocate memory for this buffer. The width and height of this buffer should be same as crop window dimensions divided by the <code>dctScaleFactor</code> . eg: <code>outBuffer</code> size for luminance component is ((crop window Height / <code>dctScaleFactor</code>) * (crop window Width / <code>dctScaleFactor</code>))

NOTE : If the width and height of the input JPEG image (actual width, actual height) are of odd dimensions, the output width and height are extended (extended width, extended height) according to the input color format.

If input JPEG image is of YUV420 format, output (`outBuffer.width`, `outBuffer.height`) is extended to be multiples of 16.

If input JPEG image is of YUV422H format, output image width (`outBuffer.width`) is extended to be a multiple of 16 and output image height (`outBuffer.height`) is extended to be a multiple of 8.

If input JPEG image is of YUV422V format, output image width (`outBuffer.width`) is extended to be a multiple of 8 and output image height (`outBuffer.height`) is extended to be a multiple of 16.

If input JPEG image is of YUV444/RGB444/Monochrome format, output image width (`outBuffer.width`) and image height (`outBuffer.height`) is extended to be multiples of 8.

Return Value

`E_SUCCESS`

`E_FAILURE`

`E_OUT_OF_MEMORY`

`E_ERROR_ARGUMENT`

`E_CORRUPT`

2.3 Decode an Image

Function Name

`int32 JDEmz_gDecodeJpegImage(int32 *handle, uint32 numRowsToDecode)`

Description



This function decodes a JPEG Image. Output of this function is the decoded frame buffer. The decoded frame buffer is in the same planar color format as the encoded image. User can call this function multiple times by specifying the number of MCU rows to be decoded in each call.

Parameters

`handle` :[IN] Handle to JPEG image decoder.

`numRowsToDecode` :[IN] Indicates the number of MCU rows to be decoded for baseline JPEG image. For progressive JPEG images, the control is returned to the user after decoding one complete scan. If this parameter is set to '0', then entire image is decoded before control is returned to user.

Return Value

`E_SUCCESS`
`E_FAILURE`
`E_OUT_OF_MEMORY`
`E_PARTIAL_DECODED`
`E_INSUFFICIENT_DATA`
`E_CORRUPT`

2.4 Delete Decoder Instance

Function Name

```
int32 JDEmz_gDeleteJpegDecoder(int32 *handle)
```

Description

This function deletes the JPEG image decoder instance.

Parameters

`handle` :[IN] Handle to JPEG image decoder.

Return Value

`E_SUCCESS`

3 Data Types

This section describes the data types used by the APIs.

3.1 Data Structures

3.1.1 tImageInfo_JDEmz

This is the data structure for the information about the image parameters.

```
typedef struct ImageInfo_JDEmz
{
    unsigned short int width;
    unsigned short int height;
```

```

    unsigned short int  actualWidth;
    unsigned short int  actualHeight;
    unsigned char   frameType;
} tImageInfo_JDEmz;
```

width : Extended width of the Frame.

height : Extended height of the Frame.

actualWidth : Actual width of the frame.

actualHeight: Actual height of the frame.

frameType : Color format of the frame as in the compressed data. The supported color formats are: YUV420, YUV422H (chrominance sub-sampled in horizontal direction), YUV422V (chrominance sub-sampled in vertical direction), YUV444, RGB444 and MONOCHROME.

3.1.2 tRect_JDEmz

This is the data structure for cropping frames.

```

typedef struct Rect_JDEmz
{
    int32 xOffset;
    int32 yOffset;
    int32 wndWidth;
    int32 wndHeight;
} tRect_JDEmz;
```

xOffset : This is the x co-ordinate of the top left corner of the cropping window. (must be a multiple of 16)

yOffset : This is the y co-ordinate of the top left corner of the cropping window.(must be a multiple of 16)

wndWidth : This specifies the width of the crop window. (must be a multiple of 16)

wndHeight : This specifies the height of the crop window. (must be a multiple of 16)

3.1.3 tBaseVideoFrame_JDEmz

This is the data structure for the output frame buffer.

```

typedef struct BaseVideoFrame_JDEmz
{
    uint8  *lum;
    uint8  *cb;
    uint8  *cr;
    uint16 width;
    uint16 height;
    uint32 timeStamp;
} tBaseVideoFrame_JDEmz;
```

lum : Pointer to the buffer for storing the luminance component.
 cb : Pointer to the buffer for storing the Cb (chrominance) component.
 cr : Pointer to the buffer for storing the Cr (chrominance) component.
 width : Width of the luminance frame buffer.
 height : Height of the luminance frame buffer.
 timeStamp : Not used in JPEG decoding..

3.2 Macro Defines

Color Format Types:

```

#define YUV420      0x01 /* YUV4.2.0. format (Y2x2, U1x1, V1x1) */
#define YUV422H     0x02 /* YUV4.2.2.H format (Y2x1, U1x1, V1x1) */
#define YUV422V     0x03 /* YUV4.2.2.V format (Y1x2, U1x1, V1x1) */
#define YUV444      0x04 /* YUV4.4.4. format (Y1x1, U1x1, V1x1) */
#define RGB444      0x05 /* RGB4.4.4. format (R1x1, G1x1, B1x1) */
#define MONOCHROME  0x06 /* Monochrome format (Y1x1) */
#define YUYV        0x07 /* Y.U.Y.V. format (Y1, U1, Y2, V1) */
#define RGB16       0x08 /* RGB5.6.5. format (R1x5, G1x6, B1x5) */

```

Error Codes:

```

#define E_SUCCESS          0
#define E_FAILURE          (-1)
#define E_OUT_OF_MEMORY   (-2)
#define E_OUT_OF_RANGE    (-3)
#define E_FILE_CREATE_FAIL (-4)
#define E_UNDEFINED_FLAG  (-5)
#define E_FILE_READ_FAIL  (-6)
#define E_FILE_OPEN_FAIL  (-7)
#define E_END_OF_FILE      (-8)
#define E_NOT_COMPLETE     (-9)
#define E_INSUFFICIENT_OUTBUFF (-10)
#define E_INSUFFICIENT_DATA (-11)
#define E_ERR_ARGUMENT     (-12)
#define E_NOT_SUPPORTED    (-13)
#define E_PARTIAL_DECODED  (-14)
#define E_CORRUPT          (-15)
#define E_ERROR_ARGUMENT   (-16)

```

Run Time Macros:

ROM_BUILD	Set to enable the Run Time ROM build functions.
FLASH_BUILD	Set to enable the Run Time Flash build functions.
EMZ_PROG_DECODE	Set to enable Progressive decoding option.
WINDOWS_BUILD	Set to enable windows malloc / calloc function calls.
ARM_ASM	Set to enable ARM specific assembly optimization.

4 References

- [1] JPEG Standard, “Information Technology- Digital Compression and Coding of Continuous-Tone Still Images- Requirements and Guidelines,” Recommendation T.81, ITU, September 1992.