JPEG
Reference Guide

V1.0

Publication Release Date: Oct. 2011

Support Chips:

W55FA Series

Support Platforms:

Non-OS

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

Table of Contents

1. JPEG Overview

FA95/VA95 Non-OS library consists of a set of libraries. These libraries are built to access those on-chip functions such as VPOST, APU, SIC, USBH, USBD, GPIO, I2C, SPI and UART, as well as File System (NVTFAT), USB Mass Storage devices (UMAS) and NAND Flash devices (GNAND). This document describes the provided APIs of JPEG library. With these APIs, user can quickly build a binary target for JPEG library on FA95/VA95 micro processor.

These libraries are created by using ARM Development Suite 1.2. Therefore, they only can be used in ADS environment.

1.1. JPEG Overview

This library is designed to make user application to use FA95/VA95 JPEG more easily.
The JPEG library has the following features:

- JPEG Normal / Encode function
- JPEG Encode Upscale function
- JPEG Decode Downscale function
- JPEG Window Decode function
- JPEG Decode Input Wait function
- JPEG Decode Output Wait function

1.2. Programming Guide

System Overview

The JPEG Codec supports Baseline Sequential Mode JPEG still image compression and decompression that is fully compliant with ISO/IEC International Standard 10918-1 (T.81). The features and capability of the JPEG codec are listed below.

JPEG Features

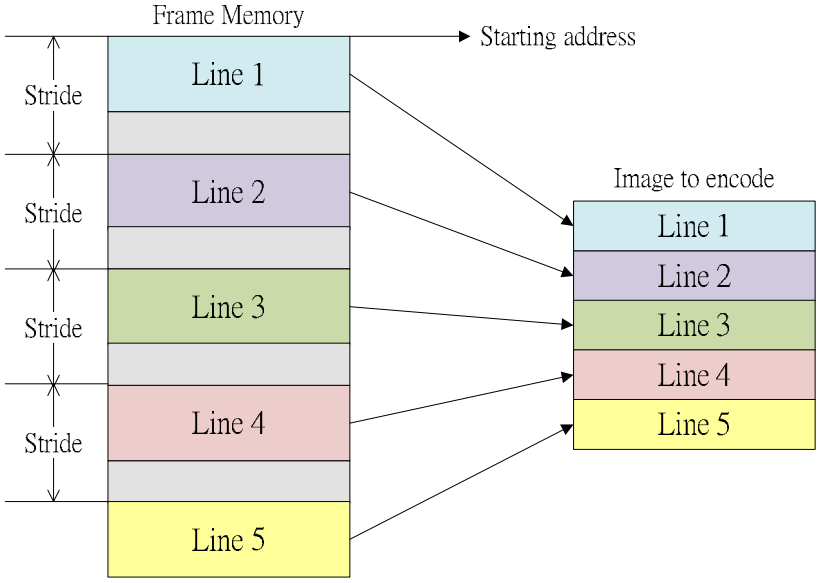
- Support to encode interleaved YCbCr 4:2:2/4:2:0 and gray-level (Y only) format image
- Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0/4:1:1 and gray-level (Y only) format image
- Support to decode YCbCr 4:2:2 transpose format

- The encoded JPEG bit-stream format is fully compatible with JFIF and EXIF standards
- Support Capture and JPEG hardware on-the-fly access mode for encode
- Support JPEG and Playback hardware on-the-fly access mode for decode
- Support software input/output on-the-fly access mode for both encode and decode
- Support arbitrary width and height image encode and decode
- Support three programmable quantization-tables
- Support standard default Huffman-table and programmable Huffman-table for decode
- Support arbitrarily 1X~8X image up-scaling function for encode mode
- Support down-scaling function for encode and decode modes
- Support specified window decode mode
- Support quantization-table adjustment for bit-rate and quality control in encode mode
- Support rotate function in encode mode

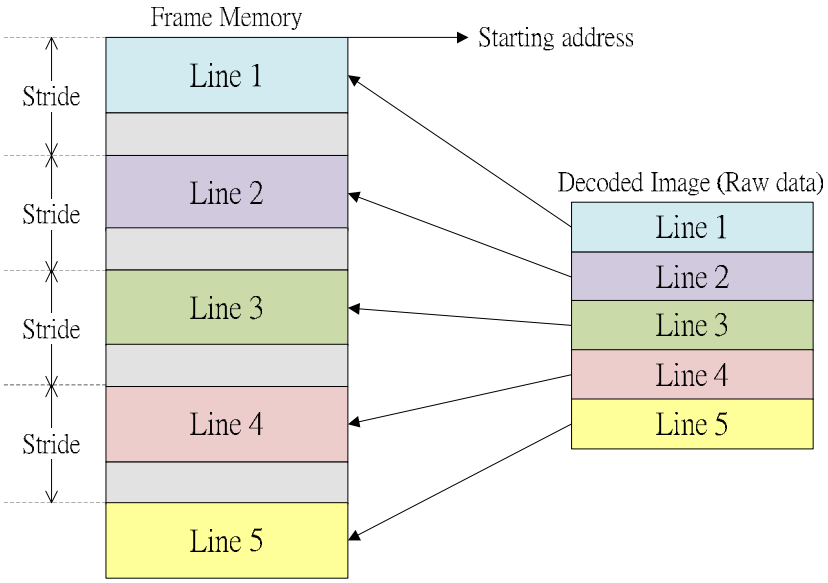
JPEG Operation Control

n Memory access

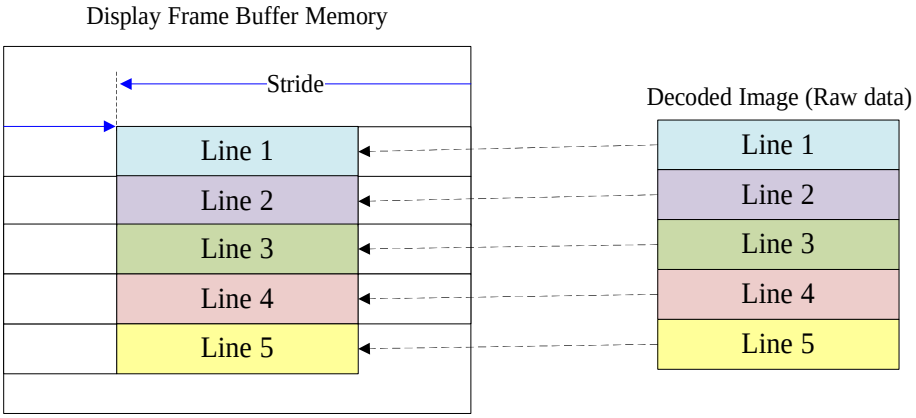
The following figure shows the encode mode to access the source data which are from sensor normally and stored on the SDRAM.



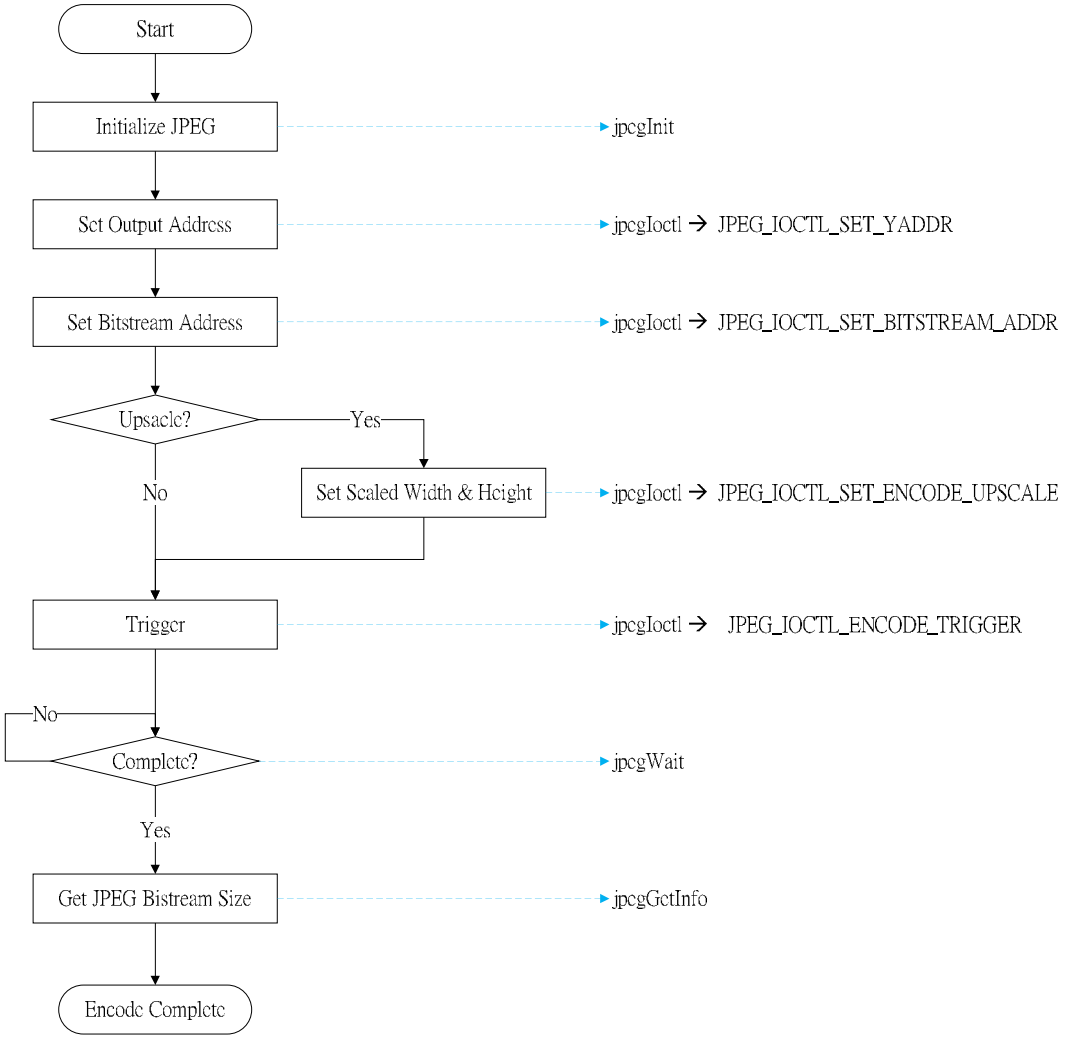
Following figure shows the decode mode to output the decoded raw data on the SDRAM.



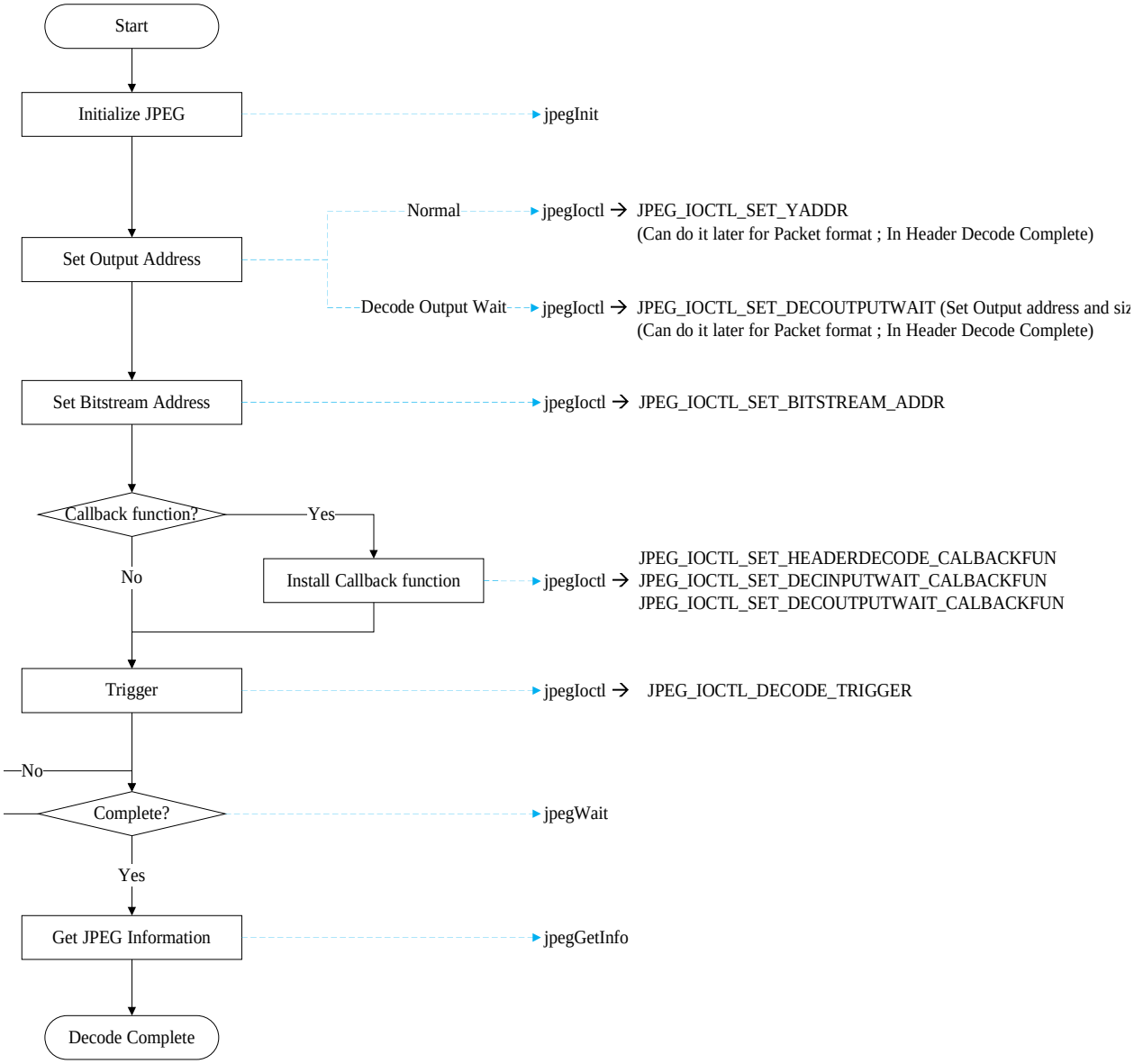
User can use stride function to output decoded image to any position on the Display Frame Buffer for Display. Following figure shows the decode mode with stride to output the decoded raw data on the Display Frame Buffer.



- **Encode operation flow**

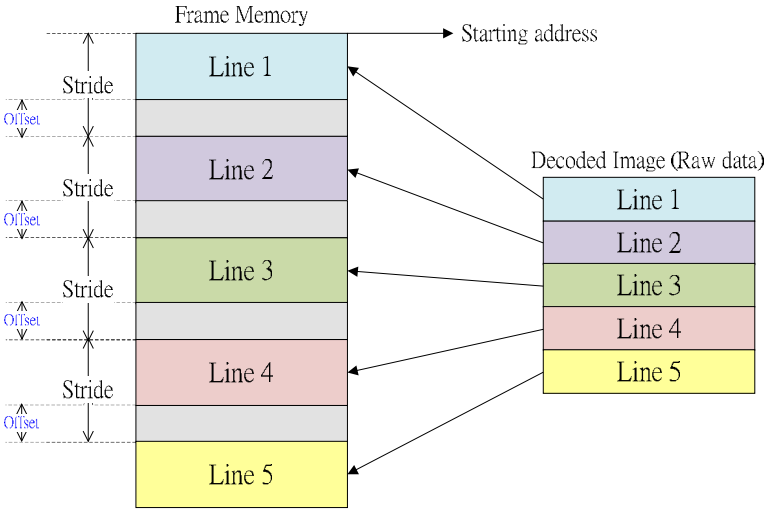

Start
Initialize JPEG
jpegInit
Set Output Address
jpegIoctl → JPEG_IOCTL_SET_YADDR
Set Bitstream Address
jpegIoctl → JPEG_IOCTL_SET_BITSTREAM_ADDR
Upsacle?
Yes
No
Set Scaled Width & Height
jpegIoctl → JPEG_IOCTL_SET_ENCODE_UPSCALE
Trigger
jpegIoctl → JPEG_IOCTL_ENCODE_TRIGGER
No
Complete?
jpegWait
Yes
Get JPEG Bistream Size
jpegGetInfo
Encode Complete

n Decode operation flow



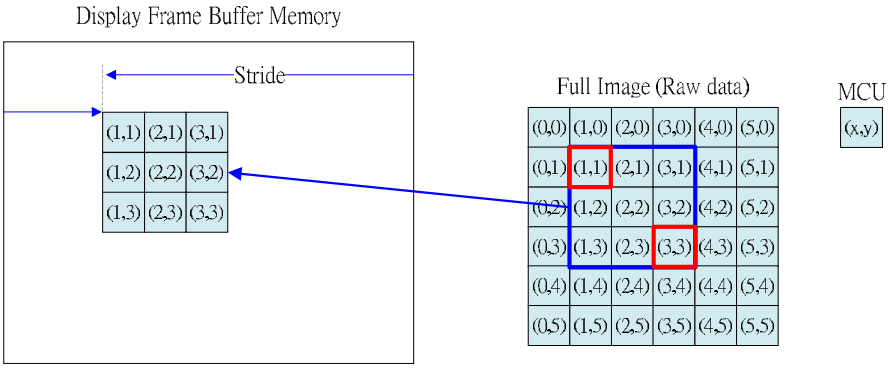
n Decode stride

Before clearing Header Decode End interrupt, the value of stride must be set to stride value instead of original width. Offset is the difference between Stride and Image width. If Offset is 0, the decoded Raw data is continuous.



n Window Decode

The JPEG decoder supports specified window decode mode. This function allows user to specify a sub-window region within the whole image to be decoded as shown in the following figure. Only the specified window region image will be decoded and stored to frame memory.

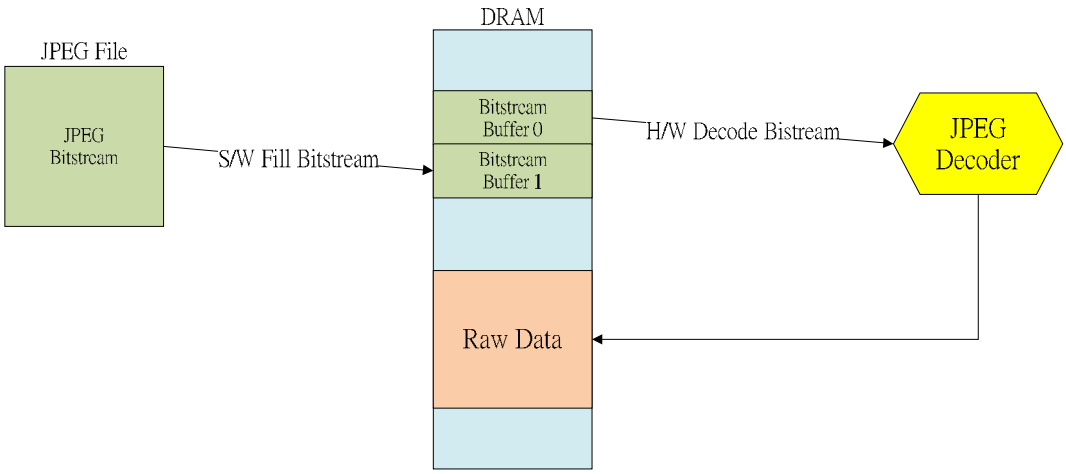


n Decode Input Wait

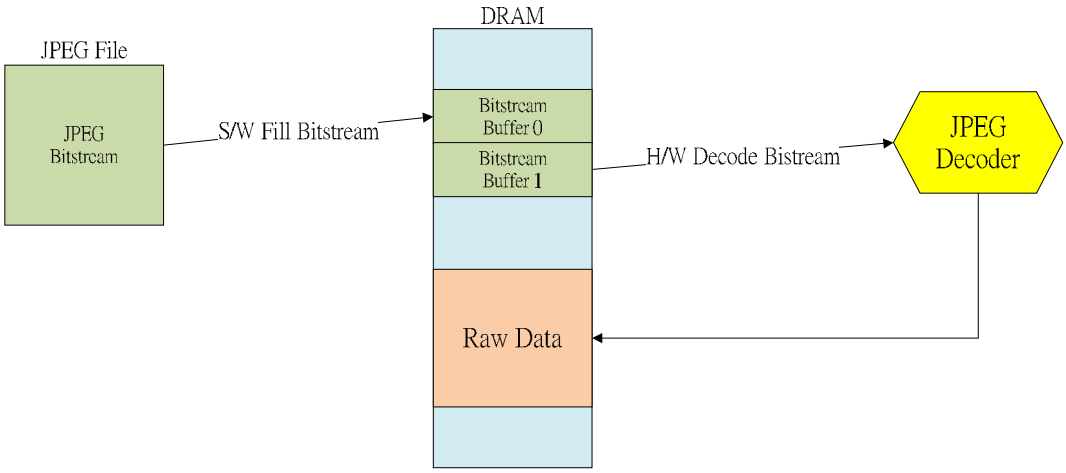
When the JPEG is in decoding mode, the input source is the JPEG bit-stream written by software. The bit-stream buffer size is in 2K unit dual-buffer manner. If the buffer-size is 2KB,user needs to fill 1KB bit-stream into one of the half buffer region before resuming JPEG operation when an input-wait interrupt is generated.

When JPEG engine decodes one of the half buffers, the Decode Input Wait call back function will be called. The Only thing user needs to do is to fill bit stream to the other buffer like the following Step 1 & Step 2 untill entired bistream is filled into the buffer.

[Step 1] JPEG engine decodes the data in Buffer 0 and S/W fills the data into Buffer 1



[Step 2] JPEG engine decodes the data in Buffer1 and S/W fills the data into Buffer 0

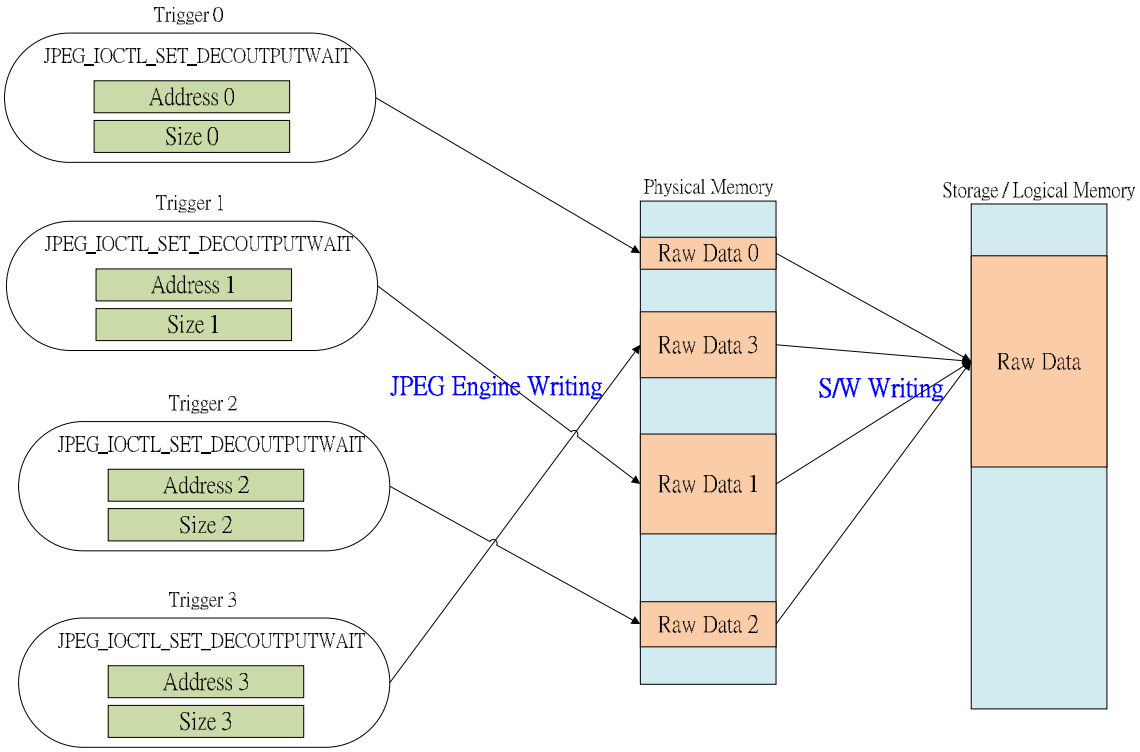


n Decode Output Wait

When there is not enough continuous space to store the decode output raw data, JPEG engine support a function to output data partially. User can get the whole data by assigning several output data address and size settings (JPEG_IOCTL_SET_DECOUTPUTWAIT). Using this function, user can get the JPEG decoded image that larger than the available continuous memory space.

The decode output wait call back function will be called when the Output Data size is equal to the Size assigned by IOCTL - JPEG_IOCTL_SET_DECOUTPUTWAIT. In the call back function, user should move the data to it's destination address and call the IOCTL again to set next address and data size.

[Note 1]. The data size of the final ICOTL must equal to the extract output size. Otherwise, user will get the Decode mplete interrupt instead of Decode output wait interrupt.



- **Header Decode Complete**

  In the callback function, user can get JPEG image width and height by calling jpegGetInfo(). After getting the information, user can use jpegIoctl to

  - Allocate and set output buffer → JPEG_IOCTL_SET_YADDR for Packet format Only
  - Change output buffer address → JPEG_IOCTL_SET_YADDR for Packet format Only
  - Set Downscale → JPEG_IOCTL_SET_DECODE_DOWNSCALE
  - Set Decode output Stride → JPEG_IOCTL_SET_DECODE_STRIDE
  - Set windows decode → JPEG_IOCTL_SET_WINDOW_DECODE

JPEG Library Constant Definition

- **Encode operation**

Name	Value	Description
Encode format		
JPEG_ENC_PRIMARY	0	Encode operation : Primary JPEG
JPEG_ENC_THUMBNAIL	1	Encode operation : Thumbnail JPEG
JPEG_ENC_SOURCE_PLANAR	0	Encode source : planar format
JPEG_ENC_SOURCE_PACKET	1	Primary Encode source : packet format

JPEG_ENC_PRIMARY_YUV420	0xA0	Primary Encode image format : YUV 4:2:0
JPEG_ENC_PRIMARY_YUV422	0xA8	Primary Encode image format : YUV 4:2:2
JPEG_ENC_PRIMARY_GRAY	0xA1	Primary Encode image format : GRAY
JPEG_ENC_THUMBNAIL_YUV420	0x90	Thumbnail Encode image format : YUV 4:2:0
JPEG_ENC_THUMBNAIL_YUV422	0x98	Thumbnail Encode image format : YUV 4:2:2
JPEG_ENC_THUMBNAIL_GRAY	0x91	Thumbnail Encode image format : GRAY
Encode Header control		
JPEG_ENC_PRIMARY_DRI	0x10	Restart Interval in Primary JPEG Header
JPEG_ENC_PRIMARY_QTAB	0x20	Quantization-Table in Primary JPEG Header
JPEG_ENC_PRIMARY_HTAB	0x40	Huffman-Table in Primary JPEG Header
JPEG_ENC_PRIMARY_JFIF	0x80	JFIF Header in Primary JPEG Header
JPEG_ENC_THUMBNAIL_DRI	0x1	Restart Interval in Thumbnail JPEG Header
JPEG_ENC_THUMBNAIL_QTAB	0x2	Quantization-Table in Thumbnail JPEG Header
JPEG_ENC_THUMBNAIL_HTAB	0x4	Huffman-Table in Thumbnail JPEG Header
JPEG_ENC_THUMBNAIL_JFIF	0x8	JFIF Header in Thumbnail JPEG Header

- **Decode operation**

Name	Value	Description
Decode output format		
JPEG_DEC_PRIMARY_PLANAR_YUV	0x8021	Primary Decode output format : planar format
JPEG_DEC_PRIMARY_PACKET_YUV422	0x0021	Primary Decode output format : planar YUV422
JPEG_DEC_PRIMARY_PACKET_RGB555	0x04021	Primary Decode output format : packet RGB555
JPEG_DEC_PRIMARY_PACKET_RGB565	0x06021	Primary Decode output format : packet RGB565
JPEG_DEC_PRIMARY_PACKET_RGB888	0x14021	Primary Decode output format : packet RGB888
JPEG_DEC_THUMBNAIL_PLANAR_YUV	0x8031	Thumbnail Decode output format : planar YUV
JPEG_DEC_THUMBNAIL_PACKET_YUV422	0x0031	Thumbnail Decode output format : packet RGB555
JPEG_DEC_THUMBNAIL_PACKET_RGB555	0x4031	Thumbnail Decode output format : packet RGB565

JPEG format		
JPEG_DEC_YUV420	0x000	JPEG format is YUV420
JPEG_DEC_YUV422	0x100	JPEG format is YUV422
JPEG_DEC_YUV444	0x200	JPEG format is YUV444
JPEG_DEC_YUV411	0x300	JPEG format is YUV411
JPEG_DEC_GRAY	0x400	JPEG format is Gray
JPEG_DEC_YUV422T	0x500	JPEG format is YUV422 Transport

JPEG Library Property Definition

The JPEG library provide property structure to set JPEG property.

JPEG_INFO_T;

Name	Value	Description
yuvformat	JPEG_DEC_YUV420 JPEG_DEC_YUV422 JPEG_DEC_YUV444 JPEG_DEC_YUV411 JPEG_DEC_GRAY JPEG_DEC_YUV422T	JPEG format (Decode only)
width	< 8192	Decode Output width (Decode only)
height	< 8192	Decode Output height (Decode only)
jpeg_width	< 65535	JPEG width (Decode only)
jpeg_height	< 65535	JPEG height (Decode only)
stride	< 8192	Decode output Stride (Decode only)
bufferend	Reserved	Reserved
image_size[2]	< 2^{24} -1	Encode Bitstream Size (Encode Only)

The JPEG library provide window decode function, user can partially decode the JPEG image by MCU unit (16 pixels *16 pixels).

JPEG_WINDOW_DECODE_T

Name	Value	Description
u16StartMCUX	0~511	Decode MCU Horizontal Start index
u16StartMCUY	0~511	Decode MCU Vertical Start index
u16EndMCUX	0~511	Decode MCU Horizontal End index

u16EndMCUY	0~511	Decode MCU Vertical End index
u32Stride	< 8192	Decode output Stride

1.3. JPEG API

jpegOpen

Synopsis

INT jpegOpen(VOID)

Description

This function initializes the software resource, sets the engine clock and enables its interrupt

Parameter

None

Return Value

E_SUCCESS - Always successes

Example

jpegOpen();

jpegClose

Synopsis

VOID jpegClose(VOID)

Description

Disable clock of JPEG engine and disable its interrupt

Parameter

None

Return Value

None

Example

jpegClose();

jpegInit

Synopsis

VOID jpegInit(VOID)

Description

Reset JPEG engine and set default value to its registers

Parameter

None

Return Value

None

Example

jpegInit();

jpegGetInfo

Synopsis

VOID jpegGetInfo(JPEG_INFO_T *info)

Description

This function can get JPEG width and height after header decode complete and get JPEG bit stream size after encode complete.

Parameter

info	JPEG Data type pointer stores the returned JPEG header information

Return Value

None

Example

```
JPEG_INFO_T jpegInfo;
/* Get JPEG Header information */
jpegGetInfo(&jpegInfo);
```

jpegWait

Synopsis

INT jpegWait(VOID)

Description

After triggers JPEG engine, application need to wait the completion flag while JPEG engine completes it job.

Parameter

None

Return Value

E_FAIL	Error happen
E_SUCCESS	Action is done

Example

```
jpegWait();
```

jpegIsReady

Synopsis

```
BOOL jpegIsReady(VOID)
```

Description

The function can get the JPEG engine status.

Parameter

None

Return Value

TRUE	Engine is ready
FALSE	Engine is busy

Example

```
jpegIsReady ();
```

jpegSetQTAB

Synopsis

```
INT jpegSetQTAB(
        PUINT8      puQTable0,
        PUINT8      puQTable1,
        PUINT8      puQTable2,
        UINT8       u8num
);
```

Description

The function can specify the Quantization table

Parameter

puQTable0	Specify the address of Quantization table 0
puQTable1	Specify the address of Quantization table 1
puQTable2	Specify the address of Quantization table 2
u8num	Specify the number of Quantization table

Return Value

E_SUCCESS : Success

E_JPEG_TIMEOUT : Set Quantization table timeout

Example

```
jpegSetQTAB(g_au8QTable0,g_au8QTable1, 0, 2);
```

jpegIoctl

Synopsis

VOID jpegIoctl(UINT32 cmd, UINT32 arg0, UINT32 arg1)

Description

This function allows programmers configure JPEG engine, the supported command and arguments listed in the table below.

Command	Argument 0	Argument 1	comment
JPEG_IOCTL_SET_YADDR	JPEG Y component frame buffer address		Specify the JPEG Y component frame buffer address.
JPEG_IOCTL_SET_YSTRIDE	JPEG Y component frame buffer stride		Specify the JPEG Y component frame buffer stride
JPEG_IOCTL_SET_USTRIDE	JPEG U component frame buffer stride		Specify the JPEG U component frame buffer stride
JPEG_IOCTL_SET_VSTRIDE	JPEG V component frame buffer stride		Specify the JPEG V component frame buffer stride
JPEG_IOCTL_SET_BITSTREAM_ADDR	JPEG bit stream buffer starting address		Specify the bit stream frame buffer starting address
JPEG_IOCTL_SET_SOURCE_IMAGE_HEIGHT	The encode source image height in pixel		Specify the encode source image height in pixel
JPEG_IOCTL_ENC_SET_HEADER_CONTROL	JPEG_ENC_PRIMARY_DRI JPEG_ENC_PRIMARY_QTAB JPEG_ENC_PRIMARY_HTAB JPEG_ENC_PRIMARY_JFIF		Specify the header information includes in the encoding bit stream
JPEG_IOCTL_SET_DEFAULT_QTAB			Specify the Quantization table
JPEG_IOCTL_SET_DECODE	JPEG_DEC_PRIMARY_PLANAR_YUV		Specify the decoded image output

_MODE	JPEG_DEC_PRIMARY_PACKET_YUV422 JPEG_DEC_PRIMARY_PACKET_RGB555 JPEG_DEC_PRIMARY_PACKET_RGB565 JPEG_DEC_PRIMARY_PACKET_RGB888		format
JPEG_IOCTL_SET_ENCODE_MODE	JPEG_ENC_SOURCE_PLANAR JPEG_ENC_SOURCE_PACKET	JPEG_ENC_PRIMARY_YUV420 JPEG_ENC_PRIMARY_YUV422	Specify the encode source format and encoding image format
JPEG_IOCTL_SET_DIMENSION	Image height	Image width	Set the encode image dimension or decode output image dimension
JPEG_IOCTL_ENCODE_TRIGGER			Trigger the JPEG operation for encoding
JPEG_IOCTL_DECODE_TRIGGER			Trigger the JPEG operation for decoding
JPEG_IOCTL_WINDOW_DECODE	JPEG_WINDOW_DECODE_T		Enable window decode mode and set the decode window region
JPEG_IOCTL_SET_DECODE_STRIDE	Decode Output Stride (in pixel)		Specify the decode output stride
JPEG_IOCTL_SET_DECODE_DOWNSCALE	Scaled Height	Scaled Width	Set Decode downscale function
JPEG_IOCTL_SET_ENCODE_UPSCALE	Scaled Height	Scaled Width	Set Encode Upscale function
JPEG_IOCTL_SET_HEADERDECODE_CALBACKFUN	Header Decode Complete Call Back function pointer		Set Header Decode Complete Call Back function pointer
JPEG_IOCTL_SET_DECINPUTWAIT_CALBACKFUN	Decode Input Wait Call Back function pointer		Set Decode Input Wait Call Back function pointer
JPEG_IOCTL_ADJUST_QTAB	JPEG_ENC_PRIMARY JPEG_ENC_THUMBNAIL	Quantization-Table Adjustment and control values[0]	Set Quantization-Table Adjustment and control
JPEG_IOCTL_ENC_RESERVED_FOR_SOFTWARE	Reserved size		Reserve memory space for user application
JPEG_IOCTL_SET_UADDR	Address for U Component		Set address for U Component
JPEG_IOCTL_SET_VADDR	Address for V Component		Set address for V Component
JPEG_IOCTL_SET_ENCODE_PRIMARY_RESTART_INTERVAL	Primary Restart interval		Set Primary Restart interval size
JPEG_IOCTL_SET_ENCODE_THUMBNAIL_RESTART_INTERVAL	Thumbnail Restart interval		Set Thumbnail Restart interval size
JPEG_IOCTL_GET_ENCODE_PRIMARY_RESTART_INTERVAL	The pointer to store Primary Restart interval size		Get Primary Restart interval size
JPEG_IOCTL_GET_ENCODE_THUMBNAIL_RESTART_INTERVAL	The pointer to store Thumbnail Restart interval size		Get Thumbnail Restart interval size
JPEG_IOCTL_SET_THUMBNAIL_DIMENSION	Thumbnail Heightt	Thumbnail Width	Set Thumbnail Dimension
JPEG_IOCTL_SET_ENCODE_SW_OFFSET	Offset		Set Software Encode Offset
JPEG_IOCTL_GET_THUMBNAIL_DIMENSION	The pointer to store Thumbnail Heightt	The pointer to store Thumbnail Width	Get Thumbnail Dimension

JPEG_IOCTL_GET_ENCODE_SW_OFFSET	The pointer to store Encode Offset		Get Software Encode Offset
JPEG_IOCTL_SET_ENCODE_PRIMARY_DOWNSCALE	Primary Downscaled Heightt	Primary Downscaled Width	Set Primary Encode downscale Size (Planar format only)
JPEG_IOCTL_SET_ENCODE_THUMBNAIL_DOWNSCALE	Thumbnail Downscaled Heightt	Thumbnail Downscaled Width	Set Thumbnail Encode downscale Size (Planar format only)
JPEG_IOCTL_SET_ENCODE_PRIMARY_ROTATE_RIGHT			Encode rotate right (Planar format only)
JPEG_IOCTL_SET_ENCODE_PRIMARY_ROTATE_LEFT			Encode rotate left (Planar format only)
JPEG_IOCTL_SET_ENCODE_PRIMARY_ROTATE_NORMAL			Encode no rotate (Planar format only)
JPEG_IOCTL_SET_DECOUTPUTWAIT_CALBACKFUN	Decode Output Wait call back function pointer		Set Decode Output Wait call back function (Packetformat Only)
JPEG_IOCTL_SET_DECOUTPUTWAIT	Data Output Address	Data Output Size	Set Decode Output Wait address and size
JPEG_IOCTL_GET_DECOUTPUTWAIT_ADDR	The pointer to store Decode Output Wait Address		Get Decode Output Wait Address
JPEG_IOCTL_GET_DECOUTPUTWAIT_SIZE	The pointer to store Decode Output Wait Size		Get Decode Output Wait Size

Parameter

cmd Command

arg0 First argument of the command

arg1 Second argument of the command

Return Value

None

Example

```
/* Set Downscale to QVGA */
jpegIoctl(JPEG_IOCTL_SET_DECODE_DOWNSCALE, 240, 320);

/* Set Decode Stride to Panel width (480 pixel)*/
jpegIoctl(JPEG_IOCTL_SET_DECODE_STRIDE, 480, 0);

/* Set Decoded Image Address */
jpegIoctl(JPEG_IOCTL_SET_YADDR, u32FrameBuffer, 0);

/* Set Bit stream Address */
 jpegIoctl(JPEG_IOCTL_SET_BITSTREAM_ADDR, u32BitStream, 0);
```

```
/* Set Decode Input Wait mode (Input wait buffer is 8192) */
jpegIoctl(JPEG_IOCTL_SET_DECINPUTWAIT_CALBACKFUN, (UINT32)
JpegDecInputWait, 8192);

 /* Decode mode */
jpegIoctl(JPEG_IOCTL_SET_DECODE_MODE,
JPEG_DEC_PRIMARY_PACKET_YUV422, 0);

/* Set JPEG Header Decode End Call Back Function */
jpegIoctl(JPEG_IOCTL_SET_HEADERDECODE_CALBACKFUN, (UINT32)
JpegDecHeaderComplete, 0);

/* Trigger JPEG decoder */
jpegIoctl(JPEG_IOCTL_DECODE_TRIGGER, 0, 0);

/* Set Source Y/U/V Stride */
jpegIoctl(JPEG_IOCTL_SET_YSTRIDE, u16Width, 0);
jpegIoctl(JPEG_IOCTL_SET_USTRIDE, u16Width/2, 0);
jpegIoctl(JPEG_IOCTL_SET_VSTRIDE, u16Width/2, 0);

/* Primary Encode Image Width / Height */
jpegIoctl(JPEG_IOCTL_SET_DIMENSION, u16Height, u16Width);

/* Encode upscale 2x */
jpegIoctl(JPEG_IOCTL_SET_ENCODE_UPSCALE, u16Height * 2, u16Width * 2);

/* Set Encode Source Image Height */
jpegIoctl(JPEG_IOCTL_SET_SOURCE_IMAGE_HEIGHT, u16Height, 0);

/* Include Quantization-Table and Huffman-Table */
jpegIoctl(JPEG_IOCTL_ENC_SET_HEADER_CONTROL, JPEG_ENC_PRIMARY_QTAB
| JPEG_ENC_PRIMARY_HTAB, 0);

/* Use the default Quantization-table 0, Quantization-table 1 */
jpegIoctl(JPEG_IOCTL_SET_DEFAULT_QTAB, 0, 0);
```

Note [0]

8 bits Quantization-Table Adjustment and control value.

7	6	5	4	3	2	1	0
P_QADJUST				**P_QVS**			

Bits	Descriptions	
[7:4]	P_QADJUST	**Primary Quantization-Table Adjustment** **If the sum of the position (x, y) of quantization-table is greater than P_QADJUST, the quantization value will be set to 127. Otherwise the value will keep as the original.** **8x8 DCT block: x = 0~7, y = 0~7** **if ((x+y) > P_QADJUST) => Q′ = 127** **else => Q′ = Q**
[3:0]	P_QVS	**Primary Quantization-Table Scaling Control** **Q′ = (P_QVS[3]*2*Q)+(P_QVS[2]*Q)+(P_QVS[1]*Q/2)+(P_QVS[0]*Q/4)**

1.4. Example code

This demo code has sample code for “Normal Encode”, “Encode Upscale”, “Normal Decode”, “Decode Downscale & Stride”, “Decode Input Wait”, and “Decode Output Wait” (write/read from SD Card). Please refer to the JPEG sample code of SDK Non-OS.

2. Revision History

Version	Date	Description
V1	Oct. 21, 2011	▪ Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.