XR872 JPEG Developer Guide

Revision 1.1

Nov 04, 2019

Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADIO TECHNOLOGY ("XRADIO"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADIO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE PURCHASED PRODUCTS, SERVICES AND FEATURES ARE STIPULATED BY THE CONTRACT MADE BETWEEN XRADIO AND THE CUSTOMER. PLEASE READ THE TERMS AND CONDITIONS OF THE CONTRACT AND RELEVANT INSTRUCTIONS CAREFULLY BEFORE USING, AND FOLLOW THE INSTRUCTIONS IN THIS DOCUMENTATION STRICTLY. XRADIO ASSUMES NO RESPONSIBILITY FOR THE CONSEQUENCES OF IMPROPER USE (INCLUDING BUT NOT LIMITED TO OVERVOLTAGE, OVERCLOCK, OR EXCESSIVE TEMPERATURE).

THE INFORMATION FURNISHED BY XRADIO IS PROVIDED JUST AS A REFERENCE OR TYPICAL APPLICATIONS, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE.

NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADIO.THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Date	Summary of Changes
1.0	2019-08-15	Initial Version
1.1	2019-11-04	Optimize Format

Table 1- 1 Revision History

Contents

Tables

Figures

1 概述

1.1 属性介绍

XR872 JPEG 模块是一款高性能、低功耗、较少资源依赖的编码器，其具备的属性如下。

- 支持 online 模式 640*480@60fps， offline 模式 640*480@30fps
- 支持 online 模式 1280*720@40fps， offline 模式 1280*720@20fps
- 支持最大分辨率为 1920*1088 online/offline 编码
- 支持 1/2 缩小(scaler)后编码，支持抠图(cropwin)后编码
- online 模式支持分块输出功能，可减小 sram 开销，并提高带宽利用率
- 支持非常规分辨率的 online/offline(如:192*192, 304*224)编码
- 支持常规分辨率图像的 online/offline 编码

1.2 基本功能

1.2.1 编码格式

XR872 JPEG 模块只支持 YUV420 格式的编码，因此 CSI 捕获的数据输出给 JPEG 模块编码的图像格式必须是 YUV420。若 CSI 输入 JPEG 模块是 JPEG 码流，JPEG 模块也能正常将其输出。

1.2.2 编码模式

XR872 JPEG 模块支持 online 及 offline 模式编码。

online 模式即在线模式， CSI 每接收到 16 行数据就自动进行 JPEG 编码，当前帧图像接收完，编码也随即完成。 该模式 CSI 不会将接收的原始图像数据保存起来，只输出 JPEG 编码后的数据。编码数据输出的方式又有：整帧模式和分块模式。

offline 模式即离线模式， CSI 接收到的数据会先存到内存中，待一帧完整数据全部存储完成后，由软件启动 JPEG 编码。 所以此时 JPEG 不是实时处理，可以对任何已经保存好的 YUV420 图像数据进行编码。

1.2.2.1 online 模式

Online 模式的通路框图如图 1-1 所示。

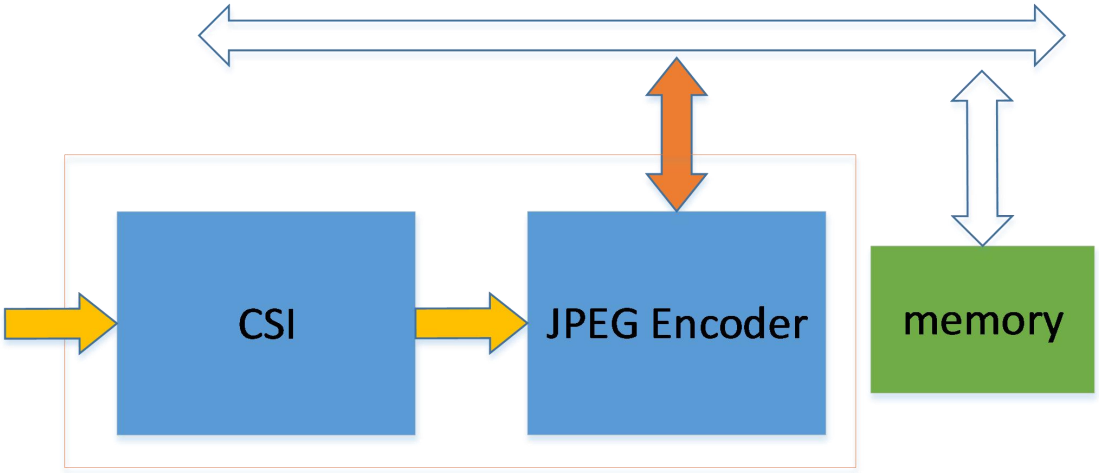

图 1-1 online 模式框图

Sensor(摄像头)输出 YUV422 格式数据到 CSI， CSI 接收到 YUV422 后处理成 YUV420 格式，每接收到 16 行数据后，硬件会自动启动 JPEG encoder 进行一次编码操作，编码输出的码流通过总线直接写到设定好的内存中，故可认为 Online 模式下图像的接收和编码是同时进行的。在一帧数据接收完并编码结束后， JPEG encoder 会产生 ve finish(编码完成)中断。因此，对图像分辨率的要求是行列数为 16 的整数倍，支持的最小分辨率为 32*32。

Online 分块模式与整帧模式的区别在于，分块模式可以在 JPEG 编码输出数据量达到设定值(例如 2KB/4KB)后产生中断，并且可以在一帧编码过程中循环使用编码输出空间，例如只分配 8KB 的编码输出空间，而一帧图像编码数据有 20KB，则在第一次写满 8KB 后， JPEG 将会从这 8KB 的首地址开始存储，循环使用，故需要软件配合将之前的数据读走，否则之前的数据会被覆盖。

1.2.2.2 offline 模式

Offline 模式的通路框图如图 1-2 所示。

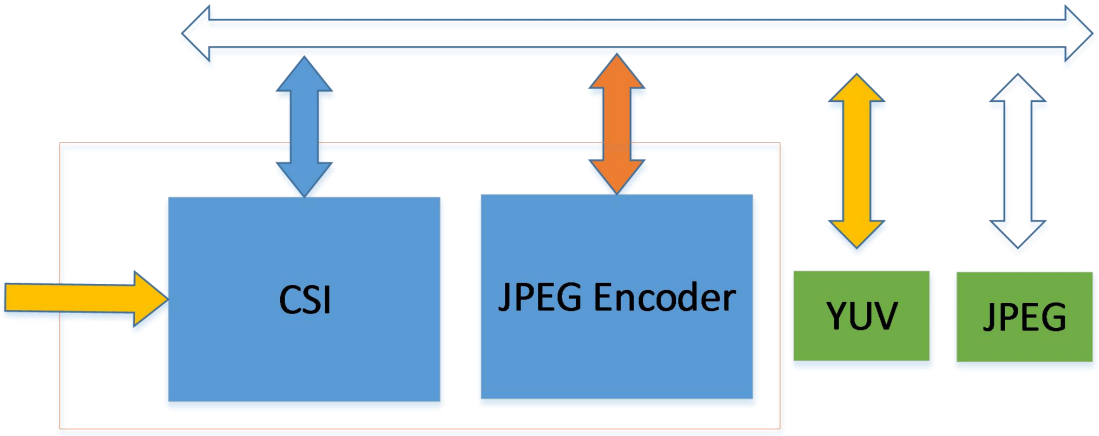

图 1-2 offline 模式框图

Offline 模式下， CSI 会将 YUV420 的原始图像数据存储到 YUV memory 中，存放格式为 NV12。一帧图像

全部存完后，产生写回中断(wb finish)，然后由软件启动 JPEG 开始编码， JPEG 编码器 会读取 YUV memory 中的原始数据送给 Encoder 进行编码，编码后的数据写到 JPEG memory 中。

1.3 代码位置

1.3.1 模块源代码位置

表 1-1 模块源代码位置

文件名	位置
hal_csi_jpeg.c、	sdk/src/driver/chip/
hal_csi_jpeg.h	sdk/include/driver/chip/
jpegenc.c 、 jpeglib.c 、 jpeglib.h 、 jpeg_marker.h	sdk/src/jpeg
jpegenc.h	sdk/include/jpeg

1.3.2 模块示例代码位置

表 1-2 模块示例代码位置

文件名	位置
main.c	sdk/project/example/jpeg/main.c
board_config.c	sdk/project/common/board/xradio_evb/board_config.c

2 模块结构体

2.1 CSI_JPEG_Priv 结构体

CSI_JPEG_Priv 结构体主要保存了 CSI/JPEG 模块的配置信息。

```
typedef struct {
    CSI_CapType       capMode;//CSI 的捕获模式，0：图像模式，1：视频模式
    JPEG_Mode         encMode;//JPEG 的工作模式：online 或 offline
    uint8_t *         jpgOutStmAddr;//JPEG 数据的输出起始地址
    uint8_t           jpgOutStmSize;//JPEG 数据的输出起始地址
    uint8_t           jpgOutBufNum;//JPEG 数据缓存的 buff 数量
    uint32_t          jpgMemSize;//JPEG 数据存放的内存大小
    uint8_t              memPartEn;//JPEG 分块模式的使能
    uint8_t           memPartNum;//JPEG 分块模式的块数量
    uint32_t          memPartSize;//JPEG 分块模式的块大小
    uint32_t          memPartCnt;//JPEG 分块模式的块累计数量
    uint32_t          memPartOffSet;//JPEG 分块模式下所在块的偏移
    uint32_t          memCurSize;//JPEG 数据拷贝的当前大小
    uint8_t           *memPartDstBuf;//JPEG 分块模式数据拷贝的目的地址
    uint8_t           jpgVeEn;//JPEG 使能控制
    uint8_t           capRun;//CSI 捕获的使能状态
    CSI_State         state;//CSI/JPEG 的运行状态
    CSI_JPEG_IRQCallback cb;//CSI/JPEG 的硬件中断的回调函数
    uint32_t          ve_finish_count;//JPEG 数据编码完成的数量
    JpegCtx           *jpegCtx;
} CSI_JPEG_Priv;
```

2.2 CSI_ConfigParam 结构体

CSI_ConfigParam 结构作为对 CSI 模块的配置参数。

```
typedef struct {
    CSI_YUV420Mask       yuv420_mask;//表示 YUV420 的奇偶行掩码
    CSI_OutputMode       out_mode;//表示 CSI 的输出数据格式
    CSI_YUV420LineOrder yuv420_line_order;//表示 CSI 输出的 YUV420 的排列类型
    CSI_InputSeq     input_seq;//表示 CSI 输入的图像排列类型
    CSI_InputFmt     input_fmt;//表示 CSI 输入的图像格式
    CSI_SignalPol    vref_pol;//表示 CSI 帧信号的极性
    CSI_SignalPol    href_pol;//表示 CSI 行信号的极性
    CSI_SignalPol    clk_pol;//表示 CSI 像素点信号的极性
    CSI_SyncType     sync_type;//表示 CSI 同步的类型

    uint16_t         hor_len;//CSI 捕获的图像行长度
    uint16_t         hor_start;//CSI 捕获的图像行起始
```

```
    uint16_t          ver_len;//CSI 捕获的图像列长度
    uint16_t          ver_start;//CSI 捕获的图像列起始
} CSI_ConfigParam;

    CSI_State         state;//CSI/JPEG 的运行状态
    CSI_JPEG_IRQCallback cb;//CSI/JPEG 的硬件中断的回调函数
    uint32_t          ve_finish_count;//JPEG 数据编码完成的数量
    JpegCtx           *jpegCtx;
} CSI_JPEG_Priv;
```

2.3 JPEG_ConfigParam 结构体

JPEG_ConfigParam 结构作为对 JPEG 模块的配置参数。

```
typedef struct {
    uint8_t       jpeg_mode;//JPEG 的工作模式
    uint8_t       sensor_out_type;//JPEG 的输入图像格式

    uint32_t      csi_output_addr_y;//CSI 图像 Y 分量的输出地址
    uint32_t      csi_output_addr_uv;//CSI 图像 UV 分量的输出地址

    uint32_t      pic_size_width;//图像的宽
    uint32_t      pic_size_height;//图像的高

    uint32_t      jpe_input_addr_y;//JPEG Y 分量的输入地址
    uint32_t      jpe_input_addr_uv;//JPEG UV 分量的输入地址

    uint32_t      outstream_start_addr;//JPEG 图像的输出起始地址
    uint32_t      outstream_end_addr;//JPEG 图像的输出结束地址
    uint32_t      outstream_offset;//JPEG 图像的输出的偏移
    uint32_t      outstream_mem_size;//JPEG 存放图像的内存大小
    uint32_t      outstream_buff_num;//JPEG 存放图像的 buff 数量

    uint32_t      quality;//JPEG 图像的质量等级（0~99）

    uint8_t           jpeg_en;//JPEG 使能配置
    uint8_t       mem_part_en;//JPEG 分块模式使能配置
    JPEG_MemPartNum mem_part_num;//JPEG 分块模式的块数量
    uint8_t           *mem_part_buf;//JPEG 分块模式的数据拷贝目的地址
} JPEG_ConfigParam;
```

3 模块接口

3.1 CSI/JPEG 初始化

HAL_Status HAL_CSI_JPEG_Init(CSI_JPEG_InitParam *param)	
功能	CSI/JPEG 模块的初始化，主要是硬件时钟等的配置 。
参数	param：初始化配置的参数。
返回值	成功：HAL_OK 失败：HAL_INVALID，HAL_ERROR

3.2 CSI/JPEG 反初始化

HAL_Status HAL_CSI_JPEG_Deinit(void)	
功能	反初始化 CSI/JPEG 模块。
参数	无
返回值	成功：HAL_OK 失败：HAL_ERROR

3.3 CSI 模块配置

HAL_Status HAL_CSI_Config(CSI_ConfigParam *cfg)	
功能	CSI 模块配置。
参数	cfg：指向 CSI_ConfigParam 类型的配置参数，具体结构体说明可以参见章节 2。
返回值	成功：HAL_OK 失败：HAL_ERROR

3.4 JPEG 模块配置

HAL_Status HAL_JPEG_Config(JPEG_ConfigParam *cfg)	
功能	JPEG 模块配置。
参数	cfg：指向 JPEG_ConfigParam 类型的配置参数，具体结构体说明可以参见章节 2。
返回值	成功：HAL_OK 失败：HAL_ERROR
备注	重新配置 JPEG 模块时，需要先调用 HAL_JPEG_Reset 接口进行 reset 操作。

3.5 CSI/JPEG 模块重启

void HAL_JPEG_Reset(void)	
功能	reset CSI/JPEG 模块。
参数	无
返回值	无
备注	当 CSI/JPEG 运行出现“excption”时，需要先调用该接口进行 reset，然后重新配置 CSI/JPEG 模块，才能再次去捕获图像。

3.6 JPEG 写帧头

void HAL_JPEG_WriteHeader(uint8_t *baseAddr)	
功能	给 JPEG 图像进行帧头信息的编写。
参数	baseAddr：指向 JPEG 图像帧头信息的基地址，应用需要提供至少 623bytes 内存进行帧头的编写。
返回值	无
备注	JPEG 模块输出的数据是 JPEG 裸数据，需要添加帧头信息才能正确显示出图像来。

3.7 CSI/JPEG 开启图像捕获

HAL_Status HAL_CSI_StartCapture(CSI_CapType mode)	
功能	CSI/JPEG 启动图像的捕获。
参数	mode：捕获的模式，可以是静态图像模式或是视频模式。
返回值	成功：HAL_OK 失败：HAL_ERROR

3.8 CSI/JPEG 停止图像捕获

HAL_Status HAL_CSI_StopCapture(void)	
功能	CSI/JPEG 停止图像的捕获。
参数	无
返回值	成功：HAL_OK 失败：HAL_ERROR
备注	图像模式的捕捉，当一帧图像捕获完毕，CSI 则停止捕获，所以该接口更多是应用于视频模式。

4 模块使用示例

4.1 模块使用流程

具体代码可参考 sdk/project/example/jpeg/main.c。