

# 一、wifi\_camera目录下的C文件说明

## ( 1 ) apps/wifi\_camera目录下

①apps/wifi\_camera

app\_database.c ：主要是保存在flash vm区的数据。

app\_main.c：系统任务名字和优先级、堆栈空间配置，应用层入口等。

get\_yuv\_data.c：获取摄像头的YUV数据。

jpeg2yuv.c：jpeg解码成YUV

simple\_avi\_unpkg.c：AVI简单解码器

spi\_video.c：SPI摄像头

usb\_hid\_keyboard.c：USB HID键盘

video\_rec.c：SD卡录像的事件处理和视频打开关闭处理接口

yuv2jpeg\_thumbnail.c：YUV编码大分辨率的JPEG，而且带上小分辨率的缩略图

②apps/wifi\_camera/board/wl82或者apps/wifi\_camera/board/wl80

board\_config.h：板级配置，选择不同的板级，和板级的一些配置。

board\_xxx.c：板级配置，包括格式外设和设备、打印等。

③apps/wifi\_camera/include，公共头文件

④apps/wifi\_camera/ui\_camera\_demo

camera\_demo.c：UI的摄像头测试例子

camera\_lcd\_only\_show\_demo：摄像头显示到屏幕例子

key\_demo.c：UI的按键例子

ui\_demo.c ：UI主工程例子

④apps/wifi\_camera/wifi

ctp\_cmd.c：与APP通信的CTP命令

ftpd\_vfs\_reg.c：FTP的文件操作接口，主要用在OTA升级

net\_fs\_update.c：网络升级，与ftpd\_vfs\_reg.c配合使用

net\_video\_rec.c：与APP图传的事件处理和视频打开关闭处理接口

network\_mssdp.c：量产测试配置，详情可看：量产测试说明文档

stream\_protocol.c：视频流的协议转发，与user\_video\_rec.c、stream/video\_rt\_xxx.c配合使用

user\_video\_rec.c：用户自定义的视频操作接口，与stream\_protocol.c、stream/video\_rt\_xxx.c配合使用

wifi\_app\_task.c：wifi启动和wifi的速率配置等。

stream/video\_rt\_tcp.c：与APP通信的TCP socket初始化、数据传输、关闭等

stream/video\_rt\_udp.c：与APP通信的UDP socket初始化、数据传输、关闭等

stream/video\_rt\_usr.c：用户自定义协议的初始化、数据传输、关闭等

# 二、app\_config.h配置宏说明

## ( 1 ) 视频相关宏

```
#define CONFIG_OSC_RTC_ENABLE //外接RTC32768使能
#define CONFIG_OSD_ENABLE //视频图像的时间OSD使能
#define CONFIG_VIDEO_REC_PPBUF_MODE //视频应用层的buffer 乒乓模式，屏蔽则使用lbuf模式
#define CONFIG_VIDEO_SPEC_DOUBLE_REC_MODE //视频双录使能功能，VGA+VGA/720P
#define RTOS_STACK_CHECK_ENABLE //启用定时检查任务栈打印

#define CONFIG_ROOT_PATH CONFIG_STORAGE_PATH"/C/" //sd卡根目录路径
#define CONFIG_REC_PATH_0 CONFIG_STORAGE_PATH"/C/"CONFIG_REC_DIR_0//sd0录像路径

#define CONFIG_UDISK_STORAGE_PATH "storage/udisk0" //USB主机U盘盘符路径
```

```
#define CONFIG_UDISK_ROOT_PATH          CONFIG_UDISK_STORAGE_PATH"/C/"//USB主机U盘盘符根目录路径

#define CONFIG_UVC_VIDEO2_ENABLE  //usb主机UVC模式使能
#define TCFG_UDISK_ENABLE          0  //1主机读取U盘功能使能，0不使用
#define TCFG_PC_ENABLE             0//1打开连接PC电脑模式，0不使用

#define CONFIG_VIDEO_720P  //使用720p摄像头，屏蔽则是VGA摄像头
#define VIDEO_REC_AUDIO_SAMPLE_RATE      8000 //视频流的音频采样率,注意：硬件没MIC则为0
#define VIDEO_REC_FPS                    20   //录像SD卡视频帧率设置,0为默认
#define CONFIG_USR_VIDEO_ENABLE          //用户VIDEO使能，自定义接收数据流，详情demo_video
```

( 2 ) 音频相关宏

```
#define VIDEO_REC_AUDIO_SAMPLE_RATE      8000 //视频流的音频采样率,注意：硬件没MIC则为0
```

音频配置说明部分详情demo\_audio\使用说明的 [DEMO AUDIO readme.md](#) .

( 3 ) 网络相关宏

```
#define CONFIG_MASS_PRODUCTION_ENABLE  //量产测试模式使能（STA wifi出图）

#define AP_WIFI_CAM_PREFIX      "wifi_camera_wl8x_"//AP模式热点名字（后面跟随mac地址）
#define AP_WIFI_CAM_WIFI_PWD   "12345678"//AP模式的默认密码

#define CONFIG_WIFI_STA_MODE          //打开:STA连接固定路由器
#define STA_WIFI_SSID                "GJ1" //STA连接路由器名称，量产模式的路由器名称
#define STA_WIFI_PWD                  "8888888899" //STA连接路由器密码，也为量产模式的路由器密码

#define CONFIG_NET_UDP_ENABLE //与DVrunning2的APP图传协议才有UDP（UDP和TCP只能使用一种）
//#define CONFIG_NET_TCP_ENABLE //与DVrunning2的APP图传协议才有TCP（UDP和TCP只能使用一种）

/*-----与APP数据传输的网络端口-----*/
#define CTP_CTRL_PORT      3333
#define CDP_CTRL_PORT     2228
#define NET_SCR_PORT       2230
#define VIDEO_PREVIEW_PORT 2226
#define VIDEO_PLAYBACK_PORT 2223
#define HTTP_PORT          8080
#define RTSP_PORT          554

/*-----与APP图传的视频库内使用-----*/
#define _DUMP_PORT         2229
#define _FORWARD_PORT      2224
#define _BEHIND_PORT       2225

#define NET_VIDEO_BUFF_FRAME_CNT      2 //TCP和UDP图传缓冲区帧数，超过则丢弃
#define NET_VIDEO_REC_FPS0            20//前视图传帧率，图传帧率与摄像头输出原始帧率有关
#define NET_VIDEO_REC_FPS1            20 //后视图传帧率（暂不使用）
```

( 4 ) 其他相关宏

```
#define CONFIG_DOUBLE_BANK_ENABLE      1 //双备份升级方式，详情example/upgrade
#define CONFIG_UPGRADE_FILE_NAME       "update.ufw" //升级文件名字
#define CONFIG_UPGRADE_PATH            //升级文件路径
```

三、 board\_config.h配置宏说明

( 1 ) SD、KEY、RTC、音频接口等配置

```
#ifndef CONFIG_BOARD_7916A
#define __FLASH_SIZE__      (8 * 1024 * 1024)
#define __SDRAM_SIZE__      (8 * 1024 * 1024)

//#define CONFIG_VIDEO1_ENABLE          //spi video

#define TCFG_SD0_ENABLE          1
#define TCFG_ADKEY_ENABLE       1 //AD按键
#define CONFIG_OSC_RTC_ENABLE   //RTC时钟开关
//#define CONFIG_PRESS_LONG_KEY_POWERON //长按开关机功能
```

```
#define TCFG_DEBUG_PORT          IO_PORTB_08
#define TCFG_DAC_MUTE_PORT      0xff
#define TCFG_DAC_MUTE_VALUE     0

#define TCFG_SD_PORTS            'A'          //SD0/SD1的ABCD组(默认为开发板SD0-D),注意:IO占用问题
#define TCFG_SD_DAT_WIDTH       4            //1:单线模式, 4:四线模式
#define TCFG_SD_DET_MODE        SD_CLK_DECT  //检测模式:命令检测, 时钟检测, IO检测
#define TCFG_SD_DET_IO          IO_PORTB_08  //SD_DET_MODE为SD_IO_DECT时有效
#define TCFG_SD_DET_IO_LEVEL    0           //IO检卡上线的电平(0/1),SD_DET_MODE为SD_IO_DECT时有效
#define TCFG_SD_CLK              30000000    //SD时钟

//*****//
//                                AUDIO_ADC应用的通道配置                                //
//*****//
//#define CONFIG_AUDIO_ENC_SAMPLE_SOURCE      AUDIO_ENC_SAMPLE_SOURCE_MIC
// #define CONFIG_AUDIO_ENC_SAMPLE_SOURCE      AUDIO_ENC_SAMPLE_SOURCE_PLNK0
#if CONFIG_AUDIO_ENC_SAMPLE_SOURCE == AUDIO_ENC_SAMPLE_SOURCE_PLNK0
#define TCFG_MIC_CHANNEL_MAP      LADC_CH_MIC3_P_N
#define TCFG_MIC_CHANNEL_NUM      1
#else
#define TCFG_MIC_CHANNEL_MAP      (LADC_CH_MIC0_P_N | LADC_CH_MIC1_P_N | LADC_CH_MIC3_P_N)
#define TCFG_MIC_CHANNEL_NUM      3
#endif
#define TCFG_LINEIN_CHANNEL_MAP    (LADC_CH_AUX1 | LADC_CH_AUX3)
#define TCFG_LINEIN_CHANNEL_NUM    2

#define CONFIG_AISP_DIFFER_MIC_REPLACE_LINEIN //用差分mic代替aec回采
#define CONFIG_ASR_CLOUD_ADC_CHANNEL 1      //云端识别mic通道
#define CONFIG_VOICE_NET_CFG_ADC_CHANNEL 1   //声波配网mic通道
#define CONFIG_AISP_MIC0_ADC_CHANNEL 1       //本地唤醒左mic通道
#define CONFIG_AISP_MIC_ADC_GAIN      80     //本地唤醒mic增益
#define CONFIG_AISP_LINEIN_ADC_CHANNEL 3      //本地唤醒LINEIN回采DAC通道
#define CONFIG_AISP_MIC1_ADC_CHANNEL 0       //本地唤醒右mic通道
#define CONFIG_REVERB_ADC_CHANNEL      1      //混响mic通道
#define CONFIG_PHONE_CALL_ADC_CHANNEL 1      //通话mic通道
#define CONFIG_UAC_MIC_ADC_CHANNEL     1      //UAC mic通道
#define CONFIG_AISP_LINEIN_ADC_GAIN     10     //本地唤醒LINEIN增益
#endif

//*****//
//                                SD 配置(暂只支持打开一个SD外设)                                //
//*****//
//SD0  cmd,  clk,  data0, data1, data2, data3
//A    PB6  PB7  PB5   PB5   PB3   PB2
//B    PA7  PA8  PA9   PA10  PA5   PA6
//C    PH1  PH2  PH0   PH3   PH4   PH5
//D    PC9  PC10 PC8   PC7   PC6   PC5

//SD1  cmd,  clk,  data0, data1, data2, data3
//A    PH6  PH7  PH5   PH4   PH3   PH2
//B    PC0  PC1  PC2   PC3   PC4   PC5

#if (TCFG_SD0_ENABLE || TCFG_SD1_ENABLE)
#if (defined CONFIG_BOARD_7911B_DEVELOP_AERIAL)
#define TCFG_SD_PORTS            'D'          //SD0/SD1的ABCD组(默认为开发板SD0-D,用户可针对性更改,注意:IO占用问题)
#define TCFG_SD_DAT_WIDTH       1            //1:单线模式, 4:四线模式
#define TCFG_SD_DET_MODE        SD_CMD_DECT  //检测模式
#define TCFG_SD_DET_IO          IO_PORTA_01  //SD_DET_MODE为SD_IO_DECT时有效
#define TCFG_SD_DET_IO_LEVEL    0           //IO检卡上线的电平(0/1),SD_DET_MODE为SD_IO_DECT时有效
#define TCFG_SD_CLK              24000000    //SD时钟
#endif
#endif
#endif
```

## 四、图传、录卡或者UVC显示的摄像头-高帧率说明

摄像头高帧率：20帧以上，则需要修改工程配置才能达到高帧率传输或显示，以下修改为：**wifi\_camera**工程，其他工程大同小异。

### ( 1 ) 修改app\_config.h的VIDEO\_REC buffer模式和输出帧率

注释乒乓buff模式，恢复使用lbuf（lbuf详情 [readme.md](#) ），如下：

```
//（1）buffer模式：注释CONFIG_VIDEO_REC_PPBUF_MODE。
//#define CONFIG_VIDEO_REC_PPBUF_MODE

//（2）图传或UVC显示的输出帧率：如设置25。
#define NET_VIDEO_REC_FPS0 25 //不使用0作为默认值，写具体数值

//（3）音频数据：有则采样率为8000，没有则为0，如下为0。
#define VIDEO_REC_AUDIO_SAMPLE_RATE 0

//（4）SD卡录像的帧率：没有SD卡则0或不修改，如25帧。
#define VIDEO_REC_FPS 25
```

（2）确定工程没有开CONFIG\_NO\_SDRAM\_ENABLE宏，工程使用sdram

windows环境，coldblocks：打开cbp工程，右键工程，build options--->compiler--->#define选项，查看是否含有：CONFIG\_NO\_SDRAM\_ENABLE，有则说明不需要sdam，没有则需要sdram。在高帧率情况下，都是需要sdram，内存才足够，否则是无法满足高帧率。因此需要去掉：CONFIG\_NO\_SDRAM\_ENABLE。

linux环境，makefile：关闭CONFIG\_NO\_SDRAM\_ENABLE宏。

（3）修改video\_buf\_config.h的应用层视频缓冲区大小

video\_buf\_config.h对应内存配置说明如下，一般修改图传和UVC显示：NET\_VREC0\_FBUF\_SIZE为300K即可。

```
#define VIDEO0_REC_FORMAT VIDEO_FMT_AVI
#define VIDEO1_REC_FORMAT VIDEO_FMT_AVI
#define VIDEO2_REC_FORMAT VIDEO_FMT_AVI
#ifdef CONFIG_PSRAM_ENABLE
#define VREC0_FBUF_SIZE (512*1024)
#define VREC1_FBUF_SIZE (256*1024)
#define VREC2_FBUF_SIZE (256 * 1024)
#define AUDIO_BUF_SIZE (256*1024)
#else
#if (!defined CONFIG_NO_SDRAM_ENABLE && __SDRAM_SIZE__ >= (2 * 1024 * 1024))
#define VREC0_FBUF_SIZE (500*1024)//开sdram，SD卡录像视频缓冲区
#define VREC1_FBUF_SIZE (0)
#define VREC2_FBUF_SIZE (0)
#define AUDIO_BUF_SIZE (64*1024)//开sdram，SD卡录像音频缓冲区，没有音频写0
#else
#define VREC0_FBUF_SIZE (150*1024)//不开sdram，SD卡录像修改这里
#define VREC1_FBUF_SIZE (0)
#define VREC2_FBUF_SIZE (0)
#define AUDIO_BUF_SIZE (16*1024)//不开sdram，SD卡录像音频缓冲区，没有音频写0
#endif
#endif

#define USB_CAMERA_BUF_SIZE (1 * 1024 * 1024) // + 512 * 1024)
#define CAMERA_CAP_BUF_SIZE (1 * 1024 * 1024)

#if (!defined CONFIG_NO_SDRAM_ENABLE && __SDRAM_SIZE__ >= (2 * 1024 * 1024))
#define NET_VREC0_FBUF_SIZE (200*1024)//开sdram，图传或UVC显示视频缓冲区
#define NET_VREC1_FBUF_SIZE (0)
#define NET_AUDIO_BUF_SIZE (64*1024)//开sdram，图传或UVC显示音频缓冲区，没有音频写0
#else
#define NET_VREC0_FBUF_SIZE (100*1024)//不开sdram，图传或UVC显示视频缓冲区
#define NET_VREC1_FBUF_SIZE (0)
#define NET_AUDIO_BUF_SIZE (16*1024)//开sdram，图传或UVC显示音频缓冲区，没有音频写0
#endif
```

- ①当使用SD卡录像则可以修改：VREC0\_FBUF\_SIZE 和 AUDIO\_BUF\_SIZE大小；当需要提高视频帧率或者视频卡顿现象，改大：VREC0\_FBUF\_SIZE（一般VGA 500K，720P 800K）；当需要提高音频卡顿现象，改大：AUDIO\_BUF\_SIZE（一般128K）。
- ②当使用网络图传或UVC显示则可以修改：NET\_VREC0\_FBUF\_SIZE 和 NET\_AUDIO\_BUF\_SIZE大小；当需要提高视频帧率或者视频卡顿现象，改大：NET\_VREC0\_FBUF\_SIZE（一般VGA 300K，720P 500K）；当需要提高音频卡顿现象，改大：NET\_AUDIO\_BUF\_SIZE（一般128K）。
- ③确定user\_video\_rec.c有没有添加到工程，wifi\_camera工程默认不加

当user\_video\_rec.c添加到工程编译，则在文件中修改：USER\_VREC0\_FBUF\_SIZE 大小（一般为300K，使用音频需改善音频卡顿则修改：USER\_AUDIO\_BUF\_SIZE（一般128K），没有音频则USER\_AUDIO\_BUF\_SIZE为0），其他demo则USER\_VREC0\_FBUF\_SIZE在对应的app\_config.h。

（4）修改提高摄像头源视频输出帧率，确定摄像头源输出高帧率

- ①例如DVP-GC0308摄像头驱动设置25帧：

```
#define CONFIG_INPUT_FPS 25
```

注意：摄像头输出帧率需要确保是源头帧率，不能只修改CONFIG\_INPUT\_FPS的值，如原本驱动只能15帧，修改驱动软件的CONFIG\_INPUT\_FPS为25是不能提高摄像头帧率的，需要修改摄像头寄存器配置。

②UVC摄像头则默认20-25帧，具体需要查看UVC摄像头数据手册如何更改帧率，一般不修改。

### （5）使用网络图传则更换lwip库和wifi库

使用网络进行视频传输（图传功能）则：

①lwip\_2\_1\_2\_sfc.a换lwip\_2\_1\_2.a；

②wl\_wifi\_ap\_sfc.a换wl\_wifi\_ap.a；

或使用STA的wl\_wifi\_sta\_sfc.a换wl\_wifi\_sta.a；

或者wl\_wifi\_sfc.a换wl\_wifi.a。

### （6）在图传中，需要提高图片质量

在 [net\\_video\\_rec.c](#) 修改码率，适当提高码率（1000-6000），如下。

```
/*码率控制，根据具体分辨率设置*/
static int net_video_rec_get_abr(u32 width)
{
    /*视频码率kbps使用说明：
    码率：一帧图片以K字节为单位大小 * 帧率 * 8，比如：一帧图片为30KB，帧率为20帧，则码率为30*20*8=4800
    VGA图片大小说明：低等质量(小于20K)，中等质量(20K-40K)，高质量(大于40K，极限70K)
    720P图片大小说明：低等质量(小于50K)，中等质量(50k-100K)，高质量(大于100K，极限150K)
    */

    if (width <= 640) {
        return 1800;
    } else {
        return 1000;
    }
}
```

如果是SD卡录像，则修改码率在 [video\\_rec.c](#) 的video\_rec\_get\_abr()函数。

## 五、图传卡顿问题

### wifi图传卡顿问题分析步骤

- （1）先确定SDK的buf已经优化，修改SDK的步骤详情上一章：**四、图传、录卡或者UVC显示的摄像头-高帧率说明**
  - （2）卡顿问题，确保天线匹配、PA参数、晶振频偏校准值都已经配置完好，先确保这些参数没有问题，才能往下一步看看。
  - （3）在确保PA参数和晶振频偏参数配置好之后，增加发送设功率，包括每个速率功率和总功率。
  - （4）确保上述3个步骤完成，若还是卡顿则：确定是不是干扰问题，可以先把路由器信道换到干扰比较少的信道。
- 以上4步骤基本上可以解决，如果上述步骤都没有解决，有可能是芯片问题，可以换芯片试试。在测试多个芯片还是没有解决，则说明干扰过大，无法解决！

