

# IMB 模块使用说明

## 1.模块介绍

IMB 模块为彩屏控制器，可以支持 Smart Panel、Dump Panel 类型的彩屏，数据口可以支持 1bit,3bit,6bit,8bit,9bit,16bit,18bit,24bit 的数据宽度，可以同时支持多图层显示,带 alpha 参数。IMB 模块是按帧操作的，对于 Smart Panel 类型的彩屏，需要与 PAP 模块(即 EMI 模块)配合使用，PAP 模块负责配置彩屏的控制器，完成初始化工作，初始化完成以后就可以启用 IMB 模块，只负责写数据。

## 2.特性

支持 3 位、6 位、8 位、9 位、16 位、18 位、24 位的位宽

支持 Parallel RGB 24 Bit、Serial RGB 8Bit、Serial RGB 8Bit + Dummy、YUV640、YUV720、CCIR656、RAW DATA 接口

支持 HDMI 输出，需外接 HDMI 发送器

支持 AVOUT 输出(PAL、NTSC 制式)

最大支持分辨率 800x600

64 级透明度可调

2 层图片层(1 层带 MASK 层)，2 层 OSD 层，每层 OSD 有三个窗口，每层可独立开关

DCLK 时钟频率可配

## 3.IMB 函数列表

```
/*-----*/
```

```
/**@brief 初始化 IMB 模块
```

```
@param dev:设备参数
```

```
@return 无
```

```
@note void imb_init(DISP_DEV_CFG* dev);
```

```
*/
```

```
/*-----*/
```

```
void imb_init(DISP_DEV_CFG* dev);
```

```
/*-----*/
```

```
/**@brief 配置 OSD 层的颜色表
```

```
@param osd: OSD 层索引
```

```
@param osd_tab: 颜色表(YUV 格式)
```

```
@param num: 颜色表大小
```

```
@return 无
```

```
@note    void imb_init(DISP_DEV_CFG* dev);
*/
/*-----*/
extern void imb_osd_pal_cfg(u8 osd, const u32 *osd_tab,u16 num);

/*-----*/
/**@brief  获取图片层句柄
    @param  layer:图层序号
    @return 图片层句柄
    @note   IMGLAYER *imb_get_image_hdl(u8 layer);
*/
/*-----*/
IMGLAYER *imb_get_image_hdl(u8 layer);

/*-----*/
/**@brief  图片层配置
    @param  img:图片层句柄
    @return 无
    @note   void imb_image_cfg(IMGLAYER *img);
*/
/*-----*/
void imb_image_cfg(IMGLAYER *img);

/*-----*/
/**@brief  获取 OSD 层句柄
    @param  layer:图层序号
    @return OSD 层句柄
    @note   OSDWIN *imb_get_osd_hdl(u8 layer);
*/
/*-----*/
OSDWIN *imb_get_osd_hdl(u8 layer);

/*-----*/
/**@brief  OSD 层配置
    @param  oshdl:OSD 层句柄
    @return 无
    @note   void imb_osd_win_cfg(OSDWIN *osdhdl);
*/
/*-----*/
void imb_osd_win_cfg(OSDWIN *osdhdl);
```

```
/*-----*/
```

```
/**@brief  启动 IMB 模块  
    @param  无  
    @return 无  
    @note   void imb_start(void);
```

```
*/
```

```
/*-----*/
```

```
void imb_start(void);
```

```
/*-----*/
```

```
/**@brief  打开 IMB 模块  
    @param  无  
    @return 无  
    @note   void imb_open(void);
```

```
*/
```

```
/*-----*/
```

```
void imb_open(void);
```

```
/*-----*/
```

```
/**@brief  关闭 IMB 模块  
    @param  无  
    @return 无  
    @note   void imb_close(void);
```

```
*/
```

```
/*-----*/
```

```
void imb_close(void);
```

以下为初始化 IMB 模块所需要的参数，是跟彩屏类型相关的。

```
typedef struct
```

```
{
```

```
    u16 out_format;      //输出数据格式  
    u8  drive_mode;      //驱动类型  
    u8  data_width;      //数据接口位宽  
    u8  isr_prior;       //中断优先级  
    u8  ncycle;          //输出时钟数(1~4 clk)  
    u8  signal_polarity; //信号极性  
    u8  scan_mode;       //扫描方式 非隔行/时序隔行/数据隔行  
    u8  dclk_freq;       //DCLK 频率选择  
    u32 bkcolor;         //背景颜色  
    u8  *r_gma_tab;      //red gamma 表
```

```
u8 *g_gma_tab;          //green gamma 表
u8 *b_gma_tab;          //blue gamma 表
u8 raw_phase;           //奇偶行初始相位 只对 RAW 模式有效
L0_PARA_CFG *para;      //时钟配置
CGAIN *gain;            //颜色增益
void (*hook) (u16 x,u16 y,u16 w,u16 h); //配置彩屏控制器, 只针对 MCU 接口的屏
}DISP_DEV_CFG;
```

```
typedef struct
{
    u16 h_total_width; //水平时钟总数
    u16 h_sync_width;   //水平同步时钟
    u16 h_act_start;    //水平起始时钟
    u16 h_act_width;    //水平宽度
    u16 v_total_height; //垂直时钟总数
    u16 v_sync_height;  //垂直同步时钟
    u16 v_act_start;    //垂直起始时钟
    u16 v_act_startl;   //垂直起始时钟(隔行模式, 非隔行模式设为 0)
    u16 v_act_width;    //垂直宽度
}L0_PARA_CFG;
```

#### 参数说明:

其中 drive\_mode,data\_width,out\_format 参数根据屏类型选择相应的宏。

#### out\_format:

```
#define FORMAT_RGB333    0x01
#define FORMAT_RGB666    0x02
#define FORMAT_RGB888    0x04
#define FORMAT_RGB565    0x08
#define FORMAT_RAWPOS    0x10    //正向 R->G->B
#define FORMAT_RAWNEG    0x20    //逆向 B->G->R
#define FORMAT_YUV888    0x40
#define FORMAT_YUV422    0x80
#define FORMAT_ITU656    0x180   //YUV422
```

#### drive\_mode:

```
#define MODE_RGB_SYNC    0x00//Dump Panel 硬件接 HSYNC VSYNC 信号
#define MODE_RGB_DE      0x01//Dump Panel 硬件接 DE 信号
#define MODE_MCU          0x02//Smart Panel 自带显存, 单帧显示
#define MODE_HDMI         0x03//HDMI 显示
#define MODE_AVOUT        0x04//AV OUT 显示
#define MODE_CCIR656      0x05//嵌入同步信号(8BITS+DCLK, 不需要 HSYNC、VSYNC 信号)
```

#### data\_width:

```
#define PORT_1BIT      0x00//AV OUT
#define PORT_3BITS     0x01
#define PORT_6BITS     0x03
#define PORT_8BITS     0x07
#define PORT_9BITS     0x0f
#define PORT_16BITS    0x1f
#define PORT_18BITS    0x3f
#define PORT_24BITS    0x7f
```

#### isr\_prior:

imb 模块的中断优先级设置，7 级中断优先级可设，其中 0 为最低优先级，7 为最高优先级

#### ncycle:

```
#define ONE_CYCLE      0x00    //RGB-24BIT、MCU-16BIT
#define TWO_CYCLE      0x01    //MCU-8BIT
#define THREE_CYCLE    0x02    //RGB-8BIT
#define FOUR_CYCLE     0x03    //RGB-8BIT+dummy
```

#### signal\_polarity:

以下宏可组合使用

```
#define DEFAULT_POSITIVE 0x00//默认的 DCLK/HSYNC/VSNC 极性为正向，上升沿有效
#define DCLK_NEGATIVE   0x01//DCLK 负极性，下降沿有效
#define HSYNC_NEGATIVE  0x02//HSYNC 负极性，下降沿有效
#define VSYNC_NEGATIVE  0x04//VSYNC 负极性，下降沿有效
```

#### scan\_mode:

```
#define NON_INTERLACED  0x00//非隔行
#define INTERLACED_TIMING 0x01//时序隔行
#define INTERLACED_DATA  0x02//数据隔行
#define INTERLACED_ALL   0x03//时序数据隔行
```

#### dclk\_freq:

预设常用的 DCLK 频率

```
#define DCLK_6MHZ      0x00
#define DCLK_10MHZ     0x01
#define DCLK_24MHZ     0x02
#define DCLK_27MHZ     0x03
```

#### bkcolor:

背景层颜色设置，当所有图层都关闭时，默认显示背景层颜色

r\_gma\_tab: 红色分量 gamma 表

g\_gma\_tab: 绿色分量 gamma 表

b\_gma\_tab: 蓝色分量 gamma 表

**raw\_phase:**

FORMAT\_RAWPOS 与 FORMAT\_RAWNEG 格式有效，奇行相位和偶行相位组合使用，例如 ODD\_PHASE\_R|EVEN\_PHASE

//奇行相位

```
#define ODD_PHASE_R    0x00
```

```
#define ODD_PHASE_G    0x01
```

```
#define ODD_PHASE_B    0x02
```

//偶行相位

```
#define EVEN_PHASE_R    0x00
```

```
#define EVEN_PHASE_G    0x10
```

```
#define EVEN_PHASE_B    0x20
```

**para:**

配置彩屏的驱动时钟，请参考彩屏相关的技术手册进行设置

**gain:**

调节彩屏色彩，由于每种彩屏的色彩都会有偏差，可通过该参数对颜色进行调整

**hook:**

回调函数，用于设置彩屏的有效窗口区域，Smart Panel 需要实现该回调函数

## 4.函数调用流程

imb\_init -> imb\_osd\_pal\_cfg -> imb\_image\_cfg/imb\_osd\_win\_cfg -> imb\_start