

## BaseParameter 存储格式说明

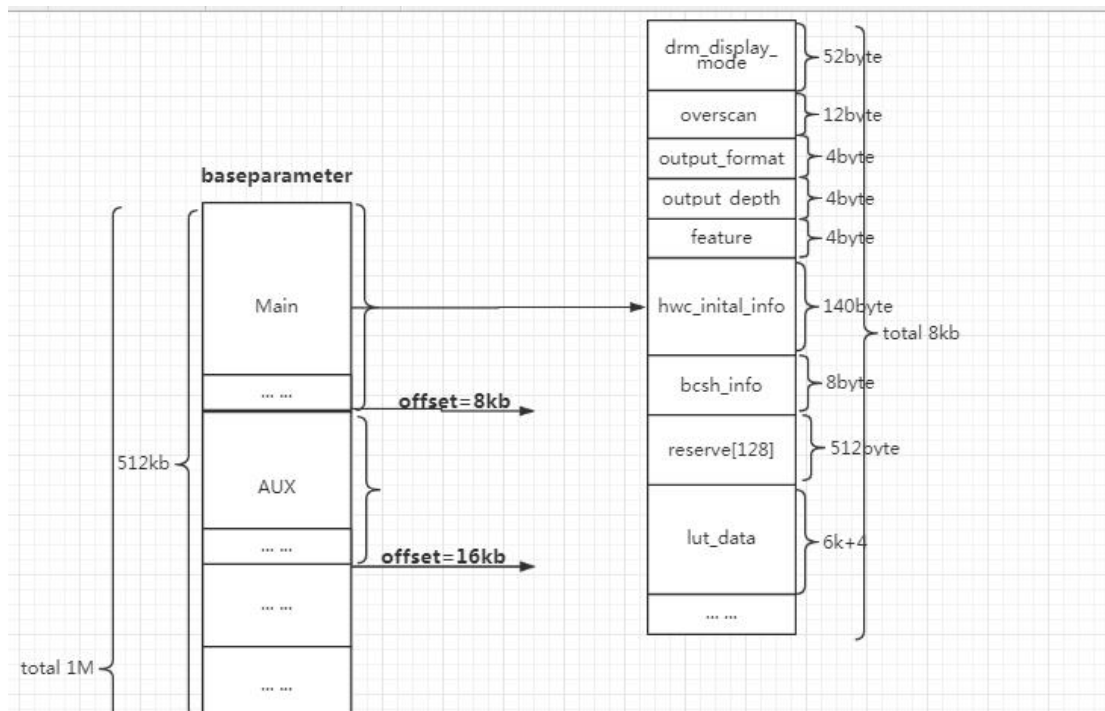
### 1. baseparameter 结构:

```
struct disp_info{
    struct drm_display_mode resolution;
    struct oversean scan;
    enum output_format  format;
    enum output_depth depth;
    unsigned int feature;
    struct hwc_init_info hwc_info;
    struct bcsb_info bcsb;
    unsigned int reserve[128];
    struct lut_data mlutdata;
};

struct file_base_parameter
{
    struct disp_info main;
    struct disp_info aux;
};
```

main,aux 分别保存主副屏信息，main 的信息存储在 baseparameter 前 8kb 的空间内，aux 偏移为 8kb。

disp\_info 的大小不为 8kb，因此 main 和 aux 读取和写入的时候需要分开，lseek 8kb 之后再读写 aux。



## 2. Disp\_info 说明

- `drm_display_mode`:  
存储分辨率的时序信息

```
struct drm_display_mode {  
    /* Proposed mode values */  
    int clock;      /* in kHz */  
    int hdisplay;  
    int hsync_start;  
    int hsync_end;  
    int htotal;  
    int vdisplay;  
    int vsync_start;  
    int vsync_end;  
    int vttotal;  
    int vrefresh;  
    int vscan;  
    unsigned int flags;  
    int picture_aspect_ratio;  
};
```

- `Overscan`:  
存放缩放相关:

```
struct overscan {  
    unsigned int maxvalue;  
    unsigned short leftscale;  
    unsigned short rightscale;  
    unsigned short topscale;  
    unsigned short bottomscale;  
};
```

- `output_format`:  
颜色格式, 属性字符串为 `Auto` 时, 选择 `output_ycbcr_high_subsampling`

```
enum output_format {  
    output_rgb=0,  
    output_ycbcr444=1,  
    output_ycbcr422=2,  
    output_ycbcr420=3,  
};
```

```

output_ycbcr_high_subsampling=4, // (YCbCr444 > YCbCr422 > YCbCr420 > RGB)
output_ycbcr_low_subsampling=5, // (RGB > YCbCr420 > YCbCr422 > YCbCr444)
invalid_output=6,
};

```

- Output\_depth:

色深, 属性字符串为 Auto 时, 选择 Automatic

```

enum output_depth{
    Automatic=0,
    depth_24bit=8,
    depth_30bit=10,
};

```

- Feature:

feature 目前有如下 flag, 配置分辨率以及颜色的 AUTO 模式, 是否开启 hdcp1x, 是否过滤分辨率列表

```

#define RESOLUTION_AUTO          (1<<0)
#define COLOR_AUTO              (1<<1)
#define HDCP1X_EN               (1<<2)
#define RESOLUTION_WHITE_EN     (1<<3)

```

- hwc\_inital\_info:

配置 FrameBuffer 大小, fps, 以及主副屏的挂载设备

```

struct hwc_inital_info{
    char device[128];
    unsigned int framebuffer_width;
    unsigned int framebuffer_height;
    float fps;
};

```

device 例子 : HDMI-A,TV

Fps 可以用来限制帧率

- bcsb\_info :

保存 bcsb 信息, 取值范围 0~100, 默认值都是 50

```

struct bcsb_info {
    unsigned short brightness;
};

```

```
    unsigned short contrast;  
    unsigned short saturation;  
    unsigned short hue;  
};
```

- Reserve:

```
    unsigned int reserve[128];
```

预留信息，供以后使用。

- lut\_data:

保存 lut 表信息，size 表示每个 rgb lut 表中有多少个数据，最大为 1024。

```
struct lut_data{  
    uint16_t size;  
    uint16_t lred[1024];  
    uint16_t lgreen[1024];  
    uint16_t lblue[1024];  
};
```

### 3. Backup

由于有恢复出厂设置的需求，baseparameter 的后半部分用来保存初始数据，起始偏移 512kb。

