

1.三色RGB驱动库使用流程:

1.硬件要求及移植过程, 参考RGB_har.h

2.函数应用实例:

(1) 实现呼吸灯, 心跳灯, 常亮灯

```
1  #include "RGB_app.h"
2
3  /*RGB 输出相关结构体定义*/
4  struct RGB_COLOR_STRUCT *Red, *Green, *Yellow;
5  struct RGB_SHINGING_MODE_STRUCT_BREATH *Breath_red, *Breath_green, *Breath_yellow;
6  struct RGB_SHINGING_MODE_STRUCT_CO *CO_red, *CO_green, *CO_yellow;
7  struct RGB_SHINGING_MODE_STRUCT_HEART *Heart_red, *Heart_green, *Heart_yellow;
8
9  int main( void )
10 {
11
12     Peripheral_Init(); //初始化片内外设, 板载外设
13     Parameter_Init(); //初始化全局变量
14
15     /*定义各颜色对象*/
16     /*红色分量, 绿色分量, 蓝色分量, 三种分量应该在0 - 1之内, 分量之和可以不为1*/
17     Red = RGB_getColorStruct(1, 0, 0);
18     Green = RGB_getColorStruct(0, 1, 0);
19     Yellow = RGB_getColorStruct(0.8, 0.2, 0);
20
21     /*定义呼吸灯模式对象*/
22     /*最亮亮度, 最暗暗度, 呼吸速度*/
23     Breath_red = RGB_getShingingModeStruct_breath(RGB_TIMER_PERIOD - 2000, 0, 100);
24     Breath_green = RGB_getShingingModeStruct_breath(RGB_TIMER_PERIOD - 2000, 0, 100);
25     Breath_yellow = RGB_getShingingModeStruct_breath(RGB_TIMER_PERIOD - 2000, 0, 100);
26
27     /*定义心跳灯模式对象*/
```

```

28  /*最亮亮度，最暗暗度，心跳点，心跳持续时间，心跳周期*/
29  Heart_red = RGB_getShingingModeStruct_heart(RGB_TIMER_PERIOD - 2000, 0,
10, 2, 50);
30  Heart_green = RGB_getShingingModeStruct_heart(RGB_TIMER_PERIOD - 2000,
0, 10, 2, 50);
31  Heart_yellow = RGB_getShingingModeStruct_heart(RGB_TIMER_PERIOD - 2000,
0, 10, 2, 50);
32
33  /*定义常亮灯模式对象*/
34  /*最亮亮度*/
35  CO_red = RGB_getShingingModeStruct_CO(RGB_TIMER_PERIOD - 2000);
36  CO_green = RGB_getShingingModeStruct_CO(RGB_TIMER_PERIOD - 2000);
37  CO_yellow = RGB_getShingingModeStruct_CO(RGB_TIMER_PERIOD - 2000);
38
39  while(1) {
40
41  //执行以下其中一个函数，实现呼吸灯，心跳灯或常亮灯
42
43  RGB_breath_output(*Red, Breath_red, ENABLE); //呼吸灯
44  RGB_heartBeat_output(*Red, Heart_red, ENABLE); //心跳灯
45  RGB_ConstantOn_output(*Red, CO_red, ENABLE); //常亮灯
46
47  /*循环计数*/
48  delay_ms(20), counter++;
49  }
50  }

```

(2) 根据三种颜色分量点亮RGB

```

1  #include "RGB_app.h"
2
3  int main( void )
4  {
5
6  Peripheral_Init(); //初始化片内外设，板载外设
7  Parameter_Init(); //初始化全局变量
8
9  while(1) {
10
11  RGB_mixOutput(1, 0.5, 0.5); //红色分量，绿色分量，蓝色分量，三种分量应该在0
- 1之内，分量之和可以不为1
12
13  }

```

```
14 }
```

(3) 根据预定义的颜色，点亮RGB

```
1 #include "RGB_app.h"
2
3 int main( void )
4 {
5
6     Peripheral_Init(); //初始化片内外设，板载外设
7     Parameter_Init(); //初始化全局变量
8
9     while(1) {
10
11         RGB_mixOutput(RGB_RED, ENABLE); //输出红色
12
13     }
14 }
```

2.库函数详细讲解

```
1 /*根据输入参数，获得一个颜色对象句柄*/
2 // 输入：三种颜色分量，范围应该在0-1之内，三种分量之和可以不为1
3 // 返回值：颜色对象结构体
4 struct RGB_COLOR_STRUCT* RGB_getColorStruct(f32 red_vector, f32 green_vector, f32 blue_vector);
```

```
1 2./*根据输入参数，获得闪烁模式对象句柄*/
2 //输入：亮度（捕获比较模块的最大值），暗度（捕获比较模块的最小值），呼吸速度（每次执行呼吸函数时捕获比较模块的增量）
3 //返回值：闪烁模式对象-呼吸灯模式结构体
4 struct RGB_SHINGING_MODE_STRUCT_BREATH* RGB_getShingingModeStruct_breath(
    BITWIDTH brightness, BITWIDTH darkness, BITWIDTH speed);
5 //输入：亮度（捕获比较模块的最大值），暗度（捕获比较模块的最小值），
6 // 心跳点（捕获比较模块的值等于该值时会发生跳动，第一次在0点跳动，第二次在定义
    的点跳动）
7 // 心跳时间（跳起到落下的捕获比较模块计数值），心跳周期（不是捕获比较模块的数
    值）
8 //返回值：闪烁模式对象-心跳灯模式结构体
9 struct RGB_SHINGING_MODE_STRUCT_HEART* RGB_getShingingModeStruct_heart(
    BITWIDTH brightness, BITWIDTH darkness, \
```

```

10 //输入：亮度（捕获比较模块的最大值）
11 //返回值：闪烁模式对象-常亮灯模式结构体
    BITWIDTH beatPoint, BITWIDTH beatTime, BITWIDTH period);
12 struct RGB_SHINGING_MODE_STRUCT_CO* RGB_getShingingModeStruct_CO(BITWIDTH brightness);

```

```

1 /*定时执行一次，便能输出呼吸灯，心跳灯或常亮灯*/
2 //输入：颜色模式结构体和闪烁模式结构体，使能/失能位
3 void RGB_heartBeat_output(struct RGB_COLOR_STRUCT colorStruct, struct RGB_SHINGING_MODE_STRUCT_HEART* modeStruct, u8 status);
4 void RGB_breath_output(struct RGB_COLOR_STRUCT colorStruct, struct RGB_SHINGING_MODE_STRUCT_BREATH* modeStruct, u8 status);
5 void RGB_ConstantOn_output(struct RGB_COLOR_STRUCT colorStruct, struct RGB_SHINGING_MODE_STRUCT_CO* modeStruct, u8 status);

```

```

1 /*根据输入的颜色，点亮三色RGB*/
2 //输入：颜色枚举对象， 使能/失能位
3 void RGB_quickOutput(RGB_COLOR color, BitStatus status);

```

```

1 //以三种颜色的合成颜色点亮RGB
2 //输入：三种颜色分量，范围应该在0-1之内，三种分量之和可以不为1
3 void RGB_mixOutput(f32 red_vector, f32 green_vector, f32 blue_vector);

```

3.结构体，枚举对象详细讲解

```

1 /*闪烁循环方向枚举定义*/
2 //只适用于呼吸灯
3 typedef enum {
4     RGB_SHINGING_DIR_UP = 0, //吸气，亮度增强中
5     RGB_SHINGING_DIR_DOWN //呼气，亮度减弱中
6
7 } RGB_SHINGING_DIR;

```

```

1 /*闪烁模式枚举定义*/
2 typedef enum {
3     RGB_SHINGING_MODE_BREATH = 0, //呼吸模式

```

```
4     RGB_SHINGING_MODE_HEART_BEAT, //心跳模式
5     RGB_SHINGING_MODE_CO //常亮模式
6 } RGB_SHINGING_MODE;
```

```
1 /*预定义颜色枚举定义*/
2 typedef enum {
3     RGB_RED = 0, //红
4     RGB_BLUE, //蓝
5     RGB_GREEN, //绿
6     RGB_WHITE, //白
7     RGB_PINK, //粉
8     RGB_PURPLE, //紫
9     RGB_YELLOW //黄
10 } RGB_COLOR;
```

```
1 /*颜色对象结构体定义*/
2 struct RGB_COLOR_STRUCT {
3     f32 red_vector; //各颜色分量，实际亮度为 brightness * vector，要保证在
0 - 1之间，无参数检查机制
4     f32 green_vector;
5     f32 blue_vector;
6 };
```

```
1 /*呼吸灯模式对象结构体定义*/
2 struct RGB_SHINGING_MODE_STRUCT_BREATH {
3     BITWIDTH speed; //速度，每次执行函数时的捕获比较寄存器增量
4     BITWIDTH brightness; //最大亮度，捕获比较寄存器最大值
5     BITWIDTH darkness; //最大暗度，捕获比较寄存器最小值
6
7     u8 dir; //当前计数方向，标记方向用，不用初始化赋值
8     BITWIDTH count; //当前时间，计时用，不用初始化赋值
9 };
```

```
1 /*常亮模式对象结构体定义*/
2 struct RGB_SHINGING_MODE_STRUCT_CO {
3     BITWIDTH brightness; //亮度
4 };
```

```
1 /*心跳灯模式对象结构体定义*/
```

```
2 struct RGB_SHINGING_MODE_STRUCT_HEART {
3     BITWIDTH brightness;
4     BITWIDTH darkness;
5
6     BITWIDTH beatPoint; //第一次在0点跳动，第二次在哪跳动
7     BITWIDTH beatTime; //从跳起到落下时间
8
9     BITWIDTH count; //当前时间，计时用，不用初始化赋值
10    BITWIDTH period; //心跳周期
11 };
```