

void OLED\_Init(void)

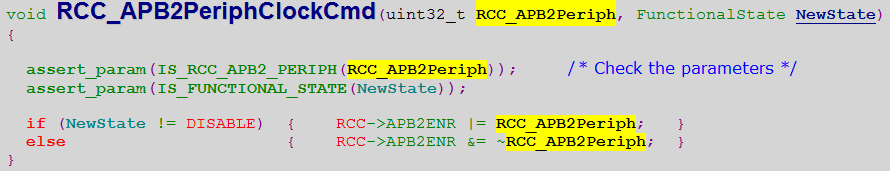
{

GPIO\_InitTypeDef GPIO\_InitStructure;

**RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_AFIO, ENABLE);**

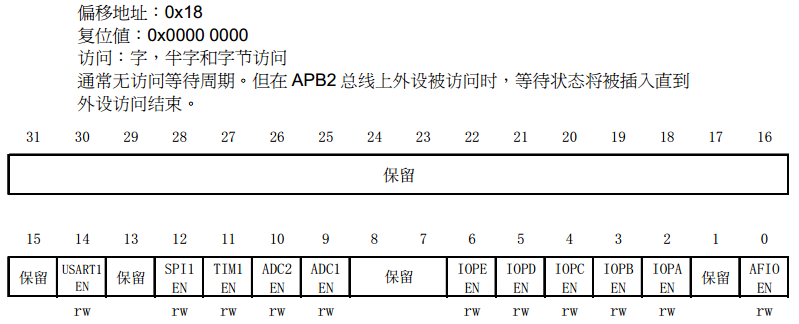
上面RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_AFIO, ENABLE);的操作：

#define RCC\_APB2Periph\_AFIO ((uint32\_t)0x00000001)



就是说，外设时钟使能寄存器(RCC\_APB2ENR)的最低位设置成1，也就是辅助功能IO始终开启。

4.3.7 APB2 外设时钟使能寄存器(RCC\_APB2ENR)





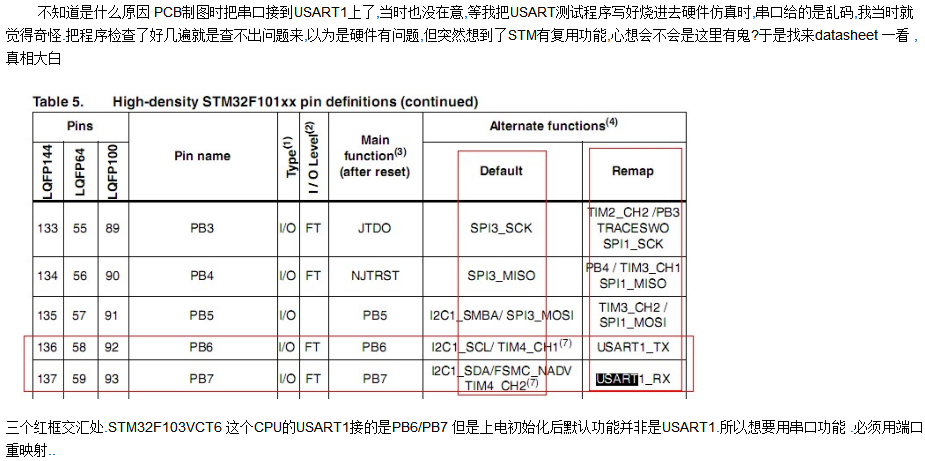
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

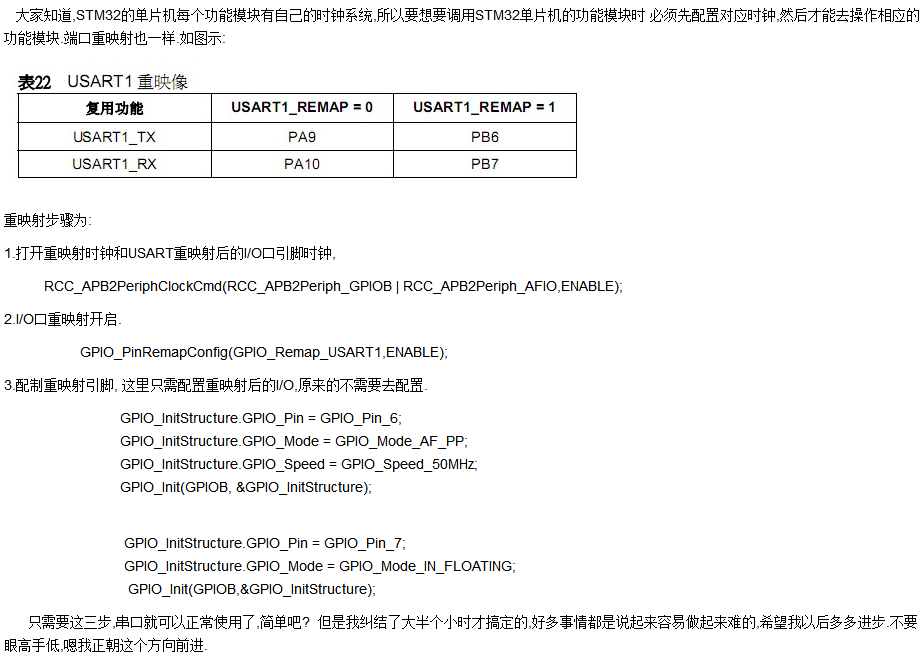
**GPIO\_PinRemapConfig(GPIO\_Remap\_SWJ\_JTAGDisable, ENABLE);**

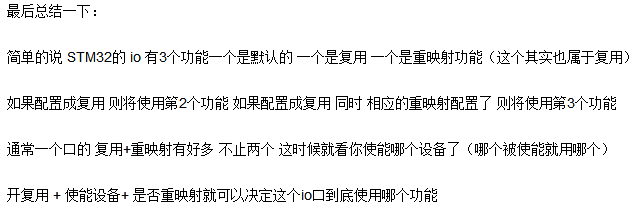
#define GPIO\_Remap\_SWJ\_JTAGDisable ((uint32\_t)0x00300200) /\*!< JTAG-DP Disabled and SW-DP Enabled \*/

简单来说，应该是讲该引脚设置成JTAG-DP disable而SW-DP enable.

这里涉及到“引脚重定义remap”，类似的知识点找到：





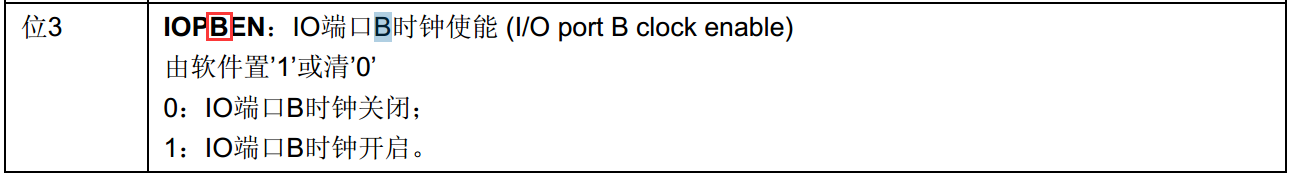


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**RCC\_APB2PeriphClockCmd**(RCC\_APB2Periph\_GPIOB, ENABLE); //使能PD端口时钟

又使用RCC\_APB2PeriphClockCmd()调用。

#define RCC\_APB2Periph\_GPIOB ((uint32\_t)0x00000008) // APB2 bit3置1



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_3|GPIO\_Pin\_4|GPIO\_Pin\_5|GPIO\_Pin\_0|GPIO\_Pin\_1; // 用到了PB\_0/1/3/4/5这几个引脚

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP; //推挽输出

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz; //速度50MHz

GPIO\_Init(GPIOB, &GPIO\_InitStructure); // 初始化GPIOD3,6

#define GPIOB ((GPIO\_TypeDef \*) GPIOB\_BASE) // (uint32\_t)0x40000000 + 0x10000 + 0x0C00 = 0x40010C00

#define GPIOB\_BASE (APB2PERIPH\_BASE + 0x0C00)

#define APB2PERIPH\_BASE (PERIPH\_BASE + 0x10000)

#define PERIPH\_BASE ((uint32\_t)0x40000000) /\*!< Peripheral base address in the alias region \*/

#define GPIO\_Pin\_0 ((uint16\_t)0x0001) /\*!< Pin 0 selected \*/ bit0

#define GPIO\_Pin\_1 ((uint16\_t)0x0002) /\*!< Pin 1 selected \*/ bit1

#define GPIO\_Pin\_2 ((uint16\_t)0x0004) /\*!< Pin 2 selected \*/ bit2

#define GPIO\_Pin\_3 ((uint16\_t)0x0008) /\*!< Pin 3 selected \*/ bit3

#define GPIO\_Pin\_4 ((uint16\_t)0x0010) /\*!< Pin 4 selected \*/ bit4

#define GPIO\_Pin\_5 ((uint16\_t)0x0020) /\*!< Pin 5 selected \*/ bit5

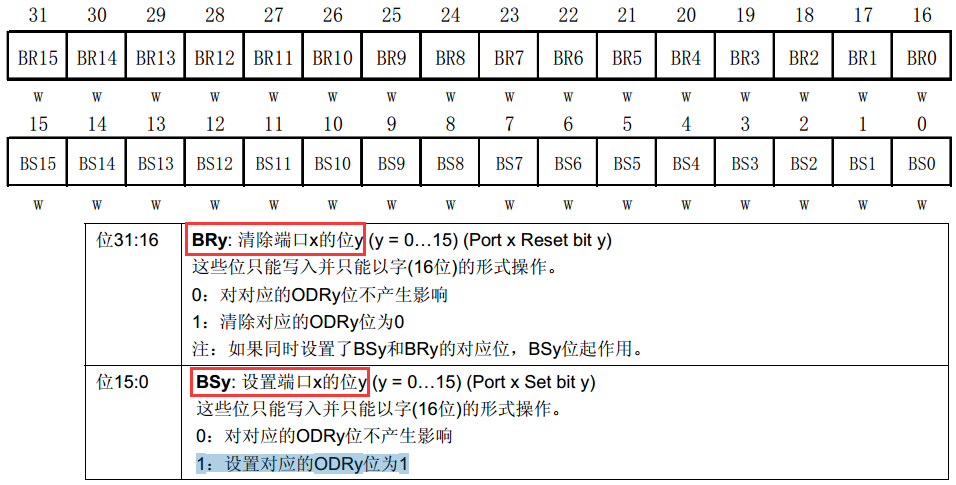
#define GPIO\_Pin\_6 ((uint16\_t)0x0040) /\*!< Pin 6 selected \*/ bit6

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

GPIO\_SetBits(GPIOB,GPIO\_Pin\_3|GPIO\_Pin\_4|GPIO\_Pin\_5|GPIO\_Pin\_0|GPIO\_Pin\_1); // **这个函数设置寄存器干什么？？？？？**

该函数会调用GPIOx->BSRR = GPIO\_Pin; /\* 端口位设置/清除寄存器(GPIOx\_BSRR) (x=A..E) \*/

其中BSRR寄存器是端口位设置/清除寄存器



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

OLED\_RST\_Set(); // #define OLED\_RST\_Set() **GPIO\_SetBits**(GPIOB,GPIO\_Pin\_3) // **这样就拉高了GPIO\_Pin\_3？？？？**

delay\_ms(100);

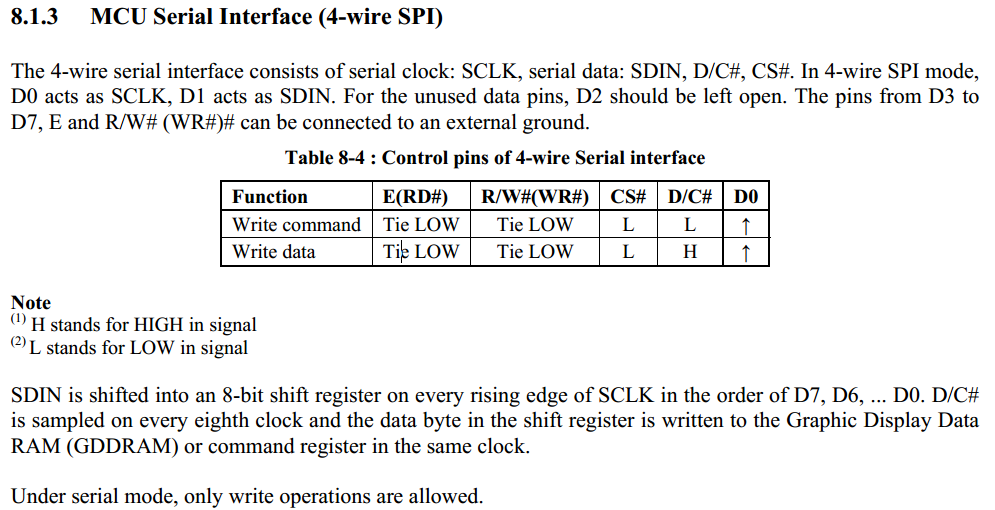
OLED\_RST\_Clr(); // #define OLED\_RST\_Clr() **GPIO\_ResetBits**(GPIOB,GPIO\_Pin\_3) //RES

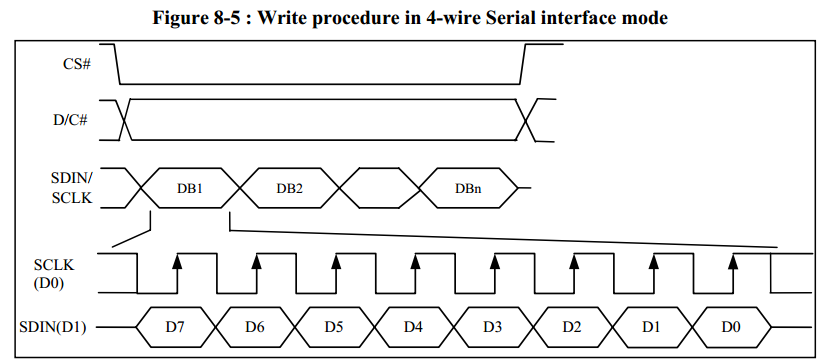
delay\_ms(100);

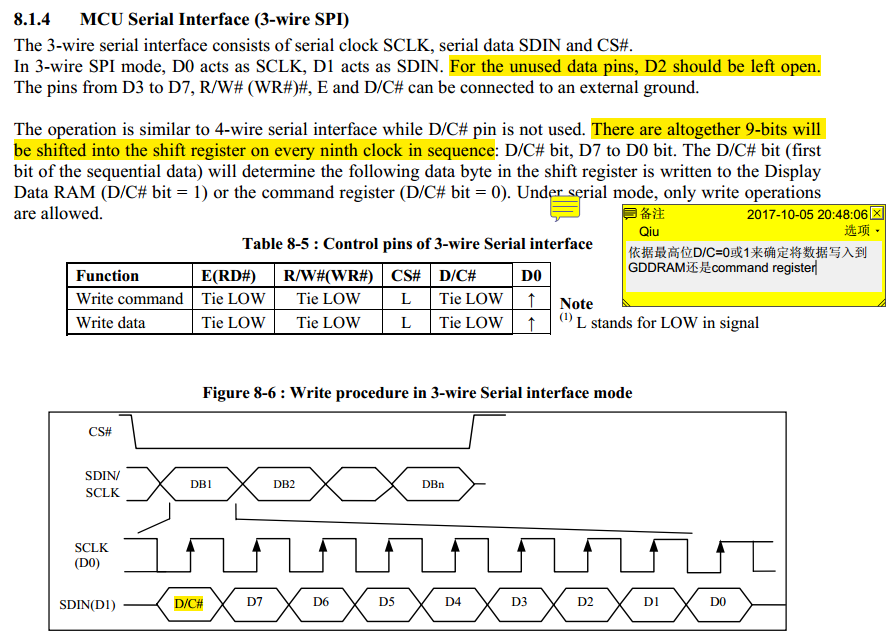
OLED\_RST\_Set();

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

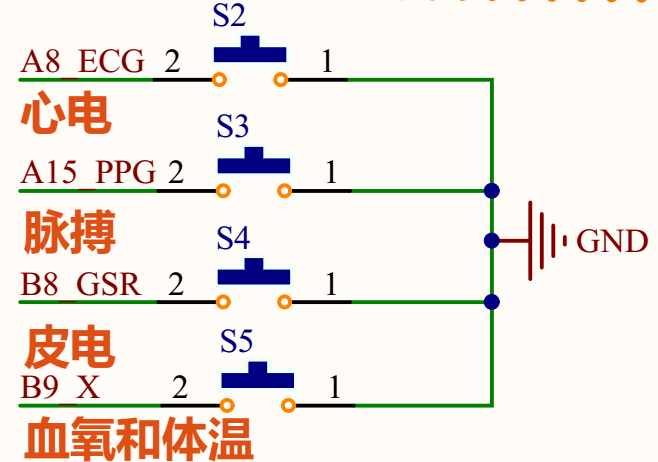
关于OLED的写时序：(分3线SPI和4线SPI)





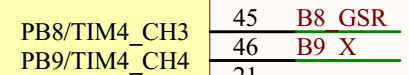


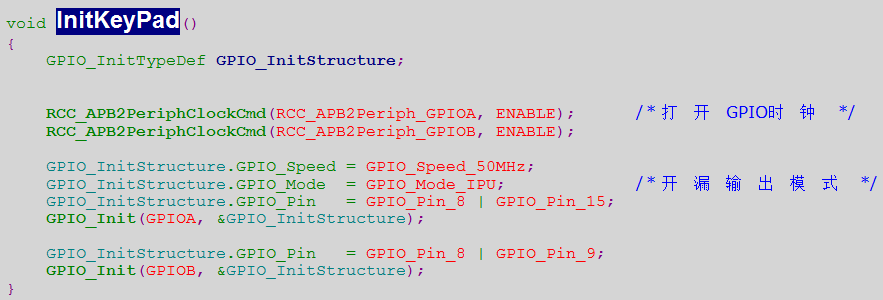
=============================按键keypad=========================================================











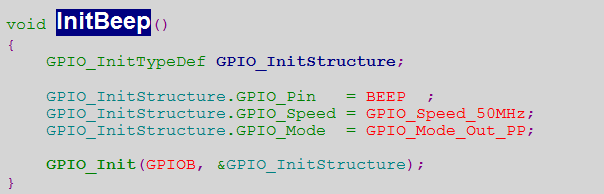
#define RCC\_APB2Periph\_GPIOA ((uint32\_t)0x00000004) // bit2

#define RCC\_APB2Periph\_GPIOB ((uint32\_t)0x00000008) // bit3



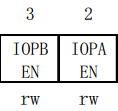
GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IPU; 看注释说GPIO\_Mode\_IPU; /\*上拉输入 \*/

=============================蜂鸣器beep=========================================================

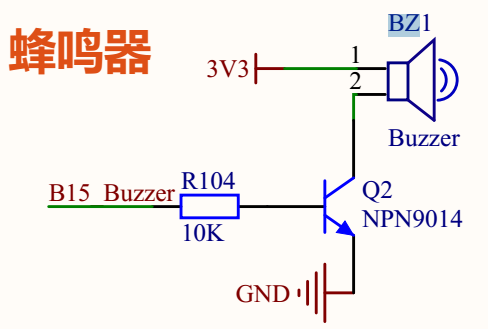


为什么不这么做：RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB, ENABLE); /\* 打开GPIO时钟 \*/

因为上面键盘初始化已经做过。我觉得应该再做一下更好。



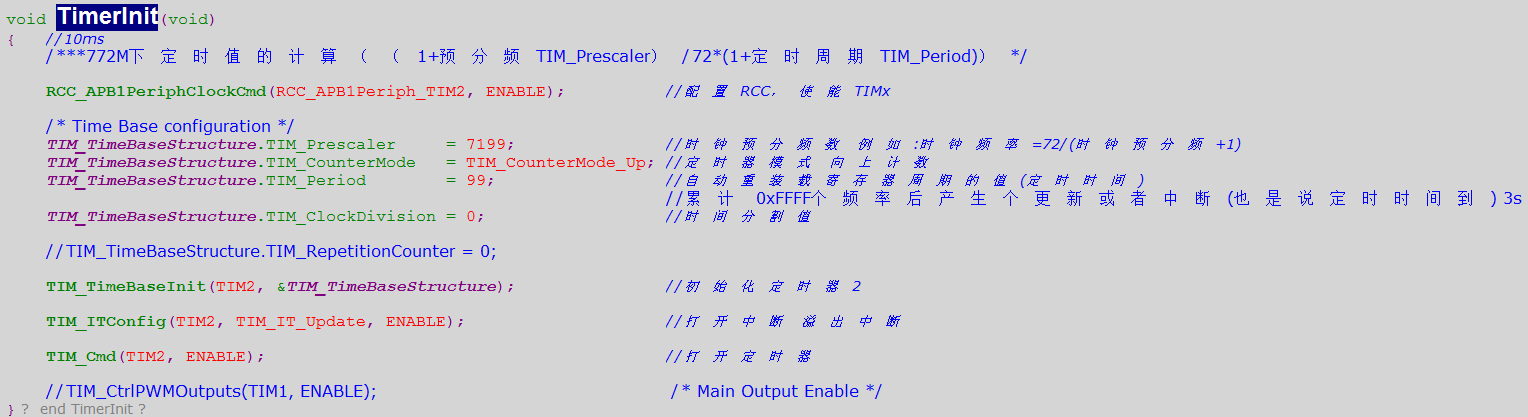
这2位是可读可写的。只是目前不清楚读的函数。



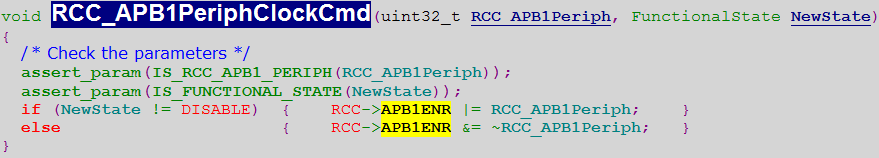


GPIO\_Mode\_Out\_PP是 推挽输出，代码中还有GPIO\_Mode\_Out\_OD是开漏输出

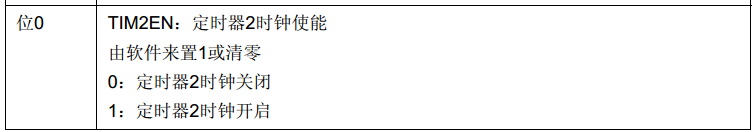
=============================定时器timer=========================================================



#define RCC\_APB1Periph\_TIM2 ((uint32\_t)0x00000001)



**APB1 外设时钟使能寄存器(RCC\_APB1ENR)**



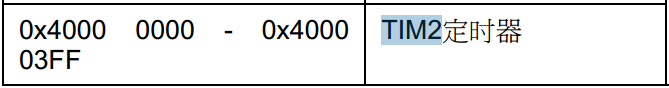
**TIM\_TimeBaseInit(TIM2, &TIM\_TimeBaseStructure); //初始化定时器2**

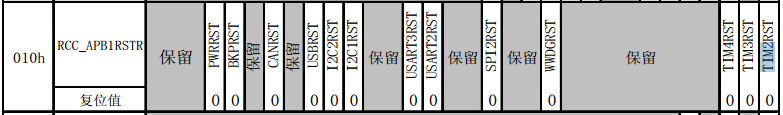
#define TIM2 ((TIM\_TypeDef \*) TIM2\_BASE) // 0x40000000 + 0x0000 = 0x40000000

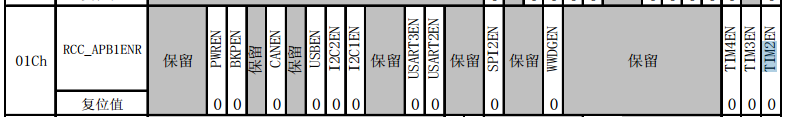
#define TIM2\_BASE (APB1PERIPH\_BASE + 0x0000)

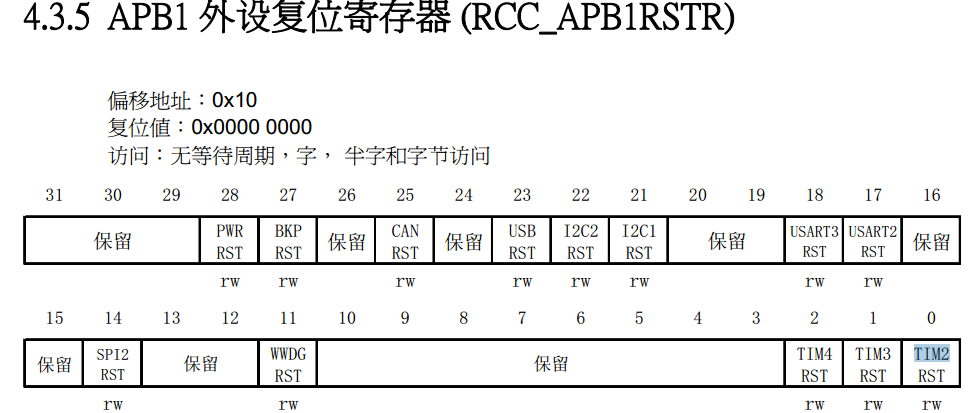
#define APB1PERIPH\_BASE PERIPH\_BASE

#define PERIPH\_BASE ((uint32\_t)0x40000000) /\*!< Peripheral base address in the alias region \*/



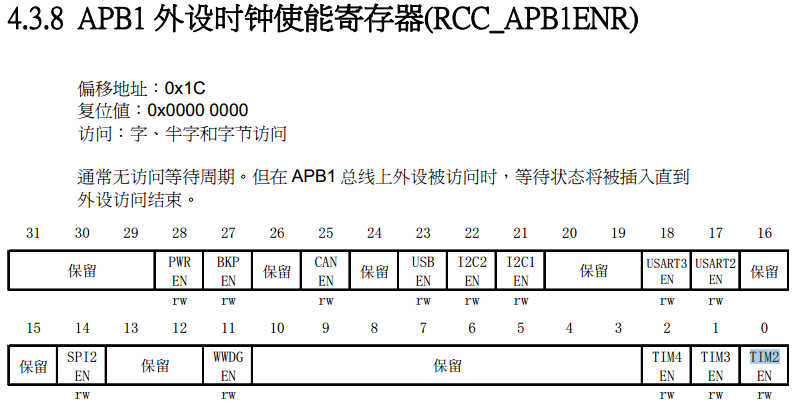


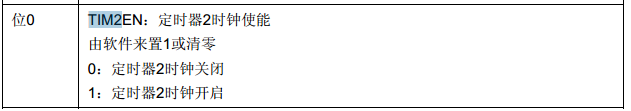






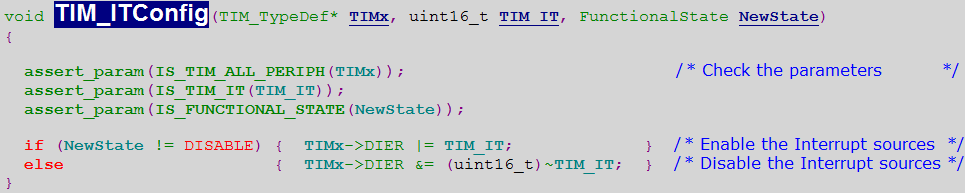
手册上还看到相关寄存器：

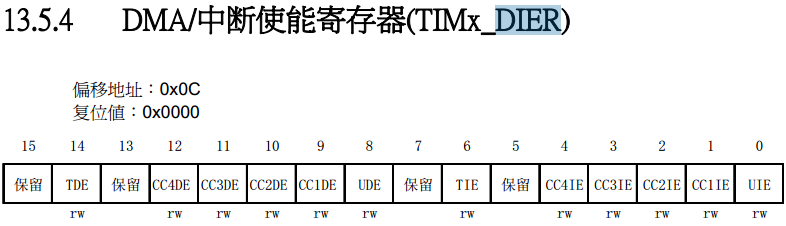


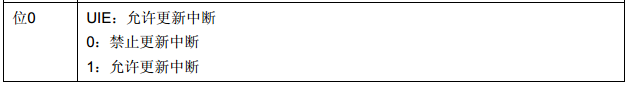


**TIM\_ITConfig(TIM2, TIM\_IT\_Update, ENABLE);**  //打开中断溢出中断

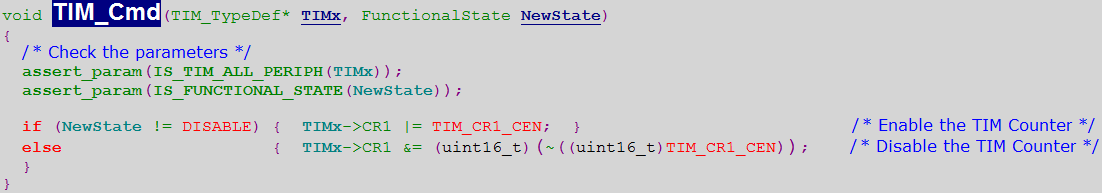
#define TIM\_IT\_Update ((uint16\_t)0x0001)







**TIM\_Cmd(TIM2, ENABLE);**  //打开定时器



#define TIM\_CR1\_CEN ((uint16\_t)0x0001) /\*!< Counter enable \*/

