

第十一章 BEEP 及其应用实例

STM8S207 芯片自带了 BEEP 接口，通过程序控制选择 1、2、4KHz 的输出，只要在 BEEP 接口接上蜂鸣器就可以实现作为报警作用，BEEP 接口简单而实用

11.1 STM8S207 引脚的第二功能

STM8S207 大部分引脚都有附加功能，而第二功能的引脚是在原理图中用方括号括起来的。例如这章实用的 BEEP 引脚就是在 PD 端口的第 4 个引脚。既然是第二功能，那么如何使用这个功能引脚？所以本章除了详细说明 BEEP 编程之外，更为重要的是理解 STM8S207 芯片一个更为特色的编程，选项字编程。当然选项字可以直接使用 ST-Link 在 STVP 中修改，但是不方便在程序中修改，除此之外的另一种方式就是通过在线程序对选项字编程，这里会涉及到 Flash 编程，为了降低难易程度，这里只是说明操作方法以及如何使用。

11.2 BEEP 引脚

BEEP 引脚在 PD4 中，除此之外还是定时 2 的通道 1 TIM2_CH1，TIM2_CH1 是默认功能，当初始化定时器 2 通道 1 时就可以使用这个功能，但是却使用不了 BEEP 功能。这里就要修改选项字了，当再次想把 PD4 改回 TIM2_CH4 功能的时候也要通过选项字才能修改回去。

11.3 选项字介绍

在 STM8S20XX 数据手册中的第 26 页有详细的说明，这里我们简单概述一下

首先是如何实现选项字节编程，有两种方法

1、SWIM 方式，也就是借助 ST-Link 仿真器可以人为修改，好处是方便而不容易出错。缺点是不能程序控制

2、另一种是在线编程修改，也就是在程序中直接修改，我们采取了第二种方法。同时自己参考 ST 公司的 Flash 库，自己封装好了这些函数。这是因为头

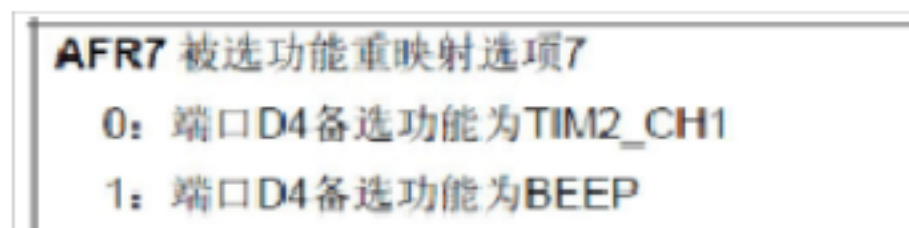
文件用了 IAR 的封装，这和 ST 的库不兼容，而且 IAR 的直接操作寄存器对应自己更为方便操作，不过存在学习周期长而容易出错的问题

下图展示了选项字中所有的地址，这些地址不能直接操作。

表6 选项字节

Addr.	Option name	Option byte no.	Option bits								Factory default setting
			7	6	5	4	3	2	1	0	
4800h	Read-out protection (ROP)	OPT0	ROP[7:0]								00h
4801h	User boot code(UBC)	OPT1	UBC[7:0]								00h
4802h		NOPT1	NUBC[7:0]								FFh
4803h	Alternate function remapping (AFR)	OPT2	AFR7	AFR6	AFR5	AFR4	AFR3	AFR2	AFR1	AFR0	00h
4804h		NOPT2	NAFR7	NAFR6	NAFR5	NAFR4	NAFR3	NAFR2	NAFR1	NAFR0	FFh
4805h	Watchdog option	OPT3	Reserved				LSI_EN	IWDG_HW	WWDG_HW	WWDG_HALT	00h
4806h		NOPT3	Reserved				NLSI_EN	NWWDG_HW	NWWDG_HALT		FFh
4807h	Clock option	OPT4	Reserved				EXT_CLK	CKAWU_SEL	PRS_C1	PRS_C0	00h
4808h		NOPT4	Reserved				NEXT_CLK	NCKAWU_SEL	NPR_SC1	NPR_SC0	FFh
4809h	HSE clock startup	OPT5	HSECNT[7:0]								00h
480Ah		NOPT5	NHSECNT[7:0]								FFh
480Bh	Reserved	OPT6	Reserved								00h
480Ch		NOPT6	Reserved								FFh
480Dh	Flash wait states	OPT7	Reserved							Wait state	00h
480Eh		NOPT7	Reserved							Nwait state	FFh
487Eh	Bootloader	OPTBL	BL[7:0]								00h
487Fh		NOPTBL	NBL[7:0]								FFh

上表是全部的选项字节，除了 0x4800 地址之外，其它地址都有一个互补备份数据。而我们这节使用的功能在 OPT2 的 AFR7 中，见下图：



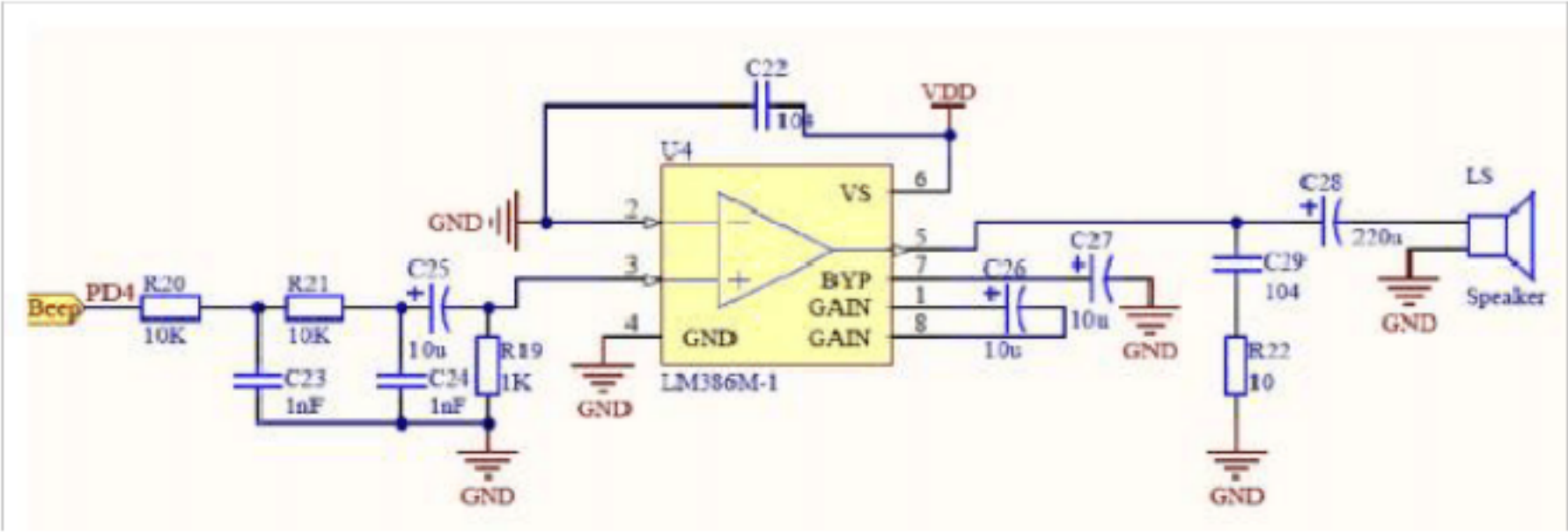
所以我们的目的就是修改 OPT2 的第 7 位为 1，而修改 NOPT2 的第七位为 0。不过我们需要用到 Flash 操作，我已经为大家封装好这些函数，大家如果有兴趣研究，可以直接打开对应的 PDF 文件查阅源码，已经附上大量注释

因为这一节只是用到 BEEP, 所以我们以 BEEP 的选项字节编程为例程，其它的相信动了这个例程后都可以自己实现选项字节编程。不过需要注意的是，当需要这个引脚原来的功能是，请记得重新修改对应的选项字节。

11.4 BEEP编程

实验的硬件：

我们的实验板采用了无源蜂鸣器，关于有源与无源的区别是，无源的只对频率有所反应而不是直接接一个高电平就会响。除此之外，我们增加了外围电路的支持，如下：

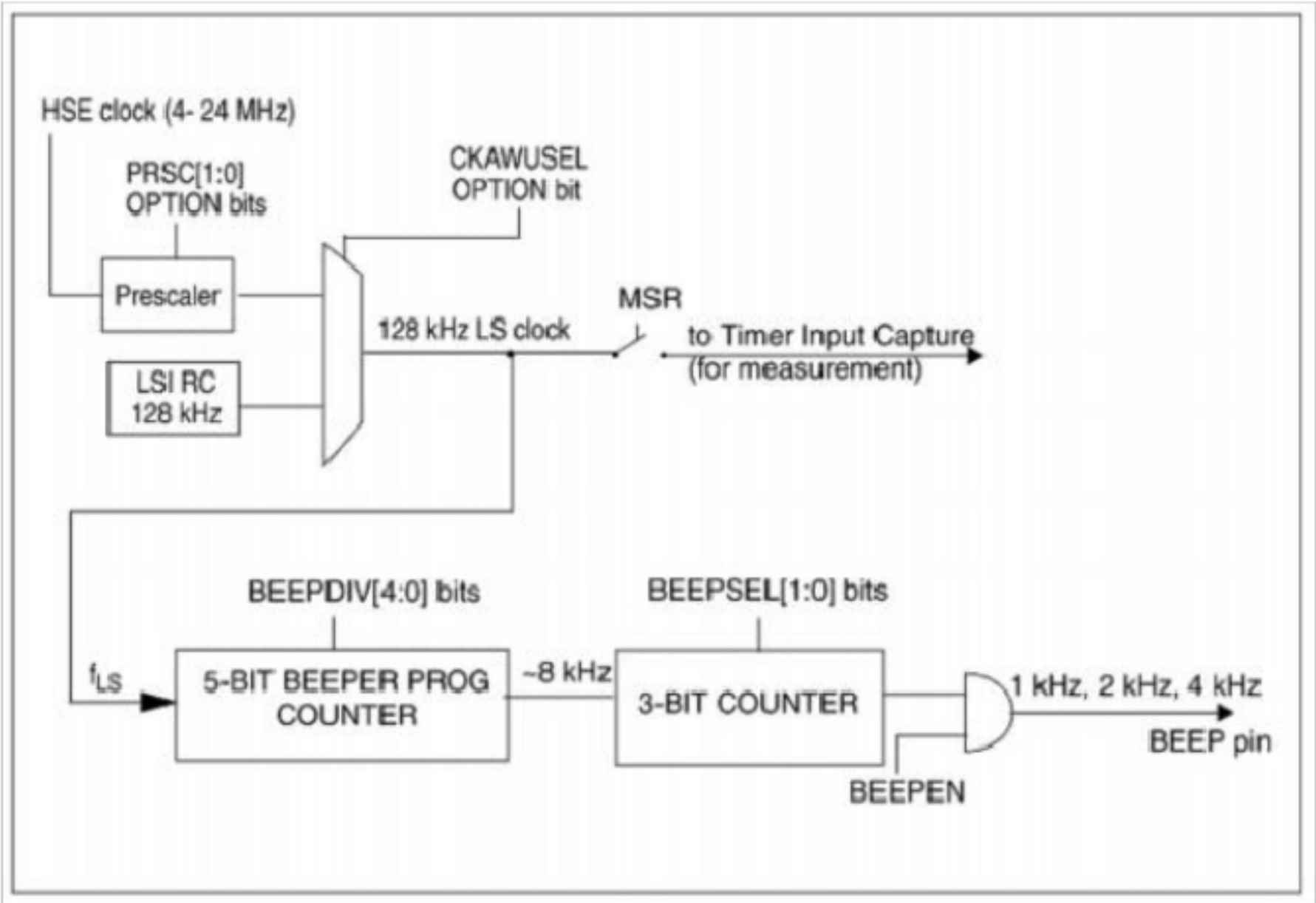


由上图可以知道我们添加了一个 LM386 的运放，并作了相应的滤波效果。可以实现更为逼真的效果。同时 PD4 引脚是连在 TIM2_CH1 的，这个引脚可以实现 PWM 的输出。所以在以后的章节会实现这个电路的第二种功能，可以播放简单的音符

软件介绍

BEEP 的功能介绍

当 LS 时钟工作在 128kHz 时可产生频率为 1kHz，2 kHz 或者是 4 kHz 的蜂鸣信号。



BEEP 功能只有一个寄存器 BEEP_CSR

7	6	5	4	3	2	1	0
BEEPSEL[1:0]		BEEPEN	BEEP DIV[4:0]				
rw		rw	rw				

BEEPSEL是控制 BEEP 的输出频率， 0 = 1K 1 = 2K 2 = 4K

BEEPEN是 BEEP 的使能开关

BEEP DIV 是 f_LS 分频而得到 8K 的频率，如我们的实验是 128/8 = 16

BEEPDIV[4:0] 蜂鸣器预分频器

此位由软件置位和清零。设置蜂鸣器分频因数BEEPDIV。

00h: BEEP_{DIV} = 2

01h: BEEP_{DIV} = 3

...

0Eh: BEEP_{DIV} = 16

0Fh: BEEP_{DIV} = 17

....

1Eh: BEEP_{DIV} = 32

注意：此寄存器不能设置成其初始复位值(0x1F)

由上图可以知道我们设置为 BEEP_{DIV} = 0x0E

蜂鸣器操作

为了使用蜂鸣功能，按顺序执行如下的步骤：

1. 根据 13.2.2 中描述的方法确定 BEEP_{DIV}[4:0]的值来校准 LS 时钟的频率；
2. 通过写 BEEP_CSR的 BEEPSEL[1:0]位来选择 1 kHz, 2 kHz 或 4 kHz 的输出频率；
3. 置位 BEEP_CSR的 BEEPEN位来使能 LS 的时钟源；

下面进行软件代码编写，因为这里只关于 BEEP, Flash 编程的就只知道是为了修改对应选项字节就可以了

```
#include "iostm8s207rb.h"
```

```
#include "flash.h"
```

```
// 声明函数
```

```
void delay_ms(int value);
```

```
void FLASH_INIT(void);
```

```
int main( void )
```

```
{
```

```
    CLK_CKDIVR = 0; //修改分频系数， 16M
```

```
    FLASH_INIT();
```

```
    if(FLASH_ReadOptionByte(0x4803) != 0x807F)
```

```
    {
```

```
        FLASH_ProgramOptionByte(0x4803,0x80); // 将 PD4 修改为 beep
```

```
        WWDG_CR = 0x80; //复位 stm8
```

```
    }
```

```
    BEEP_CSR = 0x0E; //校验时钟 128k/16 = 8k
```

```
    BEEP_CSR_BEEPEN = 1; //使能蜂鸣器
```

```
    BEEP_CSR_BEEPSEL = 1; //2K
```

```
    while(1)
```

```
    {
```

```
        delay_ms(100);
```

```
        BEEP_CSR_BEEPEN = 1;
```

```
        delay_ms(100);
```

```
        BEEP_CSR_BEEPEN = 0;
```

```

    }
}
/*****
* FLASH初始化函数
* 复位 FLASH的寄存器
* FLASH_DeInit();
* 解锁对 Data Flash Memory 进行操作
* FLASH_Unlock(FLASH_MEMTYPE_DATA);
* 标准编程时间
* FLASH_SetProgrammingTime(FLASH_PROGRAMTIME_STANDARD)
*/
void FLASH_INIT(void)
{
    FLASH_DeInit();
    FLASH_Unlock(FLASH_MEMTYPE_DATA);
    FLASH_SetProgrammingTime(FLASH_PROGRAMTIME_STANDARD);
}
/*
*****

```

简单延时程序

```

*****

*/
void delay_ms(int value)
{
    int i,j;
    if(value < 1)
        value = 1;
    for(i=0;i!=value;++i)
        for(j=0;j!=5000;++j);
}

```

编译下载后就可以听到刺耳的声音了，当然如果不想
面的程序中修改为：

```

if(FLASH_ReadOptionByte(0x4803) != 0x00FF)
FLASH_ProgramOptionByte(0x4803,0x00);
    复位后就不是 BEEP 功能了

```

PD4 一直是 BEEP 功能，就在上