

stm8 特有蜂鸣器外设

实验目的：

- 1, 学会使用最基本的延时函数，空指令延时
- 2, 学习修改 stm8 的系统时钟源
- 3, 学习使用 stm8 特有的 beep 输出功能

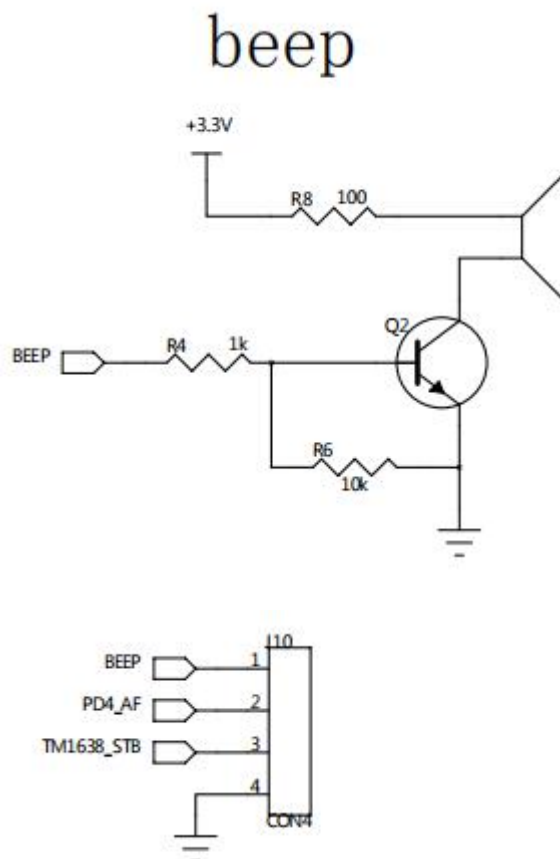
实现功能：

- 1, 用 gpio 实现驱动蜂鸣器鸣叫
- 2, 用 stm8 片上 beep 功能驱动蜂鸣器鸣叫

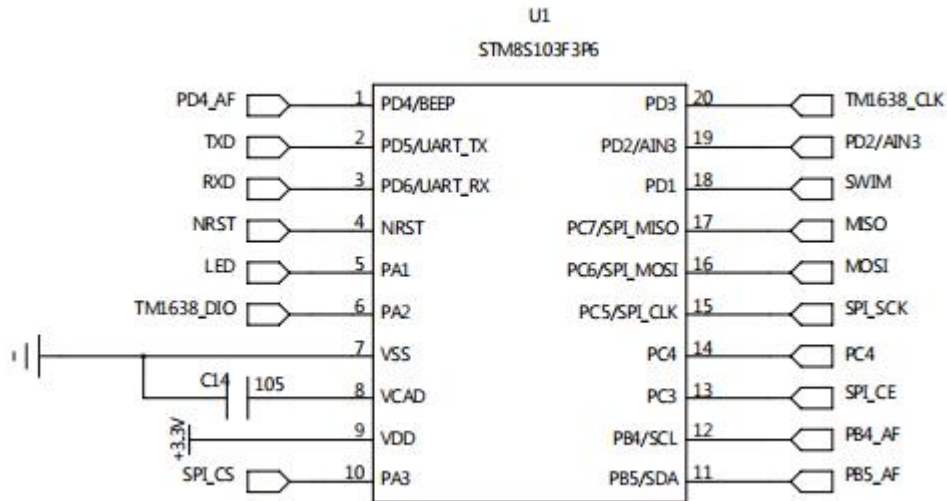
下面进入正题：

- 1, 如果是有源蜂鸣器，那么蜂鸣器通电就鸣叫，断电就停止；
如果是无源蜂鸣器，那么蜂鸣器需要周期性通断电才能发声。
笔者的开发板用的是无源蜂鸣器。

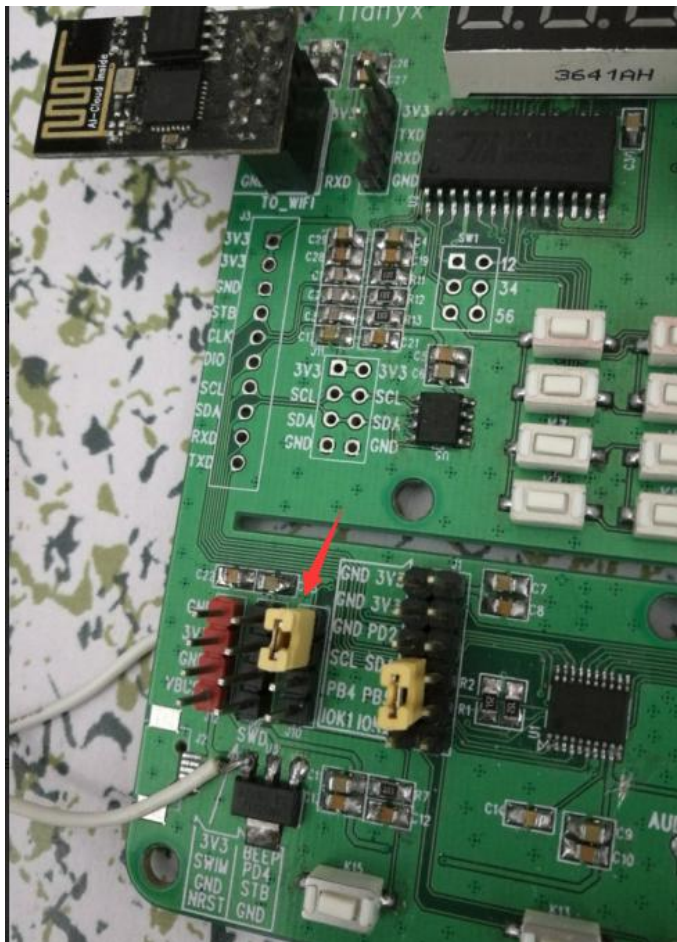
查看原理图，mcu 的 PD4 经过排针，跳线到蜂鸣器或者 tm1638，这里通过短路帽跳线控制蜂鸣器，PD4 输出低电平蜂鸣器断电，输出高电平蜂鸣器通电。



蜂鸣器与tm1638共用PD4，注意跳线



在硬件上使用了一个短路帽，将 PD4 与 BEEP 的插针连接到一起，如下图：



1，同 led 中实验，只需将 PA1 改成 PD4 即可，代码如下：

```

void main(void)
{
    volatile int i;

    i = 0;
    i = 5;
    i = -1;

    // 初始化 PD4 为推挽输出 // 库文件中提供的函数
    GPIO_DeInit(GPIOD);
    GPIO_Init(GPIOD, GPIO_PIN_4, GPIO_MODE_OUT_PP_HIGH_SLOW);
    // 控制 PD4 输出低电平 // 库文件中提供的函数
    GPIO_WriteLow(GPIOD, GPIO_PIN_4);

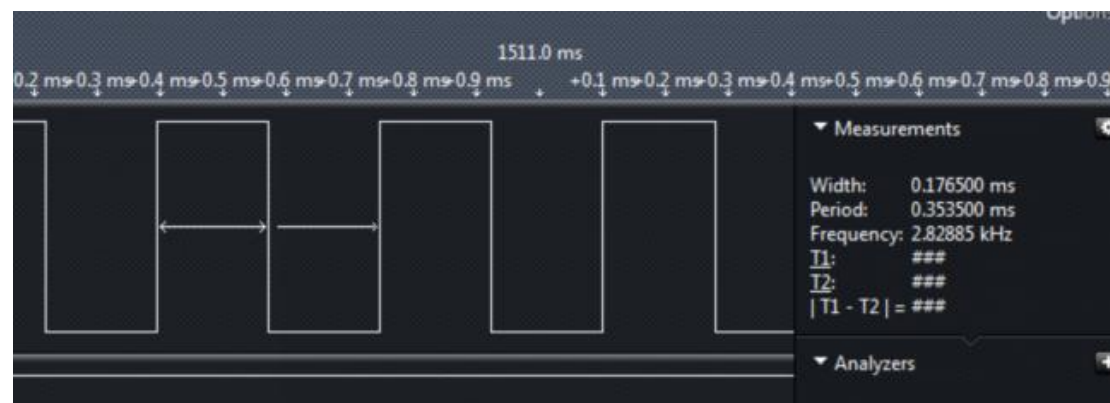
    // 延时, 等待系统稳定
    delay(100);

    /* Infinite loop */
    while (1)
    {
        #if 0
            //delay(20000);
            GPIO_WriteHigh(GPIOA, GPIO_PIN_1);    // PA1输出高, 灯灭
            //delay(20000);
            GPIO_WriteLow(GPIOA, GPIO_PIN_1);      // PA1输出低, 灯亮
        #else
            delay(20);
            GPIO_WriteReverse(GPIOD, GPIO_PIN_4); // PD4 输出电平翻转
        #endif
    }
} « end main »

```

编译调试即可听到蜂鸣器鸣叫。

使用逻辑分析仪查看 PD4 脚信号如下图, 频率为 2.8kHz



2, 空指令延时函数

```

// 空指令延时函数
void delay(u16 d)
{
    u16 i;

    for(i = 0; i < d; i++)
        ; // 空指令
}

```

这个 delay() 函数啥也不做，就执行空指令，消耗时间。

这样的话，cpu 在被消耗的时间内实际上是在做无用功，那么 cpu 的有效使用率就低了，所以我们实际开发中要尽量少用这类延时函数。

新手会问：有些操作间必须要有一定的时间间隔，不延时怎么实现时间间隔？

答案是可以实现的。我这里说少用这类延时函数，是因为还有其他的实现延时的方法，在实现延时的同时，又可以让 cpu 去做其他的有效工作，而且延时时间更精确，这里先卖个关子，下一节再讲怎么实现。

4， 根据 stm8s103 手册，芯片在上电启动时，缺省使用 HSI 的 8 分频作为系统时钟，其中 HSI 是高速内部时钟 16MHz，所以缺省系统时钟是 2MHz。

这里教大家怎么将系统时钟改成 HSI，即 16MHz。

4.1 代码如下：

```
void clk_config(void)
{
    // HSI 时钟 16MHz
    CLK_HSPrescalerConfig(CLK_PRESCALER_HSIDIV1); // HSI时钟预分频，分频系数1
    CLK_SYSCLKConfig(CLK_PRESCALER_HSIDIV1); // 系统时钟配置，HSI，分频系数1
    CLK_HSICmd(ENABLE); // 使能HSI
    while(RESET == CLK_GetFlagStatus(CLK_FLAG_HSIRDY)); // 等待HSI ready
}

/**
 * author : tianyx
 * email : zzztyx55@sina.com
 * qq : 609421258
 * github : https://github.com/zzztyx55
 */
void main(void)
{
    // 配置系统时钟为HSI
    clk_config();

    // 初始化 PD4 为推挽输出 // 库文件中提供的函数
    GPIO_DeInit(GPIOD);
    GPIO_Init(GPIOD, GPIO_PIN_4, GPIO_MODE_OUT_PP_HIGH_SLOW);
    // 控制 PD4 输出低电平 // 库文件中提供的函数
    GPIO_WriteLow(GPIOD, GPIO_PIN_4);

    // 延时，等待系统稳定
    delay(100);

    /* Infinite loop */
    while (1)
    {
        delay(20);
        GPIO_WriteReverse(GPIOD, GPIO_PIN_4); // PD4 输出电平翻转
    }
} // « end main »
```

4.2 由于函数 clk_config() 中调用了库函数，需要将库文件 stm8s_clk.c 和 stm8s_clk.h 添加到工程，添加方法前面建立工程里面已经讲过了。

4.3 编译调试即可听到蜂鸣器发声，

使用逻辑分析仪查看 PD4 脚信号如下图，频率为 22.7kHz

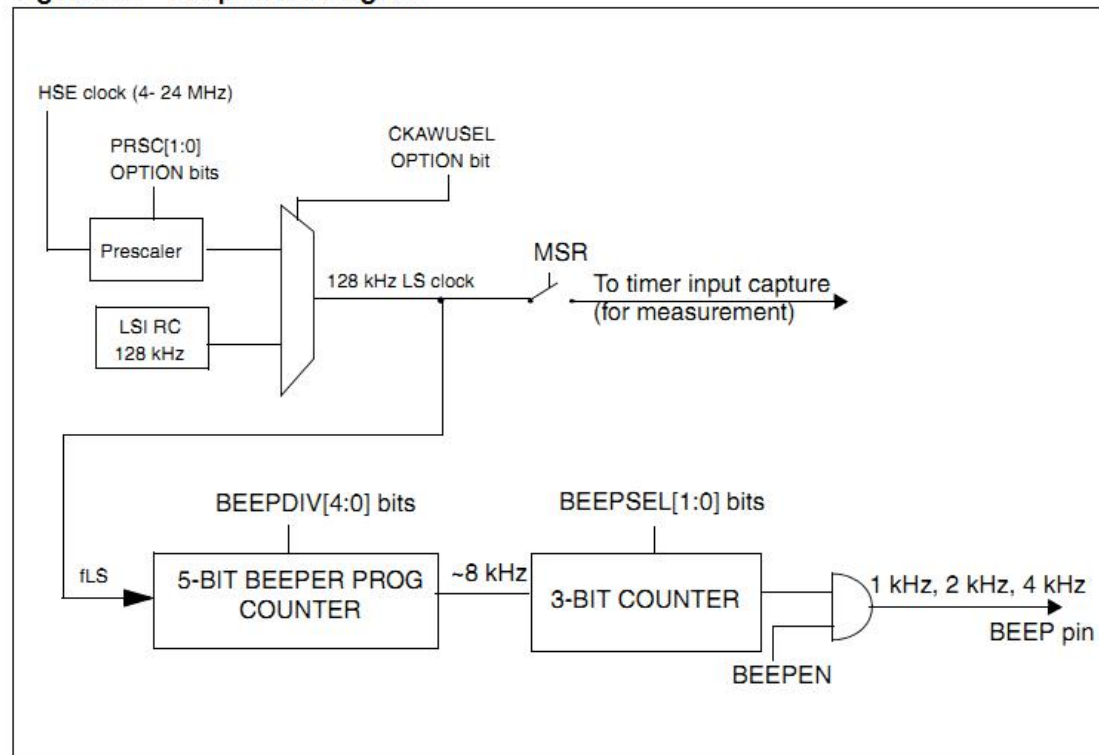
系统时钟由 2MHz 变成 16MHz，快了 8 倍，这里 PD4 的输出信号频率由 2.8KHz 变成 22.7KHz。



5, stm8s103f3p6 芯片集成了 beep 功能, 主要是通过引脚 PD4 对外输出 1kHz / 2kHz / 4kHz 的频率, 下图摘自手册, 欲知详情请看手册的 beep 章节。

由图中可知, 根据选项字的位 CKAWUSEL OPTION bit 可以选择使用外部时钟 HSE 还是内部时钟 LSI 作为 Beep 的输入时钟 fLS, 然后有 5bits (BEEPDIV[4:0]) 计数器分频, 再然后有 3bits (BEEPSEL[1:0]) 计数器分频, 最终输出 1kHz / 2kHz / 4kHz 的 Beep 驱动信号。

Figure 23. Beep block diagram



选项字的修改可以通过 swim 或者在程序中修改,

6 代码如下:

```
/**
 * author : tianyx
 * email : zzztyx55@sina.com
 * qq : 609421258
 * github : https://github.com/zzztyx55
 */
/* Private defines -----*/

/* Private function prototypes -----*/
void delay(u32 dly);
void Beep_inputClkSelect(void);

/* Private functions -----*/
void delay(u32 dly)
{
    u32 i;

    for(i = 0; i < dly; i++)
    {
        ; //空操作
    }
}

// 选择LSI RC 128KHz 为beep输入时钟
void Beep_inputClkSelect(void)
{
    uint16_t value;

    value = FLASH_ReadOptionByte(0x4807); // 读出选项字OPT4地址0x4807 存储的数据
    if(value&0x0400) // 判断CKAWUSEL 位是否为1,若是1,将该位清零
        FLASH_ProgramOptionByte(0x4807,(uint8_t)((value&0xFBFF)>>8));
}

void clk_config(void)
{
    // HSI 时钟 16MHz
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // HSI时钟预分频,分频系数1
    CLK_SYSCLKConfig(CLK_PRESCALER_HSIDIV1); // 系统时钟配置, HSI, 分频系数1
    CLK_HSICmd(ENABLE); // 使能HSI
    while(RESET == CLK_GetFlagStatus(CLK_FLAG_HSIRDY)); // 等待HSI ready
}

void LSI_enable(void)
{
    CLK_LSICmd(ENABLE); // 使能LSI
    while(RESET == CLK_GetFlagStatus(CLK_FLAG_LSIRDY)); // 等待LSI ready
}

void init_beep(void)
{
    BEEP_DeInit();
    //Set_Beep_OptionByte(); // select LSI 128k as LS clock
    LSI_enable();
    BEEP_LSI-CalibrationConfig(128000);
    BEEP_Init(BEEP_FREQUENCY_1KHZ);
    // BEEP_Cmd(ENABLE);
}

void change_beep_freq(u8 freq_group)
{
    //
```

```

void change_beep_freq(u8 freq_group)
{
    //
    BEEP_Cmd(DISABLE);

    switch(freq_group)
    {
        case 0: BEEP_Init(BEEP_FREQUENCY_1KHZ); break;
        case 1: BEEP_Init(BEEP_FREQUENCY_2KHZ); break;
        case 2: BEEP_Init(BEEP_FREQUENCY_4KHZ); break;
        default: BEEP_Init(BEEP_FREQUENCY_1KHZ); break;
    }

    //BEEP_Cmd(ENABLE);
    //
}

/**
 * author : tianyx
 * email : zzztyx55@sina.com
 * qq : 609421258
 * github : https://github.com/zzztyx55
 */
void main(void)
{
    int dly = 1500;

    // 关中断
    disableInterrupts();
    // 系统时钟配置
    clk_config();

    GPIO_DeInit(GPIOD);
    // 初始化Beep
    init_beep();
    // 开中断
    enableInterrupts();

    /* Infinite loop */
    while (1)
    {
        // 使用1kHz的频率驱动蜂鸣器
        BEEP_Cmd(DISABLE);
        change_beep_freq(0); //beep 的输出频率1kHz
        BEEP_Cmd(ENABLE);
        delay(dly); // 延时

        // 关闭蜂鸣器
        BEEP_Cmd(DISABLE);
        delay(dly); // 延时

        // 使用2kHz的频率驱动蜂鸣器
        BEEP_Cmd(DISABLE);
        change_beep_freq(1); //beep 的输出频率2kHz
        BEEP_Cmd(ENABLE);
        delay(dly); // 延时

        // 关闭蜂鸣器
        BEEP_Cmd(DISABLE);
        delay(dly); // 延时

        // 使用4kHz的频率驱动蜂鸣器
        BEEP_Cmd(DISABLE);
        change_beep_freq(2); //beep 的输出频率4kHz
        BEEP_Cmd(ENABLE);
        delay(dly); // 延时

        // 关闭蜂鸣器
        BEEP_Cmd(DISABLE);
        delay(dly); // 延时
    } « end while 1 »
} « end main »

#ifdef USE_FULL_ASSERT

```

email : zzztyx55@sina.com qq : 609421258 github : <https://github.com/zzztyx55>

6.2 由于 Beep 功能用到了库函数，所以需要将相应的函数库文件 `stm8s_beep.c`，`stm8s_flash.c` 添加到工程中，以及对应的 `.h` 头文件，后续提到库文件 `.c` 文件，不再说明需要相应的 `.h` 头文件。

6.3 编译调试即可听到蜂鸣器有 3 种不同的鸣叫声音 循环交替。

使用逻辑分析仪查看 PD4 脚信号如下图，分别为 1KHz，2KHz，4KHz。

