

实验二：定时器及中断实验

一、 实验要求

- (1) 了解 STM32-F1 系列处理器定时器及定时中断的工作原理及编程方法。
- (2) 编写定时中断服务程序，完成周期性工作，并为其他模块提供时间控制。

二、 实验内容

设定定时器周期，设计定时中断服务程序。

三、 目标要求

不依靠软件延时 Delay(unsigned tDly)，在主程序实现 1Hz 及 10Hz 周期性简单处理任务（可通过计数变量如 Cntx 观察）。

四、 原理简述

配置定时器中断：

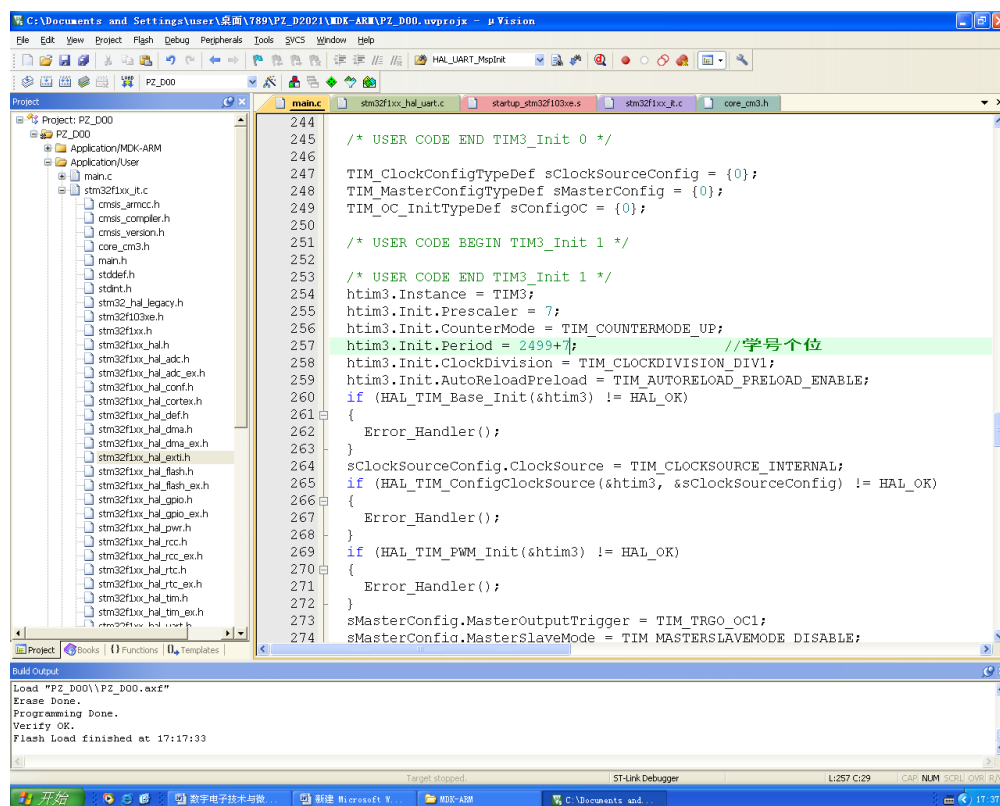


图 1：定时器基本配置

- (1) 定时器时钟。TIM3 时钟接在 APB1 上面,使用 RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE) 打开 TIM3 时钟。
- (2) 清除中断挂起位。在固件库中使用: void TIM_ClearITPendingBit(TIM_TypeDef*TIMx, u16 TIM_IT) 来清除中断挂起位, 中断程序在开始前可能会被置位, 因此需要消除中断挂起位。
- (3) 初始化定时器基本配置。主要使用固件库中的 TIM_TimeBaseInit() 函数进行操作。
- (4) 使能定时器 TIMx。直接使用 TIM_Cmd() 函数就可以。
- (5) 使能 TIMx 中断。在库函数里面定时器中断使能是通过 TIM_ITConfig 函数来实现。
void TIM_ITConfig(TIM_TypeDef* TIMx, uint16_t TIM_IT, FunctionalState NewState);
该函数的形参: 第一个参数是选择定时器号, 取值为 TIM1-TIM17; 第二个参数是用来指明使能的定时器中断的类型, 定时器中断的; 类型有很多种, 包括更新中断 TIM_IT_Update, 触发中断 TIM_IT_Trigger, 以及输入捕获中断等等; 第三个参数就是失能还是使能。
- (6) 配置中断优先级。函数的原型为 void NVIC_Init(NVIC_InitTypeDef* NVIC_InitStruct)
- (7) 编写中断服务程序。先清除中断挂起位, 接着再编写中断处理内容即可。

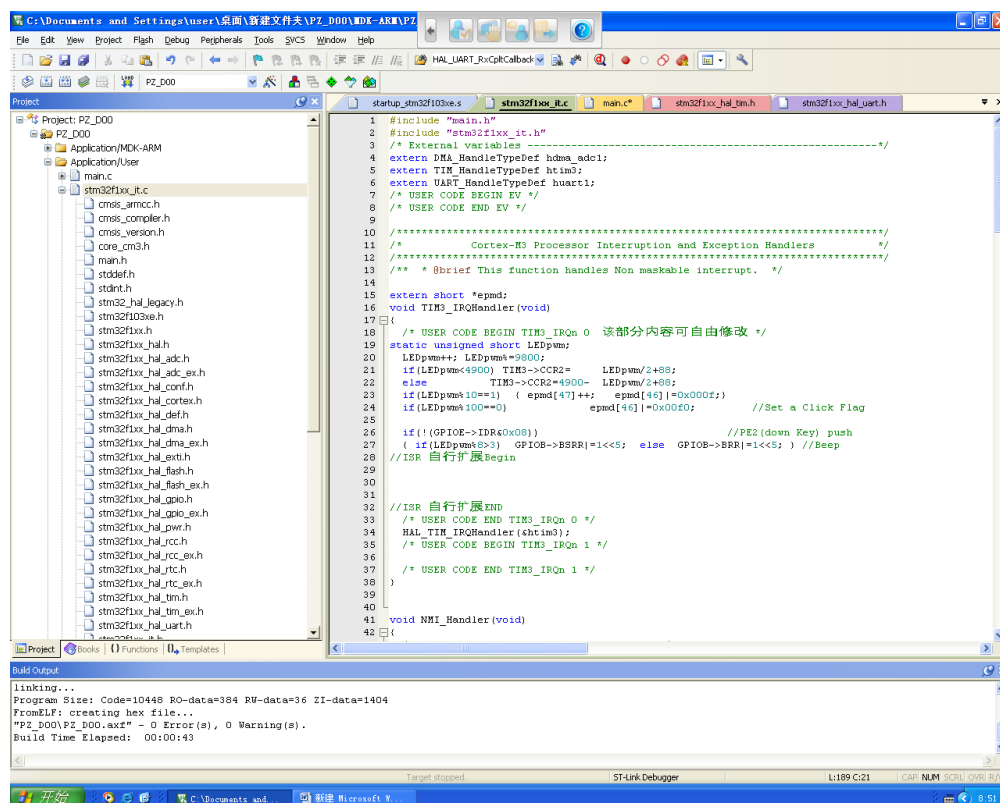


图 2: 中断服务函数界面

五、 实验程序和现象

按下 Down 按键之后，蜂鸣器按照一定频率开始响声。

```
void TIM3_IRQHandler(void)
{
    /* USER CODE BEGIN TIM3_IRQn 0 */
    static unsigned short LEDpwm;
    LEDpwm++; LEDpwm%=9800;
    if(LEDpwm<4900) TIM3->CCR2= LEDpwm/2+88;
    else TIM3->CCR2=4900- LEDpwm/2+88;
    if(LEDpwm%10==1) { epmd[50]++; epmd[46] |=0x000f;}
    if(LEDpwm%100==0) {
        epmd[49]++;
        epmd[46] |=0x00f0; } //Set a Click Flag
    if(!(GPIOE->IDR&0x08))
        //PE2(down Key) push
        { if(LEDpwm%8>3) GPIOB->BSRR|=1<<5; else GPIOB->BRR|=1<<5; }
        //Beep
    /* USER CODE END TIM3_IRQn 0 */
    HAL_TIM_IRQHandler(&htim3);
    /* USER CODE BEGIN TIM3_IRQn 1 */
    /* USER CODE END TIM3_IRQn 1 */
}
```