（三）系统时钟定时器

一、 本课内容概述

这一课的主要内容是 STM32 系统时钟定时器 systick 的配置以及如何产生精确延时。

通常实现 Delay(N)函数的方法为：

for(i = 0; i <= x; i ++);

x --- 对应于 对应于 N 毫秒的循环值

对于 STM32 系列微处理器来说，执行一条指令只有几十个 ns，进行 for 循环时，要实现 N 毫秒的 x 值非常大，而且由于系统频率的宽广，很难计算出延时 N 毫 秒的精确值。针对 STM32 微处理器，需要重新设计一个新的方法去实现该功能，以实现在程序中使用 Delay(N)。

对于 STM32 系列微处理器来说，执行一条指令只有几十个 ns，进行 for 循环时，要实现 N 毫秒的 x 值非常大，而且由于系统频率的宽广，很难计算出延时 N 毫 秒的精确值。针对 STM32 微处理器，需要重新设计一个新的方法去实现该功能，以实现在程序中使用 Delay(N)。

注： 全局变量 TimingDelay 必须定义为 volatile

二、 SysTick 的配置使用方法

外部晶振为 8MHz，9 倍频，系统时钟为 72MHz，SysTick 的最高频率为 9MHz（最大为 HCLK/8），在这个条件下，把 SysTick 效验值设置成 9，将 SysTick 时钟设置为 9MHz， 就能够产生 1us 的时间基值，即 SysTick 产生 1us 的中断。

使用 ST 的函数库使用 systick 的方法

1、调用 SysTick_CounterCmd() 失能 SysTick 计数器

2、调用 SysTick_ITConfig () 失能 SysTick 中断

3、调用 SysTick_CLKSourceConfig() 设置 SysTick 时钟源。

4、调用 SysTick_SetReload() 设置 SysTick 重装载值。

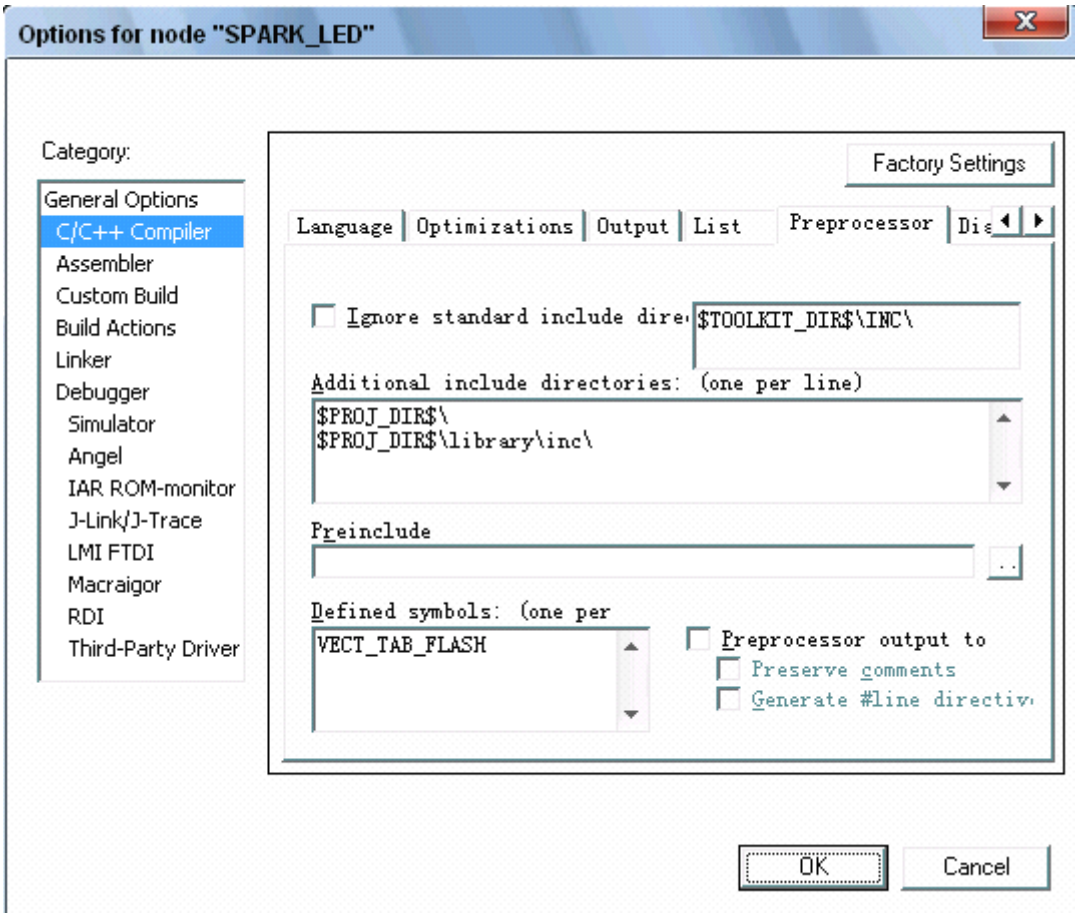
5、调用 SysTick_ITConfig () 使能 SysTick 中断

6、调用 SysTick_CounterCmd() 开启 SysTick 计数器

7、去掉 stem32f10x_conf.c 文件里面关于 SysTick 的注释，包含编译相关文件

8、在 FWLIB 里面加入 stm32f10x_systick.c

9、修改工程设置，把中断向量表指向 FLASH 空间：project-option-C/C++ Complier-Processor-Defined symbols 改为 VECT_TAB_FLASH



三、 相关程序

(1)建立 systick.c 文件

新建 systick.c 文件，作为 systick 相关函数的子函数，子函数里面应包括：SysTick_Init,SysTickDelayUs, TimingDelayMs_Decrement 三个函数，由于程序用到一个存放中断计数值的全局变量，而我们将次变量与 main.c 文件中定义，因此还要在这个文件中做外部引用定义：extern vu32 TimingDelay;，再把系统初始化头文件包括，就完成了此文件的建立：#include "systemInit.h"。

```
#include "systemInit.h"
extern vu32 TimingDelay;
```

SysTick 初始化函数

```
/*******************************************************************************
* Function Name  : SysTick_Config
* Description    : Configures SysTick
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
//SysTick 设置
void SysTick_Init(void)
```

```
{
    /* Disable SysTick Counter */
    SysTick_CounterCmd(SysTick_Counter_Disable);

    /* Disable the SysTick Interrupt */
    SysTick_ITConfig(DISABLE);

    /* Configure HCLK clock as SysTick clock source */
    SysTick_CLKSourceConfig(SysTick_CLKSource_HCLK_Div8);

    /* SysTick interrupt each 1000 Hz with HCLK equal to 72MHz */
    SysTick_SetReload(9);

    /* Enable the SysTick Interrupt */
    SysTick_ITConfig(ENABLE);
}
```

SysTickDelayMs 延迟一微秒函数

```
/*******************************************************************************
* Function Name  : SysTickDelayUs
* Description    : Inserts a delay time.
* Input          : nTime: specifies the delay time length, in milliseconds.
* Output         : None
* Return         : None
*******************************************************************************/
void SysTickDelayUs(u32 nTime)
{
  /* Enable the SysTick Counter */
  SysTick_CounterCmd(SysTick_Counter_Enable);

  TimingDelay = nTime;

  while(TimingDelay != 0);

  /* Disable SysTick Counter */
  SysTick_CounterCmd(SysTick_Counter_Disable);
  /* Clear SysTick Counter */
  SysTick_CounterCmd(SysTick_Counter_Clear);
}
```

TimingDelayMs_Decrement 中断调用函数

```
/*******************************************************************************
* Function Name  : TimingDelayMs_Decrement
* Description    : Decrements the TimingDelay variable.
* Input          : None
* Output         : TimingDelay
```

```
* Return             : None
*******************************************************************************/
void TimingDelay_Decrement(void)
{
  if (TimingDelay != 0x00)
  {
    TimingDelay--;
  }
}
```

SysTickHandler 中断进入函数

```
/*******************************************************************************
* Function Name  : SysTickHandler
* Description    : This function handles SysTick Handler.
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
void SysTickHandler(void)
{
  TimingDelay_Decrement();
}
```

NVIC_Configuration 中断向量表配置

```
/*******************************************************************************
* Function Name  : NVIC_Configuration
* Description    : Configures NVIC and Vector Table base location.
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
void NVIC_Configuration(void)
{
#ifdef  VECT_TAB_RAM
  /* Set the Vector Table base location at 0x20000000 */
  NVIC_SetVectorTable(NVIC_VectTab_RAM, 0x0);
#else  /* VECT_TAB_FLASH  */
  /* Set the Vector Table base location at 0x08000000 */
  NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
#endif
}
```

(2)修改 systemInit.h 文件

由于在其他 C 文件中要用到 systick.c 定义的一些函数，因此我们在 systemInit.h 中进行外部函数声明：

```
extern void SysTick_Init(void);
extern void SysTickDelayUs(u32);
```

```
extern void TimingDelay_Decrement(void);
```

(3)修改中断文件 stm32f10x_it.c

由于我们用到了 systick 系统时钟计数器中断计数的方法来产生延时，因此，我们必须在此文件中修改相应的中断函数，打开 stm32f10x_it.c 找到相应的函数，加入中断计数函数。

```
/*******************************************************************************
* Function Name  : SysTickHandler
* Description    : This function handles SysTick Handler.
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
void SysTickHandler(void)
{
  TimingDelay_Decrement();      //中断计数函数
}
```

由于在此文件中调用了 TimingDelay_Decrement()函数，因此必须将对此函数进行了外部声明的头文件 systemIni.h 包括进来：

```
/* Includes ------------------------------------------------------------------*/
#include "stm32f10x_it.h"
#include "systemInit.h"
```

(3)修改 main.c 文件

由于使用了全局变量 TimingDelay,因此要进行定义：vu32 TimingDelay;，而要完成 sistick 的初始化，我们要在系统初始化总程序 System_Init()里引用 SysTick_Init();函数。

根据需要编写我们的应用主程序，给出一个实现流水灯的例程：

```
/*******************************************************************************
* Function Name  : main
* Description    : Main program.
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
int main(void)
{
  System_Init();
  while (1)
  {
    LED1_H;
    SysTickDelayUs(500000);
    LED1_L;
    SysTickDelayUs(500000);
```

```
        LED2_H;
        SysTickDelayUs(500000);
        LED2_L;
        SysTickDelayUs(500000);
        LED3_H;
        SysTickDelayUs(500000);
        LED3_L;
        SysTickDelayUs(500000);
        LED4_H;
        SysTickDelayUs(500000);
        LED4_L;
        SysTickDelayUs(500000);
    }
}
```

至此，全部的修改与配置结束，enjoy it。