杂项库函数

杂项库函数一览

void NVIC_SetVectorTable (u32 NVIC_VectTable, u32 NVIC_Offset) **设置 NVIC 中断向量表位置和偏移**
void NVIC_LowPowerConfig (u8 NVIC_LowPowerMode, ControlStatus NewState) **选择低功耗模式**
void NVIC_CoreReset (void) **产生内核复位操作**
void NVIC_SetPendingSystemHandler (u32 SystemHandler) **设置系统中断挂起操作**
void SYSTICK_ClockSourceConfig (u32 SysTick_ClockSource) **Configure the SysTick clock source.** **配置系统定时器时钟源**
void SYSTICK_CounterCmd (u32 SysTick_Counter) **使能或禁止系统时钟计数**
void SYSTICK_IntConfig (ControlStatus NewState) **使能或禁止系统时钟中断**
void SYSTICK_SetReloadValue (u32 SysTick_Reload) **设置系统定时器重载计数值**

杂项库函数说明

NVIC_CoreReset 函数

```
void NVIC_CoreReset  ( void  )
```

产生内核复位操作.

返回值：无

NVIC_LowPowerConfig **函数**

```
void NVIC_LowPowerConfig  ( u8  NVIC_LowPowerMode,
  ControlStatus  NewState
 )
```

选择系统低功耗模式

参数:

NVIC_LowPowerMode,: 指定系统低功耗运行模式，这个参数是下面 3 个值之一。

NVIC_LOWPOWER_SEVONPEND

NVIC_LOWPOWER_SLEEPDEEP

NVIC_LOWPOWER_SLEEPONEXIT

NewState,: 新的运行模式状态. 这个参数是 ENABLE 或 DISABLE.

返回值：无

NVIC_SetPendingSystemHandle 函数

```
void NVIC_SetPendingSystemHandler  ( u32  SystemHandler )
```

设置系统中断挂起操作.

参数:

SystemHandler,: 指定系统中断. 这个值是下面 3 个值之一:

SYSTEMHANDLER_NMI

SYSTEMHANDLER_PSV

SYSTEMHANDLER_SYSTICK

返回值：无

NVIC_SetVectorTable 函数

```
void NVIC_SetVectorTable   ( u32   NVIC_VectTable,
  u32   NVIC_Offset
 )
```

功能：设置 NVIC 中断向量表的位置和偏移

参数：

NVIC_VectTable,: 指定中断向量表位于 FLASH 或 RAM 中. 这个参数是下面的 2 个值之一：

NVIC_VECTTABLE_RAM

NVIC_VECTTABLE_FLASH

NVIC_Offset,: 向量表的偏移地址. 这个值一定是 0x100 的倍数.

返回值：无

SYSTICK_ClockSourceConfig 函数

```
void SYSTICK_ClockSourceConfig   ( u32   SysTick_ClockSource )
```

配置系统定时器时钟源 Configure the SysTick clock source.

参数:

SysTick_ClockSource,: 指定系统定时器时钟源. 这个参数是下面 2 个值之一:

SYSTICK_SRC_STCLK : 外部参考时钟作为系统定时器时钟源。

SYSTICK_SRC_FCLK : AHB 时钟作为系统定时器时钟源。

返回值：无

举例：利用系统定时器产生精确的硬件延时 n 毫秒

原理：设置系统定时器选用外部 9Mhz 时钟，这样计数 9000 个时钟脉冲就是 1ms。没 1ms 产生一个系统时钟中断，在中断服务程序中将延时函数的延时以毫秒为单位的参数减 1，减到 0 时延时时间到。

```
#define SYSTICKRELOAD    9000
vu32 DelayTime=0；中断服务程序使用的延时变量。
void SPI_LCD_SysTick_Config(void)
{
    SYSTICK_ClockSourceConfig(SYSTICK_SRC_STCLK);    // STCLK = 9MHz
    SYSTICK_SetReloadValue(SYSTICKRELOAD);             // 1ms
    SYSTICK_IntConfig(ENABLE);                          //  使能系统定时器中断
}
```

供中断服务程序调用的延时函数

```
void SYSTICK_TimingDelay(void)
{
```

```
    if(DelayTime != 0x00)
      DelayTime--;
}
```

延时毫秒的延时函数，nTime 的值是程序需要的延时毫秒数。

```
void DelayMS(u32 nTime)
{
    SYSTICK_CounterCmd(SYSTICK_COUNTER_ENABLE); /* 使能系统定时器 */
    DelayTime = nTime;
    while(DelayTime != 0);
    SYSTICK_CounterCmd(SYSTICK_COUNTER_DISABLE); /* 禁止系统定时器计数 r */
    SYSTICK_CounterCmd(SYSTICK_COUNTER_CLEAR); /* 系统定时器清零 */
}
```

在 ht32f176x_276x_it.c 文件中，找到 SysTick_Handler 函数

```
void SysTick_Handler(void)
{
     SYSTICK_TimingDelay ();
}
```