零死角玩转STM32—M3系列

SysTick—系统定时器

淘宝：firestm32.taobao.com

论坛： www.firebbs.cn

扫描进入淘宝店铺

主讲内容

01 SysTick简介

02 SysTick功能框图讲解

03 SysTick定时实验讲解

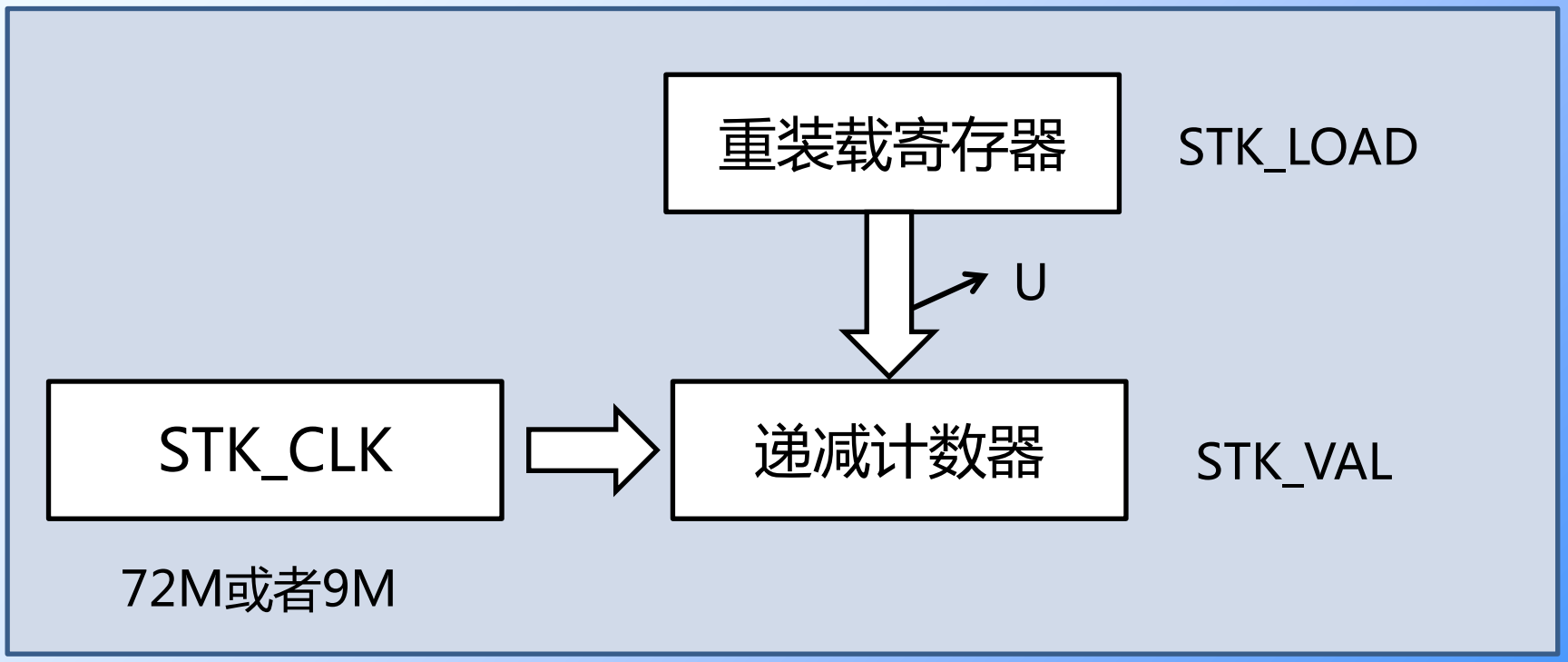
参考资料:《零死角玩转STM32》

“SysTick—系统定时器”章节

SysTick简介

SysTick：系统定时器，24位，只能递减，存在于内核，嵌套在NVIC中，所有的Cortex-M内核的单片机都具有这个定时器。

SysTick功能框图



counter在时钟的驱动下，从reload初值开始往下递减计数到0，产生中断和置位COUNTFLAG标志。然后又从reload值开始重新递减计数，如此循环。

SysTick寄存器

表 18-2 SysTick 控制及状态寄存器

位段	名称	类型	复位值	描述
16	COUNTFLAG	R/W	0	如果在上次读取本寄存器后， SysTick 已经计到了 0，则该位为 1。
2	CLKSOURCE	R/W	0	时钟源选择位，0=AHB/8，1=处理器时钟 AHB
1	TICKINT	R/W	0	1=SysTick 倒数计数到 0 时产生 SysTick 异常请求，0=数到 0 时无动作。也可以通过读取 COUNTFLAG 标志位来确定计数器是否递减到 0
0	ENABLE	R/W	0	SysTick 定时器的使能位

表 18-3 SysTick 重装载数值寄存器

位段	名称	类型	复位值	描述
23:0	RELOAD	R/W	0	当倒数计数至零时，将被重装载的值

表 18-4 SysTick 当前数值寄存器

位段	名称	类型	复位值	描述
23:0	CURRENT	R/W	0	读取时返回当前倒计数的值，写它则使之清

SysTick定时时间计算

- 1-t：一个计数循环的时间，跟reload和CLK有关
- 2-CLK：72M或者9M，由CTRL寄存器配置
- 3-RELOAD：24位，用户自己配置

SysTick定时时间计算

- t = reload * (1/clk)
- Clk = 72M时，t = (72) *(1/ 72 M)= 1US
- Clk = 72M时，t = (72000) *(1/ 72 M)= 1MS

时间单位换算:

1s = 1000ms = 1000 000 us = 1000 000 000ns

SysTick寄存器

SysTick寄存器结构体

在固件库文件：core_cm3.h中定义

```
typedef struct
{
  __IO uint32_t CTRL;                    /*!< 控制及状态寄存器 */
  __IO uint32_t LOAD;                    /*!< 重装载数值寄存器*/
  __IO uint32_t VAL;                     /*!< 当前数值寄存器*/
  __I  uint32_t CALIB;                   /*!< 校准寄存器 */
} SysTick_Type;
```

SysTick库函数

SysTick配置库函数
在固件库文件：core_cm3.h中定义

```
// 这个 固件库函数 在 core_cm3.h中
static __INLINE uint32_t SysTick_Config(uint32_t ticks)
{
  // reload 寄存器为24bit, 最大值为2^24
  if (ticks > SysTick_LOAD_RELOAD_Msk)  return (1);

  // 配置 reload 寄存器的初始值
  SysTick->LOAD  = (ticks & SysTick_LOAD_RELOAD_Msk) - 1;

  // 配置中断优先级为 1<<4-1 = 15, 优先级为最低
  NVIC_SetPriority (SysTick_IRQn, (1<<__NVIC_PRIO_BITS) - 1);

  // 配置 counter 计数器的值
  SysTick->VAL   = 0;

  // 配置systick 的时钟为 72M
  // 使能中断
  // 使能systick
  SysTick->CTRL  = SysTick_CTRL_CLKSOURCE_Msk |
                   SysTick_CTRL_TICKINT_Msk   |
                   SysTick_CTRL_ENABLE_Msk;
  return (0);
}
```

SysTick库函数

SysTick配置库函数

在固件库文件：core_cm3.h中定义

```
static __INLINE void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)
{
  /* 设置优先级 for Cortex-M3 系统中断 */
  if(IRQn < 0) {
    SCB->SHP[((uint32_t)(IRQn) & 0xF)-4] = ((priority << (8 - __NVIC_PRIO_BITS)) & 0xff); }
  /* 设置优先级 for 外设中断  */
  else {
    NVIC->IP[(uint32_t)(IRQn)] = ((priority << (8 - __NVIC_PRIO_BITS)) & 0xff);    }
}
```

SysTick中断优先级

1-SysTick属于内核里面的外设，他的中断优先级跟片上的外设的中断优先级相比，哪个高？

2-systick中断优先级配置的是scb->shprx寄存器；而外设的中断优先级配置的是nvic->iprx，有优先级分组，有抢占优先级和子优先级的说法。

SysTick中断优先级

1-STM32里面无论是内核还是外设都是使用4个二进制位来表示中断优先级。

2-中断优先级的分组对内核和外设同样适用。当比较的时候，只需要把内核外设的中断优先级的四个位按照外设的中断优先级来分组来解析即可，即人为的分出抢占优先级和子优先级。

实验设计

1-编写一个微秒延时函数

2-编写一个毫秒延时函数

零死角玩转STM32—M3系列

THANKS

论坛： www.firebbs.cn

淘宝：firestm32.taobao.com

扫描进入淘宝店铺