

## MDK 配置向导详解

//----- <<< Use Configuration Wizard in Context Menu >>> -----

<<< Use Configuration Wizard in Context Menu >>>

配置向导开始(必须的)

<<< end of configuration section >>>

配置向导结束 (可以不要的)

关键字	说明
<h>	头标签: 表示包含多个子项的一个组, 无开关控制
<e>	使能标签: 表示包含多个子项的一个组, 该组开关由这项控制 (后面有个 <input checked="" type="checkbox"/> )
<e0>, <e1> <e1.4>	0, 1, 表示操作对象序号, 后面有解释 1 表示操作对象序号, 4 表示 BIT4
</h> or </e>	头标签、使能标签结束
<i>	提示信息, 可以连续写几条提示信息。当鼠标放在要操作项目上将出现提示信息
<q> <sup>s</sup>	位操作用一个 <input type="checkbox"/> 控制, <input type="checkbox"/> =0, <input checked="" type="checkbox"/> =1
<o> <sup>s</sup> <o0>, <o1> <o1.4..6>	数字标签 0, 1, 表示操作对象序号 1 表示操作对象序号, 4..6 操作对象的 BIT6~BIT4
<s> <s1.30>	字符串标签: 操作的对象是字符串 操作的对象是字符串, 并且限制字符串长度 (这里是 30)
<0-31>	数值范围, 超过该范围无意义
<0-100:10>	10 进制数值范围
<0x40-0x1000:0x10>	16 进制数值范围
<0=> <1=> <2=>.....	数值选择, 后面可以跟注释文字, 如: <0=> SYSTICKCLK = HCLK/8
<#+1> <#-1> <#*8> <#/3>	数值做运算, 分别表示+1, -1, *8, /3

### <h>标签

最简单的<h></h>例子是.s 文件中的堆栈配置

```
Stack Configuration
  Stack Size (in Bytes)      0x0000 0200
```

下面是代码

```
;// <h> Stack Configuration
;// <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
;// </h>
Stack_Size EQU 0x00000200
```

### <e>标签

<e>标签, 开关控制模块

<i> Default: SYSTICKCLK = HCLK/8 //提示信息

```
System Timer Configuration
  System Timer clock source selection SYSTICKCLK = HCLK/8
  SYSTICK period [ms] 10
  System Timer interrupt enabled  Default: SYSTICKCLK = HCLK/8
```

### 系统时钟配置

0 表示操作对象序号:0

2 表示 BIT2

```

//===== System Timer Configuration
// <e0> System Timer Configuration
// <o1.2> System Timer clock source selection
// <i> Default: SYSTICKCLK = HCLK/8
// <0=> SYSTICKCLK = HCLK/8
// <1=> SYSTICKCLK = HCLK
// <o2> SYSTICK period [ms] <1-1000:10>
// <i> Set the timer period for System Timer.
// <i> Default: 1 (1ms)
// <o1.1> System Timer interrupt enabled
// </e>

#define __SYSTICK_SETUP          1 //操作对象序号: 0
#define __SYSTICK_CTRL_VAL      0x00000006 //操作对象序号: 1
#define __SYSTICK_PERIOD        0x0000000A //操作对象序号: 2

```

<i>提示信息

<0=> <1=> 值和文字选择

#define \_\_SYSTICK\_PERIOD 0x0000000A //操作对象序号: 2

解释:

### <o>标签

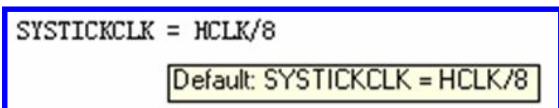
<e0> <o1.2> <o2> <o1.1>

字母后的第 1 个数值表示要操作对象的序号, .1 .2 ‘.’ 点后面的数字表示要操作对象的位  
例: .1=BIT1, .2= BIT2

<o1.2>表示操作的第 1 个对象的 BIT2

<o1.2..5>表示操作的第 1 个对象的 BIT5, BIT4, BIT3, BIT2

<i> 前一个项目提示帮助, 将鼠标放在要操作项目上将出现提示信息



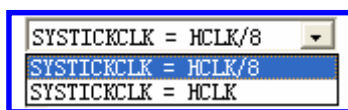
<0=>, <1=>: 值和文字的选择

例:

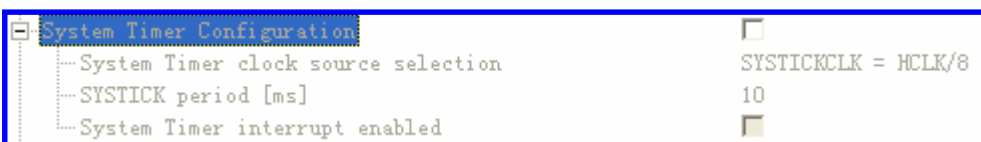
```

// <0=> SYSTICKCLK = HCLK/8
// <1=> SYSTICKCLK = HCLK

```



<e> </e>是成对出现, 包含一对<e> </e>里面的所以项通过该复选框启用/禁止



#define \_\_SYSTICK\_SETUP 0 //注意这里为 0 表示未禁止

下面 4 行整体意思表示:

```

// <o1.2> System Timer clock source selection
// <i> Default: SYSTICKCLK = HCLK/8
// <0=> SYSTICKCLK = HCLK/8
// <1=> SYSTICKCLK = HCLK

```

<i>提示信息

<0=> <1=> 值和文字选择

系统定时器时钟频率由第 1 个操作数的 BIT2 决定

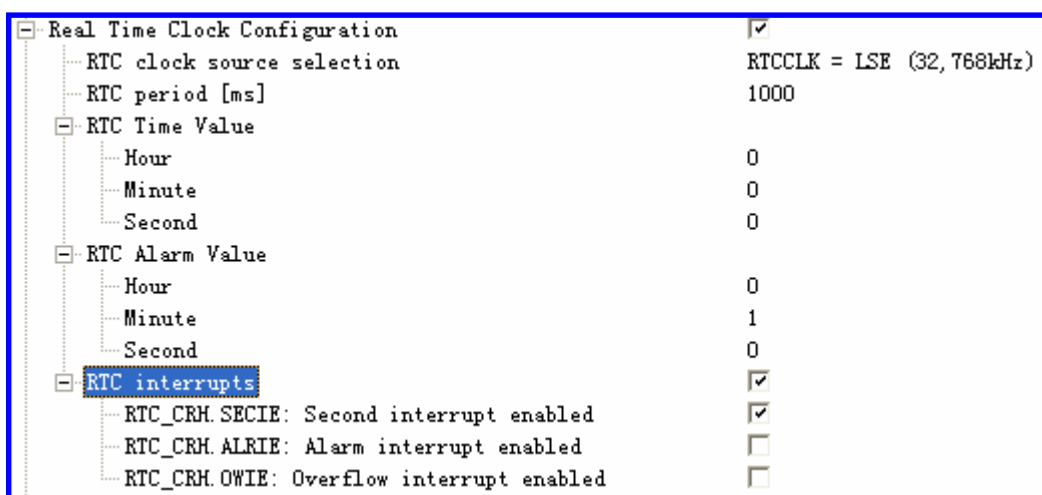
0: SYSTICKCLK = HCLK/8;      1: SYSTICKCLK = HCLK

操作对象序号说明:

```
// <e1.13> Enable Chip Select 0 (CS0)
// </e>
_WDWORD(0xFFE00000, 0x010024A9); // EBI_CS0: Flash
```

上面一行表示 2 个操作对象

例 2: RTC 配置



```
//===== Real Time Clock Configuration
// <e0> Real Time Clock Configuration
// <o1.8..9> RTC clock source selection
// <i> Default: No Clock
//           <0=> No Clock
//           <1=> RTCCLK = LSE (32,768kHz)
//           <2=> RTCCLK = LSI (32 kHz)
//           <3=> RTCCLK = HSE/128
// <o2> RTC period [ms] <10-1000:10>
// <i> Set the timer period for Real Time Clock.
// <i> Default: 1000 (1s)
// <h> RTC Time Value
//   <o3> Hour <0-23>
//   <o4> Minute <0-59>
//   <o5> Second <0-59>
// </h>
// <h> RTC Alarm Value
//   <o6> Hour <0-23>
//   <o7> Minute <0-59>
//   <o8> Second <0-59>
// </h>
// <e9> RTC interrupts
//   <o10.0> RTC_CRH_SECIE: Second interrupt enabled
//   <o10.1> RTC_CRH_ALRIE: Alarm interrupt enabled
//   <o10.2> RTC_CRH_OWIE: Overflow interrupt enabled
```

```
// </e>
// </e>
#define __RTC_SETUP 0 //操作对象序号: 0
#define __RTC_CLKSRC_VAL 0x00000100 //操作对象序号: 1
#define __RTC_PERIOD 0x000003E8 //操作对象序号: 2
#define __RTC_TIME_H 0x00 //操作对象序号: 3
#define __RTC_TIME_M 0x00 //操作对象序号: 4
#define __RTC_TIME_S 0x00 //操作对象序号: 5
#define __RTC_ALARM_H 0x00 //操作对象序号: 6
#define __RTC_ALARM_M 0x01 //操作对象序号: 7
#define __RTC_ALARM_S 0x00 //操作对象序号: 8
#define __RTC_INTERRUPTS 0x00000001 //操作对象序号: 9
#define __RTC_CRH 0x00000001 //操作对象序号: 10
```

解释:

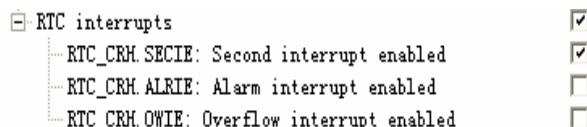
```
// <h> RTC Time Value
// <o3> Hour <0-23>
// <o4> Minute <0-59>
// <o5> Second <0-59>
// </h>
```



<h></h>是成对出现, 表示里面的子项都属于同一类, 同一组, 可以嵌套使用

<0-23> 数值范围, 超过范围是无效的

```
// <e9> RTC interrupts
// <o10.0> RTC_CRH.SECIE: Second interrupt enabled
// <o10.1> RTC_CRH.ALRIE: Alarm interrupt enabled
// <o10.2> RTC_CRH.OWIE: Overflow interrupt enabled
// </e>
```



注意: <h></h>和<e> </e>区别是<h>是无参数的, 只是表示包含子项是一个组, <e>是有参数的比如<e0>, <e1>,

<e>项目后面还有个  , 这是<h>没有的。

## <q>标签

<q>标签实际就是最简单的位操作标签, 位操作作用一个  控制,  =0,  =1



```
// <q0.4> q 标签测试
#define QLABEL 0x00000000
```



BIT4 变化了

```
// <q0.4> q 标签测试
#define QLABEL 0x00000010
```

## <s>标签

<q>标签就是字符串标签啦, 这个标签很少用到

```
Change ID My User ID 8
Change Password String My Password1234567890123456789
```

```
// <s> Change ID
```

```
#define ID "My User ID 8"
```

无限制的字符

```
// <s0.30> Change Password String
```

```
char pw[] = "My Password1234567890123456789";
```

限制的字符串长度为 30