例程十　独立看门狗

介绍

独立看门狗模块可以用于解决处理器因为硬件或软件的故障所发生的错误。它由一个内部的128kHz的LSI阻容振荡器作为时钟源驱动，因此即使是主时钟失效时它仍然照常工作。

独立看门狗功能说明

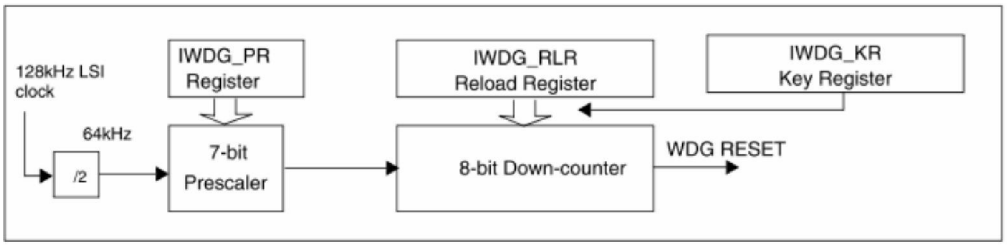
*图24*是独立看门狗模块的功能框图。

当在键寄存器(IWDG_KR)中写入数值0xCC后，独立看门狗就被启动了，计数器开始从它的复位值0xFF开始递减计数，当计数减到0x00时就会产生一个复位信号(WDG RESET)。

使用IWDG_PR和IWDG_RLR寄存器配置独立看门狗。IWDG_PR寄存器是用于选择驱动计数器时钟的预分频系数。每当KEY_REFRESH的数值(0xAA)写入到IWDG_KR寄存器时，独立看门狗将用IWDG_RLR的数值刷新计数器的内容，从而避免了产生看门狗的复位。

IWDG_PR和IWDG_RLR寄存器具有写保护功能，要修改它们前，需首先在IWDG_KR寄存器写入KEY_ACCESS代码(0x55)；在IWDG_KR写入0xAA将恢复写保护状态。

图24　独立看门狗框图



硬件看门狗功能

如果在IWDG_HW选择字节中使能了硬件看门狗的功能，在芯片上电时看门狗的功能被自动开启，如果软件不能及时操作键寄存器，则在计数器达到0x00时产生复位。关于选择字节的内容请参考数据手册中的说明。

超时周期

超时周期由计数器数值和时钟预分频器决定，下表列出了它们的数值

表26　看门狗超时周期(假定计数器时钟为64kHz)

预分频系数	PR[2:0]	最短超时(RL[7:0]=0x00)	最长超时(RL[7:0]=0xFF)
/4	0	62.5 µs	15.90 ms
/8	1	125 µs	31.90 ms
/16	2	250 µs	63.70 ms
/32	3	500 µs	127 ms
/64	4	1.00 ms	255 ms
/128	5	2.00 ms	510 ms
/256	6	4.00 ms	1.02 s

14.3 IWDG寄存器

14.3.1 键寄存器(IWDG_KR)

地址偏移值：0x00

复位值：未定义

7	6	5	4	3	2	1	0
KEY[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw

位7:0	**KEY[7:0]：**键值 软件必须在规定的时间内写入KEY_REFRESH数值，否则当计数器数值达到0时，看门狗会产生一个复位。 **KEY_ENABLE** 数值=0xCC 写入KEY_ENABLE数值将启动IWDG。 **KEY_REFRESH** 数值=0xAA 写入KEY_REFRESH数值将刷新IDDG。 **KEY_ACCESS** 数值=0x55 写入KEY_ACCESS数值将允许对受保护的IWDG_PR和IWDG_RLR寄存器的操作(见*14.2*)。

14.3.2 预分频寄存器(IWDG_PR)

地址偏移值：0x01

复位值：0x00

7	6	5	4	3	2	1	0
保留					PR[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw

位7:3	保留，必须保持为0。
位2:0	**PR[2:0]：**预分频系数 这些位是写保护的(见*14.2*)。它们用于指定对计数器时钟分频的分频系数。 000：分频系数=4 001：分频系数=8 010：分频系数=16 011：分频系数=32 100：分频系数=64 101：分频系数=128 110：分频系数=256 111：保留

14.3.3 重装载寄存器(IWDG_RLR)

地址偏移值：0x02

复位值：0xFF

7	6	5	4	3	2	1	0
RL[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw

位7:0	**RL[7:0]：**看门狗计数器重装载数值 这些位是写保护的(见*14.2*)。每次在IWDG_KR寄存器中写入0xAA时，这个寄存器中的内容会被传送到看门狗的计数器中，看门狗的计数器将重新从这个值开始计数。超时时间由这个数值和时钟的预分频系数决定，见*表26*。

14.3.4 IWDG寄存器映像和复位数值

表27 IWDG寄存器映像

地址偏移值	寄存器	7	6	5	4	3	2	1	0
0x00	IWDG_KR	KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0
	复位值	x	x	x	x	x	x	x	x

当你打开独立看门狗的时候，IWDG->KR=0xcc，在你看门狗的定时的范围内就要不断 Reloads IWDG counter，IWDG->KR=0xaa，俗称“喂狗”。

下面从主函数看起，一步一步看清看门的设置。

```
int main(void)
{

  /* Infinite loop */

  /*设置内部时钟16M为主时钟*/

  CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);
  /*!<Set High speed internal clock  */
   LED_Init();
   SetLedON();

   Test_IWDGReset();
   IWDG_Configuration();
   Tim1_Init();
   __enable_interrupt();

   while (1)
   {

     LED_Display();


    }
```

主要看看 2 个函数就行了，看门测试函数 Test_IWDGReset() 和看门狗配置函数 IWDG_Configuration()，先看看门狗配置函数原型

```
void IWDG_Configuration(void)
{
  /* Enable the IWDG*/
  IWDG_Enable();
  /* Enable the access to the IWDG registers*/
  IWDG_WriteAccessCmd(IWDG_WriteAccess_Enable);
  /* Fixe IWDG Reset period */
  IWDG_SetPrescaler(IWDG_Prescaler_256);
  IWDG_SetReload(0xFF);
  /* Refresh IWDG */
  IWDG_ReloadCounter();
}
```

这个函数就是配置看门狗定时时间为 1.02s，也就是说在这 1.02S 的时间内你至少要喂一次狗，否则的话就会产生复位了。如果想定时其他的时间的话，就修改这 2 个的函数就行，IWDG_SetPrescaler(IWDG_Prescaler_256)、IWDG_SetReload(0xFF);至于参数可以参考以上开头所说的看门狗介绍。

```
void Test_IWDGReset(void)
{
  FlagStatus IwdgFlag;

  /*Get IWDG Reset Status */
  IwdgFlag = RST_GetFlagStatus(RST_FLAG_IWDGF);
  /* Test if a IWDG Reset has occured */
  if (IwdgFlag)
  {
    LED_ON_OFF();
    /* Clear IWDGF Flag */
    RST_ClearFlag(RST_FLAG_IWDGF);

   }
}
```

这个函数是查询看门是否复位了，如果是看门狗复位了，4 盏灯不断在闪，记得要清楚看门狗复位标志位。

实验效果：

下载本例程到风驰电子 STM8 开发板上方向每隔一定的时间的看门狗复位，4 盏灯一闪一闪的。

风驰电子祝您学习愉快！！！ ～～～