

看门狗

1. 实验目的

- 了解 STC15W4K32S4 系列单片机系统复位的方式，掌握看门狗定时器的原理。
- 掌握单片机看门狗定时器的寄存器配置及程序设计。

2. 实验内容

- 编写程序观察开启看门狗 WDT 后及时喂狗系统正常运行的现象。
- 编写程序观察开启看门狗 WDT 但未及时喂狗系统复位的现象。

3. 硬件设计

3.1. 单片机系统复位

STC15W4K32S4 系列单片机的复位分为硬件复位和软件复位两种。

硬件复位时，所有的寄存器的值会复位到初始值，系统会重新读取所有的硬件选项。同时根据硬件选项所设置的上电等待时间进行上电等待。

软件复位时，除与时钟相关的寄存器保持不变外，其余的所有寄存器的值会复位到初始值，软件复位不会重新读取所有的硬件选项。

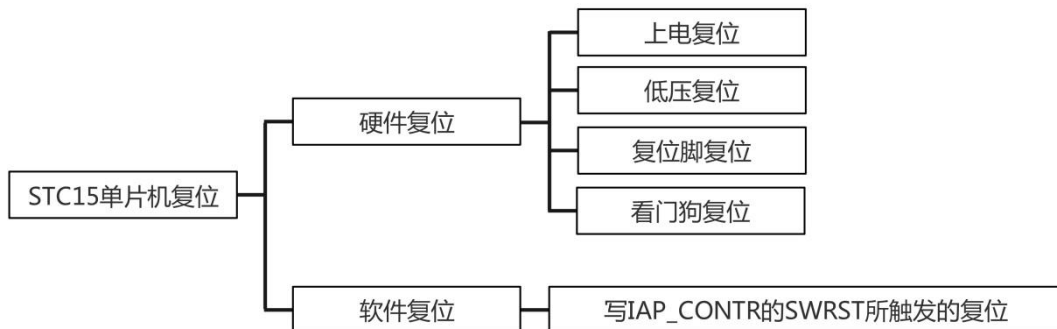


图 1: STC15W4K32S4 系列单片机复位分类

STC15W4K32S4 系列单片机软件复位是通过 IAP 控制寄存器 IAP_CONTR 的 SWRST 使能的，IAP_CONTR 寄存器具体如下。

IAP_CONTR (IAP 控制寄存器)										
① 寄存器名	符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
	IAP_CONTR	C7H	IAPEN	SWBS	SWRST	CMD_FAIL	-	WT [2:0]		
② 启动选择	SWBS: 软件复位启动选择									
	0: 软件复位后从用户程序区开始执行代码。用户数据区的数据保持不变。									
	1: 软件复位后从系统 ISP 区开始执行代码。用户数据区的数据会被初始化。									
③ 软件复位使能位	SWRST: 软件复位触发位									
	0: 对单片机无影响									
	1: 触发软件复位									

图 2: IAP 控制寄存器 IAP_CONTR

✧ 注：如果开启软件复位后，还需操作寄存器 SWBS 位以确定开始执行代码的区域。

3.2. 单片机看门狗简介

在工业控制/汽车电子/航空航天等需要高可靠性的系统中，为了防止“系统在异常情况下，受到干扰，MCU/CPU 程序跑飞，导致系统长时间异常工作”的情况发生，而设计了一种用于检测程序运行状态的芯片或电路，俗称看门狗（即 Watchdog）。

看门狗通常有一个输入信号和一个输出信号。输入信号也称“喂狗”，输出信号关联 MCU 的复位端。系统运行后启动看门狗计数器，看门狗开始计数，当计数满后未收到清零信号（喂狗），看门狗计数器溢出，触发看门狗中断，输出控制 MCU 复位。

看门狗可分为独立于单片机芯片外的看门狗芯片和集成于单片机芯片内的看门狗外设两种。按照工作原理，独立看门狗和窗口看门狗是最为常见的两种看门狗类型。

✧ 注：以上描述是针对看门狗的通用性描述，不同型号的单片机看门狗可能会有差异。

3.3. STC15W4K32S4 系列单片机看门狗介绍

STC15W4K32S4 系列单片机看门狗实际上是一个 15 位的定时器或计数器，简称 WDT。单片机运行工作中，如果开启了看门狗外设，则看门狗会对单片机机器周期脉冲不停计数，单片机程序如果不能及时对看门狗的定时器复位（俗称“喂狗”），看门狗就会在其定时器计满溢出时产生一个复位信号，从而使系统重启。

下图可比较形象的诠释看门狗原理，单片机 CPU 必须在看门狗计数器没有溢出前喂狗，否则计数器一旦溢出，则系统重启。

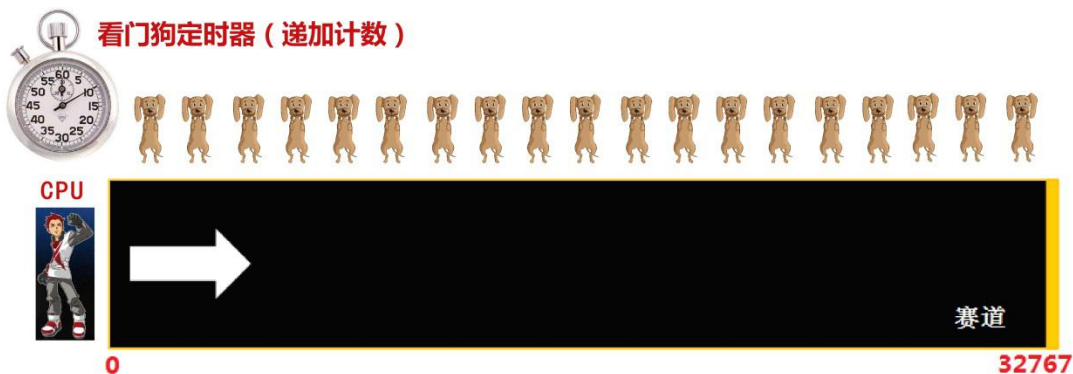


图 3：看门狗工作原理示意图

✧ 注：所谓的喂狗，就是将看门狗计数器计数清零。

3.3.1. 看门狗控制寄存器 WDT_CONTR

看门狗控制寄存器 WDT_CONTR 不支持位寻址，该寄存器的 B0~B2 位用于设置看门狗定时器时钟分频系数，寄存器 B3 位是单片机空闲模式时的看门狗控制位，寄存器 B4 位是清零看门狗计数器计数控制位，寄存器 B5 位是看门狗使能控制位，寄存器 B7 位是看门狗溢出标志位。



图 4: 看门狗控制寄存器 WDT_CONTR

3.3.2. 看门狗溢出时间计算

下面给出看门狗溢出时间计算公式:

$$\begin{aligned}
 \text{看门狗溢出时间} &= \text{计数器单次计数时间} \cdot \text{溢出前计数总次数} \\
 &= \frac{1}{f_{\text{CNT}}} \cdot (1 + 32767) \\
 &= \frac{\text{PSC}}{f_{\text{WDT}}} \cdot 32768 = \frac{2^{\text{PS}[2:0]+1}}{f_{\text{SYSCLK}}/12} \cdot 32768
 \end{aligned}$$

1 看门狗定时器输入时钟频率

2 分频因子

3 看门狗定时器时钟频率

4 系统时钟

5 看门狗定时器时钟分频系数

图 5: 看门狗溢出时间计算公式

✧ 注: STC 官方手册上面没有给出看门狗溢出时间的推导过程, 也没有明确说明看门狗定时器的时钟频率, 以系统时钟 12 分频后给看门狗定时器提供时钟是最合理解释。

举例, 配置 WDT_CONTR 寄存器 B2 位 PS2 为 0, 配置 WDT_CONTR 寄存器 B1 位 PS1 和 B0 位 PS0 均为 1, 系统时钟为 11.0592MHZ, 计算看门狗溢出时间。

- 看门狗定时器时钟分频系数 PS[2:0]值为 011, 转成十进制是 3, 进而得出分频因子 PSC 值为 16。
- 看门狗溢出时间: $(16 \cdot 12 \cdot 32768) / 11059200 = 0.5688\text{s} = 568.8\text{ms}$ 。

3.4. 看门狗配置步骤

针对 STC15W4K32S4 系列单片机看门狗配置过程如下:

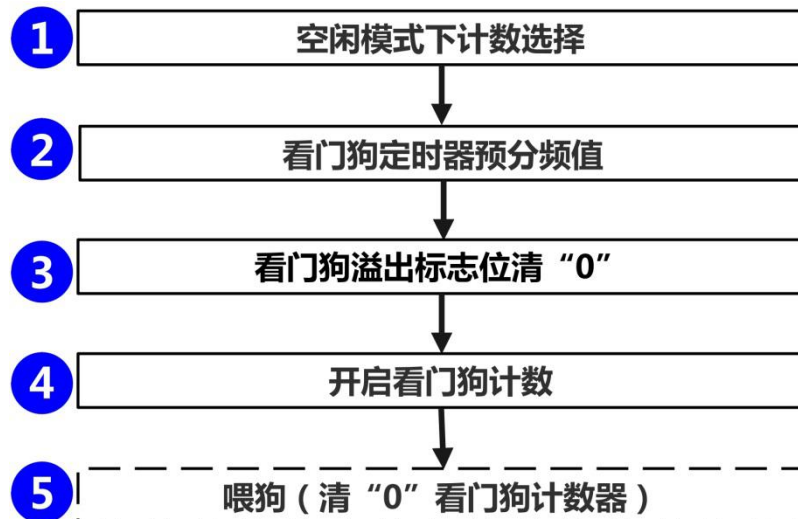


图 6: 看门狗配置步骤

✧ 注：实验例程即是按照上述配置步骤操作寄存器相关位实现，后有详述。

4. 软件设计

4.1. 看门狗实验实验（及时喂狗）

✧ 注：本节的实验源码是在“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”的基础上修改。本节对应的实验源码是：“实验 2-7-1：WDT 看门狗实验（及时喂狗）”。

4.1.1. 工程需要用到的 c 文件

本例需要用到的 c 文件如下表所示，工程需要添加下表中的 c 文件。

表 1: 实验需要用到的 c 文件

序号	文件名	后缀	功能描述
1	led	.c	包含与用户 led 控制有关的用户自定义函数。
2	wdt	.c	看门狗有关的用户自定义函数。
3	delay	.c	包含用户自定义延时函数。

4.1.2. 头文件引用和路径设置

■ 需要引用的头文件

```

1. #include "delay.h"
2. #include "wdt.h"
3. #include "led.h"
  
```

■ 需要包含的头文件路径

本例需要包含的头文件路径如下表：

表 2：头文件包含路径

序号	路径	描述
1	..\ Source	led.h、wdt.h 和 delay.h 头文件在该路径，所以要包含。
2	..\User	15W4KxxS4.h 头文件在该路径，所以要包含。

MDK 中点击魔术棒，打开工程配置窗口，按照下图所示添加头文件包含路径。

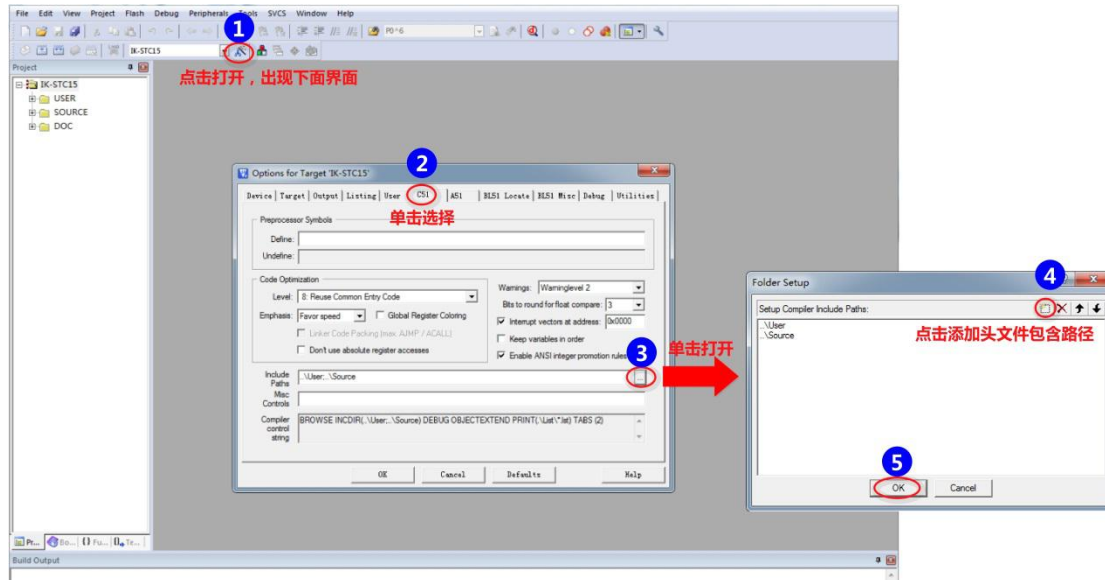


图 7：添加头文件包含路径

4.1.3. 编写代码

首先，在 wdt.c 文件中编写看门狗初始化函数 WDT_Init，代码如下。

程序清单：看门狗初始化函数

```

1.  /*****
2.  * 描 述：看门狗初始化
3.  * 入 参：无
4.  * 返回值：无
5.  *****/
6.  void WDT_Init(void)
7.  {
8.      WDT_CONTR &= 0xF7;          //IDLE_WDT 位置 0，看门狗定时器在空闲模式下不计数
9.      WDT_CONTR &= 0xFB;          //PS2 位置 0，设置看门狗定时器预分频值
10.     WDT_CONTR |= 0x03;          //PS1、PS0 位置 1，设置看门狗定时器预分频值
11.     WDT_CONTR &= 0x7F;          //WDT_FLAG 位置 0，看门狗溢出标志位清零
12.     WDT_CONTR |= 0x20;          //EN_WDT 位置 1，开启看门狗定时器
13. }

```

然后，编写喂狗函数，代码如下。

程序清单：喂狗函数

```
1.  /*****
2.  * 描 述：喂狗
3.  * 入 参：无
4.  * 返回值：无
5.  *****/
6. void WDG_Feed(void)
7. {
8.     WDT_CONTR |= 0x10;          //CLR_WDT 位置 1，看门狗定时器重新计数
9. }
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制蓝灯亮，主循环中有软件延时和喂狗操作，喂狗之后会翻转红灯并熄灭蓝灯，以方便观察实验现象。

代码清单：主函数

```
1. int main()
2. {
3.     ///////////////////////////////////
4.     //注意：STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
5.     //      高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
6.     //相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
7.     //      P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
8.     ///////////////////////////////////
9.     P0M1 &= 0x3F; P0M0 &= 0x3F;    //设置 P0.6~P0.7 为准双向口
10.
11.     leds_off();                    //熄灭开发板上的 2 个指示灯(DS1 和 DS2)
12.     WDT_Init();                    //看门狗初始化,看门狗溢出时间约 568.8ms (11.0592MHZ)
13.     delay_ms(200);
14.     led_on(LED_1);                  //点亮蓝色指示灯 DS1
15.     while(1)
16.     {
17.         //软件延时 200ms,该延时是模拟喂狗前实际执行用户应用代码所用时长
18.         delay_ms(200);
19.         //喂狗
20.         WDG_Feed();
21.         led_off(LED_1);              //熄灭蓝色指示灯 DS1
22.         led_toggle(LED_2);           //翻转红色指示灯 DS2
23.         //软件延时 200ms,该延时是模拟喂狗后实际执行用户应用代码所用时长
24.         delay_ms(200);
25.     }
26. }
```

4.1.4. 实验步骤

1. 解压“···\第3部分：配套例程源码\1-基础实验程序\”目录下的压缩文件“实验2-7-1：WDT看门狗实验（及时喂狗）”，将解压后得到的文件夹拷贝到合适的目录，如“D:\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“···\WDT\projec”目录下的工程“WDT.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WDT.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：
 - 1) 蓝灯先亮，约 200s 后蓝灯灭红灯亮。
 - 2) 之后红灯间隔 400ms 闪烁，蓝灯一直灭。

4.2.看门狗实验（未及时喂狗）

✧ 注：本节的实验源码是在“实验2-7-1：WDT看门狗实验（及时喂狗）”的基础上修改。本节对应的实验源码是：“实验2-7-2：WDT看门狗实验（未及时喂狗）”。

4.2.1. 工程需要用到的c文件

本实验需要用到的头文件以及添加头文件包含路径的方法请参考“实验2-7-1：WDT看门狗实验（及时喂狗）”部分。

4.2.2. 编写代码

首先，在 wdt.c 文件中编写看门狗初始化函数 WDT_Init，代码如下。

程序清单：看门狗初始化函数

```
1.  /*****
2.  * 描 述：看门狗初始化
3.  * 入 参：无
4.  * 返回值：无
5.  *****/
6. void WDT_Init(void)
7. {
8.     WDT_CONTR &= 0xF7;          //IDLE_WDT 位置 0，看门狗定时器在空闲模式下不计数
9.     WDT_CONTR &= 0xFB;          //PS2 位置 0，设置看门狗定时器预分频值
10.    WDT_CONTR |= 0x03;           //PS1、PS0 位置 1，设置看门狗定时器预分频值
11.    WDT_CONTR &= 0x7F;           //WDT_FLAG 位置 0，看门狗溢出标志位清零
12.    WDT_CONTR |= 0x20;           //EN_WDT 位置 1，开启看门狗定时器
13. }
```

然后，编写喂狗函数，代码如下。

程序清单：喂狗函数

```
1.  /*****
2.  * 描 述：喂狗
3.  * 入 参：无
4.  * 返回值：无
5.  *****/
6.  void WDG_Feed(void)
7.  {
8.      WDT_CONTR |= 0x10;          //CLR_WDT 位置 1，看门狗定时器重新计数
9.  }
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制蓝灯亮，主循环中有软件延时和喂狗操作，喂狗之后会翻转红灯并熄灭蓝灯，以方便观察实验现象。

代码清单：主函数

```
1.  int main()
2.  {
3.      //注意：STC15W4K32S4 系列的芯片，上电后所有与 PWM 相关的 IO 口均为
4.      //      高阻态，需将这些口设置为准双向口或强推挽模式方可正常使用
5.      //相关 IO：P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
6.      //      P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
7.      //
8.      //设置 P0.6~P0.7 为准双向口
9.      P0M1 &= 0x3F; P0M0 &= 0x3F;
10.
11.     leds_off();          //熄灭开发板上的 2 个指示灯(DS1 和 DS2)
12.     WDT_Init();          //看门狗初始化，看门狗溢出时间约 568.8ms (11.0592MHZ)
13.     delay_ms(200);
14.     led_on(LED_1);       //点亮蓝色指示灯 DS1
15.     while(1)
16.     {
17.         //软件延时 200ms，该延时是模拟喂狗前实际执行用户应用代码所用时长
18.         delay_ms(200);
19.         //喂狗
20.         WDG_Feed();
21.         led_off(LED_1);   //熄灭蓝色指示灯 DS1
22.         led_toggle(LED_2); //翻转红色指示灯 DS2
23.         //软件延时 500ms，该延时是模拟喂狗后实际执行用户应用代码所用时长
24.         delay_ms(500);
25.     }
26. }
```

4.2.3. 实验步骤

1. 解压“···\第3部分：配套例程源码\1 - 基础实验程序\”目录下的压缩文件“实验2-7-2：WDT看门狗实验（未及时喂狗）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“···\WDT\projec”目录下的工程“WDT.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WDT.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：
 - 1) 蓝灯先亮，约 200s 后蓝灯灭红灯亮。
 - 2) 之后红灯和蓝灯交替闪烁。