嵌入式系统原理及应用

第9章　定时器

张帆

中南大学自动化学院

第9章 定时器

9.1 S5PC100 PWM 定时器（不讲）

9.2 S5PC100 看门狗定时器

小结

思考与练习

我们为什么需要定时器？

- 问题：

  既然嵌入式处理器（MCU）执行每条指令的时间都是固定的，可以通过软件延时的方法实现任何长度的定时。为什么还需要一个成为“定时/计数器”的硬件电路？

- 回答：

1. 定时期间CPU不能做任何其它事情，因为一旦软件定时被其它事情打断定时将不再准确。
2. 定时程序将占用宝贵的数据/程序存储器资源。

定时器是什么？

1、是MCU上的一种硬件电路，是一种集成的片上外设。

2、其本质是时序计数器。

3、时钟源是可配置的。

4、可配置增/减计数器构成。

5、初值是可配置的。

定时器的应用

- 在嵌入式系统应用中，经常会遇到以下情况：
  - 周期性执行某任务，如每隔固定时间完成一次AD采集；
  - 延时一定时间执行某个任务，如交通灯信号变化；
  - 显示实时时间，如万年历；
  - 产生不同频率的波形，如MP3软解码与播放；
  - 产生不同脉宽的波形，如驱动伺服电机；
  - 测量脉冲的个数，如测量转速；
  - 测量脉冲的宽度，如测量频率。

定时器的应用

- 定时器的应用根据计数脉冲的来源，可以归纳为两种情况：
  - 一种是**定时**（即对内部脉冲的计数操作），完成与时间相关的任务；
  - 另一种是**计数**（即对外部输入的计数操作），完成脉冲个数、宽度和频率的测量等。

9.2 S5PC100 看门狗定时器

定时器/计数器简称定时器，其作用主要包括产生各种时间间隔、记录外部事件的数量等，是计算机中最常用、最基本的部件之一。

看门狗（WatchDog）定时器的特点是， 需要不停地接受信号（一些外置看门狗芯片）或重新设置计数值（如 S5PC100 的 看门狗控制器），保持计数值不为 0。一旦一段时间接收不到信号，或计数值为 0， 看门狗将发出复位信号复位系统或产生中断。

看门狗的作用是微控制器受到干扰进入错误状态后，使系统在一定时间间隔内复位 。因此看门狗是保证系统长期、可靠和稳定运行的有效措施。

目前大部分的嵌 入式芯片内都集成了看门狗定时器来提高系统运行的可靠性。

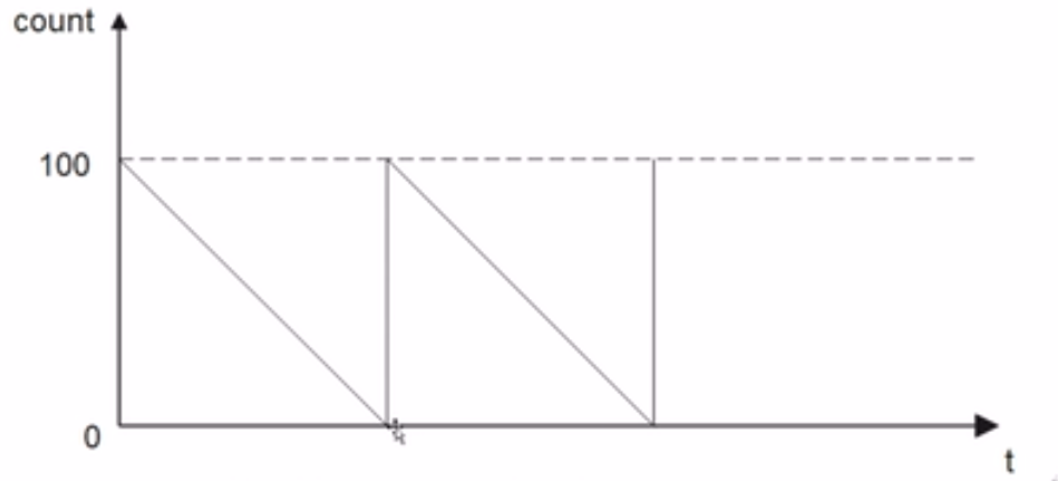
9.2 S5PC100 看门狗定时器

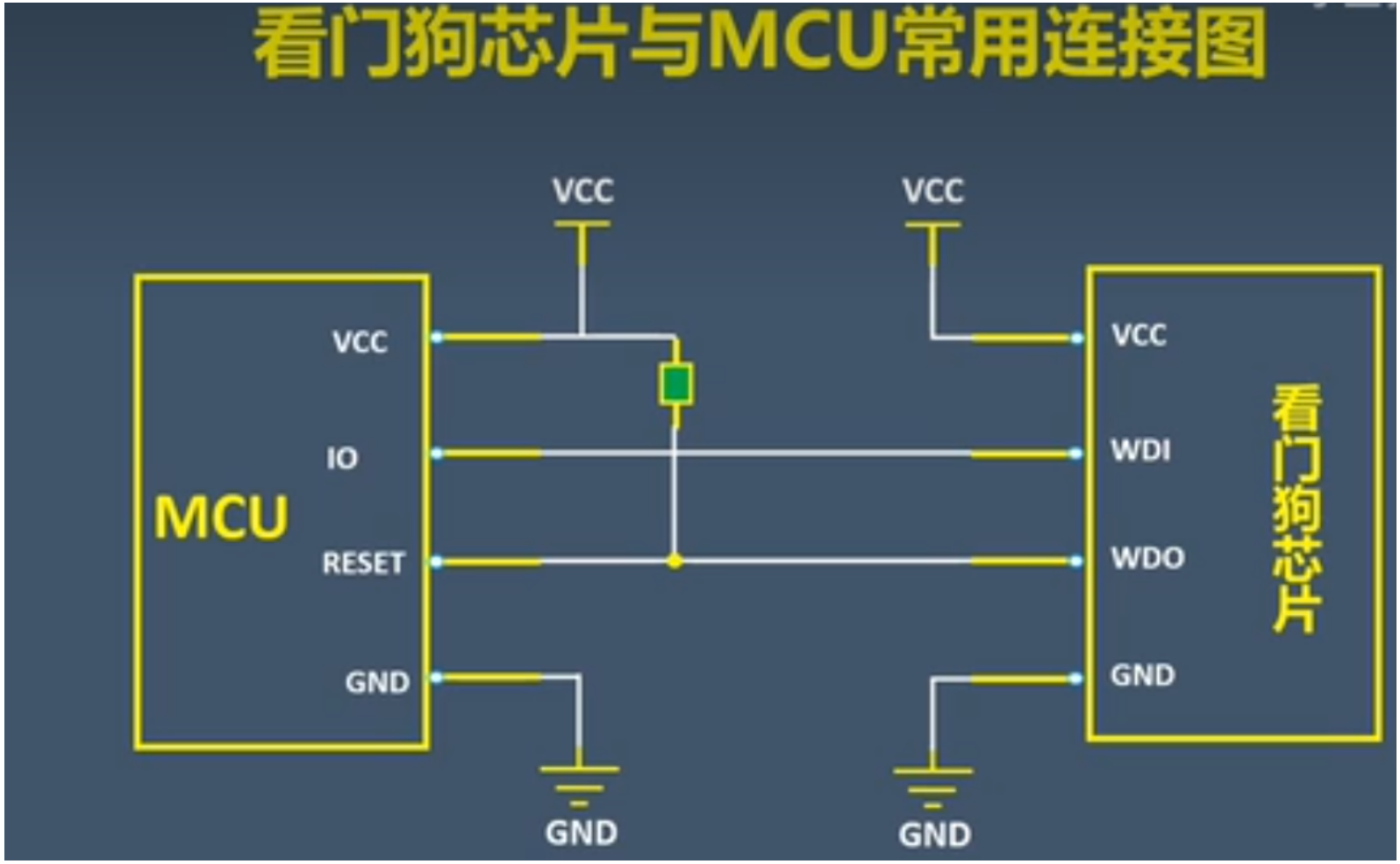
S5PC100 处理器的看门狗是当系统被故障（如噪声或者系统错误）干扰时，用 于微处理器的复位操作，也可以作为一个通用的 16 位定时器来请求中断操作。

主要特性有如下两个。

（1）通用的中断方式的 16 位定时器。

（2）当计数器减到 0（发生溢出）时，产生 一个128 PCLK（时钟源，可分频）周期的复位信号




看门狗芯片与MCU常用连接图
VCC
VCC
VCC
IO
RESET
GND
MCU
VCC
WDI
WDO
GND
看门狗芯片
GND
GND

9.2 S5PC100 看门狗定时器

概述

包括一个预比例因子放大器？，一个四分频的分频器，一个 16 位计数器

看门狗的时钟信号源来自 PCLK，为了得到宽范围的看门狗信号，PCLK 先被预分频，然后再经过分频器分频。

预分频比例因子和分频器的分频值，都由看门狗控制寄存器（WTCON）决定，预分频比例因子的范围是 0～255，分频器的分频比可以是 16、32、64 或者 128。

看门狗定时器时钟周期的计算如下：

t_watchdog = 1/(PCLK/(Prescaler value + 1)/Division_factor)

Prescaler value + 1（why+1）；

Division_factor 是四分频的分频比，可以是 16、32、64 或者 128。

S5PC100 看门狗定时器概述

一旦看门狗定时器被允许，看门狗定时器数据寄存器（WTDAT）的值就不能被自动地装载到看门狗计数器（WTCNT）中。

因此，看门狗启动前要将一个初始值写入看门狗计数器（WTCNT）中（初始化要配置）。

当 S5PC100 用嵌入式 ICE 调试的时候，看门狗定时器的复位功能就不被启动，看门狗定时器能从 CPU 内核信号判断出当前 CPU 是否处于调试状态。

如果看门狗定时器确定当前模式是调试模式，尽管看门狗能产生溢出信号，但是仍然不会产生复位信号。

看门狗定时器寄存器

1．看门狗定时器控制寄存器（WTCON）

包括：是否启用看门狗定时器、4 个分频比的选择（不仅仅）、 是否允许中断产生、是否允许复位操作等。

如果想把看门狗定时器当作一般的定时器使用，应该中断使能，禁止看门狗定时器复位。

WTCON	位	描　　述	复位值
保留	[31:16]	保留	0
预分频值	[15:8]	预分频值： 有效数值范围位<0 to 255>	0x80
保留	[7:6]	保留	00
看门狗定时器	[5]	看门狗时钟使能位： 0 = 禁止 1 = 使能	1
始终选择	[4:3]	时钟分频值： 00 = 16 01 = 32 10 = 64 11 = 128	00
中断产生器	[2]	使能/屏蔽中断功能 0 = 禁止 1 = 使能	0
保留	[1]	保留	0
复位使能/屏蔽	[0]	1 =打开 S5PC100 看门狗产生复位信号 0 =禁止上述功能	1

看门狗定时器寄存器

2．看门狗定时器数据寄存器（WTDAT）

WTDAT 用于指定超时时间，在看门狗把复位功能禁止并打开中断使能后，此时看门狗定时器就是一个普通的定时器，使用方法和普通定时器一样。当使能复位的功能后，由于 WTCNT 的值减到 0 时，系统就会复位，所以 WTDAT 的值装不进看门狗计数寄存器（WTCNT）中。复位后初始值为 0x8000。

WTDAT	位	描　　述	复位值
保留	[31:16]	保留	0
计数重载值	[15:0]	看门狗重载数值寄存器	0x8000

看门狗定时器寄存器

3．看门狗计数寄存器（WTCNT）

WTCNT 放置看门狗定时器工作时计数器的当前计数值（时刻变化，做减法）。

注意：不会自动装载WTDAT的初值，使能看门狗之前必须手动设置初值

WTCNT	位	描　　述	复位值
保留	[31:16]	保留	0
计数值	[15:0]	看门狗当前计数寄存器	0x8000

看门狗定时器程序编写

1．看门狗软件程序设计流程

看门狗的工作结果：系统复位或者产生中断，所以不需要外围硬件电路。

通过对寄存器组操作来实现功能，即控制寄存 器（WTCON）、数据寄存器（WTDAT）、计数寄存器（WTCNT）的操作。

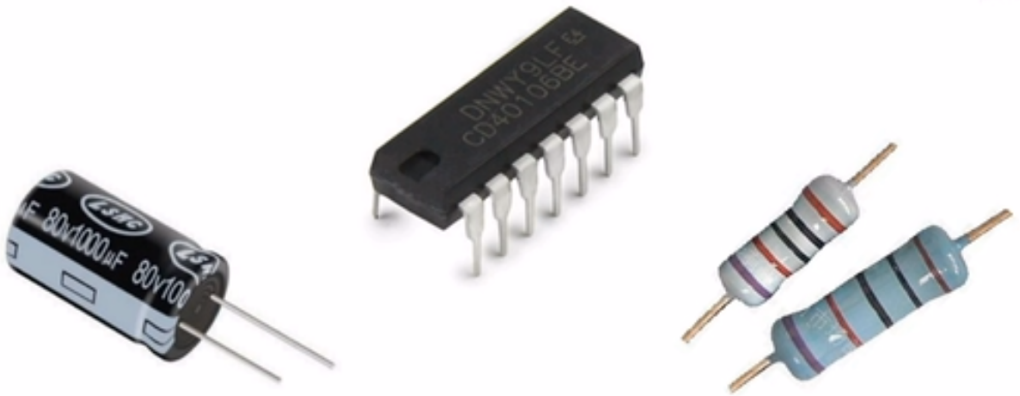
一般流程如下。

（1）设置看门狗中断操作，包括全局中断和看门狗中断的使能及看门狗中断向 量的定义，如果只是进行复位操作，这一步可以不用设置。

（2）对看门狗控制寄存器（WTCON）的设置，包括设置预分频比例因子、分频器的分频值、中断使能和复位使能等。

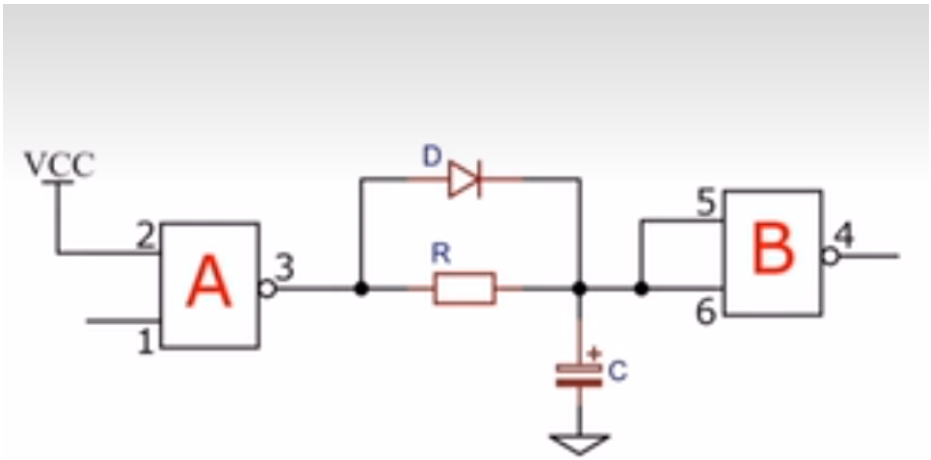
（3）对看门狗数据寄存器（WTDAT）和看门狗计数寄存器（WTCNT）的设置。

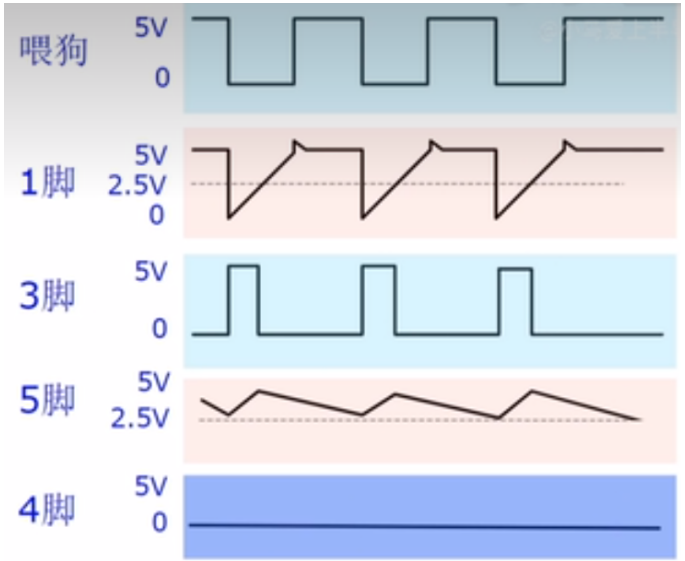
（4）启动看门狗定时器



一个硬件的看门狗电路

两个与非门





看门狗定时器程序编写

2．看门狗寄存器的定义

```
/*
*WATCHDOG 寄存器的定义
*/
typedef struct {
        unsigned int WTCON ;
        unsigned int WTDAT ;
        unsigned int WTCNT ;
        unsigned int WTCLRINT ;
}wdt;
#define WDT (* (volatile wdt *)0xEA200000 )
```

看门狗定时器程序编写

3．看门狗寄存器的初始化

```
void wdt_init()
{
        WDT.WTCNT = 0X277e;
        WDT.WTDAT = 0X277e; //
        WDT.WTCON = (1<<0)|(3<<3)|(1<<5)|(255<<8);
                                        // 66MHZ 预分频 255 得到 255824Hz
                                        再进行 128 分频得到 f = 2022HZ
                                        // data * 1/f = 5 延时 5 s 得到 data = 0x277e
}
```

看门狗定时器程序编写

4．看门狗主程序的编写

```
#include "s5pc100.h"
int main()
{
int i;
GPG3.GPG3CON = (~(0xf<<4)&GPG3.GPG3CON) | (0X1<<4);
GPG3.GPG3DAT = 0x2; // 点亮 LED 用来测试看门狗的复位功能
wdt_init();
while(1);
return 0;
 }
```

5．观察实验结果

程序运行 5 s 后，LED 就会熄灭，因为此时的 CPU 发生了复位

小知识

- 喂狗：独立看门狗一般用来检测和解决由程序引起的故障，比如一个程序正常运行的时间是50ms，在运行完这个程序之后紧接着进行喂狗，我们设置独立看门狗的定时溢出时间为60ms，比我们需要监控的程序 50ms 多一点，如果超过 60ms 还没有喂狗，那就说明我们监控的程序出故障了，跑飞了，那么就会产生系统复位，让程序重新运行。

```
void main(void)
{
    /* 各种初始化 */
    {
        ......
    }

    /* 启动独立看门狗 */
    IWDG_Start();
    while(1)
    {
        /* 需要被监控的代码 */
        {
            ......
        }

        /* 喂狗 */
        IWDG_Feed();
    }
}
```

小知识

```
// 喂狗
void IWDG_Feed(void)
{
   WDT.WTCNT = 0X277e;
   WDT.WTDAT = 0X277e;// 把重装载寄存器的值放到计数器中，喂狗，防止IWDG复位
}


   while(1)
   {
      if( Key_Scan(KEY1_GPIO_PORT,KEY1_PIN) == KEY_ON  )
      {
         // 喂狗，如果不喂狗，系统则会复位，复位后亮红灯，如果在1s
         // 时间内准时喂狗的话，则会亮绿灯
         IWDG_Feed();
         //喂狗后亮绿灯
         LED_GREEN;
      }
```

小结

- 看门狗控制器的工作原理
- S5PC100芯片中看门狗控制器的操作方法

思考与练习

9-1 在控制系统中为何要加入看门狗功能？

9-2 编程实现1秒内不对看门狗实现喂狗操作，看门狗会自动复位。

谢谢！