

单片机（AT89C51）定时/计数器实验案例

原创

叶玉恒 于 2020-04-21 15:30:22 发布 9479 收藏 74

分类专栏：[单片机](#) 文章标签：[单片机](#)

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_45629315/article/details/105657169

版权



[单片机](#) 专栏收录该内容

8 篇文章 23 订阅

订阅专栏

继上篇的《单片机（AT89C51）定时/计数器详解及其实验案例》由于各种原因里面没有实验案例现在在此补上。

单片机（AT89C51）定时/计数器详解见上篇：https://blog.csdn.net/weixin_45629315/article/details/105418365

目录

案例分析

实验一（已知8051单片机的fosc=12MHz用T1定时，试编程由P1.0引脚分别输出周期为2ms的方波）

方法一（查询法）：

方法二（中断法）：

实验结果图：

实验二（已知8051单片机的fosc=12MHz用T1定时。试编程由P1.0引脚分别输出周期为2s的方波）：

方法一（查询法）：

方法二（中断法）：

实验结果图：

实验三（使用定时器1，以定时方法在P1.0输出周期为20ms，占空比为20%的矩形脉冲，设单片机晶振频率fosc为12MHz，编程实现）：

代码：

实验结果图：

案例分析

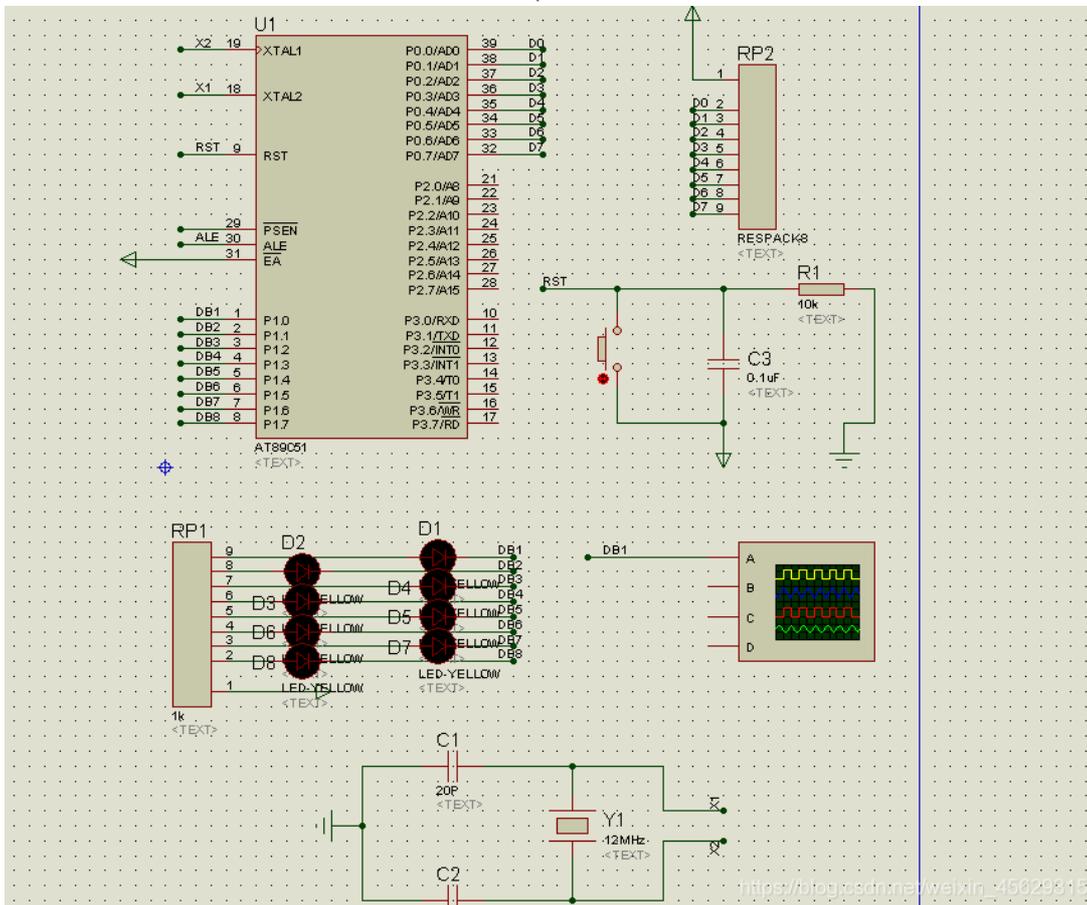
总结了经常遇到的三种类型：

已知8051单片机的fosc=12MHz用T1定时，试编程由P1.0引脚分别输出周期为2ms的方波。

2. 已知8051单片机的fosc=12MHz用T1定时。试编程由P1.0引脚分别输出周期为2s的方波。

3. 使用定时器1,以定时方法在P1.0输出周期为20ms,占空比为20%的矩形脉冲,设单片机晶振频率fosc为12MHz,编程实现。

为了方便分析我们先把实验图给出来(示波器是为了让我们的实验结果更加的明显):



实验一（已知8051单片机的fosc=12MHz用T1定时，试编程由P1.0引脚分别输出周期为2ms的方波）

先计算我们的初值(结果可以不用详细的计算，直接在代码中表示就好了): **T初值=2¹⁶-1ms/1us**，什么你还不知道这个算式怎么写？看看我上一篇吧，链接给你丢这里

了: https://blog.csdn.net/weixin_45629315/article/details/105418365

方法一（查询法）：

```

#include <reg51.h>
sbit P1_0=P1^0; //将P1_0位定义为 P1.0
void main()
{
char i;
TMOD=0x10; //使用定时器T1的模式1
TR1=1; //启动定时器T1
for( ; ; ) //无限循环
{ TH1=(65536-1000)/256;
TL1=(65536-1000)%256; //定时器T1赋初值
do { } while (!TF1); //查询计数溢出
TF1=0;
P1_0=! P1_0; //取反
}
}

```

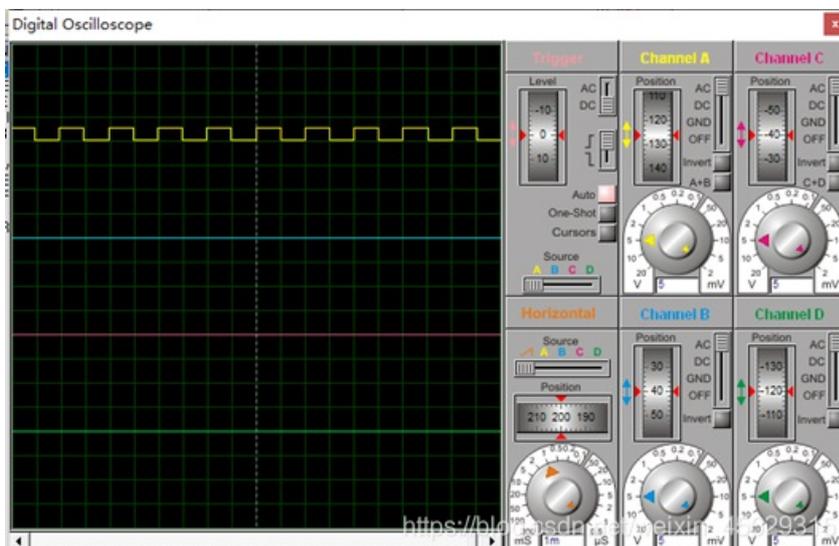
方法二（中断法）：

```

#include <reg51.h> //包含特殊功能寄存器库
sbit P1_0=P1^0;
void main()
{
EA=1; //开总中断
ET1=1; //定时器T1中断允许
TMOD=0x10; //使用定时器T1的模式1
TH1=(65536-1000)/256; //定时器T1的高8位赋初值
TL1=(65536-1000)%256; //定时器T1的低8位赋初值
TR1=1; //启动定时器T1
while(1);
}
void time1_int(void) interrupt 3 //中断服务程序
{
P1_0=!P1_0; //取反
TH1=(65536-1000)/256; TL1=(65536-1000)%256; //定时器T1赋初值
}

```

实验结果图：



实验二（已知8051单片机的 $f_{osc}=12\text{MHz}$ 用T1定时。试编程由P1.0引脚分别输出周期为

2s的方波)：

先计算我们的初值(结果可以不用详细的计算，直接在代码中表示就好了): **T初值=2¹⁶-1s/1us**,通过计算会发现我们需要的初值是个负数说明了我们想要的周期超过了定时计数器的最大范围，那我们需要怎么解决呢？可以想把这个周期的时间划分为几个相等小块，然后再计数一下，就可以实现大的周期化成小的周期。

注：周期不要化的太小否则会出问题（大家可以尝试一下）要很小的话会超出char的范围，要是真的想改的很小，把count的类型改为int即可。

我们可以将周期化为初值为20ms一个小周期然后循环的执行50遍就是我们的1s，那么初值就是：T初值=2¹⁶-20ms/1us（当然这边的方法只要是可行的就行，不一定要和我一样进行划分奥）

方法一（查询法）：

```
# include <reg51.h>
sbit P1_0=P1^0; //将P1_0位定义为 P1.0
void main()
{
char i;
char count=0;
TMOD=0x10; //使用定时器T1的模式1
TR1=1; //启动定时器T1
for( ; ;) //无限循环
{ TH1=(65536-20000)/256;
TL1=(65536-20000)%256; //定时器T1赋初值
do { } while (!TF1); //查询计数溢出
TF1=0; //计数器溢出后,将TF1清0
count=count+1;
if (count==50)
{
P1_0=! P1_0; //取反
count=0;
}
}
}
```

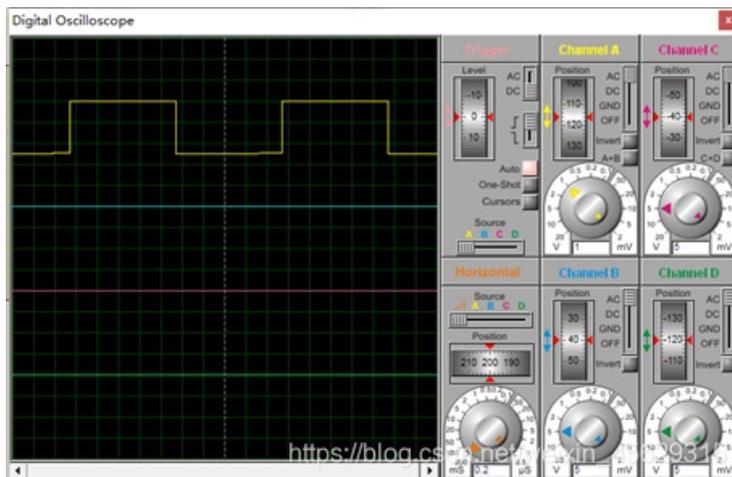
方法二（中断法）：

```

#include <reg51.h> //包含特殊功能寄存器库
sbit P1_0=P1^0;
char count;
void main()
{
    TMOD=0x10; //使用定时器T1的模式1
    TH1=(65536-20000)/256; //定时器T1的高8位赋初值
    TL1=(65536-20000)%256; //定时器T1的低8位赋初值
    EA=1; //开总中断
    ET1=1; //定时器T1中断允许
    TR1=1; //启动定时器T1
    count=0;
    while(1);
}
void time0_int(void) interrupt 3 //中断服务程序
{
    count++;
    TH1=(65536-20000)/256; TL1=(65536-20000)%256; //定时器T1赋初值
    if(count==50)
    {
        P1_0=!P1_0; //取反
        count=0;
    }
}

```

实验结果图：



实验三（使用定时器1,以定时方法在P1.0输出周期为20ms，占空比为20%的矩形脉冲，设单片机晶振频率 f_{osc} 为12MHz，编程实现）：

这个实验较前面的几个实验有不同之处：首先要了解什么叫**占空比**？（占空比是指在一串理想的脉冲序列中（如方波），正脉冲的持续时间与脉冲总周期的比值），这样我们就可以计算了：

占空比为20%，那么高电平的占比时间就是 $20ms * 20\% = 4ms$ ；低电平的占比时间就是 $20ms * 80\% = 16ms$

可以看出高电平时间比上低电平时间为--1:4，可以借鉴实验二的方法设置count

那么我们就可以设置初值为4ms，然后执行一次跳低电平，执行5次跳高电平（并且计数器count置零）

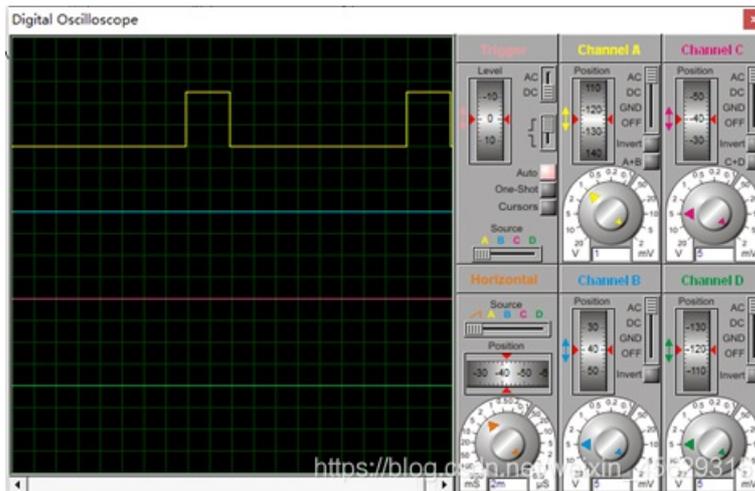
代码：

```

#include <reg51.h> //包含特殊功能寄存器库
sbit P1_0=P1^0;
char count=0; //设置计数器的count为全局变量
void main()
{
    TMOD=0x10; //使用定时器T1的模式1
    TH1=(65536-4000)/256; //定时器T1的高8位赋初值(设置一个循环为4000)
    TL1=(65536-4000)%256; //定时器T1的低8位赋初值
    EA=1; //开总中断
    ET1=1; //定时器T1中断允许
    TR1=1; //启动定时器T1
    P1_0=1;
    while(1);
}
void time0_int(void) interrupt 3 //中断服务程序
{
    count++;
    TH1=(65536-4000)/256;
    TL1=(65536-4000)%256; //定时器T1赋初值
    if(count==1)
    {
        P1_0=0; //取反
    }
    if(count==5)
    {
        P1_0=1;
        count=0;
    }
}

```

实验结果图：



写的不好，仅供参考

No pains No results