

# 单片机精确软件延时设计方法研究

韩薇薇

(扬州高等职业技术学校 电子工程系 江苏 扬州 225003)

**摘 要:** 在单片机控制应用中,经常会遇到需要短时间延时的情况,有时还需要很高的精度。然而每次延时都使用定时器/计数器是没有必要的也是不现实的,利用软件延时程序来完成一些需要延时的部分势在必行。从研究软件延时程序模式出发,指出计算整个延时时间的具体方法,并在此基础上归纳一种设计精确软件延时的数学算法。

**关键词:** 单片机; 软件延时; 数学算法

**中图分类号:** NT **文献标识码:** A **文章编号:** 1671—7597 (2011) 0310081—02

## 1 软件延时的基本原理

单片机在进行控制时,CPU过一段时间再去作某件事情,称之为延时。延时分为硬件延时和软件延时。硬件延时是指使用定时/计数器来实现延时,这种方法不占用CPU的工作时间。软件延时是利用程序中的指令在被执行时要花费一定的机器周期来实现的,占用CPU的工作时间。软件延时主要与两个因素有关,一是所用的晶振,二是延时程序的循环次数。单片机所用的晶振一经确定之后,延时时间就被指令的循环次数所唯一确定了,如单片机的晶振频率为6MHz,则一个机器周期为 $2\mu s$ 。下面根据指令循环的次数的不同,先来介绍软件延时时长计算方法(设晶振频率为12MHz)。

## 2 软件延时时长的计算方法

软件延时常用来汇编指令。

表1 常用的汇编指令及执行时间

指令	占用的机器周期(T)
MOV Rn, #data	1
DJNZ Rn,rel	2
RET	2
NOP	1
ACALL addr11	2
LCALL addr16	2

### 1) 单循环延时程序

DEL: MOV R0, #X; 1T

DEL1: DJNZ R0, DEL1; 2XT

RET; 2T

延时时间:  $t=1+2X+2=2X+3=2R0+3$

因为X的取值范围是[0, 255], X=0时延时时间最长。所以延时时间t的取值范围是[5, 515] $\mu s$ 。

### 2) 二重循环程序

DEL: MOV R1, #Y ; 1T

DEL1: MOV R0, #X ; 1YT

DEL2: DJNZ R0, DEL2; 2XYT

DJNZ R1, DEL1; 2YT

RET; 2T

延时时间:  $t=1+1Y+2XY+2Y+2$

$=2XY+3Y+3=2R0R1+3R1+3$

因为X、Y的取值范围均为[0, 255], X=Y=0时延时时间最长。所以延时时间t的取值范围是[8, 131843] $\mu s$ 。

### 3) 三重循环程序

DEL: MOV R2, #Z ; 1T

DEL1: MOV R1, #Y ; 1ZT

DEL2: MOV R0, #X ; 1Y2T

DEL3: DJNZ R0, DEL3; 2XYZT

DJNZ R1, DEL2; 2Y2T

DJNZ R2, DEL1; 2ZT

RET; 2T

延时时间:

$t=1+1Z+1YZ+2XYZ+2YZ+2Z+2$

$=2XYZ+3YZ+3Z+3$

$=2R0R1R2+3R1R2+3R2+3$

同理, 延时时间t的取值范围是[11, 33751811] $\mu s$ 。

综上所述, 由数学归纳法可得出N(N一般不大于8)重循环的延时时间计算公式:

$$2\prod_{i=0}^{N-1} R_i + 3 \sum_{j=1}^{N-1} \left( \prod_{k=j}^{N-1} R_k \right) + 3$$

## 3 C语言中延时程序的设计方法

在用C语言进行编程时涉及到的软件延时程序,其设计与汇编语言的设计方法本质相同。例如,在C语言的一个实例程序中用for语句编写了一个延时子程序(该程序要求延时100ms)。

```
delay(void)
{
    unsigned char i, j;
    for(i=250; i>=1; i--)
        for(j=200; j>=1; j--)
            {}
}
```

利用keil c的反汇编功能,通过调用disassembly window命令查看到的汇编程序如下。

```
C:0X0060 7F7D MOV R7, #0XFA
C:0X0062 7F32 MOV R6, #0XC8
C:0X0064 DEFE DJNZ R6, C:0064
C:0X0066 DFFA DJNZ R7, C:0062
C:0X0068 RET
```

带入所得公式计算结果是100753 $\mu s$ ,利用keil c的测试工具得出的延时时间为100755 $\mu s$ ,误差是由调用指令引起,后有详细分析。可以得出结论,此处C语言中的for语句不仅与汇编语言中的DJNZ命令起到的作用类似且计算延时的方法也完全相同。即上面得出的公式同样适用于计算用C语言编写的延时程序。下面考虑一个相反的问题,也是本文着重讨论的。

## 4 软件延时程序的设计算法

在实际的编程过程中,人们往往想得到一个规定了延时长度的延时子程序,即在已知延时时间的前提下设计循环次数。设计延时程序时应注意使循环的重数尽量少,以减少寄存器的使用量,同时也使程序更加简单,出错的可能性随之降低。其次尽量不使用寄存器R0、R1,因为它们经常用在间接寻址及需要传送外部数据的地方。延时程序具体的编写步骤如下:

- 1) 明确晶振频率,定出延时时间所需的最小循环数;
- 2) 将内层循环设为最大值,即内层循环的寄存器值取0;
- 3) 根据公式确定出不大于所要的延时时间的最外层循环的寄存器的取值;
- 4) 用所要得的延时时间减去上述步骤得出的延时时间,差值作为第

二个所要的延时时间,重复步骤1)的做法,直至差值缩小至 $10\mu\text{s}$ 以内;

5)最后用添加NOP指令的方式来得到精确的延时时间。

例如设计一个延时时间为 $1\text{s}$ 的延时程序。首先 $1\text{s}$ 所属的最小循环为三重循环,将内部寄存器的值设置为0,其次根据公式算出不大于 $1\text{s}$ 的外层循环的最大取值为7,此时得到的延时时间为 $922904\mu\text{s}$ ,与 $1\text{s}$ 相减余数为 $77096\mu\text{s}$ 。而 $77096\mu\text{s}$ 属于的最小循环为二重,当外重循环取149时,得到的延时时间为 $76738\mu\text{s}$ ,与 $77096\mu\text{s}$ 相减余数为 $358\mu\text{s}$ 。 $358\mu\text{s}$ 通过一次循环程序配合NOP指令就可以得到。注意到三重循环与二重循环计算公式中占两个机器周期的RET指令已不存在,要用NOP指令补充。因此可得到延时时间为 $1\text{s}$ 的精确程序。

```
DEL:  MOV R5, #7
DEL1: MOV R6, #0
DEL2: MOV R7, #0
DEL3: DJNZ R7, DEL3
      DJNZ R6, DEL2
      DJNZ R5, DEL1
MOV R6, #149
DEL4: MOV R7, #0
DEL5: DJNZ R7, DEL5
      DJNZ R6, DEL4
      MOV R7, #177
DEL6: DJNZ R7, DEL6
NOP
NOP
NOP
NOP
NOP
RET
```

利用keil c的测试工具,在延时程序的开始与结尾设置两个断点,两个断点处的时间差值为 $1.00000200\text{s}$ 。以上程序中 $2\mu\text{s}$ 的误差是由于没有考虑调用延时程序的指令所占用的机器周期,如ACALL、LCALL指令,将这些指令考虑进去后减去两个NOP指令,这样就得到了精确的 $1\text{s}$ 延时程序。这种方法对于用C语言来设计的延时程序同样适用,下面是利用此法并考虑了调用指令的C语言延时程序。

```
delay(void)
{ unsigned char i, j, k;
```

```
for(k=7; k>=1; k--)
for(j=255; j>=1; j--)
for(i=255; i>=1; i--)
{ }
for(j=164; j>=1; j--)
for(i=255; i>=1; i--)
{ }
for(i=66; i>=1; i--)
{ }
_NOP_();
_NOP_();
_NOP_();
}
```

## 5 总结

本文通过对软件延时程序计算方法的公式化推导,总结了一种精确设计软件延时的数学算法,为分析和设计程序提供了一种新的方法和技巧。无论是对设计精度要求较高的程序还是对设计精度要求一般的程序都应当有使程序更加精确的好习惯,这样不仅可以提高程序的精确度,还可减少程序出错的几率。希望本文在实际问题解决的过程中对大家有所帮助。

## 参考文献:

- [1]余锡存、曹国华,单片机原理及接口技术(第二版)[M].西安:西安电子科技大学出版社,2007.
- [2]沈旻,延时程序延时时间的计算[J].电工技术与自动化,2005,(5):154-157.
- [3]王超峰,单片机延时程序时间长度的记忆方法[J].单片机与嵌入式系统应用,2002,(7):77-78.
- [4]彭志刚,51系列单片机延时程序运行时间的计算[J].仪器仪表用户,2008,(5):102-103.
- [5]文桦,彭彦华.单片机应用系统定时时间的误差分析[J].天中学刊,2006,(5):38-39.
- [6]admin.C语言编程细节总结[EB].IT达人网,2009-7-15.

## 作者简介:

韩薇薇(1985-),女,南京理工大学在职硕士研究生,扬州高等职业技术学院电子工程系,助教,主要研究方向:应用电子技术。

(上接第86页)

Vs2005提供的部署工具有以下选项:

1)允许更新此预编译站点。选中这一项时Asp.net的页面文件(.aspx文件)的内容按原样部署,页面文件对应的后置代码文件.aspx.cs被编译成为.dll,这些.dll统一放在一个名为bin的文件夹中。可以看到这些文件名是一些无规律的随机文件名(除非选择上面的使用固定命名和单页程序集)。

不选中这一项,编译出来的aspx中没有界面信息,只有一行静态文本,就是不允许部署后修改页面。如果是想此编译方式具有最大的安全性,应去掉“允许更新此预编译站点”,这样代码文件(code,即cs文件)和内容文件(content file,即aspx)都会预编译,以这种方式部署网站则所有的aspx都是受保护不可见的,但这种方式不允许部署后修改单个页面,修改某个页面后只能重新部署。

2)使用固定命名和单页程序集会按照网站项目中所有的类编译出一一对应的很多名字固定的dll。

3)对预编辑程序集启用强命名。在多数情况下,完全预编译方式正是所需要的方式,但是有时候因为内容文件变化不大,你可能希望在网站部署后,不用每次把所有的代码与内容文件全部编译,也许内容文件就不用再次编译,只需编译代码文件即可,这种情况下,就选中“允许更新此预编译站点”,这种方式称为“只预编译代码文件(pre-compilation of code only)”方式.此方式与“完全预编译方式”相比较,只有一点区别,即内容文件仍是原始版本,而不是存根(stub)版本,其它效果

相同.在内容文件部署后也可以对其进行编辑,其变动在以后的请求到来时起作用,对于访问此站的用户来说是透明的。

## 4 总结

本文首先分析了Asp.net网站的特点,根据实际使用中存在的问题,分别提出利用XCOPY部署、使用Visual studio的复制网站功能部署、使用Visual studio的部署网站功能部署,在使用时可根据实际情况来选用不同的操作方式。

## 参考文献:

- [1]Jeffrey Richter,Microsoft .NET框架程序设计[M].清华大学出版,2003年11月.
- [2]内格尔(Nagel, C.),C#高级编程(第6版)[M].清华大学出版社,2008年10月.
- [3]Dino Esposito,ASP.NET 2.0技术内幕[M].清华大学出版社,2006年8月.
- [4]CHRIS HART,JOHN KAUFFMAN,ASP.NET 2.0入门经典(第4版)[M].清华大学出版社,2006年8月.

## 作者简介:

胡小凤(1975-),女,汉族,湖南省冷水江人,本科,讲师,主要研究方向:软件工程。