**第七章 51单片机的定时/计数器**

**1. 89C51定时器有哪几种工作方式？有何区别，请用表格阐述？**

答：有四种工作模式：方式0，方式1，方式2，方式3

|  |  |  |  |
| --- | --- | --- | --- |
| M1 M0 | 工作方式 | 功能 | 最大计数值 |
| 0 0 | 方式0 | 13位定时/计数器，由THX（X=0，1）的8位和TLX的低5位构成。 | M＝213＝8192 |
| 0 1 | 方式1 | 16位定时/计数器，由THX和TLX构成。 | M＝216＝65536 |
| 1 0 | 方式2 | 可自动重装初值的8位计数器，TLX作计数器，THX保存计数初值，一旦计数器计满溢出，初值自动装入，继续计数重复不止。 | M＝28＝256 |
| 1 1 | 方式3 | 仅适用于T0，分为两个8位计数器，T1停止计数。 | M＝28＝256 |

**2. 选择定时器0工作方式1时，若时钟频率12MHz，定时50ms，TH0和TL0怎样装初值？写出计算过程。**

答：假设时钟频率为12MHz（就是单片机最小系统18脚和19脚之间的晶振大小），那么时钟周期（振荡周期）就是（1/12000000）Hz，由于我们用的单片机为12T单片机，所以一个机器周期等于12个时钟周期，也就是1μs，无论TH0和TL0初值是多少，计满都是2^16-1=65535，再加一个数就会溢出，TH0和TL0都变为0（循环），随即向CPU发出中断申请。如果从0开始计数那么只能加65536个数溢出进入中断，也就是最多65536μs即65.536ms，为了定时1s，我们可以让基本的定时时间取50000μs即50ms，取20次即为1s，那么定时50000μs如何设定初值那？TH0和TL0是定时器的高八位和第八位，低8位都是1时对应的数是2^8-1=255，再加1就会向高八位进1位，所以存放初值时TH0和TL0总数除以256的整数应放在TH0里，用“/”表示除取整，余数应放在TL0里，用“%”表示除取余。要计50000个数，TH0和TL0应装入总的数值为65536-50000=15536，TH0=15536/256=60，TL0=15536%256=176。实际上经常写法为：TH0=(65536-50000)/256，TL0=（65536-50000）%256，其实要计多少个数溢出产生中断只需要将50000改为要写的数即可。

**3.如何书写定时器0中断服务函数？**

Void T0\_time() interrupt 1

{.................}

T0\_time 可以随意改动

1. **如果外部晶振11.0592MHz，使用定时器0工作方式1，定时50ms，在程序初始阶段请对有关的寄存器进行设置？**

TMOD=0x01; //设置定时器0为工作1

TH0=(65536-45872)/256; //装初值11.0592MHZ晶振

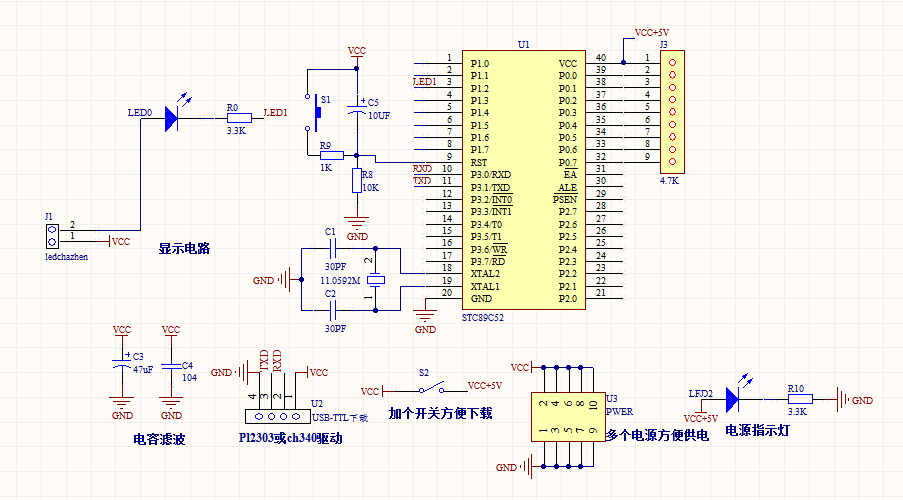
TL0=(65536-45872)%256; //定时50毫秒数为45872

EA=1; //开总中断

ET0=1; //开定时器0中断

TR0=1; //启动定时器0

**5.编写程序实现小灯1s一亮的功能？**

****

#include<reg52.h> //52系列单片机头文件

#define uchar unsigned char //宏定义

#define uint unsigned int

sbit LED1=P1^2; //声明单片机P1口第三位

uchar n;

void delayms(uint xms);

void main() //主函数

{

TMOD=0x01; //设置定时器0为工作1

TH0=(65536-45872)/256; //装初值11.0592MHZ晶振

TL0=(65536-45872)%256; //定时50毫秒数为45872

EA=1; //开总中断

ET0=1; //开定时器0中断

TR0=1; //启动定时器0

while(1); //程序停止在这里等待中断

}

void T0\_time()interrupt 1

{

TH0=(65536-45872)/256; //重装初值

TL0=(65536-45872)%256;

n++; //n每加一次判断一次是否到20次

if(n==20) //如果n到20次，说明1秒时间到

{

n=0; //然后把n清零重新再计数

LED1=0; //让发光二极管状态取反

delayms（10）;

LED1=1;

}

}

void delayms(uint xms)

{

uint i,j;

for(i=xms;i>0;i--)

for(j=110;j>0;j--);

}