|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 51单片机中断级别   |  |  |  | | --- | --- | --- | | 中断源 | 默认中断级别 | 序号（C语言用） | | INT0---外部中断0 | 最高 | 0 | | T0---定时器/计数器0中断 | 第2 | 1 | | INT1---外部中断1 | 第3 | 2 | | T1----定时器/计数器1中断 | 第4 | 3 | | TX/RX---串行口中断 | 第5 | 4 | | T2---定时器/计数器2中断 | 最低 | 5 |   ***中断允许寄存器IE***   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 位序号 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | 符号位 | EA | ------- | ET2 | ES | ET1 | EX1 | ET0 | EX0 |   **EA---全局中允许位。** EA=1，打开全局中断控制，在此条件下，由各个中断控制位确定相应中断的打开或关闭。 EA=0，关闭全部中断。 -------，无效位。 **ET2---定时器/计数器2中断允许位。 EA总中断开关，置1为开；** ET2=1，打开T2中断。 EX0为外部中断0（INT0）开关，…… ET2=0，关闭T2中断。 ET0为定时器/计数器0（T0)开关，……  **ES---串行口中断允许位。**EX1为外部中断1（INT1）开关，…… ES=1，打开串行口中断。 ET1为定时器/计数器1（T1)开关，…… ES=0，关闭串行口中断。 ES为串行口（TX/RX）中断开关，…… **ET1---定时器/计数器1中断允许位。**ET2为定时器/计数器2（T2)开关，…… ET1=1，打开T1中断。 ET1=0，关闭T1中断。 **EX1---外部中断1中断允许位。** EX1=1，打开外部中断1中断。 EX1=0，关闭外部中断1中断。 **ET0---定时器/计数器0中断允许位。** ET0=1，打开T0中断。 ET0=0，关闭T0中断。 **EX0---外部中断0中断允许位。** EX0=1，打开外部中断0中断。 EX0=0，关闭外部中断0中断。 ***中断优先级寄存器IP***   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 位序号 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | 位地址 | --- | --- | --- | PS | PT1 | PX1 | PT0 | PX0 |   -------，无效位。 **PS---串行口中断优先级控制位。** PS=1，串行口中断定义为高优先级中断。 PS=0，串行口中断定义为低优先级中断。 **PT1---定时器/计数器1中断优先级控制位。** PT1=1，定时器/计数器1中断定义为高优先级中断。 PT1=0，定时器/计数器1中断定义为低优先级中断。 **PX1---外部中断1中断优先级控制位。** PX1=1，外部中断1中断定义为高优先级中断。 PX1=0，外部中断1中断定义为低优先级中断。 **PT0---定时器/计数器0中断优先级控制位。** PT0=1，定时器/计数器0中断定义为高优先级中断。 PT0=0，定时器/计数器0中断定义为低优先级中断。 **PX0---外部中断0中断优先级控制位。** PX0=1，外部中断0中断定义为高优先级中断。 PX0=0，外部中断0中断定义为低优先级中断。 ***定时器/计数器工作模式寄存器TMOD***   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 位序号 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | 位符号 | GATE | C/T | M1 | M0 | GATE | C/T | M1 | M0 |   |-----------------定时器1------------------------|--------------------定时器0----------------------| **GATE---门控制位。** GATE=0，定时器/计数器启动与停止仅受TCON寄存器中TRX(X=0,1)来控制。 GATE=1，定时器计数器启动与停止由TCON寄存器中TRX(X=0,1)和外部中断引脚（INT0或INT1）上的电平状态来共同控制。 **C/T---定时器和计数器模式选择位。** C/T=1，为计数器模式；C/T=0，为定时器模式。 **M1M0---工作模式选择位。**   |  |  |  | | --- | --- | --- | | M1 | M0 | 工作模式 | | 0 | 0 | 方式0，为13位定时器/计数器 | | 0 | 1 | 方式1，为16位定时器/计数器 | | 1 | 0 | 方式2，8位初值自动重装的8位定时器/计数器 | | 1 | 1 | 方式3，仅适用于T0，分成两个8位计数器，T1停止工作 |   **定时器/控制器控制寄存器TCON**   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **位序号** | **DB7** | **DB6** | **DB5** | **DB4** | **DB3** | **DB2** | **DB1** | **DB0** | | **符号位** | **TF1** | **TR1** | **TF0** | **TR0** | **IE1** | **IT1** | **IE0** | **IT0** |   **TF1---定时器1溢出标志位。** 当定时器1记满溢出时，由硬件使TF1置1，并且申请中断。进入中断服务程序后，由硬件自动清0。需要注意的是，如果使用定时器中断，那么该位完全不用人为去操作，但是如果使用软件查询方式的话，当查询到该位置1后，就需要用软件清0。 **TR1---定时器1运行控制位。** 由软件清0关闭定时器1。当GATE=1，且INIT为高电平时，TR1置1启动定时器1；当GATE=0时，TR1置1启动定时器1。 **TF0---定时器0溢出标志，其功能及其操作方法同TF1。** **TR0---定时器0运行控制位，其功能及操作方法同TR1。** **IE1---外部中断1请求标志。** 当IT1=0时，位电平触发方式，每个机器周期的S5P2采样INT1引脚，若NIT1脚为定电平，则置1，否则IE1清0。 当IT1=1时，INT1为跳变沿触发方式，当第一个及其机器周期采样到INIT1为低电平时，则IE1置1。IE1=1，表示外部中断1正向CPU中断申请。当CPU响应中断，转向中断服务程序时，该位由硬件清0。 **IT1外部中断1触发方式选择位。** IT1=0，为电平触发方式，引脚INT1上低电平有效。 IT1=1，为跳变沿触发方式，引脚INT1上的电平从高到低的负跳变有效。 **IE0---外部中断0请求标志，其功能及操作方法同IE1。** **IT0---外部中断0触发方式选择位，其功能及操作方法同IT1。** 从上面的知识点可知，每个定时器都有4种工作模式，可通过设置TMOD寄存器中的M1M0位来进行工作方式选择。 方式1的计数位数是16位，对T0来说，由TL0寄存器作为低8、TH0寄存器作为高8位，组成了16位加1计数器。 关于如何确定定时器T0的初值问题。定时器一但启动，它便在原来的数值上开始加1计数，若在程序开始时，我们没有设置TH0和TL0，它们的默认值都是0，假设时钟频率为12MHz，12个时钟周期为一个机器周期，那么此时机器周期为1us，记满TH0和TL0就需要216 -1个数，再来一个脉冲计数器溢出，随即向CPU申请中断。因此溢出一次共需65536us,约等于65.6ms，如果我们要定时50ms的话，那么就需要先给TH0和TL0装一个初值，在这个初值的基础上记50000个数后，定时器溢出，此时刚好就是50ms中断一次，当需要定时1s时，我们写程序时当产生20次50ms的定时器中断后便认为是1s，这样便可精确控制定时时间啦。要计50000个数时，TH0和TL0中应该装入的总数是65536-50000=15536.，把15536对256求模：15536/256=60装入TH0中，把15536对256求余：15536/256=176装入TL0中。 以上就是定时器初值的计算法，总结后得出如下结论：当用定时器的方式1时，设机器周期为TCY，定时器产生一次中断的时间为t，那么需要计数的个数为N=t/TCY ，装入THX和TLX中的数分别为： **THX=(65536-N)/256 , TLX=(65536-N)%256 <x为0或1>** **中断服务程序的写法** void 函数名()interrupt 中断号 using 工作组 { 中断服务程序内容 } 在写单片机的定时器程序时，在程序开始处需要对定时器及中断寄存器做初始化设置，通常定时器初始化过程如下： （1）对TMOD赋值，以确定T0和 T1的工作方式。 （2）计算初值，并将初值写入TH0、TL0或TH1、TL1。 （3）中断方式时，则对IE赋值，开放中断。 （4）使TR0和TR1置位，启动定时器/计数器定时或计数。 **例：利用定时器0工作方式1，实现一个发光管以1s亮灭闪烁。** **程序代码如下：** **#include<reg52.h>** **#define uchar unsigned char** **#define uint unsigned int** **sbit led1=P1^0;** **uchar num;** **void main()** **{** **TMOD=0x01; //设置定时器0位工作模式1（M1,M0位0，1）** **TH0=(65536-45872)/256; //装初值11.0592M晶振定时50ms数为45872** **TL0=(65536-45872)%256;** **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)** **{** **if(num==20) //如果到了20次，说明1秒时间** **{** **led1=~led1; //让发光管状态取反** **num=0;** **}** **}** **}** **void T0\_time()interrupt 1** **{** **TH0=(65536-45872)/256; //重新装载初值** **TL0=(65536-45872)%256;** **num++;** **}** |

51单片机1个机器周期=12个时钟周期，频率为12MHZ，则一个机器周期为1US，具体到定时器程序就是，假如你想定1MS，那么单片机每次加一个一，就要过1US，那么1MS就要加1000次，所以用65535-1000=64535；再把64535换成16进制为FC17，把FC付给TH0,17给TLO,即可定时1MS，因为65535他就溢出进入中断。