

51单片机-通过定时器控制LED的闪烁和流水灯

定时器

是单片机内部的系统，每隔一段时间进行一次中断，中断后会跳到中断函数执行内部操作，使用前应先进行初始化

```
1 void Timer0_Init(){
2     //TMOD = 0x01;           //定时器模式为0000 0001，可用下两行代替
3     TMOD &= 0xF0;           //把TMOD的低四位清零，高四位保持不变
4     TMOD |= 0x01;           //把TMOD的最低位置1，高四位保持不变
5     TFO = 0;                 //中断溢出标志位
6     TR0 = 1;                 //定时器是否开启
7     /*该计数器只能从0计到65535个数，每隔1us计数加1，总共定时65535us*/
8     TH0 = 64535/256;         //64535的高八位，使计数器计1ms
9     TL0 = 64535%256;         //64535的低八位，使计数器计1ms
10    ET0 = 1;
11    EA = 1;
12    PT0 = 0;
13 }
```

中断函数

当计数器计完一个周期后，会跳到该函数

```
1 void Timer0_Routine() interrupt 1{
2     static unsigned int T0Count;
3     TH0 = 64535/256;
4     TL0 = 64535%256;
5     T0Count++;
6     if(T0Count>=100){ //设定时间为100ms
7         T0Count = 0;
8         P2 = ~(~P2 << 1); //流水灯操作
9         if(P2 == 0xFF) P2 = 0xFE; //当P2=0xFF左移都为0xFF，因此需要重置P2为0xFE
10    }
11 }
```

完整代码

```
1 #include "REG51.h"
2
3 void Timer0_Init(){
4     //TMOD = 0x01;           //定时器模式为0000 0001，可用下两行代替
5     TMOD &= 0xF0;           //把TMOD的低四位清零，高四位保持不变
6     TMOD |= 0x01;           //把TMOD的最低位置1，高四位保持不变
7     TFO = 0;                 //中断溢出标志位
8     TR0 = 1;                 //定时器是否开启
9     /*该计数器只能从0计到65535个数，每隔1us计数加1，总共定时65535us*/
```

```
10     TH0 = 64535/256;    //64535的高八位，使计数器计1ms
11     TL0 = 64535%256;    //64535的低八位，使计数器计1ms
12     ET0 = 1;
13     EA = 1;
14     PT0 = 0;
15 }
16 int main(){
17     P2 = 0xFE;
18     Timer0_Init();
19     while(1){
20
21     }
22 }
23
24 void Timer0_Routine() interrupt 1{
25     static unsigned int T0Count;
26     TH0 = 64535/256;
27     TL0 = 64535%256;
28     T0Count++;
29     if(T0Count>=100){    //设定时间为100ms
30         T0Count = 0;
31         P2 = ~(~P2 << 1); //流水灯操作
32         if(P2 == 0xFF) P2 = 0xFE; //当P2=0xFF左移都为0xFF，因此需要重置P2为0xFE
33     }
34 }
```