**51单片机 主循环和中断共享全局变量的冲突**

现象：工作中遇到一个大坑，STC11F32的运行灯闪的不均匀，大多数是500ms低电平和500ms高电平，但偶尔会有40ms左右的高、低电平出现，对，就这么简单的一个程序，奇哉怪哉！

//定义全局变量：ms计数器

unsigned int cnt\_ms;

//定义全局变量：ms计数器备份

unsigned int cnt\_ms\_bk;

//定义LED对应的IO口

sbit LED = P3 ^ 3;

//主循环读访问

int main(void)

{

......

if ((cnt\_ms - cnt\_ms\_bk) > 500)

{

led = ~led;

cnt\_ms\_bk = cnt\_ms;

}

......

}

void timer0\_interrupt(void) interrupt 1

{

......

TL0 = 0x0cd; //装在1ms定时初值

TH0 = 0x0f8;

count\_1ms++;

......

}

解决方法：主循环程序修改如下，用示波器观察led的闪烁，结果是正常的。

int main(void)

{

......

EA = 0; //关中断

if ((cnt\_ms - cnt\_ms\_bk) > 500)

{

led = ~led;

cnt\_ms\_bk = cnt\_ms;

}

EA = 1; //开中断

......

}

原因分析：反汇编如下，为分析方便采用16位示例。

......

101: if ((cnt\_ms - cnt\_ms\_bk) >= 500)

C : 0x0F18 900036 MOV DPTR, #msLedCtr(0x0036)

C : 0x0F1B E0 MOVX A, @DPTR

C : 0x0F1C F8 MOV R0, A

C : 0x0F1D A3 INC DPTR

C : 0x0F1E E0 MOVX A, @DPTR

C : 0x0F1F F9 MOV R1, A

C : 0x0F20 A3 INC DPTR

C : 0x0F21 E0 MOVX A, @DPTR

C : 0x0F22 FA MOV R2, A

C : 0x0F23 A3 INC DPTR

C : 0x0F24 E0 MOVX A, @DPTR

C : 0x0F25 FB MOV R3, A

C : 0x0F26 900031 MOV DPTR, #count\_1ms(0x0031)

C : 0x0F29 E0 MOVX A, @DPTR

C : 0x0F2A FC MOV R4, A

C : 0x0F2B A3 INC DPTR

C : 0x0F2C E0 MOVX A, @DPTR //在此处被中断

C : 0x0F2D FD MOV R5, A

C : 0x0F2E A3 INC DPTR

C : 0x0F2F E0 MOVX A, @DPTR

C : 0x0F30 FE MOV R6, A

C : 0x0F31 A3 INC DPTR

C : 0x0F32 E0 MOVX A, @DPTR

C : 0x0F33 C3 CLR C

C : 0x0F34 9B SUBB A, R3

C : 0x0F35 FF MOV R7, A

C : 0x0F36 EE MOV A, R6

C : 0x0F37 9A SUBB A, R2

C : 0x0F38 FE MOV R6, A

C : 0x0F39 ED MOV A, R5

C : 0x0F3A 99 SUBB A, R1

C : 0x0F3B FD MOV R5, A

C : 0x0F3C EC MOV A, R4

C : 0x0F3D 98 SUBB A, R0

C : 0x0F3E FC MOV R4, A

......

......

282 : cnt\_ms++;

C : 0x09D1 900031 MOV DPTR, #cnt\_ms(0x0031)

C : 0x09D4 E0 MOVX A, @DPTR

C : 0x09D5 FC MOV R4, A

C : 0x09D6 A3 INC DPTR

C : 0x09D7 E0 MOVX A, @DPTR

C : 0x09D8 FD MOV R5, A

C : 0x09D9 A3 INC DPTR

C : 0x09DA E0 MOVX A, @DPTR

C : 0x09DB FE MOV R6, A

C : 0x09DC A3 INC DPTR

C : 0x09DD E0 MOVX A, @DPTR

C : 0x09DE 2401 ADD A, #0x01

C :0x09E0 FF MOV R7, A

C : 0x09E1 E4 CLR A

C : 0x09E2 3E ADDC A, R6

C : 0x09E3 FE MOV R6, A

C : 0x09E4 E4 CLR A

C : 0x09E5 3D ADDC A, R5

C : 0x09E6 FD MOV R5, A

C : 0x09E7 E4 CLR A

C : 0x09E8 3C ADDC A, R4

C : 0x09E9 FC MOV R4, A

C : 0x09EA 900031 MOV DPTR, #count\_1ms(0x0031)

C : 0x09ED 1218B7 LCALL C ? LSTXDATA(C : 18B7)

......

假设在上述位置被中断前cnt\_ms的值为0x00FF，cnt\_ms\_bk的值为0x0000时：R4装载的值为0xFF，本打算给R5装载值0x00，结果被中断打断，中断后cnt\_ms++，后其值变为0x0100，中断执行完返回后继续主循环中装载R5，结果给R5装载值0x01，各位现在主循环中使用的cnt\_ms为0x01FF了，这样就会造成脉冲的变窄。

我起先老纠结中断时会压栈，会将变量保存。实际中断时保存的是ACC, B, DPTR等寄存器，但是全局变量cnt\_ms被保存到了xdata区，DPTR中断前后不变，但其指向的内容（xdata区）已被修改。

---------------------

作者：breeze991073

来源：CSDN

原文：https://blog.csdn.net/breeze991073/article/details/80844724

版权声明：本文为博主原创文章，转载请附上博文链接！