

教你学单片机 (十一)

■周兴华

MCS-51 中断系统及实验

什么是“中断”？顾名思义中断就是中断某一工作过程去处理一些与本工作过程无关或间接相关或临时发生的事件，处理完后，则继续原工作过程。比如：你在看书，电话响了，你在书上做个记号后去接电话，接完后在原记号处继续往下看。如有多个中断发生，依优先法则，中断还具有嵌套特性。又比如：看书时，电话响了，你在书上做个记号后去接电话，你拿起电话和对方通话，这时门铃响了，你让打电话的对方稍等一下，你去开门，并在门旁与来访者交谈，谈话结束，关好门，回到电话机旁，拿起电话，继续通话，通话完毕，挂上电话，从作记号的地方继续往下看。由于一个人不可能同时完成多项任务，因此只好采用中断方法，一件一件地做。

类似的情况在单片机中也同样存在，通常单片机中只有一个 CPU，但却要应付诸如运行程序、数据输入输出以及特殊情况处理等多项任务，为此也只能采用停下一个工作去处理另一个工作的中断方法。

在单片机中，“中断”是一个很重要的概念。中断技术的进步使单片机的发展和应用大大地推进了一步。所以，中断功能的强弱已成为衡量单片机功能完善与否的重要指标。

单片机采用中断技术后，大大提高了它的工作效率和处理问题的灵活性，主要表现在三方面：

1. 解决了快速 CPU 和慢速外设之间的矛盾，可使 CPU、外设并行工作（宏观上看）。

2. 可及时处理控制系统中许多随机的参数和信息。

3. 具备了处理故障的能力，提高了单片机系统自身的可靠性。

中断处理程序类似于程序设计中的调用子程序，但它们又有区别，

主要是：

中断产生是随机的，它既保护断点，又保护现场，主要为外设服务和为处理各种事件服务。保护断点是由硬件自动完成的，保护现场须在中断处理程序中用相应的指令完成。

调用子程序是程序中事先安排好的，它只保护断点，主要为程序服务（与外设无关）。

中断的种类

中断的应用是很广泛的，因此能引起中断的原因也是多种多样的，也就是说，要求共享 CPU 的任务很多，因此有必要对中断加以分类，通常把中断分为外中断和内中断两大类。

1. 外中断

外中断是由 CPU 以外的原因引起的，通过硬件电路发出中断请求，因此把这类中断称之为硬件中断。外中断主要用于实现外设的数据传送、实时处理以及人机联系等。

属于外中断的中断源主要有：

- 1) 输入输出设备及外存储设备。
- 2) 实时时钟或计数电路。
- 3) 电源故障等。

2. 内中断

内中断是指由 CPU 内部原因引起的中断，由于这类中断发生在 CPU 的内部，因此称之为内中断。内中断包括陷阱中断和软件中断两种。

1) 陷阱中断是指由 CPU 内部事件引起的中断，例如程序执行中的故障，或 CPU 内部的硬件故障等。

2) 而软件中断则是由一些专用的软件中断指令或系统调用指令引起，通过软件中断可以引入程序断点，便于进行程序调试和故障检测。

MCS-51 单片机的中断系统

一、中断源及控制

MCS-51 单片机共有三类 5 个中断源，二个优先级，中断处理程序可实现两级嵌套，有较强的中断处理能力。5 个中断源中，其中 2 个为外部中断

请求 $\overline{INT0}$ 和 $\overline{INT1}$ (由 P3.2 和 P3.3 输入)，2 个为片内定时/计数器 T0 和 T1 的溢出中断请求 TFO 和 TF1，另一个为片内串行口中断请求 TI 或 RI，这些中断请求信号分别锁存在特殊功能寄存器 TCON 和 SCON 中。

TCON：定时/计数器控制寄存器，字节地址 88H。其锁存中断请求标志的格式如下表所示。

其中与中断有关的控制位有六

TCON	TF1	TR1	TFO	TRO	IE1	IT1	IE0	IE0
位地址	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H

位：ITO(IT1)、IE0(IE1)、TFO(TF1)。

ITO：外部中断 0 请求方式控制位。ITO=0，为电平触发方式， $\overline{INT0}$ 低电平有效；ITO=1， $\overline{INT0}$ 为边沿触发方式， $\overline{INT0}$ 输入脚上电平由高到低的负跳变有效。ITO 可由软件置“1”或清“0”。

IE0：外部中断 0 请求标志位。CPU 采样到 $\overline{INT0}$ 端出现有效中断请求时，该位由硬件置位；当 CPU 响应中断，转向中断服务程序时由硬件清零 IE0。

IT1：外部中断 1 请求方式控制位，和 ITO 类似。

IE1：外部中断 1 请求标志位，和 IE0 相同。

TFO：片内定时/计数器 T0 溢出中断申请标志，在启动 T0 计数后，定时/计数器 T0 从初值开始加 1 计数，当最高位产生溢出时，由硬件置位 TFO，向 CPU 申请中断，CPU 响应 TFO 中断时清除该标志位，TFO 也可用软件查询后清除。

TF1：片内的定时/计数器 T1 的溢出中断申请标志，功能和 TFO 类同。当 MCS-51 复位后，TCON 被清“0”。

SCON：串行口控制寄存器，字节地址为 98H。SCON 的低二位锁存串行口的接收中断和发送中断标志，其格式如下表所示。

TI：串行口的发送中断标志。当

SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

发送完一帧 8 位数据后，由硬件置位 TI。由于 CPU 响应发送器中断请求后，转向执行中断服务程序时并不清除 TI，TI 必须由用户在中断服务程序中清除。

RI：串行口接收中断标志。当接

收完一帧 8 位数据时置位 RI。同样 RI 必须由用户的中断服务程序清零。

MCS-51 复位以后,SCON 也被清零。

对于每个中断源,其开放与禁止由专用寄存器 IE 中的某一位控制,其中断次序可由专用寄存器 IP 中相应位是置 1 还是清 0 决定其为高优先级还是低优先级,这在硬件上有相应的优先级触发器予以保证。IE 和 IP 寄存器格式分述如下:

中断允许寄存器(IE)

IE	EA	/	ET2	ES	ET1	EX1	ET0	EX0
位地址	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H

与中断有关的控制位共六位: EA、ES、ET1(ET0)、EX1(EX0)。

EA: 中断总允许控制位。EA=0, 禁止总中断。EA=1, 开放总中断, 随后每个中断源分别由各自的允许位的置位或清除确定开放或禁止。

ES: 串行中断允许控制位。ES=0, 禁止串行中断。ES=1, 允许串行中断。
ET1: 定时/计数器 T1 中断允许控制位。ET1=0, 禁止 T1 中断。ET1=1, 允许 T1 中断。

EX1: 外部中断源 1 中断允许控制位。EX1=0, 禁止外部中断 1。EX1=1, 允许外部中断 1。

ET0: 定时/计数器 T0 中断允许控制位。ET0=0, 禁止 T0 中断。ET0=1, 允许 T0 中断。

EX0: 外部中断源 0 中断允许控制位。EX0=0, 禁止外部中断 0。EX0=1, 允许外部中断 0。

中断优先级寄存器(IP)

IP	/	/	/	PS	PT1	PX1	PT0	PX0
位地址	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H

其中:

PS: 串行中断优先级设定位。PS=1, 则编程为高优先级。

PT1: 定时器 T1 中断优先级设定位。PT1=1, 则编程为高优先级,

PX1: 外中断 1 优先级设定位。PX1=1, 则编程为高优先级。

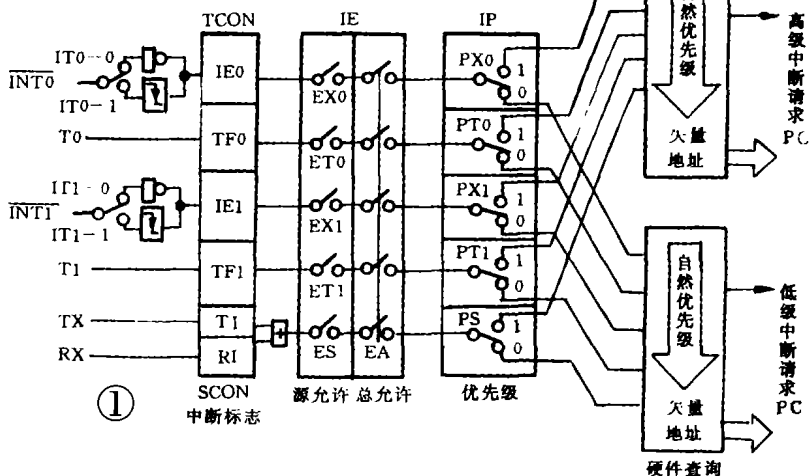
PT0: 定时器 T0 中断优先级设定位。PT0=1, 则编程为高优先级。

PX0: 外中断 0 优先级设定位。PX0=1, 则编程为高优先级。

需要说明的是,单片机复位之后 IE 和 IP 均被清 0。用户可按需要置位或清除 IE 的相应位,来允许或禁止各

中断源的中断申请。

为使某中断源允许中断,必须同时使 EA=1,首先使 CPU 开放中断,所以 EA 相当于中断允许的“总开关”。至于中断优先级寄存器 IP,其复位清 0 将会把各个中断源置为低优先级中断,同样,用户也可对相应位置“1”或清“0”,来改变各中断源的中断优先级。整个中断系统结构如图 1 所示。



MCS-51 单片机对中断优先级的处理原则是:

1. 不同级的中断源同时申请中断时,先处理高优先级后处理低优先级。
2. 处理低级中断又收到高级中断请求时,停止处理低优先级转而处理高优先级。
3. 正在处理高级中断却收到低级中断请求时,不理睬低优先级。
4. 同一级的中断源同时申请中断时,通过内部查询按自然优先级顺序确定应响应哪个中

断申请。对于同一优先级,单片机对其中断次序安排如下:

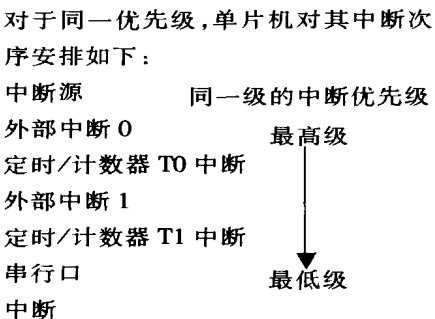


图 2 和 3 为单片机响应中断的流程图及中断嵌套流程图。

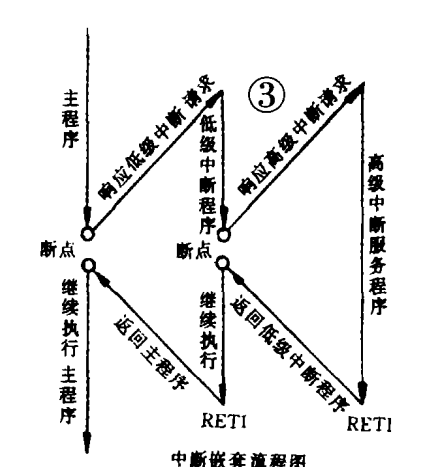
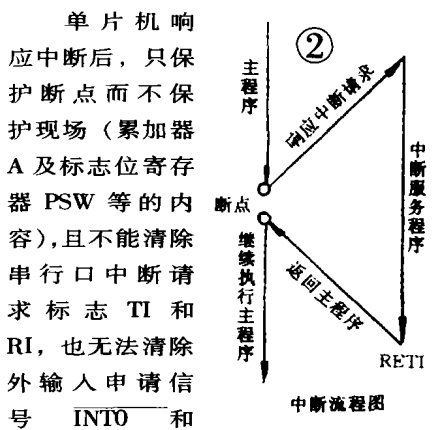
二、中断响应

单片机响应中断的基本条件是:

中断源有请求,中断允许寄存器 IE 相应位置“1”,总中断开放(EA=1)。

单片机中断响应过程:单片机一旦响应中断,首先置位相应的优先级有效触发器,然后执行一个硬件子程序调用,把断点地址压入堆栈,再把与各中断源对应的中断服务程序首地址送程序计数器 PC,同时清除中断请求标志(TI 和 RI 除外),从而控制的

程序转移到中断服务程序。以上过程均由中断系统自动完成。



INTI, 因而进入中断服务子程序后, 如用到上述寄存器就会破坏它原来存在的内容, 一旦中断返回, 将造成主程序的混乱。所以在进入中断服务子程序后, 一般都要保护现场, 然后再执行中断服务程序。在返回主程序前再恢复现场。所有这些应在用户编制中断处理程序时予以考虑。

各中断源所对应的中断服务程序入口地址如下:

中断源	入口地址
外部中断 INT0	0003H
定时器/计数器 T0	000BH
外部中断 INT1	0013H
定时器/计数器 T1	001BH
串行口中断	0023H

CPU 从此地址开始执行中断服务程序直至遇到一条 RETI 指令为止。RETI 指令表示中断服务程序的结束, CPU 执行该指令, 一方面清除中断响应时所置位的优先级有效触发器, 一方面由栈顶弹出断点地址送程序计数器 PC, 从而返回主程序。若用户在中断服务程序开始安排了保护现场指令(相应寄存器内容入栈), 则在 RETI 指令前应有恢复现场(相应寄存器内容出栈)指令。

下面做实验, 认识 MCS-51 单片机中断系统的作用。

在 S1 板上做一个实验, 使用定时器 T1 以方式 0 使单片机产生周期为 1000 μ S 等宽方波脉冲(1000Hz 音频), 在 P1.7 输出驱动蜂鸣器发音。这个实验同讲座(九)的第一个实验类似, 但讲座(九)的实验采用查询方式, 这里采用中断方式完成。

S1 板使用 11.0592MHz 晶振, 可近似认为其为 12MHz。这样一个机器周期为 1 μ S。欲产生 1000 μ S 周期方波脉冲, 只需在 P1.7 以 500 μ S 时间交替输出高低电平即可。

1. T1 为方式 0, 则 M1M0=00H。使用定时功能, C/T=0。GATE=0。T0 不用, 其有关位设为 0。这样, TMOD=00H。

2. 方式 0 为 13 位长度计数结构, 设计数初值为 X, 则: $(2^{13}-X) \times 1 \times 10^{-6} = 500 \times 10^{-6}$ 得 X=7692D

X=111100001100B 转成 16 进制后, 高 8 位=FOH, 低 8 位=0CH。即 TH1=FOH, TL0=0CH。

3. 开放定时器/计数器 T1 的中断, IE=#88H(或 EA=1, ET1=1)。

4. 由控制寄存器 TCON 中的 TR1 位来控制定时的启动和停止, TR1=1 启动, TR1=0 停止。

在我的文档中建立一个文件目录(S17), 然后建立一个 S17.uv2 的工程项目, 最后建立源程序文件(S17.asm)。

输入下面的程序:

```

序号:1      ORG 0000H
2          LJMP MAIN
3          ORG 001BH
4          LJMP INSER
5          ORG 030H
6 MAIN:    MOV TMOD,#00H
7          MOV TH1,#0F0H
8          MOV TL1,#0CH
9          SETB EA
10         SETB ET1
11 LOOP:   SETB TR1
12 HERE:   SJMP HERE
13         ORG 0200H
14 INSER:  MOV TH1,#0F0H
15         MOV TL1,#0CH
16         CPL P1.7
17         RETI
18         END
  
```

编译通过后, 将 S17 文件夹中的 hex 文件烧录到 89C51 芯片中, 将芯片插入到 S1 型 LED 试验板上, 接上 5V 稳压电源, 蜂鸣器中立即响起悦耳的 1KHz 音频声。

下面我们对程序进行分析解释。

序号 1(程序解释, 以下同): 程序开始。

序号 2: 跳转到 MAIN 主程序处。

序号 3: 定时器 T1 中断入口地址 001BH。

序号 4: 跳转到中断服务程序 INSER 处。

序号 5: 主程序 MAIN 从地址 0030H 开始。

序号 6: 置 T1 为方式 0。

序号 7、8: 载入定时初值。

序号 9: 开总中断。

序号 10: 定时器 T1 允许中断。

序号 11: 启动定时器 T1。

序号 12: 等待中断, 虚拟主程序。

序号 13: 中断服务子程序从地址 0200H 开始。

序号 14、15: 重装定时初值。

序号 16: P1.7 输出端取反。

序号 17: 中断返回。

序号 18: 程序结束。

再在 S1 板上做一个实验, 利用外中断方式进行数据采集。实验的原理框图如图 4。

将 P1 口的外接短路块取下, 由随机配带的试验线来置 P1 口各端的



高、低电平。试验线一端插到标示有“0”电平的排针上, 另一端如插入 P1 口的某一端排针, 则此端被置低电平; 如不插入, 则由于 P1 口内部具有上拉电阻, 此端为高电平。置好一组数据后, 用另一根试验线将 P3.2/INT0 接地(只需碰一下), 向单片机发出中断申请, 单片机响应后执行中断服务子程序, 读入这组数据, 经内部处理后再送给 P0 口输出显示。我们做的试验是: 采用外部中断 0 方式进行数据采集, 向 P1 口置数 10101010B, 单片机采集这组数据并取反, 然后从 P0 口输出 01010101B 并显示。

在我的文档中建立一个文件目录(S18), 然后建立一个 S18.uv2 的工程项目, 最后建立源程序文件(S18.asm)。

输入下面的程序:

```

序号:1      ORG 0000H
2          LJMP MAIN
3          ORG 0003H
4          LJMP 0100H
5          ORG 030H
6 MAIN:    MOV A,#0FFH
7          SETB EA
8          SETB EX0
9          LOOP:MOV P0,A
10         LCALL DEL
11         AJMP LOOP
12 DEL:    MOV R7,#0FFH
13 DEL1:   MOV R6,#0FFH
14 DEL2:   DJNZ R6,DEL2
15         DJNZ R7,DEL1
16         RET
17         ORG 0100H
18         MOV A,P1
19         CPL A
20         MOV P0,A
21         RETI
22         END
  
```

编译通过后, 将 S18 文件夹中的 hex 文件烧录到 89C51 芯片中, 将芯片插入到 S1 型 LED 试验板上, 取下 P1 口及 P3.2 处的短路块, 接上 5V 稳压电源, 这时 P0 口外接的 8 个 LED 均不亮(输出状态为 FFH)。将 P1 口置成 10101010B, 用另一根试验线将 P3.2/INT0 接地(只需碰一下), 这时 P0 口输出 01010101B 并点亮相应的

LED 作显示。读者朋友也可自己多试几组数据。单片机的数据采集系统非常有用,如你将来要设计一个数码温度表时,先将温度量转换成电压(流量),然后进行模/数转换,转换完成后向单片机发出中断申请,单片机响应后即将数据读入,经运算处理后以数码方式进行显示。

下面我们对程序进行分析解释。

序号 1(程序解释,以下同):程序开始。

序号 2:跳转到 MAIN 主程序处。

序号 3:外部中断 0 入口地址 0003H。

序号 4:跳转到 0100H 地址处。

序号 5:主程序 MAIN 从地址 0030H 开始。

序号 6:累加器 A 置初值 0FFH。

序号 7:开放总中断。

序号 8:开放外部中断 $\overline{INT0}$ 。

序号 9:将累加器内容送 PO 口显示。

序号 10:调用延时子程序,维持 PO 口点亮时间,便于观察。

序号 11:跳转到标号 LOOP 处循环运行。

序号 12~17:延时子程序。

序号 18:中断服务程序从地址 0100H 处开始。

序号 19:将 P1 口数据读入累加器。

序号 20:累加器内容取反。

序号 21:将累加器内容送 PO 口显示。

序号 22:中断返回。

序号 23:程序结束。

最后做一个实验,在 S1 板上做一个中断嵌套实验。利用外部中断源 $\overline{INT0}$ 、 $\overline{INT1}$ 实现中断,其中 $\overline{INT1}$ 设置为高优先级。开机后执行主程序,PO 口输出 00H,PO 口外接的 8 个 LED 全部点亮。有低优先级中断 $\overline{INT0}$ 产生时,P1 口输出 10 秒钟的 00H,P1 口外接的 8 个 LED 全部点亮,其它 LED 均熄灭,然后返回。当有高优先级中断 $\overline{INT1}$ 产生时,P2 口输出 2 秒钟的 00H,P2 口外接的 8 个 LED 全部点亮,P1 口外接的 8 个 LED 均熄灭,然后返回。

在我的文档中建立一个文件目录(S19),然后建立一个 S19.uv2 的工程项目,最后建立源程序文件(S19.asm)。

输入下面的程序:

```
序号:1      ORG 0000H
          2      LJMP MAIN
```

```
3      ORG 0003H
4      LJMP INSERO
5      ORG 0013H
6      LJMP INSER1
7      ORG 030H
8      MAIN:MOV SP,#70H
9      MOV IE,#85H
10     SETB PX1
11     LOOP:MOV P0,#00H
12     MOV P1,#0FFH
13     MOV P2,#0FFH
14     SJMP LOOP
15     ORG 0100H
16     INSERO:MOV R5,#3FH
17     DS0:  MOV P0,#0FFH
18     MOV P1,#00H
19     MOV P2,#0FFH
20     LCALL DEL
21     DJNZ R5,DS0
22     RETI
23     ORG 0150H
24     INSER1:MOV A,R5
25     PUSH ACC
26     MOV R5,#0FH
27     DS1:  MOV P0,#0FFH
28     MOV P1,#0FFH
29     MOV P2,#00H
30     LCALL DEL
31     DJNZ R5,DS1
32     POP ACC
33     MOV R5,A
34     RETI
35     ORG 0200H
36     DEL:  MOV R4,#0FFH
37     DEL1:MOV R3,#0FFH
38     DEL2:DJNZ R3,DEL2
39     DJNZ R4,DEL1
40     RET
41     END
```

编译通过后,将 S19 文件夹中的 hex 文件烧录到 89C51 芯片中,将芯片插入到 S1 型 LED 试验板上,取下 P3.2 及 P3.3 处的短路块,接上 5V 稳压电源,这时 PO 口外接的 8 个 LED 点亮。用一根试验线将 P3.2/ $\overline{INT0}$ 接地(只需碰一下),这时 PO 口的 LED 熄灭,而 P1 口的 8 个 LED 点亮 10 秒钟,即处理中断 0 的服务子程序,10 秒后又回到 PO 口 LED 点亮、其它口的 LED 熄灭状态。如果在 P1 口 LED 点亮的过程中(中断 0 服务子程序处理),用试验线将 P3.3/ $\overline{INT1}$ 接地(只需碰一下),则这时 P1 口 LED 熄灭,而 P2 口 LED 点亮 2 秒钟,即优先处理中断 1 的服务子程序,处理完后再去处理中断 0 的服务子程序,随后恢复。通过这个实验,我们可深刻体会中断及嵌套的过程。

下面我们对程序进行分析解释。

序号 1(程序解释,以下同):程序开始。

序号 2:跳转到 MAIN 主程序处。

序号 3:外部中断 0 入口地址 0003H。

序号 4:跳转到标号 INSERO 处。

序号 5:外部中断 1 入口地址 0013H。

序号 6:跳转到标号 INSER1 处。

序号 7:主程序 MAIN 从地址 0030H 开始。

序号 8:主程序开始。设置堆栈指针为 70H。

序号 9:开放总中断。开放总中断及中断 0、中断 1。

序号 10:设置中断 1 为高优先级。

序号 11~13:PO 置 00H,点亮 8 个 LED。P1、P2 口的 LED 熄灭。

序号 14:跳转到标号 LOOP 处循环运行。

序号 15:中断 0 服务子程序从地址 0100H 处开始。

序号 16:R5 置初值 3FH。

序号 17~19:P1 置 00H,点亮 8 个 LED。PO、P2 口的 LED 熄灭。

序号 20:调用延时子程序。

序号 21:判 10 秒到否? 10 秒到向下执行,否则转回 DS0 处。

序号 22:中断返回。

序号 23:中断 1 服务子程序从地址 0150H 处开始。

序号 24:将 R5 内容送累加器中。

序号 25:R5 内容入栈保护。

序号 26:R5 重载初值 0FH。

序号 27~29:P2 置 00H,点亮 8 个 LED。PO、P1 口的 LED 熄灭。

序号 30:调用延时子程序。

序号 31:判 2 秒到否? 2 秒到向下执行,否则转回 DS1 处。

序号 32:弹出栈中内容至累加器。

序号 33:恢复 R5 中原先保护的内容。

序号 34:中断返回。

序号 35:延时子程序从地址 0200H 处开始。

序号 36~40:延时子程序。

序号 41:程序结束。

配文优惠邮购(每次邮费保价费 12 元):Keil 51 Windows 集成开发环境(已汉化光盘,邮购代号:K1):46 元。TOP851 多功能编程器(邮购代号:B1):400 元。LED 输出试验板(邮购代号:S1):90 元。LED 数码管输出试验板(邮购代号:S2):140 元。5V 高稳定专用稳压电源(邮购代号:D1):35 元。邮购时只需在附言栏中写明邮购代号及数量并附上联系电话即可。

邮购地址:201103 上海市闵行区莲花路 2151 弄 57 号 201 室

联系人:吕超亚

电话:021-64066571 13044152947

技术支持 E-mail:zch2151@sohu.com ◀