

51单片机汇编语言教程：第25课-单片机键盘接口程序设计

（基于 HJ-1G、HJ-3G 实验板）

键盘是由若干按钮组成的开关矩阵，它是单片机系统中最常用的输入设备，用户能通过键盘向计算机输入指令、地址和数据。一般单片机系统中采用非编码键盘，非编码键盘是由软件来识别键盘上的闭合键，它具有结构简单，使用灵活等特点，因此被广泛应用于单片机系统。

按钮开关的抖动问题

组成键盘的按钮有触点式和非触点式两种，单片机中应用的一般是由机械触点组成的。在下图中，当开

〈键盘结构图〉

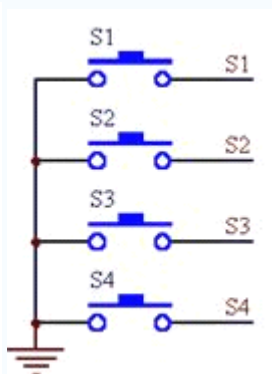


关 S 未被按下时，P1.0 输入为高电平，S 闭合后，P1.0 输入为低电平。由于按钮是机械触点，当机械触点断开、闭合时，会有抖动，P1.0 输入端的波形如图2所示。这种抖动对于人来说感觉不到的，但对计算机来说，则是完全能感应到的，因为计算机处理的速度是在微秒级，而机械抖动的时间至少是毫秒级，对计算机而言，这已是一个“漫长”的时间了。前面我们讲到中断时曾有个问题，就是说按钮有时灵，有时不灵，其实就是这个原因，你只按了一次按钮，可是计算机却已执行了多次中断的过程，如果执行的次数正好是奇数次，那么结果正如你所料，如果执行的次数是偶数次，那就不对了。

为使 CPU 能正确地读出 P1 口的状态，对每一次按钮只作一次响应，就必须考虑如何去除抖动，常用的去抖动的办法有两种：硬件办法和软件办法。单片机中常用软件法，因此，对于硬件办法我们不介绍。软件法其实很简单，就是在单片机获得 P1.0 口为低的信息后，不是立即认定 S1 已被按下，而是延时 10 毫秒或更长一些时间后再次检测 P1.0 口，如果仍为低，说明 S1 的确按下了，这实际上是避开了按钮按下时的抖动时间。而在检测到按钮释放后（P1.0 为高）再延时 5-10 个毫秒，消除后沿的抖动，然后再对键值处理。不过一般情况下，我们常常不对按钮释放的后沿进行处理，实践证明，也能满足一定的要求。当然，实际应用中，对按钮的要求也是千差万别，要根据不一样的需要来编制处理程序，但以上是消除键抖动的

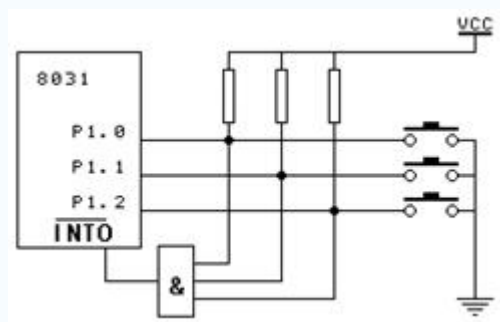
原则。

键盘与单片机的连接



〈键盘连接〉

图3



〈单片机与键盘接口图〉

图4

1、通过I/O口连接。将每个按钮的一端接到单片机的 I/O 口，另一端接地，这是最简单的办法，如图3所示是实验板上按钮的接法，四个按钮分别接到 P3.2 、P3.3、P3.4和 P3.5。对于这种键各程序能采用持续查询的办法，功能就是：检测是否有键闭合，如有键闭合，则去除键抖动，判断键号并转入对应的键处理。下面给出一个例程。其功能很简单，四个键定义如下：

P3.2: 开始，按此键则灯开始流动（由上而下）

P3.3: 停止，按此键则停止流动，所有灯为暗

P3.4: 上，按此键则灯由上向下流动

P3.5: 下，按此键则灯由下向上流动

UpDown EQU 00H ;上下行标志

StartEnd EQU 01H ;起动及停止标志

LAMPCODE EQU 21H ;存放流动的数据代码

ORG 0000H

AJMP MAIN

ORG 30H

MAIN:

MOV SP, #5FH

MOV P1, #0FFH

CLR UpDown ;启动时处于向上的状态

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

```
CLR StartEnd ;启动时处于停止状态

MOV LAMPCODE, #0FEH ;单灯流动的代码

LOOP:

ACALL KEY ;调用键盘程序

JNB F0, LNEXT ;若无键按下, 则继续

ACALL KEYPROC ;不然调用键盘处理程序

LNEXT:

ACALL LAMP ;调用灯显示程序

AJMP LOOP ;反复循环, 主程序到此结束

DELAY:

MOV R7, #100

D1: MOV R6, #100

DJNZ R6, $

DJNZ R7, D1

RET

;-----延时程序, 键盘处理中调用

KEYPROC:

MOV A, B ;从 B 寄存器中获取键值

JB ACC. 2, KeyStart ;分析键的代码, 某位被按下, 则该位为1 (因为在键盘程序中已取反)

JB ACC. 3, KeyOver

JB ACC. 4, KeyUp

JB ACC. 5, KeyDown

AJMP KEY_RET

KeyStart:

SETB StartEnd ;第一个键按下后的处理

AJMP KEY_RET

KeyOver:

CLR StartEnd ;第二个键按下后的处理

AJMP KEY_RET

KeyUp: SETB UpDown ;第三个键按下后的处理
```

推荐使用慧净 51 实验板。推荐 51 学习网 WWW.HLMCU.COM 淘宝网: <http://shop37031453.taobao.com/>

[51 单片机汇编语言教程-慧净电子会员收集整理](#) （全部 28 课）

```
AJMP KEY_RET  
  
KeyDown:  
  
CLR UpDown ;第四个键按下后的处理  
  
KEY_RET:RET  
  
KEY:  
  
CLR F0 ;清 F0，表示无键按下。  
  
ORL P3, #00111100B ;将 P3口的接有键的四位置1  
  
MOV A, P3 ;取 P3的值  
  
ORL A, #11000011B ;将其余4位置1  
  
CPL A ;取反  
  
JZ K_RET ;如果为0则一定无键按下  
  
ACALL DELAY ;不然延时去键抖  
  
ORL P3, #00111100B  
  
MOV A, P3  
  
ORL A, #11000011B  
  
CPL A  
  
JZ K_RET  
  
MOV B, A ;确实有键按下，将键值存入 B 中  
  
SETB F0 ;设置有键按下的标志  
  
K_RET:  
  
ORL P3, #00111100B ;此处循环等待键的释放  
  
MOV A, P3  
  
ORL A, #11000011B  
  
CPL A  
  
JZ K_RET1 ;直到读取的数据取反后为0说明键释放了，才从键盘处理程序中返回  
  
AJMP K_RET  
  
K_RET1:  
  
RET  
  
D500MS: ;流水灯的延迟时间  
  
PUSH PSW
```

推荐使用慧净 51 实验板。推荐 51 学习网 WWW.HLMCU.COM 淘宝网: <http://shop37031453.taobao.com/>

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

```
SETB RS0
MOV R7, #200
D51: MOV R6, #250
D52: NOP
NOP
NOP
NOP
DJNZ R6, D52
DJNZ R7, D51
POP PSW
RET
LAMP:
JB StartEnd, LampStart ;如果 StartEnd=1, 则启动
MOV P1, #0FFH
AJMP LAMPRET ;不然关闭所有显示, 返回
LampStart:
JB UpDown, LAMPUP ;如果 UpDown=1, 则向上流动
MOV A, LAMPCODE
RL A ;实际就是左移位而已
MOV LAMPCODE, A
MOV P1, A
LCALL D500MS
AJMP LAMPRET
LAMPUP:
MOV A, LAMPCODE
RR A ;向下流动实际就是右移
MOV LAMPCODE, A
MOV P1, A
LCALL D500MS
LAMPRET:
```

[51 单片机汇编语言教程-慧净电子会员收集整理（全部 28 课）](#)

RET

END

以上程序功能很简单，但它演示了一个单片机键盘处理程序的基本思路，程序本身很简单，也不很实用，实际工作中还会有好多要考虑的因素，比如主循环每次都调用灯的循环程序，会造成按钮反应“迟钝”，而如果一直按着键不放，则灯不会再流动，一直要到松开手为止，等等，大家能仔细考虑一下这些问题，再想想有什么好的解决办法。

2、采用中断方式：如图4所示。各个按钮都接到一个与非上，当有任何一个按钮按下时，都会使与门输出为低电平，从而引起单片机的中断，它的好处是不用在主程序中持续地循环查询，如果有键按下，单片机再去做对应的处理

[51 实验板推荐\(点击下面的图片可以进入下载资料链接\)](#)

