

手把手教你学 PIC 单片机 C 语言教程 第 13 课

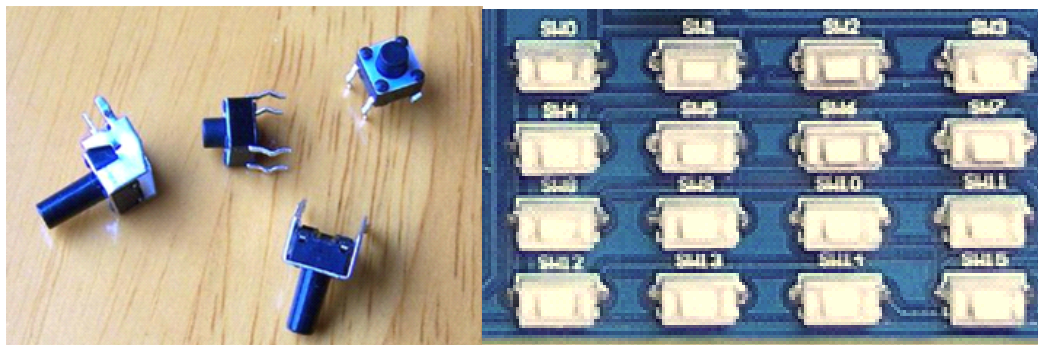
(按钮实验)

参考例程所在位置：HL-K18 配套例程\7 KEY1X4

参考例程所在位置：HL-K18 配套例程\8 KEY4X4

1. 开发板键盘模块简介

作为人机对话的主体，键盘是一个实验系统或者开发系统中最为常见的部件。



一般键盘可以构成两类模式：一种是动态键盘（也称矩阵键盘），投入的 I/O 接口线较少而可以设置较多的键盘；另一种是静态键盘（也称独立式键盘），接口线的数量与键盘数相等。

1.1 动态键盘（也称矩阵键盘）SW0-SW15

动态键盘在应用系统中使用比较广泛，常用作数字键和功能键，TOPPIC PLUS 单片机实验开发系统选用一组 4X4 动态键盘，分别为 SW0、SW1、SW2、SW3、SW4、SW5、SW6、SW7、SW8、SW9、SW10、SW11、SW12、SW13、SW14、SW15。4X4 动态键盘占用 8 条口线。

为确保辨别出每一个按键，列线经过一个限流电阻后接至高电平。在此种连接方式下，一般可用两种方式进行软件编程：键盘的动态扫描和静态扫描。此两种编程设计方案对于控制系统的构架有较大的影响，工程技术人员应给予充分重视。下面对这两种设计思想进行概述。

(1) 动态扫描方式

一般配合 4 位八段动态数码显示器动态扫描进行工作：列线 COL₄-COL₁ 依次置 0(对应的，也可以是 COL₄-COL₁ 依次置 1，“COL₄-COL₁”和“COL₄-COL₁”是反向关系)，如果发现行线为 0，即表示该列线上有键输入。为了实现实时判键输入，动态扫描过程必须周而复始循环，会占用较多的单片机工作资源，在有些场合应谨慎采用。

(2) 静态扫描方式

上述动态扫描方式应用比较普通，对应某些特殊中断方式的单片机才能够采用静态扫描方式，譬如 PIC 单片机 RB₄~7 引脚具有的电平变化中断的方式就可以实现这种静态扫描功能，同时，HL-K18 开发板的 SW₀、SW₁ 应当用软跳线（也叫实验插线、杜帮线）改接到 RB₆ 或者 RB₇。列线 COL₄-COL₁ 与低电平稳定连接(对应的，也可以是 COL₄-COL₁ 与高电平稳定连接)，若出现键盘按下，则由 RB₄~7 所连接的行线（SW₀-SW₃）出现低电平而引起电平变化中断，在中断处理程序中，用与上述动态扫描方式相同的检查方法检查具体是哪个键按下，检查出是哪个键按下后退出在中断处理程序，此种方式比较节省单片机工作资源。PIC

单片机一旦采用 RB3、RB6 和 RB7 的引脚，在线调试功能就不再有效，RB4~7 引脚必须处于一种脱机方式下才能够使用。

1.2 静态键盘（也称独立式键盘）SW0-SW3

通过对 4*4 动态键盘的特殊设置，可以将其变换成静态键盘使用，具体方法有两种：

(1) 软件方式：

将 COL1 置高，对应的 COL_1 始终为低，这样 SW0、SW1、SW2、SW3 可以作为四个静态键盘使用。

(2) 硬件方式：

去掉“COL1-RA1”短路帽，用软跳线（也叫实验插线、杜帮线）连接 COL1 插针和 J1 座的 VCC 插针，这样 SW0、SW1、SW2、SW3 可以作为三个静态键盘使用。建议采用软件方式，比较方便。

静态键盘的使用比较随意，一般作为某种信号的输入或功能的设定。一般可用下述 2 种方式进行软件编程：

(1) 查询方式。按键处于正常状态下，外引出接线柱为高电平；当按下键盘后，外引出接线柱为低电平。通过判断接线柱的电平状态，可以很方便地获得按键的工作情况。具体的，先判断静态键盘是否按下，若按下就调用 10~20 ms 延时；进而再判断按键是否释放，若释放即为一次完整的按键过程。10~20 ms 延时具有很强的物理含意，主要用来防止机械或其他原因引起的抖动。在一次完整的按键后，可以连接其他的程序，以实现特定的功能。查询方式下，单片机需要不断地查询按键是否按下，会占用较多的单片机工作资源。

(2) 中断方式。接在 RB0 上的 SW0 可使用中断方式。若出现键盘按下，则出现低电平而引起中断，在中断处理程序中，先完成与该按键对应的处理程序，再延时 10~20ms，退出中断处理程序。此种方式比较节省单片机工作资源。

2. 编程实例

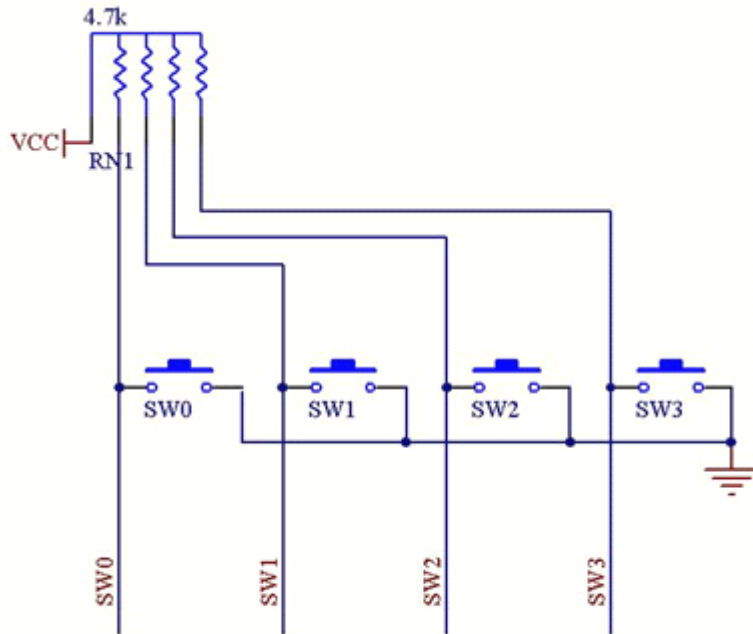
2.1 独立式键盘编程

从前面的介绍中，我们知道一般键盘可以构成两类模式：一种是矩阵键盘，另一种是独立式键盘，我们先介绍比较简单的独立式键盘（也叫静态键盘）。独立式键盘一般可用查询方式或中断方式编程，这里我们只介绍查询方式，它相对而言比较简单些。

按键处于正常状态下，外引出接线柱为高电平；当按下键盘后，外引出接线柱为低电平。通过判断接线柱的电平状态，可以很方便地获得按键的工作情况。单片机不断地查询 SW0、SW1、SW2、SW3 这四个按键是否按下，并在 1 位 8 段静态数码显示器显示按键按下的情况。

注：关于 1 位八段静态数码显示器的原理和设置要点，大家可以参考后面“动态数码自动计数显示实验”这一讲。

简化电路原理图如下：

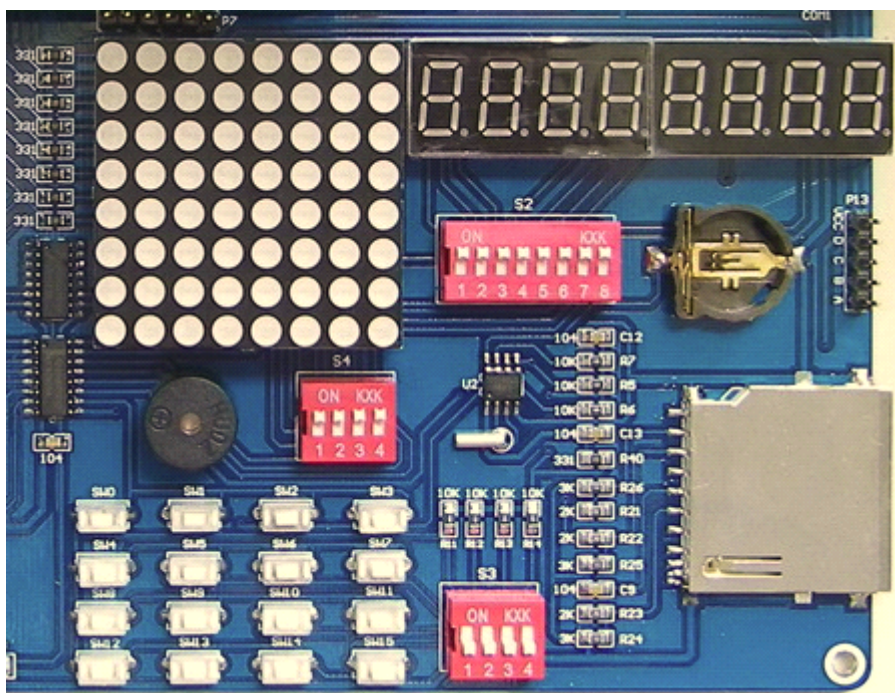


独立式键盘例程如下：

```
#include <p18F4520.h>
#include "k52.h"
#include "lcd1602.h"
void main(void)
{
    K52_init();
    TRISD=0X00; /*设置 D 口为输出*/
    PORTD=0X00;
    COL1=1; /*将数码管设置成 1 位静态方式*/
    COL2=0;
    COL3=0;
    COL4=0;
    while(1)
    {
        if (SW0==0)
        {
            PORTD=0x3f; /*显示 SW0 曾经按下*/
        }
        if (SW1==0)
        {
            PORTD=0x06; /*显示 SW1 曾经按下*/
        }
        if (SW2==0)
        {
            PORTD=0x5b; /*显示 SW2 曾经按下*/
        }
    }
}
```

```
}  
if (SW3==0)  
{  
PORTD=0x4f; /*显示 SW3 曾经按下*/  
}  
}  
}
```

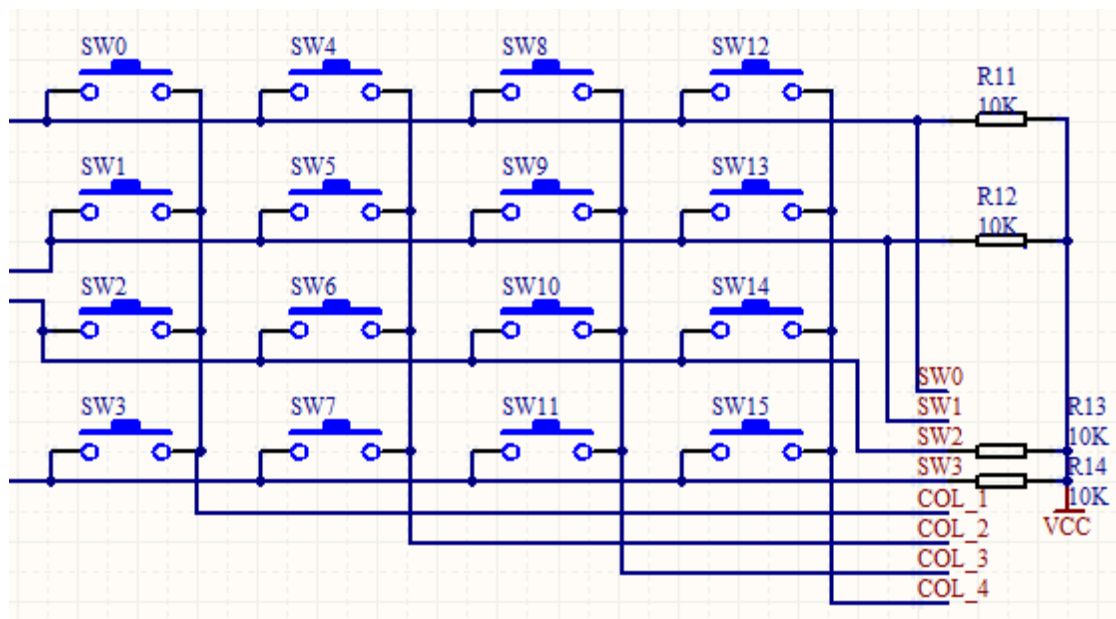
while(1)死循环体内的语句为判断按键及数码管显示的程序代码。分别用 if 语句来对开发板上的 SW0—SW3 按键的状态进行判断，按下相应的键，则在最右边一位数码管上显示对应的键号。同时由于这些语句放在 while(1)循环中，程序会不停地检查 SW0—SW3 按键的状态。以下是程序运行时的效果图：



2. 2 动态键盘（矩阵键盘）编程

动态键盘（矩阵键盘）的编程相对复杂一些，通常将 4*4 键盘的动态扫描和 4 位数码管或者 8*8 点阵管的动态刷新结合在一起，程序分析时要注意这一点。本例程要实现的功能是：在点阵管 C1 列显示列键号，在点阵管 C2 列显示行键号，在点阵管 C3 列显示全亮，在点阵管 C4 列显示“01010101”，注意列键号、行键号都是用 8 位二进制数方式显示的。例如，当我们按下 SW0 时，点阵管 C1 列显示“00000001”（列键号 1），在点阵管第 2 列显示“00000001”（行键号 1），当我们按下 SW11 时，点阵管 C1 列显示“00000100”（列键号 4），在点阵管 C2 列显示“00000011”（行键号 3）。

简化电路原理图如下：



动态键盘例程如下：

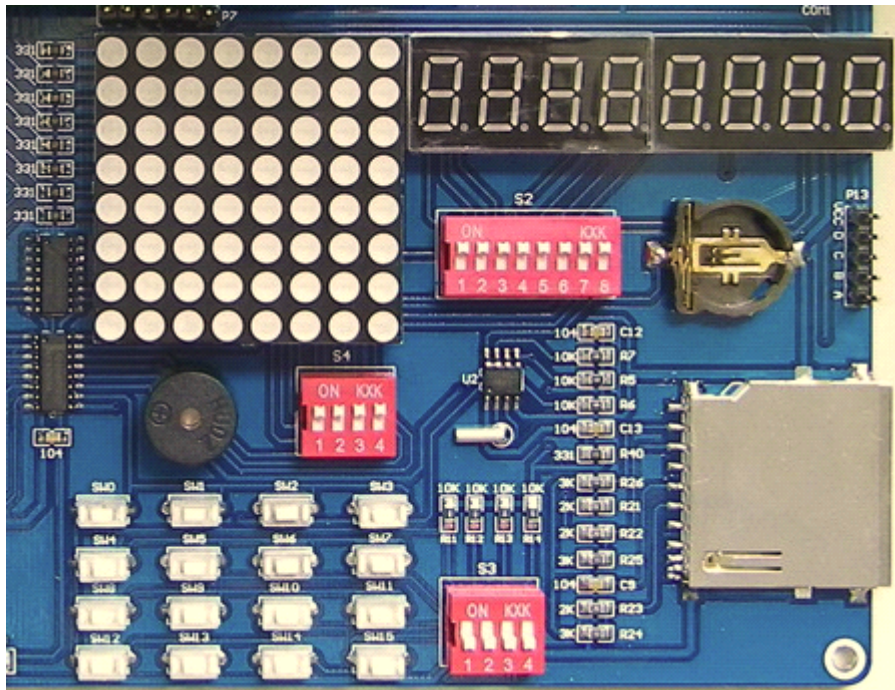
```
#include <pl8cxxx.h>
#include "k18.h"
#include "Delay.h"
unsigned char col;
unsigned char row;
unsigned char keyon;
void readcol(void);
void main(void)
{
k18_init();/*HL-K18 主板初始化*/
COL1=0;
COL2=0;
COL3=0;
COL4=0;
while(1)
{
PORTD=col;//在点阵管第 1 列显示列号
COL1=1;
readcol();
if (keyon==1) row=1;
//有键按下时，更新行信息
COL1=0;
PORTD=row;//在点阵管第 2 列显示行号
COL2=1;
readcol();
if (keyon==1) row=2;
//有键按下时，更新行信息
```

```
COL2=0;
PORTD=0XFF;//在点阵管第 3 列显示全亮
COL3=1;
readcol();
if (keyon==1) row=3;
//有键按下时，更新行信息
COL3=0;
}
PORTD=0b01010101;//在点阵管第 4 列显示“01010101”
COL4=1;
readcol();
if (keyon==1) row=4;
//有键按下时，更新行信息
COL4=0;
}
}
```

```
void readcol(void)
{
unsigned char TEMP;
TEMP=PORTB;
TEMP=TEMP&0b00110101;
keyon=1;
if (TEMP==0b00110100)
col=1;
*else if (TEMP==0b00110001) col=2;
*else if (TEMP==0b00100101) col=3;
*else if (TEMP==0b00010101) col=4;
else keyon=0;
```

其实这个程序也可以改到 4 位数码管上显示行列键号，这里我们主要考虑 8*8 点阵管上编程简单，便于初学者看懂程序。

以下是程序运行时的效果图：



版权声明：（部分资料图片来源于网络）

- 1、本教程为慧净电子会员整理修改，欢迎网上下载、转载、传播、免费共享给各位单片机爱好者！
- 2、该教程可能会存在错误或不当之处，欢迎朋友们指正。
- 3、未经协商便做出不负责任的恶意评价(中评, 差评)，视为自动放弃一切售后服务的权利！
- 4、我们的产品收入一部分是赠送给慈善机构的, 以免影响到你的善心. 大家好, 才是真的好（双方好评）。

下面是有缘人看的，谢谢理解

善有善报，恶有恶报，不是不报，时候未到。
从古至今，阴司放过谁，大家得多行善。
行善积德，爱护动物，哪怕小蚂蚁也是生命。
可改变命运，可心想事成，有利保佑子孙后代更昌盛。
学习弟子规，教我们如何做人，看和谐拯救危机，教我们看宇宙。
看为什么不能吃它们，教我们慈悲心，看因果轮回纪录，教我们懂得因果报应。
切勿造恶，种瓜得瓜种豆得豆，一切都有过程，待成熟之时，福德或果报自来找你。

慧净
2008年8月8日