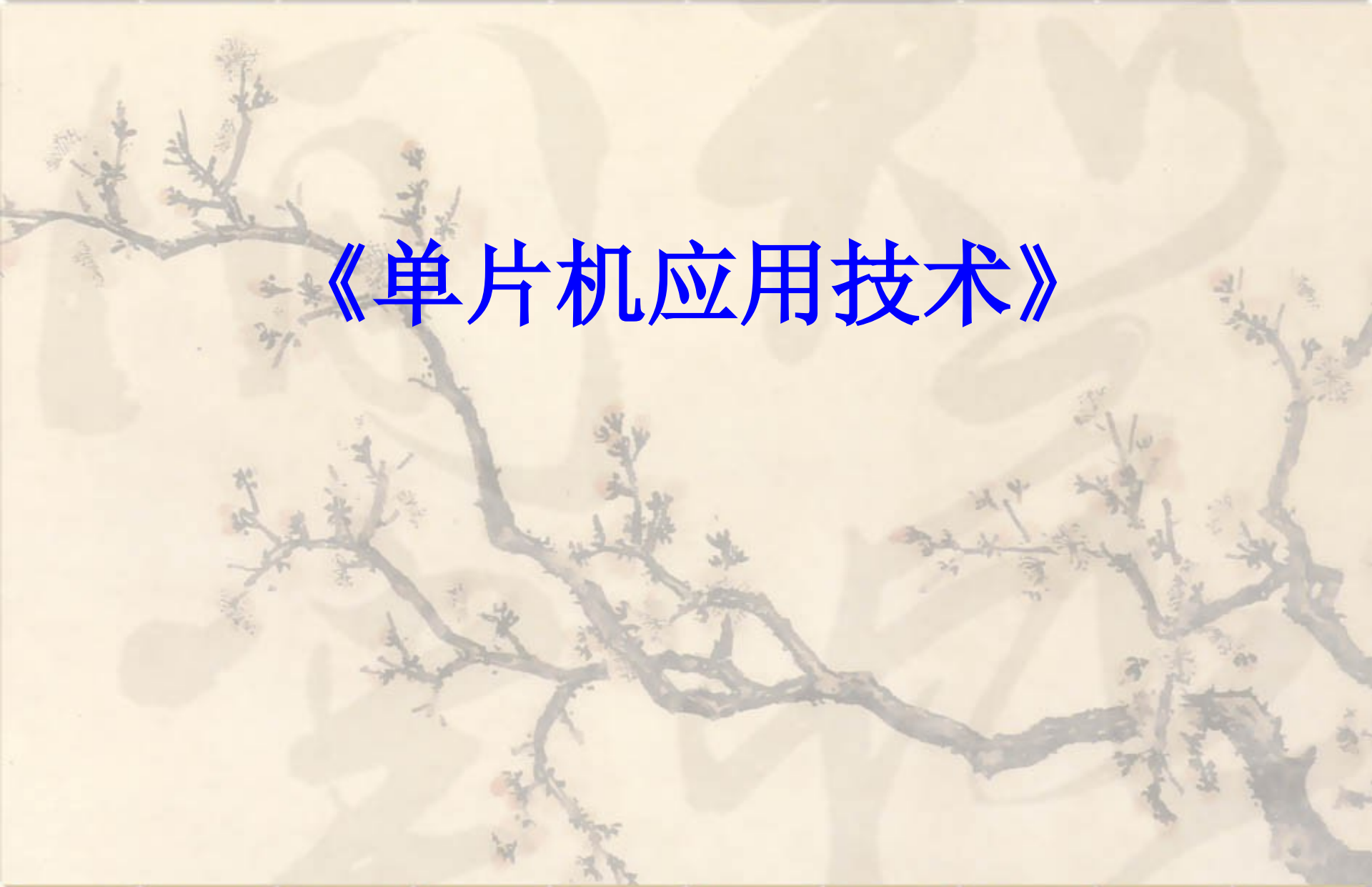




《单片机应用技术》

项目六 按键识别应用

- ❖ 任务 1 项目知识点学习
- ❖ 任务 2 多功能按键控制 LED 灯
- ❖ 任务 3 矩阵式键盘键值识别
- ❖ 任务 4 电话拨号显示控制

【知识目标】

- ❖ 了解按键的确认与去抖原理以及方法
- ❖ 学会矩阵键盘的工作原理与内部连接
- ❖ 学会矩阵键盘的键值识别方法

【能力目标】

- ❖ 掌握按键的去抖方法
- ❖ 掌握矩阵键盘的应用系统的设计
- ❖ 掌握矩阵键盘的程序编写、调试方法。

任务 1 项目知识点学习

一、单片机的 IO 输入接口

二、按键

三、简单键盘接口

四、矩阵式键盘接口

一、单片机的 IO 输入接口

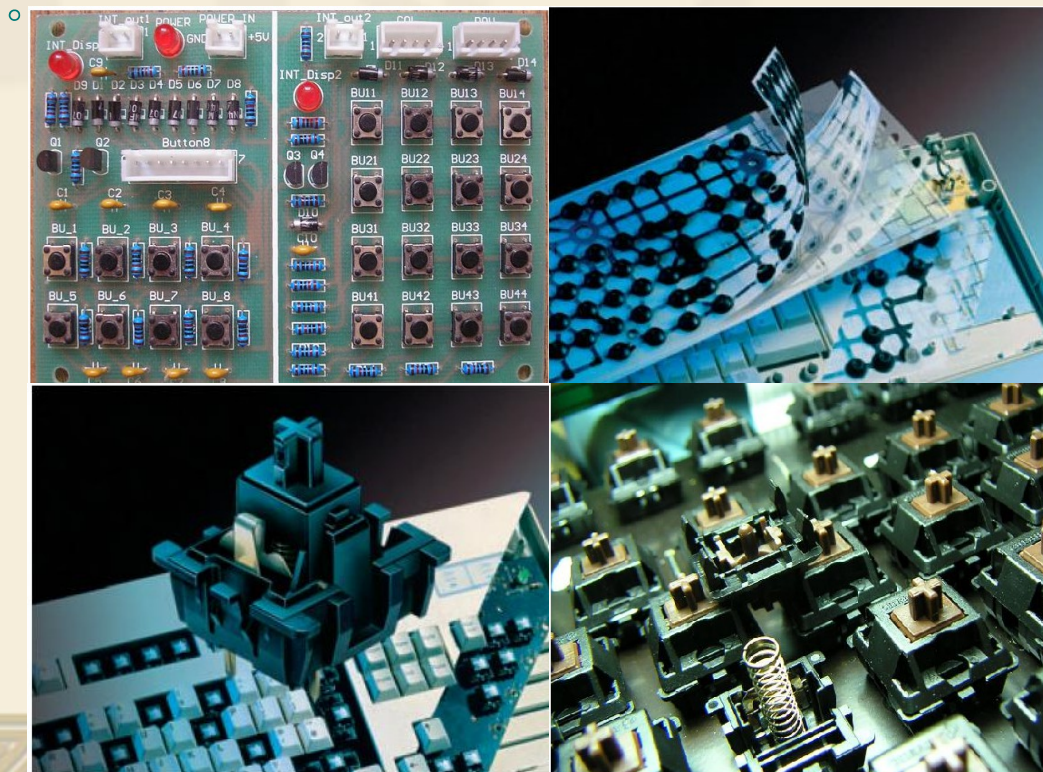
假如把一个单片机嵌入式系统比做一个人的话，那么单片机就相当于人的心脏和大脑，而输入接口就好似人的感官系统，用于获取外部世界的变化、状态等各种信息，并把这些信息输送进人的大脑。嵌入式系统的人机交互通道、前向通道、数据交换和通信通道的各种功能都是由单片机的输入接口及相应的外围接口电路实现的。

对于一个电子系统来讲，外部现实世界各种类型和形态的变化和状态都需要一个变换器将其转换成电信号，而且这个电信号有时还需要经过处理，使其成为能被 MCU 容易识别和处理的数字逻辑信号，这是因为单片机常用的输入接口通常都是数字接口。

单片机 I/O 接口的逻辑是数字逻辑电平，即以电压的高和低作为逻辑“1”和“0”，因此进入单片机的信号要求是电压信号。这些电压信号又可分为单次信号和连续信号。

二、按 键

键盘由一组规则排列的按键组成，一个按键实际上是一个开关元件，也就是说键盘是一组规则排列的开关。单片机使用的按键是一种常开型的开关，平时按键的两个触点处于断开状态，按下键时它们才闭合



1. 按键的分类

2. 按键输入原理

3. 按键开关的抖动问题

1. 按键的分类

按键按照结构原理可分为两类：

1. 触点式开关按键，如机械式开关、导电橡胶式开关等；
2. 无触点开关按键，如电气式按键，磁感应按键等。

前者造价低，后者寿命长。目前单片机系统中最常见的是触点式开关按键。

按键按照接口原理可分为

:

1. 编码键盘
2. 非编码键盘

编码键盘主要是用硬件来实现对键的识别，非编码键盘主要是由软件来实现键盘的定义与识别。

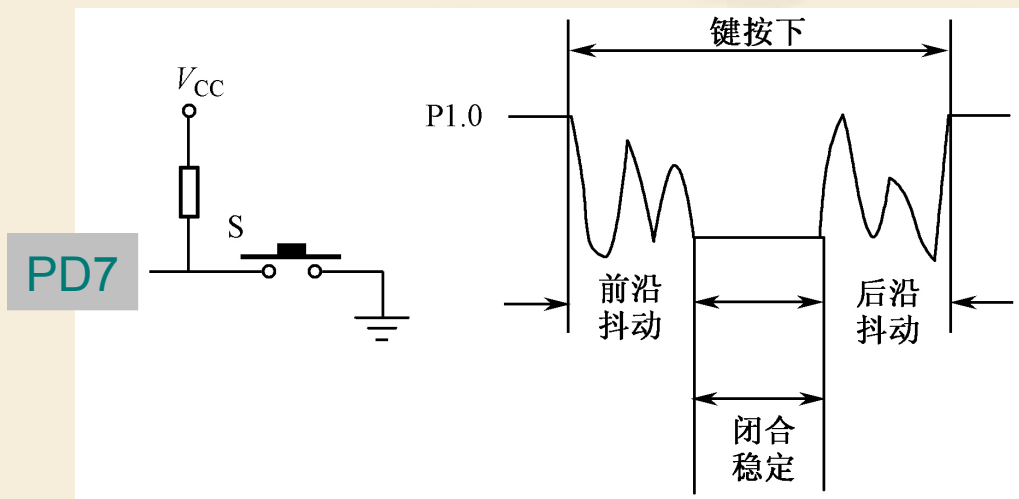
2. 按键输入原理

在单片机应用系统中，除了复位按键有专门的复位电路及专一的复位功能外，其他按键都是以**开关状态**来设置控制功能或输入数据。当所设置的功能键或数字键按下时，单片机系统应完成该按键所设定的功能，按键信息输入是与软件结构密切相关的过程。

对于一个键盘或一组按键，单片机系统中总有一个接口电路与 **CPU** 相连。**CPU** 可以采用**查询或中断方式**了解有无按键输入并检查是哪一个键按下，然后转入执行该键的功能程序，执行完后再返回主程序。

3. 按键开关的抖动问题

单片机系统中应用的一般是由机械触点构成的按键。当机械触点断开、闭合时，会有抖动，如下图所示。在触点抖动期间检测按键的通与断状态，可能导致判断出错。即按键一次按下或释放被错误地认为是多次操作，这种情况是不允许出现的。



常用的去抖动方法有两种

:

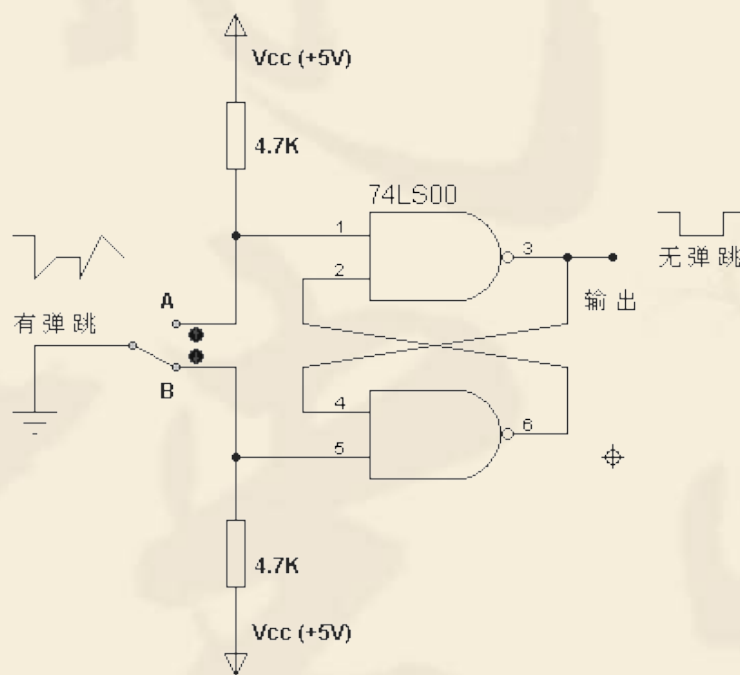
a.

硬件方法

b. 软件方法

a. 硬件方法

在硬件上可采用在键输出端加 **R-S** 触发器（双稳态触发器）或单稳态触发器构成去抖动电路，如图所示。



电路工作过程

电路工作过程:

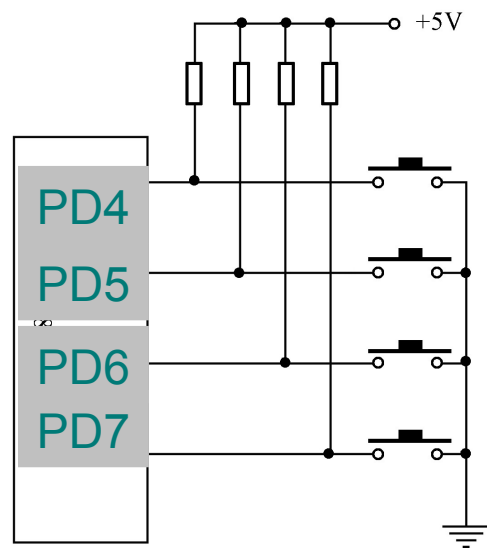
按键未按下时, $a = 0$, $b = 1$, 输出 $Q = 1$, 按键按下时, 因按键的机械弹性作用的影响, 使按键产生抖动, 当开关没有稳定到达 b 端时, 因与非门 2 输出为 0 反馈到与非门 1 的输入端, 封锁了与非门 1, 双稳态电路的状态不会改变, 输出保持为 1, 输出 Q 不会产生抖动的波形。当开关稳定到达 b 端时, 因 $a = 1$, $b = 0$, 使 $Q = 0$, 双稳态电路状态发生翻转。当释放按键时, 在开关未稳定到达 a 端时, 因 $Q = 0$, 封锁了与非门 2, 双稳态电路的状态不变, 输出 Q 保持不变, 消除了后沿的抖动波形。当开关稳定到达 a 端时, 因 $a = 0$, $b = 1$, 使 $Q = 1$, 双稳态电路状态发生翻转, 输出 Q 重新返回原状态。由此可见, 键盘输出经双稳态电路之后, 输出已变为规范的矩形方波。

b. 软件方法

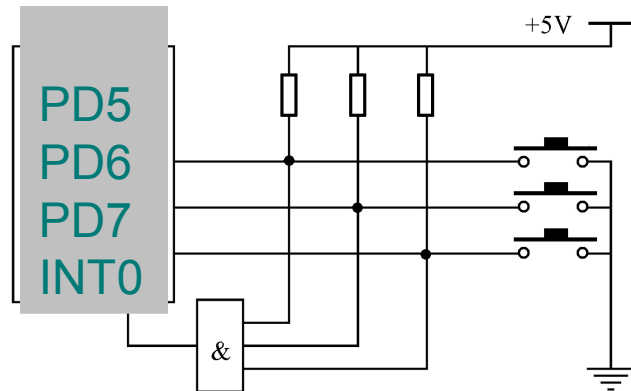
软件方法其实很简单，就是在单片机获得 P1.0 口为低电平的信息后，不是立即认定 S 已被按下，而是延时 10ms 或更长一些时间后再次检测 P1.0 口，如果仍为低电平，说明 S 的确按下了，这实际上是避开了按键按下时的抖动时间。而在检测到按键释放后（P1.0 为高电平）再延时 5 ~ 10ms，消除后沿的抖动，然后再对键值处理。当然，在实际应用中，对按键的要求也是千差万别的，要根据不同的需要来编制处理程序，但以上是消除按键抖动的原则。

三、键盘与单片机的连接

如果系统只需几个按键，可直接采用 I/O 线构成单个按键电路，各个按键之间相互独立，一根线上的按键状态不会影响其他输入线上的工作状态，又称独立式键盘接口电路。



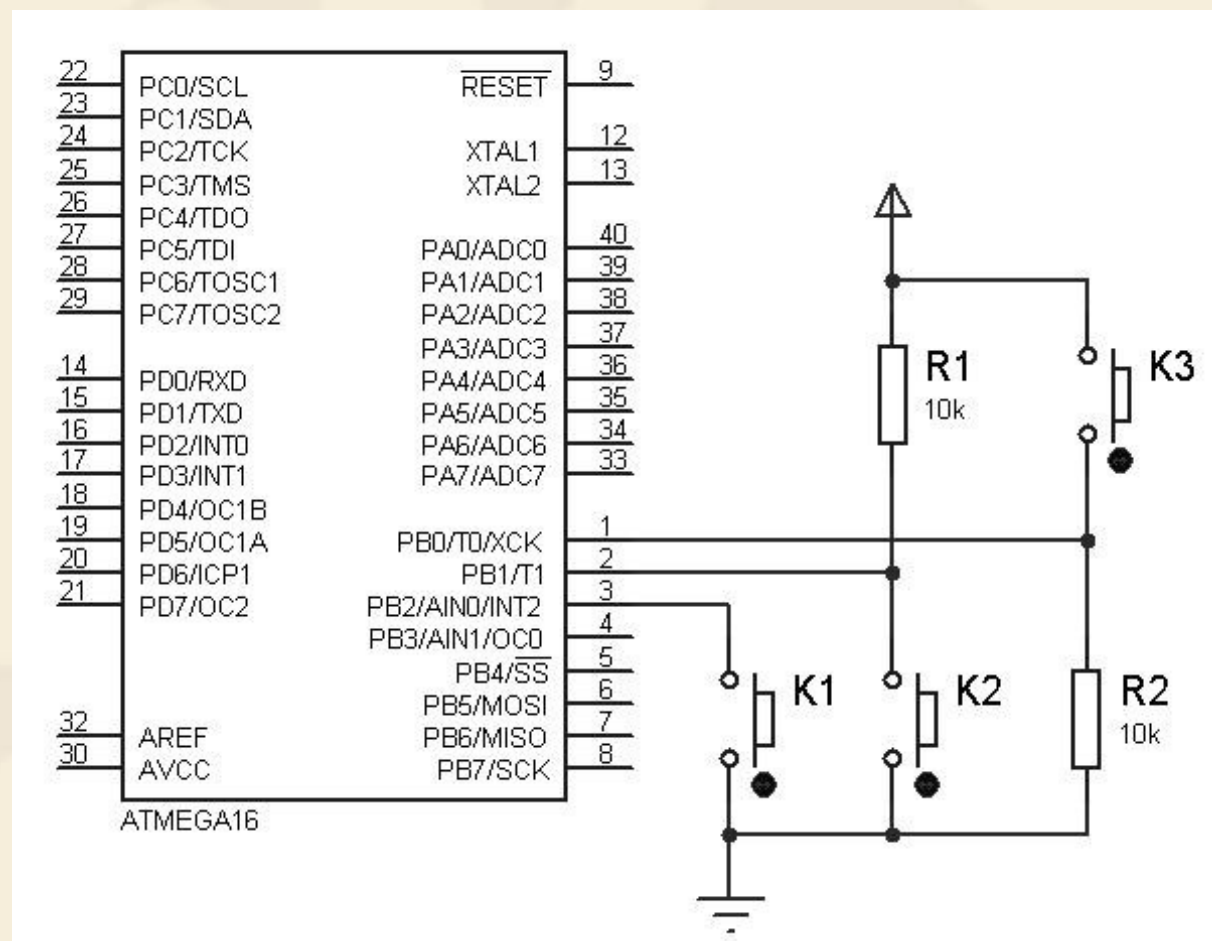
(a) 查询方式



(b) 中断方式

这种电路每个按键需要占用一根 I/O 线，在按键数量较多时，输入口浪费大且电路结构很烦琐，因此这种键盘接口电路只适用于按键较少或要求较高操作速度的场合。

三、键盘与单片机的连接



四、矩阵式键盘

❖ 1.

矩阵式键盘的结构与工作原理

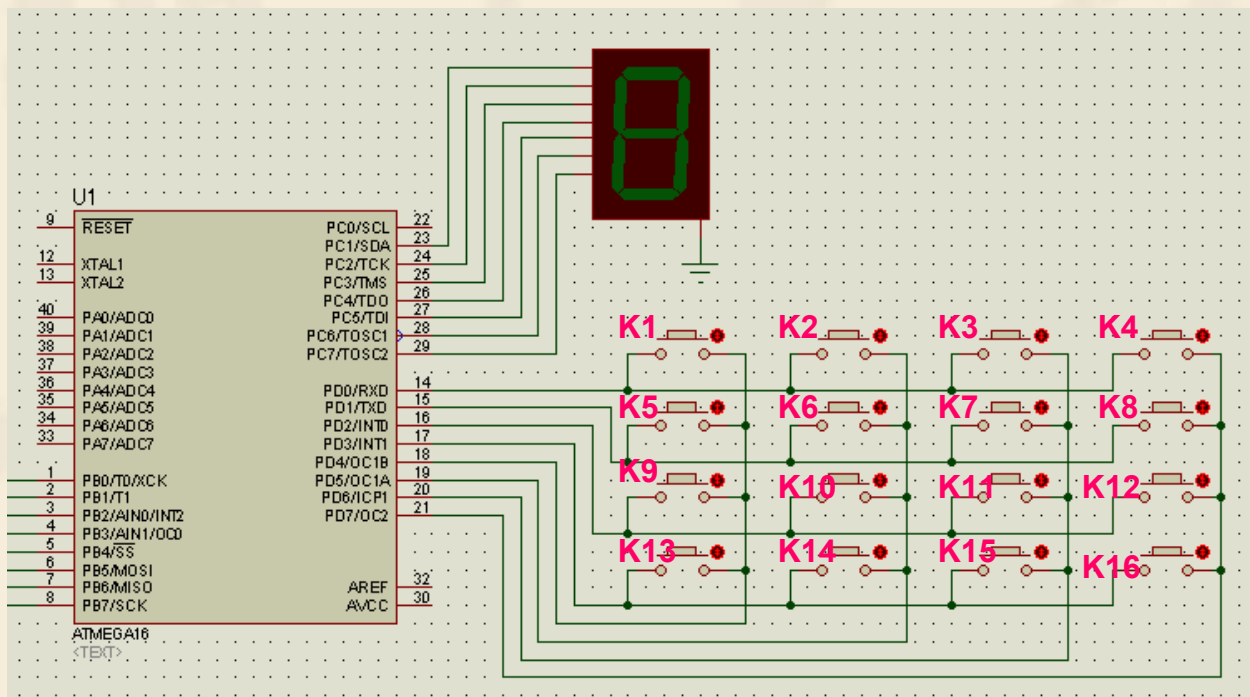
❖ 2. 矩阵式键盘按键的识别

❖ 3. 键盘的编码

❖ 4. 键盘的工作方式

1. 矩阵式键盘的结构与工作原理

用 I/O 口线组成行、列结构，行列线分别连在按键开关的两端，按键处在行列线的交叉点上

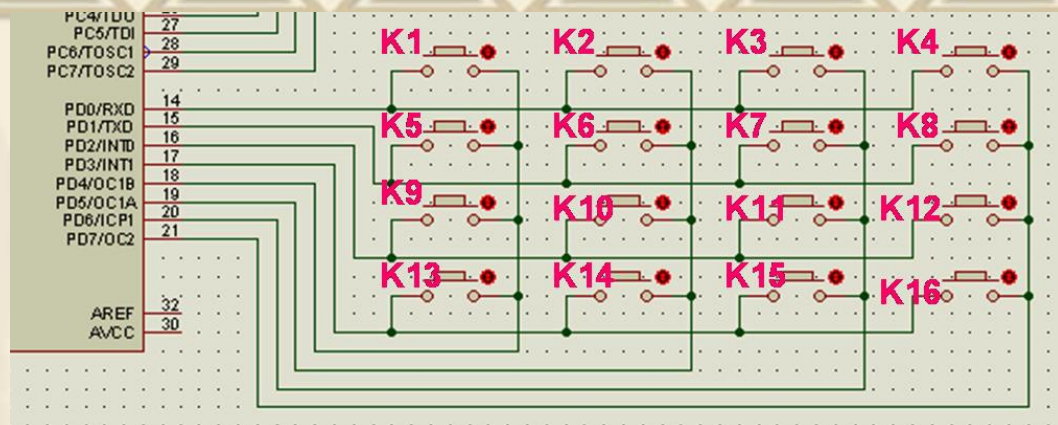


2. 矩阵式键盘按键的识别

a. 扫描法

b. 线反转法

a. 扫描法



以 4×4 键盘为例：

（1）判断是否有键按下：使行线都输出 0，检测列线的电平。如果列线上的电平全为高，则表示没有键被按下。如果列线上的电平不全为高，则表示有键被按下。

（2）按键键值：如果没有键闭合，就返回扫描。如果有键闭合，再进行逐行扫描，找出闭合键的键号。先使 D0=0，D1 ~ D3=1，检测 D4 ~ D7 上的电平，如果 D4=0，表示 K1 键被按下；同理，如果 D5 ~ D7=0，分别表示 K2、K3、K4 键被按下；如果 D4 ~ D7=1，则表示这一行没有键被按下。再使 D1=0，D0、D2、D3 为 1，对第二行进行扫描，这样依次进行下去，直到把闭合的键找到为止。

uchar key_scan()// 扫描法

```
{
  uchar i,j,m,n;
  uchar key=0;
  uchar t[4]={0xf7,0xfb,0xfd,0xfe};

  for(i=0;i<4;i++)
  {
    DDRD=0X0f;
    j=t[i];

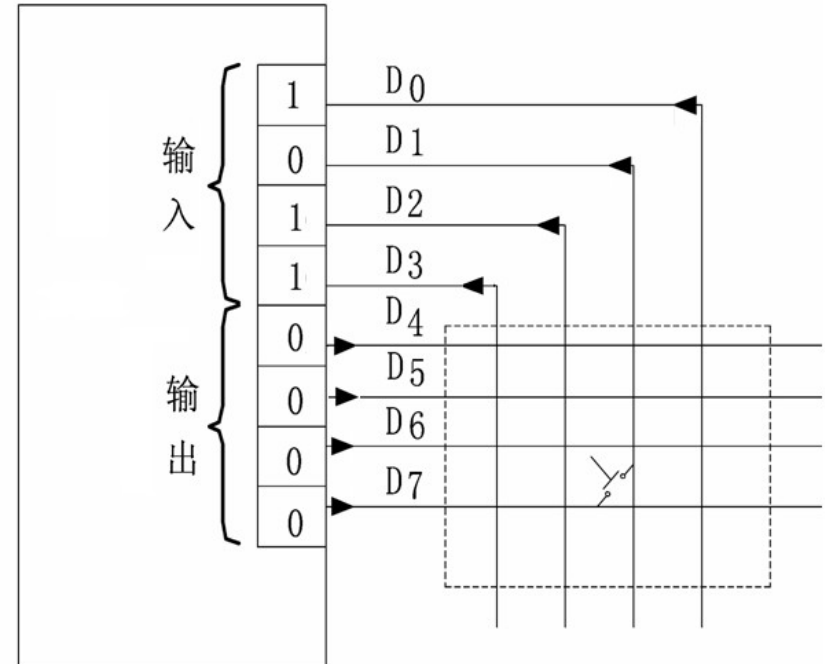
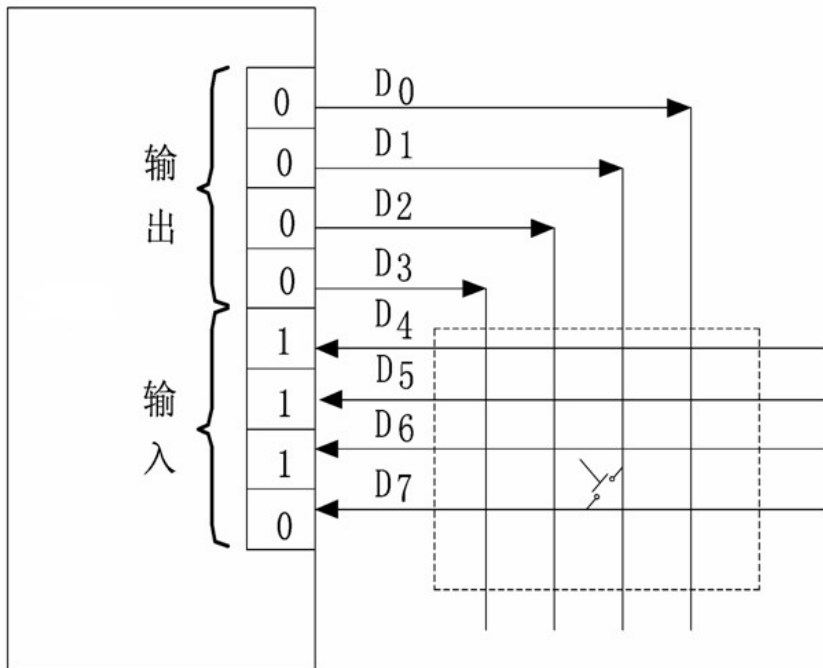
    PORTD=j;

    if((PIND&0X10)==0)
    {
      delay(5);
      if((PIND&0X10)==0)
      {
        while((PIND&0X10)==0);
        return key;
        break;
      }
    }
    key++;
  }
```

```
if((PIND&0X20)==0)
{
  delay(5);
  if((PIND&0X20)==0)
  {
    while((PIND&0X20)==0);
    return key;
    break;
  }
}
key++;
if((PIND&0X40)==0)
{
  delay(5);
  if((PIND&0X40)==0)
  {
    while((PIND&0X40)==0);
    return key;
    break;
  }
}
key++;
```

```
if((PIND&0X08)==0)
{
  delay(5);
  if((PIND&0X08)==0)
  {
    while((PIND&0X08)==0);
    return key;
    break;
  }
}
key++;
}
return 0xff;// 如果没有键按下返回 0xff
}
```

b. 线反转法

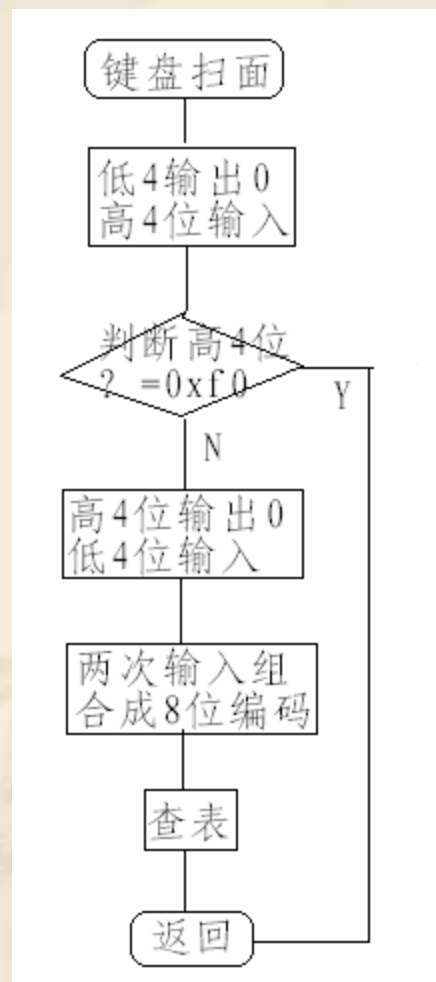
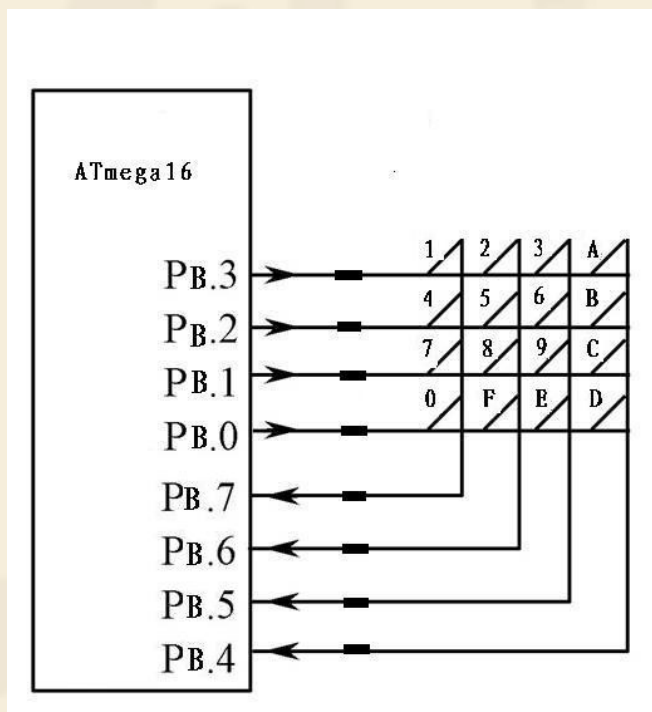


(a) 线反转法第一步
线反转法第二步

(b)

图 线反转法原理

b. 线反转法



3. 键盘的编码

- a. 独立式键盘 按键数量少，可根据实际需要灵活编码。
- b. 矩阵式键盘 按键的位置由行号和列号唯一确定，因此可分别对行号和列号进行二进制编码，然后将两值合成一个字节，高 4 位是行号，低 4 位是列号。

总结：

无论以何种方式编码，均应以处理问题方便为原则，而最基本的就是键所处的物理位置即行号和列号，它是各种编码之间相互转换的基础，编码相互转换可通过计算或查表的方法实现。

4. 键盘的工作方式

- ❖ a. 编程扫描方式
- ❖ b. 定时扫描方式
- ❖ c. 中断扫描方式

a. 编程扫描方式

编程扫描方式是利用 **CPU** 完成其他工作的空余调用键盘扫描子程序来响应键盘输入的要求。在执行键功能程序时，**CPU** 不再响应键输入要求，直到 **CPU** 重新扫描键盘为止。

键盘扫描程序一般应包括以下内容：

- （1）判别有无键按下。
- （2）键盘扫描取得闭合键的行、列值。
- （3）用算法或查表法得到键值。
- （4）判断闭合键是否释放，如没释放则继续等待。
- （5）将闭合键键号保存，同时转去执行该闭合键的功

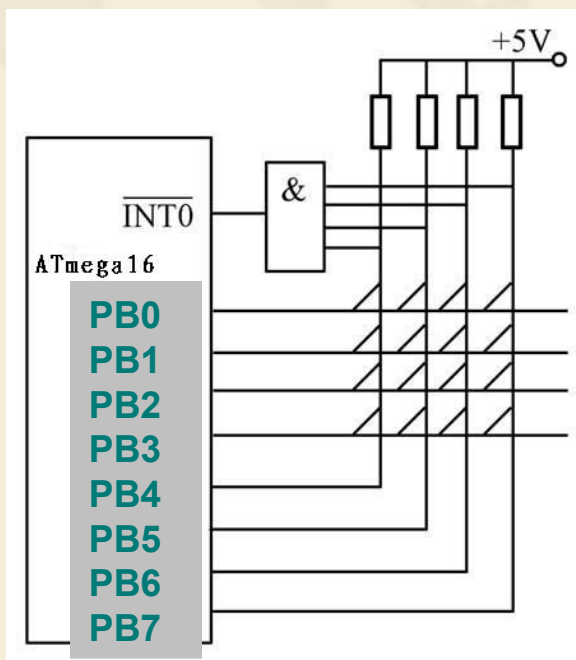
能。

b. 定时扫描方式

定时扫描方式就是每隔一段时间对键盘扫描一次，它利用单片机内部的定时器产生一定时间（例如 **10ms**）的定时，当定时时间到就产生定时器溢出中断，**CPU** 响应中断后对键盘进行扫描，并在有键按下时识别出该键，再执行该键的功能程序。

c. 中断扫描方式

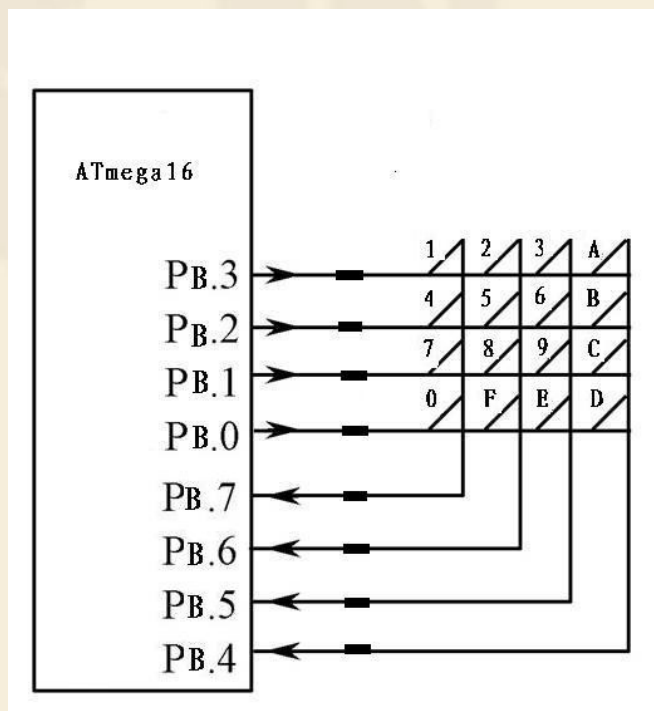
当无键按下时，CPU 处理自己的工作，当有键按下时，产生中断请求，CPU 转去执行键盘扫描子程序，并识别键号。中断扫描方式可以提高 CPU 工作效率。



图中的 4 输入与门用于产生按键中断，其输入端与各列线相连，再通过上拉电阻接至 +5V 电源，输出端接至 16 的外部中断输入端。具体工作如下：当键盘无键按下时，与门各输入端均为高电平，保持输出端为高电平；当有键按下时，端为低电平，向 CPU 申请中断，若 CPU 开放外部中断，则会响应中断请求，转去执行键盘扫描子程序。

五、矩阵键盘的接口实例

ATmega16 单片机的 PB 口作为键盘 I/O 口，键盘的列线接到 PB 口的高 4 位，键盘的行线接到 PB 口的低 4 位。作为输入时要打开内部上拉电阻功能。在按键和 I/O 之间有一排限流电阻。4 根行线和 4 根列线形成 16 个相交点，即：4*4 键盘。



(1) 检测当前是否有键被按下。检测的方法是 PB0 ~ PB3 输出全“ 0 ”，读取 PB4 ~ PB7 的状态，若 PB4 ~ PB7 为全“ 1 ”，则无键闭合，否则有键闭合。

(2) 高四位 PB4 ~ PB7 输出 “ 0 ”，低四位 PB0 ~ PB3 为输入。读取 PB0 ~ PB3 状态。

(3) 把两次输入的值组合成一个 8 位的数据，即按键编码，由于每一行和列的交叉点唯一确定一个按键，所以这个按键编码对应一个唯一的按键。

如果 0111 0111 ，则这个编码对应着十六进制数？

(4) 后可采用算法或查表法将闭合键的行值和列值转换成所定义的键值。

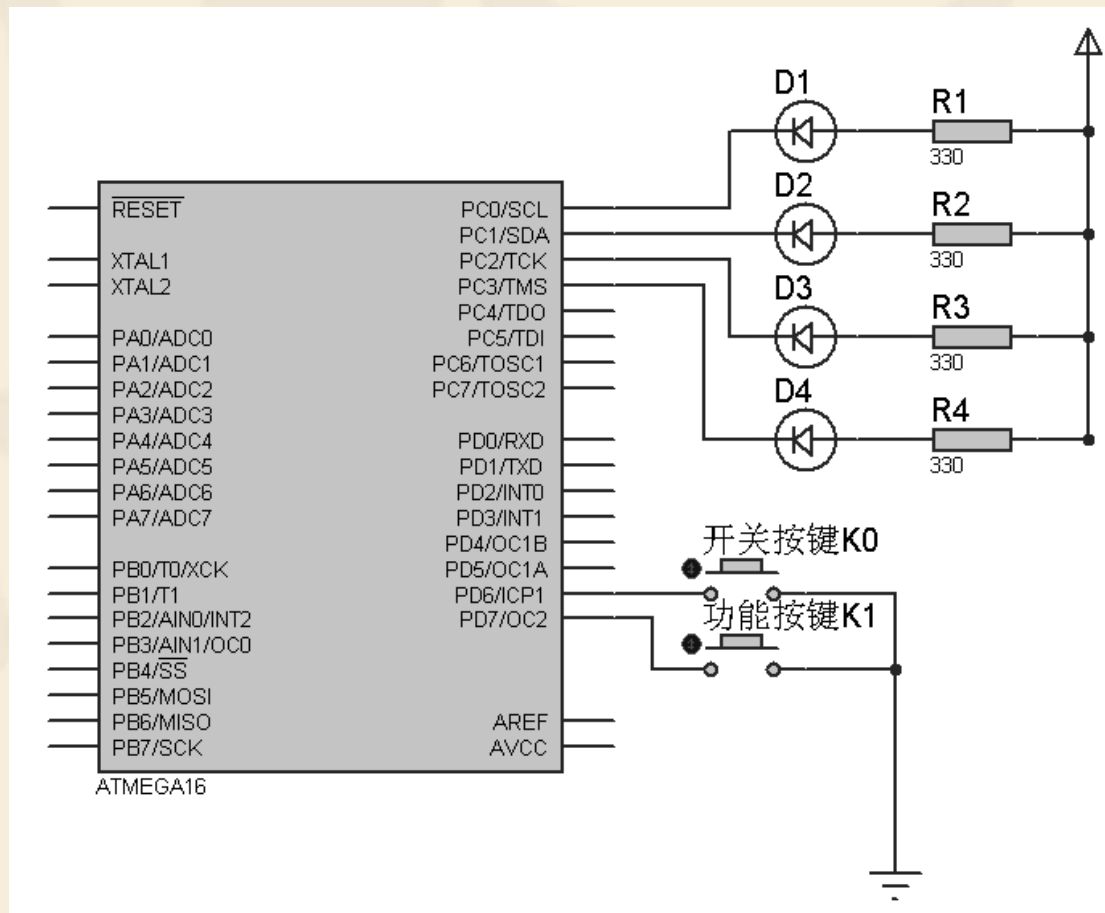
(5) 执行相应的按键对应程序。

任务 2 多功能按键控制 LED 灯

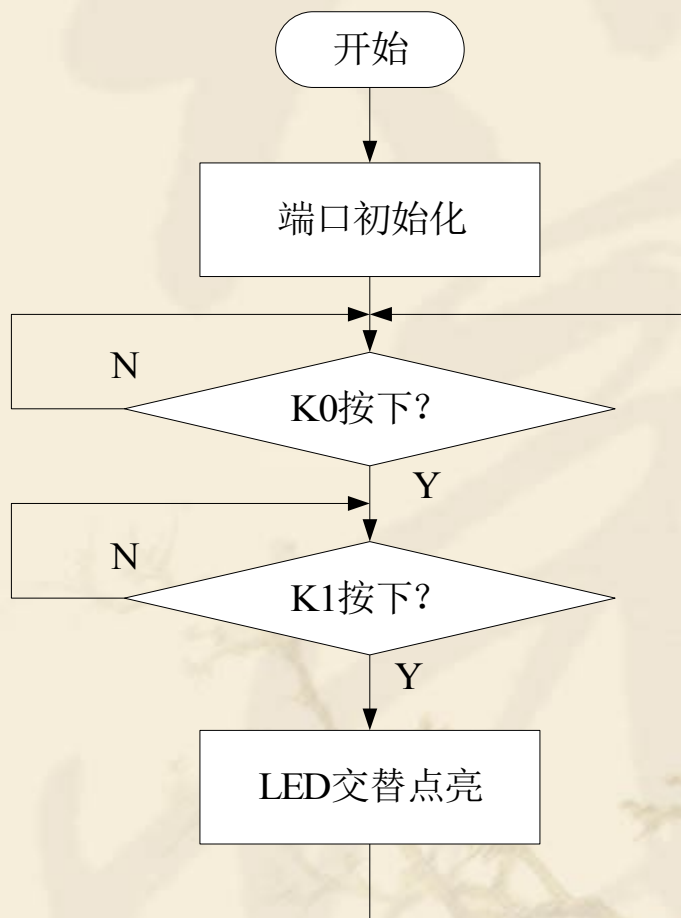
❖ 一、任务要求：

✎ 利用 ATmega16 单片机数字 I/O 口，实现对按键状态的读取，控制霓虹灯的不同显示花样。例如：当按下开关按键 K0 按下 LED 进入可以点亮状态，K1 有效，再按一下 LED 进入熄灭状态，K1 无效；连续按功能按键 K1 可以实现彩灯 D1，D2，D3，D4 的交替点亮，模拟家庭多功能彩灯功能。

❖ 二、硬件 电路



❖ 三、程序设计



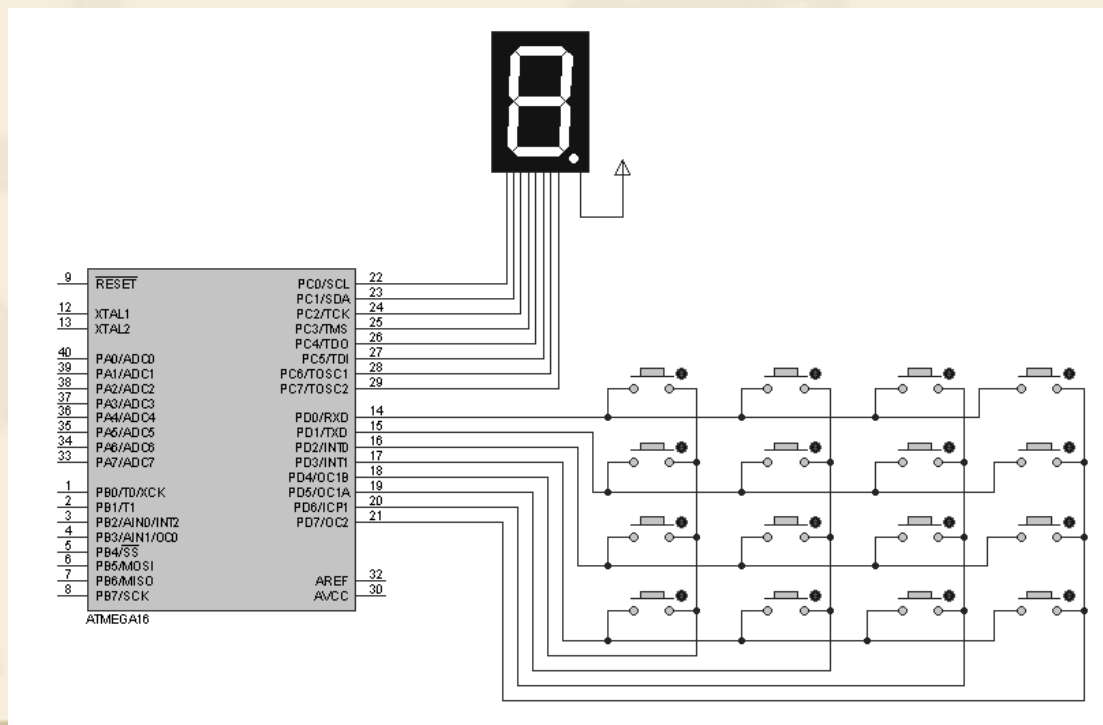
任务 3 矩阵式键盘键值识别

❖ 一、任务要求：

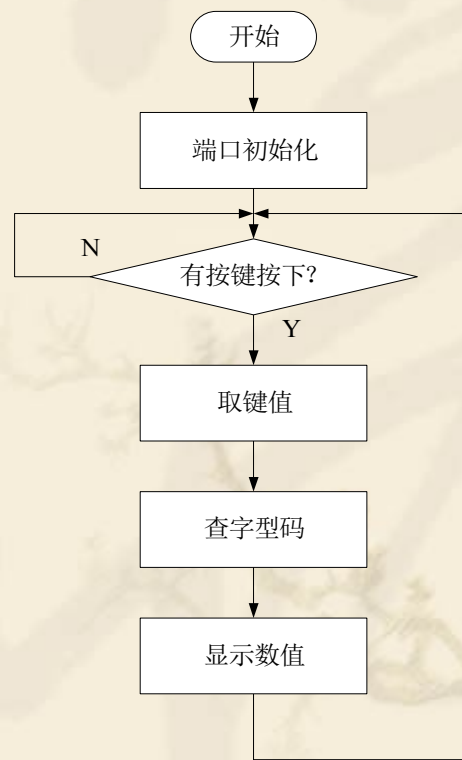
✎ 设计一个 4×4 的矩阵键盘识别显示系统，利用 ATmega16 单片机的 4 个数字 I/O 口作为行线、4 个数字 I/O 端口作为列线，有按键按下数码管显示对应键值，并通过数码管显示按键的键值。

❖ 二、硬件电路

❧ 将十六个按键按照 4×4 的矩阵排列，按键两端按照下图所示分别接到 PD0~PD7，显示部分采用了单个数码管。



❖ 三、程序设计

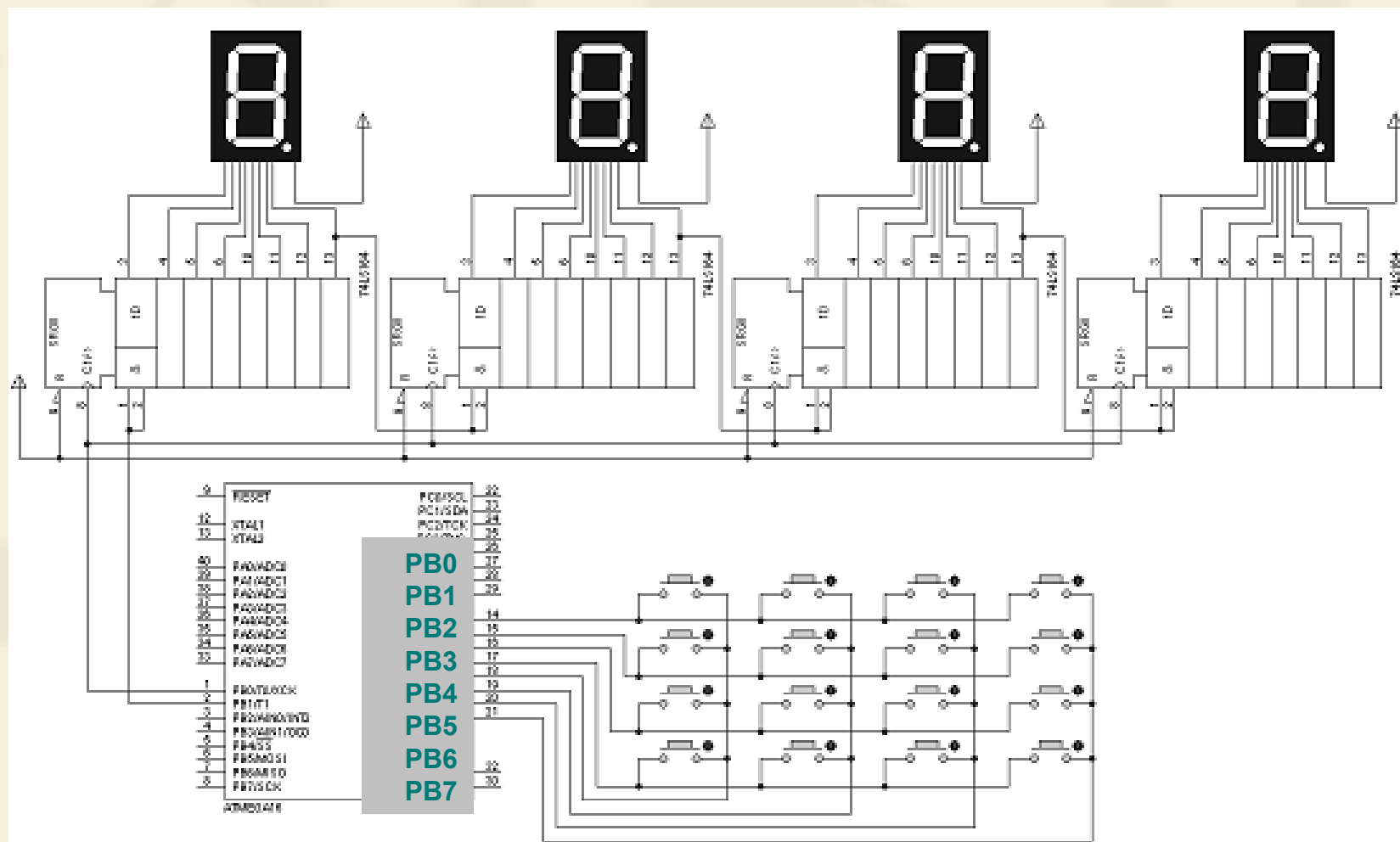


任务 4 电话拨号显示控制

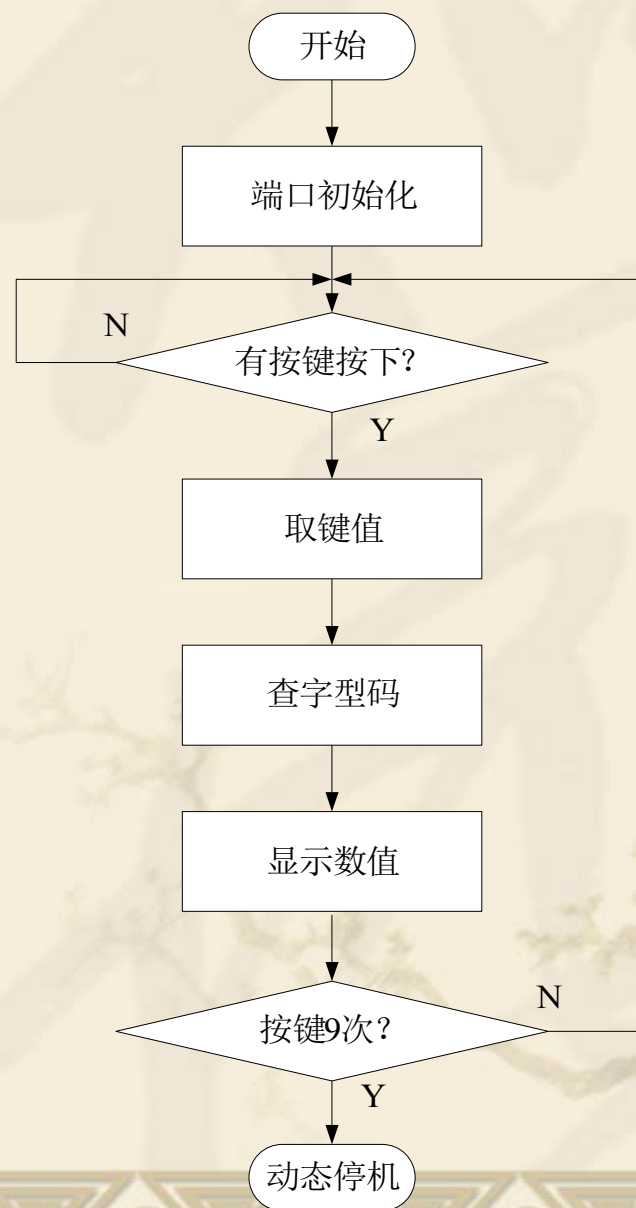
❖ 一、任务要求：

- ❧ 设计一个电话拨号显示系统，利用 ATmega16 单片机作为控制器， 4×4 的矩阵键盘作为电话拨号盘，9 个数码管作为号码显示器。每按下一个号码，在数码管上显示相应的号码同时所有显示内容左移。

二、硬件电路



三、软件设计



【项目实施】

1. 根据元器件清单选择合适的元器件。
2. 根据硬件设计原理图，在万能电路板进行元器件布局，并进行焊接工作。
3. 焊接完成后，重复进行线路检查，防止短路、虚接现象。
4. 在 **AVR Studio** 软件中创建项目，输入源代码并生成 *.hex 文件。
5. 在确认硬件电路正确的前提下，通过 **JTAG** 仿真器进行程序的下载与硬件在线调试。

谢 谢