

# 红外线遥控器专用 IC

## AT8A21 用户手册

版本：V 1.0

日期：2005-7-28

### 1. 上电的地址 (1FFH)

有几种方法的复位都是从 01FFH 地址开始执行的

- 复位后程序从此地址开始执行
- 看门狗复位后程序从此地址开始执行
- 外部复位脚引发的复位程序从此地址开始执行

```

例如:          ORG    01FFH    ; 程序的开始地址为 1FFH
                GOTO  RESET    ; 跳到用户程序的开始地址
                .....
                ORG    00H     ; 从地址 00H 开始写程式
RESET:
                .....
                .....
                RETURN      ; 程序结束

```

### 2. 数据存贮器的定义:

- 数据存贮器可以用来定义变量也可以定义常量

例如:

```

.....
DATA_H    EQU    010H    ; 定义变量 DATA_H 的地址为 010H
DATA_L    EQU    011H    ; 定义变量 DATA_L 的地址为 011H
.....
MAX       EQU    026H    ; 定义常量 MAX 的内容为 06H
MIN       EQU    01FH    ; 定义常量 MIN 的内容为 1FH
.....

```

**注意:** 此处的伪指令“EQU”也可以用“=”来代替, 效果是相同的。

### 3. 定时/计数器的用法:

- AT8A21 有一个 8 位的定时/计数器, 它的时钟源可以来自内部指令周期, 也可以是来自外部 T0CKI 引脚的上升或下降沿触发的信号, 它的定时/计数是在后台执行的, 并不会产生中断, 但是我们可以在不同的时间段去查询定时/计数器是否定时/计数时间到了或是没有到。定时器寄存器的值可以清掉。

定时器的计算公式:  $T = (256 - M) * Pre * Tm / Osc$

其中: M 为定时器预置数, Pre 为 TCC 的分频数

Tm 为 Oscillator periods, Osc 为振荡频率

例如: 要定时 100MS, Pre 选 256, Tm 选 2, Osc 为 455KHz, 则 M 的值应该为  $167 = A7H$

例如:

```

.....
MOVIA     044H        ; 把立即数 4 送累加器 ACC
OPTION    ; 设定 TIMER 的预分频为 32, 时钟源来自内部的指令周期

```

---

```

CLRR    TIMER    ; 清 TIMER 的的值
.....
LOOP:
MOVRR   TIMER,0  ; 将 TIMER 的值送累加器 ACC
XORIA   03H      ; TIMER 的值与 3 比较, TIMER 是否到 3 了?
BTRSS   STATUS,2 ; 判断状态寄存器的 ZF 位是否为 1 (即 ACC=0)?
GOTO    LOOP     ; 若时间没有, 则继续去等待查询
CLRR    TIMER    ; 否则清掉定时器, 程序向下执行
NOP
.....
.....

```

---

#### 4. 如何查数据表:

- 在 AT8A21 中用指令 RETIA 可以用来查数据表:

例如: 要读第 3 行数据表的值到 DATA\_H 中:

```

.....
MOVIA   03H      ; 将立即数 3 送给 ACC
CALL    TABLE   ; 调用查表子程序
MOVAR   DATA_H  ; 将查表得到的值送 DATA_H
.....
TABLE:
ADDAR   PCL,1    ; PC 当前值加上 ACC 的值送给 PCL 去查表
RETIA   02H      ; 当 ACC=0 时把 2 返回给 ACC
RETIA   0FH      ; 当 ACC=1 时把 0FH 返回给 ACC
RETIA   013H     ; 当 ACC=2 时把 013H 返回给 ACC
RETIA   044H     ; 当 ACC=3 时把 0 返回给 ACC
RETIA   06H      ; 当 ACC=4 时把 06H 返回给 ACC
.....

```

程序运行的结果是: DATA\_H 中的值为 44H。

---

#### 5. IR 载波输出:

- AT8A21 有一个 IR 载波发生器输出, 它的输出由 PCON 寄存器的 IRSC 和 IREN 位来控制, IRCYCLE 和 IRDUTY 寄存器可以用来设定 IR 输出的频率及输出载波的比率, IR 的驱动模式分 drive 和 sink 两种方式。

IR 输出频率 = IC 的振荡频率/IRCYCLE

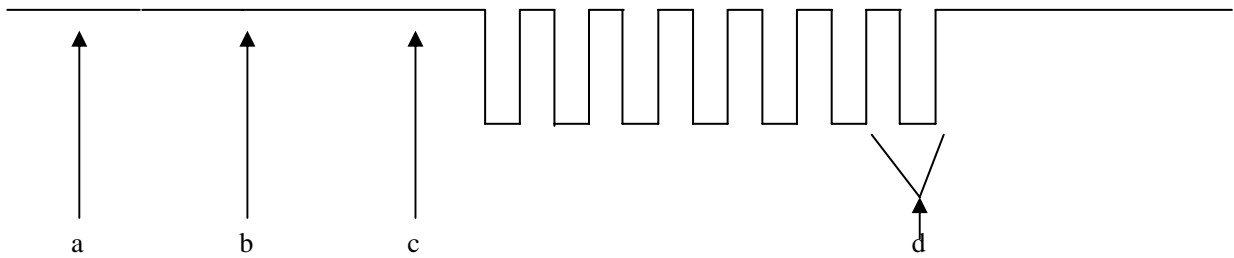
IR 输出载波的比率 = IRDUTY/IRCYCLE

- 例如: 假设 IC 的振荡频率为 455KHz, IRCYCLE=12, IRDUTY=6, IR 驱动模式为 sink 方式。

IR 输出频率 = 455KHz/12 = 37.9KHz

IR 输出载波的比率 = 6/12 = 1/2 (一个周期中高/低电平的占空比)

**注意:** 在设定 IR 输出有效之前必须设定 IOB0 输出高电平, 且 IRDUTY 的值必须小于 IRCYCLE 的值。  
下图为 IR 输出口的波形图:



上图中：  
 a 段：设定 IOB0 输出高电平  
 b 段：设定 IRDUTY, IRCYCLE 及 PCON 寄存器的 IRSC 位  
 c 段：设定 IREN 位=1 即 IR 输出有效，开始输出载波  
 d 段：清掉 IREN 位=0 即 IR 输出无效，载波输出停止

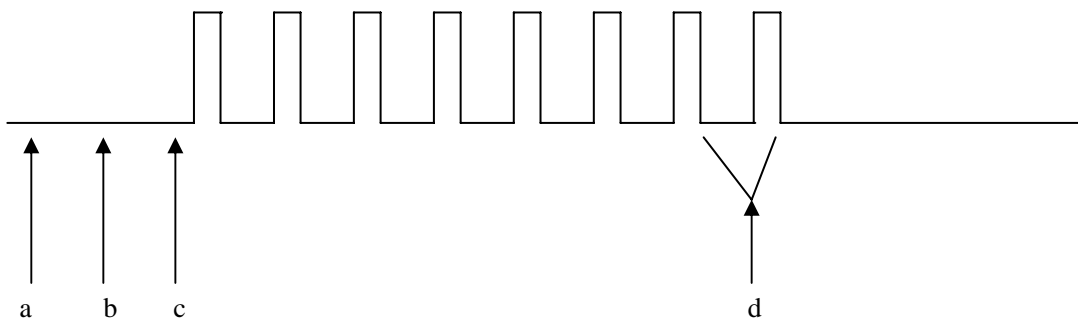
例如：

```

.....
MOVIA 00H
IOST  IOSTB          ; 设定 B 口为输出口
BSR   RB,0          ; B0 口输出高电平
MOVIA 0CH            ; 设定 IRCYCLE 为 12
IOST  IRCYCLE
MOVIA 06H            ; 设定 IRDUTY 为 6
IOST  IRDUTY
MOVIA 07H
IOST  PCON           ; 设定 IR 载波输出有效，载波输出
CALL  DELAY_100MS   ; 延时 100ms
MOVIA 03H           ; 设定 IR 载波输出无效，无载波输出
IOST  PCON
.....
.....

```

- 假设 IC 的振荡频率为 455KHz，IRCYCLE=12，IRDUTY=4，IR 驱动模式为 drive 方式  
 IR 输出频率 =  $455\text{KHz}/12 = 37.9\text{KHz}$   
 IR 输出载波的比率 =  $4/12 = 1/3$  (一个周期中高/低电平的占空比)  
 下图为 IR 输出口的波形图：



上图中: a 段: 设定 IOB0 输出低电平

b 段: 设定 IRDUTY, IRCYCLE 及 PCON 寄存器的 IRSC 位

c 段: 设定 IREN 位=1 即 IR 输出有效, 开始输出载波

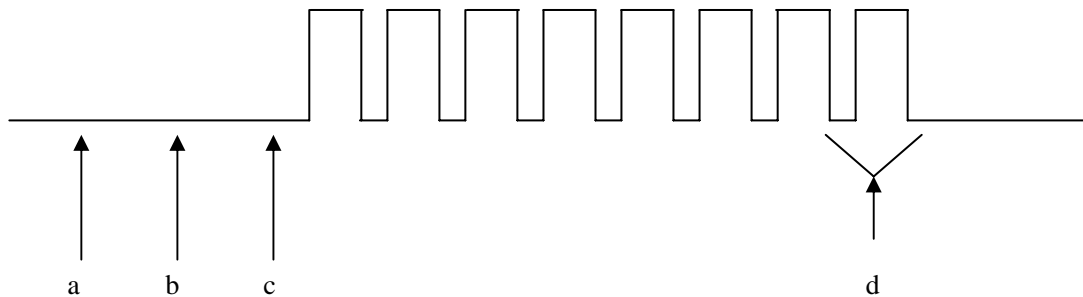
d 段: 清掉 IREN 位=0 即 IR 输出无效, 载波输出停止

➤ 假设 IC 的振荡频率为 455KHz, IRCYCLE = 12, IRDUTY = 8, IR 驱动模式为 drive 方式

IR 输出频率 =  $455\text{KHz}/12 = 37.9\text{KHz}$

IR 输出载波的比率 =  $8/12 = 2/3$  (一个周期中高/低电平的占空比)

下图为 IR 输出口的波形图:



上图中: a 段: 设定 IOB0 输出低电平

b 段: 设定 IRDUTY, IRCYCLE 及 PCON 寄存器的 IRSC 位

c 段: 设定 IREN 位=1 即 IR 输出有效, 开始输出载波

d 段: 清掉 IREN 位=0 即 IR 输出无效, 载波输出停止

**注意:** IR 口平时输出为高电平还是低电平要由 IR 的模式(drive/sink)来决定, 当 IR 的驱动模式是 drive 方式时, IR 平时输出为低电平; 当 IR 的驱动模式是 sink 方式时, IR 平时输出为高电平。

## 6. SLEEP 唤醒模式:

在 SLEEP 模式下, AT8A21 有三种唤醒模式:

- 看门狗溢出唤醒 SLEEP 模式
- PORTA 或 PORTB 输入下降沿唤醒
- 复位脚 (RSTB) 唤醒

外部复位脚和看门狗溢出的复位会使整个 IC 重启, 判断 /PD 和 /TO 位的值是 0 或 1 就可以知道是有哪种方式引发的复位。

对于 PORTA 或 PORTB 输入状态改变引发的 SLEEP 模式唤醒, 程序将从 SLEEP 指令的下一条指令开始执行, 但是在 SLEEP 前, 必须要读 PORTA 或 PORTB 的状态, 当任意一个口唤醒后此脚的唤醒功能将会无效, 因此, 在下次进入 SLEEP 模式之前, 必须重新设定此脚的唤醒功能, 系统的唤醒延时的长短有 18ms/140us/36ms/72ms 四种, 它主要由 SUT0 和 SUT1 位的状态来决定。

例如:

```
.....
MOVIA 0FFH
IOST  IOSTA      ;设定 PORTA 为输入口
MOVIA 00H
```

```

IOST  IOSTB      ;设定 PORTB 为输出口
MOVAR RB          ;PORTB 口输出为低电平
MOVIA 0FFH
IOST  AWUCON     ;设定 A 口可以唤醒 SLEEP 模式
MOVR  RA,1       ;读 A 口的状态
NOP
SLEEP           ;进入 SLEEP 模式
NOP             ;当 A 口唤醒后程序将从此处开始往下执行
.....

```

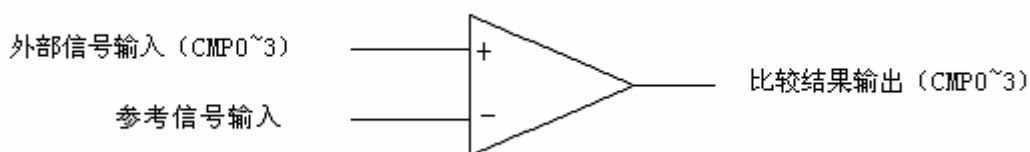
## 7. 比较器方式:

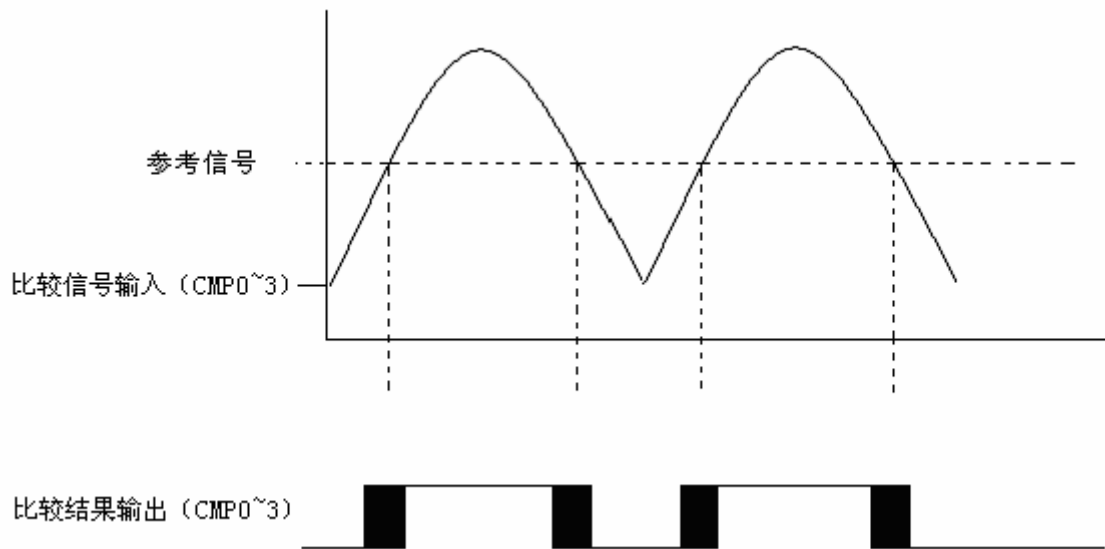
- 比较器有四个分析比较通道，这四个输入脚和 IOA0~IOA3 是公用的，可以由控制寄存器 CMPCON 来决定此四个脚是作为比较输入脚还是普通输入输出脚
- 一个外部信号输入的比较可以通过 VREF0 和 VREF1 两个位来选择内部的四种不同的参考信号，

VREF1	VREF0	比较器参考信号
0	0	1/4 Vdd
0	1	2/4 Vdd
1	0	3/4 Vdd
1	1	VREF 脚 (IOA4/VRFE 脚必须设为输入)

- 当比较输入电平比内部参考信号的电平低时，比较输出的结果是低电平；当比较输入电平比内部参考信号的电平高时，比较输出的结果是高电平。
- 比较输入电平电压的范围是：0V~ (Vdd-0.8V)。
- 比较输出的结果可以通过 IOA0~IOA3 来读取。
- 在 SLEEP 模式下，若比较器是有效的，则比较器仍然是有效的，并且工作电流要比静态电流大，而且假如此信号唤醒 SLEEP 模式有效的的话则通过比较器的输出信号的变化也可以唤醒 SLEEP 模式，一般为了使静态电流尽可能小，要在进入 SLEEP 模式前关闭比较器（通过清 CMPON 的位）。

下图是信号比较器的波形图：





例如: 设 IOA0 口作为比较信号输入口, 比较参考信号为  $1/4V_{dd}$ ,

```

.....
MOVIA    0FFH
IOST     IOSTA    ;设定 A 口为输入脚
MOVIA    041H
MOVAR    CMPCON   ;A0 作为比较信号输入脚, 比较参考信号为 1/4Vdd, 比较器模式有效
MOVR     PORTA, 0 ;读 PORTA 口的结果在 ACC 中
MOVAR    A_RESULT ;再把结果送到 A_RESULT 中
.....

```

## 8. RESET(复位)

AT8A21 有以下几种复位方法:

- 上电复位 (POR)
- 低电压复位 (BOR)
- 复位脚复位
- 看门狗溢出复位

对于不同的复位方式, 可以用状态寄存器的 P 和 T 位的不同来判断是哪种方式引起的复位. 如果没有用到复位脚 (RESET), 为了方便也可以直接将 RESET 脚接到电源 Vdd 脚.

例如:

```

ORG     01FFH
JMP     RESET
.....
.....

```

RESET:

```

BTRSS   STATUS, 3    ;判断 P 是否为 1?
JMP     JUDGE_1
BTRSS   STATUS, 4    ;判断 T 是否为 1?
JMP     WDT_ERROR

```

```

    JMP     BEGIN
JUDGE_1:
    BTRSS  STATUS,4      ;判断 T 是否为 1?
    JMP     WDT_RESET
RSTB_RESET:             ;P=0&T=1→RSTB 脚在 SLEEP 模式下复位
    .....
WDT_ERROR:              ;P=1&T=0→是正常操作中看门狗溢出复位
    .....
WDT_RESET:              ;P=0&T=0→看门狗在 SLEEP 模式下溢出复位
    .....
BEGIN:                  ;P=1&T=1→是上电或低电压复位跳去 BEGIN 开始执行
    .....
    .....

```

## 9. 专用寄存器的用法:

- AT8A21 共有 16 个专用寄存器
  - 操作寄存器:
    - INDF:        间接寻址寄存器

例如: 要把 BANK1 中 A0 和 A1 地址的内容送到地址 012H 和 013H 中去:

```

    .....
A0     EQU    038H      ;内容为 0AH
A1     EQU    039H      ;内容为 012H
B0     EQU    012H
B1     EQU    013H
-----
    .....
MOVR   A0,0           ;将 A0 的内容送 ACC
MOVAR  FSR            ;再把 A0 的地址装入 FSR
MOVR   INDF,0         ;读 INDF 的值到 ACC
MOVAR  B0             ;再将 ACC 的值送给 B0
INCR   FSR,1          ;FSR 地址+1 即 39H 地址
MOVAR  FSR            ;再把 039H 的地址装入 FSR
MOVR   INDF,0         ;读 INDF 的值到 ACC
MOVAR  B1             ;再将 ACC 的值送给 B1
    .....

```

结果是 A0 和 A1 的值分别被送到以 012H 和 013H 为地址的寄存器中.

```

TMR0:   8 位实时计时/计数器寄存器
PCL:    程序计数器/堆栈的低字节寄存器
STATUS: 状态寄存器
FSR:    数据存储器间接寻址指针寄存器
PORTA:  A 口寄存器

```



PORTB: B 口寄存器  
 PORTC: C 口寄存器  
 CMPCON: 比较器控制寄存器

-----  
 ➤ 由 OPTION 或 IOST 指令控制的寄存器:

OPTION: 选项控制寄存器  
 IOSTA: A 口输入输出控制寄存器  
 IOSTB: B 口输入输出控制寄存器  
 PCON: 模式控制寄存器  
 AWUCON: A 口输入状态改变唤醒控制寄存器  
 BWUCON: B 口输入状态改变唤醒控制寄存器  
 IRCYCLE: IR 输出周期控制寄存器  
 IRDUTY: IR 占空比系数控制寄存器

以上其它寄存器的用法详细见 AT8A21 规格书上的说明.

### 10. 看门狗定时器:

看门狗定时器是一个自由运行的定时器, 可以通过设置或清除 WDTE 位的状态来设定它是有效或无效, 一般情况下, 看门狗的溢出时间为 18ms(没有分频), 如果想要看门狗溢出的时间变长, 可以设定不同的分频数改变它的溢出时间, 它的最长溢出时间可以达到 2.3 秒.

### 11. 复位的顺序

当上电复位, 低电压复位, 复位脚复位或看门狗复位被检测到时, 复位的顺序动作如下:

1. 复位锁存器被设为 1, 上电复位定时器和振荡定时器被清掉.
2. 当内部上电复位脉冲完成之后, 上电复位定时器开始计数.
3. 当上电复位定时器溢出后, 振荡开始.
4. 当振荡延时完成之后, 复位锁存器被清掉, 芯片开始工作.

### 12. 振荡结构

AT8A21 共有 4 种振荡模式:

1. 晶振模式
2. 外部时钟输入方式
3. 外部 RC 振荡方式
4. 内部 RC 振荡方式(455KHz)

振荡模式的选择是在烧写的时候由硬件来设定的, 具体参考电路见 AT8A21 规格书.

### 13. 矩阵按键的用法:

对于 AT8A21 来说, 它共有 8 个输入输出脚, 可以由硬件设置上拉有效或无效, 但是只能整体而不能独立设置. 这 8 个脚作为输入脚时可以设置唤醒 SLEEP 模式有效或无效.

例如: 对于矩阵式按键 4\*4 来说, 用 PORTA 和 PORTB 口来做, A0~A3 作为输出口, B0~B3 作为输入口, PORTB 口上拉电阻有效, PORTA 口上拉电阻无效.

```
.....
.....
MOVIA    OFFH
```

---

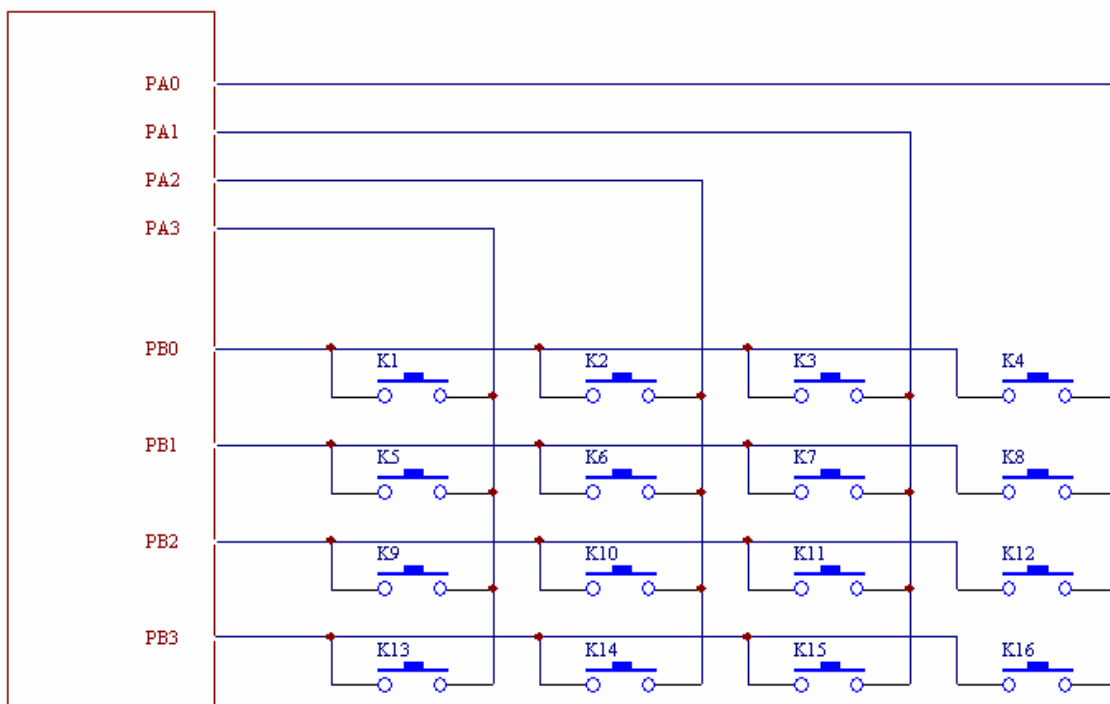
```

IOST      IOSTB      ;PORTB 口作为输入口
MOVIA     00H
IOST      IOSTA      ;设置 PORTA 为输出口,输出低电平
MOVAR     PORTA
.....
.....
MOVR      PORTB,1    ;读 PORTA 的状态保存在 PORTA 寄存器中
BTRSS     PORTB,0
JMP       PORTB_0_LOW
BTRSS     PORTB,1    ;若 PORT0 输入为高则没有按键按下再去判断 PORTA1
.....
.....
PORTB_0_LOW:      ;若 PORT0 口为低电平则 PORTB0 口有按键被按下
MOVIA     0FEH
MOVAR     PORTA      ;PORTA0 输出低电平
NOP
MOVR      PORTB,1    ;读 PORTB 口的状态
MOVIA     0FH
XORAR     PORTB,0    ;PORTB 的输入是否与 0FH 相等?
BTRSS     STATUS,2   ;ZF=1?
GOTO     JUDGE_0
MOVIA     0FDH
MOVAR     PORTA      ;PORTA1 输出低电平
NOP
MOVR      PORTB,1    ;读 PORTB 口的状态
MOVIA     0FH
XORAR     PORTB,0    ;PORTB 的输入是否与 0FH 相等?
BTRSS     STATUS,2
GOTO     JUDGE_1
.....
.....
JUDGE_0:
DECRSZ   KEY_COUNT   ;KEY_COUNT-1=0?
GOTO     PORTB_0_LOW
MOVIA     01H
MOVAR     KEY_NAME,1  ;KEY_NAME=1
.....
.....
JUDGE_1:
.....
.....

```

---

下图为 4\*4 的矩阵按键电路图, PA0~PA3 作为输出口, PB0~PB3 作为输入口:



下图为 7\*8 的矩阵按键电路图, PB 口作为输出, PA 口作为输入:

