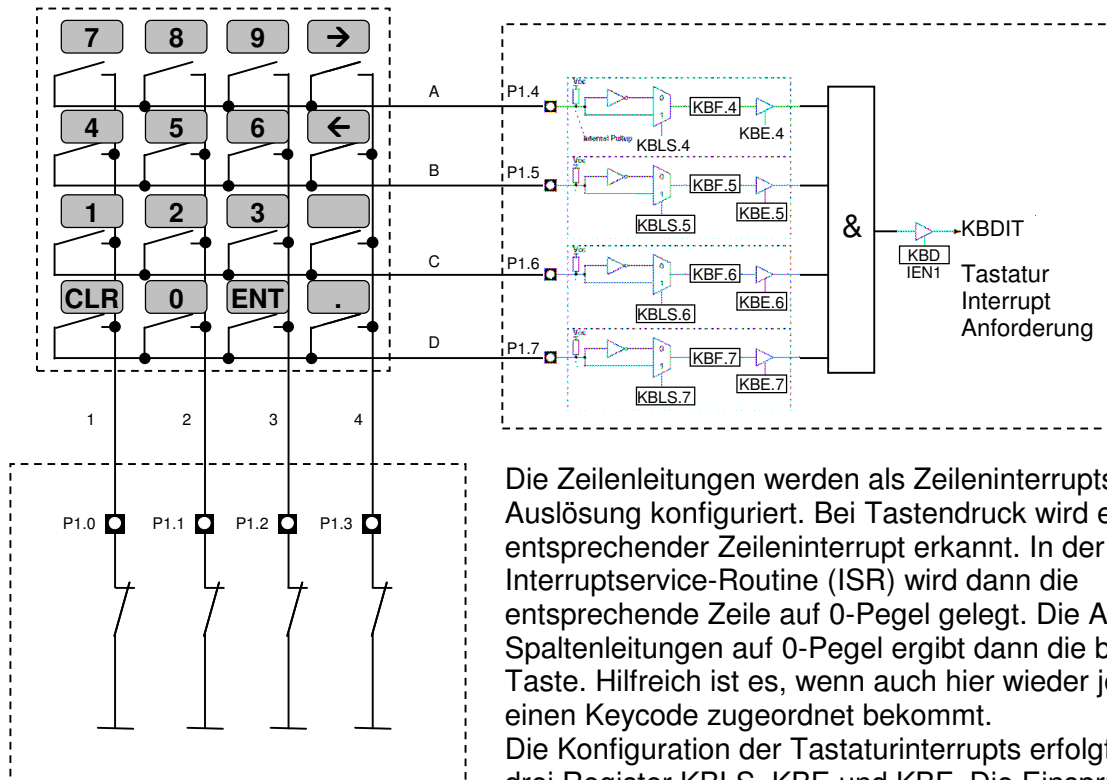
 Friedrich-Ebert-Schule Esslingen	MIKROCONTROLLER	Name:
	4.7.6.1	Matrix-Tastenfeld mit Keyboard-Interrupt

Der AT89C51ED2 (RC2) verfügt über ein Keyboardinterface für Matrixtastaturen. Zum Anschluss geeignet sind alle 8 x n Tastenfelder. Das Keyboardinterface enthält für die Portanschlüsse P1.0 ...P1.7 jeweils eigene Interrupteingänge, die sich alle den selben Interruptvektor (Einsprungadresse) teilen. Im Beispiel ist der Anschluss eines 4 x 4 – Tastenfeldes an Port 1 gezeigt.




Die Zeilenleitungen werden als Zeileninterrupts mit 0-Pegel Auslösung konfiguriert. Bei Tastendruck wird ein entsprechender Zeileninterrupt erkannt. In der Interruptservice-Routine (ISR) wird dann die entsprechende Zeile auf 0-Pegel gelegt. Die Abfrage der Spaltenleitungen auf 0-Pegel ergibt dann die betätigte Taste. Hilfreich ist es, wenn auch hier wieder jede Taste einen Keycode zugeordnet bekommt. Die Konfiguration der Tastaturinterrupts erfolgt über die drei Register KBLs, KBE und KBF. Die Einsprungadresse des Tastaturinterrupts ist 003Bh

7	6	5	4	3	2	1	0	KBLS Keyboard Level Selector Register (9Ch)
KBLS7	KBLS6	KBLS5	KBLS4	KBLS3	KBLS2	KBLS1	KBLS0	Resetwert: 0000 0000b
Keyboard Line Level Selection bit (Auswahl des Auslösepegels für einen Tastatur-Zeileninterrupt) 1: H-Pegel Auslösung 0: L-Pegel Auslösung								

7	6	5	4	3	2	1	0	KBE Keyboard Input Enable Register (9Dh)
KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0	Resetwert: 0000 0000b
Keyboard Line Enable bit (Einzelfreigaben für Tastatur-Zeileninterrupt) 1: Zeilen-Interrupt freigeschaltet! 0: Normale Port-IO (Kein Interrupt!)								

7	6	5	4	3	2	1	0	KBF Keyboard Flag Register (9Eh)
KBF7	KBF6	KBF5	KBF4	KBF3	KBF2	KBF1	KBF0	Resetwert: 0000 0000b
Keyboard Line Flag (Zeigt einen Tastatur-Zeileninterrupt an) 1: Wird bei Erkennung des programmierten Interruptpegels von der Hardware gesetzt. Ein Keyboard-Interrupt wird angefordert, wenn das zugehörige KBE-Bit in Register KBE gesetzt ist. 0: Der Zugriff auf KBF ist nur-lesend. Das Flag wird durch einen Lesezugriff jedoch automatisch zurückgesetzt..								

 Friedrich-Ebert-Schule Esslingen	MIKROCONTROLLER	Name:
4.7.6.2	Matrix-Tastenfeld mit Keyboard-Interrupt	Datum:

Tastatur_interrupt.a51

```

;*****
;* Testprogramm für Tastaturinterrupt, Rahm, 7.3.2007
;*****
    include c51rd2.inc
    include lcd.inc

    extern code init_disp, out_char, set_cursor, text_zeile1, text_zeile3

    Keycode equ    7Fh
    Spalte  equ    7Eh
    ctr     equ    7Dh

    IEN1    data    0B1h
    KBL5    data    9Ch
    KBE     data    9Dh
    KBF     data    9Eh

    code at 0

start:  mov     sp,#080h
        mov     ctr,#0
        ljmp    hauptprogramm

        org     003Bh                ; Einsprungadresse KBD-Interrupt
KBDIT:  mov     KBE,#0                ; Keyboardinterrupt sperren
        mov     a,KBF                ; Tastatur Zeilenflags lesen
        jb     acc.4,ZeileAint        ; Zeilen-Interrupt Erkennung
        jb     acc.5,ZeileBint
        jb     acc.6,ZeileCint
        jb     acc.7,ZeileDint

        mov     p1,#11110000b        ; Spaltenleitungen = 0
        mov     a,KBF                ; DummyRead (setzt Flags zurück
        mov     KBE,#0f0h           ; Kein Interrupt erkannt, dann ITfreigabe
        reti                          ; und wieder ins Hauptprogramm

ZeileAint:
        mov     p1,#11101111b        ; Zeile A (P1.4) = 0
        mov     Keycode,#4
        ajmp    fertigZeile

ZeileBint:
        mov     p1,#11011111b        ; Zeile B (P1.5) = 0
        mov     Keycode,#8
        ajmp    fertigZeile

ZeileCint:
        mov     p1,#10111111b        ; Zeile C (P1.6) = 0
        mov     Keycode,#12
        ajmp    fertigZeile

ZeileDint:
        mov     p1,#01111111b        ; Zeile D (P1.7) = 0
        mov     Keycode,#16

fertigZeile:
        acall   Spaltentest

Ausgabe:
        mov     a,#Z2_Start_Adress
        lcall   set_cursor

        mov     a,Keycode
        mov     dptr,#key_tab
        dec     a
        movc    a,@a+dptr

        lcall   out_char

        mov     p1,#11110000b
        mov     a,KBF                ; DummyRead (setzt Flags zurück)
        mov     KBE,#0f0h           ; KBint freigeben!

        reti

```



```
*****  
Spaltentest:  
    mov     a,p1  
    cpl     a                ; Akku = [x,x,x,x,S4,S3,S2,S1]  
    anl     a,#00001111b    ; Akku = [0,0,0,0,S4,S3,S2,S1]  
    swap    a                ; Akku = [S4,S3,S2,S1,0,0,0,0]  
    mov     Spalte,#4        ; Spaltenzähler = #4  
    clr     c                ; Carry-Flag löschen  
next_Spalte:  
    rlc     a                ; Prüfbit in Carryflag schieben  
    jc     break            ; Taste betätigt? Dann Prüfung erfolgreich!  
                        ; Sonst  
    dec     Keycode         ; Keycode = Keycode - 1  
    djnz   Spalte,next_Spalte  
                        ; Spaltenzähler = Spaltenzähler-1  
break:                                ; sonst, fertig  
  
    ret  
  
*****  
;* Hauptprogramm  
*****  
Hauptprogramm:  
    lcall   init_disp  
    mov     dptr,#text1  
    lcall   text_zeile1  
  
initKBDint:  
    mov     p1,#11110000b    ; Spaltenleitungen = 0  
    mov     KBLs,#00000000b ; L-Pegel Auslösung für Keyboardinterrupt  
    mov     KBE,#11110000b  ; Tastatur-Zeileninterrupts freigeben  
    mov     IEN1,#00000001b ; KeyboardInterrupt freigeben (KBD)  
    setb    EA              ; Generelle Interruptfreigabe  
  
loop:    sjmp    loop        ; Endlosschleife  
  
*****  
key_tab:    ;Keycode 1  2  3  4  
            db     7Eh,'9','8','7'    ; Zeile A  
            ;Keycode 5  6  7  8  
            db     7Fh,'6','5','4'    ; Zeile B  
            ;Keycode 9  10 11 12  
            db     ' ','3','2','1'    ; Zeile C  
            ;Keycode 13 14 15 16  
            db     ' ','E','0','C'    ; Zeile D  
  
text1:    db "Keyboard-Int",0  
  
end
```