

<p>Střední Průmyslová Škola Elektrotechnická Havířov</p>	<p>Protokol do MIT</p>	<p>Třída: 4.C</p>
<p>Zabezpečení budovy</p>		<p>Skupina: 3</p>
		<p>Zpráva číslo: 1</p>
	<p>Dne: 02.010.2006</p>	
	<p>Soupis použitých přístrojů: přípravek s μC 8051 přípravek LCD přípravek s tlačítky pro přerušení</p>	
	<p>Jméno učitele: Ing. Baarová</p>	
	<p>Jméno:</p>	
	<p>Známka:</p>	

ZADÁNÍ:

Napište program pro zabezpečení budovy s použitím přerušení a LC displeje. Program bude hlídat otevřené dveře a okna. Jakékoliv otevřené dveře signalizuje T1 => na LCD vypíše Pozor! dveře. Jakékoliv otevřené okno signalizuje INT1 => na LCD vypíše Pozor! okno. Do 60s musíte stisknout tlačítko T0 jinak se budova po 60s zamkne.

TEORIE:

PŘERUŠENÍ:

Obvody 8051 mají 5 zdrojů přerušení:

	Zdroj přerušení:	Adresa vektoru přerušení:
Vnější přerušení 0 (INT0)	IE0	03h
Přerušení C/T0 (T0)	TF0	0Bh
Vnější přerušení 1 (INT1)	IE1	13H
Přerušení C/T1 (T1)	TF1	1Bh
Přerušení sériového kanálu	RI+TI	23h

OBSLUHA PŘERUŠENÍ:

Je-li jeden z příznaků přerušení nastaven, výběrový cyklus ho najde a přerušovací systém generuje instrukci LCALL vhodnou pro přechod do odpovídajícího obslužného programu přerušení. Hardwarové generování instrukce LCALL proběhne pouze tehdy, není-li blokováno žádnou z těchto podmínek:

- 1) Již se zpracovává přerušení se stejnou nebo vyšší úrovní priority.
- 2) Obvyklý(výběrový) cyklus není v posledním cyklu prováděné instrukce.
- 3) Právě prováděná instrukce je RETI nebo jiná, která zasahuje do registrů IE nebo IP.

Druhá podmínka zajišťuje, že se právě prováděná instrukce zcela dokončí před skokem do některého obslužného programu přerušení.

Po hardwarovém vygenerování instrukce LCALL naplní Program Counter adresou vektoru přerušení, kde jsou obslužné programy. Mikroprocesor vykonává obslužný program dokud nenarazí na instrukci RETI, která ukončí obsluhu přerušení a mikroprocesor se vrátí do hlavního programu kde byl přerušen a pokračuje dál.

IP – REGISTR PRIORITY PŘERUŠENÍ:

prioritu lze nastavit na adrese D8h, což je registr IP(7-0bit, - - - PS PT1 PX1 PT0 PX0)

PS – priorita přerušení od sériového kanálu

PT1 – priorita přerušení C/T1

PX1 – priorita vnějšího přerušení 1 (INT1)

PT0 – priorita přerušení C/T1

PX0 – priorita vnějšího přerušení 0 (INT0)

IE – REGISTR POVOLENÍ PŘERUŠENÍ:

před použitím musíme přerušení povolit na adrese A8h, což je registr IE(7-0bit, EA - - ES ET1 EX1 ET0 EX0)

EA – povoluje nebo zakazuje celý přerušovací systém

ES – povoluje nebo zakazuje přerušení od sériového kanálu

ET1 – povoluje nebo zakazuje přerušení od C/T1

EX1 – povoluje nebo zakazuje vnější přerušení 1

ET0 – povoluje nebo zakazuje přerušení od C/T0

EX0 – povoluje nebo zakazuje vnější přerušení 0

Vnější přerušení INT0 a INT1 se mohou vyvolat buď logickou úrovní nebo změnou log. úrovně. Rozhoduje o tom stav bitů IT0 a IT1 v registru speciálních funkcí TCON.

Přerušení od časovače 0 a časovače 1 se vyvolávají signály TF0 a TF1, které se nastaví do logické úrovně 1, když přetečou odpovídající registry čítačů/časovačů.

Přerušení od sériového portu se generuje logickým součtem OR signálů RI a TI.

SOUBOR BUDOVA.ASM:

;vodice k LCD

LCD_EN bit P2.0

LCD_RW bit P2.1

LCD_RS bit P2.2

org 0

sjmp Init

org 0bh ;adresa preruseni citace/casovace0

call OK ;vypise na LCD budova OK

reti

org 13h ;adresa externiho preruseni 1

mov dptr, #Okna

call Osetreni ;osetreni preruseni

reti

org 1bh ;adresa preruseni citace/casovace1

mov dptr, #Dvere

call Osetreni ;osetreni preruseni

reti

Init:

clr tr0 ;vypnuti citace/casovace0

clr tr1 ;vypnuti citace/casovace1

call ResetLCD

call OK ;vypise na LCD budova OK

mov r1, #0

;osetreni vypisu OK a osetreni citacu

clr f0 ;smazani pro dalsi pouziti

Main:

mov tmod, #01100110b ;nastaveni citacu/casovacu

mov ie, #10001111b ;povoleni preruseni

mov th0, #255

mov tl0, #255

mov th1, #255

mov tl1, #255

setb tr0 ;spusteni citace/casovace0

setb tr1 ;spusteni citace/casovace1

Repeat:

jb f0, View ;osetreni preruseni

jmp Repeat

Osetreni:

setb f0 ;nastaveni, aby mohl provest odpocet a vypis casu

mov r1, #0

call TextLCD

ret

```

View:                                ;odpocet
        clr          et1              ;zakaz preruseni citace/casovace1, aby pri osdoctu
nefungovalo tlacitko dvere
        clr          ex1              ;zakaz externiho preruseni 1, aby pri osdoctu nefungovalo
tlacitko okno
        clr          ex0              ;zakaz externiho preruseni 0, aby pri osdoctu nefungovalo
tlacitko okno

        cjne         r1, #0, Init     ;osetreni vypisu OK a osetreni citacu

        call         Delay01

        mov          r6, #9           ;nastaveni jednotek sekund
        mov          r5, #5           ;nastaveni desitek sekund

        mov          a, #0cbh         ;zapis minut na LCD
        call         RidLCD
        mov          a, #'0'
        call         ZnakLCD

        mov          a, #0ceh         ;zapis jednotek sekund na LCD
        call         RidLCD
        mov          a, r6
        add          a, #'0'
        call         ZnakLCD
        dec         r6

        mov          a, #0cdh         ;zapis desitek sekund na LCD
        call         RidLCD
        mov          a, r5
        add          a, #'0'
        call         ZnakLCD
        dec         r5

        cjne         r1, #0, Init     ;osetreni vypisu OK a osetreni citacu

        call         Delay01

ViewJ:                                ;zapis jednotek sekund na LCD

        mov          a, #0ceh
        call         RidLCD
        mov          a, r6
        add          a, #'0'
        call         ZnakLCD
        call         Delay01
        dec         r6
        cjne         r1, #0, Init     ;osetreni vypisu OK a osetreni citacu
        cjne         r6, #255, ViewJ
        mov          r6, #9

ViewD:                                ;zapis desitek sekund na LCD

        mov          a, #0cdh
        call         RidLCD
        mov          a, r5
        add          a, #'0'
        call         ZnakLCD

```

```

    dec    r5
    cjne   r5, #254, ViewJ

    jmp    Lock           ;vypise budova zamcena

Delay01:
                                ;zpozdeni cca 1s
    mov    r4, #255
    mov    r3, #255
    mov    r2, #14

Delay02:
    djnz   r4, Delay02
    djnz   r3, Delay02
    djnz   r2, Delay02
    ret

OK:
                                ;vypise na LCD budova OK
    mov    dptr, #True
    call   TextLCD

    mov    r1, #1

    ret

Lock:
                                ;vypise budova zamcena
                                ;zastaveni citace/casovace0
                                ;zastaveni citace/casovace1
    clr    tr0
    clr    tr1

    mov    dptr, #Locked
    call   TextLCD

    call   Delay02

    jmp    Lock

True:
    db     ' Budova:   '
    db     '      OK! '

Okna:
    db     ' Pozor! okno '
    db     ' odpocet: 1:00 '

Dvere:
    db     ' Pozor! dvere '
    db     ' odpocet: 1:00 '

Locked:
    db     'Pozor! vetrelec'
    db     ' budova zamcena '

#include<LCDkit.inc>

nop
end

```

ZHODNOCENÍ:

Při psaní psaní tohoto programu si musíme dávat pozor na ukončení přerušení, abychom mohli využít přerušení další. Na začátku programu je nutné přeskočit příkazem `sjmp` adresy přerušení, protože bychom si je přepsali a program by nefungoval jak má. Musíme dát pozor taky při odčítání sekund, abychom napsali správnou podmínku, jinak by program na LCD mohl vypisovat na LCD i ostatní znaky ASCII tabulky, nebo při špatném použití instrukce `djnz` by nedopočítal až do 0:00 ale jen do 0:11 a skončil.