Střední Průmyslová Škola Elektrotechnická Havířov	**Protokol do MIT**	Třída: 4.C
		Skupina: 3
Klimatizace		Zpráva číslo: 3
		Dne: 08.01.2007
		Soupis použitých přístrojů: přípravek s μC 8051 přípravek s LCD přípravek s tlačítky pro přerušení teploměr DS18B20
		Jméno učitele: Ing. Baarová
		Jméno:
		Známka:

ZADÁNÍ:

Napište program, který po stisku tlačítka t0 bude zobrazovat teplotu na LCD. Po stisku tlačítka t1 se spusti klimatizace => napis na LCD, klimatizace se spusti jen tehdy, přesáhne-li teplota 25°C a zároveň stiskneme tlačítko t1. Tlačítko int1 bude sloužit pro ruční vypnutí klimatizace.

TEORIE:

DIGITÁLNÍ TEPLOMĚR DS18B20:

Připojuje se na 1-vodičovou sběrnici a vrací teplotu v rozsahu -55°C až 125°C, teplota se vrací ve 2 bytech, LSB a MSB. Význam bitů u MSB, horních 5-bitů vyjadřuje polaritu, následuje celočíselná hodnota. Význam bitů u LSB, horní 4-bity vyjadřují celočíselnou hodnotu a nižší 4-bity vyjadřují desetinné číslo. Každý teploměr obsahuje vnitřní ROM paměť, ve které je uložen vnitřní 64-bitový kód, který jednoznačně určuje zařízení na sběrnici. Každému zaadresovaní sběrnice musí předcházet reset, kdy master (v našem případě 8051) stáhne datový vodič do logické 0, následně jej uvolní a čeká na odezvu.

VÝZNAMY BITŮ:

S (bit 11 až bit 15) – signalizují znaménko, jsou-li v log 0 => kladná teplota, jsou-li v log 1 => záporná teplota
2^0 až 2^6 (bit 4 až bit 10) – celočíselná část teploty
2^{-4} až 2^{-1} (bit 0 až bit 3) – desetinná část teploty

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	**bit 14**	**bit 13**	**bit 12**	**bit 11**	**bit 10**	**bit 9**	**bit 8**
MS Byte	S	S	S	S	S	2^6	2^5	2^4

PŘÍKAZY PRO DS18B20:

Skip ROM [CCh]:
Zařízení master používá tento příkaz k zaadresování všech zařízení na sběrnici současně bez vyslání nějaké ROM kód informace.

Convert [44h]:
Příkaz inicializuje převod jedné teploty. Následuje převod, výsledek je uložen ve 2-registrech v paměti a DS18B20 se vrátí úsporného režimu.

Read scratchpad [BEh]:
Tento příkaz dovolí, aby master mohl přečíst obsah paměti. Přenos dat začne nejníž významným bitem 0.

SOUBOR TEST.ASM:

```
;digitalni teplomer - testovaci program
ds_data     bit         P0.0

LSB                     data        30h
MSB                     data        31h

stack                   equ         0fh

                        org         0

                        mov         sp, #stack
Init:
                        call        ds_reset

Main:

                        call        ds_reset

                        mov         a, #0cch ;skip rom
                        call        ds_write_byte           ;povel pro prevod teploty
                        mov         a, #044h
                        call        ds_write_byte
                        setb        ds_data ;cekani na prevod teploty
                        jnb         ds_data, $
                        call        ds_reset
                        mov         a, #0cch ;skip rom
                        call        ds_write_byte
                        mov         a, #0beh ;nacteme teplotu
                        call        ds_write_byte

                        call        ds_read_byte
                        mov         LSB, a ;nacte a ulozi nizsi byte z DS18B20
                        call        ds_read_byte
                        mov         MSB, a ;nacte a ulozi vyssi byte z DS18B20

                        mov         P2, LSB
                        mov         P3, MSB

                        jmp         Main

                        #include <ds18b20.inc>

                        nop
                        end
```

SOUBOR KLIMATIZACE.ASM:

```
;LCD P2.4 az P2.7 => datove vodice k LCD
LCD_EN      bit         P2.0
LCD_RW      bit         P2.1
LCD_RS      bit         P2.2

ds_data     bit         P0.0
```

```
LSB                     equ     30h ;zaloha nizsiho bytu z DS18B20
MSB                     equ     31h ;zaloha vyssiho bytu z DS18B20

stack                   equ     0fh ;adresa pro posunuti zasobniku

Temp25      equ         25 ;teplota pri ktere se ma zapnout klimatizace

                        org     0

                        sjmp    Init

                        org     0bh ;adresa vektoru preruseni pro casovac/citac0
                        setb    f0 ;uzivatelsky nastavitelny priznak dame do 1 => v hlavnim programu
je dulezity do
;podminky => jestli je v 1 => bude merit teplotu, jestli bude v 0 => nebude merit teplotu
                        reti

                        org     13h ;adresa vektoru preruseni pro vnejsi preruseni1
                        call    KlimaOff ;vypne klimatizaci
                        reti

                        org     1bh ;adresa vektoru preruseni pro casovac/citac1
                        call    KlimaOn ;zapne klimatizaci
                        reti
Init:
                        mov     sp, #stack ;posunitu zasobniku
                        call    ResetLCD ;reset LCD
                        call    ds_reset ;reset DS18B20
                        mov     tmod, #01100110b ;nastaveni citace0 a citace1 do rezimu 2
                        mov     ie, #10001110b ;povoleni preruseni
                        mov     th0, #0ffh
                        mov     tl0, #0ffh
                        mov     th1, #0ffh
                        mov     tl1, #0ffh
                        setb    tr0 ;spusteni citace0
                        setb    tr1 ;spusteni citace1

                        clr     f0 ;vynulovani uzivatelsky nastavitelneho priznaku
Main:
                        jb      f0, Mer ;podminka jestli bude nebo nebude merit teplotu
                        jmp     Main
Mer:
                        call    MerTemp ;zmeri teplotu a ulozi ji do LSB a MSB
                        call    ConvertTemp ;prevede teplotu na celou cast a desetinnou cast
                        call    LCDTemp ;zobrazi teplotu na LCD

                        jmp     Main
MerTemp:
                        call    ds_reset

                        mov     a, #0cch ;skip rom
                        call    ds_write_byte
                        mov     a, #044h ;povel pro prevod
                        call    ds_write_byte

                        setb    ds_data
```

```
                jnb     ds_data, $ ;ceka na prevod teploty

                call    ds_reset

                mov     a, #0cch ;skip rom
                call    ds_write_byte

                mov     a, #0beh ;nacteme teplotu
                call    ds_write_byte
                call    ds_read_byte
                mov     LSB, a ;nacteni a ulozeni nizsiho bytu LSB DS18B20
                call    ds_read_byte
                mov     MSB, a ;nacteni a ulozeni vyssiho bytu MSB DS18B20

                call    ds_reset

                ret
ConvertTemp:
                mov     a, LSB ;nacteme do akumulatoru nizsi byte teploty
                swap    a ;prohodime vyssi 4-bity s nissimi 4-bity
                anl     a, #00001111b ;maskovani => zustanou nam jen data ktera chceme
                mov     r7, a ;vysledek prevodu LSB je v registru r7

                mov     a, MSB ;nacteme do akumulatoru vyssi byte teploty
                swap    a ;prohodime vyssi 4-bity s nissimi 4-bity
                add     a, r7 ;secteme celou cast s desetinnou casti teploty
                mov     r7, a ;vysledek operace ulozime do registru r7

                mov     a, LSB ;nacteme do akumulatoru nizsi byte teploty
                anl     a, #00001111b ;maskovani => zustanou nam jen data která chceme
                mov     r4, a ;ulozi desetinnou cast do registru r4

                ret
LCDTemp:
                mov     dptr, #ActualTemp ;do registru ukazatele dat dame adresu na to, co
chceme zobrazit na LCD
                call    TextLCD ;vypiseme napis na LCD

                mov     a, r7 ;do akumulatoru dame hodnotu z registru r7
                mov     b, #10 ;do registru B dame hodnotu 10
                div     ab ;podelime registr B akumulatorem
                mov     r6, a ;vysledek deleni => zaloha po deleni
                mov     r5, b ;zbytek po deleni => jednotky

                mov     a, #0c6h
                call    RidLCD
                mov     a, r5
                add     a, #30h
                call    ZnakLCD

                mov     a, r6 ;do akumulatoru dame hodnotu z registru r6
                mov     b, #10 ;do registru B dame hodnotu 10
                div     ab ;podelime registr B akumulatorem
                mov     r6, a ;vysledek deleni => zaloha po deleni
                mov     r5, b ;zbytek po deleni => desitky
```

```
                mov     a, #0c5h
                call    RidLCD
                mov     a, r5
                add     a, #30h
                call    ZnakLCD

                mov     a, r6 ;do akumulatoru dame hodnotu z registru r6
                mov     b, #10 ;do registru B dame hodnotu 10
                div     ab ;podelime registr B akumulatorem
                mov     r6, a ;vysledek deleni => zaloha po deleni
                mov     r5, b ;zbytek po deleni => stovky

                mov     a, #0c4h
                call    RidLCD
                mov     a, r5
                add     a, #30h
                call    ZnakLCD

                mov     a, r4 ;do akumulatoru dame hodnotu z registru r4
                mov     b, #10 ;do registru B dame hodnotu 10
                div     ab ;podelime registr B akumulatorem
                mov     r6, a ;vysledek deleni => zaloha po deleni
                mov     r5, b ;zbytek po deleni => setiny

                mov     a, #0c9h
                call    RidLCD
                mov     a, r5
                add     a, #30h
                call    ZnakLCD

                mov     a, r5 akumulatoru dame hodnotu z registru r4
                mov     b, #10 ;do registru B dame hodnotu 10
                div     ab ;podelime registr B akumulatorem
                mov     r6, a ;vysledek deleni => zaloha po deleni
                mov     r5, b ;zbytek po deleni => setiny

                mov     a, #0c8h
                call    RidLCD
                mov     a, r5
                add     a, #30h
                call    ZnakLCD

                ret
KlimaOff:
                clr     f0 ;vynuluje uzivatelsky nastavitelny priznak, aby se vyplo zobrazovani a
mereni teploty

                mov     dptr, #KlimaVyp ;do registru ukazatele dat dame adresu na to, co
chceme zobrazit
;na LCD
                call    TextLCD ;vypise napis na LCD

                ret
KlimaOn:
                clr     f0
```

```
                mov     a, r7
                mov     b, #100
                div     ab
                mov     a, b ;zbytek po deleni => jednotky

                clr     c
                cjne    a, Temp25, KlimaJC ;porovnavame aktualni teplotu v akumulatoru s konstantou Temp5

KlimaJC:
                jc      KlimaEnd ;jeli C=1 => akumulator byl pri porovnavani mensi nez konstanta Temp25
;klimatizace se nezpusti

                mov     dptr, #KlimaZap ;do registru ukazatele dat dame adresu na to, co chceme zobrazit
;na LCD
                call    TextLCD ;vypise napis na LCD

KlimaEnd:
                ret

ActualTemp:
                db      ' Akt. teplota:  '
                db      '      ,   C    '

KlimaVyp:
                db      ' Klimatizace    '
                db      '       vypnuta '

KlimaZap:
                db      ' Klimatizace    '
                db      '       zapnuta '

                #include<LCDkit.inc>
                #include<ds18b20.inc>

                nop
                end
```

ZHODNOCENÍ:

Jakmile jsem vytvořil funkční testovací program, bylo snadné napsat program pro zobrazení aktuální teploty na LCD po stisku tlačítka. Důležité bylo dát si pozor při psaní obslužného programu pro přerušení, protože je nutné obslužný program pro přerušení ukončit, před tím, než mikroprocesor začne zpracovávat další instrukce. Další krok, který se mi povedl ze zadání splnit je vypnutí klimatizace, zde bylo nejdůležitější vynulovat bit f0, který byl důležitý pro rozhodování mikroprocesoru, jestli má měřit nebo nemá měřit teplotu.

Velký problém ale pro mě bylo vytvořit podprogram, který měl spustit „klimatizaci“ po stisku tlačítka a zároveň překročení teploty 25°C. Myslel jsem si, že podprogram mi bude fungovat, fungoval ale ne tak jak měl. Při stisku tlačítka totiž spustil „klimatizaci“ vždy! Z tohoto vyvozuji, že jsem musel udělat nějakou chybu v podmínce, která určovala, zda se měla spustit „klimatizace“.