

<p>Střední Průmyslová Škola Elektrotechnická Havířov</p>	<p><b>Protokol do MIT</b></p>	<p>Třída: 3.C</p>
<p><b>Digitální teploměr s LCD</b></p>		<p>Skupina: 3</p>
		<p>Zpráva číslo: 5</p>
<p>Dne: 22.05.2006</p>		
<p>Soupis použitých přístrojů: přípravek s <math>\mu</math>C 8051 přípravek s LCD digitální teploměr DS18B20</p>		
<p>Jméno učitele: Ing. Baarová</p>		
<p>Jméno:</p>		
<p>Známka:</p>		

## ZADÁNÍ:

Napište program DIGITÁLNÍ TEPLOMĚR pro mikroprocesor 8051. Na port P0 připojte digitální teploměr DS18B20 a na port P2 připojte LCD. Přečtete teplotu změřenou digitálním teploměrem DS18B20 a zobrazte ji na LCD, nejdříve ale napište testovací program a na port P0 připojte DS18B20 a na porty P2 a P3 připojte LEDky.

## TEORIE:

### DIGITÁLNÍ TEPLOMĚR DS18B20:

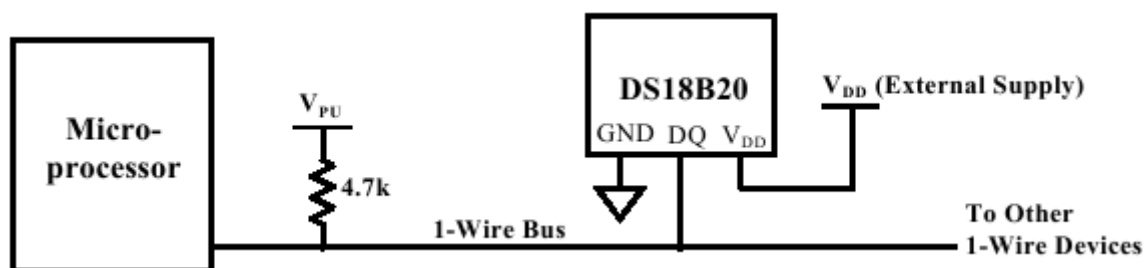
Připojuje se na 1-vodičovou sběrnici a vrací teplotu v rozsahu  $-55^{\circ}\text{C}$  až  $125^{\circ}\text{C}$ , teplota se vrací ve 2 bytech, LSB a MSB. Význam bitů u MSB, horních 5-bitů vyjadřuje polaritu, následuje celočíselná hodnota. Význam bitů u LSB, horní 4-bity vyjadřují celočíselnou hodnotu 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 zaadresování 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	$2^6$	$2^5$	$2^4$

### ZAPOJENÍ DS18B20 K MIKROPROCESORU:



## TESTOVACÍ PROGRAM (TEST.ASM):

```
ds_data    bit        P0.0

LSB        equ        30h ;zaloha nizsiho bytu z DS18B20
MSB        equ        31h ;zaloha vyssiho bytu z DS18B20
stack      equ        0fh

            org        0

            mov        sp, #stack ;posunuti zasobniku
            call       ds_reset ;reset DS18B20

main:

            call       ds_reset ;skip rom

            mov        a, #0cch
            call       ds_write_byte
            mov        a, #44h ;povel pro prevod
            call       ds_write_byte
            setb       ds_data
            jnb        ds_data, $ ;cekame na prevod teploty

            call       ds_reset

            mov        a, #0cch ;skip rom
            call       ds_write_byte

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

            call       ds_reset

            mov        P2,MSB ;vyslani MSB na LEDky na P2
            mov        P3,LSB ;vyslani LSB na LEDky na P3
            jmp        main

#include<ds18b20.inc>

            nop
            end
```

## HLAVNÍ PROGRAM DIGITÁLNÍ TEPLOMĚR (DIG\_TEMP.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

org 0

Init:

mov sp, #stack ;posunuti zasobniku

call ResetLCD ;reset LCD

mov DPTR, #Temp ;vypis zakladni obrazovky s 0 teplotou

call TextLCD

call ds\_reset ;reset DS18B20

Main:

call RWTemp ;cteni a zapis teploty do pameti

call LSBRotate ;format nizsiho bytu

call MSBRotate ;format vyssiho bytu + secteni obou kladnych bytu

call DesTemp ;format desetinne casti teploty

call LCDDesTemp ;vypis desetinne casti teploty na LCD

call LCDCelaTemp ;deleni bytu a vypis cele casti teploty na LCD

jmp Main

RWTemp:

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, \$ ;cekame na prevod teploty

call ds\_reset

mov a, #0cch ;skip rom

call ds\_write\_byte

mov a, #0beh ;nacteme si teplotu

call ds\_write\_byte

call ds\_read\_byte

mov LSB, a ;nacteni a ulozeni nizsiho bytu LSB z DS18B20

call ds\_read\_byte

mov MSB, a ;nacteni a ulozeni vyssiho bytu MSB z DS18B20

call ds\_reset

ret

```
LSBRotate: ;format nizsiho bytu
            mov     a, LSB
            swap    a ;prohozeni 4-vyssich a 4-nizsich bitu
            anl     a, #00001111b ;maskovani
            mov     r6, a

            ret
```

```
MSBRotate: ;format vyssiho bytu
            mov     a, MSB
            swap    a ;prohozeni 4-vyssich a 4-nizsich bitu
            add     a, r6 ;secteni MSB a LSB
            mov     r6, a ;vysledek souctu je v r6

            ret
```

```
DesTemp:
            mov     a, LSB
            anl     a, #00001111b ;makovani
            mov     r5, a ;desetinna cast teploty je v r5

            ret
```

```
LCDDesTemp: ;vypis desetinne casti teploty na LCD
            cjne    r5, #0, LCDDes01
            mov     DPTR, #Temp
            call    TextLCD

            ret
```

```
LCDDes01: ;vypis desetinne casti teploty na LCD
            cjne    r5, #1, LCDDes02
            mov     DPTR, #Des01
            call    TextLCD

            ret
```

```
LCDDes02: ;vypis desetinne casti teploty na LCD
            cjne    r5, #2, LCDDes03
            mov     DPTR, #Des02
            call    TextLCD

            ret
```

```
LCDDes03: ;vypis desetinne casti teploty na LCD
            cjne    r5, #3, LCDDes04
            mov     DPTR, #Des03
            call    TextLCD

            ret
```

```
LCDDes04: ;vypis desetinne casti teploty na LCD
            cjne    r5, #4, LCDDes05
            mov     DPTR, #Des04
            call    TextLCD
```

ret

LCDDes05: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #5, LCDDes06
mov     DPTR, #Des05
call   TextLCD
```

ret

LCDDes06: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #6, LCDDes07
mov     DPTR, #Des06
call   TextLCD
```

ret

LCDDes07: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #7, LCDDes08
mov     DPTR, #Des07
call   TextLCD
```

ret

LCDDes08: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #8, LCDDes09
mov     DPTR, #Des08
call   TextLCD
```

ret

LCDDes09: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #9, LCDDes10
mov     DPTR, #Des09
call   TextLCD
```

ret

LCDDes10: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #10, LCDDes11
mov     DPTR, #Des10
call   TextLCD
```

ret

LCDDes11: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #11, LCDDes12
mov     DPTR, #Des11
call   TextLCD
```

ret

LCDDes12: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #12, LCDDes13
mov     DPTR, #Des12
call   TextLCD
```

ret

LCDDes13: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #13, LCDDes14
mov     DPTR, #Des13
call    TextLCD
```

ret

LCDDes14: ;vypis desetinne casti teploty na LCD

```
cjne    r5, #14, LCDDes15
mov     DPTR, #Des14
call    TextLCD
```

ret

LCDDes15: ;vypis desetinne casti teploty na LCD

```
mov     DPTR, #Des15
call    TextLCD
```

ret

LCDCelaTemp: ;deleni bytu a jeho vypis na LCD

```
mov     a, r6
mov     b, #10
div     ab
mov     r7, a ;zaloha cele casti po deleni
mov     r5, b ;r5 -> jednotky
```

```
mov     a, #0C7h ;vypis jednotek
call    RidLCD
mov     a, r5
add     a, #030h
call    ZnakLCD
```

```
mov     a, r7
mov     b, #10
div     ab
mov     r7, a ;zaloha cele casti po deleni
mov     r5, b ;r5 -> desitky
```

```
mov     a, #0C6h ;vypis desitek
call    RidLCD
mov     a, r5
add     a, #030h
call    ZnakLCD
```

```
mov     a, r7
mov     b, #10
div     ab
mov     r7, a ;zaloha cele casti po deleni
mov     r5, b ;r5 -> stovky
```

```
mov     a, #0C5h ;vypis stovek
call    RidLCD
```

```

mov     a, r5
add     a, #030h
call    ZnakLCD

ret

Temp:   db      ' Teplota:  '
        db      ' 0,0000 C '
Des01:  db      ' Teplota:  '
        db      ' 0,0625 C '
Des02:  db      ' Teplota:  '
        db      ' 0,1250 C '
Des03:  db      ' Teplota:  '
        db      ' 0,1875 C '
Des04:  db      ' Teplota:  '
        db      ' 0,2500 C '
Des05:  db      ' Teplota:  '
        db      ' 0,3125 C '
Des06:  db      ' Teplota:  '
        db      ' 0,3750 C '
Des07:  db      ' Teplota:  '
        db      ' 0,4375 C '
Des08:  db      ' Teplota:  '
        db      ' 0,5000 C '
Des09:  db      ' Teplota:  '
        db      ' 0,5625 C '
Des10:  db      ' Teplota:  '
        db      ' 0,6250 C '
Des11:  db      ' Teplota:  '
        db      ' 0,6875 C '
Des12:  db      ' Teplota:  '
        db      ' 0,7500 C '
Des13:  db      ' Teplota:  '
        db      ' 0,8125 C '
Des14:  db      ' Teplota:  '
        db      ' 0,8750 C '
Des15:  db      ' Teplota:  '
        db      ' 0,9375 C '

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

nop
end

```

### **ZHODNOCENÍ:**

Tato úloha byla celkem jednoduchá jelikož jsme už uměli pracovat s LCD z minulé úlohy. Asi nejtěžší pro mne bylo přečtení dat z teploměru, menší problém jsem měl i s maskováním, ale ten jsem odhalil při krokování programu.

Úloha by se dala rozšířit, mohli bychom zapojit více teploměrů a pak se tlačítkem přepínat mezi obrazovkami, která by obsahovala číslo čidla a změřenou teplotu, nebo bychom mohli ovládat klimatizaci, při překročení určité teplotní hranice by se klimatizace zapnula a při poklesu pod hranici by se vypnula.