

```
1: // LCD module connections
2: sbit LCD_RS at RB4_bit;
3: sbit LCD_EN at RB5_bit;
4: sbit LCD_D4 at RB0_bit;
5: sbit LCD_D5 at RB1_bit;
6: sbit LCD_D6 at RB2_bit;
7: sbit LCD_D7 at RB3_bit;
8:
9: sbit LCD_RS_Direction at TRISB4_bit;
10: sbit LCD_EN_Direction at TRISB5_bit;
11: sbit LCD_D4_Direction at TRISB0_bit;
12: sbit LCD_D5_Direction at TRISB1_bit;
13: sbit LCD_D6_Direction at TRISB2_bit;
14: sbit LCD_D7_Direction at TRISB3_bit;
15: // End LCD module connections
16:
17: // Set TEMP_RESOLUTION to the corresponding resolution of used DS18x20 sensor:
18: // 18S20: 9 (default setting; can be 9,10,11,or 12)
19: // 18B20: 12
20: const unsigned short TEMP_RESOLUTION = 9;//9;
21: char *text = "000.0000";
22: unsigned temp,tempi; // -původní deklarace
23: //unsigned int temp;
24: void Display_Temperature(unsigned int temp2write) {
25:     const unsigned short RES_SHIFT = TEMP_RESOLUTION - 8;
26:     char temp_whole;
27:     unsigned int temp_fraction;
28:     unsigned char Znak;
29:     Znak = 1; //Hodnota 1 při kladné teplotě
30:
31:     // Check if temperature is negative
32:     if (temp2write & 0x8000)
33:     {
34:         Znak = 0; //Hodnota nula při záporné teplotě
35:         temp2write = ~temp2write + 1;
36:     }
37:
38:     // Extract temp_whole
39:     temp_whole = temp2write >> RES_SHIFT ;
40:
41:     // Convert temp_whole to characters
42:     if (temp_whole/100)
43:         text[0] = temp_whole/100 + 48;
44:     else
45:         text[0] = ' ';
46:
47:     text[1] = (temp_whole/10)%10 + 48; // Extract tens digit
48:     //Potlačení zobrazení zbytečných nul
49:     if ((text[0]) == ' ')
50:     {
51:         if ((text[1]) == 48)
52:         {
53:             text[1] = ' ';
54:         }
55:     }
56:     //Zobrazení záporného znaménka
57:     if (Znak == 0)
58:     {
59:         text[0] = '-';
60:     }
61:
62:     text[2] = temp_whole%10 + 48; // Extract ones digit
```

```

63:
64: // Extract temp_fraction and convert it to unsigned int
65: temp_fraction = temp2write << (4-RES_SHIFT);
66: temp_fraction &= 0x0F;
67: temp_fraction *= 625;
68:
69: // Convert temp_fraction to characters
70: text[4] = temp_fraction/1000 + 48; // Extract thousands digit
71: text[5] = (temp_fraction/100)%10 + 48; // Extract hundreds digit
72: text[6] = (temp_fraction/10)%10 + 48; // Extract tens digit
73: text[7] = temp_fraction%10 + 48; // Extract ones digit
74:
75: // Print temperature on LCD
76: Lcd_Out(2, 5, text);
77: }
78:
79: display(unsigned int temp2write) {
80: unsigned int i,x;
81: x=0;
82: //temp << 1;
83: for ( i = 16; i > 0; i--) {
84: //Lcd_Ch(1,x++, '0');
85: if((temp & (1<< i)) == 0)Lcd_Ch(1,x++, '0'); else Lcd_Ch(1,x++, '1');
86: }
87: }
88:
89:
90: void main() {
91: //ANSEL = 0; // Configure AN pins as digital
1 I/O
92: //ANSELH = 0;
93: //C1ON_bit = 0; // Disable comparators
94: //C2ON_bit = 0;
95: TRISA1_bit = 1; //uprava tlacitko
96: CMCON = 0x07;
97: Lcd_Init(); // Initialize LCD
98: Lcd_Cmd(_LCD_CLEAR); // Clear LCD
99: Lcd_Cmd(_LCD_CURSOR_OFF); // Turn cursor off
100: Lcd_Out(1, 1, " Temperature: ");
101: // Print degree character, 'C' for Centigrades
102: Sound_Init(&PORTA, 0);
103: Sound_Play(980, 1000);
104: Lcd_Ch(2,13,223); // Different LCD displays have di
// Different char code for degree
105: // If you see greek alpha letter
// try typing 178 instead of 223
106:
107: Lcd_Ch(2,14,'C');
108:
109: //--- Main loop
110: do {
111: //--- Perform temperature reading
112: Ow_Reset(&PORTA, 2); // Onewire reset signal
113: Ow_Write(&PORTA, 2, 0xCC); // Issue command SKIP_ROM
114: Ow_Write(&PORTA, 2, 0x44); // Issue command CONVERT_T
115: Delay_ms(1000); //Delay_us(10000); // puvodne 120
116:
117: Ow_Reset(&PORTA, 2);
118: Ow_Write(&PORTA, 2, 0xCC); // Issue command SKIP_ROM
119: Ow_Write(&PORTA, 2, 0xBE); // Issue command READ_SCRATCHPAD
120:

```

```
121:     temp = Ow_Read(&PORTA, 2);
122:     tempi = temp;
123:     temp = (Ow_Read(&PORTA, 2) << 8) + temp;
124:
125:     //--- Format and display result on Lcd
126:     if (Button(&PORTA, 1, 1, 1)){
127:         display(tempi);
128:     }
129:     Display_Temperature(temp);
130:     Delay_ms(500);
131: } while (1);
132: }
```