

```
/* ATB_B1_Temp_1.c Created: 19.10.2014 20:02:09 Author: AS+Timo G. */
```

```
#include <stdbool.h>
#include <avr/pgmspace.h>
#include "main.h"
#include <util/delay.h>
#include "i2clcd.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include "util/delay.h"
#include "avr/interrupt.h"

#define lm75_r 0x91 // Leseadresse des LM75
uint8_t msb_temp; // Oberes Temperatur-Byte
uint8_t lsb_temp; // Unteres Temperatur-Byte
uint16_t temp_wrd; // Ganzes Temperatur-Wort
uint8_t ret; // Kontrollvariable für I2C Kommunik.
uint8_t x; // X-Position der Kommastelle
signed char temperatur; // Variable m. Vorzeichen für die Temp.berechnung
char Buffer[20]; // Umwandlungs-Variable für LCD Anzeige
void temperfassung(void) // Unterprogramm Temperaturerfassung
{
    ret = i2c_start(lm75_r); // Start Lesen des LM75
    if (ret == 0) // Wenn LM75 ein OK sendet...
    { // ...speichere oberes Bit
        msb_temp = i2c_readAck(); // ...speichere unteres Bit
        lsb_temp = i2c_readNak();
    }
    else // Fehlererkennung
    { // Wenn LM75 kein OK sendet
        lcd_command(LCD_CLEAR); // Leere Display
        lcd_printlc(1,13,"READ"); // "Lesevorgang"
        lcd_printlc(2,13,"NOK"); // "Nicht OK (NOK)"
    }
}

int main(void) // Start Hauptprogramm
{
    cli(); // Interrupts deaktiviert
    i2c_init(); // Starte I2C Bus
    lcd_init(); // Starte I2CLCD

    // Display Befehle - Display ein, Cursor aus, Blinken aus
    lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);
    lcd_command(LCD_CLEAR); // Leere Display
    _delay_ms(2); // Warte 2ms
    lcd_printlc(1,6,"Boxtec"); // Zeile 1
    lcd_printlc(2,2,"Display Modul 1"); // Zeile 2
    lcd_printlc(3,2,"und Temp-Modul"); // Zeile 3
    lcd_printlc(4,2,"(achim S.+TG)"); // Zeile 4
    _delay_ms(5000); // Warte 5000ms
}
```

```

lcd_command(LCD_CLEAR);           // Lösche Display
_delay_ms(2);                     // Warte 2ms
while(1)                           // Start while Schleife
{
    temperfassung();              // Unterprg Temp.auslesung
    temp_wrd = (msb_temp << 8 | lsb_temp); // Zusammensetzung von o. & u. Byte
    temperatur = msb_temp;         // Umwandlung in ein Signed Char

    // Nachkommastelle erkennen
    if (temperatur <= -100) {x = 5;} // X-Pos. bei <= -100°C
    else if (temperatur <= -10) {x = 4;} // X-Pos. bei <= -10°C
    else if (temperatur <= 9) {x = 3;} // X-Pos. bei <= 9°C
    else if (temperatur <= 99) {x = 4;} // X-Pos. bei <= 99°C
    else {x = 5;} // X-Pos. bei > 99°C

    if (temp_wrd & (1<<7))
    {
        lcd_printlc(3,x,".5"); // ,5 an Pos x schreiben
    }
    else
    {
        lcd_printlc(3,x,".0"); // ,0 an Pos x schreiben
    }
    if (temp_wrd & (1<<15))
    {
        // Wenn Minus-Grade angezeigt werden
        if (temp_wrd & (1<<7))
        {
            // Nachkommastelle erkennen
            temperatur = temperatur +1;
        }
        itoa(temperatur,Buffer, 10 ); // Variable umwandeln...
        lcd_printlc(3,1,Buffer); // ...und anzeigen
    }
    else
    {
        lcd_printlc(3,1,"+"); // Ausgabe „ + „
        itoa(temperatur, Buffer, 10 );
        lcd_printlc(3,2,Buffer);
    }
    lcd_printlc(1,1,"Temperatur ist:"); // Ausgabe Text
    _delay_ms(100); // Pause
}
}

```