```
/*Programm zum programmieren des DS1821 als Thermostat.
 * Die Ein- und Ausschalttemperatur ist einzeln festgelegt in den Definitionen.
 * Die Definition "zuruecksetzen" muss "0" sein, wenn der IC als Thermostat programmiert werden soll.
Bei einer "1" wird der IC in den Ursprungsmodus zurueckgesetzt, und kann neu programmiert werden,
oder als Temperatursensor verwandt werden.

Prozessor: ATTINY85V e-Fuse FFh; h-Fuse DFh; l-Fuse EDh; lock 3Fh. Quarz 3,6864MHz
Version 0 6/17 */

#define F_CPU 3686400UL

#include <stdlib.h>
#include <avr/eeprom.h>
#include <util/delay_basic.h>
#include <util/delay.h>

//Geräte LED
#define Spannung 2
#define Spannung_an (PORTB |= (1<<Spannung))
#define Spannung_aus (PORTB &= ~(1<<Spannung))
#define Sensor 1
#define Sensor_Ausgang_L (PORTB &= ~(1<<Sensor))    //Sensor auf Low
#define Sensor_Ausgang_H (PORTB |= (1<<Sensor)) //Sensor auf High
#define Sensor_Pin_lesen (DDRB &= ~(1<<Sensor)) //Sensor auf Eingang
#define Sensor_Pin_schreiben (DDRB |= (1<<Sensor))  //SEnsor auf Ausgang
#define Sensor_Eingang (PINB & (1<<Sensor))
//Ende Geräte

#define temp_puls(tL,tH) \
    { \
    DDRB |= (1<<Sensor); \
    PORTB &= ~(1<<Sensor); \
    _delay_us(tL); \
    DDRB &= ~(1<<Sensor); \
    _delay_us(tH); \
    }
#define schreibe_TH 0x01
#define schreibe_TL 0x02
#define lese_TH 0xA1
#define lese_TL 0xA2
#define schreibe_Status 0x0C
#define lese_Status 0xAC
#define lese_temperatur 0xAA
#define start_konvertierung 0xEE
#define stop_konvertierung 0x22

#define zuruecksetzen 0 //wenn "1" dann wird in Normalmodus zurückgesetzt; bei "0" Thermostat
#define Einschalttemperatur 27
#define Ausschalttemperatur 29

//Globale Variable im EEPROM
uint8_t ee_Titel[] EEMEM =   {"Programmierung DS1821"};
uint8_t ee_Version[] EEMEM = {"0 6/17 pkr"};



uint8_t reset()
{
    //Resetimpuls (1: erfolgreich, 0: fehlgeschlagen)
    uint8_t ok;

   temp_puls(600,100)
    if(Sensor_Eingang == 0) ok = 1; else ok = 0;
    _delay_us(410);
    return(ok);
}

void Byte_schreiben(uint8_t Byte)
{
    uint8_t n;

    for (n = 0; n < 8; n++)
    {
        if (Byte & (1<<n))
            temp_puls(6,60)
```

```
 74         else
 75             temp_puls(60,6)
 76     }
 77     return;
 78 }
 79
 80 uint8_t Byte_lesen()
 81 {
 82     uint8_t n;
 83     uint8_t byte = 0;
 84
 85     for (n = 0; n < 8; n++)
 86     {
 87         temp_puls(2,9)
 88         if (Sensor_Eingang)
 89         {
 90             byte |= (1<<n);
 91         }
 92         _delay_us(60);  //warte zur Erholung t(slot)
 93     }
 94     return(byte);
 95 }
 96
 97 void Temperatur_programmieren(uint8_t unten, uint8_t oben)
 98 {
 99     if (reset())
100     {
101         Byte_schreiben( schreibe_TH );
102         Byte_schreiben( oben );
103         reset();
104         Byte_schreiben( schreibe_TL );
105         Byte_schreiben( unten );
106         reset();
107         Byte_schreiben( schreibe_Status );
108         Byte_schreiben( 0x06 );  //Thermostat modus, Ausgang aktiv high
109     }
110     return;
111 }
112
113 void Thermostat_modus_verlassen()
114 {
115     uint8_t n;
116     Spannung_an;
117     Sensor_Pin_schreiben;
118     Sensor_Ausgang_H;
119     Spannung_aus;
120     _delay_us(5);
121     Sensor_Ausgang_L;
122     _delay_us(5);
123
124     for (n=0; n<16; n++)
125     {
126         Sensor_Ausgang_L;
127         _delay_us(5);
128         Sensor_Ausgang_H;
129         _delay_us(5);
130     }
131     Spannung_an;
132     return;
133 }
134
135
136 void Ports_init()
137 {
138     DDRB |= (1<<Spannung) | (1<<Sensor); //Sensor und Spannungsport als Ausgang
139 }
140
141 int main()
142 {
143
144     Ports_init();
145        if (zuruecksetzen)
146        {
147             Thermostat_modus_verlassen();
148        }
```

```
149         else
150         {
151              Spannung_an;
152              _delay_ms(20);
153              Temperatur_programmieren(Einschalttemperatur, Ausschalttemperatur);
154         }
155
156
157     while(1)
158     {
159
160     }
161
162 return (0);
163 }
164
165
166
```