

Bilaga 1. Programmeringskod

```
/*
 * berra.c
 *
 * Created: 2015-04-07 15:31:40
 * Author: digpi02
 */
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
void disp_init();
void disp_clear();
void write_data(char val);
void write_cmd(char val);
void RS_high();
void RS_low();
void E_high();
void E_low();
void plus();
void minus();
int preftemp;
int main(void){
    preftemp = 20;
    DDRA=0b00000110;
    DDRB=0b11111111;
    DDRD=0b11100111;
    startTempRead();

while(1)
{
    int pbtn = (0b00010000 & PIND);
    int mbtn = (0b00001000 & PIND);
    if(pbtn == 0b00010000){
        plus();
    }

    if(mbtn == 0b00001000){
        minus();
    }

disp_init();
    disp_clear();

    char s[14];
    sprintf(s, "Pref: %d Act: %d", preftemp, checkTemp());

    disp_print(s);
    _delay_ms(200);
```

```
}  
}  
  
    //Startar termometern och omvandlaren  
void startTempRead() {  
    ADCSRA = 0xDC; //Startar termometern  
    ADMUX = 0b11100000; //Startar omvandlaren  
    //SFIOR = 0x00;  
}  
  
    //Läser temperaturen från termometern  
int tempRead() {  
    int cycles = 80;  
    int res=0;  
    for(int i = 0; i < cycles; i++) {  
        int t1 = ADCH;  
        float t = t1*10*100/1024; //ADCH ger temperaturen  
        res += t;  
    }  
    int temp = res/cycles;  
    return temp; //5=volt  
}  
  
    //Jämför temperaturen mot preferensvärdet och tänder rätt dioder  
int checkTemp(){  
    int tempTest = tempRead();  
    if (tempTest>preftemp) {  
        //set_pin('A',PA2,1);  
        //set_pin('A',PA1,0);  
        PORTA = 0b00000100; //Grön dioder lyser  
    } else {  
        //set_pin('A', PA1, 1);  
        //set_pin('A', PA2, 0);  
        PORTA = 0b00000100; //Röd ldioder lyser  
    }  
    return tempTest;  
}  
  
void plus(){  
    preftemp++;  
    //char s[24];  
    //sprintf(s, " Pref temp: %d", preftemp);  
    //disp_print("%d", preftemp);  
    //pd3  
}  
  
void minus(){  
    preftemp--;  
    //pd4  
}  
  
void write_cmd(char val){  
    PORTB=val;  
    RS_low();
```

```
        E_flip();
    }

void write_data(char val){
    PORTB=val;
    RS_high();
    E_flip();
    //disp_reset();
}

void disp_init(){
    RS_low();
    E_low();
    PORTB=0b00111100; //kod för function set
    E_flip();
    PORTB=0b00001111; //kod för on
    E_flip();
}

void disp_clear() {
    E_low();
    RS_low();
    PORTB=0b00000001; //kod för clear displ
    E_flip();
}

void E_low(){
    PORTD &= ~(1<<0);
}

void E_high(){
    PORTD |= (1<<0);
}

void RS_low(){
    PORTD &= ~(1<<2);
}

void RS_high(){
    PORTD |= (1<<2);
}

void E_flip(){
    E_high();
    E_low();
}

void disp_print(char *str){
    int i = 0;
    for(i=0; str[i]!='\0'; i++){
        write_data(str[i]);
    }
}
```

```
}  
void set_pin(char port, char pin, char state) {  
    char set = 1 << pin;  
    if(port == 'A'){  
        set = PORTA;  
        if(set && !state){ //ändra från 1 > 0  
            PORTA ^= set;  
        }  
        if(!set && state){ //ändra från 0 > 1  
            set = 1 << pin;  
            PORTA ^= set;  
        }  
    }  
    else if(port == 'B'){  
        set = PORTB;  
        if(set && !state){ //ändra från 1 > 0  
            PORTB ^= set;  
        }  
        if(!set && state){ //ändra från 0 > 1  
            set = 1 << pin;  
            PORTB ^= set;  
        }  
    }  
    else if(port == 'C'){  
        set = PORTC;  
        if(set && !state){ //ändra från 1 > 0  
            PORTC ^= set;  
        }  
        if(!set && state){ //ändra från 0 > 1  
            set = 1 << pin;  
            PORTC ^= set;  
        }  
    }  
    else if(port == 'D'){  
        set = PORTD;  
        if(set && !state){ //ändra från 1 > 0  
            PORTD ^= set;  
        }  
        if(!set && state){ //ändra från 0 > 1  
            set = 1 << pin;  
            PORTD ^= set;  
        }  
    }  
}
```