

```

/*
 * berra.c
 *
 * Created: 2017-04-03 09:35:22
 * Author : ine14abe
 */
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 8000000UL;
#include <util/delay_basic.h>

//-----Globala variabler-----
int goalTemp = 40;
int currentTemp = 0;

//----- Port Setup -----
void setUpPorts(){ //Sätter riktningen på alla portar rätt.
    DDRA = 0x06; //0b00000110
    DDRB = 0xff; //Sätter hela B-porten utåt.
    DDRC = 0x00;
    DDRD = 0xff; //Alla D-portarna utåt.
    ADMUX = 0b11000000;
    SFIOR = 0x00; //Testar att koppla denna från föreg rapport
    //DDRD |= _BV(PD1);
    //DDRD |= _BV(PD5);
    //DDRD |= _BV(PD6); Pekar Rs, Rb, E -portarna utåt.
}

//-----LED-----
void lightOn(){
    PORTA |= _BV(PA1);
}
void lightOff(){
    PORTA &= ~_BV(PA1);
}

//-----Värmeelement-----
void heatOn(){
    PORTA |= _BV(PA2);
}

void heatOff(){
    PORTA &= ~_BV(PA2);
}

//-----Termometer-----
int readTemp(){
    _delay_ms(1000);

    int sumRead = 0;
    for (int i = 0; i < 8; i = i + 1){
        ADCSRA = 0b11000110;
    }
}

```

```

        while(ADCSRA == 0b11000110){//prescaler 64
            // lämnar while-loopen då bit 6 slås om till 0 av
            // hårdvaran vid färdig konvertering
        }
        int readVolt = ADC; //potentiell felkälla
        sumRead = sumRead + readVolt;
    }

    int temp = (sumRead/8)*2.56;
    temp = temp/10;
        if(currentTemp!=temp){
            currentTemp = temp;
            printCurrentTemp();
        }
}

// ----- Display -----

void displayCommand(char c){    //Utför ett command; vilket anges genom c.
    _delay_ms(200);
    PORTD &= ~_BV(PD1); // RS 0
    PORTD &= ~_BV(PD5); // RW 0
    PORTD |= _BV(PD6); // E=1 FÖR ATT SÄKERT VETA VAD UTGÅNGSLÄGET
ÄR

    PORTB = c;    //Skickar datan.

    PORTD &= ~_BV(PD6); //Sätter E = 0. Nu läses data av.

    PORTD |= _BV(PD6); // E=1 FÖR ATT SÄKERT VETA VAD UTGÅNGSLÄGET
ÄR (ev. kaka på kaka men better safe than sorry)
}

void displayOn(){    //Gör skärmen helt redo för vår användning; sätter på skärmen och
pekar alla databuss-pins utåt.
    DDRB = 0xff;    //Sätter hela B-porten utåt.
    DDRD |= _BV(PD1);
    DDRD |= _BV(PD5);
    DDRD |= _BV(PD6); // Pekar Rs, Rb, E -portarna utåt.

    PORTD &= ~_BV(PD1); // RS 0
    PORTD &= ~_BV(PD5); // RW 0
    PORTD |= _BV(PD6); // E=1

    displayCommand(0b00111000); //Ställer in FÖRUTSÄTTNINGARNA; antal
bitar i databussen, antal rader osv.
    displayCommand(0b00001100); //Display on, cursor på och blinkande.
    //displayCommand(0b00000001); //Clearar
    //displayCommand(0b00000010); //Sätter cursorn på Home
}

void writeChar(char c) {

```

```

    _delay_ms(200);
    PORTD |= _BV(PD1); //Sätter RS till data mode
    PORTD &= ~_BV(PD5); //Sätter R/W = 0, dvs till write mode.
    PORTD |= _BV(PD6); //Sätter E = 1.
    PORTB = c; //Skickar datan.

    PORTD &= ~_BV(PD6); //Sätter E = 0. Nu läses data av.

    PORTD |= _BV(PD6); //Sätter E = 1.
}

void writeString(char string[]){
    int i = 0;
    while(string[i] != '\0'){ // "\0" betyder "null" i ASCII.

        writeChar(string[i]); //Skriver ut bokstaven.
        i++;
    }
}

void printStationaryText(){
    writeString("Temp: ");
    displayCommand(0b10001110);
    writeChar(0b11011111); //Skriver ut gradertecknet
    writeString("C");

    displayCommand(0b11000000); //Andra raden, första platsen

    writeString("Goal Temp:");
    displayCommand(0b11001110);
    writeChar(0b11011111);
    writeString("C");
}

void printGoalTemp(){
    displayCommand(0b11001100); //Flyttar cursorn till rätt plats
    char c = goalTemp;
    char str[3];
    sprintf(str, "%d", goalTemp);
    writeString(str);
}

void printCurrentTemp(){
    displayCommand(0b10001100); //Flyttar cursorn till rätt plats
    char c = currentTemp;
    char str[3];
    sprintf(str, "%d", currentTemp);
    writeString(str);
}

void welcomeMessage1(){

```

```

        writeString("Welcome to");
        displayCommand(0b11000000);
        writeString("Optima Sous Vide");
        _delay_ms(10000);
        _delay_ms(10000);
        _delay_ms(10000);
        _delay_ms(10000);
        _delay_ms(10000);
        _delay_ms(10000);
        displayCommand(0b00000001);    //Rensar displayen
        displayCommand(0x00);    //Sätter cursor i ursprungsläge
    }

```

```

void welcomeMessage2(){
    writeString("Perfect Food.");
    displayCommand(0b11000000);
    writeString("Every Time.");
    _delay_ms(10000);
    _delay_ms(10000);
    _delay_ms(10000);
    _delay_ms(10000);
    _delay_ms(10000);
    _delay_ms(10000);
    displayCommand(0b00000001);    //Rensar displayen
    displayCommand(0x00);    //Sätter cursor i ursprungsläge'
}

```

//-----Händelser-----

```

void manageTemp(){
if(currentTemp<goalTemp){
    heatOn();
    }

else{
    heatOff();
}
}

```

```

void signalTemp(){
    if(currentTemp>goalTemp-2 && currentTemp<goalTemp+2){
lightOn();
    }
else{
    lightOff();
}
}
}

```

```

int main(void)
{
    setUpPorts();
    displayOn();
    welcomeMessage1();
    welcomeMessage2();
    printStationaryText();
}

```

```
        printGoalTemp();
        readTemp();
while (1)
        {
        readTemp();
        manageTemp();
        signalTemp();

if(PIND & (1<<PD2)){ //Om PD2 är hög dvs knappen trycks ned
        _delay_ms(700);
        goalTemp = goalTemp + 1;
        printGoalTemp();
        }
else if(PIND & (1<<PD3)){ //Om PD3 är hög dvs knappen trycks ned
        _delay_ms(700);
        goalTemp = goalTemp - 1;
        printGoalTemp();
        }
        }
}
```