

Lunds Universitet  
Institutionen för Elektro- och informationsteknik  
EITF11 Digitala projekt

## Sous vide

Handledare: Bertil Lindvall  
Marcus Carlson  
Victor Bernhardsson  
Oskar Bengtson

2016-05-23

### Abstract

The following describes the process of constructing and developing a prototype of a sous vide machine and it also serves as a specification for the prototype.

## Innehållsförteckning

Abstract .....	1
1. Inledning.....	3
2. Kravspecifikation.....	3
2.1 Funktionella krav.....	3
2.1.1 Krav 1 .....	3
2.1.2 Krav 2 .....	3
2.1.3 Krav 3 .....	3
2.1.4 Krav 4 .....	3
2.1.5 Krav 5 .....	3
3. Teori.....	3
3.1 Hårdvara .....	3
3.1.1 Kopplingschema.....	3
3.1.2 Processor .....	3
3.1.3 Display.....	4
3.1.4 LED .....	4
3.1.5 Knappar .....	4
3.1.6 Resistorer.....	4
3.1.7 Temperaturmätare .....	4
3.1.8 Transistor.....	4
3.1.9 Vattenkokare.....	4
3.2 Mjukvara.....	4
4. Arbetsprocessen.....	4
4.1 Planering.....	5
4.2 Konstruktion.....	5
4.3 Mjukvaruframtågning.....	5
5. Resultat .....	5
5.1 Maskinen mäter vattnets aktuella temperatur.....	5
5.2 Maskinen värmer temperaturer tills måltemperaturen uppnåts .....	5
5.3 Två knappar användes för att ställa in temperaturen.....	5
5.4 Displayen visar nuvarande temperatur samt måltemperatur .....	5
5.5 Lysdioden tänds när nuvarande temperaturen är inom intervallet från måltemperaturen ...	5
6. Diskussion och slutsatser .....	6
7. Referenser .....	7

8.	Appendix.....	8
8.1	Bilaga 1 Krettschema.....	8
8.2	Bilaga 2 Mjukvara.....	8

## 1. Inledning

Syftet med denna rapport är att beskriva utvecklingsarbetet av en prototyp för en sous vide-maskin. Rapporten skall också utgöra en dokumentation av prototypen som möjliggör vidareutveckling av produkten.

En sous vide-maskin är ett köksredskap som används för att tillaga mat under en kontrollerade temperatur för att bevara smak. Förenklar skulle man kunna beskriva maskinen som en kokgryta där man kan ställa in vattnets temperatur.

## 2. Kravspecifikation

### 2.1 Funktionella krav

#### 2.1.1 Krav 1

Maskinen skall kunna mäta vattnets aktuella temperatur.

#### 2.1.2 Krav 2

Maskinen skall kunna värma vattnet tills måltemperaturen uppnåts.

#### 2.1.3 Krav 3

Genom två knappar skall maskinen kunna ställas in på att arbeta mot en specifik måltemperatur.

#### 2.1.4 Krav 4

Maskinen skall kunna visa måltemperatur och aktuelltemperatur på en bildskärm.

#### 2.1.5 Krav 5

Maskinen skall indikera att måltemperaturen uppnåts genom att en grön lampa tänds.

## 3. Teori

### 3.1 Hårdvara

#### 3.1.1 Kopplingsschema

För att underlätta konstruktionen av prototypen och vidareutvecklingen av den skapades ett kopplingsschema i PowerLogic, se bilaga 1.

#### 3.1.2 Processor

En ATmega16 med 40 stycken pinnar användes för att kontrollera sous vide-maskinen. Av dessa 40 pinnar var 32 stycken I/O. Dessa var indelade på de fyra portarna, A, B, C och D. Port B användes exklusivt för att skicka data till display. Port A användes för att ta emot input ifrån termometern. Port D användes för att ta emot input ifrån de två knapparna, kontrollera output till vattenkokaren och för

några styrfunktioner till displayen. Port C användes för att styra dioden. 8 av pinnarna på processorn användes av en Jtag för att debugga programvaran. Atmega16 har 16 kilo byte programmerbart flashminne som användes för att lagra mjukvaran på. För mer detaljerad information se referenser.

### 3.1.3 Display

Den display som användes är en SHARP Dot-Matrix som är en alfanumerisk teckendisplay. Displayen kontrolleras av processorn av en instruktionskod bestående av nio bittar. RS och RW användes för att styra displayen och DB0-7 för att ge den data. I prototypen användes displayen för att skriva ut information om aktuell temperatur och om måltemperaturen. För mer detaljerad information se referenser.

### 3.1.4 LED

För att indikera huruvida måltemperaturen uppnåtts eller inte användes en grön LED. Denna lyser grönt om den aktuella temperaturen är  $\pm 1^\circ\text{C}$  från måltemperaturen, och lyser inte alls om temperaturen är utanför intervallet.

### 3.1.5 Knappar

För att möjliggöra inställning av måltemperaturen, vars ursprungsvärde är  $25^\circ\text{C}$ , användes två knappar. En knapp användes för att sänka måltemperaturen med  $1^\circ\text{C}$  och en knapp användes för att höja måltemperaturen med  $1^\circ\text{C}$ .

### 3.1.6 Resistorer

Tre resistorer användes. Två i koppling med knapparna för att säkerställa att frånvaro av signal kan registreras som 0 av processorn och inte som "ingenting". Den sista resistorn på  $5,6\text{ k}\Omega$  användes till termometern.

### 3.1.7 Temperaturmätare

En LM335 temperaturmätare användes för att läsa av aktuell temperatur i  $^\circ\text{K}$ . Processorn är programmerad att omvandla temperaturen till  $^\circ\text{C}$ . Pin adj ska inte vara ansluten till något. Pluspolen ska vara ansluten till en strömkälla via en resistor på  $5,6\text{ k}\Omega$ , samt till en PA-port (ej via resistorn). Minuspolen ska direkt vara ansluten till ground. För mer detaljerad information se referenser.

### 3.1.8 Transistor

Prototypen drevs av en alltför svag ström för att kunna värma vatten nämnvärt. Därför kopplades processorn till en transistor som släpper igenom en högre spänning till vattenkokaren.

### 3.1.9 Vattenkokare

En vattenkokare på 12 volt användes för att värma vattnet.

## 3.2 Mjukvara

Mjukvaran skrevs i #C i Atmel studio 6.2 och debuggades med en Jtag. Översiktligt så består mjukvaran av en setup-funktion som startar och initierar displayen och sedan en oändlig while-loop som kontinuerligt kollar input ifrån knappar, uppdaterar temperaturinformationen och som styr vattenkokaren. För mer specifik beskrivning se bilaga 2.

## 4. Arbetsprocessen

## 4.1 Planering

Arbetet inleddes med en planering och diskussion kring vad som skulle uppnås och vad målet med prototypen var. Vad sous vide-maskinen skulle kunna göra och vad den inte skulle kunna göra. Efter denna diskussion gjordes en skiss av ett kopplingschema och en diskussion kring vilken hårdvara som skulle användas inleddes. Efter att Lindvall gett återkoppling på detta gjordes ett utförligt kopplingschema, se bilaga 1.

## 4.2 Konstruktion

Efter att kopplingschemat skapats inleddes arbetet med att koppla samman hårdvaran i prototypen. Detta gjordes genom att löda fast komponenterna på ett kretskort. Inledningsvis fungerade inte hårdvaran som önskat men efter att kopplingarna debuggats med Atmel Studio 6.2 så fortskred arbetet i mjukvaruutveckling.

## 4.3 Mjukvaruframttagning

Mjukvaruutvecklingen inleddes med att områdesansvariga läste på om programmeringsspråket C eftersom ingen hade använt språket tidigare. Därefter läste gruppen på om hur kommunikation skulle ske mellan processorn och hårdvaran. Därefter påbörjades den faktiska programmeringen, och debuggningprocessen. För specifik kod se bilaga 2.

# 5. Resultat

Projektets resultat blev en fungerande prototyp av en sous vide-maskin som uppfyller alla krav i kapitel 2.1.

## 5.1 Maskinen mäter vattnets aktuella temperatur

Efter en kalibrering av termometern, kunde den visa rätt vattentemperatur inom en rimlig noggrannhet för detta projekt.

## 5.2 Maskinen värmer temperaturer tills måltemperaturen uppnåtts

Vi testade värma vattnet till en maxtemperatur av 50 °C, och det klarade vattenkokaren av. När väl temperaturen nått 50 °C stängdes vattenkokaren av, och den gröna lysdioden tändes. Vi genomförde ett flertal tester med olika måltemperaturer och sous-vide maskinen klarade samtliga fall.

## 5.3 Två knappar användes för att ställa in temperaturen

Detta kravet är uppfyllt.

## 5.4 Displayen visar nuvarande temperatur samt måltemperatur

Displayen kan visa temperaturer mellan 0-99 °C.

## 5.5 Lysdioden tänds när nuvarande temperaturen är inom intervallet från måltemperaturen

Som nämnt i 5.2 så lyser lysdioden, när temperaturen är inom intervallet.

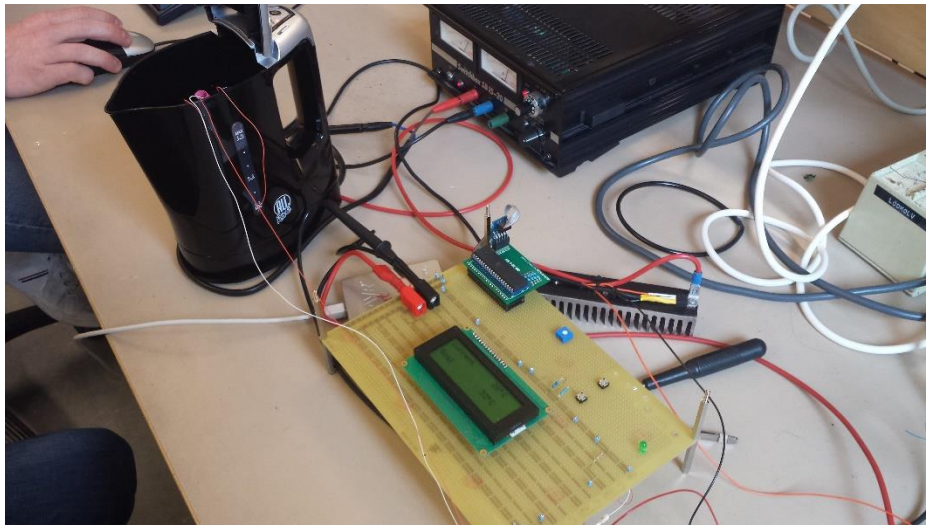


Bild 1. En översikt av prototypen

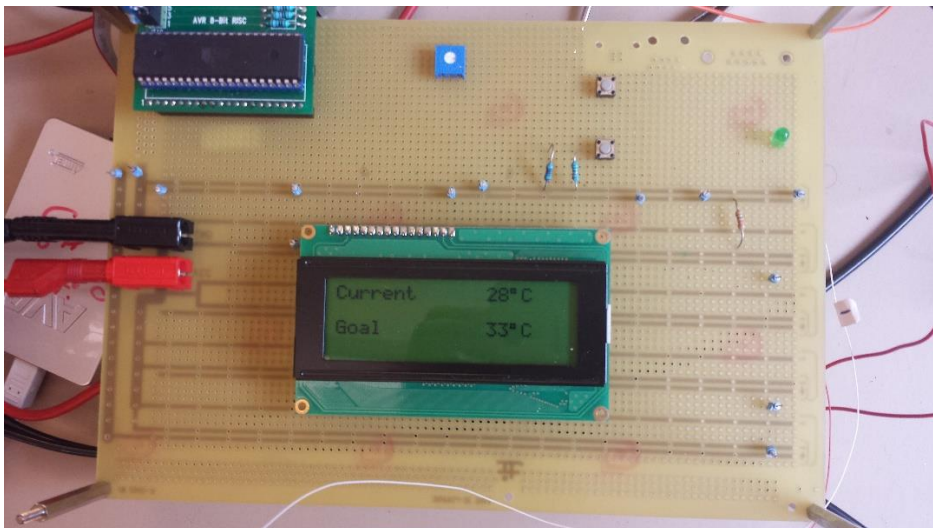


Bild 2. En bild på sous vide-maskinens display

## 6. Diskussion och slutsatser

Att vara med och utveckla, och med hjälp av Bertil, planera och leda framtagningen av en prototyp har varit mycket givande. Vi har dels fått en ökad insikt i processen för att ta fram en prototyp men också lärt oss programmera i C, läsa och förstå datablad och felsökning. Vår grupp är också nöjda med slutresultatet. Prototypen fungerar enligt avsikt. Även om den kanske inte är den snyggaste sous vide-maskin vi har sett.

Under utvecklingsprocessen stötte vi på en del problem. Vi hade inte alla kunskaper som krävdes men vi anskaffade oss dem. Vi kopplade fel och vi skrev fel i vår kod men genom handledning och hårt arbete så löste vi alla fel. En stor insikt arbetet har gett oss är att produktframtagning inte är en linjär process. Utan det krävs att man gång på gång går tillbaka och förbättrar och förändrar det arbete man redan gjort.

Sammanfattningsvis har detta varit ett spännande och givande kurs. Och vårt arbete har förhoppningsvis förberett oss för arbetslivet.

## 7. Referenser

Databladet AVR-ATmega16 High-performance AVR 8-bit Microcontroller

[http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16\\_adv.pdf](http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16_adv.pdf)

Datum: 2016-05-11

Databladet Sharp Dot-Matrix - LCD Units Alfanumerisk teckendisplay

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Display/LCD.pdf>

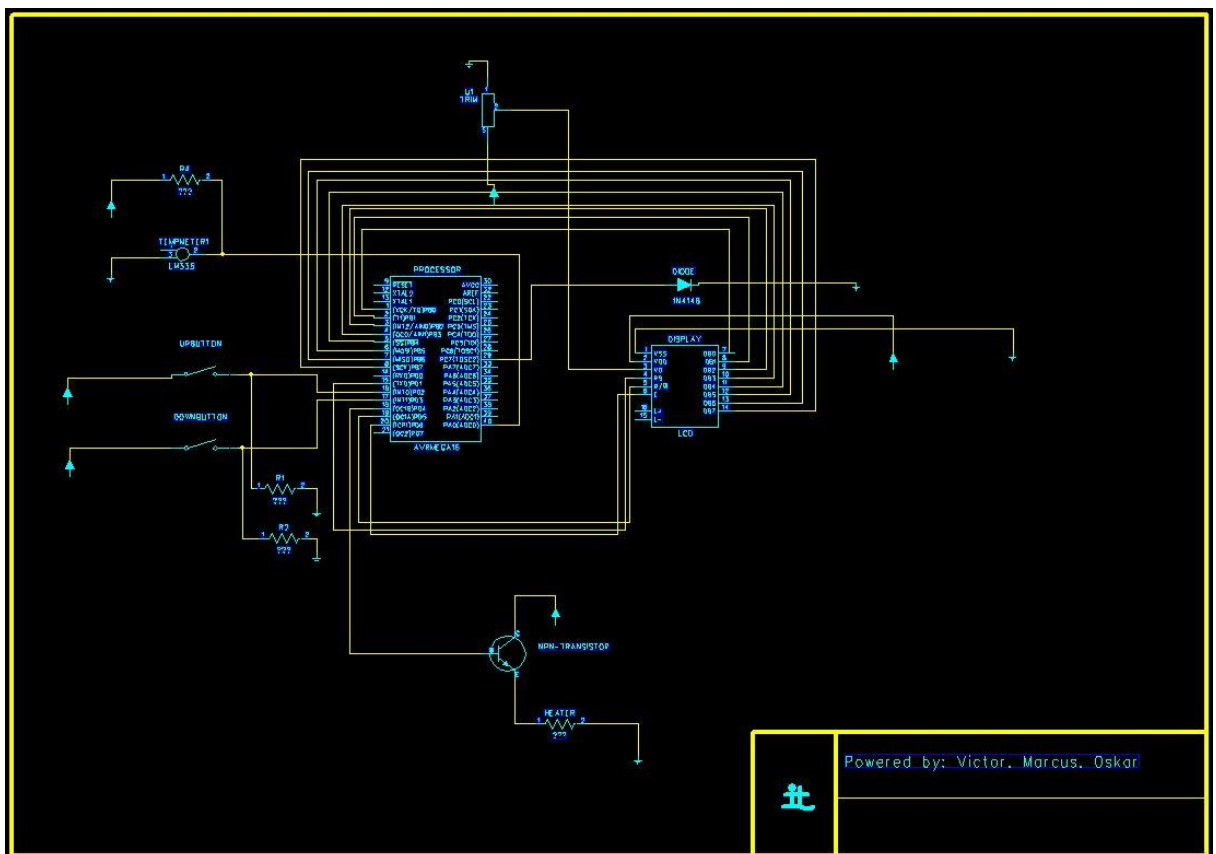
Datum: 2016-05-11

Databladet LM445 Kelvin temperature sensor

Datum: 2016-05-11

## 8. Appendix

### 8.1 Bilaga 1 Kretsschema



### 8.2 Bilaga 2 Mjukvara

```
/*
 * berra.c
 *
 * Created: 2016-04-15 14:31:25
 * Author: digpi08 marcus victor och oskar
 */
```

```
#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
```

```
int goaltemp = 25;
```



```

int i=2;//siffran på plats ett för goaltemp
int k=5;//siffran på plats två för goaltemp
int currenttemp = 30;
int i1 = 3;
int k1 = 0;

void heatOn(){
    PORTD |= _BV(PD4);//skickar en etta
}

void heatOff(){
    PORTD &= ~_BV(PD4);//skickar en nolla
}

void writeCmd(char b){//FUNKAR SÅ JÄKLA BRA JUST NU; RÖR DU SÅ DÖR DU
    rsHigh();
    rwHigh();
    eHigh();

    PORTB= b;

    rwLow();
    rsLow();
    eLow();
    eHigh();
}

void writeData(char b){
    eHigh();//test
    rsHigh();//test
    rwHigh();//test

    PORTB = b;
    rwLow();
    eLow();
    eHigh();
    rwHigh();
}

void setUp(){
    DDRB = 0xff;//alla pins på port b ska skicka (dvs vara inställda på
output)
    DDRC = 0b10000000;
    DDRA = 0b00000001;
    DDRD = 0b01110010;
    rsHigh();
    rwHigh();
    eHigh();
    writeCmd(0b00111000);//function set
    writeCmd(0b00001100);//display on
    writeCmd(0b10000000);//set initial address
}

```

```

void setAdrSecondLine(){
    writeCmd(0xc0);
    //PORTB = 0xc0;
}

void setAdrThirdLine(){
    writeCmd(0x94);
}

void setAdrFourthLine(){
    writeCmd(0xD4);
}

void setAdrCurrentTemp(){//anger var siffrorna börjar skrivas på skärmen
    writeCmd(0x8D);
}

void setAdrGoalTemp(){//Anger var siffrorna börjar skrivas på skärmen
    writeCmd(0xA1);
}

void setAdrCelcCurrent(){//skriver grader celsius på fast plats vid currentemp
    writeCmd(0b10001111);
    _delay_ms('500');
writeData(0b11011111);
_delay_ms('500');
writeData('C');
}

void setAdrCelcGoal(){//skriver grader celsius vid goaltemp
    writeCmd(0b10100011);
    _delay_ms('500');
writeData(0b11011111);
_delay_ms('500');
writeData('C');
}

void checkTemp(){// kollar om nuvarande temperaturen är inom rimligt intervall. OBS
denna metod kanske vi vill göra till boolean och använda till kaffekokaren "if true
stop boiling water"...
if (currenttemp < goaltemp+2 && currenttemp > goaltemp-2)
{
    lightOn();
}
else{
    lightOff();
}
}

void checkHeat(){
if(currenttemp<goaltemp){
    heatOn();
}
else{
    heatOff();
}
}

```

```
}  
}
```

```
void clearD(){//denna metod funkar bra  
    rsHigh();  
    rwHigh();  
    eHigh();  
  
    PORTB=0b00000001;  
  
    rwLow();  
    rsLow();  
    eLow();  
    eHigh();  
    //rsHigh();  
    //rwHigh();  
  
}  
void clearDisplay(){//test av cmdmetoden FUNKAR  
    writeCmd(0b00000001);  
}  
  
void rsLow(){  
    //PD1 = 0;  
    PORTD &= ~_BV(PD1);  
  
}  
  
void rsHigh(){  
    //PD1 = 1;  
    PORTD |= _BV(PD1);  
}  
  
void rwLow(){  
    //PD5 = 0;  
    PORTD &= ~_BV(PD5);  
}  
  
void rwHigh(){  
    //PD5 = 1;  
    PORTD |= _BV(PD5);  
}  
  
void eLow(){  
    //PD6 = 0;  
    PORTD &= ~_BV(PD6);  
}  
  
void eHigh(){  
    //PD6 = 1;  
    PORTD |= _BV(PD6);  
}  
  
void lightOn(){//FUNKAR  
    PORTC |= _BV(PC7);
```

```

}

void lightOff(){//FUNKAR
    PORTC &= ~_BV(PC7);
}

void writeNum(int b){
    eHigh();
    rsHigh();
    rwHigh();
    PORTB = b;
    rwLow();
    eLow();
    //eHigh();
    //rwHigh();
    //return?
}

int tempRead(){
    ADCSRA = 0b1100010;// just nu har vi prescaler på 16
    while(ADCSRA==0b1100010){
    }
    int rawvoltage = ADC;
    return rawvoltage; //just nu läser den volten från 5V
}

void printTemp(){

    int read = tempRead();

    read = read*5;//nu har vi antalet millivolt egentligen ska det vara 4.88
    read = read/10;//grader i kelvin

    int celc = read - 273;//kalibrering till celsius

    if(currenttemp!=celc){
        currenttemp = celc;

    int i1 = celc/10;
    int k1 = celc%10;
    _delay_ms('500');
    setAdrCurrentTemp();
    _delay_ms('500');
    writeData(i1 + '0');
    _delay_ms('500');
    writeData(k1 + '0');

    }
}

```

```

int main(void)
{

    setUp();

    _delay_ms('500');
    clearDisplay();
    ADMUX = 0b01000000;
    SFIOR = 0x00;

    _delay_ms('500');
    writeData('C');
    _delay_ms('500');
    writeData('u');
    _delay_ms('500');
    writeData('r');
    _delay_ms('500');
    writeData('r');
    _delay_ms('500');
    writeData('e');
    _delay_ms('500');
    writeData('n');
    _delay_ms('500');
    writeData('t');

    _delay_ms('500');

    setAdrThirdLine();

    _delay_ms('500');
    writeData('G');
    _delay_ms('500');
    writeData('o');
    _delay_ms('500');
    writeData('a');
    _delay_ms('500');
    writeData('l');
    _delay_ms('500');

    /*
    setAdrCurrentTemp();//här skriver vi current tempen
    _delay_ms('500');
    writeData(i1 + '0');
    _delay_ms('500');
    writeData(k1 + '0');

    _delay_ms('500');
    setAdrCelcCurrent();
    */

    printTemp();
    _delay_ms('500');

```

```
setAdrCelcCurrent();
```

```
_delay_ms('500');
```

```
setAdrGoalTemp();//här skriver vi goal tempen
```

```
_delay_ms('500');
```

```
writeData(i + '0');
```

```
_delay_ms('500');
```

```
writeData(k + '0');
```

```
_delay_ms('500');
```

```
setAdrCelcGoal();
```

```
_delay_ms('10000');
```

```
while(1) {
```

```
    checkTemp();
```

```
    checkHeat();
```

```
    tempRead();
```

```
    printTemp();
```

```
    if(PIND & (1<<PD2)){//den övre knappen
```

```
        goaltemp = goaltemp + 1;
```

```
        i = goaltemp/10;//plockar siffran på plats ett
```

```
        k = goaltemp%10;//plockar siffran på plats två
```

```
        setAdrGoalTemp();
```

```
        _delay_ms('500');
```

```
        writeData(i + '0');//konverterar int till char
```

```
        _delay_ms('100');
```

```
        writeData(k + '0');
```

```
    }
```

```
    else if(PIND & (1<<PD3)){//den undre knappen
```

```
        goaltemp = goaltemp - 1;
```

```
        i = goaltemp/10;
```

```
k = goaltemp%10;

setAdrGoalTemp();
_delay_ms('500');
writeData(i + '0');//konverterar int till char
_delay_ms('100');
writeData(k + '0');
```

```
}
```

```
}  
}
```