

Rapportsammanfattning

Optima Sous Vide

Elektro- och informationsteknik, Lunds Tekniska Högskola
Anton Bengtsson, Hannes Helin, Ingrid Hyltander
Handledare: Bertil Lindvall
2017-05-20

Abstract

The following report describes the project to build a functioning sous vide, a cooking device which allows a chef to heat water to a desired temperature and to maintain that temperature. The project is a result from the course EITF11 which is held at the Faculty of Engineering at Lund University. The report covers the process from specifying requirements, design, construction to reaching the final prototype. The reader will be given insight in chosen hardware, C-code and difficulties that was discovered and how they were handled.

Innehållsförteckning

<u>Rapportsammanfattning</u>	<u>0</u>
<u>Abstract</u>	<u>1</u>
<u>Innehållsförteckning</u>	<u>2</u>
<u>Inledning</u>	<u>3</u>
<u>Produktbeskrivning</u>	<u>3</u>
<u>Kravspecifikation</u>	<u>3</u>
<u>Teori</u>	<u>4</u>
<u>Genomförande</u>	<u>5</u>
<u>Resultat</u>	<u>6</u>
<u>Diskussion och slutsatser</u>	<u>7</u>
<u>Appendix A - Kretskoppling</u>	<u>9</u>
<u>Appendix B- Källkod</u>	<u>10</u>

Inledning

Studentens liv är en ständig balansgång mellan studier och vardagsbestyr. Böcker ska läsas och matlådor ska lagas. Grundtanken med författarnas Sous Vide-prototyp är att studenten inte ska behöva välja mellan att glänsa på tentor och att glänsa i lunchrummet. Med Sous Vide-prototypen sköter matlagningen sig till stor del själv och studenten kan ägna sig åt hårdvarunära programmering och andra mer kultiverade sysslor, utan att för den delen behöva skämmas i mikrokön.

Syftet med projektet är att producera en välfungerande Sous Vide-prototyp. På ett mer abstrakt plan är syftet att gruppledningarna i arbetet mot en sådan prototyp, ska skaffa sig kunskaper inom elektronik och prototyputveckling. Vad som är en "välfungerande Sous Vide-prototyp" specificeras närmare i avsnitt *Kravspecifikation*.

Återstoden av rapporten består av produktbeskrivning, kravspecifikation och en beskrivning av den använda mjuk- och hårdvaran. Därefter följer en beskrivning av genomförandet och resultatet. Rapporten avslutas med diskussionsdel.

Produktbeskrivning

I en Sous Vide tillagas livsmedel genom att de vakuumpförpackas och placeras i ett vattenbad vars temperatur kan regleras. Detta gör att livsmedel bibehåller saftighet och blir jämnt tillagat. I projektet används en vattenkokare som behållare för vattenbadet och till uppvärmning av detta. Via återkoppling av vattentemperaturen till processorn reglerar denna vattenkokaren så att vattenbadet uppnår önskad måltemperatur.

Kravspecifikation

Den färdiga prototypen ska uppfylla följande krav:

- Mäta nuvarande temperatur i vattnet
- Redovisa den nuvarande temperaturen på en display
- Ställa in måltemperatur
- Redovisa vald måltemperatur på en display
- Se till att vattnet når vald måltemperatur
- Se till att vattnet behåller vald måltemperatur

Teori

Hårdvara:

Processor

Som processor användes en ATmega16, en 8-bitars Microcontroller. Processorn har 40 pinnar. 8 av dessa är reserverade till ett JTAG-interface vilken används för att programmera processorn.

Display

Som display användes en SHARP Dot-Matrix display. Displayen kan visa både siffror och bokstäver. Vilket tecken som ska visas på displayen avgörs av via en 8-bitars signal från processor.

Termometer

Som termometer användes en LM35. Processorn mottar en spänning som är proportionell mot uppmätt temperatur mätt i Celcius. En grad Celsius motsvaras av en spänning på 10 mV.

Knappar

Två knappar kopplades till processorn för att möjliggöra för användaren att öka eller minska önskad måltemperatur. När en knapp trycks ner skickas en spänning till processorn och beroende på vilken knapp som signalen kommer ifrån höjer eller sänker processorn måltemperaturen.

Kondensatorer

För att motverka brus och felaktiga signaler kopplades kondensatorer till processorn och displayen.

Resistorer

Flera resistorer användes för att kunna reglera strömmen i kretsen.

Vattenkokare

Som behållare och för uppvärmning av vattenbadet användes en varmvattenkokare på 12 volt.

LED-lampa

En grön LED användes för att visa när varmvattenbadet uppnådde inställd måltemperatur.

Mjukvara:

Mjukvaran är skriven i programspråket C och skapades i AtmelStudio. En JTAG användes för att föra över kod till processorn och för att debugga. Huvudmetoden inleds av att alla portar ställs in på in- respektive utsignal, skärmen sätts på och ett välkomstmeddelande visas på skärmen. Därefter påbörjas en oändlig while-loop där vattentemperatur mäts av, vattenkokaren följer vald måltemperatur och ev. knapptryck registreras och därmed ändrar måltemperaturen. Källkoden finns i sin helhet i Appendix B.

Genomförande

Projektgenomförandet hade fyra huvudsakliga skeden: projektplanering, sammansättning av hårdvara, kodning samt testning. De fyra momenten genomfördes i huvudsak i nämnd ordning, men processen var inte helt linjär. Exempelvis reviderades ritningen och hårdvarans kopplingar under kod- och teststegen.

Projektplaneringen var projektets första moment. I denna initiala del av projektet fastställdes projektets mål, det vill säga vilken prototyp som skulle byggas. Vidare upprättades även en enkel kravspecifikation och ett kopplingschema.

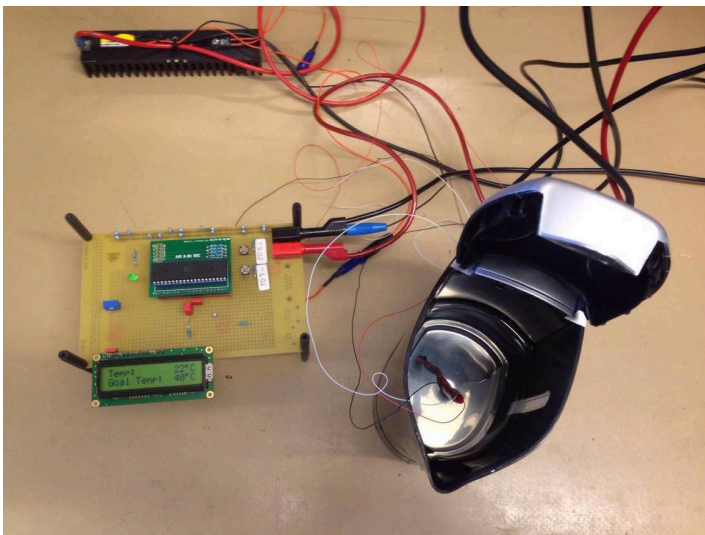
Sammansättningen av hårdvaran var projektets andra moment och innebar att komponenterna löddes fast i en platta och kopplades ihop med koppartråd och virpistol. Kopplingschemat som upprättats i projektplaneringen var ritningen som användes.

I projektets tredje steg, hårdvarukodningen, kopplades Sous Vide-prototypen, mer precist ATMEGA16-processorn, via JTAG till en dator. I Atmel Studio 7 skrevs koden. Programspråket C användes. Kodningsarbetet delades upp i moduler, exempelvis kod till display och kod för temperaturregulering, och arbetet med varje modul fördelades mellan gruppens medlemmar.

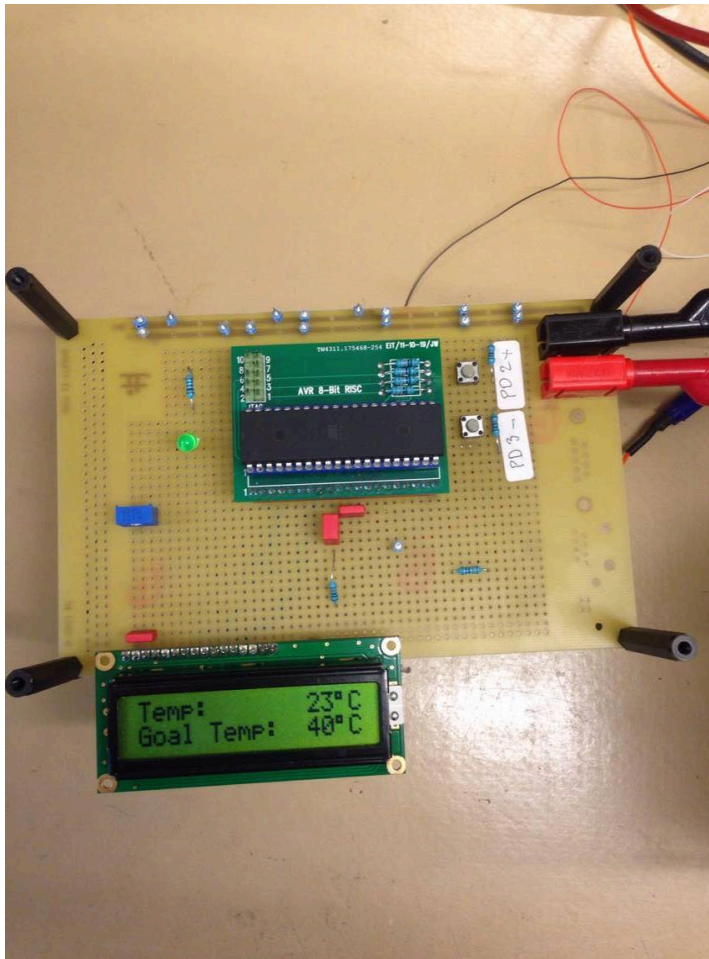
Testningssteget hörde tätt samman med hårdvarukodningen. Testningen genomfördes löpande i samband med att nya kodartefakter togs fram. I de fall artefakterna inte uppfyllde kraven återgick arbetet från testning till kodning. Genom sådana kodning-testning-iterationer arbetades en fungerande kod fram. Det praktiska testförfarandet var att via JTAG överföra kod till Sous Vide-prototypens ATMEGA16-processor och därefter använda Atmel Studio 7:s debugger för att lokalisera fel.

Resultat

Den färdiga prototypen uppfyller alla uppsatta krav i kravspecifikationen. Dock fungerar den endast felfritt när JTAG är inkopplad. Om JTAG inte är inkopplad kan processorn ibland återställa sig ("reset") då måltemperatur uppnås, troligtvis på grund av brus. Figur 1 och Figur 2 visar hur prototypen såg ut.



Figur 1, prototypen i sin helhet



Figur 2, närmare vy över display, processor och knappar

Diskussion och slutsatser

Sammanfattningsvis lyckades själva prototypbygget väl, men inte helt perfekt. Problemet med att processorn återställde sig i tid och otid var ett som inte lyckades lösas, trots omfattande försök. Gruppmedlemmarna anser trots den mindre defekten på slutprodukten att projektet har varit lyckat. Gruppens handledare med sin stora kunskap och många års erfarenhet kunde inte hitta anledningen till defekten i prototypen. Därmed anser gruppmedlemmarna med sin begränsade kunskap och erfarenhet att det är rimligt att de ej själva lyckas åtgärda defekten och att de är nöjda med sin slutprodukt trots allt.

Vidare behöver ett mindre problem med slutprototypen inte nödvändigtvis stå i vägen för projektets större syfte, att lära gruppmedlemmarna om elektronik och prototyputveckling. Snarare är det tvärt om. En lärdom som författarna dragit av problemet med processoråterställningen är att prototyputveckling inte är så rättfram som det kan verka på pappret. Detta kan tyckas som en banal insikt, vilket det också är om man bara fått höra den. Om man emellertid fått uppleva processen och problemen själv rör det sig om en insikt på en helt annan nivå och en insikt som gruppmedlemmarna tror kan vara en viktig pusselbit i framtida arbetsroller.

Ser man till detta mer övergripande syfte, är författarna alltså mycket nöjda och har lärt sig mycket. Från att tidigare aldrig ha sett ett datablad kan nu gruppmedlemmarna ta fram väsentlig information om olika hårdvarukomponenter. Ingen hade erfarenhet varken av programmeringsspråket C eller att koppla ihop hårdvarukomponenter men alla har efter att ha genomfört projektet avsevärt bättre förståelse för dessa områden. En av de viktigaste lärdomarna är hur mjukvara och hårdvara faktiskt fungerar tillsammans, inte minst på binär


```

* 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();
    }
    }
}
```