

LUNDS TEKNISKA HÖGSKOLA

## TEMPERATURMÄTARE MED GRAFRITARE

---

**EITF11**

**Grupp 6**

**Jesper Henrikson & Viktor Sköld**

*Handledare: Bertil Lindvall*

## Innehållsförteckning

<b>Innehållsförteckning</b>	2
<b>Inledning</b>	3
<b>Kravspecifikation</b>	4
<b>Teori</b>	
Processor	5
Temperatursensor	5
LCD-display	5
Knapp	5
Resistorer	5
<b>Genomförande</b>	6
<b>Resultat</b>	7
<b>Diskussion</b>	8
<b>Referenser</b>	9
<b>Kopplingschema</b>	10
<b>Källkod</b>	11

## Inledning

EITF11, eller Digitala Projekt, är en kurs vars syfte är att ge grundläggande kunskaper inom elektronik, digitalteknik samt att skapa färdigheter i själva konstruktionen. Alla kursdeltagande ges möjlighet att valfritt välja projekt inom vissa rimliga ramar. Projektet går sedan ut på att skapa en fungerande prototyp och samtidigt dokumentera detta.

Gruppen valde att konstruera en temperaturmätare som mäter temperatur inomhus såväl som utomhus. Temperaturen redovisas dels som en graf av temperaturen som en funktion av tiden under den senaste veckan, men även utskrivet i teckenform. Skiftandet från redovisning i form av en graf till teckenform och vice versa sker genom ett knapptryck. Eftersom gruppen ej hade någon förkunskap inom ämnet var det svårt att förutse vad som skulle komma att vara utmanande och vad som skulle gå smärtfritt.

## Kravspecifikation

Temperaturstationen ska kunna

- Avläsa aktuell temperatur
- Visa senaste veckans timvisa temperatur i ett linjediagram
- Visa inomhus- respektive utomhustemperaturen i teckenform
- Byta mellan de två alternativa redovisningarna med hjälp av ett knapptryck



## Genomförande

Inledande tillgavs gruppen de komponenter som ansågs esentiella för vårt ändamål samt en verktygslåda med de instrument som skulle kunna användas vid konstruktion. Ett kopplingsschema skapades sedan i programmet Power Logic. Detta fungerade sedan som en mall för hur kopplingarna skulle utföras på det verkliga kretskortet. Till vår hjälp, med syfte att förstå hur kopplingsschemat skulle kopplas, användes datablad till de olika komponenterna. Processor, display, temperatursensorer, knapp, resistorer samt spänning och jord kopplades sedan samman genom lödning och virning av trådsladdar mellan de olika pinnarna. Slutligen skrevs C-kod i syfte att få hårdvaran att göra som önskat. Källkoden skrevs och testades i programmet Atmel Studio 6.0.

## Resultat

Temperaturgivaren slutställdes till slut efter mycket om och men. Gruppen ändrade efterhand några av sina krav vilket gjorde att slutprodukten blev lite skild från vad som önskades initialt. Gruppen lyckades framställa en dubbeldisplay som redovisar inomhustemperaturen i det ena halvplanet och utomhustemperaturen i det andra halvplanet. Temperaturerna representeras i form av linjediagram. Initialt önskades en knappfunktion som skulle skifta från grafisk till skriftlig redovisning. Denna slopades dock senare då vi insåg att det skulle krävas mer tid än vad vi hade, på grund av de problem vi stötte på.

## Diskussion

Arbetet i kursen Digitala Projekt (EITF11) har varit mycket lärorikt. Mycket tålamod har krävts men samtidigt har nya vet- respektive kunskaper erhållits. Under arbetets gång har vi till slut enats om att databladens information är vettigare och matnyttigare än vad gruppmedlemmarna initialt trodde. Således gick gruppen från att till en början vara närmast helt beroende av assistans till att senare leta efter information på ett mer självständigt sätt. Gruppmedlemmarna har även lärt sig att skriva källkod i språket "C". I det stora hela har det varit en mycket lärorik kurs där kursdeltagarna har fått testa sig själv inom områden som tidigare varit helt okända.

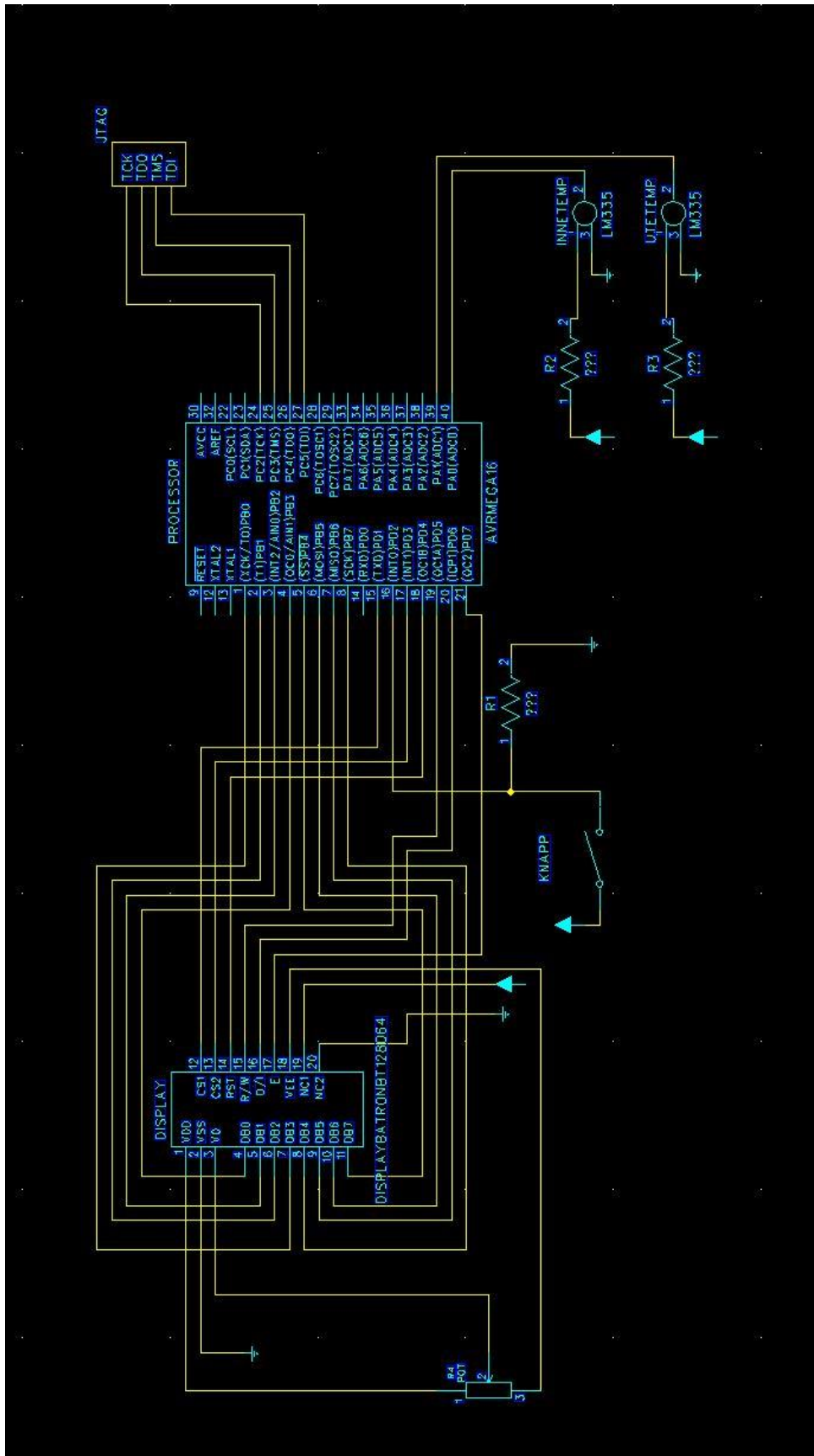


## Referenser

[http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16\\_sum.pdf](http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16_sum.pdf)

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Sensors/lm335.pdf>

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Display/GDM12864H.pdf>



KOD\*\*\*\*\*

```
/*
 * GccApplication1.c
 * Created: 2014-03-26 15:47:35
 * Author: digpi06
 */

#include <avr/io.h>
#include <util/delay.h>
#include <math.h>
#include <avr/interrupt.h>

char adValOut;
char adValIn;
long int count=0;

void init(void) {
    set_data_direction();
    init_disp();
    PORTD=0b11010001; //display i utgångsläge
    //cs1, cs2, rst, RS och E höga. RW låg
    write_cmd_right(11000001); //set display start line
}

void clear_display(void) {
    PORTD=0b00011011; //utgångsläge fast ändrat pin 6 och 7
    PORTB=0b01000000; //vi har satt Y-adress till 0.
    for(int x=184;x<192;x++) { //vi anger PORTB i decimalform. 184 motsvarar
//att vi ska skriva x-adressen 000.
        for(int y=64;y<128;y++) { // samma som ovan.
            set_dot_left(x, y, 0);
            set_dot_right(x,y,0);
            _delay_ms(2);
        }
    }
}

void init_disp(void) {
    write_cmd_right(0b00111111); //höger halvplans display sätts på
    write_cmd_left(0b00111111);
}

//ange vilket håll datan ska gå i de olika portarna
void set_data_direction(void) {
    DDRD=0b11111010; //nollan motsv knappen som ska peka in
    DDRA=0b11111100;
    DDRB=0b11111111;
    DDRC=0b11000011;
}

void write_cmd_right(char value) {
    PORTD=0b10011001; //RS (D/I) och CS1 ändras till låga
    PORTB=value;
}
```

```

        toggle_E();
        PORTD=0b11010001; //PORTD tbx till utgångsläget
    }

void write_cmd_left(char value) {
    PORTD=0b10010011; //RS och CS2 ändras till låga.
    PORTB=value;
    toggle_E();
    PORTD=0b11010001; //PORTD tbx till utgångsläget
}

void write_data_right(char val) {
    PORTB=val;
    PORTD=0b11011001; //CS1 till låg
    toggle_E();
    PORTD=0b11010001; //PORTD tbx till utgångsläget
}

void write_data_left(char val) {
    PORTB=val;
    PORTD=0b11010011; //CS2 till låg.
    toggle_E();
    PORTD=0b11010001; //PORTD tbx till utgångsläget
}

void toggle_E(void) {
    PORTD |= (1<<PIND7); //PD7 sätts till 1
    PORTD &= ~(1<<PD7); //PD7 sätts till 0
    PORTD |= (1<<PIND7); //PD7 sätts till 1
}

void set_dot_right(char x, char y, char val) {
    write_cmd_right(y); //koordinat för prick i y-led
    write_cmd_right(x); //koordinat för prick i X-led
    write_data_right(val);
}

void set_dot_left(char x, char y, char val) {
    write_cmd_left(y); //koordinat för prick i y-led
    write_cmd_left(x); //koordinat för prick i X-led
    write_data_left(val);
}

void timer1_init(void) {
    TCCR1A=0b00000000;
    TCCR1B=0b00000101; //prescale 1024
    TCNT1=57734; //sätter initialt värde.. dvs den behöver bara räkna
//häriifrån till 65.355 typ
    TIMSK=0b0000100; //overflow interrupt enable
}

void adc_init(void) {
    ADCSRA=0b10001100; //ADC sätts på. Interrupt möjliggörs. prescaler
//sätts till 16
}

void create_frame_left(void) {

```

```

        for(int y=64;y<128;y++) { //översta linjen
            int x=184;
            set_dot_left(x,y,1);
            _delay_ms(10);
        }

        for(int y=64;y<128;y++) { //nedersta linjen
            int x=191;
            set_dot_left(x,y,128);
            _delay_ms(10);
        }

        for(int x=184;x<192;x++) { //vi anger PORTB i decimalform. 184
//motsvarar att vi ska skriva x-adressen 000.
            int y=64;
            set_dot_left(x,y,255);
            _delay_ms(10);
        }
    }

    void create_frame_right(void) {
        for(int y=64;y<128;y++) { //översta linjen
            int x=184;
            set_dot_right(x,y,1);

            _delay_ms(10);
        }

        for(int y=64;y<128;y++) { //nedersta linjen
            int x=191;
            set_dot_right(x,y,128);
            _delay_ms(10);
        }

        for(int x=184;x<192;x++) { //vi anger PORTB i decimalform. 184
//motsvarar att vi ska skriva x-adressen 000.
            int y=64;
            set_dot_right(x,y,255);
            _delay_ms(10);
        } //xxxx

        for(int x=184;x<192;x++) { // höger linje,
            int y=127;
            set_dot_right(x,y,255);
            _delay_ms(10);
        }
    }

    char convert_x(char kelvTemp) {
        if(kelvTemp< 56){
            return 184;
        }
        if(kelvTemp>= 56 && kelvTemp<64){
            return 185;
        }
        if(kelvTemp>= 64 && kelvTemp<72){
            return 186;
        }
    }

```

```
    if(kelvTemp>= 80 && kelvTemp<88){  
        return 187;  
    }  
    if(kelvTemp>= 88 && kelvTemp<96){  
        return 188;  
    }  
    if(kelvTemp>= 96 && kelvTemp<104){  
        return 189;  
    }  
    if(kelvTemp>= 104 && kelvTemp<112){  
        return 190;  
    }  
    if(kelvTemp>= 112){  
        return 191;  
    }  
}
```

```
char convert_value(char value) {  
    if(value%8 == 0) {  
        return 1;  
    }  
    if(value%8 == 1) {  
        return 2;  
    }  
    if(value%8 == 2) {  
        return 4;  
    }  
    if(value%8 == 3) {  
        return 8;  
    }  
    if(value%8 == 4) {  
        return 16;  
    }  
  
    if(value%8 == 5) {  
        return 32;  
    }  
    if(value%8 == 6) {  
        return 64;  
    }  
    if(value%8 == 7) {  
        return 128;  
    }  
}
```

```
int main(void) {  
    init();  
    clear_display();  
    adc_init();  
    timer1_init();  
    create_frame_left();  
    create_frame_right();  
    sei();  
    while(1) {  
  
    }  
}
```

```
ISR(TIMER1_OVF_vect) {
    TCNT1=57734; //startvärde för timern
    if(count%2==0) {
        ADMUX = 0b00100000;
    } else {
        ADMUX = 0b01100001;
    }
    //Starta A/D-omvandlaren
    ADCSRA = 0b11001100; //omvandling påbörjas
    TIFR=0b00000100; //infogar en overflow flag.
    count++;
    if(count>=1000000) {
        count=0;
    }
    //sammanfattning: startvärde för timern sätts. Var och en av sensorerna konverterar i
    //varannan interrupt. Flagga sätts
}

ISR(ADC_vect) {
    //läs A/D
    if(count%2==0) {
        adValOut=ADCH; //kontrollera vilken som är inne resp ute
    } else {
        adValIn=ADCH;
    }
    ADCSRA=0b10011100; //interrupt flag har satts
    set_dot_left(convert_x(adValIn),1,convert_value(adValIn));
    set_dot_right(convert_x(adValOut),0,convert_value(adValOut));
}
```