

Digitala projekt, EDI021 Rapport  
Handledare: Bertil Lindvall

Zoltan Darvas (dt06zd8)  
Nicklas Nidhogg(dt06nn8)

2010-03-01

# 1 Abstract

I detta project har vi skapat ett "huslarm" där det finns två stycken sensorer, en analog och en digital. Den analoga sensorn känner av ett föremåls temperaturskillnad mot omgivningen, och den digitala sensorn reagerar på en av- och på signal. I denna rapport kommer vi att bearbeta de olika komponenter vi använt samt hur vi skapat larmet med hjälp av vår AVR mega16 processor.

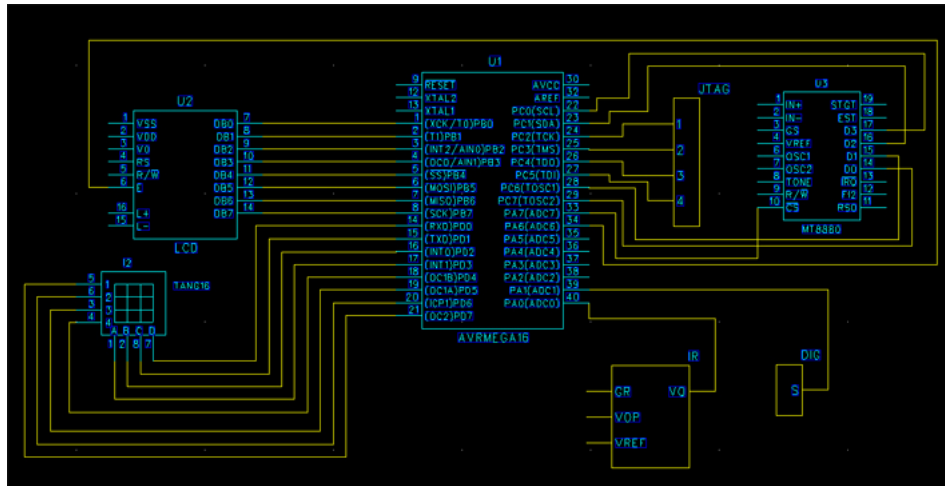
# Innehåll

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Inledning</b>	<b>3</b>
<b>3</b>	<b>Blockschema</b>	<b>4</b>
<b>4</b>	<b>Kravspecen</b>	<b>5</b>
4.1	Funktionella krav . . . . .	5
4.2	Icke-funktionella krav . . . . .	5
<b>5</b>	<b>Konstruktion</b>	<b>5</b>
<b>6</b>	<b>Hårdvara</b>	<b>6</b>
6.1	Processor . . . . .	6
6.2	Display . . . . .	6
6.3	Knappssats . . . . .	6
6.4	Digitalsensor . . . . .	6
6.5	Analogsensor . . . . .	6
<b>7</b>	<b>Mjukvara</b>	<b>6</b>
<b>8</b>	<b>Problem under konstruktionen</b>	<b>6</b>
<b>9</b>	<b>Slutsats</b>	<b>7</b>
<b>A</b>	<b>C-Kod</b>	<b>8</b>
A.1	Huvudprogram . . . . .	8
<b>B</b>	<b>Bilder</b>	<b>10</b>
B.1	Sekvens (huvudprogram) . . . . .	10
B.2	Konstruktionen . . . . .	11

## 2 Inledning

Med en stor optimism trodde vi att detta skulle vara ett lätt projekt att skapa, dock så insåg vi snabbt att vi inte hade så stora kunskaper inom den hårdvara som vi skulle använda oss av. Så det blev en massa sökande och läsande i olika informationsblad samt hemsidor. Dock fixade vi den praktiska delen med att bygga ihop allt samt att linda och löda kablar ganska enkelt.

### 3 Blockschema



Figur 1: Ett blockschema där man ser delarna i konstruktionen

## 4 Kravspecen

När det kom till att göra en kravspecifikation var vi inte särskilt klara med vad vårt larm skulle ha för finesser. Men efter en kort diskussion kom vi fram de följande krav som skulle uppfyllas.

### 4.1 Funktionella krav

- När den analoga givaren detekterar en temperatursförändring som motsvarar  $5^{\circ}\text{C}$  mot omgivningen ska larmet utlösas.
- När en digital givare bryts skall en timer starta.
- Det ska finnas möjlighet att kunna sätta på/stänga av larmet.
- Möjligheten att kunna ha olika sektioner aktiva oberoende av varandra.
- Använda en kod för aktivering/deaktivering av larmet.
- När larmet utlöses ska tiden loggas.

### 4.2 Icke-funktionella krav

- När den digitala givaren bryts, skall en timer på 30 sek starta för att möjliggöra deaktivering av larmet.
- När den analoga givaren detekterar en temperatursförändring som motsvarar  $5^{\circ}\text{C}$  mot omgivningen ska larmet utlösas inom maxtiden av en sekund.

## 5 Konstruktion

Vi valde att konstruera vårt larm i form av en display, en knappsats, en analog givare samt en digital givare. Meningen med givarna vi valt är att den digitala givaren, som då fungerar så att den ger en signal som är antingen "på" eller "av" ska användas vid en dörr. Denna ska då kunna läsa av om dörren är stängd eller inte.

Däremot kommer den analoga givaren fungera som ett vakande öga i ett rum tex, då denna reagerar på temperaturskillnader i omgivningen. Denna kan då övervaka ett rum och upptäcka ifall någon går in i rummet eller inte. Givetvis är förutsättningarna att givaren ska upptäcka denna person/föremål sådan att personen/föremålet måste skilja  $5^{\circ}\text{C}$  från omgivningen.

Knappsatsen kommer att användas som en kontroll för att styra vårt larm. Med denna kommer man kunna stänga av och sätta på larmet samt att välja vilken sektion av larmet som ska vara igång.

Displayen i sig fungerar som en feedback monitor över vad man gör, samt i vilket läge larmet befinner sig i. Som tex om det är aktivt eller inte, vilken sektion som är larmad samt om larmet har utlösts eller inte.

## 6 Hårdvara

### 6.1 Processor

Processorn vi valde är en AVR Mega16, denna processor valde vi för att den räcker gott och väl till att styra vårt larm. Den har en klockfrekvens på 16MHz, ett minne på 1Kb RAM samt ett internt flashminne på 16Kb, och även 32 programmerbara I/O pinnar.

### 6.2 Display

Displayen är av 2x16 punkterstyp som är fullt programmerbar. Denna fungerar som en feedback punkt i vårt larm där man kan läsa av olika sorters information.

### 6.3 Knappssats

En vanlig 16 bitars knappssats med 4x4 system.

### 6.4 Digitalsensor

Den digitala sensorn är av på och av typ, då den endast registrerar om det går ”information” eller inte genom den.

### 6.5 Anlogsensor

En IR sensor från Siemens med betäckningen PID11. Denna sensor mäter temperaturen i omgivningen och skickar ut en signal om något inom denna omgivning avviker från temperaturen i omgivningen med mer än 5°C.

## 7 Mjukvara

Vi använde oss av C-Kod för att skriva vårt program som ska styra vårt larm. Programmet fungerar så att det styr alla komponenter i vår konstruktion, så som av och påslagning av larmet, styrning av de olika givarna osv. Själva koden hittar ni i vårt appendix.

## 8 Problem under konstruktionen

Konstruktionen hade vi rätt klart för oss hur den skulle se ut, dock stötte vi på lite missöden under vårt bygge. Först och främst blev det förmodligen en kortslutning i vår krets så att processorn brann upp. Det var dock ganska simpelt att ordna då det var bara blev tvungna att byta ut våran processor.

Nästa problem vi stötte på var att koppla in våran digitala sensor. Men efter lite klurande förstod vi hur simpelt det egentligen var att koppla in den. Man fick helt enkelt skicka en *etta* på ena benet i forma av en ström, och sedan kontrollera om man fick ut denna *etta* eller inte på det andra benet.

Ytterligare ett problem som vi stötte på var att våran kappsats inte ville fungera på rätt sätt. Felet är att första hela kolumnen inte fungerar som den ska, dock har vi enligt oss själva skrivit *koden* till knappsatsen på rätt sätt då resterade knappar fungerar ypperligt.

## 9 Slutsats

Detta projekt har lärt oss mycket hur man går tillväga för att färdigställa en prototyp. Vårt larm fungerar på ett bra sätt som kan utvecklas vidare till en mer avancerad produkt.

Som larmet är idag fungera det bra med de givare vi valt att använda. Givarna reagerar på ett sett som var förväntat och helhetskonstruktionen samarbetar väl.

Slutligen har vi under denna tid som vi arbetat med projektet lärt oss en hel del av de hårdvaror vi valde att arbeta med. Dock har det jobbigaste varit att planera tiden väl, man blir lätt tidsoptimist när man jobbar med denna sortens projekt.

Summa summarum så lyckades vi knyta ihop projektet på ett bra sätt.



## A C-Kod

### A.1 Huvudprogram

```
int larmed = 0, do_alarm = 0;
unsigned char elapsedseconds = 0;
char true_code[4];
char try_code[4];
char key;
int index;

init_time(){
    TCCR1B |= ((1 << CS10) | (1 << CS11));
}

init_code(){
    true_code[0] = 2;
    true_code[1] = 3;
    true_code[2] = 6;
    true_code[3] = 7;
}

while(1){

    key = get_key();

    if(key == 4){
        larmed = 1;
    }

    if(larmed && DIG_check() != 0){
        do_alarm = 1;
        index = 0;
        elapsedseconds = 0;
        TCNT1 = 0; //reset the timer

        while(elapsedseconds < 30 && do_alarm){
            if (TCNT1 >= 15624) {
                TCNT1 = 0;
                elapsedseconds++;
            }

            key = get_key();

            if(key > 0){
                if(key == true_code[index]){
                    index++;
                    LCD_write("*");
                }
            }
        }
    }
}
```

```

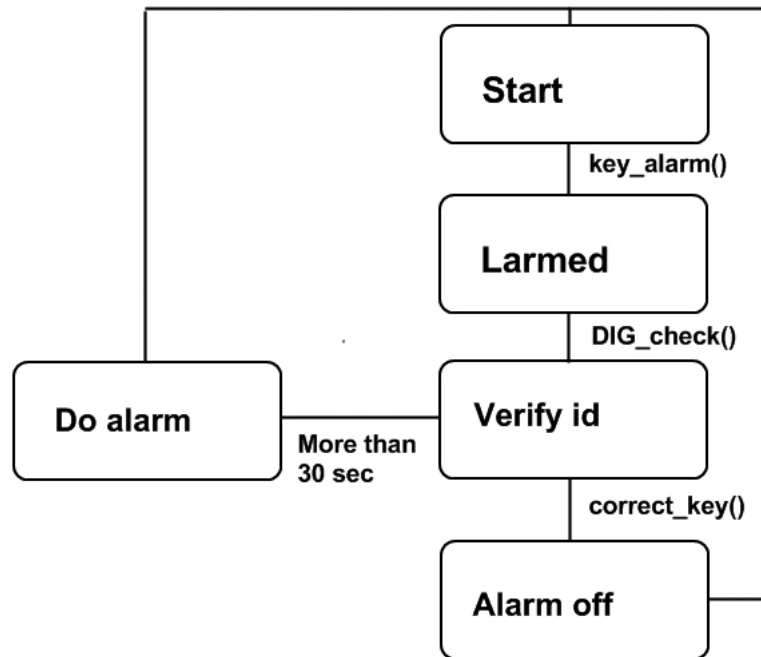
    }
    else{
        index = 0;
        LCD_clear();
    }
    if(index == 4){
        do_alarm = 0;
        LCD_clear();
        LCD_write("ALARM_OFF");
    }
}

}
if(do_alarm){
    LCD_clear();
    LCD_write("INTRUDER");
}
else
    larmed = 0;
}
}
}

```

## B Bilder

### B.1 Sekvens (huvudprogram)

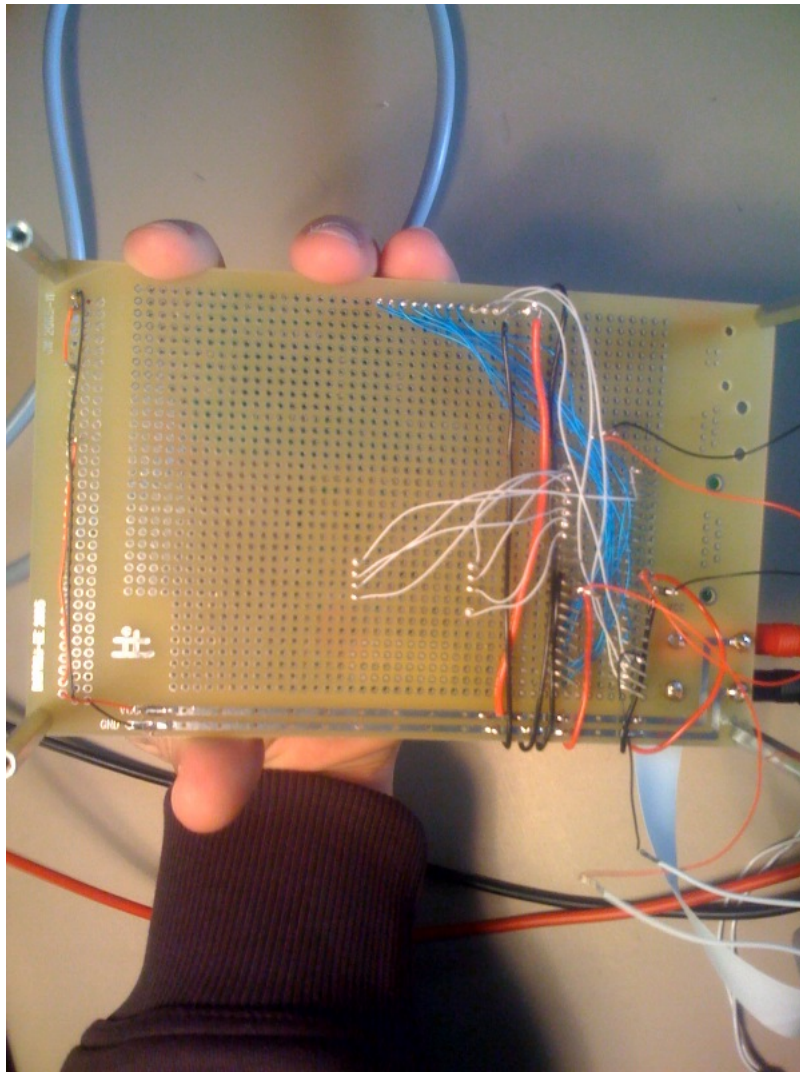


Figur 2: Sekvensen som beskriver huvudprogrammets förlopp

## B.2 Konstruktionen



Figur 3: Konstruktionen ovanifrån



Figur 4: Konstruktionen underifrån