

2015

Larmcentral

Digitala Projekt

Cecilia Olsson & Erika Björck

Handledare: Bertil Lindvall

Inledning

Denna rapport är en sammanställning av projektet i kursen Digitala Projekt. Rapporten innefattar metod, kravspecifikation, vilken hårdvara som valts och en diskussion. Dessutom finns en bilaga med den kod som använts för att programmera larmcentralen.

Syftet med projektet var att skapa ett larm anpassat för en orolig sommarhusägare. Larmet är tänkt att varna ägaren om temperaturen i huset blir för hög eller för låg eller om någon oönskad skulle få för sig att försöka öppna en dörr. Varningen sker genom att en diod lyser rött när något av ovanstående inträffar. Ägaren kan sätta på och stänga av larmet med hjälp av sin personliga pinkod.

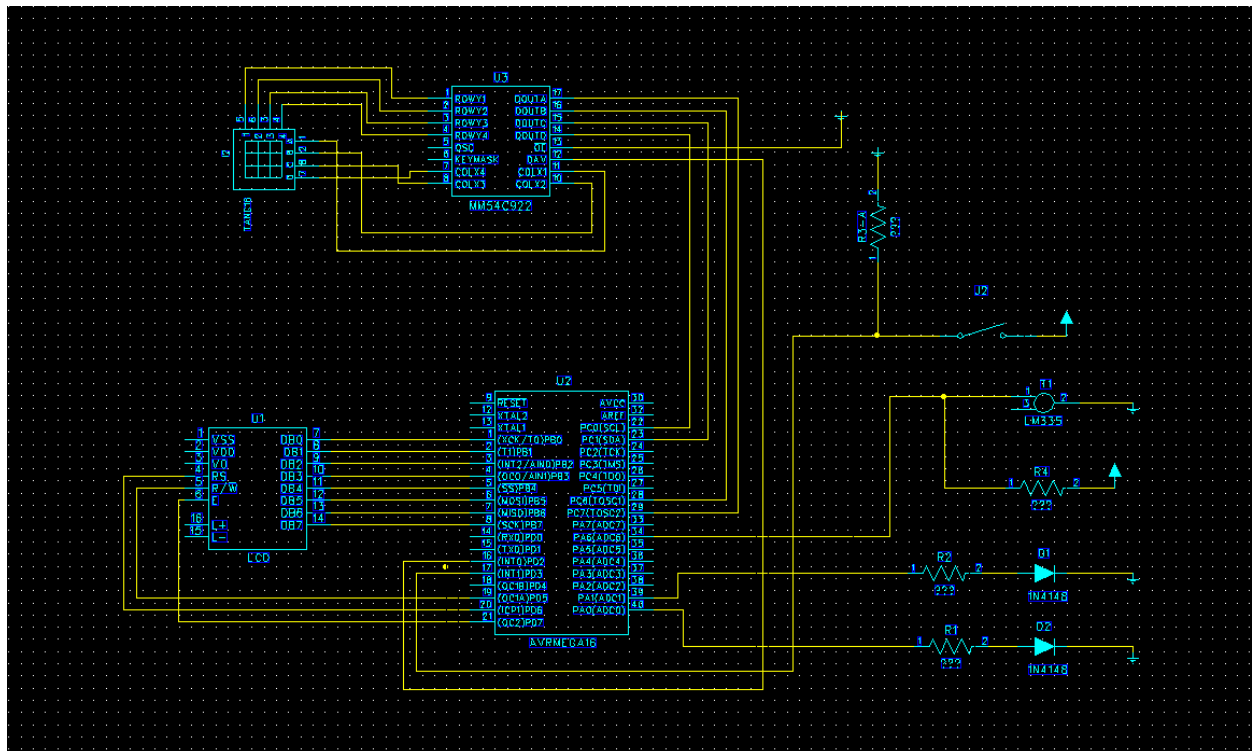
Kravspecifikation

Kravspecifikation innehåller de krav som sattes upp att larmcentralen skulle uppfylla i början av projektet.

- Diod 1 lyser grönt när larmet är påslaget och diod 2 avslagen.
- Diod 2 lyser rött när larmet är utlöst och diod 1 är avslagen.
- Larmet slås på och av med pinkod
- Larmet utlöses då temperaturen överstiger 30 grader
- Larmet utlöses då temperaturen understiger 10 grader
- Larmet utlöses vid knapptryck på knappen (symboliserar att en dörr öppnas)
- Display visar pinkoden.

Hårdvara

Kopplingschema



Processor

Som grund till larmet har en *ATmega16 High-Performance AVR 8-bit Microcontroller* använts. Denna processor har fyra olika portar (A, B C och D) och 40 pinnar.

Display

Displayen (*SHARP Dot-Matrix, LCD Units Alfanumerisk teckendisplay*) används för att underlätta för användaren när larmet ska sättas på eller stängas av genom att displayen visar vilken PIN-kod som har skrivits in, samt ger information om man har slagit in fel PIN-kod.

Knappsats

Knappsatsen har 16 knappar (fördelat på fyra rader) och används för att användaren ska kunna slå in sin PIN-kod.

Switch

Switchen används för att symbolisera att en dörr öppnas. När knappen trycks ner utlöses larmet.

Termometer

Termometern som används är en *LM335 Kelvin temperature sensor*. Termometern ger en spänning som är proportionell mot temperaturen (mätt i Kelvin) som sedan omvandlas i mjukvaran till en temperatur i Celsius.

LED-lampor

En röd och en grön LED-lampa används. Den gröna signalerar om larmet är igång och den röda lampan lyser när larmet är utlöst.

Mjukvara

Mjukvaran är uppbyggd kring tre huvudmetoder som hanterar respektive läge som larmet kan vara i (avstängt, igång och utlöst). Koden är byggd så att det är avbrott som genererar olika metoder som sedan byter mellan de tre huvudmetoderna. Avbrotten kommer huvudsakligen från tre olika hårdvarokomponenter; tangentbordet, switchen och temperaturmätaren. Programmet kodades i C.

Metod

Projektet inleddes med en diskussion kring vilka krav prototypen skulle uppfylla och en kravspecifikation upprättades. Utifrån denna kravspecifikation listades vilka delar som var nödvändiga till prototypen och därefter kunde ett kopplingsschema ritas i programmet PowerLogic.

Efter att kopplingsschemat godkänts började bygget utav prototypen, det gjordes med lödning för att hålla delarna på plats, och kopplades ihop genom att trådarna virades runt pinnarna utifrån kopplingsschemat.

Tester utfördes på prototypen i programmet Atmel Studio, där varje del av prototypen testades för sig. Efter att alla delar visat sig fungerande så kodades mjukvaran för larmcentralen, även den i Atmel Studio.

Diskussion

Arbetet har resulterat i en larmcentral som uppfyller de krav som sattes vid projektets start. Vägen dit har dock inte varit problemfri, utan det har varit en hel del utmaningar. Den största utmaningen var att gruppens (nästintill obefintliga) elektronik-kunskaper gjorde det svårt att läsa och tyda datablad, vilken skapade en del förvirring både då kopplingsschema skulle ritas, men även vid programmeringen. En annan utmaning var att ingen av gruppmedlemmarna hade programmerat i C förut, vilket skapade en liten tröskel i början av projektet.

Även om larmcentralen uppfyller de krav som sattes, finns det stor förbättringspotential. Möjliga förbättringar skulle kunna vara:

- Att larmet låter när det utlöses, istället för att en lampa lyser.
- Att göra det möjligt för användaren att välja PIN-kod själv.
- Att låta användaren själv ställa in vid vilka temperaturer som larmet ska utlösas.
- Att kunna fjärrstyra larmet från sin permanenta bostad.
- Att ge larmet inte sätts igång förrän efter en viss tid (t ex en minut), för att ge användaren tid att lämna sommarstugan efter att denne har tryckt in sin PIN-kod.

Sammanfattningsvis har projektet varit väldigt lärorikt och det har varit roligt att arbeta på ett sätt som tidigare inte påträffats i någon annan kurs.

Bilaga 1

Kod

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

//Setup
char p[]={'5','1','1','5'}; //PIN-kod
int mode;

//Temperature
int temp;
int read; //Läser spänning från termometern

//Keypad
int n;
int val;
char r;
char parray[4];

int main(void)
{
    n=0; //vektorns läge
    mode = 0; //avläge
    DDRA = 0b11111111;
    DDRB = 0b11111111;
    char var = DDRC;
    var = var &0b00111100;
    DDRC=var;
    DDRD = 0b11100000;
    MCUCR = 0b00001111;
    GICR = 0b11000000;
    sei();
    start_temp();
    print_cmd(0x38); //Functionset
    print_cmd(0x0F); //Display ON
    print_cmd(0x01); //Clear Display
    print_cmd(0x02); //Return home
    init_mode0();

    while(1){
    }
}

void red_light_on(){
    PORTA=PORTA |0b00000001;
}

void red_light_off(){
    PORTA=PORTA &0b11111110;
}
```

```

void green_light_on(){
    PORTA = PORTA | 0b00000010;
}

void green_light_off(){
    PORTA = PORTA & 0b11111101;
}

/*Knapp, ska sätta utlöst läge*/
void the_switch(){
    init_mode2();
}

/*Set up för termometern*/
void start_temp(){
    ADMUX = 0b01100110;
    ADCSRA = 0b11001000;
    SFIOR = 0b00000000;
}

/*Läser av temperaturen*/
int read_temp(){
    read = ADCH;
    temp = read*(5/255)*100-273;

    if(temp>30){
        init_mode2();
    }
    else if(temp<10){
        init_mode2();
    }
}

/*Läser av vilken tangent som var nertryckt*/
char read_keypad(){
    val = PINC&0b11000011;

    if(val== 0b11000000){
        return '0';
    }
    if(val==0b01000000){
        return '1';
    }
    if(val==0b10000000){
        return '2';
    }
    if(val==0b00000000){
        return '3';
    }
    if(val==0b11000010){
        return '4';
    }
    if(val==0b01000010){
        return '5';
    }
    if(val==0b10000010){
        return '6';
    }
    if(val==0b00000010){
        return '7';
    }
}

```

```

    }
    if(val==0b11000001){
        return '8';
    }
    if(val==0b01000001){
        return '9';
    }
    if(val==0b10000001){
        return 'A';
    }
    if(val==0b00000001){
        return 'B';
    }
    if(val==0b1100011){
        return 'C';
    }
    if(val==0b0100011){
        return 'D';
    }
    if(val==0b1000011){
        return 'E';
    }
    if(val==0b0000011){
        return 'F';
    }
}

/*/Skärm ändras så siffran skrivs */
void print_char(char ch){
    _delay_ms(10);
    PORTD=0b11010111; //RW sätts till 0, RS sätts till 1, E sätts till 1

    PORTB=ch;
    PORTD=0b01010111; //E sätts till 0
    PORTD =0b11010111; //E sätts till 1
}

void print_home_page(){
    print_cmd(0x01); //Clear Display
    print_cmd(0x02); //Return home
    print_char('P');
    print_char('I');
    print_char('N');
    print_char('-');
    print_char('k');
    print_char('o');
    print_char('d');
    print_char(':');
}

void print_wrong(){
    print_cmd(0x01); //Clear Display
    print_cmd(0x02); //Return home
    print_char('F');
    print_char('e');
    print_char('l');
    print_char(' ');
    print_char('P');
    print_char('I');
}

```

```

        print_char('N');
        print_char('!');
        print_char(' ');
        print_char(':');
        print_char('(');
        _delay_ms(1500);
        print_home_page();
    }

void print_cmd(char ch){
    _delay_ms(10);
    PORTD=0b10010111; //RW sätts till 0, RS sätts till 1, E sätts till 1

    PORTB=ch;
    PORTD=0b00010111; //E sätts till 0
    PORTD =0b10010111; //E sätts till 1
}

void print_from_keyboard(){
    _delay_ms(200);
    char r = read_keypad();
    _delay_ms(200);
    print_char(r);
    parray[n]=r;
    n++;
    if(n==4){
        check_pin();
        n=0;
    }
}
/*Kollar pinkoden, den sparade vektorn mot vektorn som hämtat info från tangent*/
check_pin(){
    if(parray[0]==p[0] && parray[1]==p[1] && parray[2]==p[2] &&
parray[3]==p[3]){
        if (mode == 0){
            _delay_ms(100);
            init_mode1();
        }
        else if(mode == 1){
            _delay_ms(100);
            init_mode0();
        }
        else if(mode == 2){
            _delay_ms(100);
            init_mode0();
        }

    } else{
        _delay_ms(500);
        print_wrong();
    }
}
/*Avbrott från tangentbord*/
ISR(INT1_vect){
    _delay_ms(500);
    print_from_keyboard();
    _delay_ms(500);
}

```



```
/*Avbrott från termometern*/
ISR(ADC_vect){
    read_temp();
}
/*Avbrott från switchen*/
ISR(INT0_vect){
    if(mode==1){
        the_switch();
    }
}

init_mode0(){ //Sätter igång AV-läge.
    mode=0;
    print_home_page();
    green_light_off();
    red_light_off();
};

init_mode1(){ //Sätter igång PÅ-läge
    mode=1;
    green_light_on();
    print_home_page();
}

init_mode2(){ //Sätter igång activate mode
    mode=2;
    green_light_off();
    red_light_on();
    print_home_page();
}

}
```