



LUNDS UNIVERSITET  
Lunds Tekniska Högskola

Laboration 2 - Operationsförstärkaren

LTH

# Projektrapport

**Kursnamn:** Digitala System

**Kurskod:** EITA15

**Datum:** 04/05-2022

## Rökdetektor

### Handledare

Bertil Lindvall

Lars-Göran Larsson

### Gruppmedlemmar

Max Persson

Edgar Portela

Eric Portela

Hassan Hussin

## Sammanfattning

Detta är avslutande delen i kursen digitala system EITA15. Projektet gick ut på att konstruera en hårdvara med en mikrokontroller och programmera en fungerande produkt med hjälp av Atmel Studio 7. Slutprodukten blev en rökdetektor som känner av rök och varnar genom att slå på ett larm och visa ett varningsmeddelande på en LCD-display.

Till att börja med var planering en essentiell faktor till att hinna med allt som skulle göras. En veckoplanering skapades för att få en tydlig struktur på arbetet. Planering av när projektgruppen skulle träffas och vad som skulle göras var delar av veckoplaneringen. Därefter konstruerades kretsschemat genom att använda programmet Goodnotes på en Ipad. Atmega 1284 är processorn som användes som sedan programmerades i Atmel Studio 7. Databladet för processorn och andra komponenter användes jämnt genom hela arbetet för att veta var dem olika pinsen är och dess funktioner. En gas sensor, MQ2, användes för att få in värden när den detekterar rök som sedan används vid programmeringen. Resterande komponenter som skulle vara på kretskortet testades. Dem flesta komponenter fungerade som de skulle och därefter löddes dem på kretskortet. Det var lite problem med en buzzer där vi sedan upptäckte att den inte fungerade. Därefter påbörjades programmeringen där koden skrevs i C. Projektarbetet utfördes under 2 månaders tid där slutprodukten blev en fungerande prototyp av en rökdetektor.

## Nyckelord

Atmega1284 - En mikrodator

Kopplingsschema - Schematisk bild över en krets

MQ2 - Gas sensor

## Abstract

The idea of this project is to create a prototype of a smoke detector. The project consisted of planning, hardware construction and C programming in Atmel Studio 7. Working on a bigger project independently enhanced the group members' problem solving skills. A microcontroller, Atmega 1284, was the main component in this project. To get done with all the parts of the project a lot of planning had to be made from the start. Making the schematics for the circuit was the first task that had to be completed. Followed by examining and soldering the components to the circuit and programming them. To summarize the work a website was created with a demonstration of the final product.

The achieved product is a smoke detector which is able to detect smoke from a distance of 15 cm. When the smoke is detected a siren is initiated and a warning message is produced on an LCD-display.

## Keywords

Atmega1284 - Microcontroller

Circuit diagram - Schematics for a circuit

MQ2 - Gas sensor

# Innehållsförteckning

<b>1 Inledning</b>	<b>5</b>
1.1 Bakgrund	5
1.2 Syfte	5
1.3 Kravspecifikation	5
1.4 Målformulering	5
1.5 Motivering av arbetet	5
<b>2 Teknisk bakgrund</b>	<b>6</b>
<b>3 Metod</b>	<b>7</b>
3.1 Planering	7
3.2 Hårdvara	7
3.3 Mjukvara	7
<b>5 Resultat</b>	<b>8</b>
<b>6 Diskussion</b>	<b>9</b>
<b>7 Källförteckning</b>	<b>10</b>
<b>8 Appendix</b>	<b>11</b>
8.1 Planering	11
8.2 Kretsschema	12
8.3 Källkod	12

# 1 Inledning

Rapporten är en sammanfattning och avslutning av projektet i kursen digitala system EITA15. En fungerande produkt ska skapas som mål för projektet. Kunskaper som kursen tillhandahållit testas vid projektets utförande. Genomförande av arbetet har bestått av planering, skapande av kretsschema, programmering och konstruktion av prototyp. Produkten som projektet resulterade i var en rökdetektor.

## 1.1 Bakgrund

Projektet utfördes som en avslutande del av kursen EITA15. Samtliga gruppmedlemmar är framtida högskoleingenjörer och därmed är färdigheter som problemlösningssförmåga, ingenjörsmässigt skrivande och samarbetsvilja viktiga att träna upp. Genom arbetandet har kunskaper inom datorteknik fördjupats samt praktisk erfarenhet av programmering med mikrokontroller.

## 1.2 Syfte

Syftet med projektet är att konstruera en fungerande prototyp, utveckla problemlösningssförmågan men även att sammarbeta med större projekt. Projektet ger inblick i att kunna realisera en fungerande prototyp från en idé. Kunskaper som planering, struktur och samarbetsvilja utvecklas inför framtida projektarbeten. Förväntat resultat är en rökdetektor som varnar vid detekterad rök.

## 1.3 Kravspecifikation

- 1.1 Ska kunna upptäcka rök inom 15 cm avstånd.
- 1.2 Varningsmeddelande på display vid detekterad rök.
- 1.3 Lysdiod blinkar vid detekterad rök.
- 1.4 Larmar med buzzer vid detekterad rök.
- 1.5 Reset knapp som återställer mjukvaran.

## 1.4 Målformulering

Målet med projektarbetet är att kunna realisera en idé och få en fungerande prototyp. Hårdvaran ska kunna detektera rök och varna genom att slå på ett larm, blinka med lysdioder men även få ett varningsmeddelande. En hemsida skapas för att sammanfatta projektarbetet.

## 1.5 Motivering av arbetet

Realisering av en idé var största motiveringen. Kunskaper som planering, samarbetsvilja och driv stärks vid utförandet av arbetet.

## 2 Teknisk bakgrund

Nedan följer en beskrivning av komponenterna som prototypen är uppbyggd av. För att se hur dem är kopplade till varandra finns kretsschemat i [Appendix 7.2](#)

Komponenter:

- **Processor:** AVR ATmega 1284 är en 8-bitars processor. Processorn har fyra portar PA-D och har 40 pinnar.
- **Gas Sensor:** MQ2 är en gas sensor som används för att detektera rök.
- **Buzzer:** kxg1205c används för att varna när sensorn detekterar rök.
- **Lysdiod:** Två röda lysdioder används för att varna när sensorn detekterar rök. Sätts på samtidigt som larmet.
- **NE555:** Integrerad krets som används för att generera en puls till lysdioderna.
- **Resistorer:** Spänningsdelare i disc för resistorer
- **LCD Display:** LCD\_BC1602Y är en 16x2 lcd display som används för att varna om där är någon rök.
- **JTAG:** Atmel-ICE används för att överföra koden från datorn till processorn.

## 3 Metod

### 3.1 Planering

Projektarbetet organiserades genom att genomföra en preliminär veckoplanering. Detta gav arbetet en tydlig struktur att följa.

### 3.2 Hårdvara

Samtliga komponenter löddes fast på kretskortet enligt kretsschemat, se [Appendix 7.2](#). Konstruktion av hårdvara inleddes med atmega1284 och en LCD-display. Displayen krävde konstruktion av en spänningsdelare, samt en potentiometer för att justera kontrasten. Därefter konstruerades en pulsgenerator med hjälp av NE555 för att få en blinkeffekt på två lysdioder.

När alla komponenter löddes fast på kretskortet modellerades en kabinett i SolidWorks som håller i kretskortet och samtidigt gömmer undan alla kablar så de inte syns.

### 3.3 Mjukvara

Programvaran programmerades i C, genom vilken den insamlade datan bearbetas. Programvaran kan i huvudsak delas in i tre olika delar. Initialisering av relevanta I/O portar och deras tillhörande register. En inledande period för inläsning av data för att erhålla ett referensvärde (för gassammansättning i rummet), och slutligen en loop som jämför referensvärdet med senast detekterad data. Inläsning av rökdetektorns spänningsvärden påbörjas cirka 5 sekunder efter att kretsen satts igång för att utesluta volatila mätpunkter och därmed säkerställa att sensorn stabiliserat sig. Därefter detekteras fem mätpunkter med en sekunds fördröjning mellan varje för att slutligen erhålla referensvärdet, vilket är ett medelvärde av de detekterade mätpunkterna. När referensvärdet erhållits jämförs detta med nyinkommen data från sensorn. En procentuell avvikelse från referensvärdet sattes som ett tröskelvärde (villkor), vilket vid överskridning resulterar i att larmsystemet utlöses.

## 5 Resultat

Resultatet blev en fungerande prototyp som varnar vid detekterad rök. Produkten känner av rök vid maximalt 15 cm avstånd under testförhållanden. Två lysdioder är kopplade så att de blinkar varierande samtidigt som en buzzer sänder ut ljud som larm. Det finns även en LCD display som visar ett varningsmeddelande när produkten detekterar rök och en reset knapp som återställer produkten. Alla krav är därmed uppfyllda.



## 6 Diskussion

Projektets resultat påvisar att målformuleringen och kravspecifikationen är uppnådd. Arbetet flöt på till stor del felfritt. Det största problem vi hade var att vi använde två redan upptagna pins (PC5 och PC4), när vi kopplade LCD-displayen till atmega. Dessa är förkopplade till debuggern, vilket gjorde att LCD-displayen inte fungerade som väntat. Efter att detta problem åtgärdats fungerade redan implementerad kod för LCD-displayen.

Eftersom vi hade en matningsspänning på 5 volt från spänningsaggregatet och Vcc för LCD-displayen krävde 4.6 volt, var vi tvungna att konstruera en spänningsdelare. Detta krävde korrekt dimensionering av resistorer.

Vi försökte först använda en annan gas sensor MQ-5 som inte är ämnad för rökdetektion. Vi valde efterhand att inte använda denna eftersom spännings utslaget på rök var både lågt och opålitligt.

Användandet av ADC-omvandlare krävde att vi kopplade en kondensator till AREF-pin för att minska eventuellt brus i signalen. MQ-2 som användes i projektet ger en varierande utspänning beroende på på gassammansättningen i rummet. Denna spänning varierade mellan 0.2-0.3 volt, vilket är ett lågt värde i relation till referensspänningen på 2.5 volt. Detta innebar att vi var tvungna att finjustera referensvärdet nog för att undvika att larmet utlöstes vid fel tillfälle, alternativt inte var känslig nog att detektera rök.

För att implementera en reset funktion som återställde register och PC till 0 använde vi en watchdog timer. Denna var ganska strulig då vi hamnade i en reset-loop. Detta löstes genom att manuellt nollställa en flagga i watchdog timers register efter varje reset.

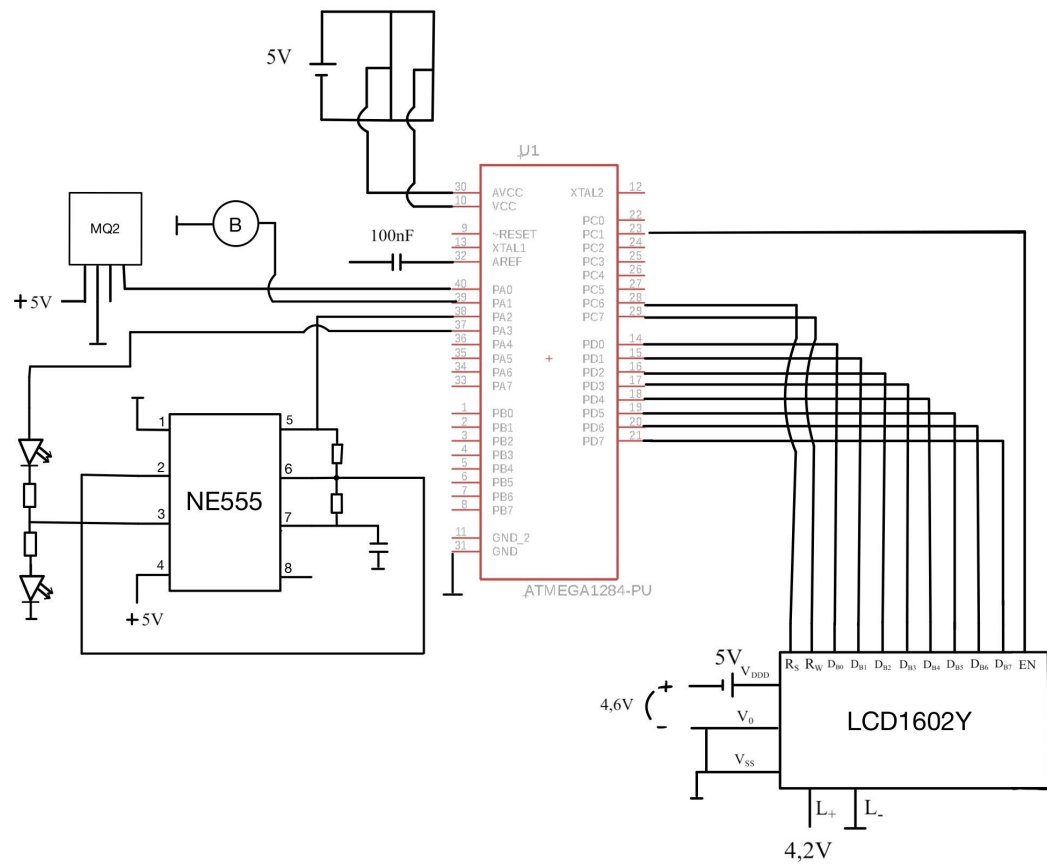
## 7 Appendix

### 7.1 Planering

Att göra	v.13	v.14	v.15	v.16	v.17	v.18	v.19	v.20	v.21	v.22
Kopplingsschema	Bestämma komponenter och börja med kopplingsschema	Fortsätta med kopplingsschemat och bli färdiga								
Konstruktion av hårdvara		Börja bygg	Fortsätt bygg	Fortsätt bygg	Fortsätt bygg	Fortsätt bygg/testa	Fortsätt bygg/testa	Finslipa/testa		
Programkod för prototypen		Börja skriv	Fortsätt skriv	Fortsätt skriv	Fortsätt skriv	Fortsätt skriv	Fortsätt skriv	Klart		
Webbsida			Börja jobba på design	Bli färdiga med första utkast			Kolla över webbsidan ifall något behövs fixas	Färdigställ webbsidan		
Rapport				Påbörja rapport	Fortsätt skriva	Fortsätt skriva	Fortsätt skriva	Fortsätt skriva	Inlämning av rapport 23/5	
Opponering									Gå igenom opponenterns arbete	2/6 Klart
Presentation								Börja med presentation	Färdigställ powerpoint	2/6 Klart
Projektdemo										2/6 Klart

Figur 1: Projektets veckoplanering.

## 7.2 Kretsschema



Figur 2: Kretsschema över rökdetektor.

## 7.3 Källkod

/\*

\* Projekt\_rok\_detektorl.c

\*

\* Created: 2022-04-14 10:44:55

\* Author : ma7646pe-s

\*/

```
#define F_CPU 16000000UL
```

```
#define DATA PORTD
```

```
#define ctrl PORTC
#define en PORTC1
#define rs PORTC6
#define rw PORTC7
#define res PINA4
#include <util/delay.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
```

```
uint16_t ref = 0;
uint8_t read_value = 0;
```

```
void reset(){
    wdt_enable(WDTO_250MS);
}
```

```
void buzzer(){
    DDRA |= 0b00000010;
    PORTA |= 0b00000010;
}
```

```
void lights(){
    DDRA |= 0b00001100;
    PORTA |= 0b00001100;
}
```

```
void adc_init(){
```

```

DDRA = 0b11111110 & DDRA; //Set pin ADC0 as input pin

PORTA |= 0x01; // Activate pull-up resistor to avoid floating value

ADMUX |= 0b10100000;    //[7,6] reference voltage AVcc, [2.5V] ADC left
adjusted,[4,0] Select ADC0 single ended input

ADCSRA |= 0b10000111; // [7] Enable ADC, [2,0] ADC prescaler division factor =
128

}

```

```

uint16_t adc_read(){

    ADCSRA |= 0b01000000;
    while((ADCSRA >> 6) == 0b00000011){

    }

    read_value = ADCH;
    return read_value;

}

```

```

void ref_value(){
    _delay_ms(5000);

    for(int i = 0; i < 5; i++){
        ref = ref + adc_read();
        _delay_ms(1000);
    }

    ref = ref / 5;
    ref = ref * 1.02;

}

```

```

void LCD_cmd(unsigned char cmd){

```

```

    DATA = cmd;
    ctrl = (0 << rs)|(0 << rw)|(1 << en);
    _delay_ms(1);
    ctrl = (0 << rs)|(0 << rw)|(0 << en);
    _delay_ms(50);
    return;
}

```

```

void LCD_init(void){

    DDRC |= 0b11000110;
    DDRD = 0xff;

    LCD_cmd(0x38); //Function set
    _delay_ms(1);

    LCD_cmd(0x01);    //Clear display
    _delay_ms(1);

    LCD_cmd(0x0f); //Display on
    _delay_ms(1);

    LCD_cmd(0x02); //Entry mode
}

```

```

void LCD_write(unsigned char chr){
    DATA = chr;
    ctrl = (1 << rs)|(0 << rw)|(1 << en);
    _delay_ms(1);
}

```

```

    ctrl = (0 << rs)|(0 << rw)|(0 << en);

    _delay_ms(50);

    return;
}

void LCD_string(const char *str){
    int i = 0;

    while(str[i]!='\0'){
        LCD_write(str[i]);

        i++;
    }

    return;
}

void is_it_burning_questionmark(){
    if(adc_read() > ref){
        LCD_cmd(0x01);

        _delay_ms(1);

        LCD_string("* Warning Fire *");

        buzzer();

        lights();

        while(1){
            if(PINA >> res == 0x01){
                PORTA = PORTA & 0b00000001;

                reset();
            }
        }
    }
}

```

```
int main(void)
{
    MCUSR = 0;
    wdt_disable();
    cli();
    LCD_init();
    LCD_string("*** No Fire! *** test");

    adc_init();
    ref_value();

    while (1) {
        is_it_burning_questionmark();
    }
}
```