



LUNDS TEKNISKA  
HÖGSKOLA  
Lunds universitet

**EDI021 Digitala projekt**

# Rapport

## LARMSYSTEM

**Utförd av:**

Niklas Eklund E03  
David Olsson E04

**Handledare:**

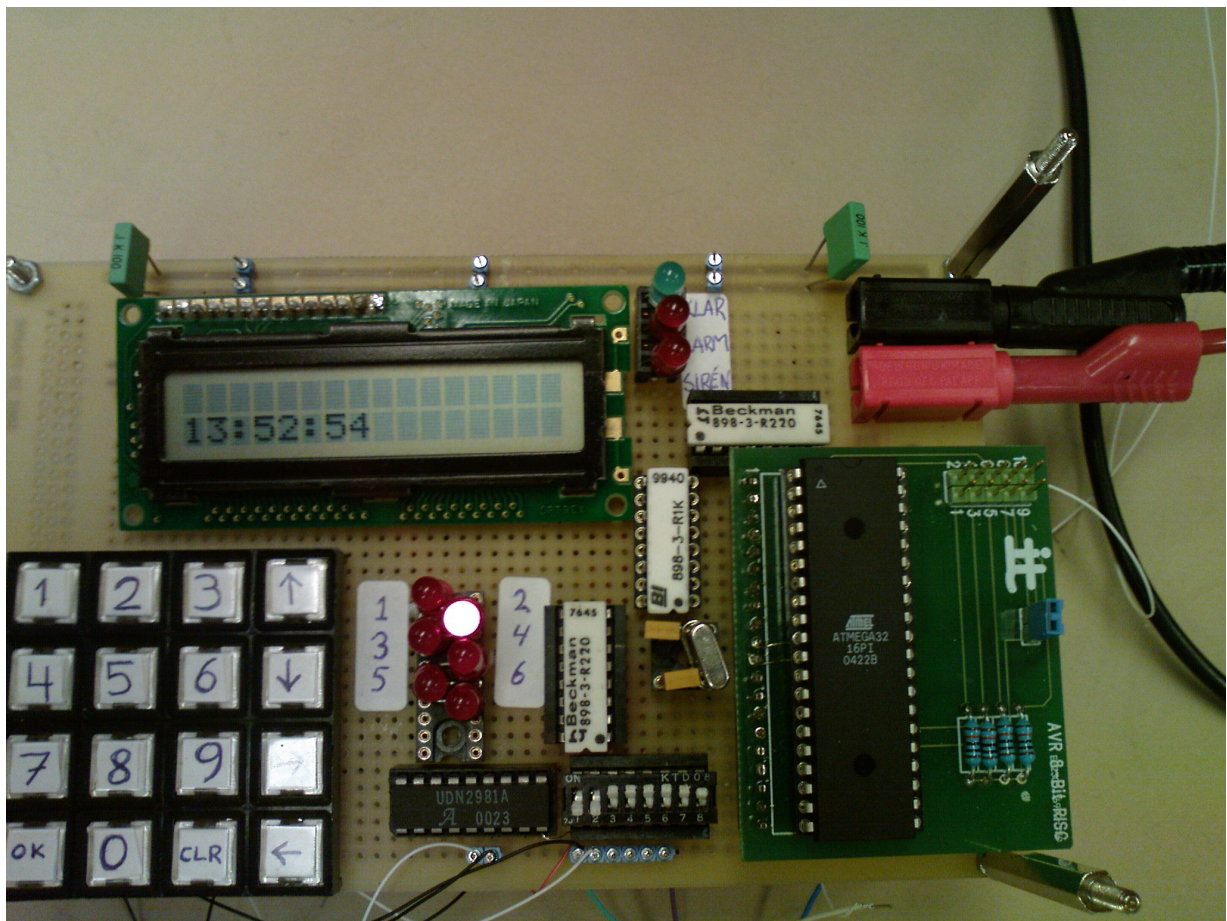
Bertil Lindvall

**Inlämnad:**

2007-12-04

## Abstract

The market for security devices, both for consumers and companies has grown exponential for the last years and it's likely to continue that way for several more years. Therefor designing an alarm system is of immediate. This rapport is a part of the course “EDI021 Digital Projects” and describes our construction of a simple alarm system based on the 8-bit AVR microcontroller ATmega 32. The alarm system has an easy controlled interface and is able to keep surveillance over 6 different detectors. The system is also password protected and capable of keeping an alarm log.



# Innehållsförteckning

<b>Abstract</b>	<b>2</b>
<b>Innehållsförteckning</b>	<b>3</b>
<b>1 Inledning</b>	<b>4</b>
<b>2 Kravspecifikation</b>	<b>4</b>
<b>3 Hårdvara</b>	<b>4</b>
3.1 AVR ATmega 32 med JTAG-modul	4
3.2 4*4 knappsats	4
3.3 16*2 alfanumerisk LCD-display	5
<b>4 Konstruktion</b>	<b>5</b>
4.1 PORTA	5
4.2 PORTB	5
4.3 PORTC	5
4.4 PORTD	5
<b>5 Mjukvara</b>	<b>6</b>
<b>6 Resultat och diskussion</b>	<b>7</b>

# 1 Inledning

Kursen EDI021 Digitala Projekt är precis som namnet antyder en projektkurs där målet är att bygga och programmera en digital konstruktion. Kursen knyter samman tidigare kurser såsom programmering och elektronik.

På senare år har säkerhetsindustrin vuxit starkt, och larmsystem har blivit allt vanligare, särskilt i bostäder. Det kändes därför rätt att konstruera ett sådant larmsystem med ett lättnavigerat och användarvänligt interface. Olika brytande detektorer av branschstandard ska kunna anslutas, t.ex. IR-detektorer, rökdetektorer och magnetdetektorer som används för att övervaka dörrar och fönster.

## 2 Kravspecifikationer

Innan konstruktionen av larmet började sattes målet att nedanstående kriterier skulle vara uppfyllda. Det visade sig under konstruktionens gång att samtliga punkter inte kunde realiseras på det sätt som planerats.

- Ett användarvänligt interface med ett menysystem som styrs via knappsatsen och visas på LCD-displayen.
- Lysdioder som fungerar som indikationslampor, d.v.s. visar larmets status.
- Lösenord krävs för att styra larmet.
- Möjlighet att ändra lösenordet och spara det på ett sådant sätt att det är kvar även om matningsspänningen till larmet kopplas bort.
- Larmet ska kunna kontrollera 8 st brytande detektorer.
- Detektorer ska kunna aktiveras/deaktiveras via interfacet.
- 8 st Lysdioder som visar om någon aktiverad detektor är bruten.
- Larmet ska kunna visa logg på vilka detektorer som utlösts och när detta inträffade.
- Larmet ska ha en klocka med acceptabel noggrannhet.
- Tiden ska kunna visas på LCD-displayen.
- Möjlighet att ställa klockan.

## 3 Hårdvara

Det slutgiltiga konstruktionen bestod av nedanstående huvudkomponenter. Resistorer, transistorer, kristaller, brytande switchar (dessa kan ersättas med brytande detektorer) och lysdioder nämns inte, men deras placeringar och värden kan ses på kretsschemat.

### 3.1 AVR ATmega 32 med JTAG-modul

En 8-bitars enchipdator med 40 pinnar och 32 KByte programminne inbyggd i en PDIP-kapsel. 32 av dessa pinnar fungerar som I/O enligt CMOS-standard. JTAG-modulen möjliggör debugging vilket underlättar programmering, och särskilt felsökning. Nackdelen med JTAG-modulen är att den tar fyra pinnar i anspråk vilka annars kunde användas till annat.

### 3.2 4\*4 Knappsats

En knappsats med 16 st knappar ordnade i fyra rader och fyra kolumner. Då en knapp trycks ner kortsluts den rad och den kolumnen som korsar knappens placering.

### 3.3 16\*2 alfanumerisk LCD-display

Denna display är HD47780-kompatibel och kan styras i både 4 och 8 bitars läge.

## 4 Konstruktion

Larmet byggdes på ett experimentkort med ledande öar. I stället för att löda gjordes kopplingarna genom virning. Samtliga komponenter placerades i IC-hållare anpassade för virning. Kretsschemat visar den slutliga konstruktionen

### 4.1 PORTA

Från den ursprungliga kravspecifikationen kan läsas att det var planerat att 8 st brytande detektorer skulle användas. Det var då planerat att LCD-displayen skulle styras i 4 bitars läge. Vi hade problem med just detta, och efter många försök bestämde vi oss för att använda 8 bitars läge i stället eftersom detta var lättare. Det blev nu för få pinnar över för att använda 8 st detektorer som vi planerat från början. Vi bestämde oss därför för att använda 6 st detektorer i stället. PA0-PA5 är inportar och kopplade till jord via de brytande detektorerna och hålls låga så länge detektorerna är slutna. Skulle någon av dessa brytas blir pinnen i fråga hög via en resistor på  $1k\Omega$  som är kopplad mellan pinnen och +5V. Enligt kravspecifikationen skulle det finnas en lysdiod för varje detektor som indikerar om detektorn brutits. Detta löste vi genom att använda en transistorarray av typ 3082 vilken består av NPN-transistorer. PA0-PA5 kopplades till varsin transistor (till basen). Kollektorn kopplades till +5V via en resistor och en lysdiod. Emittern kopplades till jord. På så sätt fungerar transistorerna som spänningsföljare och leder ström via kollektorn (lysdioden lyser) då basen är hög, d.v.s. då en detektor brutits. PA6-PA7 sattes till utportar och indikerar via lysdioder om larmets status är "Klar" eller "Larm".

### 4.2 PORTB

Samtliga pinnar på PORTB användes till knappsatsen. PB0-PB3 sattes till inportar, medan PB4-PB7 sattes till utportar. Genom att använda polling och pinnarnas inbyggda pull-up-resistorer kan man avgöra vilken knapp som är nedtryckt.

### 4.3 PORTC

PC0, PC1 och PC7 är utportar och används till LCD-displayens styrbitar. PC2-PC5 är reserverade för JTAG. PC6 är utport och indikerar via en lysdiod om sirenen är i gång.

### 4.4 PORTD

Samtliga pinnar på PORTD är utportar och används till LCD-displayens databitar.

## 5 Mjukvara

Programmet är skrivet i C och kompilerat med AVR-GCC. Som utvecklingsmiljö användes AVR Studio 4 vilket underlättade arbetets gång eftersom det har JTAG-interface som möjliggör debugging.

Programmet är anpassat efter att själva larmet har tre tillstånd; avlarmad, larmad och inloggad. Programmet körs i en huvudloop där den läser av detektorerna och knappsatsen. Är larmet i tillståndet avlarmat samtidigt som samtliga aktiverade detektorer är slutna och det inte finns några larmhändelser att kvittera kommer "klar"-dioden att lysa. Det är då i sin ordning att larma. Om larmet är i tillståndet larmat och en aktiverad detektor löses ut kommer "larm"-dioden lysa. Denna kommer lysa så länge som det finns larmhändelser som inte är kvitterade. Vid larmhändelse kommer även sirenen att tjuta i en förinställd tidsperiod eller tills man loggar in.

Om programmet läser av att knappen "OK" är intryckt på knappsatsen förflyttar sig pekaren till subrutinen `angeKod()` där man kommer vidare till huvudmenyn om man matar in rätt kod på fyra siffror. Denna kod måste stämma överens med root-lösenordet som finns lagrat i programminnet eller ett användarlösenord som finns sparad i EEPROM-minnet.

Väl inne i huvudmenyn ändras larmet tillstånd till inloggad. Här finns det 5 st undermenyer; "1. Larm till", "2. Byt kod", "3. Larmlista", "4. Sektion" och "5. Byt tid".

"Larm till" gör att larmet ändrar tillstånd till larmad.

"Byt kod" sparar en ny användarkod på fyra siffror vilken lagras i EEPROM-minnet.

"Larmlista" visar information om de 10 senaste larmhändelserna. Här visar vilken detektor som löst ut, vilken tid detta skedde och om händelsen kvitterats. I denna undermeny kan man även kvittera händelserna.

I "Sektion" kan man deaktivera olika detektorer. Samtliga detektorer aktiveras nästa gång man loggar in.

I "Byt tid" kan minuter och timmar ställas. Sekunderna kommer att nollställas.

En interrupt-rutin uppdaterar klockan varje sekund. Om larmet befinner sig i tillstånden larmad eller avlarmad kommer även tiden att skrivas ut på LCD-displayens nedre rad.

## 6 Resultat och diskussion

Bortsett från att den slutliga larmkonstruktionen hade 6 ingångar för detektorer i stället för 8 lyckades vi uppfylla samtliga kravspecifikationer. Det var från början planerat att en ATmega 16 med 16 KByte programminne skulle användas, men programmet visade sig bli större än så. Det finns optimeringsverktyg i AVR Studio 4 som kompilerar koden på ett mer effektivt sätt och som tar bort eventuella onödiga kod. Detta gjorde det kompilerade programmet betydligt mindre, men denna optimering gjorde även att alla larmets funktioner inte fungerade som tänkt. Vi bestämde oss därför att byta till ATmega 32 som i princip är samma mikroprocessor, men med dubbelt så mycket programminne.

Larmet visade sig både bli fullt funktionellt och användarvänligt. Det visade sig även fungera bra med både magnetdetektorer och IR-detektorer inkopplade.

Överlag är vi nöjda med den här kursen eftersom det gett oss möjligheten att knyta ihop olika områden som finns inom vår utbildning, som t.ex. elektronik och programmering. Möjligheten att både konstruera hårdvaran och skriva mjukvaran visa vikten av att planera sina projekt grundligt innan man påbörjar dessa. Många problem har dykt upp på vägen vilket har utvecklat vår förmåga att anpassa och tänka om tidigare lösningar.