



LUNDS
UNIVERSITET

Larmcentral



LUNDS
UNIVERSITET

LTH

**LUNDS TEKNISKA
HÖGSKOLA**

Digitala System, EITA 15

Rasmus Ivarsson
Karl Jergmo
David Petersson
Hugo Rolf
Handledare; Bertil Lindvall, Lars-Göran Larsson



LUNDS
UNIVERSITET

Abstract

During the final stages of the course EITA15 at Lund University we, the group members, were given the assignment to build an electrical circuit using components and software code. Therefore the decision to build an alarm system for homeowners became our plan of action. To disarm the alarm, a 4-digit code must be entered. During the time of the project it grew from being a simple idea towards a working product. With the following components a circuit was created; Atmega 1284 microprocessor, Hitachi HD44780 LCD-screen, Grayhill 83BB1-001 4x4 num-pad, IR-sensors to detect movement, a small buzzer to indicate the activation of the alarm and a LED-light. Overall the project served as a purpose to increase the group's knowledge of software- and hardware programming as well as computer science to create a working electrical product, based on specifications which were created by the group.



LUNDS
UNIVERSITET

Innehållsförteckning

Abstract	2
Innehållsförteckning	3
1 Inledning	4
1.1 Bakgrund	4
1.2 Syfte	4
1.3 Målformulering	4
1.4 Motivering till arbetet	5
1.5 Avgränsningar	5
1.6 Kravspecifikation	6
2 Hårdvara	7
2.1 Processor	7
2.2 Keypad	7
2.3 Display	8
2.4 IR-sensor	8
2.5 Högtalare	9
2.6 LED-lampor	9
2.7 Potentiometer	9
3 Mjukvara	10
4 Metod/Konstruktionsprocess	11
5 Diskussion	13
6 Appendix	14
7 Källförteckning	15



LUNDS
UNIVERSITET

1 Inledning

Kursen Digitala System (EITA15) ger en förståelse över hur hårdvarukomponenter kan användas för att designa och konstruera digitala kretsar, samt hur dessa kan programmeras i mjukvaruprogramspråket C. Rapporten innefattar en rad punkter där författarna beskriver syftet med projektet, metod, kravspecifikation, vilken hårdvara som har valts ut till konstruktionen samt en bilaga på kopplingsschemat.

1.1 Bakgrund

Detta projekt och utförande av denna rapport görs efter Bertil Lindvall som ett examinerande projekt i Kursen Digitala system. Projektet är obligatoriskt för studenter vid Lunds Universitet som läser högskoleingenjör i antingen Datateknik eller Elektroteknik. Handledare för projektet är Bertil Lindvall och Lars-Göran Larsson.

1.2 Syfte

Syftet med projektet är att förankra de teoretiska kunskaper vi ackumulerat under kursens gång och utveckla de praktiska färdigheterna kopplat till teorin. Med dessa kunskaper och färdigheter ska vi bygga en fungerande prototyp av ett larmsystem.

1.3 Målformulering

Målet med projektet är att konstruera ett larmsystem som använder en IR-sensor för att detektera värme, och därmed rörelse i närheten, för att trigga en signal. Det är då larmet går. Det ska sedan gå att stänga av detta larm med en 4-siffrig kod. Det ska även fungera att desarmera respektive armera larmet för att avaktivera/aktivera systemet. Projektet går således ut på, genom kunskapen vi ackumulerat genom kursen, att bygga en fungerande elektronisk krets där mjukvara och hårdvara kommunicerar och arbetar tillsammans.



LUNDS
UNIVERSITET

1.4 Motivering till arbetet

– *Larm är ett jättebra komplement till Grannsamverkan och certifierade lås mot inbrott. Det minskar tiden tjuvarna har på sig, och pockar på uppmärksamhet från grannar, men det stoppar inte tjuven från att ta sig in i ett hem,* berättar Lina Nilsson på Stöldskyddsföreningen.

Detta är en av anledningarna till att vi har valt att utveckla ett billigt larmsystem. Vi tror i längden att produktionskostnad på en större skala kommer att vara övervägande lägre än vad konsumenten är villiga att betala för den färdiga produkten. Vår ambition är att ha en 100% marginal mellan produktionskostnad och försäljningspris.

1.5 Avgränsningar

Produkten kommer att vara byggd på ett funktionellt sätt, därmed lägger vi inga krav på utseende av prototypen. Värt att nämna är också att det endast är en prototyp av en produkt och inget som är färdigt för massproducering. Därav kommer vissa komponenter inte vara de absolut optimala, som till exempel IR-sensorn vars räckvidd och effektivitet.



LUNDS
UNIVERSITET

1.6 Kravspecifikation



Larmet skall representera ett tjuvlarm som känner av när någon rör sig i rummet. Kraven och dess funktioner är följande:

- Larmet ska utlösas när någon av sensorerna triggas. (Högtalare skall låta och led blinka).
- Larmet ska kunna hantera minst 1 IR-sensor.
- Larmet skall kunna avaktiveras med en pin-kod på 4 siffror.
- Led skall lysa när larmet är aktiverat.
- Led skall inte lysa när larmet är avaktiverat.
- Larmet skall kunna aktiveras med en knapp.



LUNDS
UNIVERSITET

2 Hårdvara

 <table border="1" data-bbox="402 701 727 957"><tr><td>CINT8(XCK0/T0) PB0</td><td>1</td><td>40</td><td>PA0 (ADC0/PCINT0)</td></tr><tr><td>CINT9(CLK0/T1) PB1</td><td>2</td><td>39</td><td>PA1 (ADC1/PCINT1)</td></tr><tr><td>VT10(INT2/AIN0) PB2</td><td>3</td><td>38</td><td>PA2 (ADC2/PCINT2)</td></tr><tr><td>T11(OCA/AINT1) PB3</td><td>4</td><td>37</td><td>PA3 (ADC3/PCINT3)</td></tr><tr><td>INT12(OC2B/SS) PB4</td><td>5</td><td>36</td><td>PA4 (ADC4/PCINT4)</td></tr><tr><td>PCINT13(MOSI) PB5</td><td>6</td><td>35</td><td>PA5 (ADC5/PCINT5)</td></tr><tr><td>PCINT14(MISO) PB6</td><td>7</td><td>34</td><td>PA6 (ADC6/PCINT6)</td></tr><tr><td>PCINT15(SCK) PB7</td><td>8</td><td>33</td><td>PA7 (ADC7/PCINT7)</td></tr><tr><td>RESET</td><td>9</td><td>32</td><td>AREF</td></tr><tr><td>VCC</td><td>10</td><td>31</td><td>GND</td></tr><tr><td>GND</td><td>11</td><td>30</td><td>AVCC</td></tr><tr><td>XTAL2</td><td>12</td><td>29</td><td>PC7 (TOSC2/PCINT23)</td></tr><tr><td>XTAL1</td><td>13</td><td>28</td><td>PC6 (TOSC1/PCINT22)</td></tr><tr><td>(PCINT24/RXD0) PD0</td><td>14</td><td>27</td><td>PC5 (TOSC/PCINT21)</td></tr><tr><td>(PCINT25/TXD0) PD1</td><td>15</td><td>26</td><td>PC4 (TDO/PCINT20)</td></tr><tr><td>(PCINT26/INT0) PD2</td><td>16</td><td>25</td><td>PC3 (TMS/PCINT19)</td></tr><tr><td>(PCINT27/INT1) PD3</td><td>17</td><td>24</td><td>PC2 (TCK/PCINT18)</td></tr><tr><td>(PCINT28/OC1B) PD4</td><td>18</td><td>23</td><td>PC1 (SDA/PCINT17)</td></tr><tr><td>(PCINT29/OC1A) PD5</td><td>19</td><td>22</td><td>PC0 (SCL/PCINT16)</td></tr><tr><td>(PCINT30/OC2B/ICP) PD6</td><td>20</td><td>21</td><td>PD7 (OC2A/PCINT31)</td></tr></table>	CINT8(XCK0/T0) PB0	1	40	PA0 (ADC0/PCINT0)	CINT9(CLK0/T1) PB1	2	39	PA1 (ADC1/PCINT1)	VT10(INT2/AIN0) PB2	3	38	PA2 (ADC2/PCINT2)	T11(OCA/AINT1) PB3	4	37	PA3 (ADC3/PCINT3)	INT12(OC2B/SS) PB4	5	36	PA4 (ADC4/PCINT4)	PCINT13(MOSI) PB5	6	35	PA5 (ADC5/PCINT5)	PCINT14(MISO) PB6	7	34	PA6 (ADC6/PCINT6)	PCINT15(SCK) PB7	8	33	PA7 (ADC7/PCINT7)	RESET	9	32	AREF	VCC	10	31	GND	GND	11	30	AVCC	XTAL2	12	29	PC7 (TOSC2/PCINT23)	XTAL1	13	28	PC6 (TOSC1/PCINT22)	(PCINT24/RXD0) PD0	14	27	PC5 (TOSC/PCINT21)	(PCINT25/TXD0) PD1	15	26	PC4 (TDO/PCINT20)	(PCINT26/INT0) PD2	16	25	PC3 (TMS/PCINT19)	(PCINT27/INT1) PD3	17	24	PC2 (TCK/PCINT18)	(PCINT28/OC1B) PD4	18	23	PC1 (SDA/PCINT17)	(PCINT29/OC1A) PD5	19	22	PC0 (SCL/PCINT16)	(PCINT30/OC2B/ICP) PD6	20	21	PD7 (OC2A/PCINT31)	<h3>2.1 Processor</h3> <p>Processorn vi har valt att använda i projektet är en Atmega1284. Den har 40 pinnar och är högpresterande med en låg energiförbrukning. Mjukvaran kommer att vara i programmeringsspråket C och kommer att överföras till processorn i JTAG. Vidare information hittas i databladet för processorn.</p>
CINT8(XCK0/T0) PB0	1	40	PA0 (ADC0/PCINT0)																																																																														
CINT9(CLK0/T1) PB1	2	39	PA1 (ADC1/PCINT1)																																																																														
VT10(INT2/AIN0) PB2	3	38	PA2 (ADC2/PCINT2)																																																																														
T11(OCA/AINT1) PB3	4	37	PA3 (ADC3/PCINT3)																																																																														
INT12(OC2B/SS) PB4	5	36	PA4 (ADC4/PCINT4)																																																																														
PCINT13(MOSI) PB5	6	35	PA5 (ADC5/PCINT5)																																																																														
PCINT14(MISO) PB6	7	34	PA6 (ADC6/PCINT6)																																																																														
PCINT15(SCK) PB7	8	33	PA7 (ADC7/PCINT7)																																																																														
RESET	9	32	AREF																																																																														
VCC	10	31	GND																																																																														
GND	11	30	AVCC																																																																														
XTAL2	12	29	PC7 (TOSC2/PCINT23)																																																																														
XTAL1	13	28	PC6 (TOSC1/PCINT22)																																																																														
(PCINT24/RXD0) PD0	14	27	PC5 (TOSC/PCINT21)																																																																														
(PCINT25/TXD0) PD1	15	26	PC4 (TDO/PCINT20)																																																																														
(PCINT26/INT0) PD2	16	25	PC3 (TMS/PCINT19)																																																																														
(PCINT27/INT1) PD3	17	24	PC2 (TCK/PCINT18)																																																																														
(PCINT28/OC1B) PD4	18	23	PC1 (SDA/PCINT17)																																																																														
(PCINT29/OC1A) PD5	19	22	PC0 (SCL/PCINT16)																																																																														
(PCINT30/OC2B/ICP) PD6	20	21	PD7 (OC2A/PCINT31)																																																																														
	<h3>2.2 Keypad</h3> <p>Keypaden har 16 knappar konfigurerat i ett 4x4-upplägg. Det är här användaren kommer att skriva in kod för att armera respektive desarmera larmet. Keypaden vi valt att använda heter Grayhill 83BB1-001.</p>																																																																																



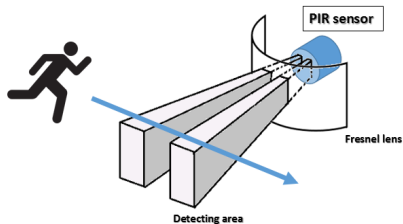
LUNDS
UNIVERSITET



- 1 VSS (Ground)
- 2 VDD (+ve)
- 3 VE (Contrast Voltage)
- 4 Register Select
- 5 Read/Write
- 6 Enable
- 7 Data 0
- 8 Data 1
- 9 Data 2
- 10 Data 3
- 11 Data 4
- 12 Data 5
- 13 Data 6
- 14 Data 7
- 15 Backlight Anode (+ve)
- 16 Backlight Cathode (Ground)

2.3 Display

Som display har vi använt en Hitachi HD44780 LCD. Detta är en alfanumerisk display med 0-9 siffror och ett alfabet som innefattar bokstäverna A-Z, försett med en dot matrix vilket är ett 2-dimensionellt mönster som ska representera dessa karaktärer. Denna display har möjligheten att skriva 2x16 tecken. Datablad till [LCD](#).



2.4 IR-sensor

Sensorn använder infrarött ljus för att kunna detektera rörelse runt om sig. Detta gör då att en signal kan skickas och larmet utlöses.



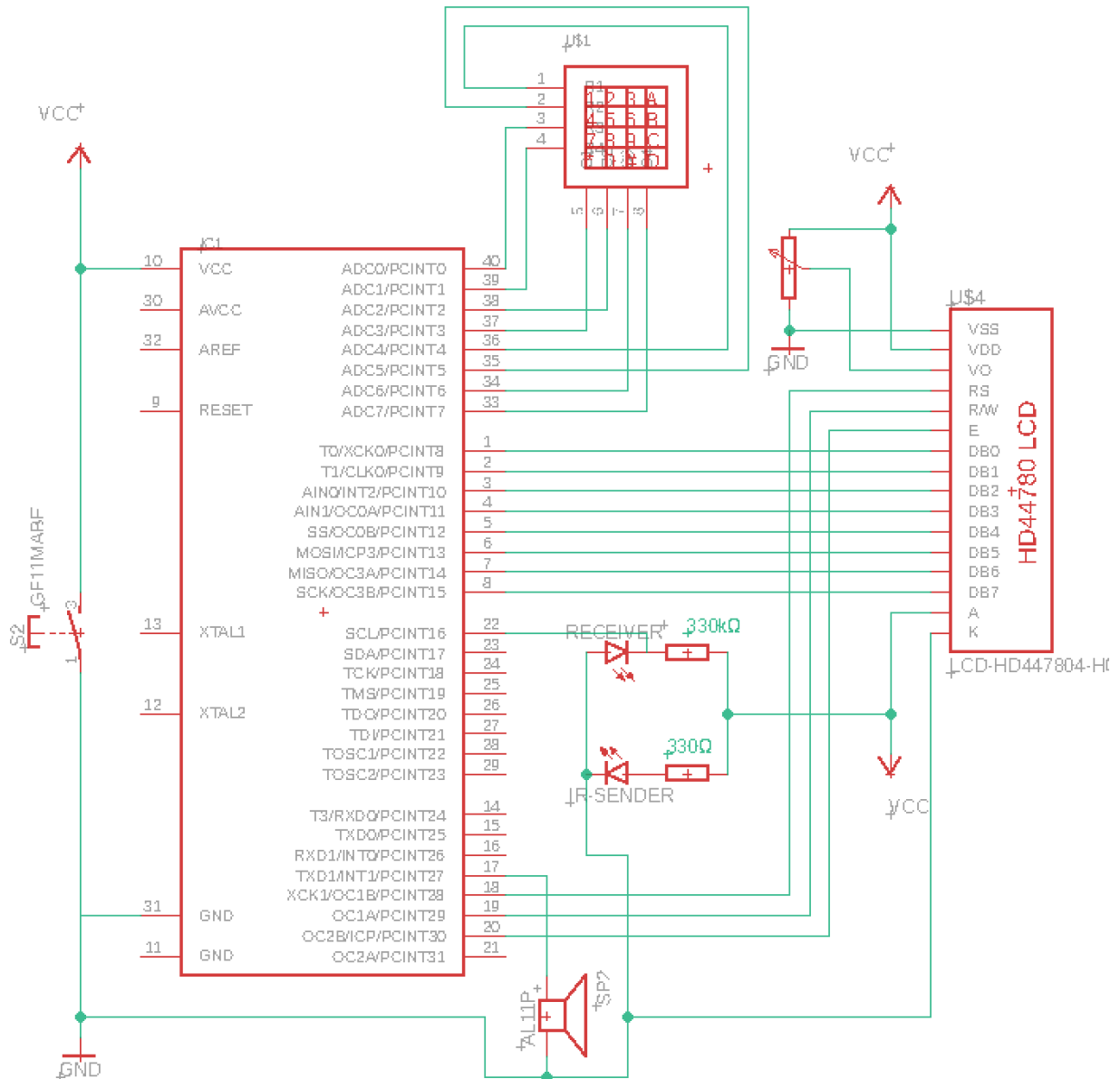
LUNDS
UNIVERSITET

	<h3>2.5 Högtalare</h3> <p>En liten “buzzer” kommer användas som högtalare. Den kommer ge ut en ton i en ljudnivå för att signalera att larmet har gått.</p>
	<h3>2.6 LED-lampor</h3> <p>Som ett komplement till ljudsignalen kommer vi använda 2 led-lampor för att signalera larmets tillstånd. En röd lampa innebär att systemet är armerat, och en grön att det inte är armerat.</p>
	<h3>2.7 Potentiometer</h3> <p>Fungerar i vår krets som en variabel resistor. Användningen är att kunna justera skärpan på våran LED-display.</p>



LUNDS
UNIVERSITET

2.8 Kopplingsschema





LUNDS
UNIVERSITET

3 Mjukvara

Olika metoder kallas beroende på vad som händer, metoderna körs i en while-loop under kravet att kretsen får en matningsspänning (+ 5V till ground). Larm utlöses, aktiveras eller avaktiveras, någon trycker på keypaden etc. Mjukvaran är skriven i C, och Atmel Studios.



LUNDS
UNIVERSITET

4 Metod/Konstruktionsprocess

Arbetet har delats upp i delmoment där delmoment ett var att skapa en övergripande plan för hur vi skulle göra. Det innebar val av material, kompetenser inom gruppen samt en övergripande tidsplanering.

Efter uppdelningen delmoment påbörjade vi ett kopplingschema för systemet. Syftet med att lägga detta i ett av de tidigare stegen var för att få en överblick av hur den fysiska produkten sedan skall kopplas.

När kopplingschemat var färdigställt började vi sedan byggandet av den fysiska produkten. Under arbetets gång utförde vi mindre test för att se att produkter presterade som de skulle, med isolerade avsnitt i koden för det ändamålet. En multimeter användes för att kontrollera spänning och ström på delar av kretsen som skulle konstrueras.

Till en början konstruerade vi processorn (Atmega 1284) tillsammans med LED-skärmen på ett breadboard för att se till att kommunikationen mellan dessa komponenter fungerade som tänkt. Genom databladet (länkat under hårdvara-beskrivning) kunde vi välja vilka portar som skulle kopplas vart. Vi valde att koppla LCD:n via port D på processorn. En tydligare beskrivning på hur finnes under mjukvara-beskrivningen. Kopplingen mellan portarna gjordes först på ett breadboard med hjälp av sladdar, för att sedan lödas fast på en slutgiltig platta.

En felaktig LCD-skärm visade sig vara ett problem under konstruktionen under detta steg. Detta resulterade i många timmars arbete med felsökning, både i den fysiska kopplingen på breadboarden och även i mjukvaran. Efter ett test av handledare visade det sig att LCD:n var felaktig och byttes ut. Koden vi skrivit kompletterades något för att få till vilken rad av LCD:n som texten printades ut på.

Efter att LCD:n var på plats och fungerade som uttänkt var nästa steg att integrera våran knappsats i kretsen. I utgångspunkt kopplades denna till Port B (ref datablad Atmega 1284) för att kunna användas till att skicka in signaler. Tanken med mjukvaran till denna komponent var att



LUNDS
UNIVERSITET

läsa en rad där en knapp blivit intryck, och sedan vilken kolumn denna låg på. En tydligare beskrivning av detta finns att läsa under mjukvara.

När samtliga produkter var på plats i systemet började vi sedan att jobba djupare med våran kod till processorn, mjukvaran till systemet. Under denna process delade vi upp koden återigen för att få de olika metoderna att fungera på respektive produkt. Detta medförde en smidigare process för att komma framåt i arbetet och även ett smidigare arbetsflöde.



LUNDS
UNIVERSITET

5 Diskussion

Arbetet har delvis uppfyllt de krav som sattes innan projektet startade. Arbetsprocessen har under projektets gång inte varit komplikationsfri, felsökningar och justering av kopplingsschema och kod har gjort att arbetet av projektet har stagnerat. Ett av de större problemen har varit brist på kunskaper inom hårdvaran och de som har använts för att bygga upp kretsen. Detta problem åtgärdades genom att noggrant läsa igenom databladerna för de olika komponenterna och därigenom bilda en tydlig uppfattning hur de olika delarna skulle kopplas på respektive portar för att kunna kommunicera med varandra på ett korrekt sätt enligt vår kravspecifikation.

Ytterligare ett problem var programmeringen av systemets mjukvara. Majoriteten av gruppens medlemmar hade begränsade kunskaper inom programspråket C och hur denna skulle användas för att få specifikationerna att fungera utifrån de kraven gruppen satt upp.

För att sammanfatta detta projekt har de varit lärorikt och logiskt krävande. Samarbetet i gruppen har varit bra där en viss uppdelning av projektet har skett där två av gruppens medlemmar har ägnat mer tid åt att sammanställa rapporten och de övriga på det praktiska med kretsen. Under de större delmomenten har alla medlemmar hjälps åt med att logiskt komma fram till lösningar om hur kopplingen av LED-displayen och keypaden skulle kopplas för komma fram till ett så bra resultat som möjligt.



LUNDS
UNIVERSITET

6 Appendix

```
/
* LCD_Testing.c
*
* Created: 2022-05-12 13:48:03
* Author : ra5327ru-s
*/
#define F_CPU 16000000UL
#include <stdio.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <time.h>

#define KEY_PRT PORTA
#define KEY_DDR DDRA
#define KEY_PIN PINA
#define LCD_Data_Dir DDRB
#define* LCD_Command_Dir DDRD
#define LCD_Data_Port PORTB
#define LCD_Command_Port PORTD
#define BUZZ PD2
#define SENSOR PC0
#define RS PD4
#define RW PD5
#define EN PD6

int alarm_activated = 0;
int triggered = 0;

void LCD_Command(unsigned char cmd)
{
    LCD_Data_Port = cmd;
    LCD_Command_Port &= ~ (1<<RS);
    LCD_Command_Port &= ~ (1<<RW);
```



LUNDS
UNIVERSITET

```
LCD_Command_Port |= (1<<EN);  
_delay_us(1);  
LCD_Command_Port &= ~(1<<EN);  
_delay_ms(3);  
}
```

```
void LCD_Char(unsigned char_data){  
  
    LCD_Data_Port = char_data;  
    LCD_Command_Port |= (1<<RS);  
    LCD_Command_Port &= ~(1<<RW);  
    LCD_Command_Port |= (1<<EN);  
    _delay_us(1);  
    LCD_Command_Port &= ~(1<<EN);  
    _delay_ms(1);  
  
}
```

```
void LCD_Init(void){  
    LCD_Command_Dir = 0xFF;  
    LCD_Data_Dir = 0xFF;  
    _delay_ms(20);  
  
    LCD_Command(0x38);  
    LCD_Command(0x0C);  
    LCD_Command(0x06);  
    LCD_Command(0x01);  
    LCD_Command(0x80);  
    _delay_ms(2);  
}
```

```
void LCD_String(char*str){  
  
    int i;  
    for(i = 0; str[i]!=0;i++){  
        LCD_Char(str[i]);  
    }  
}
```




LUNDS
UNIVERSITET

```
void LCD_String_xy (char row, char pos, char*str){

    if(row == 0 && pos < 16){
        LCD_Command((pos & 0x0F) | 0x80);
        LCD_String(str);
    }
    else if(row == 1 && pos < 16){
        LCD_Command((pos & 0x0F) | 0xC0);
        LCD_String(str);
    }

}

void LCD_Clear(){

    LCD_Command(0x01);
    LCD_Command(0x80);

}

unsigned char keypad [4][4] = { {'9', '8', '7', '*'},
                                {'6', '5', '4', '/'},
                                {'3', '2', '1', '-'},
                                {'R', '0', '=', '+'}};

unsigned char colloc, rowloc;

char keyfind()
{
    while(1)
    {
        KEY_DDR = 0xF0;    /* set port direction as input-output */
        KEY_PRT = 0xFF;

        do
```



LUNDS
UNIVERSITET

```
{
    KEY_PRT &= 0x0F; /* mask PORT for column read only */
    asm("NOP");
    colloc = (KEY_PIN & 0x0F); /* read status of column */
}while(colloc != 0x0F);

do
{
    do
    {
        _delay_ms(20); /* 20ms key debounce time */
        colloc = (KEY_PIN & 0x0F); /* read status of column */
    }while(colloc == 0x0F); /* check for any key press */

        _delay_ms(40); /* 20 ms key debounce time */
        colloc = (KEY_PIN & 0x0F);
    }while(colloc == 0x0F);

/* now check for rows */
KEY_PRT = 0xEF; /* check for pressed key in 1st row */
asm("NOP");
colloc = (KEY_PIN & 0x0F);
if(colloc != 0x0F)
{
    rowloc = 0;
    break;
}

KEY_PRT = 0xDF; /* check for pressed key in 2nd row */
asm("NOP");
colloc = (KEY_PIN & 0x0F);
if(colloc != 0x0F)
{
    rowloc = 1;
    break;
}

KEY_PRT = 0xBF; /* check for pressed key in 3rd row */
asm("NOP");
```



LUNDS
UNIVERSITET

```
colloc = (KEY_PIN & 0x0F);
if(colloc != 0x0F)
{
    rowloc = 2;
    break;
}

KEY_PRT = 0x7F;          /* check for pressed key in 4th row */
asm("NOP");
colloc = (KEY_PIN & 0x0F);
if(colloc != 0x0F)
{
    rowloc = 3;
    break;
}
}

if(colloc == 0x0E)
    return(keypad[rowloc][0]);
else if(colloc == 0x0D)
    return(keypad[rowloc][1]);
else if(colloc == 0x0B)
    return(keypad[rowloc][2]);
else
    return(keypad[rowloc][3]);
}

void sound(int length){
    LCD_Clear();
    LCD_String_xy(0,0, "Popo pull up");
    LCD_String_xy(1,0,"Skriv in kod");
    for(int i = 0; i < length; i++){
        LCD_Command_Port |= (1<<BUZZ);
        _delay_ms(300);
        LCD_Command_Port &= ~(1<<BUZZ);
        _delay_ms(300);
        if(alarm_activated == 0){
            break;
        }
    }
}
```



LUNDS
UNIVERSITET

```
    }  
}  
  
void alarm_triggered(void) {  
    DDRC &= ~(1<<SENSOR);  
    int temp = PINC & 0x01;  
    if(temp == 0x01){  
        triggered = 1;  
    } else {  
        triggered = 0;  
    }  
}  
  
ISR(PCINT2_vect){  
    alarm_triggered();  
    if(alarm_activated == 1 && triggered == 1){  
        sound(5);  
        main();  
    }  
}  
  
int main(void)  
{  
  
    PCICR = 0b000000100;  
    PCMSK2 = 0b000000001;  
    LCD_Init();  
  
    LCD_String_xy(0,0, "Ange kod:");  
  
    int i = 0;  
    char correct[6] = {'2', '2', '0', '2', '2', '4'};  
    char inserted[6] = { ' ' };  
    int right = 0;  
    sei();
```



LUNDS
UNIVERSITET

```
while(1){
    inserted[i] = keyfind();
    LCD_Char(inserted[i]);

    if(i == 5){
        for(i = 0; i < 6; i++){
            if(inserted[i] == correct[i]){
                right = 1;
            } else{
                right = 0;
                break;
            }
        }
        if (right == 1){
            _delay_ms(200);
            LCD_Clear();
            LCD_String_xy(0,0, "Korrekt kod");
            _delay_ms(500);
            if(alarm_activated == 0){
                alarm_activated = 1;
                LCD_Clear();
                LCD_String_xy(0,0, "Larmet");
                LCD_String_xy(1, 0, "Aktiverat");
                _delay_ms(700);
                main();
            } else {
                LCD_Clear();
                LCD_String_xy(0,0, "Larmet");
                LCD_String_xy(1,0, "Avaktiverat");
                alarm_activated = 0;
                _delay_ms(700);
                main();
            }
        } else{
            _delay_ms(200);
            LCD_Clear();
            LCD_String_xy(0,0, "Fel kod");
```



LUNDS
UNIVERSITET

```
        _delay_ms(700);  
        if(alarm_activated == 1){  
            sound(5);  
        }  
  
        main();  
    }  
}  
  
    i+=1;  
}  
}
```



LUNDS
UNIVERSITET

7 Källförteckning

Datablad, Processor Atmega 1284. Hämtad 28/04-2022.

<https://www.eit.lth.se/fileadmin/eit/courses/eita15/Laborationer/Datorteknik/lab2/ATmega1284.pdf>

Datablad, Hitachi HD44780 LCD-skärm. Hämtad 28/04-2022.

<https://www.eit.lth.se/fileadmin/eit/courses/datablad/Display/hd44780.pdf>

Koppling mellan olika komponenter, hjälp med kod.

<https://www.electronicwings.com/avr-atmega/interfaces>