



# The Phenomenal Doorbell

<b>Skola:</b>	Lunds Tekniska Högskola
<b>Institution:</b>	Elektro- och informationsteknik
<b>Kurs:</b>	Digitala projekt (EITF11)
<b>Projekttyp:</b>	Rapportsammanfattning
<b>Handledare:</b>	Bertil Lindvall
<b>Studenter:</b>	Louise Lyckvik, Josefin Andersson, Hedda Nilsson
<b>Datum:</b>	2015-05-11

## **Abstract:**

The following report compiles the process of constructing a prototype called “The Phenomenal Doorbell”. The doorbell is suited for interaction with clients at the door of an office, for example the door of a very busy nurses’ room. When a patient arrives, he or she rings the doorbell and then waits for the nurse to respond. In order to construct the prototype a specification of requirements was formulated and a strategy and implementation for the integration between soft- and hardware were made.

## Innehållsförteckning

1. Inledning
2. Kravspecifikation
3. Teori
  - 3.1 Hårdvara
    - 3.1.1 Kopplingsschema
    - 3.1.2 Processor
    - 3.1.3 LCD-display
    - 3.1.4 LED
    - 3.1.5 Knappar
    - 3.1.6 Summer
    - 3.1.7 Grind
    - 3.1.8 Kristall
    - 3.1.9 Resistorer
  - 3.2 Mjukvara
4. Arbetsprocessen
  - 4.1 Planering
  - 4.2 Hårdvarukonstruktion
  - 4.3 Mjukvarukonstruktion
5. Resultat
6. Diskussion och slutsats
7. Referenser/Källförteckning
8. Appendix/Bilagor
  - Bilaga 1. Kopplingsschema
  - Bilaga 2. Källkod

## 1. Inledning

Detta är en rapport som redovisar utförandet av en uppgift i kursen Digitala Projekt (EITF11). Uppgiftens syfte var att utforma en enklare digital prototyp vilket innebar planering, skapande av hårdvara samt implementering av mjukvara genom C-programmering.

Den prototyp som har skapats är en digital ringklocka hos en sjuksköterska. Sjuksköterskan ska ha möjligheten av ge respons i form av om hen är upptagen eller inte vilket ska visas för personen som ringer på i form av en ljussignal (LED) och ett meddelande på en display.

## 2. Kravspecifikation

En mottagningsringklocka som fungerar som en assistent vid dörren till en mycket upptagen sjuksköterskas rum ska skapas. Ringklockan ska bestå av en display, en knapp för att ringa på, tre lampor (grön, gul och röd) och en klocka som håller tiden samt ska kunna ringa när man trycker på knappen. Det ska även finnas en knappsats kopplad till ringklockan som sitter på andra sidan sjuksköterskans vägg, med två knappar: en för ledigt och en för upptaget. En patient kommer till mottagningen och ser displayen på vilken det står "Welcome". Ingen lampa lyser. När patienten sedan ringer på startas en klocka och olika scenarion kan inträffa. De scenarion som kan inträffa när patienten ringer på klockan är följande:

- A. Sjuksköterskan är inte närvarande.  
Den röda lampan börjar lysa efter 10 sekunder om sjuksköterskan ej trycker på en knapp.
- B. Sjuksköterskan är upptagen med en patient och det finns en väntetid.  
Om väntetiden är fem minuter trycker sköterskan en gång på knappen och väntetiden visas efter 10 sekunder från dess att klockan ringde. Om väntetiden är 10 minuter trycker sköterskan två gånger osv.
- C. Sjuksköterskan är ledig.  
Sjuksköterskan trycker på den knapp som motsvarar den gröna lampan. Lampan tänds efter de 10 sekunderna.

Övrigt:

- Alla lampor som tänds lyser i 10 sekunder och sedan slocknar dem.
- Det händer inget om en besökare trycker på ringklockan när en lampa redan lyser.

## 3. Teori

### 3.1 Hårdvara

Detta avsnitt avser att beskriva vilka hårdvarukomponenter som har använts i projektet.

#### 3.1.1 Kopplingsschema

Ett schema som visar hur hårdvarukomponenterna är sammankopplade finns i Bilaga 1. Kopplingsschema. Kopplingsschemat ritades innan prototypen sattes ihop för att underlätta vid konstruktionen.

#### 3.1.2 Processor

En processor av modellen Atmega16 används till denna prototyp. Processorn har totalt 40 pinnar, varav 32 st är I/O-pinnar och övriga pinnar är kopplade till ett JTAG interface. I/O-pinnarna är uppdelade i A, B, C respektive D-portar. Tre stycken LED och en summer har kopplats till A-porten. De tre knapparna är kopplade dels separat via varsin resistor till varsin pinne i D-porten och dels sammankopplade via en grind till en pinne i D-porten. En LCD-display är kopplad till både B- och D-porten och en kristall är kopplad till processorns crystal-port. För att testa programkoden användes en JTAG kopplad till processorns C-port.

#### 3.1.3 LCD-display

För att interagera med personen som ringer på hos sjuksköterskan används en LCD-display kopplad till B-porten samt D-porten. Denna visar det meddelande som programkoden är programmerad att skriva ut. Eftersom displayen ska visa både text och siffror (väntetid) är varianten som valts en LCD Units Alfanumerisk teckendisplay.

#### 3.1.4 LED

För att representera de ljussignaler som beskrivs i kravspecifikationen används tre LED; en grön, en gul och en röd.

#### 3.1.5 Knappar

Tre tryck-knappar används för att representera de knappar som kravspecifikationen hänvisar till. Dessa fungerar likt en strömbrytare vilket innebär att förutom att kopplas till processorn så är de kopplade till strömkällan så att en spänning i form av en etta skickas till processorns portar vid knapptryckning.

Knapp och port, samband:

- PD3 (17) - översta knappen (vänta)
- PD4 (18) - undre knappen (kom in)
- PD1 (15) - ensam knapp (ringklocka)

#### 3.1.6 Summer

En summer har kopplats in för att en signal ska ljuda inne hos sköterskan då någon trycker på knappen som motsvarar ringklockan.



### 3.1.7 Grind

Hårdvaran består av tre olika knappar som alla genererar ett avbrott då de trycks ned. Dessa är sammankopplade med en grind som förenar de tre olika knapparnas signaler till en signal som går in i processorn. Detta medför att ett och samma avbrott genereras oavsett vilken knapp som trycks ned. Därefter söker processorn upp vilken av knapparna som genererat avbrottet via knapparnas direktkoppling till processorn.

### 3.1.8 Kristall

En kristall är kopplad till processorn för att få tiden att räknas mer exakt. Detta då den inbyggda ocellatorn inte gör detta tillräckligt. Kristallen som används har en frekvens på 4 MHz.

### 3.1.9 Resistorer

Ett antal resistorer är inkopplade för att begränsa strömmen in till knapparna och LED:en. Detta för att kontrollera vilken spänning som kommer in till motsvarande knapp/LED.

## 3.2 Mjukvara

Mjukvaran till sjuksköterskans digitala ringklocka är programmerad i AVR Studio 6 vars gränssnitt är programmeringsspråket C. Huvudprogrammet inleds med en metod som utför alla grundinställningar till processorns portar, möjliggör avbrott samt skriver ut ett välkomstmeddelande på skärmen. Därefter körs en while-loop som var tiondels sekund kontrollerar om det skett ett avbrott i systemet, det vill säga undersöker om någon knapp tryckts in. Om en knapp har tryckts in granskar programmet vilken av knapparna som genererat avbrottet och skickar sedan denna information vidare. Informationen tas om hand av en metod som, beroende av knapptryck, tänds de olika LED:en samt i de fall som kravspecifikationen anger skriver ut ett meddelande på displayen.

## 4. Arbetsprocessen

### 4.1 Planering

Arbetet inleddes med att ta fram en idé över vad för slags produkt som skulle skapas. Genom att lista vilka funktioner som skulle finnas så skapades en kravspecifikation för produkten. Efter detta så ritades ett kopplingsschema i programmet PowerLogic. Här krävdes en del handledarhjälp för att förstå hur programmet fungerade. När kopplingsschemat ansågs vara klart påbörjades arbetet med hårdvarukonstruktionen.

### 4.2 Hårdvarukonstruktion

För att skapa hårdvaran tilldelades projektgruppen de komponenter som krävdes. Dessa fick kompletteras efter hand på grund av att fler delar blev nödvändiga. Hårdvarukomponenterna kopplades samman enligt kopplingsschemat. För att testa så att hårdvarukopplingen fungerade enligt plan kopplades ström in i

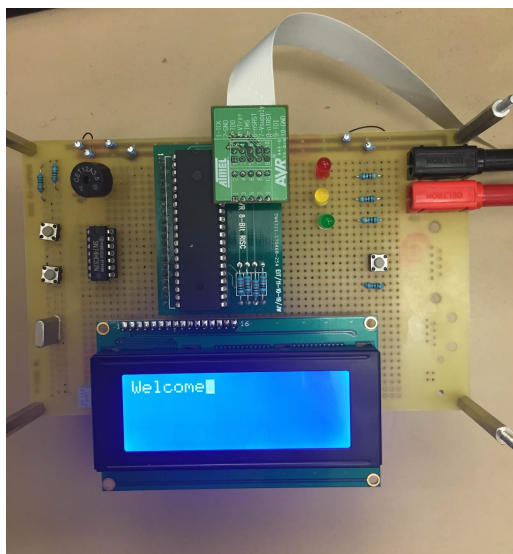
kretsen och en multimeter samt debuggern i AVR-Studio användes för att felsöka. Hårdvaran fungerade efter en del omarbetning som önskat.

### 4.3 Mjukvarukonstruktion

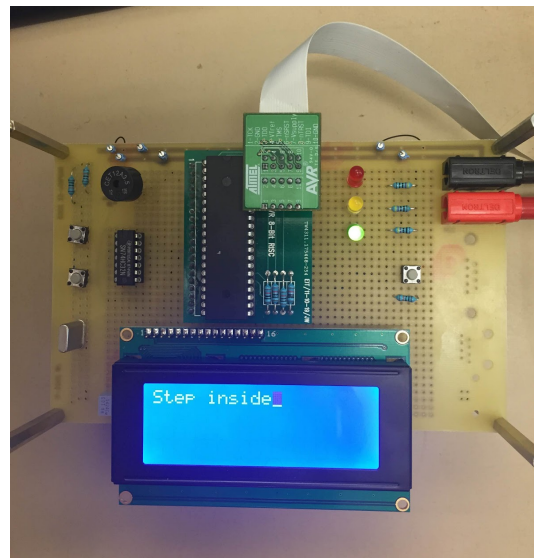
När hårdvaran var klar inleddes arbetet med att programmera mjukvaran. All programmering gjordes i C i AVR-Studio 6. Koden testades successivt genom att överföra den till hårdvaran med hjälp av en J-TAG. Detta var det mest omfattande arbetet som krävde mest tid.

## 5. Resultat

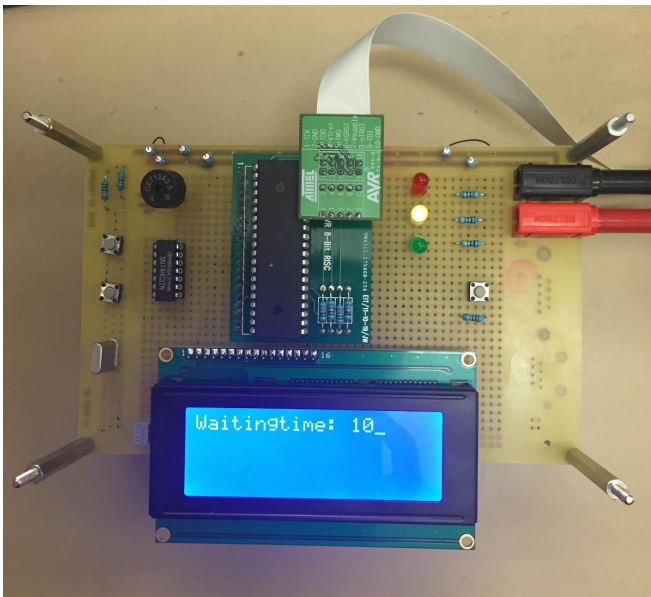
Projektet resulterade i en väl fungerande prototyp som kan hantera alla funktioner som angetts i kravspecifikationen.



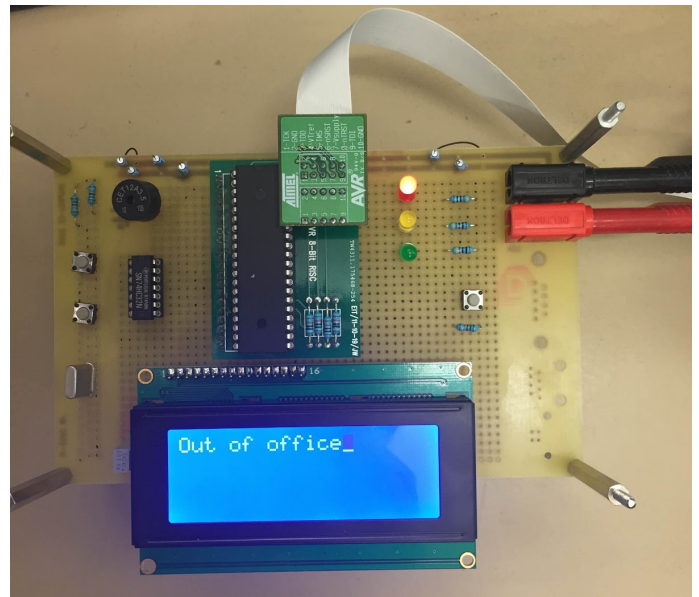
*Bild 1. När en patient kommer till mottagningen står det "Welcome" på skärmen. Ingen LED lyser.*



*Bild 2. Någon har ringt på och sjuksköterskan är inne och har tryckt på kom-in-knappen. Grön LED lyser.*



*Bild 3. Någon har ringt på men sjuksköterskan är upptagen. Vänta-  
knappen har tryckts in två gånger och på skärmen visas att väntetiden  
är 10 minuter och den gula LED:en lys.*



*Bild 4. Någon har ringt på men sjuksköterskan är inte på kontoret  
och den röda LED:en lys därför efter 10 sekunder.*

## 6. Diskussion och slutsats

Arbetet med att ta fram denna ringklocka har varit både roligt och mycket lärorikt. Genom att ta fram denna produkt har gruppen lärt sig mycket inom ämnet digital teknik som anses vara värdefullt för eventuella framtida kurser men framför allt förhoppningsvis framtida jobb. Genom utförandet av hårdvarukonstruktion och felsökning i denna, C-programmering, databladstolkningar har gruppen lärt sig mer om hur en arbetsprocess inom ämnet kan gå till.

Gruppen har stött på en del svårigheter längs vägen då kunskapen inom ämnet varit bristande. Detta har dock avhjälpats med god handledning, datablad och kreativt tänkande. Gruppen känner sig nu säkrare på att hantera nya utmaningar, speciellt inom detta område, då problemlösning har varit en stor del av arbetet.

Sammanfattningsvis så har projektet lärt oss mycket om arbete inom digital teknik och om hur konstruktionsarbeten likt detta är uppbyggda och går till. Vi är mycket nöjda med vårt projekt och med kursen i helhet. Projektets omfattning var rimlig för den tidsram som fanns men hade mer tid funnits hade det varit lärorikt och spännande att utveckla ringklockan ännu mer. Vi är ändå nöjda med den produkt vi tagit fram och väldigt nöjda med projektets resultat som helhet.

## 7. Referenser/Källförteckning

Databladet AVR-ATmega16 High-performance AVR 8-bit Microcontroller

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16.pdf>

Datum: 2015-04-14

Databladet Sharp Dot-Matrix - LCD Units Alfanumerisk teckendisplay

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Display/LCD.pdf>

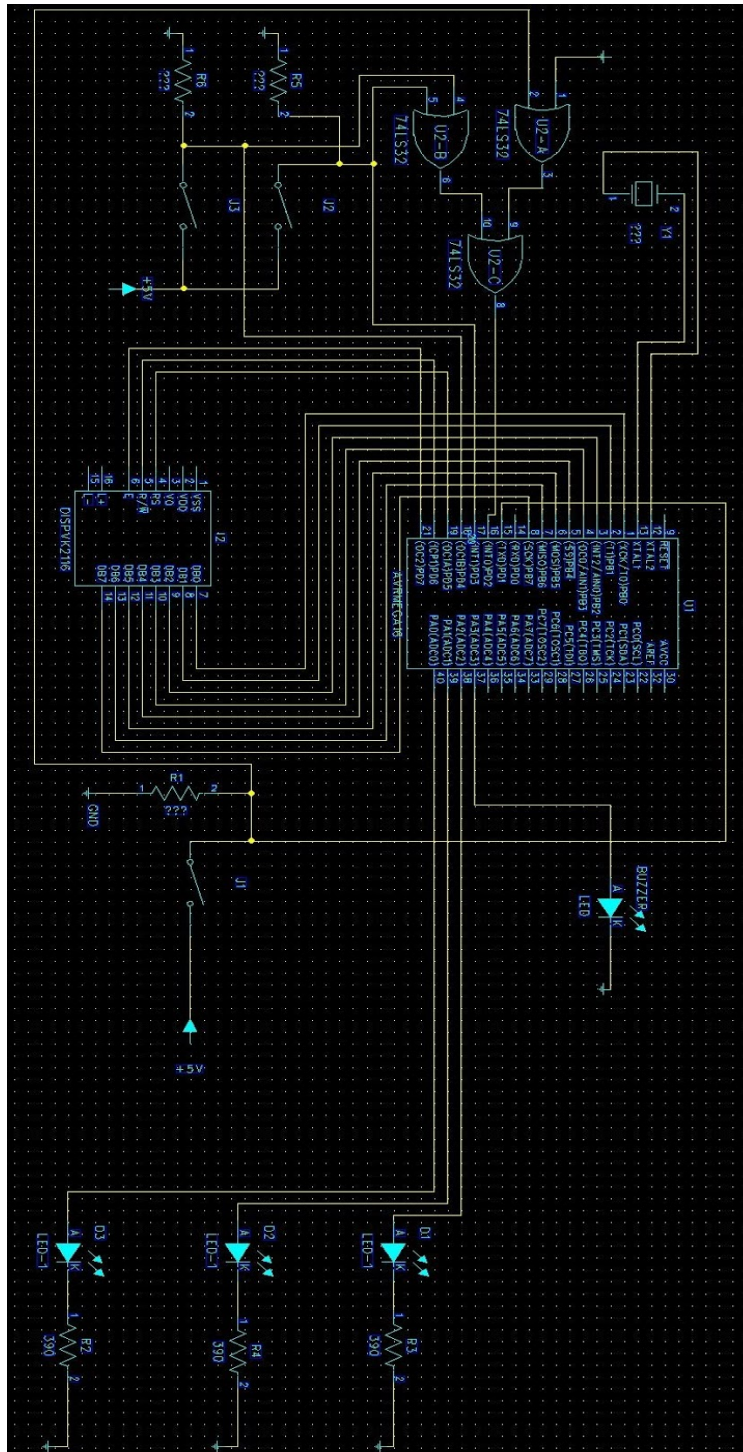
Datum: 2015-04-14

Manualen AVR Libc Users Manual-1.6.4(HTML)

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/avr-libc-user-manual/index.html>

## 8. Appendix/Bilagor

### Bilaga 1. Kopplingschema



## Bilaga 2. Källkod

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

//Variabler
int flag = 0;
int count = 0;
int timeVect[8];
int bellbutton = 0;
int waitbutton = 0;
int comeinbutton = 0;
int time = 0;
void setup() {
    DDRA = 0b00001111;
    DDRB = 0b11111111;
    DDRC = 0b00000000;
    DDRD = 0b11100000;
    disp_initial();
    disp_welcomemessage();
    enableButtonInterrupt();
    enableTimeInterrupt();
    sei();
}

void set_pin(char port, char pin, char state){

    char set = _BV(pin);

    if('A' == port){
        set &= PORTA;
        if(set != 0 && state == 0){ //Ändrar från 1 till 0
            PORTA ^= set;
        } else if(set == 0 && state == 1){ //Ändrar från 0 till 1
            PORTA ^= _BV(pin);
        }
    }
    else if('B' == port){
        set &= PORTB;
        if(set != 0 && state == 0){ //Ändrar från 1 till 0
            PORTB ^= set;
        }
    }
}
```



```
        } else if(set == 0 && state == 1){ //Ändrar från 0 till 1
        PORTB ^= _BV(pin);
        }
    }
    else if('C' == port){
        set &= PORTC;
        if(set != 0 && state == 0){ //Ändrar från 1 till 0
            PORTC ^= set;
        } else if(set == 0 && state == 1){ //Ändrar från 0 till 1
            PORTC ^= _BV(pin);
        }
    }
    else if('D' == port){
        set &= PORTD;
        if(set != 0 && state == 0){ //Ändrar från 1 till 0
            PORTD ^= set;
        } else if(set == 0 && state == 1){ //Ändrar från 0 till 1
            PORTD ^= _BV(pin);
        }
    }
}

//-----//
//----- SKÄRMINSTÄLLNINGAR -----//
//-----//

void write_cmd(char value) //RS är en nolla för att endast behöva läsa in commands
{
    PORTB = value;
    _delay_ms(5);
    set_pin('D', PD5, 0); // RS
    set_pin('D', PD6, 0); // RW
    _delay_ms(5);
    set_pin('D', PD7, 1); // E
    _delay_ms(5);
    set_pin('D', PD7, 0); // E
    _delay_ms(5);
}

void disp_writeCh(char val) // RS är en etta för att kunna läsa in tecken
{
    PORTB = val;
    _delay_ms(1);
    set_pin('D', PD6, 0); //RW
    set_pin('D', PD5, 1); //RS
}
```

```
    set_pin('D',PD7, 1); //Enable
    set_pin('D',PD7, 0); //Enable
}
void disp_nbr(int number) {
    int nbr1 = number/10;
    itoa(nbr1,timeVect,10);
    disp_writeCh(timeVect[0]);
    int nbr2 = number%10;
    nbr1 = itoa(nbr2, timeVect,10);
    disp_writeCh(timeVect[0]);
}

void disp_clear() {
    write_cmd(0b00000001); //Clear display
}
void disp_initial() //Initiala displayinställningar
{
    disp_clear();
    _delay_ms(5);
    write_cmd(0b00111000); //Function set
    _delay_ms(5);
    write_cmd(0b00001111); //ON/OFF set
    _delay_ms(5);
    write_cmd(0b00000010); //Entry mode set
}
//-----//
//----- SKRIVA PA SKÄRMEN -----//
//-----//
void disp_waitmessage(int number) {
    disp_clear();
    disp_writeCh('W');
    disp_writeCh('a');
    disp_writeCh('i');
    disp_writeCh('t');
    disp_writeCh('i');
    disp_writeCh('n');
    disp_writeCh('g');
    disp_writeCh('t');
    disp_writeCh('i');
    disp_writeCh('m');
    disp_writeCh('e');
    disp_writeCh(':');
    disp_writeCh(' ');
    disp_nbr(number); //Visa tiden beroende på knapptryck
}
}
```



```
void disp_welcomemessage() {
    disp_writeCh('W'); //Visa "Welcome" på skärmen
    disp_writeCh('e');
    disp_writeCh('l');
    disp_writeCh('c');
    disp_writeCh('o');
    disp_writeCh('m');
    disp_writeCh('e');
}
void disp_stepinsidemessage() {
    disp_writeCh('S'); //Visa "Step inside" på skärmen
    disp_writeCh('t');
    disp_writeCh('e');
    disp_writeCh('p');
    disp_writeCh(' ');
    disp_writeCh('i');
    disp_writeCh('n');
    disp_writeCh('s');
    disp_writeCh('i');
    disp_writeCh('d');
    disp_writeCh('e');
}
void disp_oomessage() {
    disp_writeCh('O'); //Visa "Out of office" på skärmen
    disp_writeCh('u');
    disp_writeCh('t');
    disp_writeCh(' ');
    disp_writeCh('o');
    disp_writeCh('f');
    disp_writeCh(' ');
    disp_writeCh('o');
    disp_writeCh('f');
    disp_writeCh('f');
    disp_writeCh('i');
    disp_writeCh('c');
    disp_writeCh('e');
}
//-----//
//----- LED -----//
//-----//
void redled_on()
{
    set_pin('A', PA0, 1);
}
```

```
void yellowled_on()
{
    set_pin('A', PA1, 1);
}
void greenled_on()
{
    set_pin('A', PA2, 1);
}
void redled_off()
{
    set_pin('A', PA0, 0);
}
void yellowled_off()
{
    set_pin('A', PA1, 0);
}
void greenled_off()
{
    set_pin('A', PA2, 0);
}
int check_button() {

    char val = PIND;

    val = val & 0b00011010;
    if(val == 8) //Sjuksköterskan trycker på vänta-knappen
    {
        return 1;
    }

    if(val == 16) //Sjuksköterskan trycker på kom in-knappen
    {
        return 2;
    }
    if (val == 2) //Någon ringer på
    {
        return 3;
    }
}

//Interrupts
void enableButtonInterrupt(){
    MCUCR = 0b0000011; // sida 67
    SREG = 0x82; //?
    GICR = (1 << INT0);
```

```
}

void enableTimeInterrupt(){
    //timer for count
    TCCR1A = 0b00000000;
    TCCR1B = 0b00000101;
    TCNT1 = 65145;

    //timer for buzzer
    TCCR0 = 0b00001001;
    TCNT0 = 0b11111111;
    TIMSK = 0b00000101;
}

ISR(INT0_vect){ //Globalt button-interrupt

    _delay_ms(1); //felsökning med Berra

    int a = check_button(); //KOLLA ALLA KNAPPAR

    if(a == 1){
        waitbutton = 1;
    }

    else if(a == 2){
        comeinbutton = 1;
    }

    else if (a == 3){
        bellbutton = 1;
    }

}

ISR(TIMER0_OVF_vect){ //Buzzer

    if(bellbutton == 1 && count < 20){
        set_pin('A', PA3, 0);
        _delay_ms(1);
        set_pin('A', PA3, 1);
    }

    TCNT0 = 0b11111111;

}

ISR(TIMER1_OVF_vect){ //Tidräknare som körs varje tiondels sekund
```

```
if(bellbutton == 1 && count < 100 && count >= 0){
    count++;
}
if(waitbutton == 1 && count > 0 && count < 100){
    time++;
    waitbutton = 0;
}

if(waitbutton == 1 && count == 0){
    waitbutton = 0;
    time = 0;
}

if(comeinbutton == 1 && count == 0){
    comeinbutton = 0;
    time = 0;
}
if(comeinbutton == 1 && count > 0 && count < 100){
    count = 0;
    time = 0;
    bellbutton = 0;
    comeinbutton = 0;
    flag = 3;    //green
}

if(count == 100 && time > 0){
    count = 0;
    bellbutton = 0;
    waitbutton = 0;
    flag = 1;    //yellow
}

if(count == 100 && time == 0){
    count = 0;
    bellbutton = 0;
    flag = 2;    //red
}

TCNT1 = 65145;
}

void actionPerformed(){
```

```
    if(flag == 3){ // tända grön led
        greenled_on();
        disp_clear();
        disp_stepinsidemessage();
        _delay_ms(500);
        disp_clear();
        greenled_off();
        disp_welcomemessage();
    }

    else if (flag == 1){// tända gul leds, skriva meddelande
        yellowled_on();
        time = time*5;
        disp_waitmessage(time);
        _delay_ms(500);
        disp_clear();
        yellowled_off();
        disp_welcomemessage();
        time = 0;
    }

    else if(flag == 2){ //tända röd led
        redled_on();
        disp_clear();
        disp_oomessage();
        _delay_ms(500);
        disp_clear();
        redled_off();
        disp_welcomemessage();
    }

    flag = 0;
}

//Huvudprogram
void main(void)
{
    setup();
    while(1)
    {
        actionPerformed();
    }
}
```