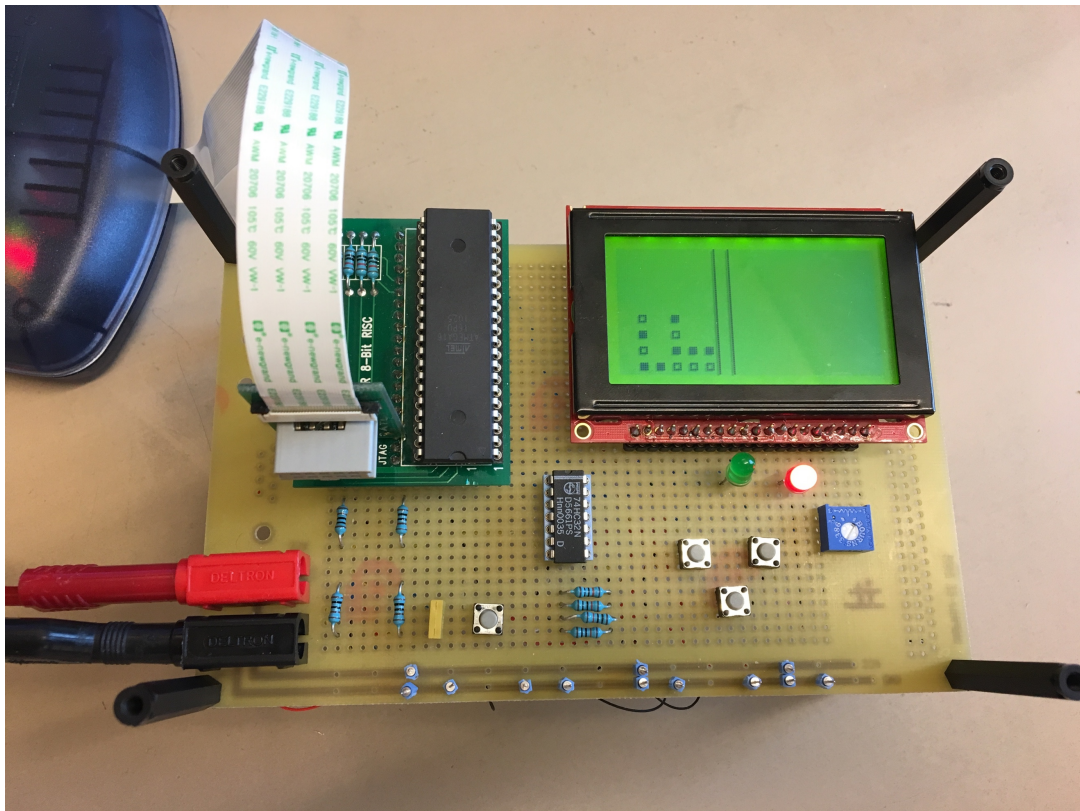


Rapport

Fyra i rad-spel



Sammanfattning: Detta dokument beskriver arbetsprocessen i det projekt som är en del av kursen Digitala Projekt(EITF11) vid Lunds Tekniska högskola. Målet med projektet är att integrera hårdvara och mjukvara för att få en fungerande prototyp. I detta projekt har en prototyp av ett fyra i rad-spel skapats och det är framställandet av denna som beskrivs och diskuteras nedan.

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

Innehållsförteckning

1. Inledning

2. Kravspecifikation

3. Hårdvara

4. Mjukvara

5. Utförande

5.1 Planering

5.2 Konstruktion

5.3 Programmering

6. Diskussion

7. Referenser

Appendix

Bilaga 1 - Kopplingschema

Bilaga 2 - Källkod

Bilaga 3 - Användarmanual

1. Inledning

I denna rapport sammanfattas arbetet med det projekt som ingår i kursen Digitala Projekt, EITF11, som ges av Lunds Tekniska Högskola. Projektets huvudsakliga uppgift är att skapa en digital prototyp där hårdvara samt mjukvara ska integreras till en helhet. Projektgruppen förväntas skapa prototypen från grunden och framställer alltså hårdvara såväl som mjukvara. I detta fall har en prototyp av ett Fyra i rad-spel skapats med en LCD-display samt knappar för att styra markören. Syftet med projektet är att skapa en förståelse för hur en konstruktionsprocess ser ut samt att tillskansa sig enklare kunskaper i programmeringsspråket C.

2. Kravspecifikation

- ☞ En LCD-skärm ska visa spelplanen som är tio kolumner bred och åtta rader hög.
- ☞ Spelet ska klara av två spelare. Spelarna har olika utseende på spelbrickorna och en markör i form av två streck visar i vilken kolumn spelaren befinner sig i.
- ☞ En grön eller röd LED lyser beroende på vems tur det är. Grön LED tillhör spelare 1, och röd LED spelare 2.
- ☞ Spelet har fyra knappar; en för att styra markören åt vänster, en för att styra markören åt höger, en för att lägga en spelbricka samt en för att starta om spelet.
- ☞ Om en spelare trycker på knappen för att lägga en bricka ska en bricka hamna ovanpå eventuella tidigare lagda brickor i markerad kolumn.
- ☞ Om en kolumn är full med brickor ska inget hända om man försöker lägga ytterligare en bricka.
- ☞ Om en spelare får fyra i rad ska LED-lampan med spelarens färg blinka.
- ☞ Om alla rutor på spelplanen fylls utan att någon vinner startar spelarna om spelet genom att trycka på knappen för omstart.
- ☞ Om en spelare vinner startar spelarna om spelet genom att trycka på knappen för omstart.
- ☞ Om en spelare trycker på knappen för omstart ska spelet startas om.

3. Hårdvara

Processor: Processorn som används är ATmega 16 AVR 8-bit Microcontroller. Denna har totalt 40 pinnar som är uppdelade i A-, B-, C- samt D-portar. Hur komponenterna är kopplade till processorn återfinns i kopplingsdiagrammet i bilaga 1.

Display: GDM 1602K. LCD-skärm bestående av 64x128 pixlar. Den består av två stycken separata delar à 64x64 som tillsammans utgör hela spelplanen.

Lysdioder: Två stycken, en röd och en grön. Används för att visa vilken spelares tur det är samt vem som har vunnit.

Knappar: Fyra stycken. En knapptryckning registreras genom att knappen är hög när den blir intryckt men är låg annars.

Resistorer: 10 stycken används för att begränsa strömmen till följande hårdvarukomponenter:

LED-lamporna - Båda lamporna är kopplade till varsin resistor på 150 ohm.

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

Knapparna - Alla fyra knappar är kopplade till en resistor på 10 kilo-ohm som sedan är kopplad till jord och en resistor på 1 kilo-ohm som sedan är kopplad till processorn.

Grind: En OR-grind används för att koppla samman knapparna. Grindens syfte är att generera ett avbrott oavsett vilken av de fyra knapparna som trycks ner.

Varistor: En resistor med motstånd som är reglerbart. Den används för att justera skärmens ljusstyrka.

4. Mjukvara

Programmeringen utfördes i programmet AtmelStudio med programspråket C. Jtag användes för att testa kod och hårdvara.

5. Utförande

5.1 Planering

Planeringen började redan vid kursens start, början av LP3. Vi bestämde oss tidigt för att göra ett spel. Det stod mellan sänka skepp och fyra i rad, och vi bestämde oss för fyra i rad. Vi tog därefter fram önskvärda krav för vår prototyp och läste datablad för de ingående komponenterna. Utifrån databladen skapade vi sedan ett kopplingschema, i dataprogrammet Eagle, som beskriver hur prototypen ska konstrueras.

5.2 Konstruktion

Efter att ha blivit tilldelade en verktygslåda samt de hårdvarukomponenter som skulle komma att krävas sattes arbetet med hårdvaran igång. Med hjälp av kopplingschema och handledare monterades komponenterna på mönsterkortet och kopplades sedan samman. Därefter användes en JTAG och enklare programrader i ATMEL studio för att kontrollera att komponenterna kopplats korrekt. Då fel upptäcktes gjordes förändringar i konstruktionen tills prototypen uppförde sig på önskat vis. Utöver ATMEL studio användes också en logikpenna samt multimeter för att felsöka.

5.3 Programmering

När prototypens hårdvara fungerade som väntat påbörjades programmeringen av mjukvaran. Programmering utfördes i AtmelStudio med programspråket C.

6. Resultat och diskussion

6.1 Resultat

Resultatet av projektet är en fungerande prototyp som uppfyller samtliga krav i kravspecifikationen ovan.

6.2 Diskussion

Projektet har varit lärorikt och spännande. Till en början kändes det som att vi saknade stora delar av den kunskap som krävdes för att lyckas skapa en fungerande prototyp. Efter att ha läst datablad samt tagit hjälp från handledare lyckades vi dock få ihop ett spel till slut. Till en början var vi väldigt ivriga att komma igång med projektet vilket gjorde att vi inte alltid tänkte igenom alla steg ordentligt innan vi utförde dem. Det ledde till att vi vid flertalet tillfällen fick göra om vissa delar av konstruktionen.

Programmeringen upplevde vi som ganska svår då det var ett nytt språk att koda i och dessutom en ny upplevelse att koda för att integrera hårdvara och mjukvara. De delar av språket som är likt Java, behärskade vi relativt bra medan de funktioner som är maskinnära blev en större utmaning. Vi provade oss fram och fick till slut fram en fungerande kod även om processen dit var tidskrävande.

För att göra arbetet mer effektivt påbörjades projektets alla olika delar ganska tidigt. När vi satt fast i exempelvis konstruktion och programmering inleddes arbetet med hemsida och rapport.

Slutligen kan vi konstatera att vi har lärt oss väldigt mycket nya saker inom ett område som till en början var relativt nytt för oss. När vi ser tillbaka på vår arbetsprocess är det främst två punkter vi kan dra lärdomar av. Först och främst tror vi att det hade lönat sig att läsa på lite mer om C-språket istället för att enbart prova sig fram, det hade antagligen genererat i ett mer effektivt arbete. Dessutom hade det med största sannolikhet varit mer förmånligt för oss att ibland tänka igenom stegen i vår process innan de utfördes. På detta sätt hade vi troligtvis undvikit mycket omarbete.

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

7. Referenser

Datablad för processorn - High-performance AVR 8-bit Microcontroller

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

Datablad för display - GDM12864C

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Display/GDM12864H.pdf>

Datablad för ellergrinden - Quad input 2 or

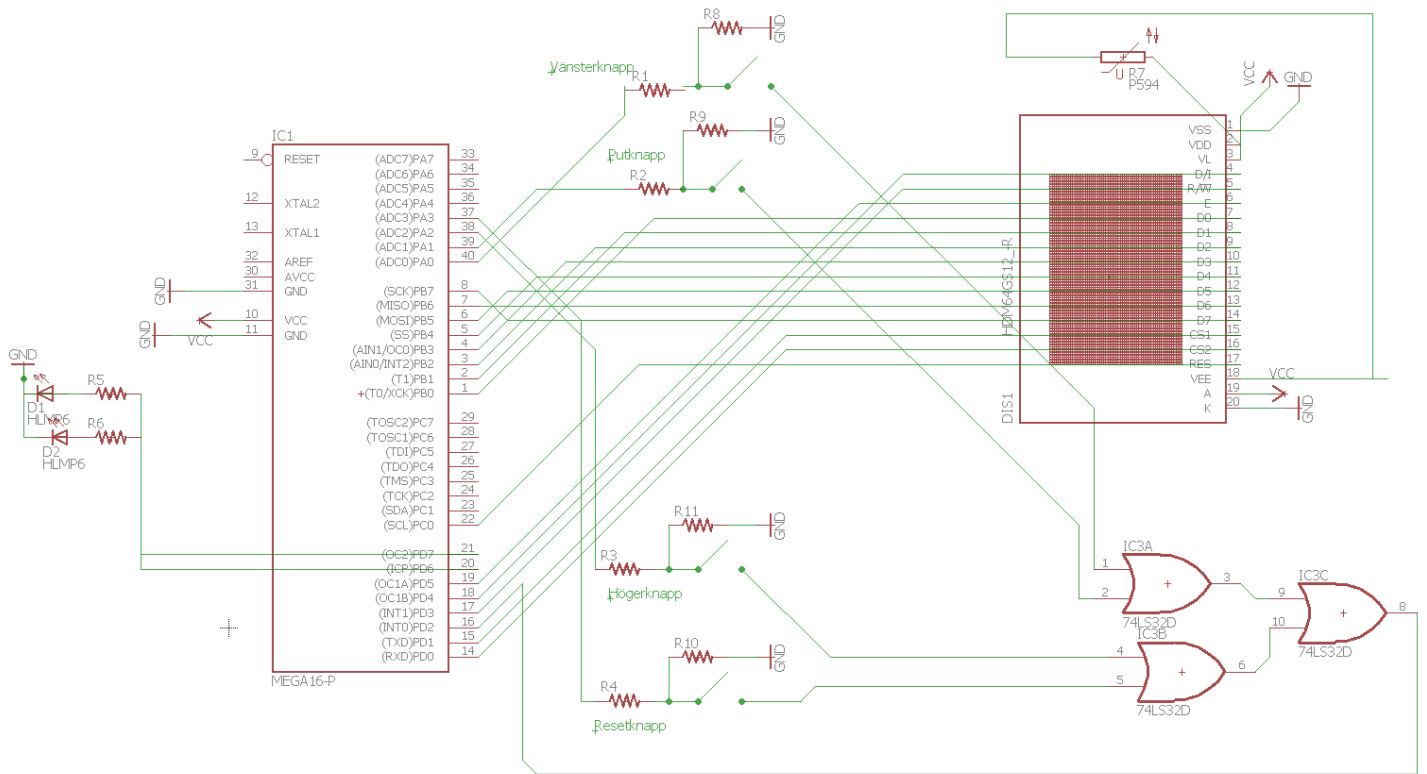
<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Logik/74HC/74HC32.pdf>

AVR bibliotek för att utläsa information om avbrott

http://www.eit.lth.se/fileadmin/eit/courses/edi021/Avr-libc-2.0.0/group_avr_interrupts.html

Appendix

Bilaga 1: Kopplingschema



Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

Bilaga 2: Källkod

```
/*
 * Fyra i rad.c
 *
 * Created: 2017-04-03 08:29:07
 * Author : inel4lra
 */

#include <avr/io.h>
#include <avr/interrupt.h>

#include <util/delay.h>
#define F_CPU 8000000UL

int matrix[8][10];
int arrow;
int side; // Vilken chipselect: side = 0 --> CS1
int player;
char global;

/* Avbrottsrutin */

ISR(INT0_vect){
    _delay_ms(100);
    global = PINA & 0x0F;
}

/* Fylla matris med nollor*/
void createMatrix(){
    for (int i = 0; i<8; i++){
        for (int j = 0; j<10; j++){
            matrix[i][j]=0;
        }
    }
}

/* Instruktioner till displayen*/

void executeInstruction(){
    _delay_ms(10);
    if(side == 0){
        PORTD = 0x21; //skriv instruction istället för data
        PORTC = 0x01;
        PORTD = 0x01; //Execute
    } else {
        PORTD = 0x22;
        PORTC = 0x01;
        PORTD = 0x02;
    }
}
```


Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

}

/* Sätter adressen på displayen*/

```
void setXY(int x, int y){
    PORTB = 0xB8 + x; //Sätt x
    executeInstruction(); //sätt adress; därför instruction
    if(y<64){ //CS storlek, CS1
        PORTB = 0x40 + y;
        side = 0;
        executeInstruction();
    } else { //CS2
        y = y - 64;
        PORTB = 0x40 + y;
        side = 1;
        executeInstruction();
    }
}
```

/* Förbereder display för att skriva*/

```
void write(){
    _delay_ms(1);
    if(side==0){
        PORTD = 0x31; //Sätter porten till att skriva data, CS1
        Behövs reset port c
        PORTD = 0x11; //Nu skriver den
    } else {
        PORTD = 0x32;
        PORTD = 0x12;
    }
}
```

/* Skriver på displayen*/

```
void draw(int x, int y, char data){
    setXY(x,y);
    PORTB = data;
    write();
}
```

/* Kollar om en spelare har vunnit */

```
int victory(int x, int y){
    int count = 0;
    int stop1 = 0;
    int stop2 = 0;
    for(int i = 1; i<4; i++){
        if (matrix[x][y] == matrix[x+i][y] && stop1 == 0){ //kolla
            neråt.
                count++;
            }else{
```

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

```

        stop1 = 1;
    }
}
if(count == 3){ //om fyra i rad
    return 1;
}
count = 0;
stop1 = 0;
for(int i = 1; i<4; i++){
    if(matrix[x][y] == matrix[x][y+i] && stop1 ==
0){//Kollar åt höger
        count++;
    }else{
        stop1 =1;
    }
    if(matrix[x][y] == matrix[x][y-i] && stop2 ==
0){//Kollar åt vänster
        count++;
    }else{
        stop2 = 1;
    }
}
if(count == 3){//om fyra i rad
    return 1;
}
count =0;
stop1=0;
stop2=0;
for(int i = 1;i<4;i++){
    if(matrix[x][y] == matrix[x+i][y+i] &&
stop1 == 0){//Kollar snett neråt höger
        count++;
    }else{
        stop1=1;
    }
    if(matrix[x][y] == matrix[x-i][y-i] &&
stop2 == 0){//Kollar snett uppåt vänster
        count++;
    }else{
        stop2=1;
    }
}
if(count==3){//om fyra i rad
    return 1;
}
count=0;
stop1=0;
stop2=0;
for(int i = 1;i<4;i++){
    if(matrix[x][y] == matrix[x-i][y+i] && stop1 ==
0){//Kollar snett uppåt höger
        count++;
    }else{
        stop1=1;
    }
}

```

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

```

        if(matrix[x][y] == matrix[x+i][y-i] && stop2 ==
0){//Kollar snett neråt vänster
                count++;
            }else{
                stop2=1;
            }
        }
        if(count==3){//om fyra i rad
            return 1;
        }
        return 0;
    }

/* Hittar plats på displayen*/

int findOurPlace(){
    int currentPlace = 0;
    if(matrix[currentPlace][arrow] !=0){//om kolonnen redan är
ifylld
        return -1;
    }
    while (matrix[currentPlace+1][arrow] == 0 && currentPlace
<7){//7 pga har kollat första platsen
        currentPlace++;
    }
    return currentPlace;
}
/* Tar bort strecken som visar vilken kolumn spelare befinner sig
i*/

void clearArrow(int y){
    y=8*y;
    for(int i = 0; i<8; i++){
        draw(i,y,0x00);
        draw(i,y+7,0x00);
    }
}

/*Rita strecken som visar vilken kolumn spelare befinner sig i*/

void drawArrow(int y) {
    if(y<10){
        y= 8*y;//gör om till dots
        for(int i =0; i<8;i++){
            draw(i,y,0xFF);
            draw(i,y+7,0xFF);
        }
    }
}

/*Ritar spelpjäs för spelare 1*/
void player1(int x, int y){
    y=y*8;
    draw(x,y+1,0x00);
    draw(x,y+2,0x3C);
}

```

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

```
        draw(x,y+3,0x3C);
        draw(x,y+4,0x3C);
        draw(x,y+5,0x3C);
        draw(x,y+6,0x00);
    }
    /*Ritar spelpjäs för spelare 2*/
    void player2(int x, int y){
        y=y*8;
        draw(x,y+1,0x00);
        draw(x,y+2,0x3C);
        draw(x,y+3,0x24);
        draw(x,y+4,0x24);
        draw(x,y+5,0x3C);
        draw(x,y+6,0x00);
    }

    /* Tömmer displayen*/
    void clearDisplay(){
        for(int i = 0; i <128; i++){//???128
            for(int j = 0; j<8; j++){
                setXY(j,i);
                PORTB = 0x00;
                write(); // töm displayen; därför write
            }
        }
    }

    /*läser av knappar */
    void readButton(){
        if(global == 0x04){ //högerknapp
            if(arrow < 10){
                clearArrow(arrow);
                arrow++;
                drawArrow(arrow);
                if(player == 1){
                    PORTD = 0x80;
                }else{
                    PORTD = 0x40;
                }
            }
        }
        else if(global == 0x01){ //vänsterknapp
            if(arrow>0){
                clearArrow(arrow);
                arrow--;
                drawArrow(arrow);
                if(player == 1){
                    PORTD = 0x80;
                }else{
                    PORTD = 0x40;
                }
            }
        }
        else if(global == 0x02){ //put-knapp
```



```

        _delay_ms(1000);
        PORTD =0x00;
        _delay_ms(1000);
        PORTD=0x40;
        _delay_ms(1000);
        PORTD =0x00;
        clearDisplay();
    }else{
        player = 1;
        PORTD = 0x80;
    }
}

}

}
else if(global ==0x08){
    createMatrix();
    arrow = 0;
    side = 0;
    clearDisplay();
    drawArrow(0);
    player = 1;
    PORTD = 0x80;
}
}

/* Sätter på displayen*/
void turnOnDisplay(){
    PORTB = 0x3F; //se sidan 10 i displayens datablad
    executeInstruction();
    side = 1;
    PORTB = 0x3F; //se sidan 10 i displayens datablad
    executeInstruction();
}

int main(void)
{
    DDRA = 0x00;
    DDRB = 0xFF;
    DDRD = 0xfb;
    global = 0x00;
    GICR = 0b01000000;
    MCUCR = 0b00000011;
    sei();
    createMatrix();
    arrow = 0;
    side =0;
    turnOnDisplay();
    clearDisplay();
    drawArrow(0);
}

```

Grupp 9

Emma Rasmusson & Louise Ragnarsson

Handledare: Bertil Lindvall

```
    player = 1;
    PORTD = 0b10000000;

    while (1)
    {
        if(global != 0x00){
            readButton();
            global = 0x00;
        }
    }
}
```

Fyra i rad

Användarmanual

- ⌘ Spelplanen är 10 kolumner bred och 8 rader hög.
 - ⌘ Tryck på höger eller vänster knapp för att välja vilken kolumn du vill lägga din spelpjäs.
 - ⌘ Tryck på mittenknappen för att lägga en spelpjäs.
 - ⌘ Om spelet blir fyllt utan att någon vunnit - tryck på resetknappen för att starta om spelet.
 - ⌘ Om en spelare vunnit - starta ett nytt spel genom att trycka på resetknappen.
- ⌘ Lycka till!