



EITF11 Digitala Projekt

TIC TAC TOE

Lukas Ljungblom & Johan Lyckenvik I-13
Handledare: Bertil Lindvall

Lunds Tekniska Högskola
VT 2016

Sammanfattning

Rapporten handlar om ett projekt i kursen EITF11 Digitala Projekt där uppgiften har varit att konstruera en hårdvara med en viss funktion och sedan programmera en C-kod för att implementera denna funktion. Gruppen har valt att ta sig an uppgiften att konstruera ett luffarschack-spel, eller Tic Tac Toe, på en LCD med tillhörande nödvändiga komponenter. Projektet resulterade i en tillfredställande prototyp som motsvarade kraven för ett standard luffarschack-spel.

Innehållsförteckning

1. Inledning	3
1.1 Bakgrund	3
1.2 Uppgift.....	3
2. Kravspecifikation.....	3
2.1 Funktionella krav	3
2.2 Kvalitetskrav	4
3. Hårdvara	4
3.1 Processor.....	4
3.2 Display	4
3.3 Varistor.....	4
3.4 Resistorer	4
3.5 Dioder.....	4
3.6 Tangentbord och Key Encoder	4
3.7 JTAG	4
3.8 Mönsterkort.....	5
3.9 Strömförsörjning.....	5
3.10 Kondensatorer.....	5
4. Mjukvara	5
5. Utförande	5
5.1 Planering.....	5
5.2 Konstruktion.....	5
5.3 Programmering	6
6. Resultat.....	6
7. Diskussion och slutsats	6
8. Bilagor.....	7
Bilaga 1. Kopplingsschema	7
Bilaga 2. Källkod.....	8

1. Inledning

1.1 Bakgrund

Teknikprofilen System- och Programvaruutveckling på civilingenjörsprogrammet Industriell Ekonomi vid Lunds Tekniska Högskola innehåller kursen EITF11 Digitala projekt, 10 hp.

Avsikten med kursen är att ge eleven ett begrepp om hur konstruktionsarbete går till. Tanken är att eleven med hjälp av handledare ska kunna prestera en fungerande prototyp med tillhörande dokumentation.

1.2 Uppgift

Uppgiften har varit att konstruera och programmera en prototyp för ett utvalt projekt inom ramen för kursen Digitala Projekt.

Utifrån de alternativ som ges i kursen så gick valet till att konstruera en LCD som man ska kunna spela ett spel på.

Efter att ha gått igenom olika tänkbara applikationer för display och övervägt intressen och förmåga bestämde sig författarna för att göra ett Luffarschack-spel, eller Tic Tac Toe, som användning för displayen.

2. Kravspecifikation

2.1 Funktionella krav

1. Konstruktionen ska ha en display innehållandes en spelplan som består av nio rutor
2. Till konstruktionen följer ett tangentbord bestående av 4x4 (16) rutor.

1	2	3	R
4	5	6	
7	8	9	A

Figur 1 Tangentbord

1. Med knapparna 1-9 väljs vilken ruta som man vill utföra någonting på.
 2. Med knappen A läggs en figur ut eller tas bort på displayen.
 3. Med knappen R startas spelet om om någon har vunnit
 4. De mörkgrå knapparna utför ingenting.
2. Systemet ska indikera när en person har vunnit.
 1. Genom att den gröna dioden lyser.
 3. Drag som strider mot spelreglerna ska inte gå att göra.

4. Varannat tur ska resultera i att ett X skrivs ut och varannat i att ett O skrivs ut.
5. Efter att sex figurer är utplacerade ska det gå att flytta figurer till tomma platser.

2.2 Kvalitetskrav

1. Systemet ska verka efter reglerna för tre i rad.
2. Spelet ska vara uppbyggt på en LCD på 128x64 pixlar.
3. Systemet ska ha en tillfredställande fördröjning för signaler.

3. Hårdvara

Fullständigt kopplingschema finns i Bilaga 1.

3.1 Processor

I projektet används mikroprocessorn AVR AT Mega32 16PI 0422B. Programkoden importeras till processorn genom en JTAG.

3.2 Display

Som display används en grafisk LCD vid namn GDM12864HLCM. Displayen består av 128x64 pixlar.

3.3 Varistor

En variabel resistor används för att justera displayens ljusstyrka.

3.4 Resistorer

Resistorer används för att få önskad spänning över dioder och display.

3.5 Dioder

En röd och en grön diod används för att indikera när ett regelbrott begås (röd diod) respektive när en spelare vunnit (grön diod).

3.6 Tangentbord och Key Encoder

Ett tangentbord används för att skicka signaler till en Key Encoder som i sin tur konverterar dessa till processorn.

3.7 JTAG

Används för att importera och exekvera mjukvaran som processorn ska fungera efter.

3.8 Mönsterkort

Ett mönsterkort används för montering av samtliga delar samt sammankoppling av dem.

3.9 Strömförsörjning

Till kretsen ansluts en strömkälla på 5 V och 5 A.

3.10 Kondensatorer

Kondensatorer används för att få Key Encodern att oscillera vilket behövs för att skapa en insignal till processorn.

4. Mjukvara

Spelet skrivs som ett program i språket C i platformen Atmel Studio 6.2. En JTAG används enligt rubrik 3.7 JTAG.

Koden i programmet möjliggör önskad kommunikation mellan ett antal hårdvarukomponenter och beskriver själva spelet. När spelet körs används metoder för att rita upp en spelplan och för att sedan placera ut figurer på den. Andra viktiga delar av koden beskriver spelreglerna så att spelet fungerar på önskat vis.

5. Utförande

5.1 Planering

Författarna ville skapa ett simpelt spel som de själva kände till. Valet föll på en variant av "luffarschack" även kallat "Tic Tac Toe".

Första steget var sedan att skapa en kravspecifikation för att få en översiktlig bild över vad projektet skulle innehålla.

En skiss på kopplingsschema ritades upp och i samråd med handledaren bestämdes sedan önskvärda komponenter vilket resulterade i att kopplingsschemat reviderades till ett första utkast.

5.2 Konstruktion

Med kopplingsschema färdigt började arbetet med att placera ut och fästa komponenter på mönsterkort. Under processen fick kopplingsschema och kravspecifikation revideras ett antal gånger för att matcha verkligheten.

5.3 Programmering

Första steget med att skriva mjukvaran var att få olika komponenter på mönsterkortet att kommunicera med varandra. Under detta arbete skrevs metoder som inte sedan kom att användas i den slutgiltiga koden och dessa syftade mest till att testa hur kommunikationen mellan exempelvis dioder och knappar fungerade.

Steg två blev att söka rita upp spelplanen på displayen. Vid detta skede uppdagades det att processorn inte var tillräckligt kraftig för att kunna hantera spelet vilket resulterade i ett byte till en dubbelt så stor processor.

Slutfasen av programmeringen blev sedan att skriva metoderna som utför själva spelandet och kontrollerar att spelreglerna följs.

För att se fullständig källkod, se Bilaga 2.

6. Resultat

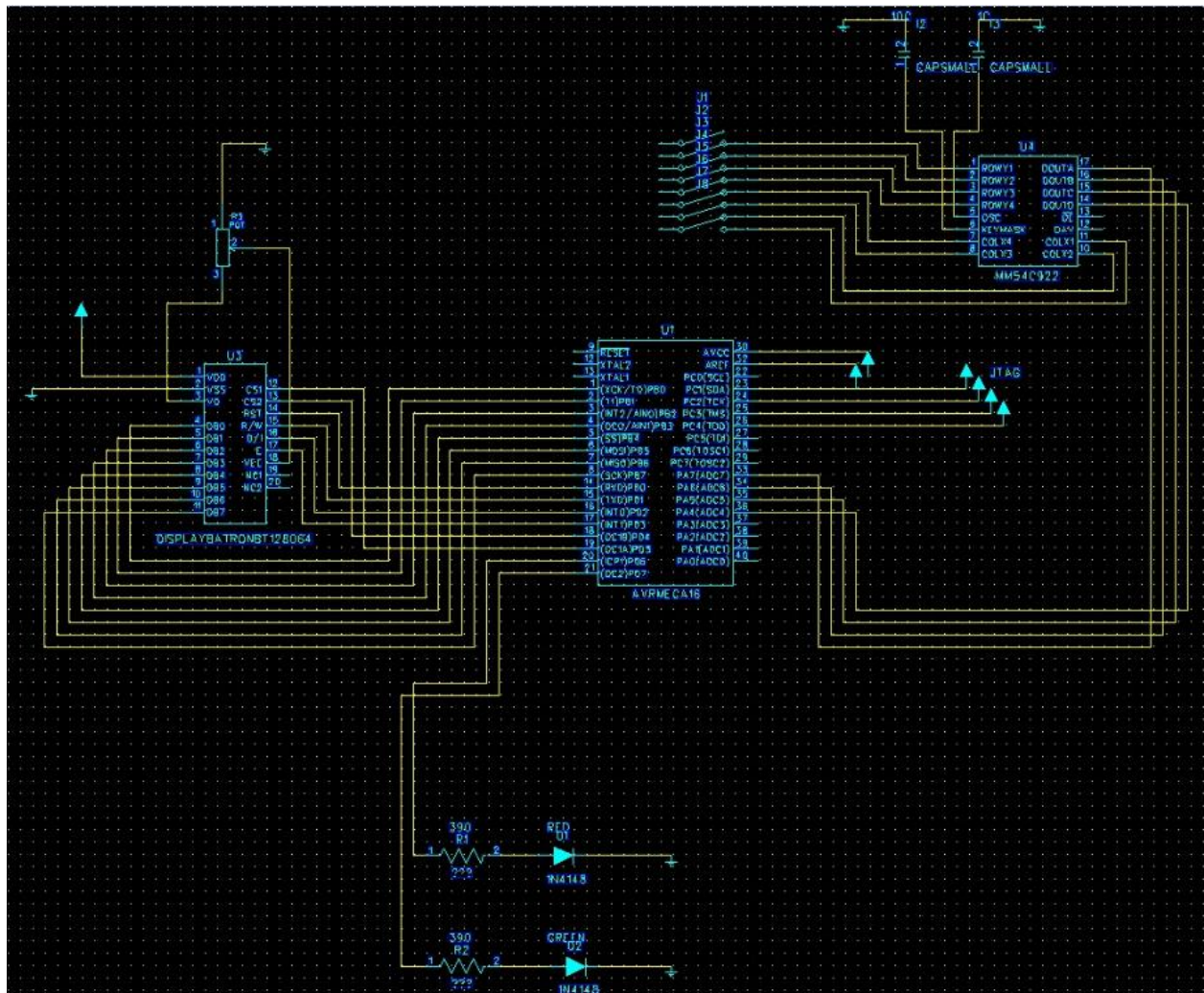
Den slutgiltiga konstruktionen uppfyllde alla de krav som fanns på den. Vissa av kraven krävde dock en smärre revidering från de ursprungliga för att göra implementeringen bättre och lättare att genomföra.

7. Diskussion och slutsats

Efter en relativt lång inlämningsprocess om hur hårdvaran fungerade och hur alla komponenter skulle höra ihop flöt projektet på bra. Projektet har varit intressant och lärorikt då det gett en djupare förståelse för hur konstruktionsarbete och produktutveckling går till och hur mjukvara kan kommunicera med hårdvara.

8. Bilagor

Bilaga 1. Kopplingschema



Figur 2 Kopplingschema

Bilaga 2. Källkod

```
/*
 * TicTac.c
 *
 * Created: 2016-04-28 10:11:07
 * Author: digpi16
 */

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

char val;
char virtualDisplay [8][128];
char gameStarted = 0;

char knapp1 = 0b11001111;
char knapp2 = 0b01001111;
char knapp3 = 0b10001111;
char knapp4 = 0b11101111;
char knapp5 = 0b01101111;
char knapp6 = 0b10101111;
char knapp7 = 0b11011111;
char knapp8 = 0b01011111;
char knapp9 = 0b10011111;

char sq1 = 0;
char sq2 = 0;
char sq3 = 0;
char sq4 = 0;
char sq5 = 0;
char sq6 = 0;
char sq7 = 0;
char sq8 = 0;
char sq9 = 0;
char win = 0;
char i = 1;
char countO = 0;
char countX = 0;
char stage = 0;

int main(void){
    setDataDirection();
    startDisplay();
    clearDisplay();
    setupBoard();
    eraseAll();
    while(1){
        if(win == 0){
            play();
        }
    }
}
```

```

while(win == 1){
    greenHigh();
    if(PINA == 0b00001111){
        eraseAll();
        sq1 = 0;
        sq2 = 0;
        sq3 = 0;
        sq4 = 0;
        sq5 = 0;
        sq6 = 0;
        sq7 = 0;
        sq8 = 0;
        sq9 = 0;
        win = 0;
        i = 1;
        count0 = 0;
        countX = 0;
        stage = 0;
        greenLow();
        redLow();
    }
}
}
}

```

```

void play(){
    while(stage == 0 && win == 0){
        checkWin();
        if(count0 == 3){
            stage = 1;
        }

        if(i == 1 && win == 0){
            checkWin();

            if(PINA == knapp1 && sq1 == 0){
                while(PINA == knapp1){
                }
                if(PINA == 0b00011111){
                    placeX(1,1);

                    sq1 = 1;
                    countX++;
                }
            }
        }
    }
}

```

```

        i = 0;
    }
}
if(PINA == knapp2 && sq2 == 0){
    while(PINA == knapp2){

        }
        if(PINA == 0b00011111){
            placeX(2,1);

            sq2 = 1;
            countX++;
            i = 0;

        }
    }
if(PINA == knapp3 && sq3 == 0){
    while(PINA == knapp3){
        }
        if(PINA == 0b00011111){
            placeX(3,1);
            i = 0;
            sq3 = 1;
            countX++;
            i = 0;

        }
    }
if(PINA == knapp4 && sq4 == 0){
    while(PINA == knapp4){

        }
        if(PINA == 0b00011111){
            placeX(1,2);

            sq4 = 1;
            countX++;
            i = 0;

        }
    }
if(PINA == knapp5 && sq5 == 0){
    while(PINA == knapp5){

        }
        if(PINA == 0b00011111){
            placeX(2,2);
            i = 0;
            sq5 = 1;
            countX++;

        }
    }
if(PINA == knapp6 && sq6 == 0){
    while(PINA == knapp6){

        }
        if(PINA == 0b00011111){
            placeX(3,2);

            sq6 = 1;
            countX++;

```

```

        i = 0;
    }
}
if(PINA == knapp7 && sq7 == 0){
    while(PINA == knapp7){

    }
    if(PINA == 0b00011111){
        placeX(1,3);

        sq7 = 1;
        countX++;
        i = 0;
    }
}
if(PINA == knapp8 && sq8 == 0){
    while(PINA == knapp8){

    }
    if(PINA == 0b00011111){
        placeX(2,3);

        sq8 = 1;
        countX++;
        i = 0;
    }
}
if(PINA == knapp9 && sq9 == 0){
    while(PINA == knapp9){

    }
    if(PINA == 0b00011111){
        placeX(3,3);

        sq9 = 1;
        countX++;
        i = 0;
    }
}
if(PINA == knapp1 && sq1 != 0){
    while(PINA == knapp1){
        redHigh();
    }
    if(PINA == 0b00011111){

        redLow();
        i = 1;
    }
}
if(PINA == knapp2 && sq2 != 0){
    while(PINA == knapp2){
        redHigh();
    }
    if(PINA == 0b00011111){

        redLow();
        i = 1;
    }
}

```

```

}
if(PINA == knapp3 && sq3 != 0){
    while(PINA == knapp3){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}
if(PINA == knapp4 && sq4 != 0){
    while(PINA == knapp4){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}
if(PINA == knapp5 && sq5 != 0){
    while(PINA == knapp5){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}
if(PINA == knapp6 && sq6 != 0){
    while(PINA == knapp6){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}
if(PINA == knapp7 && sq7 != 0){
    while(PINA == knapp7){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}
if(PINA == knapp8 && sq8 != 0){
    while(PINA == knapp8){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}

```

```

    }
}
if(PINA == knapp9 && sq9 != 0){
    while(PINA == knapp9){
        redHigh();
    }
    if(PINA == 0b00011111){
        redLow();
        i = 1;
    }
}
}
if(i == 0 && win == 0){
    checkWin();

    if(PINA == knapp1 && sq1 == 0){
        while(PINA == knapp1){
        }
        if(PINA == 0b00011111){
            place0(1,1);

            sq1 = 2;
            count0++;
            i = 1;
        }
    }
    if(PINA == knapp2 && sq2 == 0){
        while(PINA == knapp2){
        }
        if(PINA == 0b00011111){
            place0(2,1);

            sq2 = 2;
            count0++;
            i = 1;
        }
    }
    if(PINA == knapp3 && sq3 == 0){
        while(PINA == knapp3){
        }
        if(PINA == 0b00011111){
            place0(3,1);

            sq3 = 2;
            count0++;
            i = 1;
        }
    }
    if(PINA == knapp4 && sq4 == 0){
        while(PINA == knapp4){

```

```

    }
    if(PINA == 0b00011111){
        place0(1,2);

        sq4 = 2;
        count0++;
        i = 1;
    }
}
if(PINA == knapp5 && sq5 == 0){
    while(PINA == knapp5){

    }
    if(PINA == 0b00011111){
        place0(2,2);

        sq5 = 2;
        count0++;
        i = 1;
    }
}
if(PINA == knapp6 && sq6 == 0){
    while(PINA == knapp6){

    }
    if(PINA == 0b00011111){
        place0(3,2);

        sq6 = 2;
        count0++;
        i = 1;
    }
}
if(PINA == knapp7 && sq7 == 0){
    while(PINA == knapp7){

    }
    if(PINA == 0b00011111){
        place0(1,3);

        sq7 = 2;
        count0++;
        i = 1;
    }
}
if(PINA == knapp8 && sq8 == 0){
    while(PINA == knapp8){

    }
    if(PINA == 0b00011111){
        place0(2,3);

        sq8 = 2;
        count0++;
        i = 1;
    }
}

```

```

    }
}
if(PINA == knapp9 && sq9 == 0){
    while(PINA == knapp9){

        }
        if(PINA == 0b00011111){
            place0(3,3);

            sq9 = 2;
            count0++;
            i = 1;

        }
    }
if(PINA == knapp1 && sq1 != 0){
    while(PINA == knapp1){
        redHigh();
    }
    if(PINA == 0b00011111){

        redLow();
        i = 0;

    }
}
if(PINA == knapp2 && sq2 != 0){
    while(PINA == knapp9){
        redHigh();
    }
    if(PINA == 0b00011111){

        redLow();
        i = 0;

    }
}
if(PINA == knapp3 && sq3 != 0){
    while(PINA == knapp9){
        redHigh();
    }
    if(PINA == 0b00011111){

        redLow();
        i = 0;

    }
}
if(PINA == knapp4 && sq4 != 0){
    while(PINA == knapp9){
        redHigh();
    }
    if(PINA == 0b00011111){

        redLow();
        i = 0;

    }
}
if(PINA == knapp5 && sq5 != 0){
    while(PINA == knapp9){
        redHigh();
    }
}

```



```

        if(PINA == 0b00011111){
            redLow();
            i = 0;
        }
    }
    if(PINA == knapp6 && sq6 != 0){
        while(PINA == knapp9){
            redHigh();
        }
        if(PINA == 0b00011111){
            redLow();
            i = 0;
        }
    }
    if(PINA == knapp7 && sq7 != 0){
        while(PINA == knapp9){
            redHigh();
        }
        if(PINA == 0b00011111){
            redLow();
            i = 0;
        }
    }
    if(PINA == knapp8 && sq8 != 0){
        while(PINA == knapp9){
            redHigh();
        }
        if(PINA == 0b00011111){
            redLow();
            i = 0;
        }
    }
    if(PINA == knapp9 && sq9 != 0){
        while(PINA == knapp9){
            redHigh();
        }
        if(PINA == 0b00011111){
            redLow();
            i = 0;
        }
    }
}
while(stage == 1 && win == 0){

    if(i = 1 && win == 0){
        checkWin();

        if(PINA == knapp1 && sq1 == 1 && countX == 3){
            while(PINA == knapp1){

            }
        }
    }
}

```

```

        if(PINA == 0b00011111){
            eraseX(1,1);

            sq1 = 0;
            countX = 2;
        }
    }
    if(PINA == knapp2 && sq2 == 1 && countX == 3){
        while(PINA == knapp2){

        }
        if(PINA == 0b00011111){
            eraseX(2,1);

            sq2 = 0;
            countX = 2;
        }
    }
    if(PINA == knapp3 && sq3 == 1 && countX == 3){
        while(PINA == knapp3){

        }
        if(PINA == 0b00011111){
            eraseX(3,1);

            sq3 = 0;
            countX = 2;
        }
    }
    if(PINA == knapp4 && sq4 == 1 && countX == 3){
        while(PINA == knapp4){

        }
        if(PINA == 0b00011111){
            eraseX(1,2);

            sq4 = 0;
            countX = 2;
        }
    }
    if(PINA == knapp5 && sq5 == 1 && countX == 3){
        while(PINA == knapp5){

        }
        if(PINA == 0b00011111){
            eraseX(2,2);

            sq5 = 0;
            countX = 2;
        }
    }
    if(PINA == knapp6 && sq6 == 1 && countX == 3){
        while(PINA == knapp6){

        }
        if(PINA == 0b00011111){
            eraseX(3,2);

```

```

        sq6 = 0;
        countX = 2;
    }
}
if(PINA == knapp7 && sq7 == 1 && countX == 3){
    while(PINA == knapp7){

        }
        if(PINA == 0b00011111){
            eraseX(1,3);

            sq7 = 0;
            countX = 2;

        }
    }
if(PINA == knapp8 && sq8 == 1 && countX == 3){
    while(PINA == knapp8){

        }
        if(PINA == 0b00011111){
            eraseX(2,3);

            sq8 = 0;
            countX = 2;

        }
    }
if(PINA == knapp9 && sq9 == 1 && countX == 3){
    while(PINA == knapp9){

        }
        if(PINA == 0b00011111){
            eraseX(3,3);

            sq9 = 0;
            countX = 2;

        }
    }
}
if(PINA == knapp1 && sq1 == 0){
    if(countX == 2){
        while(PINA == knapp1){

            }
            if(PINA == 0b00011111){
                placeX(1,1);
                sq1 = 1;
                countX = 3;
                i = 0;

            }

        }
    }
}
if(PINA == knapp2 && sq2 == 0){
    if(countX == 2){
        while(PINA == knapp2){

```

```

        }
        if(PINA == 0b00011111){
            placeX(2,1);
            sq2 = 1;
            countX = 3;
            i = 0;
        }
    }

}
if(PINA == knapp3 && sq3 == 0){
    if(countX == 2){
        while(PINA == knapp3){
            }
            if(PINA == 0b00011111){
                placeX(3,1);
                sq3 = 1;
                countX = 3;
                i = 0;
            }
        }
    }

}
if(PINA == knapp4 && sq4 == 0){
    if(countX == 2){
        while(PINA == knapp4){
            }
            if(PINA == 0b00011111){
                placeX(1,2);
                sq4 = 1;
                countX = 3;
                i = 0;
            }
        }
    }

}
if(PINA == knapp5 && sq5 == 0){
    if(countX == 2){
        while(PINA == knapp5){
            }
            if(PINA == 0b00011111){
                placeX(2,2);
                sq5 = 1;
                countX = 3;
                i = 0;
            }
        }
    }
}

```

```

}
if(PINA == knapp6 && sq6 == 0){
    if(countX == 2){
        while(PINA == knapp6){
            }
            if(PINA == 0b00011111){
                placeX(3,2);
                sq6 = 1;
                countX = 3;
                i = 0;
            }
        }
    }
}

if(PINA == knapp7 && sq7 == 0){
    if(countX == 2){
        while(PINA == knapp7){
            }
            if(PINA == 0b00011111){
                placeX(1,3);
                sq7 = 1;
                countX = 3;
                i = 0;
            }
        }
    }
}

if(PINA == knapp8 && sq8 == 0){
    if(countX == 2){
        while(PINA == knapp8){
            }
            if(PINA == 0b00011111){
                placeX(2,3);
                sq8 = 1;
                countX = 3;
                i = 0;
            }
        }
    }
}

if(PINA == knapp9 && sq9 == 0){
    if(countX == 2){
        while(PINA == knapp9){

```

```

        }
        if(PINA == 0b00011111){
            placeX(3,3);
            sq9 = 1;
            countX = 3;
            i = 0;
        }
    }

}

}
while(i == 0 && win == 0){
    checkWin();
    if(count0 == 3){

        if(PINA == knapp1 && sq1 == 2){
            while(PINA == knapp1){

            }
            if(PINA == 0b00011111){
                erase0(1,1);

                sq1 = 0;
                count0 = 2;

            }
        }
        if(PINA == knapp2 && sq2 == 2){
            while(PINA == knapp2){

            }
            if(PINA == 0b00011111){
                erase0(2,1);

                sq2 = 0;
                count0 = 2;

            }
        }
        if(PINA == knapp3 && sq3 == 2){
            while(PINA == knapp3){

            }
            if(PINA == 0b00011111){
                erase0(3,1);

                sq3 = 0;
                count0 = 2;

            }
        }
        if(PINA == knapp4 && sq4 == 2){
            while(PINA == knapp4){

            }
            if(PINA == 0b00011111){

```

```

erase0(1,2);

sq4 = 0;
count0 = 2;
}
}
if(PINA == knapp5 && sq5 == 2){
    while(PINA == knapp5){
        }
        if(PINA == 0b00011111){
            erase0(2,2);

            sq5 = 0;
            count0 = 2;
        }
    }
}
if(PINA == knapp6 && sq6 == 2){
    while(PINA == knapp6){
        }
        if(PINA == 0b00011111){
            erase0(3,2);

            sq6 = 0;
            count0 = 2;
        }
    }
}
if(PINA == knapp7 && sq7 == 2){
    while(PINA == knapp7){
        }
        if(PINA == 0b00011111){
            erase0(1,3);

            sq7 = 0;
            count0 = 2;
        }
    }
}
if(PINA == knapp8 && sq8 == 2){
    while(PINA == knapp8){
        }
        if(PINA == 0b00011111){
            erase0(2,3);

            sq8 = 0;
            count0 = 2;
        }
    }
}
if(PINA == knapp9 && sq9 == 2){
    while(PINA == knapp9){
        }
        if(PINA == 0b00011111){
            erase0(3,3);

```

```

sq9 = 0;
count0 = 2;
    }
}
}
if(count0 == 2){
    if(PINA == knapp1 && sq1 == 0){
        while(PINA == knapp1){
            }
            if(PINA == 0b00011111){
                place0(1,1);
                sq1 = 2;
                count0 = 3;
                i = 1;
            }
        }
    }
    if(PINA == knapp2 && sq2 == 0){
        while(PINA == knapp2){
            }
            if(PINA == 0b00011111){
                place0(2,1);
                sq2 = 2;
                count0 = 3;
                i = 1;
            }
        }
    }
    if(PINA == knapp3 && sq3 == 0){
        while(PINA == knapp3){
            }
            if(PINA == 0b00011111){
                place0(3,1);
                sq3 = 2;
                count0 = 3;
                i = 1;
            }
        }
    }
    if(PINA == knapp4 && sq4 == 0){
        while(PINA == knapp4){
            }
            if(PINA == 0b00011111){
                place0(1,3);
                sq4 = 2;
                count0 = 3;
                i = 1;
            }
        }
    }
    if(PINA == knapp5 && sq5 == 0){
        while(PINA == knapp5){

```



```

    }
    if(PINA == 0b00011111){
        place0(2,2);
        sq5 = 2;
        count0 = 3;
        i = 1;
    }
}
if(PINA == knapp6 && sq6 == 0){
    while(PINA == knapp6){

    }
    if(PINA == 0b00011111){
        place0(3,2);
        sq6 = 2;
        count0 = 3;
        i = 1;
    }
}
}
if(PINA == knapp7 && sq7 == 0){
    while(PINA == knapp7){

    }
    if(PINA == 0b00011111){
        place0(1,3);
        sq7 = 2;
        count0 = 3;
        i = 1;
    }
}
}
if(PINA == knapp8 && sq8 == 0){
    while(PINA == knapp8){

    }
    if(PINA == 0b00011111){
        place0(2,3);
        sq8 = 2;
        count0 = 3;
        i = 1;
    }
}
}
if(PINA == knapp9 && sq9 == 0){
    while(PINA == knapp9){

    }
    if(PINA == 0b00011111){
        place0(3,3);
        sq9 = 2;
        count0 = 3;
        i = 1;
    }
}
}

```

```

    }
}

void checkWin(){
    if (sq1 == 1 && sq2 == 1 && sq3 == 1){
        win = 1;
    }
    if ( sq4 == 1 && sq5 == 1 && sq6 == 1){
        win = 1;
    } if (sq7 == 1 && sq8 == 1 && sq9 == 1){
        win = 1;
    } if (sq1 == 1 && sq4 == 1 && sq7 == 1){
        win = 1;
    } if (sq2 == 1 && sq5 == 1 && sq8 == 1){
        win = 1;
    } if (sq3 == 1 && sq6 == 1 && sq9 == 1){
        win = 1;
    } if (sq1 == 1 && sq5 == 1 && sq9 == 1){
        win = 1;
    } if (sq3 == 1 && sq5 == 1 && sq7 == 1){
        win = 1;
    } if (sq1 == 2 && sq2 == 2 && sq3 == 2){
        win = 1;
    } if (sq4 == 2 && sq5 == 2 && sq6 == 2){
        win = 1;
    } if (sq7 == 2 && sq8 == 2 && sq9 == 2){
        win = 1;
    } if (sq1 == 2 && sq4 == 2 && sq7 == 2){
        win = 1;
    } if (sq2 == 2 && sq5 == 2 && sq8 == 2){
        win = 1;
    } if (sq3 == 2 && sq6 == 2 && sq9 == 2){
        win = 1;
    } if (sq1 == 2 && sq5 == 2 && sq9 == 2){
        win = 1;
    } if (sq3 == 2 && sq5 == 2 && sq7 == 2){
        win = 1;
    }
}

void eraseAll(){
    erase0(1,1);
    erase0(1,2);
    erase0(1,3);
    erase0(2,1);
    erase0(2,2);
    erase0(2,3);
    erase0(3,1);
    erase0(3,2);
    erase0(3,3);
    eraseX(1,1);
    eraseX(1,2);
    eraseX(1,3);
    eraseX(2,1);
}

```

```

        eraseX(2,2);
        eraseX(2,3);
        eraseX(3,1);
        eraseX(3,2);
        eraseX(3,3);
    }

void enableInterrupt() {
    PINA = 0b00000000;
    DDRA = 0b00000000;
    MCUCR |= 0b00001111;
    GICR |= 0b11000000;
    sei();
}

redHigh(){
    PORTD |= 0b10000000;
}

redLow(){
    PORTD &= 0b01111111;
}

greenHigh(){
    PORTD |= 0b01000000;
}

greenLow(){
    PORTD &= 0b10111111;
}

void cs2high() {
    PORTD |= 0b00100000;
}

void cs2low() {
    PORTD &= 0b11011111;
}

void cs1high() {
    PORTD |= 0b00010000;
}

void cs1low() {
    PORTD &= 0b11101111;
}

void resetHigh() {
    PORTD |= 0b00000001;
}

void resetLow() {
    PORTD &= 0b11111110;
}

void rwHigh() {
    PORTD |= 0b00000010;
}

void rwLow() {

```

```

        PORTD &= 0b1111101;
    }

    void rsHigh() {
        PORTD |= 0b00000100;
    }

    void rsLow() {
        PORTD &= 0b11111011;
    }

    void eHigh () {
        PORTD |= 0b00001000;
    }

    void eLow() {
        PORTD &= 0b11110111;
    }
    void setX(char x) {
        rsLow();
        rwLow();

        PORTB = 0b10111000 | x;

        eLow();
        eHigh();
    }

    void setY(char y) {
        rsLow();
        rwLow();

        PORTB = 0b01000000 | y;

        eLow();
        eHigh();
    }
}
void writeData(char x, char y) {
    rsHigh();
    rwLow();

    PORTB = virtualDisplay[x / 8][y];

    eLow();
    eHigh();
}

void eraseData(char x, char y) {
    rsHigh();
    rwLow();

    PORTB = virtualDisplay[x / 8][y];

    eLow();
    eHigh();
}

```

```

}

void draw(char x, char y) {
    virtualDisplay[x/8][y] |= (1<<(x%8));

    char newY;

    if (y < 64) {
        cs2high();
        cs1low();
        newY = y;
    } else {
        cs1high();
        cs2low();
        newY = y - 64;
    }

    setX(x / 8);
    setY(newY);

    writeData(x,y);
}

void drawBoard() {
    for (char y = 64; y < 128; y++) {
        draw(0, y);
        draw(63, y);
    }
    for (char x = 0; x < 63; x++) {
        draw(x, 64);
        draw(x, 127);
    }
    for (char y = 64; y < 128; y++) {
        draw(42, y);
        draw(21, y);
    }
    for (char x = 0; x < 63; x++) {
        draw(x, 85);
        draw(x, 106);
    }
}

void setupBoard() {
    drawBoard();
}

void erase(char x, char y) {
    virtualDisplay[x/8][y] &= ~(1<<(x%8));

    char newY;

    if (y < 64) {
        cs2high();
        cs1low();
        newY = y;
    } else {
        cs1high();

```

```

        cs2low();
        newY = y - 64;
    }

    setX(x / 8);
    setY(newY);

    eraseData(x,y);
}

void clearDisplay() {
    for (char y = 0; y < 128; y++) {
        for (char x = 0; x < 64; x++) {
            erase(x,y);
        }
    }
}

void placeX(char x, char y){
    if(x == 1 && y == 1){
        char j = 65;
        for(char i = 1; i < 21; i++){
            draw(i,j);
            j++;
        }
        char k = 84;
        for(char f = 1; f < 21; f++){
            draw(f,k);
            k--;
        }
    }
    if(x == 2 && y == 1){
        char j = 86;
        for(char i = 1; i < 21; i++){
            draw(i,j);
            j++;
        }
        char k = 105;
        for(char f = 1; f < 21; f++){
            draw(f,k);
            k--;
        }
    }
    if(x == 3 && y == 1){
        char j = 107;
        for(char i = 1; i < 21; i++){
            draw(i,j);
            j++;
        }
        char k = 126;
        for(char f = 1; f < 21; f++){
            draw(f,k);
            k--;
        }
    }
}

```

```

if(x == 1 && y == 2){
    char j = 65;
    for(char i = 22; i < 42; i++){
        draw(i,j);
        j++;
    }
    char k = 84;
    for(char f = 22; f < 42; f++){
        draw(f,k);
        k--;
    }
}
if(x == 2 && y == 2){
    char j = 86;
    for(char i = 22; i < 42; i++){
        draw(i,j);
        j++;
    }
    char k = 105;
    for(char f = 22; f < 42; f++){
        draw(f,k);
        k--;
    }
}
if(x == 3 && y == 2){
    char j = 107;
    for(char i = 22; i < 42; i++){
        draw(i,j);
        j++;
    }
    char k = 126;
    for(char f = 22; f < 42; f++){
        draw(f,k);
        k--;
    }
}
if(x == 1 && y == 3){
    char j = 65;
    for(char i = 43; i < 63; i++){
        draw(i,j);
        j++;
    }
    char k = 84;
    for(char f = 43; f < 63; f++){
        draw(f,k);
        k--;
    }
}
if(x == 2 && y == 3){
    char j = 86;
    for(char i = 43; i < 63; i++){
        draw(i,j);
        j++;
    }
}

```

```

        char k = 105;
        for(char f = 43; f < 63; f++){
            draw(f,k);
            k--;
        }
    }

    if(x == 3 && y == 3){
        char j = 107;
        for(char i = 43; i < 63; i++){
            draw(i,j);
            j++;
        }
        char k = 126;
        for(char f = 43; f < 63; f++){
            draw(f,k);
            k--;
        }
    }
}

void eraseX(char x, char y){
    if(x == 1 && y == 1){
        char j = 65;
        for(char i = 1; i < 21; i++){
            erase(i,j);
            j++;
        }
        char k = 84;
        for(char f = 1; f < 21; f++){
            erase(f,k);
            k--;
        }
    }

    if(x == 2 && y == 1){
        char j = 86;
        for(char i = 1; i < 21; i++){
            erase(i,j);
            j++;
        }
        char k = 105;
        for(char f = 1; f < 21; f++){
            erase(f,k);
            k--;
        }
    }

    if(x == 3 && y == 1){
        char j = 107;
        for(char i = 1; i < 21; i++){
            erase(i,j);
            j++;
        }
        char k = 126;
        for(char f = 1; f < 21; f++){
            erase(f,k);
            k--;
        }
    }
}

```



```

    }
}

if(x == 1 && y == 2){
    char j = 65;
    for(char i = 22; i < 42; i++){
        erase(i,j);
        j++;
    }
    char k = 84;
    for(char f = 22; f < 42; f++){
        erase(f,k);
        k--;
    }
}

if(x == 2 && y == 2){
    char j = 86;
    for(char i = 22; i < 42; i++){
        erase(i,j);
        j++;
    }
    char k = 105;
    for(char f = 22; f < 42; f++){
        erase(f,k);
        k--;
    }
}

if(x == 3 && y == 2){
    char j = 107;
    for(char i = 22; i < 42; i++){
        erase(i,j);
        j++;
    }
    char k = 126;
    for(char f = 22; f < 42; f++){
        erase(f,k);
        k--;
    }
}

if(x == 1 && y == 3){
    char j = 65;
    for(char i = 43; i < 63; i++){
        erase(i,j);
        j++;
    }
    char k = 84;
    for(char f = 43; f < 63; f++){
        erase(f,k);
        k--;
    }
}

if(x == 2 && y == 3){
    char j = 86;
    for(char i = 43; i < 63; i++){

```

```

        erase(i,j);
        j++;
    }
    char k = 105;
    for(char f = 43; f < 63; f++){
        erase(f,k);
        k--;
    }
}

if(x == 3 && y == 3){
    char j = 107;
    for(char i = 43; i < 63; i++){
        erase(i,j);
        j++;
    }
    char k = 126;
    for(char f = 43; f < 63; f++){
        erase(f,k);
        k--;
    }
}

}

void place0(char x, char y){
    char newX = y - 1;
    char newY = x - 1;
    for(char i = 0; i < 6; i++){
        draw(8 + 21*newX + i, 65 + 21*newY);
    }
    for(char i = 0; i < 6; i++){
        draw(8 + 21*newX + i, 84 + 21*newY);
    }
    for(char i = 0; i < 6; i++){
        draw(1 + 21*newX, 72 + 21*newY + i);
    }
    for(char i = 0; i < 6; i++){
        draw(20 + 21*newX, 72 + 21*newY + i);
    }

    draw(2 + newX*21,70 + newY*21);
    draw(2 + newX*21,71 + newY*21);
    draw(2 + newX*21,78 + newY*21);
    draw(2 + newX*21,79 + newY*21);
    draw(19 + newX*21,70 + newY*21);
    draw(19 + newX*21,71 + newY*21);
    draw(19 + newX*21,78 + newY*21);
    draw(19 + newX*21,79 + newY*21);
    draw(6 + newX*21,66 + newY*21);
    draw(7 + newX*21,66 + newY*21);
    draw(14 + newX*21,66 + newY*21);
    draw(15 + newX*21,66 + newY*21);
    draw(6 + newX*21,83 + newY*21);
    draw(7 + newX*21,83 + newY*21);
    draw(14 + newX*21,83 + newY*21);
    draw(15 + newX*21,83 + newY*21);
    draw(5 + newX*21, 67 + newY*21);
    draw(4 + newX*21, 68 + newY*21);
}

```

```

draw(3 + newX*21, 69 + newY*21);
draw(5 + newX*21, 82 + newY*21);
draw(4 + newX*21, 81 + newY*21);
draw(3 + newX*21, 80 + newY*21);
draw(16 + newX*21, 67 + newY*21);
draw(17 + newX*21, 68 + newY*21);
draw(18 + newX*21, 69 + newY*21);
draw(16 + newX*21, 82 + newY*21);
draw(17 + newX*21, 81 + newY*21);
draw(18 + newX*21, 80 + newY*21);
}

void erase0(char x, char y){
    char newX = y - 1;
    char newY = x - 1;
    for(char i = 0; i < 6; i++){
        erase(8 + 21*newX + i, 65 + 21*newY);
    }
    for(char i = 0; i < 6; i++){
        erase(8 + 21*newX + i, 84 + 21*newY);
    }
    for(char i = 0; i < 6; i++){
        erase(1 + 21*newX, 72 + 21*newY + i);
    }
    for(char i = 0; i < 6; i++){
        erase(20 + 21*newX, 72 + 21*newY + i);
    }

    erase(2 + newX*21,70 + newY*21);
    erase(2 + newX*21,71 + newY*21);
    erase(2 + newX*21,78 + newY*21);
    erase(2 + newX*21,79 + newY*21);
    erase(19 + newX*21,70 + newY*21);
    erase(19 + newX*21,71 + newY*21);
    erase(19 + newX*21,78 + newY*21);
    erase(19 + newX*21,79 + newY*21);
    erase(6 + newX*21,66 + newY*21);
    erase(7 + newX*21,66 + newY*21);
    erase(14 + newX*21,66 + newY*21);
    erase(15 + newX*21,66 + newY*21);
    erase(6 + newX*21,83 + newY*21);
    erase(7 + newX*21,83 + newY*21);
    erase(14 + newX*21,83 + newY*21);
    erase(15 + newX*21,83 + newY*21);
    erase(5 + newX*21, 67 + newY*21);
    erase(4 + newX*21, 68 + newY*21);
    erase(3 + newX*21, 69 + newY*21);
    erase(5 + newX*21, 82 + newY*21);
    erase(4 + newX*21, 81 + newY*21);
    erase(3 + newX*21, 80 + newY*21);
    erase(16 + newX*21, 67 + newY*21);
    erase(17 + newX*21, 68 + newY*21);
    erase(18 + newX*21, 69 + newY*21);
    erase(16 + newX*21, 82 + newY*21);
    erase(17 + newX*21, 81 + newY*21);
    erase(18 + newX*21, 80 + newY*21);
}

```

```

void setDataDirection() {
    DDRD = 0b11111111;
    DDRB = 0b11111111;
}

void startDisplay() {
    PORTB = 0b00111111;

    cs1low();
    cs2low();

    resetHigh();
    rwLow();
    rsLow();
    eHigh();

    cs1high();
    cs2high();

    eLow();
    eHigh();
}

void stopDisplay() {
    PORTB = 0b00111110;

    cs1low();
    cs2low();

    resetHigh();
    rwLow();
    rsLow();
    eHigh();

    cs1high();
    cs2high();

    eLow();
    eHigh();
}

```