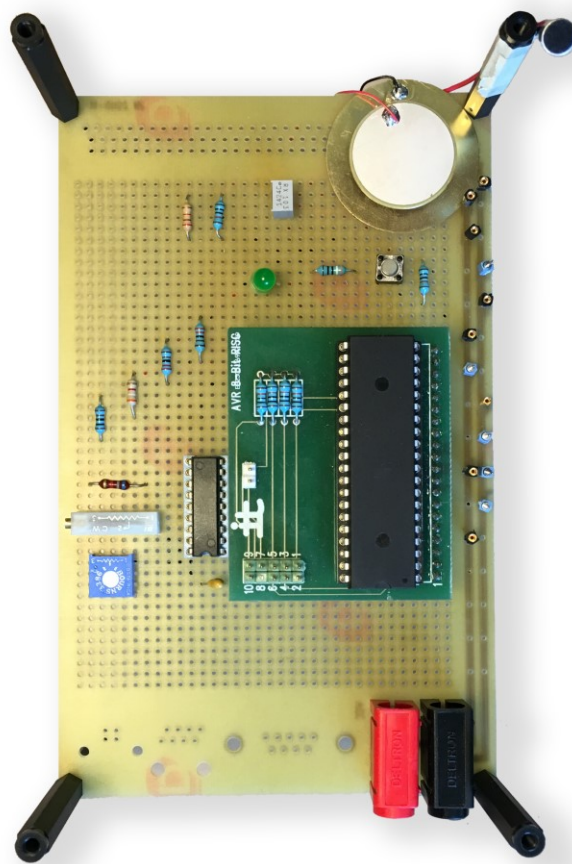


Lunds Tekniska Högskola
EITF11 Digitala projekt

Bandspelaren



Johan Hansson & Cecilia Skoglund
22 maj 2017

Abstract

This is a report describing the work process, hardware, software and results of a project in the course EITF11 "Digitala Projekt". The requirement specification set up in the beginning of the project included being able to record and reproduce sound as well as play melodies that were coded beforehand. Part of the requirements specifications were met, resulting in a melody-player that plays four different melodies. Due to lack of success with the microphone, recording of sound was not achieved and consequently neither was the reproduction of recorded sound.

Innehållsförteckning

ABSTRACT	1
INNEHÅLLSFÖRTECKNING	2
1 INLEDNING	3
2 TEORI.....	3
3 GENOMFÖRANDE	4
4 RESULTAT.....	5
5 UTVÄRDERANDE DISKUSSION	5
6 REFERENSER.....	6
APPENDIX A.....	6
APPENDIX B.....	7

1 Inledning

1.1 Projektintroduktion

Projektet att bygga en bandspelare ingick i kursen EITF11 Digitala Projekt (10 HP). Till hands fanns en ATmega16 8-bitars mikrokontroller och en uppsjö tillbehör. Det var upp till varje specifik grupp att välja vad man vill skapa utifrån hårdvaran.

Det bestämdes att en typ av bandspelare skulle byggas. Se kravspecifikationen i avsnitt 2.1 för vidare detaljer. Kravspecifikationen uppnåddes delvis och en fungerande melodispelare av förprogrammerade melodier skapades.

1.2 Syfte

Syftet med rapporten är att redovisa projektarbetets gång från planering till sammansättning av hårdvara, kodning och testning. Den är även till för att reflektera över lärdomar, motgångar och framgångar.

Syftet med projektet var att lära om hårdvara och mjukvara, och hur de två hänger ihop. Genom ett praktiskt, nyfiket och prövande tillvägagångssätt erhöles ny kunskap på ett kreativt och självlärande sätt.

1.3 Rapportens disposition

Rapporten består av en inledning med kort projekt- och syftesbeskrivning. Därefter följer en teoridel där kravspecifikationen för produkten presenteras, samt den hårdvara och mjukvara som använts i skapandet. Efter det följer genomförandet, där processen genom projektets olika stadier beskrivs från första planerandet till slutlig testning av den färdiga produkten. I resultatdelen beskrivs funktionaliteten, sammansättningen och användarmanualen för den färdiga produkten. Den utvärderande diskussionen innehåller reflektioner över arbetets gång och resultat. Därpå följer referenser och slutligen appendix.

2 Teori

2.1 Kravspecifikation

Bandspelaren hade följande kravspecifikation:

Bandspelaren ska kunna...

2.1.1 – Spela in kortare ljudsegment via en mikrofon

2.1.2 – Återge sagda ljudsegment i en högtalare

2.1.3 – Spela upp fyra olika förprogrammerade melodier

2.1.4 – Kunna växla mellan de olika melodierna efter användarens önskemål

2.1.5 – Kunna vara tyst/vilande

2.2 Hårdvara

Till produkten användes följande hårdvara:

- 1st ATmega16 8-bits mikroprocessor
- 1st D/A-omvandlare D7011C
- 1st mikrofon
- 1st piezo-högtalare
- 1st grön LED-lampa
- 2st 10 k Ω motstånd
- 1st 7.5 k Ω motstånd
- 1st 3 k Ω motstånd
- 1st 2.7 k Ω motstånd
- 2st 2.2 k Ω motstånd
- 1st 100 k Ω potentiometer
- 1st 10 k Ω potentiometer
- 1st knapp/brytare

Kopplingschema återfinns i Appendix A.

2.3 Mjukvara

För att kunna programmera bandspelaren har programvaran Atmel Studio använts. Programmeringsspråket är genomgående C. Koden i sin helhet återfinns i Appendix B.

3 Genomförande

3.1 Planering

Då det beslutats att en bandspelare i någon form skulle skapas togs kravspecifikationen fram. Då den godkännts av handledare påbörjades arbetet med att sätta sig in i hårdvarans funktionalitet, för att skapa kopplingschemat.

3.2 Montering

När handledaren hade godkänt kopplingschemat påbörjades monteringen av de olika komponenterna. Det löddes och virade kablar mellan samtliga komponenter. Testning av funktionaliteten gjordes genom att kontrollera att lampan som kopplats in lös.

3.3 Kodning & Testning

Då första monteringen var klar påbörjades kodningen, för att testa hårdvaran och sätta den i arbete. Det tog mycket lång tid att få D/A-omvandlaren att fungera på önskat vis. Flera gånger fick kopplingschemat ritas om och produktens olika delar testades självständigt, för att felsöka. Slutligen byttes vad som hade varit en högtalare ut till en piezo-högtalare, vilken belastade D/A-omvandlaren i mycket mindre utsträckning. När det gjorts kunde för första gången ljud åstadkommas genom en enkel kod.

Åtskilliga försök att få A/D-omvandlaren att uppfatta skillnader i signal från mikrofonen gjordes. Mikrofonens duglighet undersöktes i ett oscilloskop och

påvisade att mikrofonen fungerade som önskat. Trots det lyckades inte en korrekt signal plockas upp av A/D-omvandlaren från mikrofonen.

Arbetet fokuserades därför i slutskedet på att programmera tillfredställande och intresseväckande melodier, samt se till att man på ett pedagogiskt och simpelt sätt kunde växla mellan de olika låtarna.

4 Resultat

Resultatet blev något som liknar en mp3-spelare med 4 olika melodier. Kraven 2.1.3-5 uppnåddes medan krav 2.1.1 och 2.1.2 ej blev uppfyllda. De fyra melodierna är:

1. Glassbilen
2. Blinka lilla stjärna
3. Levels
4. One

De spelas i ovannämnda ordning. Varje låt loopas så länge som användaren önskar. Lampan blinkar i låtens rytm. Då nästa låt ska spelas trycker användaren på knappen. En kort slinga spelas upp för att indikera att ett låtskifte sker. Då användaren trycker på knappen då den sista låten, One, spelas blir spelaren tyst. För att starta om melodierna från början trycker användaren på knappen igen.

5 Utvärderande diskussion

Planeringen och monteringen av produkten var rolig och mycket lärorik. Det gick förvånansvärt fort att få både knapp och lampa att fungera och bete sig enligt önskemål genom kodning. Sedan blev det tung uppförsbacke då det tog mycket lång tid att få D/A-omvandlaren att fungera. Efter en hel del eftersökningar online samt hjälp från handledaren konstaterades till sist av D/A-omvandlaren fungerade men mycket, mycket svagt. När endast en vecka av projekttiden återstod insågs att högtalaren som kopplats in belastade D/A-omvandlaren alldeles för mycket, och gjorde att den analoga signalen till högtalaren blev för svag. Då kopplades istället en piezo-högtalare in och projektarbetet fick nytt liv då ljud för första gången skapades.

Parallellt med arbetet med D/A-omvandlaren pågick ett arbete att få A/D-omvandlaren att fungera. Tyvärr kunde aldrig en tillräckligt stark signal från mikrofonen uppfattas av A/D-omvandlaren för att en inspelning skulle kunna göras.

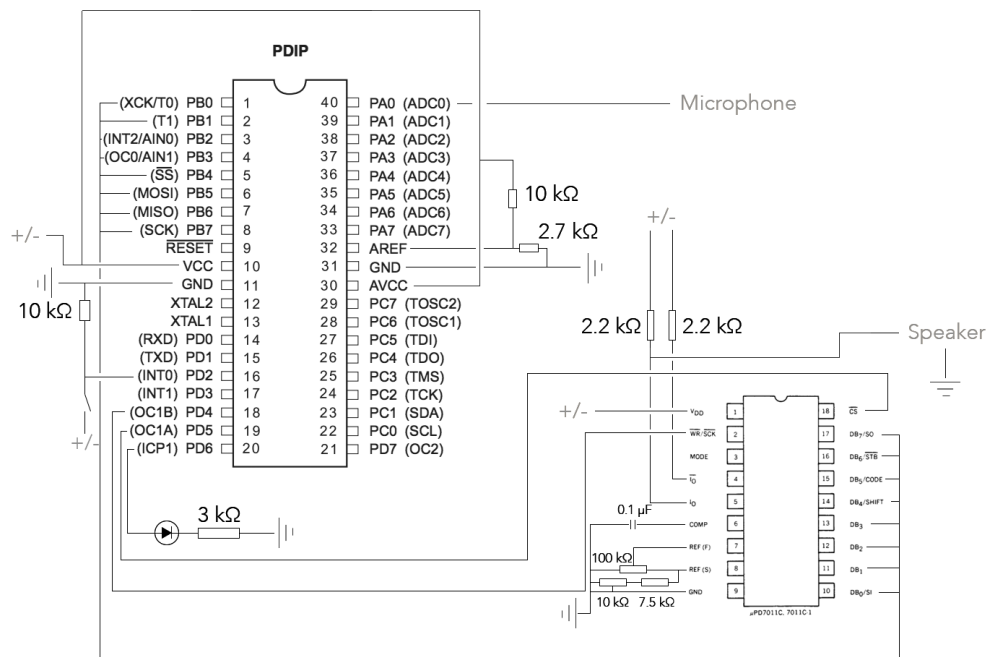
Att få koda kända melodier var ett kreativt och roligt sätt att få jobba med kod och hårdvara. Till en början kodades enkla, klassiska melodier som "Blinka lilla stjärna" och "Ha den äran idag". Mot slutet testades gränserna av piezo-högtalaren genom att skapa mer komplexa ljud som en hyllning till det svenska moderna House-musik-undret.

6 Referenser

[ATmega16\(L\) datablad](#)

[D7011C D/A converter datablad](#)

Appendix A



Figur 1: Konstruktionens kopplingschema

Appendix B

```
/*
 * test.c
 *
 * Created: 2017-04-12 11:22:42
 * Author : Cecilia Skoglund och Johan Hansson
 */

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

int main(void){

    DDRD = 0b00110000;
    PINA = 0x01;
    DDRB = 0xFF;
    int melodyCounter = 1;
    int loopToneActive = 0;

    void bOn(){
        PORTD = 0b01010000;
        PORTD = 0b01000000;
        PORTB = 0b11111111;
        PORTD = 0b01010000;
        PORTD = 0b01110000;
    }

    void bOff(){
        PORTD = 0b01010000;
        PORTD = 0b01000000;
        PORTB = 0b00000000;
        PORTD = 0b01010000;
        PORTD = 0b01110000;
    }

    void generateTone(double delay, int iteration){
        if(loopToneActive==0){_delay_ms(200);}
        for(int i = 0;i<iteration;i++){
            bOn();
            _delay_ms(delay);
            bOff();
            _delay_ms(delay);
        }
        PORTD = 0b00000000;
    }

    void loopTone(){
        loopToneActive==1;
        for(int i = 5;i<20;i++){
            generateTone(i,15);
        }
        for(int i = 20;i>3;i--){
            generateTone(i,15);
        }
        loopToneActive==0;
    }

    int checkButton(){
        if(PIND & (1<<PD2)){
            melodyCounter++;
            loopTone();
            _delay_ms(200);
            return 1;
        }
    }
}
```



```

    }
    return 0;
}

void Feliz(){
    while(1){
        /* D */
        generateTone(14,60);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(14,60);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,120);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(14,100);
        if(checkButton()==1){
            break;
        }
        /* G */
        generateTone(10,150);
        if(checkButton()==1){
            break;
        }
        /* F# */
        generateTone(10.5,300);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(14,60);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(14,60);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,120);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(14,100);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,130);
        if(checkButton()==1){
            break;
        }
        /* G */
        generateTone(10,300);
        if(checkButton()==1){
            break;
        }
    }
}

```

```

/* D */
generateTone(14,60);
if(checkButton()==1){
    break;
}
/* D */
generateTone(14,60);
if(checkButton()==1){
    break;
}
/* D */
generateTone(5.3,250);
if(checkButton()==1){
    break;
}
/* B */
generateTone(8,200);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,200);
if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,200);
if(checkButton()==1){
    break;
}
/* E */
generateTone(12,150);
if(checkButton()==1){
    break;
}
/* C */
generateTone(7.5,100);
if(checkButton()==1){
    break;
}
/* C */
generateTone(7.5,100);
if(checkButton()==1){
    break;
}
/* B */
generateTone(8,200);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,200);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,150);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,300);
if(checkButton()==1){
    break;
}
}
_delay_ms(200);

```

```

    }
}

void Twinkle(){
    while(1){
        /* D */
        generateTone(13.8,150);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(13.8,150);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,200);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,200);
        if(checkButton()==1){
            break;
        }
        /* B */
        generateTone(8,200);
        if(checkButton()==1){
            break;
        }
        /* B */
        generateTone(8,200);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,300);
        if(checkButton()==1){
            break;
        }
        /* G */
        generateTone(10,180);
        if(checkButton()==1){
            break;
        }
        /* G */
        generateTone(10,180);
        if(checkButton()==1){
            break;
        }
        /* F# */
        generateTone(10.7,175);
        if(checkButton()==1){
            break;
        }
        /* F# */
        generateTone(10.7,175);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,160);
        if(checkButton()==1){
            break;
        }
        /* E */
    }
}

```

```

generateTone(12,160);
if(checkButton()==1){
    break;
}
/* D */
generateTone(13.8,150);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,200);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,200);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,180);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,180);
if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,175);
if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,175);
if(checkButton()==1){
    break;
}
/* E */
generateTone(12,160);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,200);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,200);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,180);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,180);
if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,175);

```

```

if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,175);
if(checkButton()==1){
    break;
}
/* E */
generateTone(12,160);
if(checkButton()==1){
    break;
}
/* D */
generateTone(13.8,150);
if(checkButton()==1){
    break;
}
/* D */
generateTone(13.8,150);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,200);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,200);
if(checkButton()==1){
    break;
}
/* B */
generateTone(8,200);
if(checkButton()==1){
    break;
}
/* B */
generateTone(8,200);
if(checkButton()==1){
    break;
}
/* A */
generateTone(9,300);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,180);
if(checkButton()==1){
    break;
}
/* G */
generateTone(10,180);
if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,175);
if(checkButton()==1){
    break;
}
/* F# */
generateTone(10.7,175);
if(checkButton()==1){

```

```

        break;
    }
    /* E */
    generateTone(12,160);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,160);
    if(checkButton()==1){
        break;
    }
    /* D */
    generateTone(13.8,150);
    if(checkButton()==1){
        break;
    }
    _delay_ms(200);
}
}

void Glassbilen(){
    while(1){
        /* F */
        generateTone(11.5,80);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,100);
        if(checkButton()==1){
            break;
        }
        /* C */
        generateTone(7.5,100);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,100);
        if(checkButton()==1){
            break;
        }
        /* F */
        generateTone(11.5,160);
        if(checkButton()==1){
            break;
        }
        /* F */
        generateTone(11.5,80);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,100);
        if(checkButton()==1){
            break;
        }
        /* C */
        generateTone(7.5,100);
        if(checkButton()==1){
            break;
        }
        /* A */
        generateTone(9,100);
        if(checkButton()==1){

```

```

        break;
    }
    /* F */
    generateTone(11.5,160);
    if(checkButton()==1){
        break;
    }
    /* F */
    generateTone(11.5,80);
    if(checkButton()==1){
        break;
    }
    /* Bb */
    generateTone(8.5,200);
    if(checkButton()==1){
        break;
    }
    /* Bb */
    generateTone(8.5,110);
    if(checkButton()==1){
        break;
    }
    /* G */
    generateTone(10,160);
    if(checkButton()==1){
        break;
    }
    /* G */
    generateTone(10,90);
    if(checkButton()==1){
        break;
    }
    /* F */
    generateTone(11.5,160);
    if(checkButton()==1){
        break;
    }
    _delay_ms(500);
}
}

```

```

void Levels(){
    while(1){
        /* C# */
        generateTone(7,100);
        if(checkButton()==1){
            break;
        }
        /* B */
        generateTone(8,100);
        if(checkButton()==1){
            break;
        }
        /* G# */
        generateTone(9.5,100);
        if(checkButton()==1){
            break;
        }
        /* F# */
        generateTone(10.7,100);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,75);
        if(checkButton()==1){

```

```

        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    _delay_ms(1500);
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* D# */
    generateTone(13,85);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    /* E */
    generateTone(12,75);
    if(checkButton()==1){
        break;
    }
    _delay_ms(2000);
    /* C# */
    generateTone(7,100);
    if(checkButton()==1){
        break;
    }
    /* B */
    generateTone(8,100);
    if(checkButton()==1){
        break;
    }
    /* G# */
    generateTone(9.5,100);
    if(checkButton()==1){
        break;
    }
    /* F# */
    generateTone(10.7,100);

```



```

        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,75);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,75);
        if(checkButton()==1){
            break;
        }
        _delay_ms(1500);
        /* E */
        generateTone(12,75);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,75);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,75);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,75);
        if(checkButton()==1){
            break;
        }
        /* C# */
        generateTone(14.1,75);
        if(checkButton()==1){
            break;
        }
        /* C# */
        generateTone(14.1,75);
        if(checkButton()==1){
            break;
        }
        /* B */
        generateTone(16.4,60);
        if(checkButton()==1){
            break;
        }
        /* B */
        generateTone(16.4,60);
        if(checkButton()==1){
            break;
        }
        _delay_ms(1500);
    }
}

void OneIntro(){
    int a = 1;
    while(a == 1){
        for(int i=0;i<4;i++){
            generateTone(70,50);
            if(checkButton()==1){
                break;
            }
        }
    }
}

```

```

}
for(int i=0;i<4;i++){
    generateTone(70,25);
    if(checkButton()==1){
        break;
    }
}
for(int i=0;i<8;i++){
    generateTone(70,12.5);
    if(checkButton()==1){
        break;
    }
}
for(int i=0;i<16;i++){
    generateTone(70,5);
    if(checkButton()==1){
        break;
    }
}
for(int i=0;i<32;i++){
    generateTone(70,3);
    if(checkButton()==1){
        break;
    }
}
for(int i=0;i<32;i++){
    generateTone(70,2);
    if(checkButton()==1){
        break;
    }
}
for(int i=0;i<64;i++){
    generateTone(70,1);
    if(checkButton()==1){
        break;
    }
}
for(int i=70;i>60;i--){
    generateTone(i,1);
    if(checkButton()==1){
        break;
    }
}
for(int i=60;i>50;i--){
    generateTone(i,2);
    if(checkButton()==1){
        break;
    }
}
for(int i=50;i>40;i--){
    generateTone(i,3);
    if(checkButton()==1){
        break;
    }
}
for(int i=40;i>30;i--){
    generateTone(i,4);
    if(checkButton()==1){
        break;
    }
}
for(int i=30;i>20;i--){
    generateTone(i,5);
    if(checkButton()==1){
        break;
    }
}

```

```

    }
    for(int i=20;i>4;i--){
        generateTone(i,6);
        if(checkButton()==1){
            break;
        }
    }
    generateTone(9,900);
    if(checkButton()==1){
        break;
    }
    _delay_ms(500);
    a = 0;
}
}

void One(){
    OneIntro();
    while(1){
        /* A */
        generateTone(9,150);
        if(checkButton()==1){
            break;
        }
        /* G */
        generateTone(10,150);
        if(checkButton()==1){
            break;
        }
        /* F */
        generateTone(11.5,150);
        if(checkButton()==1){
            break;
        }
        /* E */
        generateTone(12,130);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(13.8,100);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(13.8,90);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(13.8,90);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(13.8,90);
        if(checkButton()==1){
            break;
        }
        /* D */
        generateTone(13.8,90);
        if(checkButton()==1){
            break;
        }
        _delay_ms(600);
        /* D */
    }
}

```



```

    }
}
while(1){
    DDRD = 0b011110100;
    PORTD = 0x00;
    switch(melodyCounter){
        case 1 : while(checkButton()==0){}; break;
        case 2 : Glassbilen();_delay_ms(400); break;
        case 3 : Twinkle();_delay_ms(400); break;
        case 4 : Levels(); _delay_ms(400); break; break;
        case 5 : One();_delay_ms(400);melodyCounter=1;break;
    }
}
}

```