

```
/*
 * dancing3.c
 *
 * Created: 2016-05-19 10:53:20
 * Author: digpi10
 */

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

int value; //värde från AD-omvandlaren
int limit = 0b0101000000; //Gräns för ingående värde från mikrofonen.
int countBeats; //Räknar hur länge roboten ska lyssna.
double countTime; //Gör så roboten reagerar på insignaler mer sällan.
int flag = 0; //Varierar med slagen när roboten lyssnar.
int tid = 0; //Används för att räkna ut takten.
int driveCounter=0; //Håller koll på hur länge roboten har rört sig.
int beat = 0; //Håller koll på vilket slag vi är på.
double slag; //Antal tidsavbrott per taktslag.

/*Sätter de initiala inställningarna*/
void setup() {
    DDRA = 0b00000000; //Port A - Alla pins som input
    DDRB = 0b00001111; //Port B - Pin 0-2 som output (Lysdioderna)
    DDRD = 0b00111111; //Port D - Pin 0-5 som output (Motor)

    TCCR0 = 0b00000010; //intern klocka med frekvens 2 mHz/8
    TCNT0 = 150; //initieringsvärdet, där timern börjar räkna ifrån
    TIMSK = 0b00000001; //starta klockan

    countBeats = 0;
    countTime = 0;
}

/*Metoder för att styra lampor och motorer*/
void forward() {
    PORTD = 0b00011011;
}

void backward() {
    PORTD = 0b00101101;
}

void turnRightFor() {
    PORTD = 0b00000011;
}

void turnLeftFor() {
    PORTD = 0b00011000;
}
```

```
void turnRightBack() {
    PORTD = 0b00000101;
}

void turnLeftBack() {
    PORTD = 0b00101000;
}

void spinRight() {
    PORTD = 0b00101011;
}

void spinLeft() {
    PORTD = 0b00011101;
}

void stop() {
    PORTD = 0b00000000;
}

void redLight() {
    PORTB = 0b00000001;
}

void yellowLight() {
    PORTB = 0b00000010;
}

void greenLight() {
    PORTB = 0b00000100;
}

void orangeLight() {
    PORTB = 0b00001000;
}

void allLights() {
    PORTB = 0b00001111;
}

void outerLights() {
    PORTB = 0b00001001;
}

void innerLights() {
    PORTB = 0b00000110;
}

void lightsOff() {
    PORTB = 0b00000000;
}

/*Tidsavbrott*/
ISR (TIMER0_OVF_vect) {

    TCNT0 = 150; //Sätter timerns startvärde till samma som i setup.
```

```

TIFR = 0b00000001; //Möjliggör nytt avbrott

ADMUX = 0b01000000; //AVCC med extern kondensator på AREF pin, ingen vänsterjustering,
Port ADC0 med 1x förstärkning.

SFIOR = 0b00000000; //Inga specialfunktioner aktiverade
ADCSRA = 0b11011100; //Startar A/D-omvandling

countTime++;
tid++;
driveCounter++;
}

/*A/D-avbrott */
ISR (ADC_vect) {
    value = ADC; //sparar värdet från mikrofonen

    ADMUX = 0b01000000; //AVCC med extern kondensator på AREF pin, ingen
vänsterjustering, Port ADC0 med 1x förstärkning.

    ADCSRA = 0b10011100; //nollställer flaggan och möjliggör ny A/D-omvandling
}

/* Körmönster för mellantakt (60 bpm - 120 bpm), åtta taktslag. */
void driveAround(double slag){
    lightsOff();
    driveCounter=0;

    beat = 0; //De första två slagen
    while(driveCounter < (beat+2)*slag){
        if(driveCounter < (beat+0.5)*slag) {
            outerLights(); //yttre lamporna blinkar
        }
        else if(driveCounter < (beat+1)*slag) {
            lightsOff();
        }
        else if(driveCounter < (beat+1.5)*slag) {
            innerLights(); //inre lamporna blinkar
        }
        else if(driveCounter < (beat+2)*slag) {
            lightsOff();
        }

        turnLeftFor(); //roboten svänger vänster framåt
        _delay_ms(5);
        stop();
        _delay_ms(2);
    }

    beat += 2; //De andra två slagen
    while(driveCounter < (beat+2)*slag){
        if(driveCounter < (beat+0.5)*slag) {
            outerLights(); //yttre lamporna blinkar
        }
        else if(driveCounter < (beat+1)*slag) {
            lightsOff();
        }
    }
}

```

```
    else if(driveCounter < (beat+1.5)*slag) {
        innerLights();        //inre lamporna blinkar
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
    }

    turnRightBack();        //roboten svänger höger bakåt
    _delay_ms(5);
    stop();
    _delay_ms(2);
}
```

```
beat += 2; //De tredje två slagen
while(driveCounter < (beat+2)*slag){
    if(driveCounter < (beat+0.5)*slag) {
        outerLights();        //yttre lamporna blinkar
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        innerLights();        //inre lamporna blinkar
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
    }

    turnRightFor();        //svänger höger framåt
    _delay_ms(5);
    stop();
    _delay_ms(2);
}
```

```
beat += 2; //De fjärde två slagen
while(driveCounter < (beat+2)*slag){
    if(driveCounter < (beat+0.5)*slag) {
        outerLights();        //yttre lamporna blinkar
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        innerLights();        //inre lamporna blinkar
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
    }

    turnLeftBack();        //svänger vänster bakåt
    _delay_ms(5);
    stop();
    _delay_ms(2);
}
}
```

```
/* Körmönster för långsam takt (<60 bpm), åtta taktslag. */
void driveAroundSlow(double slag){
    lightsOff();
    driveCounter=0;

    beat = 0;    //De första två slagen
    while(driveCounter < (beat+2)*slag){
        if(driveCounter < (beat+0.5)*slag) {
            orangeLight();    //orange lampa blinkar, roboten kör fram och tillbaka
            forward();
        }
        else if(driveCounter < (beat+1)*slag) {
            lightsOff();
            backward();
        }
        else if(driveCounter < (beat+1.5)*slag) {
            redLight();    //röd lampa blinkar, roboten kör fram och tillbaka
            forward();
        }
        else if(driveCounter < (beat+2)*slag) {
            lightsOff();
            backward();
        }

        _delay_ms(5);
        stop();
        _delay_ms(10);
    }

    beat += 2;    //De andra två slagen
    while(driveCounter < (beat+2)*slag){
        if(driveCounter < (beat+0.5)*slag) {
            orangeLight();    //orange lampa blinkar, roboten kör fram och tillbaka
            forward();
        }
        else if(driveCounter < (beat+1)*slag) {
            lightsOff();
            backward();
        }
        else if(driveCounter < (beat+1.5)*slag) {
            redLight();    //röd lampa blinkar, roboten kör fram och tillbaka
            forward();
        }
        else if(driveCounter < (beat+2)*slag) {
            lightsOff();
            backward();
        }

        _delay_ms(5);
        stop();
        _delay_ms(10);
    }

    beat += 2;    //De tredje två slagen
    while(driveCounter < (beat+2)*slag){
        if(driveCounter < (beat+0.5)*slag) {
            orangeLight();    //orange lampa blinkar, roboten kör fram och tillbaka
```

```

        forward();
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
        backward();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        redLight();    //röd lampa blinkar, roboten kör fram och tillbaka
        forward();
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
        backward();
    }

    _delay_ms(5);
    stop();
    _delay_ms(10);
}

beat += 2;    //De fjärde två slagen
while(driveCounter < (beat+2)*slag){
    if(driveCounter < (beat+0.5)*slag) {
        orangeLight();    //orange lampa blinkar, roboten kör fram och tillbaka
        forward();
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
        backward();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        redLight();    //röd lampa blinkar, roboten kör fram och tillbaka
        forward();
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
        backward();
    }

    _delay_ms(5);
    stop();
    _delay_ms(10);
}
}

/* Körmönster för snabb takt (> 120 bpm), åtta taktslag. */
void driveAroundFast(double slag){
    lightsOff();
    driveCounter = 0;

    beat = 0;    //De första två slagen
    while(driveCounter < (beat+2)*slag){
        if(driveCounter < (beat+0.5)*slag) {
            allLights();    //alla lampor blinkar
        }
        else if(driveCounter < (beat+1)*slag) {
            lightsOff();

```

```
}
else if(driveCounter < (beat+1.5)*slag) {
    allLights();           //alla lampor blinkar
}
else if(driveCounter < (beat+2)*slag) {
    lightsOff();
}

spinLeft();           //roboten snurrar åt vänster
_delay_ms(5);
stop();
_delay_ms(2);
}

beat += 2; //De andra två slagen
while(driveCounter < (beat+2)*slag){
    if(driveCounter < (beat+0.5)*slag) {
        allLights();           //alla lampor blinkar
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        allLights();           //alla lampor blinkar
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
    }

    spinLeft();           //roboten snurrar åt vänster
    _delay_ms(5);
    stop();
    _delay_ms(2);
}

beat += 2; //De tredje två slagen
while(driveCounter < (beat+2)*slag){
    if(driveCounter < (beat+0.5)*slag) {
        allLights();           //alla lampor blinkar
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        allLights();           //alla lampor blinkar
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
    }

    spinRight();           //roboten snurrar åt höger
    _delay_ms(5);
    stop();
    _delay_ms(2);
}
```

```

beat += 2; //De fjärde två slagen
while(driveCounter < (beat+2)*slag){
    if(driveCounter < (beat+0.5)*slag) {
        allLights(); //alla lampor blinkar
    }
    else if(driveCounter < (beat+1)*slag) {
        lightsOff();
    }
    else if(driveCounter < (beat+1.5)*slag) {
        allLights(); //alla lampor blinkar
    }
    else if(driveCounter < (beat+2)*slag) {
        lightsOff();
    }

    spinRight(); //roboten snurrar åt höger
    _delay_ms(5);
    stop();
    _delay_ms(2);
}

}

int main(void)
{
    setup();
    sei(); //möjliggör avbrott

    while(1)
    {
        PORTD = 0b00000000;

        if(value <= limit && countTime >= 50) { //då värdet lägre än limit och det gått
        minst 50 tidsenheter

            countBeats++;
            if (countBeats <= 8){ //takten räknas under åtta slag, lamporna lyser efter
            varande för varje slag

                if(flag == 0) {
                    redLight();
                }
                else if(flag == 1) {
                    yellowLight();
                }
                else if(flag == 2) {
                    greenLight();
                }
                else if(flag == 3) {
                    orangeLight();
                }
            }

            if(flag == 0) {

```

```
        flag = 1;
    }
    else if(flag == 1) {
        flag = 2;
    }
    else if(flag == 2) {
        flag = 3;
    }
    else if(flag == 3) {
        flag = 0;
    }
    countTime = 0;
```

```
}
```

```
//efter åtta takslag räknas takten ut
```

```
if(countBeats == 8) {
    slag = ((double)tid)/countBeats; //genomsnittligt antal tidsavbrott per slag
}
```

```
//beroende på vilken takt som spelas anropas olika körmönster
```

```
if(countBeats == 9){
    int limitLow=2604*0.9; //justerad frekvens
    int limitHigh=1302*0.8+50; //justerad frekvens

    if(slag > limitLow){
        driveAroundSlow(slag);
    }
    else if(limitHigh < slag && slag<= limitLow){
        driveAround(slag);
    }
    else{
        driveAroundFast(slag);
    }
}
```

```
countTime = 0;
countBeats=0;
tid=0;
```

```
}
```

```
}
```

```
}
```

```
}
```