

```

#define F_CPU 4000000UL

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

volatile int V_START = 50;

#define V_END 85
#define V_OFF 50
#define T 18

#define PLS_RATIO 85

#define SOUND_PATTERN 5

volatile unsigned long cnt = 0;
volatile int d = 0;
volatile unsigned char up = 1;
volatile unsigned char startSpin = 0;

volatile int cnt100mS = 0;

volatile unsigned char turnedOffWhileOperating = 0;

ISR(TIMER2_OVF_vect){
    if(cnt100mS >= 122){    //Executed once in 1000mS
        if(up){
            d = V_START + (V_END - V_START) * cnt / T;
            cnt++;
            if(cnt > T){
                cnt = T;
            }
        }
        else if(cnt > 0){
            d = V_OFF + (V_END - V_OFF) * cnt / T;
            cnt--;
        }
        cnt100mS = 0;
    }
    cnt100mS++;
}

ISR(PCINT2_vect){
    //Do nothing
}

int AD_read(unsigned char pin){
    int r;

    while((ADCSRA & (1 << ADSC)) != 0){}

    ADMUX = (0x40 | (pin & 0x0F));
    ADCSRA |= (1 << ADSC);

    while((ADCSRA & (1 << ADSC)) != 0){}

    r = ADC;

    return r;
}

```

```
}
```

```
void my_delay_us(unsigned int us){
```

```
    cli();
    TCNT1 = 0;

    TIFR1 |= (1 << OCF1A);
    us >>= 1;
    OCR1A = us;
    sei();
    while((TIFR1 & (1 << OCF1A)) == 0){}
    TIFR1 |= (1 << OCF1A);
```

```
}
```

```
void mytone(int f, unsigned long t, int duty){
```

```
    unsigned long i, n;
    unsigned long th, tl;
    unsigned long tlon, tloff, thon, thoff;
    th = (1000000UL / 100) * duty / f;
    tl = (1000000UL / 100) * (100 - duty) / f;

    n = (unsigned int)(t / (tl + th));
    // PB0 = D8, PB1 = D9

    thon = (unsigned int)(th * PLS_RATIO / 100);
    thoff = (unsigned int)(th * (100 - PLS_RATIO) / 100);
    tlon = (unsigned int)(tl * PLS_RATIO / 100);
    tloff = (unsigned int)(tl * (100 - PLS_RATIO) / 100);

    for(i = 0; i < n; i++){
        PORTC = 0x05;
        my_delay_us(thon);
        PORTC = 0x01;
        my_delay_us(thoff);
        PORTC = 0x09;
        my_delay_us(tlon);
        PORTC = 0x01;
        my_delay_us(tloff);
    }
```

```
}
```

```
void changeTone(int f1, int f2, unsigned long t){
```

```
    long i, n, r, f;
    n = t / 20;
    for(i = 0; i < n; i++){
        f = f1 + (f2 - f1) * i / n;

        if(!turnedOffWhileOperating && (PIND & (1 << 6)) != 0){
            _delay_ms(20);
            if((PIND & (1 << 6)) != 0){
                turnedOffWhileOperating = 1;
            }
        }

        if(turnedOffWhileOperating && (PIND & (1 << 6)) == 0){
            break;
        }

        mytone((int)f, 20000, d);

        if(startSpin && i % 4 == 0){
            PORTC = 0x01;
        }
    }
```

```

        _delay_ms(1);
        if(AD_read(4) > 5 || d > 90){
            V_START = d;

            TIMSK2 |= (1 << TOIE2);
            startSpin = 0;
        }
        else {
            d += 1;
        }
    }
}

void freeRun(int f){
    unsigned long th, tl;
    th = 700000UL / f;
    tl = 300000UL / f;
    while((PIND & (1 << 6)) == 0 && !turnedOffWhileOperating){
        PORTC = 0x05;
        my_delay_us(th);
        PORTC = 0x01;
        my_delay_us(tl);
    }
}

void speedUp(){
    cnt = 0;
    up = 1;
    startSpin = 1;
    TIMSK2 &= ~(1 << TOIE2);
    d = 50;
    turnedOffWhileOperating = 0;
}

void speedDown(){
    cnt = T;
    up = 0;
}

void seibu6000(){
    speedUp();
    changeTone(784, 784, 3720);
    changeTone(659, 988, 2270);
    changeTone(587, 784, 1920);
    changeTone(277, 415, 10070);
    freeRun(415);
    changeTone(622, 415, 9150);
    speedDown();
    changeTone(659, 587, 1750);
    changeTone(988, 831, 2970);
    changeTone(1319, 698, 6020);
    changeTone(988, 311, 5640);
}

void e231_1000(){
    speedUp();
    changeTone(1046, 1046, 4500);
    changeTone(1046, 784, 9400);
    changeTone(784, 1865, 3420);
    changeTone(659, 740, 2300);
    changeTone(340, 466, 11370);
    freeRun(466);
    changeTone(466, 340, 10000);
    changeTone(740, 659, 2300);
}

```

```

        speedDown();
        changeTone(1865, 784, 3420);
        changeTone(784, 1046, 8400);
        changeTone(1046, 1046, 4500);
    }

    void vvvf373(){
        speedUp();
        changeTone(740, 740, 3000);
        changeTone(392, 932, 6270);
        changeTone(554, 831, 5000);
        changeTone(246, 370, 7320);
        freeRun(370);
        changeTone(370, 247, 10150);
        speedDown();
        changeTone(831, 523, 6840);
        changeTone(880, 294, 8590);
    }

    void keikyu_new1000(){
        speedUp();
        changeTone(311, 311, 200);
        changeTone(349, 349, 220);
        changeTone(392, 392, 220);
        changeTone(440, 440, 220);
        changeTone(466, 466, 220);
        changeTone(523, 523, 220);
        changeTone(587, 587, 220);
        changeTone(622, 622, 220);
        changeTone(698, 698, 220);
        changeTone(784, 784, 4700);
        changeTone(784, 988, 550);
        changeTone(784, 932, 890);
        changeTone(740, 880, 1200);
        //changeTone(622, 740, 1750);
        //changeTone(494, 554, 2100);
        changeTone(247, 466, 10500);

        freeRun(466);
        speedDown();
        changeTone(1109, 831, 5550);
        changeTone(1047, 698, 4670);
        changeTone(880, 740, 1150);
        changeTone(932, 784, 770);
        changeTone(784, 784, 3840);
    }

    void vvvf209(){
        speedUp();
        changeTone(233, 233, 1250);

        changeTone(293, 932, 1500);
        changeTone(554, 1245, 2100);
        changeTone(740, 1175, 2200);
        changeTone(370, 587, 3400);
        changeTone(220, 294, 2700);
        changeTone(232, 492, 14000);

        freeRun(492);
        changeTone(246, 138, 9600);
        speedDown();
        changeTone(466, 415, 2500);
    }

```

```

    changeTone(740, 523, 4700);
    changeTone(987, 659, 3200);
    changeTone(1244, 784, 2500);
    changeTone(1244, 440, 2500);

}

int main(void){

    int r, i;

    DDRB = 0xFF;
    DDRC = 0x4F;
    DDRD = 0xFF;

    PORTB = 0xFF;    //All pulled up (Don't use)
    PORTC = 0;
    PORTD = 0xFF;    //All pulled up (PD6: Toggle switch)

    //Timer1 setup for delay count

    TCCR1A = 0;
    TCCR1B = 0x0A;
    OCR1A = 200;

    //Timer2 setup for overflow interrupt

    TCCR2A = 0;      //Timer 2 ON, Normal operation, No output
    TCCR2B = 0x05;    //Prescaler x128

    //Enable pin-change interrupt on PD6/PCINT22

    PCICR = (1 << PCIE2);
    PCMSK2 = (1 << PCINT22);

    ADCSRA = 0x85;

    sei();

    while(1){

        while(1){

            if((PIND & (1 << 6)) != 0){

                PORTC = 0;
                set_sleep_mode(SLEEP_MODE_PWR_DOWN);

                ADCSRA &= ~(1 << ADEN);
                sleep_enable();
                sleep_cpu();

                ADCSRA |= (1 << ADEN);

            }

            PORTC = 0x01;
            _delay_ms(100);

            if((PIND & (1 << 6)) == 0){
                break;
            }

        }

        _delay_ms(1000);
    }
}

```

```

r = 0;
for(i = 0; i < 4; i++){
    r += AD_read(5);
}

r = r * SOUND_PATTERN / 4 / 1024;

switch(r){
    case 0:
        vvvf209();
        break;
    case 1:
        keikyu_new1000();
        break;
    case 2:
        vvvf373();
        break;
    case 3:
        e231_1000();
        break;
    case 4:
        seibu6000();
        break;
}

_delay_ms(1000);
}
}

```