

```
#include <avr\io.h>
#include <avr\interrupt.h>
#define C4 3822 // delay count for creating 261.63-Hz tone
#define C4S 3608 // delay count for creating 277.18-Hz tone
#define D4F 3608 // 277.18 Hz
#define D4 3405 // 293.66 Hz
#define D4S 3214 // 311.13 Hz
#define E4F 3214 // 311.13 Hz
#define E4 3034 // 329.63 Hz
#define F4 2863 // 349.23 Hz
#define F4S 2703 // 369.99 Hz
#define G4F 2703 // 369.99 Hz
#define G4 2551 // 392.00 Hz
#define G4S 2408 // 415.30 Hz
#define A4F 2408 // 415.30 Hz
#define A4 2273 // 440.00 Hz
#define A4S 2145 // 466.16 Hz
#define B4F 2145 // 466.16 Hz
#define B4 2025 // 493.88 Hz
#define C5 1911 // 523.25 Hz
#define C5S 1804 // 554.37 Hz
#define D5F 1804 // 554.37 Hz
#define D5 1703 // 587.33 Hz
#define D5S 1607 // 622.25 Hz
#define E5F 1607 // 622.25 Hz
#define E5 1517 // 659.26 Hz
#define F5 1432 // 698.46 Hz
#define F5S 1351 // 739.99 Hz
#define G5F 1351 // 739.99 Hz
#define G5 1276 // 783.99 Hz
#define G5S 1204 // 830.61 Hz
#define A5F 1204 // 830.61 Hz
#define A5 1136 // 880.00 Hz
#define A5S 1073 // 932.33 Hz
#define B5F 1073 // 932.33 Hz
#define B5 1012 // 987.77 Hz
#define ZZ 20 // 50000 Hz ( inaudible sound)
unsigned int score[] = {
    F4, G4, A4, B4F, ZZ, B4F, C5, ZZ, C5, A4, ZZ, A4, C5, B4F, A4, B4F, G4, A4, F4, G4,
    A4, B4F, ZZ, B4F, C5, ZZ, C5, A4, C5, B4F, A4, B4F, G4, F4, ZZ, C5, ZZ, C5, F5, E5, D5, C5,
    C5, A4, C5, B4F, A4, B4F, G4, A4, C5, ZZ, C5, F5, E5, D5, C5, ZZ, C5, A4, C5,
    C5, B4F, G4, F4, ZZ, C5, B4F, G4, F4, ZZ, G4, ZZ,
    A4, ZZ, C5, F5, E5, D5, C5, ZZ, C5, A4, C5, B4F, A4, B4F, G4, F4, 0};
unsigned char dur[] = {
    30, 10, 60, 20, 3, 60, 20, 3, 60, 20, 3, 40, 40, 60, 20, 40, 40, 120, 30, 10,
    60, 20, 3, 60, 20, 3, 80, 40, 40, 60, 20, 40, 40, 80, 40, 20, 3, 20, 60, 20, 60, 20,
    80, 40, 40, 60, 20, 40, 40, 120, 20, 3, 20, 60, 20, 60, 20, 3, 80, 40, 40,
    40, 80, 40, 120, 40, 160, 80, 80, 40, 40, 40, 40,
    80, 40, 40, 60, 20, 60, 20, 3, 80, 40, 40, 60, 20, 40, 40, 120};
unsigned int delay;
void delayby10msOC(unsigned char k);
void main (void)
{
    unsigned char j;
    DDRB |= 0x20; // configure OC1A/PB5 for output
    delay = score[0];
    TCCR1A = 0x40; // configure OC1A to normal mode, set compare match action to toggle
    TCCR1B = 0x02; // use clk_I/0 / 8 as Timer 1 clock source
    OCR1A = TCNT1 + delay; // start the first compare operation
    TIMSK1 = 0x02; // enable OC1A compare match interrupt
```

```
TIFR1 = OCF1A; // clear OCF1A flag
sei(); // enable interrupt globally
j = 0;
while(score[j]){
    delay = score[j];
    delayby10msOC(dur[j]);
    j++;
}
TCCR1B = 0; // stop Timer 1
while(1);
}
// -----
// OC1A interrupt service routine. It starts the next compare operation and clears the OCF1A flag.
// -----
ISR(TIMER1_COMPA_vect)
{
    OCR1A += delay; // start a new output compare operation
    TIFR1 = OCF1A; // clear OCF1A flag
}
// -----
// The following function creates a time delay that is a multiple of 10 ms using Timer 3
// OC3A function.
// -----
void delayby10msOC(unsigned char k)
{
    TCCR3A = 0; // configure Timer 3 to normal mode
    TCCR3B = 0x02; // and select clk_I/0 / 8 as clock input
    OCR3A = TCNT3 + 20000; // start an output compare operation
    TIFR3 = OCF3A; // clear OCF3A flag
    while(k) {
        while(!(TIFR3 & OCF3A)); // wait for OCF3A flag to set
        OCR3A += 20000; // start a new OC3 compare operation
        TIFR3 = OCF3A; // clear OCF3A flag
        k--;
    }
}
```