

## Пьезоэлектрический излучатель. Мелодия

**А:** А кем ты работаешь?

**Б:** Я — программист!

**А:** О, классно, а что ты программируешь?

**Б:** Акустические волны.

Кроме генерации однотонного звука, можно воспроизвести даже целые мелодии: для этого потребуется поочередно задавать ту или иную частоту колебаний. Путь это будет русская народная песня «Коробейники».



ТенорА XXI века

Русская народная песня "Коробейники" - Александр Захаров

Share



4:04

[Cookie policy](#)

Ниже представлена табулатура Коробейников для укулеле.

```
A4 | | ----- | |
E4 | | ----- | |
C4 | | -----7-8-10-8-7-5-5-8---10-8-7-7-10---8-5-5--- | |
G4 | | ---7-----7-----7----- | |

A4 | | ----- | |
E4 | | ----- | |
C4 | | -----8---10-8-7-7-10---8-5-5--- | |
G4 | | ---5-8-12-10-8-7---7-----7----- | |
```

Если записать линейно, то мелодия будет выглядеть так:

D<sub>5</sub>, G<sub>4</sub>, G<sub>4</sub>#, A<sub>4</sub>#, G<sub>4</sub>#, G<sub>4</sub>, F<sub>4</sub>, F<sub>4</sub>, G<sub>4</sub>#, D<sub>5</sub>, A<sub>4</sub>#, G<sub>4</sub>#, G<sub>4</sub>, G<sub>4</sub>, A<sub>4</sub>#, D<sub>5</sub>, G<sub>4</sub>#, F<sub>4</sub>, F<sub>4</sub>,

C<sub>5</sub>, D<sub>5</sub>#, G<sub>5</sub>, F<sub>5</sub>, D<sub>5</sub>#, D<sub>5</sub>, G<sub>4</sub>#, D<sub>5</sub>, G<sub>4</sub>#, D<sub>5</sub>, A<sub>4</sub>#, G<sub>4</sub>#, G<sub>4</sub>, G<sub>4</sub>, A<sub>4</sub>#, D<sub>5</sub>, G<sub>4</sub>#, F<sub>4</sub>, F<sub>4</sub>

В табулатуре используется первая и вторая октавы, однако наш динамик плохо воспроизводит звуки до 1 кГц. Вам необходимо транспонировать его на более высокие октавы (например, на 3 и 4 октавы, или же на 4 и 5 октавы). Нужные частоты вы можете [найти в сети](#), мы же приведём частоты 3, 4 и 5 октав:

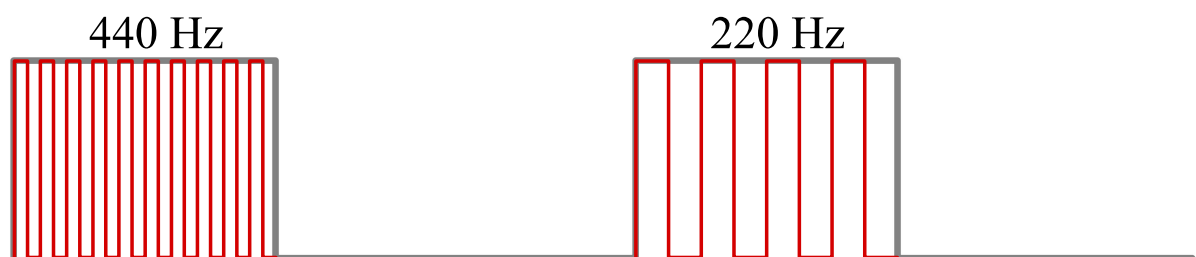
Нота	3 октава, Гц	4 октава, Гц	5 октава, Гц
C	1046,5	2093	4186
C#	1108,7	2217,4	4434,8
D	1174,6	2349,2	4698,4
D#	1244,5	2489	4978
E	1318,5	2637	5274
F	1396,9	2793,8	5587,6
F#	1480	2960	5920
G	1568	3136	6272
G#	1661,2	3322,4	6644,8
A	1760	3520	7040
A#	1864,6	3729,2	7458,4
H	1975,5	3951	7902

Поочередно меняя тон и вставляя задержки, мы получим мелодию. Добавим макросы с говорящими именами для каждого тона из таблицы.

```
// buzzer.c
#define FREQ_C_3      1046
#define FREQ_C_4      2093
#define FREQ_C_5      4186
#define FREQ_C_SHARP_3 1108
#define FREQ_C_SHARP_4 2217
#define FREQ_C_SHARP_5 4434
#define FREQ_D_3      1174
#define FREQ_D_4      2349
#define FREQ_D_5      4698
#define FREQ_D_SHARP_3 1244
#define FREQ_D_SHARP_4 2489
#define FREQ_D_SHARP_5 4978
#define FREQ_E_3      1318
#define FREQ_E_4      2637
#define FREQ_E_5      5274
#define FREQ_F_3      1396
#define FREQ_F_4      2793
#define FREQ_F_5      5587
#define FREQ_F_SHARP_3 1480
#define FREQ_F_SHARP_4 2960
#define FREQ_F_SHARP_5 5920
#define FREQ_G_3      1568
#define FREQ_G_4      3136
#define FREQ_G_5      6272
#define FREQ_G_SHARP_3 1661
```

```
#define FREQ_G_SHARP_4 3322
#define FREQ_G_SHARP_5 6644
#define FREQ_A_3 1760
#define FREQ_A_4 3520
#define FREQ_A_5 7040
#define FREQ_A_SHARP_3 1864
#define FREQ_A_SHARP_4 3729
#define FREQ_A_SHARP_5 7458
#define FREQ_H_3 1975
#define FREQ_H_4 3951
#define FREQ_H_5 7902
```

Музыка это не только поочерёдная смена тона, а ещё и паузы между нотами — это называется **ритмом**. С технической точки зрения нужно сделать ШИМ внутри другого ШИМ, причём параметры обоих могут меняться во времени.



Таким образом, одна единица мелодии состоит из трёх частей: частота звука, продолжительность ноты, продолжительность паузы. Создадим для этого структуру.

```
typedef struct {
    uint8_t note_duration; // cs (10 * ms = 0.01 s)
    uint8_t pause_duration; // cs
    uint16_t note_freq; // hz
} TONE_t;
```

Здесь по возможности нужно обойтись минимальным размером экземпляра структуры. Давайте рассуждать: частота точно не будет превышать 10 кГц, поэтому 16 бит для ноты вполне достаточно; если мы хотим уложиться в 32 бита, то на продолжительность паузы и воспроизведения нужно оставить по 8 бит; продолжительность игры и паузы вряд ли будет больше 1 секунды, но даже при этом, если мы будем хранить значения в сантисекундах ( $10^{-2}$ ), то переменной хватит аж на 2,55 с.

Создадим массив таких структур для мелодии «Коробейники».

```
static const TONE_t peddler[] = {
    { 10, 1, FREQ_D_4, },
    { 10, 1, FREQ_G_4, },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_A_SHARP_4, },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_G_4, },
    { 10, 1, FREQ_F_4, },
    { 10, 1, FREQ_F_4, },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_D_5, },
    { 10, 1, FREQ_A_SHARP_4, },
```

```

    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_G_4,      },
    { 10, 1, FREQ_G_4,      },
    { 10, 1, FREQ_A_SHARP_4, },
    { 10, 1, FREQ_D_5,      },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_F_4,      },
    { 10, 1, FREQ_F_4,      },
    { 10, 1, FREQ_C_5,      },
    { 10, 1, FREQ_D_SHARP_5, },
    { 10, 1, FREQ_G_5,      },
    { 10, 1, FREQ_F_5,      },
    { 10, 1, FREQ_D_SHARP_5, },
    { 10, 1, FREQ_D_5,      },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_D_5,      },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_D_5,      },
    { 10, 1, FREQ_A_SHARP_4, },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_G_4,      },
    { 10, 1, FREQ_G_4,      },
    { 10, 1, FREQ_A_SHARP_4, },
    { 10, 1, FREQ_D_5,      },
    { 10, 1, FREQ_G_SHARP_4, },
    { 10, 1, FREQ_F_4,      },
    { 10, 1, FREQ_F_4,      },
};

```

Осталось мелодию воспроизвести. Добавим функцию для воспроизведения.

```

// buzzer.h
#include "utils.h"

void buzzer_play(void);

// buzzer.c
#define SILENCE()      TIM2->ARR = 0; \
                        TIM2->CCR3 = 0

void buzzer_play(void) {
    uint32_t n = sizeof(peddler) / sizeof(peddler[0]);
    buzzer_turn_on();
    for (uint32_t i = 0; i < n; i++) {
        SET_FREQ(peddler[i].note_freq);
        delay(peddler[i].note_duration * 100000);
        SILENCE();
        delay(peddler[i].pause_duration * 100000);
    }
    buzzer_turn_off();
}

// main.c
while(1) {

```

```
buzzer_play();  
}
```

Код доступен на GitHub: [CMSIS](#).

Реализация наивная и имеет один существенный недостаток. Что, если вам потребуется остановить воспроизведение мелодии? Что, если данную функцию захочется вызвать из прерывания? Наверняка вам в голову уже пришло одно решение, но оно... не самое лучшее — вы просто не всё знаете.

---

[Назад](#) | [Оглавление](#) | [Дальше](#)