



Московский государственный технический университет им. Н. Э. Баумана
Факультет «Информатика, искусственный интеллект и системы управления»
Кафедра «Системы обработки информации и управления»

Отчет по лабораторной работе №4

по курсу «ЭУ АСОИУ»

«Формирование мелодии с помощью микроконтроллера ATmega16»

Выполнил: Васильев Д.А.,

Студент группы ИУ5-72Б

Вариант 3.

Москва, 2022 г.

Цель работы.

Изучить основные приёмы формирования упорядоченных звуковых сигналов в виде простейшей мелодии с помощью микроконтроллера ATmega16.

Освоить методику построения на основе микроконтроллера ATmega16 систем формирования упорядоченных звуковых сигналов, работающих как в автоколебательном, так и в ждущем режимах.

Научиться создавать программы на языке ассемблера и/или C для ATmega16 с использованием подпрограмм.

1. Задание лабораторной работы – нотная запись мелодии.

3. В лесу родилась...

Flute

Ма - лень - кой ё - лоч - ке хо - лод - но зи - мой

Fl.

Из ле - су ё - лоч - ку взя - ли мы до - мой

2. Последовательность частот и длительностей нот в мелодии.

Используются 7 нот:

	1 октава						2 октава
№	1	2	3	4	5	6	7
Нота, SPN	C4	D4	E4	F4	G4	A4	C5
Нота, русский язык	До	Ре	Ми	Фа	Соль	Ля	До
Частота, Гц	261.6	293.7	329.6	349.2	392.0	440.0	523.3

Мелодия не содержит более 8 нот → дешифратор не используется.

Мелодия в ритме 2 четверти.

№ такта	Используемые ноты и их длительности
1	G4 1/4, E4 1/8, E4 1/8
2	G4 1/4, E4 1/8, E4 1/8
3	G4 1/8, F4 1/8, E4 1/8, D4 1/8
4	C4 1/2
5	A4 1/4, C5 1/8, A4 1/8
6	G4 1/4, E4 1/8, E4 1/8
7	G4 1/8, F4 1/8, E4 1/8, D4 1/8
8	C4 1/2

Для имитации «не легато» будем делать паузу между нотами, длительностью в 0.2 от длительности последней. Данное значение будет устанавливаться параметром **NON_LEGATO_DELAY**.

Также темп мелодии можно будет задать при помощи параметра **TICK_INTERVAL_MS**, который обозначает длительность целой ноты в миллисекундах.

Для паузы между воспроизведениями мелодии будем вызывать функцию **play_note** с **h0** в качестве ноты и нужной длительностью.

3. Программа для микроконтроллера ATMEGA16, реализующая заданную мелодию.

```
#include <xc.h>
#include <avr/io.h>
#include <util/delay.h>

#define TICK_INTERVAL_MS 4000
#define NON_LEGATO_DELAY 0.2

#define h0 0b00000000

#define n1 0b00000001
#define n2 0b00000010
#define n3 0b00000100
#define n4 0b00001000
#define n5 0b00010000
#define n6 0b00100000
#define n7 0b01000000

void delay(int duration);
void play_note(const int note_port, const float duration, int legato);
void play_melody();

int main(void)
{
    DDRB = 0xFF;
    PORTB = h0;
    while(1) {
        play_melody();
    }
}

void delay(int duration) {
    for (int k = 0; k < duration; ++k)
        _delay_ms(1);
}

void play_note(const int note_port, const float duration, int legato) {
    PORTB = note_port;

    if (legato) {
        delay(duration * TICK_INTERVAL_MS);
        return;
    }

    const int sound_duration = (1 - NON_LEGATO_DELAY) * duration * TICK_INTERVAL_MS;
    const int delay_duration = NON_LEGATO_DELAY * duration * TICK_INTERVAL_MS;
    delay(sound_duration);
    PORTB = h0;
    delay(delay_duration);
}

void play_melody() {
    play_note(n5, 1./4, 0);
    play_note(n3, 1./8, 0);
    play_note(n3, 1./8, 0);

    play_note(n5, 1./4, 0);
    play_note(n3, 1./8, 0);
    play_note(n3, 1./8, 0);
}
```

```
    play_note(n5, 1./8, 0);
    play_note(n4, 1./8, 0);
    play_note(n3, 1./8, 0);
    play_note(n2, 1./8, 0);

    play_note(n1, 1./2, 0);

    play_note(n6, 1./4, 0);
    play_note(n7, 1./8, 0);
    play_note(n6, 1./8, 0);

    play_note(n5, 1./4, 0);
    play_note(n3, 1./8, 0);
    play_note(n3, 1./8, 0);

    play_note(n5, 1./8, 0);
    play_note(n4, 1./8, 0);
    play_note(n3, 1./8, 0);
    play_note(n2, 1./8, 0);

    play_note(n1, 1./2, 0);

    play_note(h0, 1, 0);
}
```

4. Функционирование разработанного генератора продемонстрировано в программе PROTEUS.

