**Московский Государственный Технический Университет имени Н. Э. Баумана**

**Лабораторная работа №4**

**«Формирование мелодии с помощью микроконтроллера ATmega16»**

**Москва 2020**

**1. Генерация заданной мелодии.**

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 как автоколебательном, так и в ждущем режимах.

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

Задание для лабораторной работы – нотная запись мелодии.

**Вариант ХХХ**

****

Описание мелодии:

1) Указан скрипичный ключ. Звукоряд расположен в первой и второй октавах.

2) Знак при ключе ФА# (фа-диез). Тональность G-dur (Соль мажор)

3) Размер такта четыре четверти

4) Мелодия начинается из-за такта

5) В последнем такте есть половинная пауза

6) В 5,8,9 тактах присутствуют половинные ноты

7) Основная длительность нот - четверть

**5. Последовательность частот и длительностей нот в мелодии.**

|  |  |  |
| --- | --- | --- |
| Значение символа | Частота, Гц | Длительность, мс |
| G1 1/4 | 392 | 100\*4 |
| G1 1/4 | 392 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| B1 1/4 | 493,8 | 100\*4 |
| G1 1/4 | 392 | 100\*4 |
| G1 1/4 | 392 | 100\*4 |
| G1 1/4 | 392 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| C2 1/4 | 523,3 | 100\*4 |
| B1 1/2 | 493,8 | 100\*8 |
| D2 1/4 | 587,3 | 100\*4 |
| B1 1/4 | 493,8 | 100\*4 |
| A1 1/4 | 440 | 100\*4 |
| A1 1/4 | 440 | 100\*4 |
| A1 1/4 | 440 | 100\*4 |
| A1 1/4 | 440 | 100\*4 |
| B1 1/4 | 493,8 | 100\*4 |
| B1 1/4 | 493,8 | 100\*4 |
| D2 1/4 | 587,3 | 100\*4 |
| B1 1/4 | 493,8 | 100\*4 |
| A1 1/2 | 440 | 100\*8 |
| A1 1/4 | 440 | 100\*4 |
| A1 1/4 | 440 | 100\*4 |
| G1 1/2 | 392 | 100\*8 |
| 1/2 | 0 | 100\*8 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Нота | G1 | C2 | B1 | D2 | A1 | Пауза |
| Частота | 392 | 523,3 | 493,8 | 587,3 | 440 | 0 |
| Кодировка ключа | 0000.0001 (0x01) | 0000.0010  (0x02) | 0000.0100  (0x04) | 0000.1000  (0x08) | 0001.0000  (0x10) | 0000.0000  (0x00) |

**6. Программа для микроконтроллера ATMEGA16, реализующая заданную мелодию.**

**#include <avr/io.h>**

**#include <avr/delay.h>**

void note\_delay(int duration, int note) // Функция вывода ноты на несколько мс

{

TCNT0=0; // Обнуление таймера

PORTB = note; // Вывод ноты

int time\_ms = 100 \* duration; // Расчёт длительности ноты, длительность целой ноты 1,6 с

while(1)

{

// Если таймер достиг нужного значения, завершить выполнение

if(TCNT0>=time\_ms) break;

// Если длительность сигнала больше максимального значения счётчика

if(TCNT0==255)

{

time\_ms -= 256;

TCNT0=0;

}

}

PORTB = 0; // Вывод на порт B нуля для разделения нот

}

int main (void)

{

DDRD = 0x00; // Назначение порта D на ввод

DDRB = 0xFF; // Назначение порта B на вывод

PORTD = 0xFF;

TCCR0 = 0b101; // Деление тактовой частоты на 1024

OSCCAL = 135; // Коррекция тактовой частоты

while(1)

{

// Такт 1

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 2

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 3

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 4

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 5

note\_delay(8, 0x04); // Нота B1, длительность 1/2

note\_delay(4, 0x08); // Нота D2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 6

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 7

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x08); // Нота D2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 8

note\_delay(8, 0x10); // Нота A1, длительность 1/2

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 9

note\_delay(8, 0x01); // Нота G1, длительность 1/2

note\_delay(8, 0x00); // Пауза, длительность 1/2

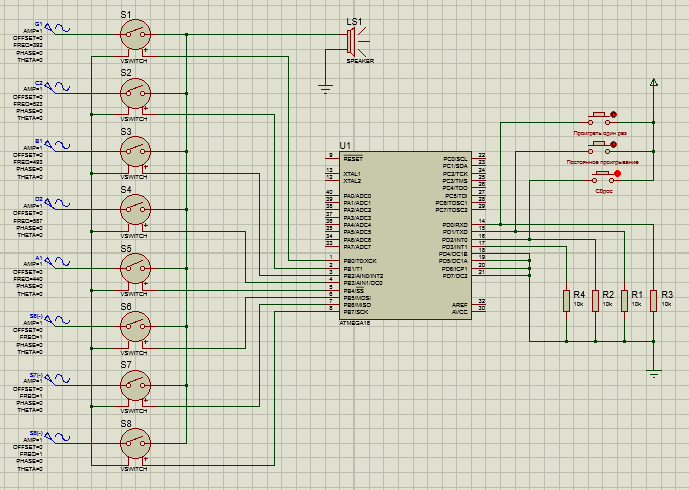
}

return 1;

}

**Дополнительное задание «Управляющие кнопки»**

**Схема:**



**Программа для микроконтроллера:**

**#include <avr/io.h>**

**#include <avr/delay.h>**

void note\_delay(int duration, int note) // Функция вывода ноты на несколько мс

{

TCNT0=0; // Обнуление таймера

PORTB = note; // Вывод ноты

int time\_ms = 100 \* duration; // Расчёт длительности ноты, длительность целой ноты 1,6 с

while(1)

{

if(TCNT0>=time\_ms) break; // Если таймер достиг нужного значения, завершить выполнение

if(TCNT0==255) // Если длительность сигнала больше максимального значения счётчика

{

time\_ms -= 256;

TCNT0=0;

}

}

PORTB = 0; // Вывод на порт B нуля для разделения нот

}

void music()

{

// Такт 1

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 2

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 3

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x01); // Нота G1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 4

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x02); // Нота C2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 5

note\_delay(8, 0x04); // Нота B1, длительность 1/2

note\_delay(4, 0x08); // Нота D2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 6

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 7

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x08); // Нота D2, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x04); // Нота B1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 8

note\_delay(8, 0x10); // Нота A1, длительность 1/2

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

note\_delay(4, 0x10); // Нота A1, длительность 1/4

note\_delay(1, 0x00); // Пауза, длительность 1/16

// Такт 9

note\_delay(8, 0x01); // Нота G1, длительность 1/2

note\_delay(8, 0x00); // Пауза, длительность 1/2

}

int main (void)

{

DDRD = 0x00; // Назначение порта D на ввод

DDRB = 0xFF; // Назначение порта B на вывод

PORTD = 0xFF;

TCCR0 = 0b101; // Деление тактовой частоты на 1024

OSCCAL = 135; // Коррекция тактовой частоты

**while(1)**

**{**

**// Кнопка «Проиграть один раз»**

**if (PIND == 0x01)**

**{**

**music();**

**}**

**// Кнопка «Постоянное проигрывание»**

**else if (PIND == 0x02)**

**{**

**while (1)**

**{**

**music();**

**// Кнопка «Сброс»**

**if (PIND == 0x04)**

**{**

**break;**

**}**

**};**

**}**

}

return 1;

}

**Комментарий:**

При нажатии кнопки «Проиграть один раз» ATmega16 проиграет мелодию один раз и остановится.

При нажатии кнопки «Постоянное проигрывание» ATmega16 начинает играть в автоколебательном режиме.

Чтобы вывести ATmega16 из этого состояния без прерывания симуляции Proteus, необходимо нажать «залипание» кнопки «Сброс». ATmega16 доиграет мелодию до конца и остановится.

Затем отключить «залипание» кнопки «Сброс», чтобы вернуть систему в исходное состояние

