

# Проект по микроконтроллерам

## Интерактивный аудиоплеер 8-битных мелодий

Работу выполнили:  
Белов Владислав, Б01-005  
Иванов Иван, Б01-004  
Фролов Даниил, Б01-005

### 1 Содержание работы

#### Содержание

1	Содержание работы	1
2	Идея проекта	2
3	Описание проекта	2
4	Использование	2
5	100 вопросов	3

## 2 Идея проекта

Проект создан в рамках курса «Микроконтроллеры», читаемого на физтех-школе радиотехники и компьютерных технологий (ФРКТ). Суть проекта — чтобы разобраться в устройстве и работе микроконтроллеров.

Идея нашего аудиоплеера базируется на проигрывании 8-битовой музыки без использования специальных для этого динамиков. Для подачи звука используется Пьезоизлучатель.

## 3 Описание проекта

Наш восьмьбитовый плеер имеет интерактивное меню, отображаемое на LCD Keypad Shield 1602. Все программы обеспечиваются через Arduino Mega 2560. Для проигрывания музыки используется Пьезоизлучатель.

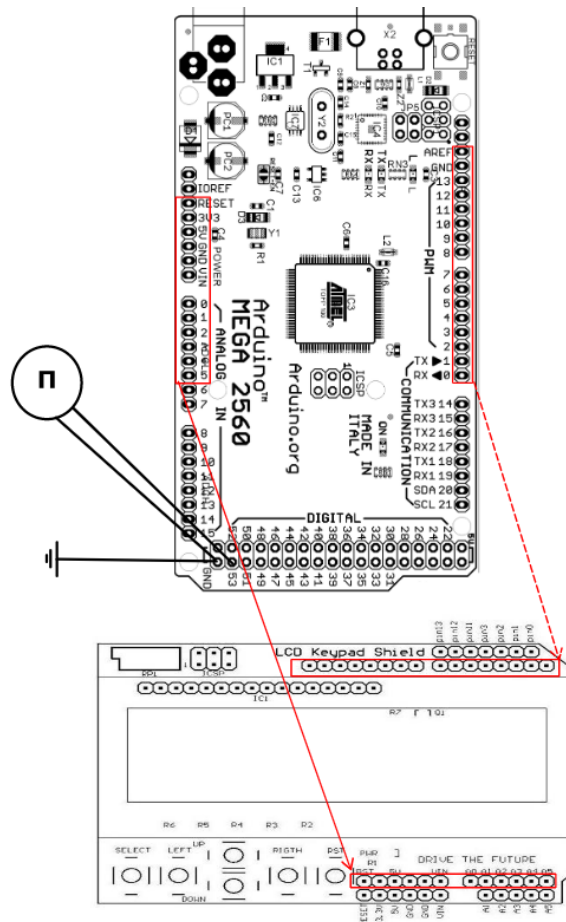


Рис. 1: Схема подключения

Код проекта можно найти по ссылке: [github.com/Exactlyub/-/tree/master/bitShataetGolovu](https://github.com/Exactlyub/-/tree/master/bitShataetGolovu)

## 4 Использование

Вы можете управлять проигрывателем, используя следующие кнопки:

- UP/DOWN — служебные кнопки, имеющие разные функции в зависимости от того, в каком режиме вы находитесь. Это может быть переключение музыки (режим выбора мелодии) или повышение оценки в режиме Rate Project
- Next/Prev — переключение музыки на следующую мелодию
- Select — паузу / выбор музыки

## 5 100 вопросов

1. Ваша фамилия, имя, отчество, номер группы.

Белов Владислав Валерьевич, Б01-005

Иванов Иван Николаевич, Б01-004

Фролов Даниил Александрович, Б01-005

2. Фамилия, имя, отчество лектора.

Донов Геннадий Иннокентьевич.

3. Чем отличается микроконтроллер от микропроцессора.

Микропроцессор – это исполняющее ядро. Микропроцессор содержит только то, что необходимо для выполнения арифметических и логических операций. Микроконтроллер имеет более сложную структуру. В микроконтроллер кроме вычислительного устройства встроены ПЗУ, ОЗУ, а также устройства ввода/вывода, АЦП. Т.е. основное отличие в том, что у микроконтроллера основные модули, необходимые для выполнения своих функций – встроенные, а микропроцессору нужно задействовать внешние устройства.

4. Какие тактовые частоты могут быть у ATmega8535.

От внутреннего генератора: 1 МГц, 2 МГц, 4 МГц. От внешнего источника: частоты в диапазоне от 0.1 МГц до 16 МГц.

5. Почему при повышении тактовой частоты микроконтроллера он начинает больше греться?

Полевые транзисторы имеют емкость затвора, при увеличении частоты увеличивается ток, с которым заряжается емкость затвора, из-за этого увеличивается рассеиваемая мощность. Получаем такую зависимость мощности:  $P = k f U^2$ .

6. Какие таймеры есть у ATmega8535.

Таймер 0 (8bit), Таймер 1(16bit), Таймер 2(8bit).

7. Сколько режимов есть у таймера 1 и режима с каким номером у него нет.

Всего 16 режимов, нет режима под номером 13.

8. Внутренняя структура МК.

МК состоит из блока управления питанием, блока управления сбросом, блока синхронизации, памяти программ, процессора, портов ввода-вывода, ОЗУ.

9. Какие значения записаны в TCCR после сигнала RESET.

Все биты станут нулями.

10. Порт A. Сколько прерываний и сколько регистров ввода/- вывода принадлежит порту A. Назначение этих регистров ввода/вывода. 0 прерываний, 3 регистра: PORTA – регистр данных порта A DDRA – регистр выбора направления передачи данных порта A PINA – нельзя ничего записать, при чтении из него будет прочитано то, что в данный момент присутствует на выводах порта A

11. Регистр SREG. Назначение его разрядов. Бит 0 – C: признак переноса Бит 1 – Z: признак нуля Бит 2 – N: признак отрицательного результата Бит 3 – V: признак переполнения Бит 4 – S: равен сумме по модулю 2 содержимого третьего и второго разрядов Бит 5 – H: признак переноса между полубайтами Бит 6 – T: временное хранение бита Бит 7 – I: глобальное разрешение прерывания Бит 3 – V: признак переполнения Бит 4 – S: равен сумме по модулю 2 содержимого третьего и второго разрядов Бит 5 – H: признак переноса между полубайтами Бит 6 – T: временное хранение бита Бит 7 – I: глобальное разрешение прерывания

12. Почему после сигнала RESET все прерывания запрещены.

Для обеспечения корректной работы МК.

13. Приведите пример использования разряда T в регистре SREG. Передача битов из одного регистра общего назначения в другой: `bst r31, 7`; запись значения седьмого разряда регистра r31 в T. `bld r0, 3`; запись из T в третий разряд регистра r0.

14. Таймер 0. Режимы работы, количество прерываний, регистры ввода/вывода, принадлежащие таймеру 0. Режимы работы:  
Normal (режим 0) – обычный суммирующий счетчик. Phase Correct PWM (режим 1) – ШИМ с точной фазой. CTC (режим 2) – счет по модулю (регистр OCR0). Fast PWM (режим 3) – быстродействующий ШИМ. Прерывания 2: по сравнению и по переполнению. Регистры: пороговый регистр, контрольный регистр, регистр-счетчик+ по два бита в маске разрешённых прерываний и в регистре флагов прерываний.
15. В каких режимах таймера 0 порог изменяется не сразу (двойная буферизация записи) при записи нового значения в регистр порога с помощью команды OUT.  
В ШИМ-режимах таймера(первый и третий).
16. Откуда приходит сигнал на вход TCNTO.  
С выхода управляемого предварительного делителя частоты(prescaler).
17. Как можно разрешить (запретить) прерывания по переполнению таймера 0. Ldi r16, 1 « TOIE0 Out TIMSK, r16 – разрешить Ldi r16, 1 « TOIE0 Out TIMSK, r16 – запретить
18. Написать программу с использованием таймера 0, вырабатывающую симметричное прямоугольное колебание на одном из выходов порта A

```
#include <avr/io.h>
#include <avr/interrupt.h>

.global TIMERO_COMP
TIMERO_COMP:
    in r16, PORTA
    eor r16, r17
    out PORTA, r16
    reti

.global main
main:
    sbi DDRA, DDA0                ;set PA0 as OUTPUT
    cbi PORTA, PORTA0             ;pull PA0 to grownd

    ldi r17, 1 << PORTA0

    sei                           ;enable interrupts
    ldi r16, 1 << OCIE0           ;enable comparison interrupt
    out TIMSK, r16
    ldi r16, 0x7f                 ;threshod on 50%
    out OCR0, r16

    ldi r16, 1 << WGM00 | 1 << CS00 ;phase-correct
    out TCCR0, r16                ;no prescaling

loop:
    nop
    rjmp loop
```

19. Какие коэффициенты деления частоты позволяет получать предварительный делитель таймера 0.  
1, 8, 64, 256, 1024.
20. Какой режим таймера 0 позволяет вырабатывать треугольные колебания, используя дополнительную интегрирующую цепочку.  
Любой: в не-ШИМ режимах достаточно поставить OC0 изменяться при совпадении с порогом; а в ШИМ режимах достаточно выставить порог 0.5 от максимального значения (скважность 0.5).

21. Как запрограммировать предварительный делитель таймера 0.  
Выставить в биты 2:0 регистра TCCR0 значение от 1 до 5.
22. Режим 0 таймера 0. Режим Normal – счетчик TCNT0 работает как обычный суммирующий счетчик, увеличивается на 1 по каждому импульсу тактового сигнала. При переходе через FF счетчик обнуляется (переполнение). Также происходит прерывание по сравнению при совпадении содержимого TCNT0 и OCR0.
23. Режим 1 таймера 0.  
Режим Phase Correct PWM – ШИМ с точной фазой – генерация сигналов с широтно-импульсной модуляцией. Работает на сложение от 0x0 до 0xF, затем на вычитание обратно до 0x00, затем происходит прерывание и смена направления счетчика. При совпадении содержимого счетчика с порогом изменяется состояние выхода C0. Особенность этого режима – двойная буферизация записи в регистр порога CR0 – новое значение сохраняется в буферном регистре, а значение OCR0 изменяется только после прохождения 0xF.
24. Режим 2 таймера 0.  
Режим CTC – Clear Timer on Compare Match – режим счета по модулю, который определяется содержимым регистра порога. Происходит прибавление до совпадения значения счетчика с содержимым OCR0, затем прерывание по сравнению, счетчик переходит в состояние \$00 и процесс повторяется.
25. Режим 3 таймера 0.  
Режим Fast PWM используется для генерации высокочастотного сигнала с широтно-импульсной модуляцией. Состояние TCNT0 меняется от \$00 до \$F F, затем он обнуляется и процесс повторяется. Также есть двойная буферизация, при совпадении содержимого счетчика с пороговым изменяется состояние выхода OC0.
26. Когда меняется порог в режиме 3 таймера 0.  
Изменение значения регистра происходит после достижения счетчиком значения \$FF, до этого новое значение находится в буферном регистре.
27. Можно ли писать в TCNT0 без остановки счета.  
Можно, но есть риск пропустить прерывание.
28. Как можно остановить счет в таймере 0.  
Обнулить биты 2:0 регистра TCCR0.
29. Система прерываний микроконтроллера ATmega8535.  
Система из 21 прерывания, чем меньше номер прерывания, тем оно приоритетнее. При генерации разрешенного прерывания все прерывания запрещаются глобально и исполнение переходит в вектор прерывания. При выходе командой reti восстанавливается ход исполнения и включаются глобально прерывания.
30. Сколько всего прерываний у ATmega8535.  
21
31. Как организовать вложенные прерывания.  
Вложенные - прерывания в начале программы обработки прерывания, для этого нужно разрешить глобально прерывания.
32. Как можно разрешить (запретить) одновременно все прерывания.  
Ассемблерные команды: sei – разрешить, cli – запретить.
33. Как организована система приоритетов при обработке прерываний.  
Каждому прерыванию присваивается номер и первым обрабатывается прерывание с наименьшим номером.
34. Какое минимальное время требуется для преобразования в АЦП.  
65 мкс.

35. Чем сигнальный процессор отличается от МК.  
У микроконтроллера основная задача – это работа с периферийными устройствами, а сигнальный процессор имеет специфичный набор команд и регистров, чтобы быстрее обрабатывать сигналы.
36. Зачем в программе надо устанавливать начальное значение Stack Pointer и чему это значение должно быть равно.  
После RESET не гарантируется правильное значение SP, поэтому выставить в регистре в конец памяти.
37. Сторожевой таймер и особенности его работы.  
WatchDog Timer предназначен для ликвидации последствий сбоев в работе микроконтроллера. Он через определенный промежуток времени перезапускает рабочую программу МК. Чтобы сторожевой таймер не срабатывал при правильной работе программы, используется команда WDR для его сброса.
38. Что такое SPI и зачем он нужен.  
Это последовательный синхронный интерфейс, который позволяет передавать данные с высокой скоростью между ATmega8535 и внешними устройствами.
39. Как инициировать передачу байта в SPI.  
Нужно записать байт в регистр SPDR у MASTER – ведущего микроконтроллера, тогда произойдет передача между ним и SLAVE — ведомым микроконтроллером.
40. Сколько прерываний и сколько регистров ввода/вывода принадлежит SPI.  
1 прерывание: *SPI\_STC*  
3 регистра ввода/вывода: *SPCR*, *SPSR*, *SPDR*.
41. Далее пойдут вопросы про однопроводный интерфейс (сеть MicroLAN).
42. Сколько проводов необходимо для реализации однопроводного интерфейса.  
Для 1 проводной шины необходимы 2 провода – сигнальный, подключенный к плюсу, и провод, подключенный к общему проводу (минусу).
43. Как выглядит физический ноль и физическая единица.  
0 – низкое напряжение (ниже некоторого порогового значения) 1 – высокое напряжение (выше некоторого порогового значения)
44. Как в однопроводном интерфейсе передается информационный ноль и информационная единица? Какова максимальная скорость передачи?  
Логически. Подается физический ноль длительностью 1-15 мкс, затем физический ноль или физическая единица длительностью до 60 мкс, следовательно, считывается 0 или 1.
45. Что такое серийный номер в однопроводном интерфейсе и какова его структура.  
Серийный номер является идентификатором устройства; его структура: 64 бита, из которых 8 бит - код семейства, 48 бит - серийный номер, 8 бит - контрольная сумма.
46. Какая команда позволяет Master определить номера всех Slave в сети MicroLAN.  
Search ROM
47. Как выглядит сигнал сброса в сети MicroLAN.  
Долгий импульс низкого уровня продолжительностью минимум 480мкс, затем долгий импульс высокого уровня, тоже минимум 480мкс.