



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)
КАФЕДРА «Информационная безопасность» (ИУ8)**

Отчёт по лабораторной работе №3

По дисциплине: «Аппаратные средства вычислительной техники»

Тема: «Использование контроллеров и прерываний. Микроконтроллер i8051»

Вариант №3

**Выполнил: Березин М.А.
студент группы ИУ8-73**

**Проверил: Рафиков А. Г.,
Старший преподаватель кафедры
ИУ**

**г. Москва,
2021 г.**

1. Цель работы

Изучение поведения прерываний в микроконтроллере i8051. Получение практических навыков в области программно-аппаратного проектирования подсистем звуковой и визуальной индикации на микроконтроллере i8051.

2. Задание

При нажатии на кнопку SB2 загорается цифра 1 и светодиод VD1 и раздаётся звуковой сигнал $f=1000$ Гц длительностью 2 сек. При нажатии на кнопку SB3 – загорается цифра 2 и светодиод VD2, раздаётся звуковой сигнал $f=2000$ Гц длительностью 2 сек. При одновременном нажатии кнопок SB2 и SB3 загорается цифра 3 и включается светодиод VD3, раздаётся звуковой сигнал $f=3000$ Гц длительностью 2 сек.

3. Ход работы

HG1 – семисегментный индикатор. Обозначения сегментов семисегментного индикатора показаны на рисунке 1.

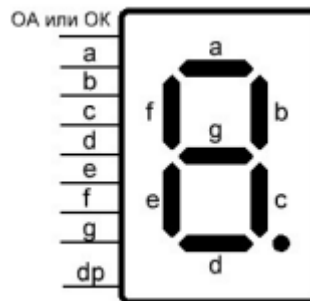


Рисунок 1 – Сегменты индикатора

Частота микроконтроллера – 12 МГц. В задании используется 2 кнопки: SB2 и SB3. При нажимании кнопок должны звучать сигналы две секунды. Следовательно, потребовалось два таймера. Один из них отвечает за отсчёт 2 секунд. Другой задаёт частоту сигнала звука.

Один такт таймера имеет частоту: $f_{Timer} = \frac{12 \text{ МГц}}{12} = 1 \text{ МГц} = 10^{-6} \text{ с}$.

Для задания времени в 2 секунды используем следующую формулу:

$$2 \text{ сек} = f_{Timer} \cdot 256 \cdot 32 \cdot 244 = 1.998848 \text{ с}$$

Для расчета длительности задержки для генерации сигнала определенной частоты применяется формула: $n = \frac{1}{2 \cdot f \cdot 10^{-6}}$, где n – количество итераций, которые должен пройти таймер, формирующий звуковой сигнал.

Для того, чтобы сканировать нажатие клавиш, возможно несколько вариантов. Первый – это использовать внешние прерывания – этот метод мною не был использован. Второй – это просто сканирование «бесконечно» на нажатие кнопок.

В связи с тем, что кнопка обладает эффектомдребезжания, приходится добавить дополнительную задержку, чтобы не было скачков состояния кнопки. Так же лучше добавить конденсатор. По условию схемы время переключения кнопки = 1 мс.

Следовательно требуется задержка сигнала > 1 мс.

Кроме того, присутствует следующая проблема: По факту, после того как происходит запуск таймера частоты сигнала, идёт отсчёт требуемой частоты и таким образом генерируется звук. Следовательно, когда происходит повторное нажатие, происходит повторный вызов запуска таймера – следовательно происходит нарушение частоты звука. Да, это будет на один сигнал, но это будет физически заметно.

Для решения этой проблемы в моем коде было добавлено несколько флагов, которые инициализируются «1»-ой тогда, когда происходит запуск таймера на 2 секунды. Это делается для того, чтобы, не происходило ненужного перезапуска генерации звука. Вместо этого просто происходит обновление только таймера на время в 2 секунды. И таким образом это и реализовано.

Для задания частоты звука используется таймер №1.

Для задания частоты в 1000 Гц таймер инициализируется значением FC17h

Для задания частоты в 2000 Гц таймер инициализируется значением F82Fh

Для задания частоты в 3000 Гц таймер инициализируется значением F447h

На рисунках 2-4 изображена диаграмма осциллографа, доказывающая частоту создаваемого звука.

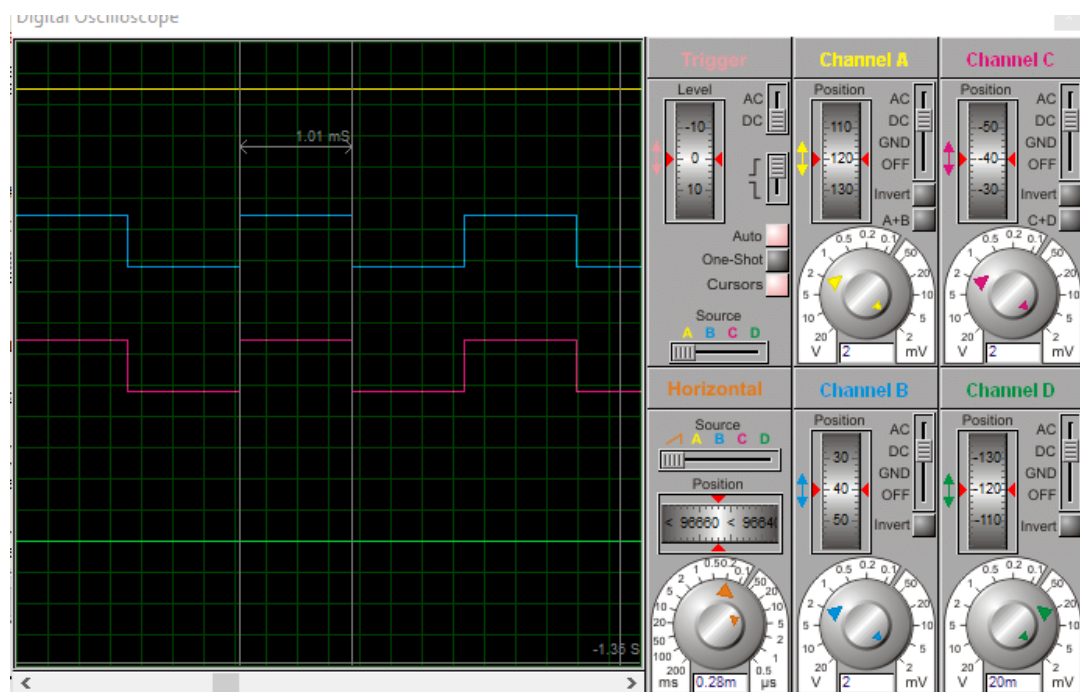


Рисунок 2 – Осциллограф звука с частотой 1000 Гц

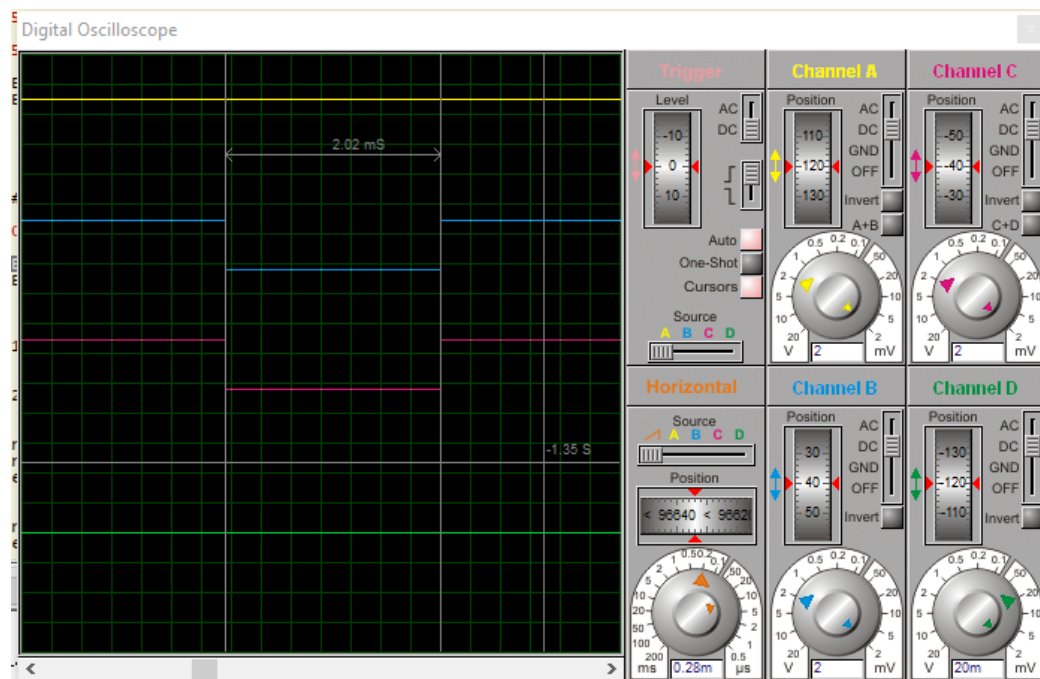


Рисунок 3 – Осциллограф звука с частотой 2000 Гц

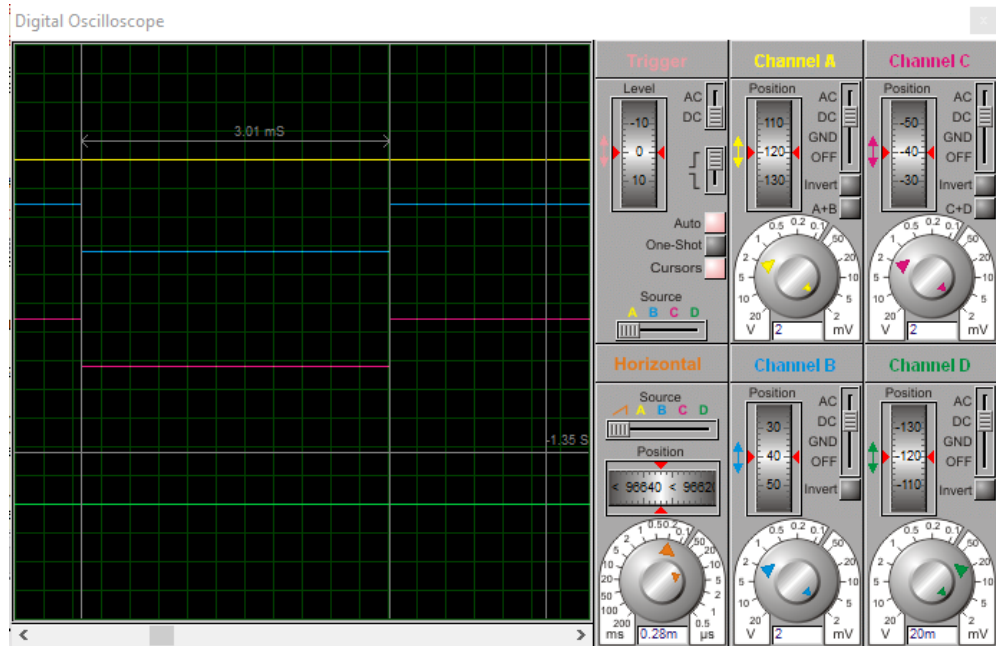


Рисунок 4 – Осциллограф звука с частотой 3000 Гц

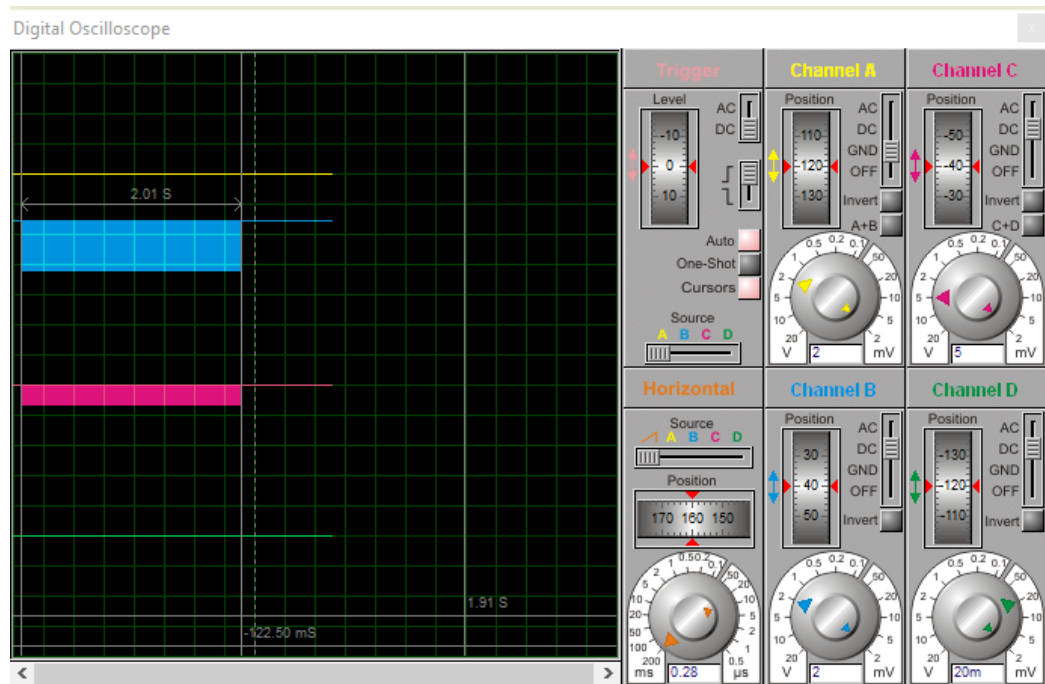


Рисунок 5 – Демонстрация длительности звукового сигнала

4. Тестирование реализованного алгоритма при помощи схемы тестирования в программе моделирования Proteus

Для тестирования данной схемы достаточно запустить и нажать на клавиши «1» и «2» на клавиатуре. Таким образом установлены клавиши для режима 1 и 2. Чтобы нажать две кнопки одновременно, достаточно просто нажать одну из кнопок. И нажмутся обе.

Схема представлена на рисунке 6.

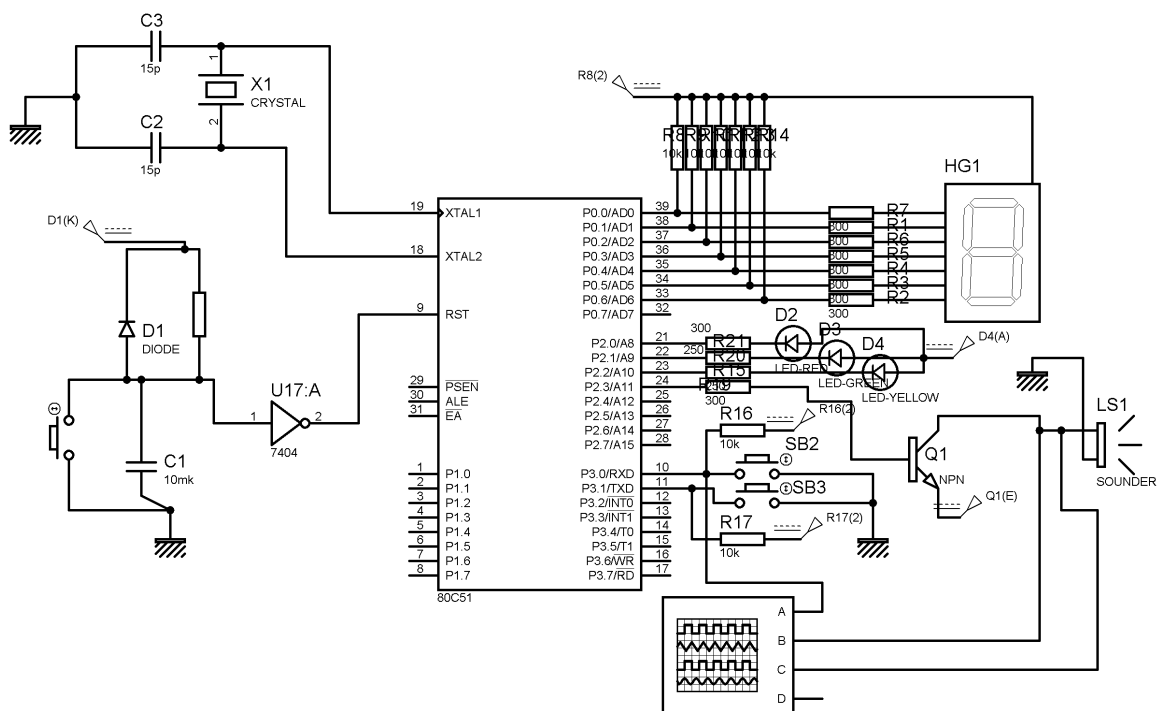


Рисунок 6 – Изображение схемы

В приложении Б представлен исходный код на языке ассемблера.

5. Выводы

Было изучено поведение прерываний в микроконтроллере i8051. Получены практические навыки в области программно-аппаратного проектирования подсистем звуковой и визуальной индикации на микроконтроллере i8051.

Были выполнены все поставленные задания и подготовлена демонстрация
схемы.

По выполненной работе был составлен отчёт.

Приложение А. Листинг для задания

```
$NOMOD51
$INCLUDE (8051.MCU)

    sound1Flag BIT 0F0H
    sound2Flag BIT 0F1H
    sound3Flag BIT 0F2H

    LED1 BIT P2.0
    LED2 BIT P2.1
    LED3 BIT P2.2

    SPEAKER BIT P2.3

    BTN1 BIT P3.0
    BTN2 BIT P3.1

    C1 BIT 0F3H
    C2 BIT 0F4H
    inWork BIT 0F5h

;=====

    org    0000h
    AJMP MAIN

org 000Bh

    ACALL T0SubRoute
    RETI

org 001Bh

    CPL SPEAKER
    MOV TH1, R5
```

```
MOV TL1, R6
RETI
```

```
org 100h
```

```
MAIN:
```

```
    SETB BTN1            ; Make it as input
    SETB BTN2            ; Make it as input
    SETB LED1
    SETB LED2
    SETB LED3
    MOV P0, #0FFh
    CLR SPEAKER
    SETB EA
    SETB ET0
    SETB ET1
    ORL TMOD, #00010001b
```

```
SubRoute:
```

```
SJMP Scan_BTNs
```

```
T0SubRoute:
```

```
DJNZ R4, next
```

```
    CLR TR1
```

```
CLR TR0
```

```
    SETB LED1
```

```
SETB LED2
```

```
SETB LED3
```

```
    CLR sound1Flag
```

```
    CLR sound2Flag
```

```
    CLR sound3Flag
```

```
    MOV P0, #01111111b
```

```
next:
```

```
MOV TL0, #0h
```

```
MOV TH0, #0E0h
```

```
RET
```


DELAY:

MOV R2, #255;#16

REP:

MOV R3, #255

REP2:

DJNZ R3, REP2

DJNZ R2, REP

RET

Scan_BTNs2:

MOV R2, #16;#16

REP11:

MOV R3, #40

REP12:

DJNZ R3, REP12

DJNZ R2, REP11

Scan_BTNs:

MOV C, BTN1 ; Scan Button 1

MOV C1, C

MOV C, BTN2

MOV C2, C

JB C1, Scan_2BTN

JNB C2, HandleTwoPressed

SJMP HandleFirstPressed

Scan_2BTN:

JB C2, Scan_BTNs2 ; When no buttons were pressed

SJMP HandleSecondPressed

RESET_TIME: ; 2 1/32

ORL TMOD, #00010001b

MOV TL0, #0h

MOV TH0, #0E0h

MOV R4, #244

SETB TR0

RET

SET_T1:

CPL SPEAKER

MOV TH1, R5

MOV TL1, R6

SETB TR1

RET

HandleFirstPressed:

ACALL RESET_TIME

CLR LED1

MOV R5, #0FCh ; #0FAh

MOV R6, #017h ; #0CBh

JB sound1Flag, noL1

ACALL SET_T1

noL1:

MOV P0, #11111001b

ACALL DELAY

SETB sound1Flag

SJMP Scan_BTNs

HandleSecondPressed:

ACALL RESET_TIME

CLR LED2

MOV R5, #0F8h;MOV R5, #0FEh

MOV R6, #2Fh;MOV R6, #0Ch

```
        JB sound2Flag, noL2
ACALL SET_T1
noL2:
MOV P0, #10100100b
ACALL DELAY
        SETB sound2Flag
SJMP Scan_BTNs
```

HandleTwoPressed:

```
ACALL RESET_TIME
        CLR LED3
MOV R5, #0F4h;#0FEh
MOV R6, #47h      ;#0B3h
        JB sound3Flag, noL3
ACALL SET_T1
noL3:
MOV P0, #10110000b
ACALL DELAY
        SETB sound3Flag
SJMP Scan_BTNs

END
```