



**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
PIC12F675»**

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 PIC12F675. Получение практических навыков в области программно-аппаратного проектирования подсистем звуковой и визуальной индикации на микроконтроллере PIC.

2. Задание

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

3. Ход работы

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

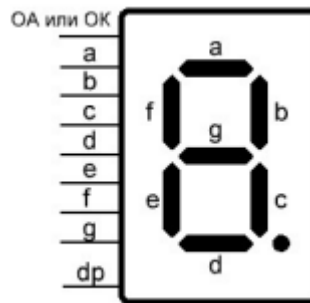


Рисунок 1 – Сегменты индикатора

Частота микроконтроллера – 4 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: $f_{Timer} = \frac{4 \text{ МГц}}{4} = 1 \text{ МГц} = 10^{-6} \text{ с}$.

Для задания времени в 2 секунды используем следующую формулу:

$$2 \text{ сек} = f_{Timer} \cdot 128 \cdot 125 \cdot 128 = 2 \text{ с}$$

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. Вторым – 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 реализации задержки на 2 секунды используется таймер T0 с установкой в него значения b«125». При этом используется предделитель для создания коэффициента деления 1:125.

Для синхронизации и задания задержки в 1 мс, используется таймер T0. И идентифицируется значением «83h». С установкой предделителя на 8.

Для задания частоты звука используется таймер 1.

Для задания частоты в 1000 Гц таймер инициализируется значением FC18h

Для задания частоты в 2000 Гц таймер инициализируется значением FE0Ch

Для задания частоты в 3000 Гц таймер инициализируется значением FEB3h

При всех этих значениях на таймер T1 устанавливается значение предделителя на коэффициент 1:1

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

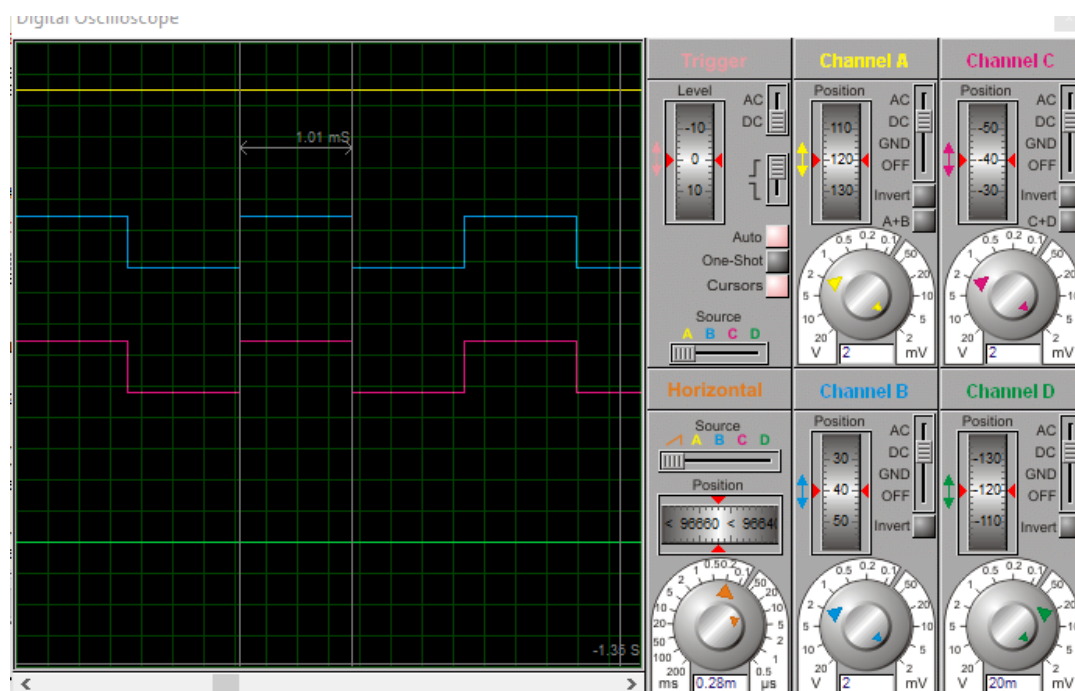


Рисунок 2 – Осциллограф звука с частотой 1000 Гц

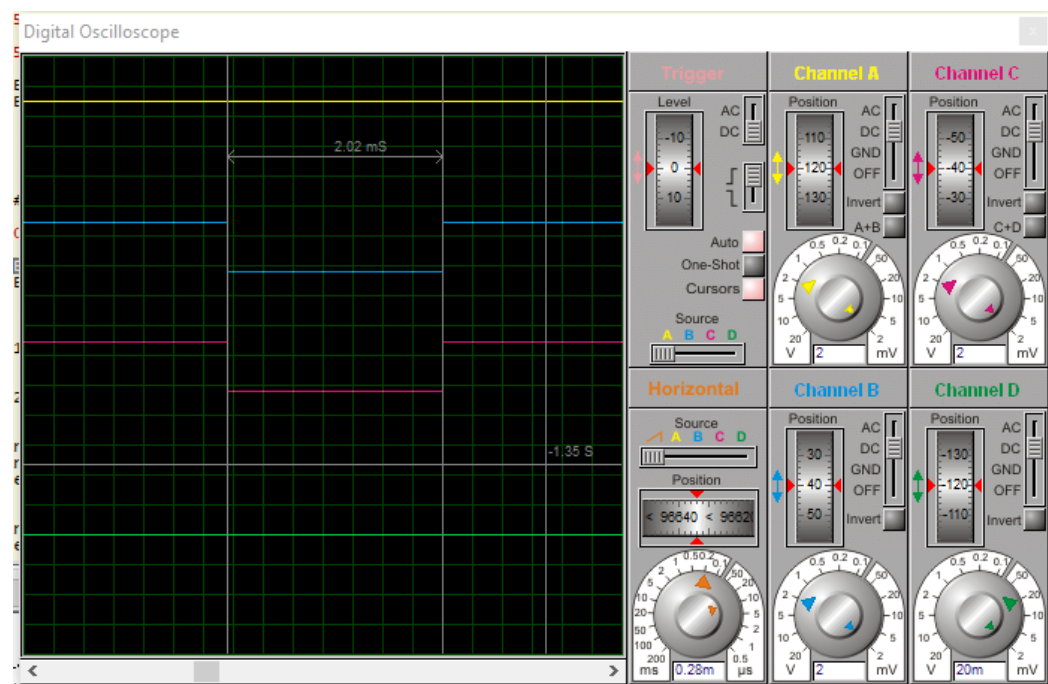


Рисунок 3 – Осциллограф звука с частотой 2000 Гц

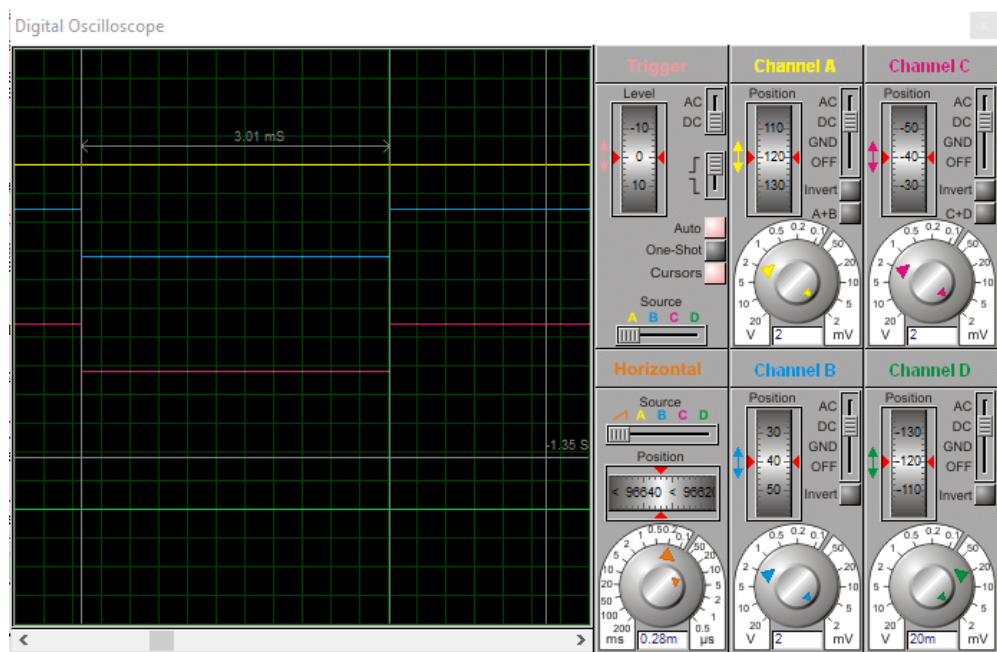


Рисунок 4 – Осциллограф звука с частотой 3000 Гц

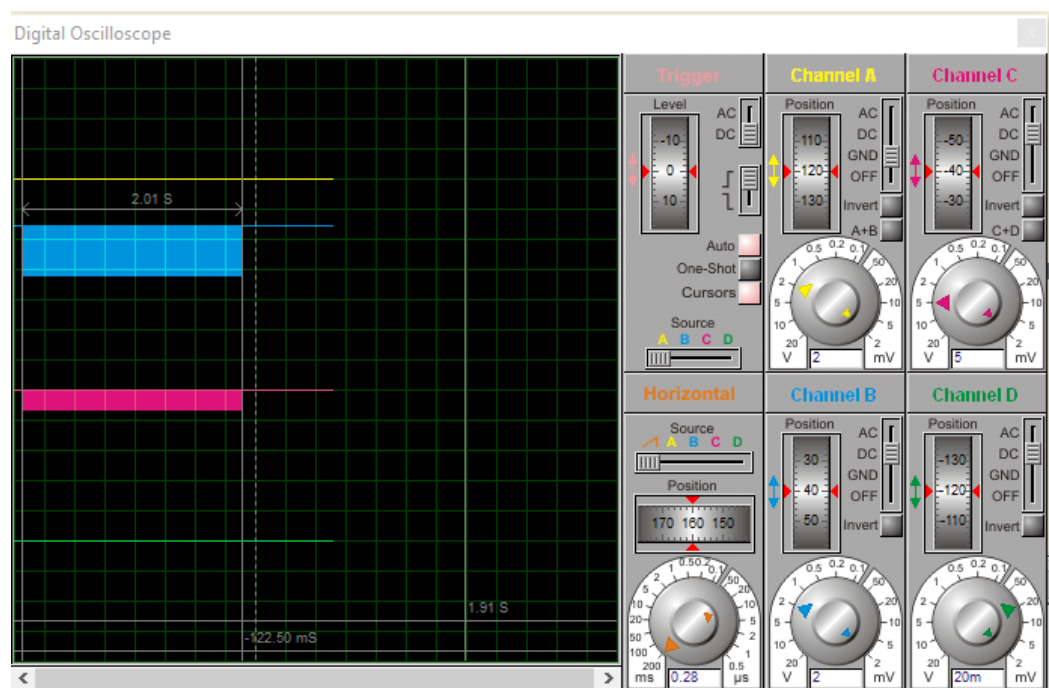


Рисунок 5 – Демонстрация длительности звукового сигнала

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.

Приложение А. Листинг для задания

```
;=====
; Main.asm file generated by New Project wizard
;
; Created:   Вс ноя 14 2021
; Processor: PIC12F675
; Compiler:  MPASM (Proteus)
;=====

;=====
; DEFINITIONS
;=====

#include p12f675.inc                ; Include register definition file
__CONFIG b'11111111010100'

;=====
; CONSTANTS
;=====

GPIO_DATA equ b'00101000'
INTERRUPT_INIT equ b'11001000'
PIE1_INIT equ b'00000000'
TMR1H_INIT equ 0x0
TMR1L_INIT equ 0x0
TMR0_INIT equ 0x0
WPU_INIT equ b'010111'
IOC_INIT equ b'101000'
T1CON_INIT equ b'00000000'

;=====
; VARIABLES
;=====

sound_flags equ 20h
snd1_flag equ 00h
snd2_flag equ 01h
snd3_flag equ 02h
```



```
;LEDS_STATES equ 21h
;LED1 equ 00h
;LED2 equ 01h
;LED3 equ 02h
```

```
sound equ GP0
```

```
btn1 equ GP3
btn2 equ GP5
```

```
pc_states equ 21h
in_work_flag equ 00h
btn_wait_flag equ 01h
btn2_st equ 02h
btn3_st equ 03h
```

```
T1H_REG equ 22h
T1L_REG equ 23h
```

```
T02SecScaler equ 24h
```

```
LastPortState equ 25h
```

```
Accum equ 26h
```

```
Lcd_data equ 27h
```

```
Loop_ident equ 28h
```

```
;=====
; RESET and INTERRUPT VECTORS
;=====
```

```
    ; Reset Vector
RST    code    0x0
        goto   START
```

```
INT code 0x4
    goto IINT_HNDL
```

```
;=====
; CODE SEGMENT
;=====
```

```
PGM    code
```

```
Init
```

```
    banksel ANSEL
    clrf ANSEL
```

```
    banksel CMCON
    movlw 0x7
    movwf CMCON
```

```
    banksel OPTION_REG
    movlw b'01001000'    ; Options
    movwf OPTION_REG
```

```
    banksel TRISIO
    movlw b'00101000'    ; GPIO Port options
    movwf TRISIO
```

```
    banksel GPIO
    movlw b'10' ; Default GPIO data
    movwf GPIO_DATA
    movwf GPIO
```

```
    banksel WPU
    movlw WPU_INIT ; Pull ups
    movwf WPU
```

```
    banksel INTCON
    movlw INTERRUPT_INIT ; Interrupts
```

```
movwf INTCON
```

```
banksel PIE1
```

```
movlw PIE1_INIT ; Enables timer 1 (16-bit) INTERRUPT (NO)
```

```
movwf PIE1
```

```
banksel TMR1H
```

```
movlw TMR1H_INIT
```

```
movwf TMR1H
```

```
movlw TMR1L_INIT
```

```
movwf TMR1L
```

```
banksel T1CON
```

```
movlw T1CON_INIT ; Enables timer 1
```

```
movwf T1CON
```

```
banksel IOC
```

```
movlw IOC_INIT
```

```
movwf IOC
```

```
CALL SavePortStates
```

```
return
```

```
IINT_HNDL
```

```
    btfsc INTCON, GPIF
```

```
    GOTO GPIO_INT
```

```
    btfsc INTCON, T0IF
```

```
    GOTO T0_INT
```

```
    banksel PIR1
```

```
    btfsc PIR1, TMR1IF
```

```
    GOTO T1_INT
```

```
    return
```

```
SavePortStates
```

```

banksel GPIO
movf GPIO, w
movwf LastPortState
return

```

CheckOnDownChanges

```

banksel GPIO
; Проверка SB2
btfss LastPortState, GPIO3
GOTO CompareSB3
CompareSB2_1
    btfsc GPIO, GPIO3
        GOTO CompareSB3
        iorlw 0FFh
        return ; Произошло нажатие SB2

```

CompareSB3

```

    btfsc LastPortState, GPIO5
    GOTO CompareSB3_1
    clrw
    return
CompareSB3_1
    btfsc GPIO, GPIO5
        GOTO Nothing
        iorlw 0FFh
        return ; Произошло нажатие SB2

```

Nothing

```

    ; Произошли только отжатия
    clrw
    return

```

Reset2SecTimer

```

CALL Start2SecTmr
CALL SavePortStates
bcf INTCON, GPIF
retfie

```

GPIO_INT

CALL CheckOnDownChanges

;movf W, w

;banksel STATUS

btfss STATUS, Z

GOTO IfUp

CALL SavePortStates

bcf INTCON, GPIF

retfie

; Только отжатия

IfUp

SB2_check

banksel GPIO

btfsc GPIO, GPIO3

GOTO SB3_check

; Обработка нажатия клавиши SB2 + проверка нажатия сразу второй

btfsc GPIO, GPIO5

GOTO ONLY_SB2

; Если нажаты две кнопки

btfsc sound_flags, snd3_flag

GOTO Reset2SecTimer; Если текущий режим подошёл

; Если не режим текущий или не запущен просто

bcf INTCON, T0IE

movlw 1h

andwf pc_states, f

clrf sound_flags

CALL Start_Mode3

bcf INTCON, GPIF

CALL SavePortStates

retfie

ONLY_SB2

; Запуск таймера на 1 мс в ожидании нажатия второй кнопки,
после этого происходит обработка кнопок (ЕСЛИ НЕ ЗАПУЩЕН УЖЕ!!!!!!!!!!!!!!)

bcf INTCON, T0IF

btfsc sound_flags, snd1_flag

GOTO Reset2SecTimer; Если текущий режим подошёл

```

; Если не режим текущий или не запущен просто
btfsc pc_states, btn_wait_flag ; Проверка, что таймер
уже запущен

GOTO ONLY_set_bitSB2
; Если не запущен таймер
movlw b'01000010'
banksel OPTION_REG
movwf OPTION_REG
movlw 0BBh
banksel TMR0
movwf TMR0 ; Тут стартовал

таймер на 1 мс

bsf INTCON, T0IE ; Тут разрешили
преоывания

bsf pc_states, btn_wait_flag
ONLY_set_bitSB2
bsf pc_states, btn2_st ; Запись, что

кнопка 2 нажата

;CALL Start_Mode1
bcf INTCON, GPIF
CALL SavePortStates
retfie

SB3_check
banksel GPIO
btfss GPIO, GPIO5
GOTO SB3_Pressed ; Если нажата все-таки клавиша 3
bcf INTCON, GPIF
CALL SavePortStates
; Ни одна кнопка не была нажата, следовательно, произошло ОТЖАТИЕ
retfie
SB3_Pressed
; Обработка нажатия клавиши SB3 + проверка нажатия сразу

первой

btfsc GPIO, GPIO3
GOTO ONLY_SB3

```

```

; Если нажаты две кнопки
btfsc sound_flags, snd3_flag
    GOTO Reset2SecTimer; Если текущий режим подошёл

; Если не режим текущий или не запущен просто
bcf INTCON, T0IE
movlw 1h
andwf pc_states, f
CALL Start_Mode3
bcf INTCON, GPIF
CALL SavePortStates
retfie

ONLY_SB3
; Запуск таймера на 1 мс в ожидании нажатия
ПЕРВОЙ кнопки, после этого происходит обработка кнопок (ЕСЛИ НЕ ЗАПУЩЕН УЖЕ!!!!!!!!!!!!)
btfsc sound_flags, snd2_flag
    GOTO Reset2SecTimer; Если текущий режим подошёл

; Если не режим текущий или не запущен просто
bcf INTCON, T0IF
btfsc pc_states, btn_wait_flag
    ;

Проверка, что таймер уже запущен
GOTO ONLY_set_bitSB3
; Если не запущен таймер
movlw b'01000010'
banksel OPTION_REG
movwf OPTION_REG
movlw 0BBh
banksel TMR0
movwf TMR0
    ; Тут

стартовал таймер на 1 мс

bsf INTCON, T0IE
    ; Тут разрешили

преоывания

bsf pc_states, btn_wait_flag
ONLY_set_bitSB3
bsf pc_states, btn3_st
    ; Запись,

что кнопка 2 нажата

```

```

;CALL Start_Mode2
bcf INTCON, GPIF
CALL SavePortStates
retfie

```

Start_Mode3

```

; Запустить таймер для времени в 2 сек (Таймер 0)
CALL Start2SecTmr
; Запустить таймер 1 для отработки звука с заданной частотой в 3 кГц
btfsc sound_flags, snd3_flag
return; Если текущий режим подошёл
; Если не режим текущий или не запущен просто
movlw 0FEh
movwf T1H_REG
banksel TMR1H
movwf TMR1H
movlw 0B3h
movwf T1L_REG
movwf TMR1L
banksel PIE1
bsf PIE1, TMR1IE
bsf INTCON, PEIE

movlw b'11110010'
movwf Lcd_data
CALL Fill_LCD

banksel T1CON
bsf T1CON, TMR1ON ; Запускаем сам таймер
bsf sound_flags, snd3_flag
return

```

Start2SecTmr

```

movlw b'01000110'
banksel OPTION_REG
movwf OPTION_REG
movlw 7Ch

```



```

movwf T02SecScaler
bsf INTCON, T0IE          ; Тут разрешили преобывания
bcf INTCON, T0IF
return

```

T0_INT

```

;bsf INTCON, GIE

btfss pc_states, btn_wait_flag
GOTO Check_on2sec

; Если по истечению 1 мс для кнопок
btfss pc_states, btn2_st
GOTO Check_btn3

; Если нажата 1 кнопка
btfss pc_states, btn3_st      ; Нажата ли вторая?
GOTO Pressed_ONLY_btn2 ; Если нажата только одна
; Если нажаты две кнопки
movlw 1h
andwf pc_states, f
clrf sound_flags
CALL Start_Mode3
retfie

Pressed_ONLY_btn2
movlw 1h
andwf pc_states, f
clrf sound_flags
CALL Start_Mode1
retfie

Check_btn3
; Если нажата ТОЛЬКО 2 кнопка
movlw 1h
andwf pc_states, f
clrf sound_flags
CALL Start_Mode2
retfie

; Если вышло время в 2 сек
Check_on2sec

```

```
decfsz T02SecScaler, f
GOTO ScalerNotZero
GOTO ScalerISZero
```

ScalerISZero

```
banksel T1CON
clrf sound_flags
bcf INTCON, T0IE ; Выключаем прерывания Таймера 0
;bcf PIE1, TMR0IE ; Выключаем прерывания Таймера 1
bcf T1CON, TMR1ON ; Выключаем Таймер 1
retfie
```

ScalerNotZero

```
movlw 83h
movwf TMR0
bcf INTCON, T0IF
bsf INTCON, T0IE
retfie
```

T1_INT

```
;bsf INTCON, GIE
banksel GPIO
btfss GPIO, sound
GOTO SET1_inSound
; Если сейчас SOUND = 1
bcf GPIO, sound
GOTO T1_refull
SET1_inSound ; Если сейчас SOUND = 0
bsf GPIO, sound
```

T1_refull

```
movf T1H_REG, w
movwf TMR1H
movf T1L_REG, w
movwf TMR1L
banksel PIR1
bcf PIR1, TMR1IF
retfie
```

```

Fill_LCD
    movlw .9
    movwf Loop_ident
    banksel GPIO
    Loop_bits
        decfsz Loop_ident
        GOTO Fill_LCD_bit
        GOTO Shift_REG
        Fill_LCD_bit
            btfss Lcd_data, 0h
            GOTO Set_0
                movlw b'00000010'
                iorwf GPIO, f

                movlw b'11101111'
                andwf GPIO, f

                movlw b'00010000'
                iorwf GPIO, f

                movlw b'11101111'
                andwf GPIO

                rrf Lcd_data, f
                GOTO Loop_bits
        Set_0
            movlw b'11111101'
            andwf GPIO, f

            movlw b'11101111'
            andwf GPIO, f

            movlw b'00010000'
            iorwf GPIO, f

            movlw b'11101111'
            andwf GPIO

```

```
rrf Lcd_data, f
GOTO Loop_bits
```

Shift_REG

```
movlw b'00000100'
iorwf GPIO
```

```
movlw b'11111011'
andwf GPIO
```

```
return
```

Start_Model

```
; Запустить таймер для времени в 2 сек (Таймер 0)
CALL Start2SecTmr
; Запустить таймер 1 для отработки звука с заданной частотой в 3 кГц
btfsc sound_flags, snd1_flag
return; Если текущий режим подошёл
; Если не режим текущий или не запущен просто
movlw 0FCh
movwf T1H_REG
banksel TMR1H
movwf TMR1H
movlw 18h
movwf T1L_REG
movwf TMR1L
banksel PIE1
bsf PIE1, TMR1IE
bsf INTCON, PEIE

movlw b'01100000'
movwf Lcd_data
CALL Fill_LCD

banksel T1CON
bsf T1CON, TMR1ON ; Запускаем сам таймер
bsf sound_flags, snd1_flag
return
```

Start_Mode2

; Запустить таймер для времени в 2 сек (Таймер 0)

CALL Start2SecTmr

; Запустить таймер 1 для отработки звука с заданной частотой в 3 кГц

btfsc sound_flags, snd2_flag

return; Если текущий режим подошёл

; Если не режим текущий или не запущен просто

movlw 0FEh

movwf T1H_REG

banksel TMR1H

movwf TMR1H

movlw 0Ch

movwf T1L_REG

movwf TMR1L

banksel PIE1

bsf PIE1, TMR1IE

bsf INTCON, PEIE

movlw b'11011010' ; '11011010'

movwf Lcd_data

CALL Fill_LCD

banksel T1CON

bsf T1CON, TMR1ON ; Запускаем сам таймер

bsf sound_flags, snd2_flag

return

START

CALL Init

Start_loop

GOTO Start_loop

;=====

END