

ГУАП

КАФЕДРА № 44

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

канд. техн. наук  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

Т.Н.Соловьёва  
\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

### РАЗРАБОТКА МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ТАЙМЕРОВ

по курсу: МИКРОКОНТРОЛЛЕРНЫЕ СИСТЕМЫ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4143

\_\_\_\_\_  
подпись, дата

Е.Д.Тегай  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2024

## **Цель работы**

Изучение принципов работы таймеров и системы прерываний микроконтроллера; приобретение навыков разработки микроконтроллерных систем, использующих таймеры.

## **Задание по работе**

Требуется разработать микроконтроллерную систему «Миллисекундомер», включающую в себя микроконтроллер семейства MCS-51, ЖКИ и одну или две кнопки (в зависимости от варианта).

При включении системы на первой строке ЖКИ выводится ФИО автора работы. При наступлении события «Старт» запускается миллисекундомер, при этом на второй строке появляется курсор в виде мигающего черного прямоугольника. При наступлении события «Стоп» останавливается миллисекундомер, при этом на второй строке гаснет курсор и выводится измеренное время с точностью до микросекунд.

В разделе «Варианты заданий» указан таймер, который необходимо использовать, максимальное время, на которое должен быть рассчитан миллисекундомер, и внешние прерывания, с которыми нужно связать события «Старт» и «Стоп». Выводы, через которые к МК требуется подключить ЖКИ, определяются вариантом подключения, указанным в работе 3. Работу системы необходимо проверить с помощью симулятора.

Задание индивидуального варианта №9 продемонстрировано на рисунке 1, где для удобства восприятия необходимые данные выделены жёлтым цветом.

Номер варианта	Таймер	T <sub>max</sub> , мс	Старт	Стоп
1	0	20	INT0	INT0
2	1	8	INT0	INT1
3	2	15	INT1	INT0
4	0	20	INT1	INT1
5	1	10	INT0	INT0
6	2	5	INT0	INT1
7	0	28	INT1	INT0
8	1	4	INT1	INT1
9	2	6	INT0	INT0

Рисунок 1 – Индивидуальное задание

На рисунке 2 продемонстрированы данные варианта №9, использованные для выполнения лабораторной работы №3.

Номер варианта	Шина управления ЖКИ			Шина данных ЖКИ	Вид курсора
	RS	RW	E		
1	P1.0	P1.1	P1.2	P2.0 – P2.7	нет
2	P1.4	P1.5	P1.6	P2.0 – P2.3	подчеркивание
3	P1.1	P1.2	P1.3	P2.4 – P2.7	мигающий
4	P1.5	P1.6	P1.7	P2.0 – P2.7	мигающий
5	P1.0	P1.1	P1.2	P2.0 – P2.3	нет
6	P1.5	P1.6	P1.4	P2.4 – P2.7	подчеркивание
7	P1.0	P1.1	P1.2	P2.0 – P2.7	нет
8	P1.1	P1.0	P1.3	P2.0 – P2.3	подчеркивание
9	P1.5	P1.4	P1.6	P2.4 – P2.7	мигающий

Рисунок 2 – Индивидуальное задание

### Разработка схемы микроконтроллерной системы

Схема состоит из ЖКИ размером 2x20, использованный также в лабораторной работе №3, а также простейшей клавиатуры. Это показано на рисунке 3.

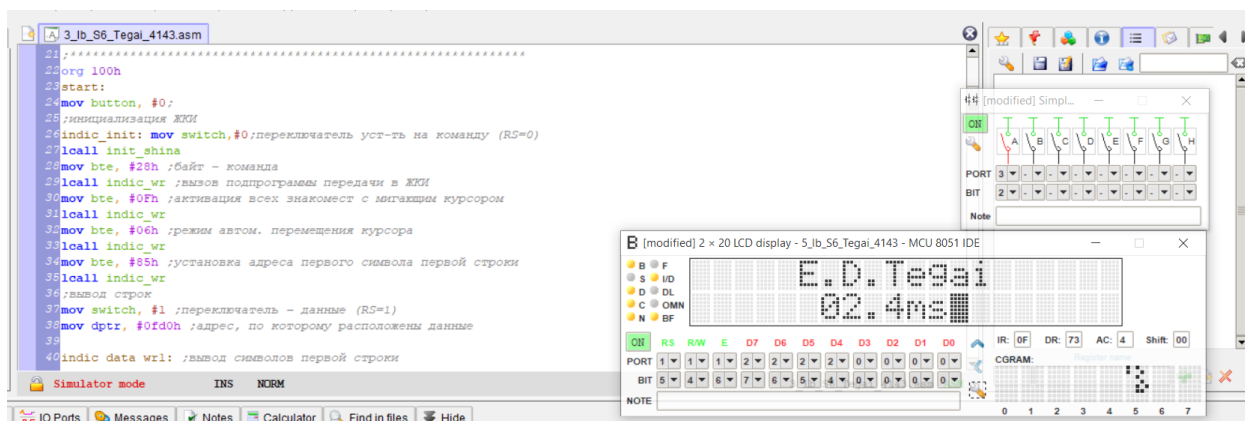


Рисунок 3 – Схема микроконтроллерной системы

## Разработка программы

Программа по своей сути является модифицированной программы из лабораторной работы №4, которая тоже является модификацией лабораторной работы №3. Рассмотрим подробнее нововведения относительно лабораторной работы №4.

Для начала добавляется новая переменная `button`, которая будет хранить в себе значение 1 или 0. Она является неким флагом факта нажатия на кнопку. Также была добавлена метка `tim_2`, которая описывает логику поведения таймера 2 (о чём – далее).

После того, как на первой строке вывелось ФИО в регистр `T2CON` передаётся определённое значение, задающее настройки таймера. Если кнопка нажата (момент «Старт»), то идёт переход к метке `tim_2`, внутри которой задаётся граничное значение таймера (в данном случае – 6 мс). Далее идёт работа меток `nexit` и `exit`, которые описывают счёт времени с помощью заранее заведённых вспомогательных переменных `ms`, `hmks`, `dms` и `hms`.

Помимо этого, в коде также фигурирует метка `int_0`, которая является неким обработчиком прерывания, связанного с `INT0`. Одним из ключевых моментов этой метки является передача определённых значений в регистры `TH2`, `TL2`, `RCAP2L`, `RCAP2H`. Это нужно для установки предварительных значений таймера. Использование данного метода связано с тем, что сам по себе таймер 2 работает в 6 раз быстрее таймеров 1 и 0, из-за чего требуется хоть как-то «замедлить» его работу.

Так как кнопка для события «Стоп» та же, то для реализации данного события есть метка int\_00, которая тоже является обработчиком прерывания относительно кнопки INTO.

### Текст программы

```
;*****; *
; Filename: 6_lb_S6_Tegai_4143.asm
; Date: 2024/05/01
; File Version: 1
; Author: Tegai E.D.
; Company: SUAI
; Description: 6 laba
; * *
;*****
; Variables
;*****
switch equ 43h ;переключатель «команда-данные» (RS)
bte equ 44h ;выдаваемый на ЖКИ байт
hmks equ 45h ;сотни мкс
ms equ 46h ;единицы мс
dms equ 47h ;десятки мс
hms equ 48h ;сотни мс
button equ 50h
;*****
; Reset Vector
;*****
org 0h ; processor reset vector
ajmp start ; go to beginning of program
;*****
org 0003h ; processor reset vector
ajmp int_0
org 002bh ; processor interrupt vector
ajmp tim_2
; MAIN PROGRAM
;*****
org 100h
start:
mov button, #0;
;инициализация ЖКИ
indic_init: mov switch,#0;переключатель уст-ть на команду (RS=0)
lcall init_shina
mov bte, #28h ;байт – команда
```

```

lcall indic_wr ;вызов подпрограммы передачи в ЖКИ
mov bte, #0Fh ;активация всех знакомест с мигающим курсором
lcall indic_wr
mov bte, #06h ;режим автом. перемещения курсора
lcall indic_wr
mov bte, #85h ;установка адреса первого символа первой строки
lcall indic_wr
;Вывод строк
mov switch, #1 ;переключатель – данные (RS=1)
mov dptr, #0fd0h ;адрес, по которому расположены данные

```

```

indic_data_wr1: ;вывод символов первой строки
clr a
movc a, @a+dptr

```

```

ind_row1:
mov bte, a ;передаваемый байт – код символа
lcall indic_wr
inc dptr
mov a, dpl ;младший байт указателя данных
cjne a, #0D9h, indic_data_wr1
;пока не введены 4 символа 1ой строки
mov switch, #0 ;RS=0 – команда
mov bte, #0C6h ;установка адреса первого символа второй строки
lcall indic_wr
mov switch, #1 ;RS=1 - данные

```

```

mov T2CON, #00000000b
setb ET2
setb EX0 ;
setb EA
setb it0
mov bte, #0Ch
mov switch, #0
lcall indic_wr
jmp main_loop

```

```

init_shina:
mov p2, #00100000b
setb p1.6 ;E:=1
clr p1.4 ;RW:=0 (запись)
mov a, switch
mov c, acc.0 ;нам нужен 0-ой бит аккумулятора
mov p1.5, c ;RS:=switch (команда/данные)
lcall indic_delay ;вызов подпрограммы задержки

```

```

clr p1.6 ;E:=0
lcall indic_delay
setb p1.6 ;E:=1
ret
indic_wr:
mov a, bte
mov p2, a ;передаваемый байт – в P2
setb p1.6 ;E:=1
clr p1.4 ;RW:=0 (запись)
mov a, switch
mov c, acc.0 ;нам нужен 0-ой бит аккумулятора
mov p1.5, c ;RS:=switch (команда/данные)
lcall indic_delay ;вызов подпрограммы задержки
clr p1.6 ;E:=0
lcall indic_delay
setb p1.6 ;E:=1
mov a, bte
swap a
mov p2, a
lcall indic_delay ;вызов подпрограммы задержки
clr p1.6 ;E:=0
lcall indic_delay
setb p1.6 ;E:=1
ret
indic_delay: ;подпрограмма задержки на 40мкс
push A ;сохраняем аккумулятор в стеке
mov A, #0Ah ; 40 = 2+2+1+A(1+2)+1+2+2
m: dec A
jnz m
nop
pop A ;восстанавливаем значение аккумулятора
ret

```

```

int_00:
clr ET2
mov switch, #1 ;переключатель уст-ть на данные (RS=1)
mov a,dms
add a,#30h
mov bte,a
lcall indic_wr
mov a,ms
add a,#30h
mov bte,a
lcall indic_wr
mov bte,#'.'

```

```

lcall indic_wr
mov a,hmks
add a,#30h
mov bte,a
lcall indic_wr
mov bte,#'m'
lcall indic_wr
mov bte,#'s'
lcall indic_wr
mov switch, #0 ;RS=0 – команда
mov bte, #0Fh
mov switch, #0
lcall indic_wr
mov bte, #1100b ;установка адреса первого символа
lcall indic_wr ;второй строки
clr EX0 ;для нового старта
setb EX0
reti

```

```

int_0:
clr tr2
inc button
mov a, button
cjne a,#1,int_00
clr IE0
mov switch, #0 ;RS=0 – команда
mov bte, #01111b ;установка адреса первого символа
lcall indic_wr ;второй строки
mov bte, #0C6h ;установка адреса первого символа
lcall indic_wr
mov TH2,#0FFh
mov TL2,#0B0h
mov RCAP2H, #0FFh
mov RCAP2L, #0B0h
setb EX0
setb tr2
mov hmks,#0
mov ms,#0
mov dms,#0
mov hms,#0
setb ET2 ;разрешение прерываний от таймера
reti

```

;ППОП таймера 2 (переполнение таймера, прошло 100 мкс)  
;максимальное время 1с



```

tim_2:
mov a, ms
cjne a, #6, next
exit:
clr TF2
reti
next: inc hmks ;десятые доли мс
mov a,hmks
cjne a,#10,exit ;не прошла 1 мс
mov hmks,#0
inc ms ;единицы мс
mov a,ms
cjne a,#10,exit ;не прошло 10 мс
mov ms,#0
inc dms ;десятки мс
mov a,dms
cjne a,#2,exit ;не прошло 100 мс
mov dms,#0
inc hms ;сотни мс
mov a,hms
cjne a,#10,exit ;не прошла 1 с
mov hms,#0

;данные располагаем в памяти программ
org 0FD0h
data: db 'E.D.Tegai'
;символов в последовательные ячейки памяти программ
; переход на бесконечный цикл
main_loop:

sjmp main_loop ; бесконечный цикл
;конец программы
end

```

## Результаты работы программы

При проверке работоспособности программы было замечено, что ФИО выводится в течение первых 3 мс, поэтому таймер запускался примерно в это время. Результаты работы программы продемонстрированы на рисунках 4 – 6 соответственно.

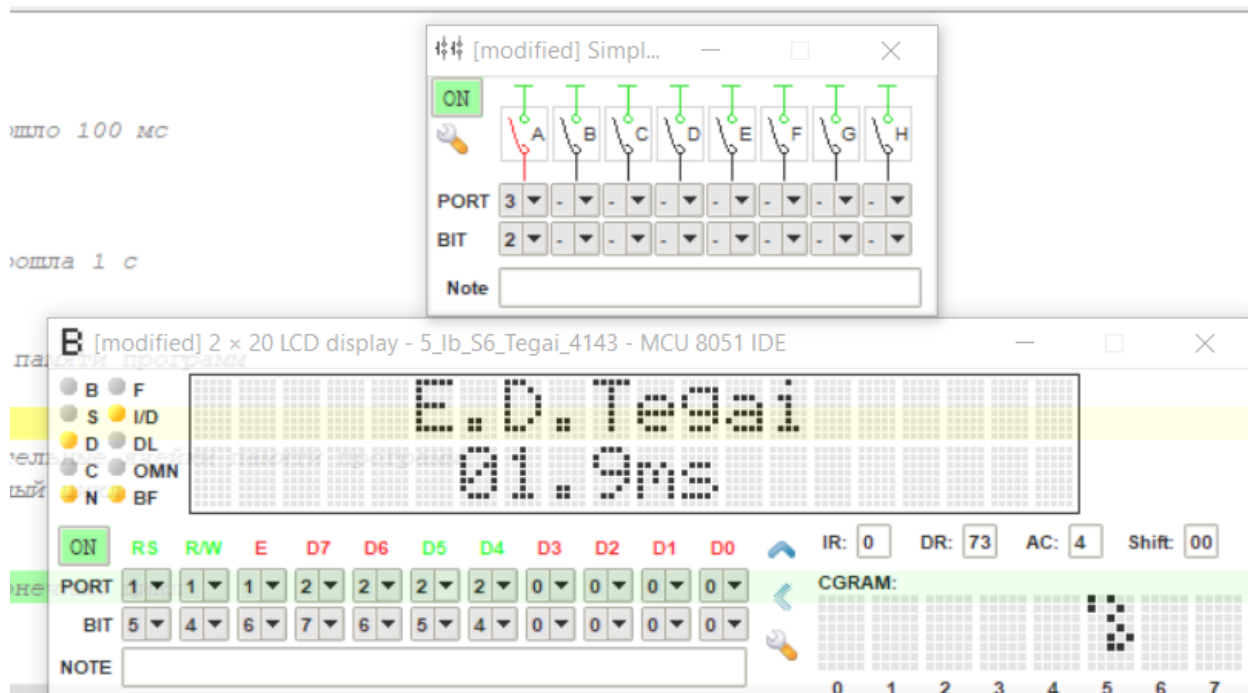


Рисунок 4 – Остановка таймера примерно спустя 2 мс

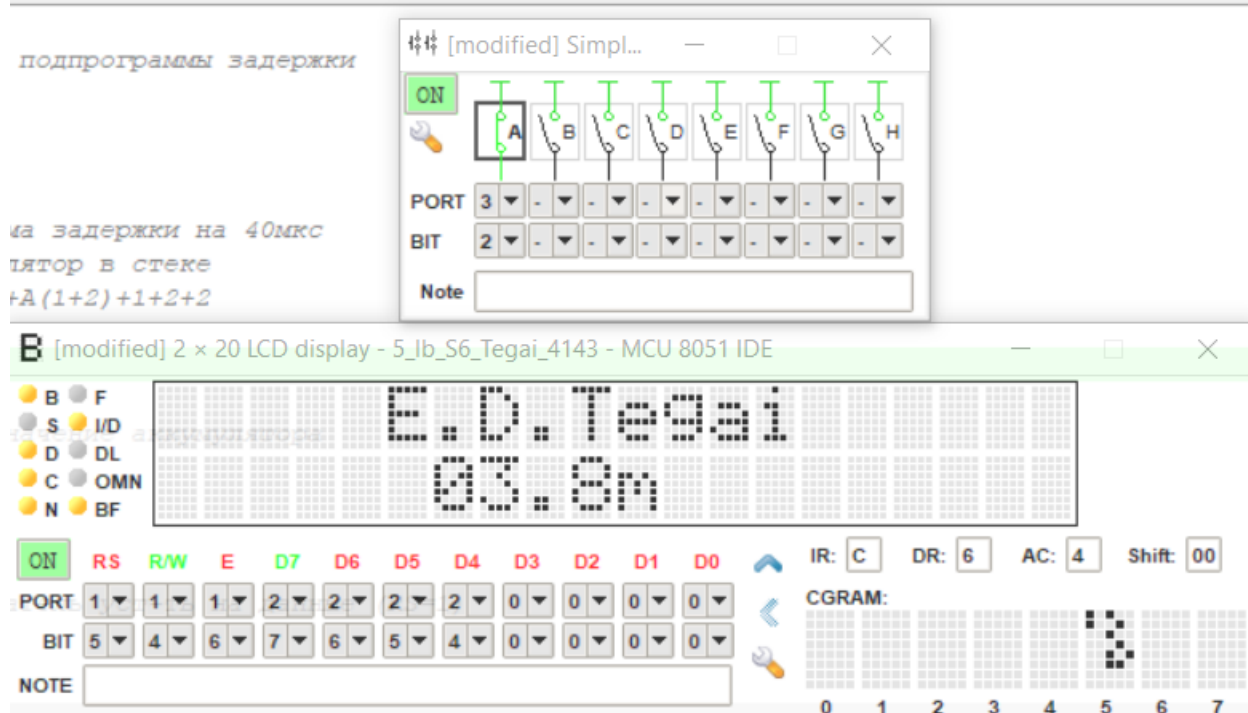


Рисунок 5 – Остановка таймера примерно спустя 3.8 мс

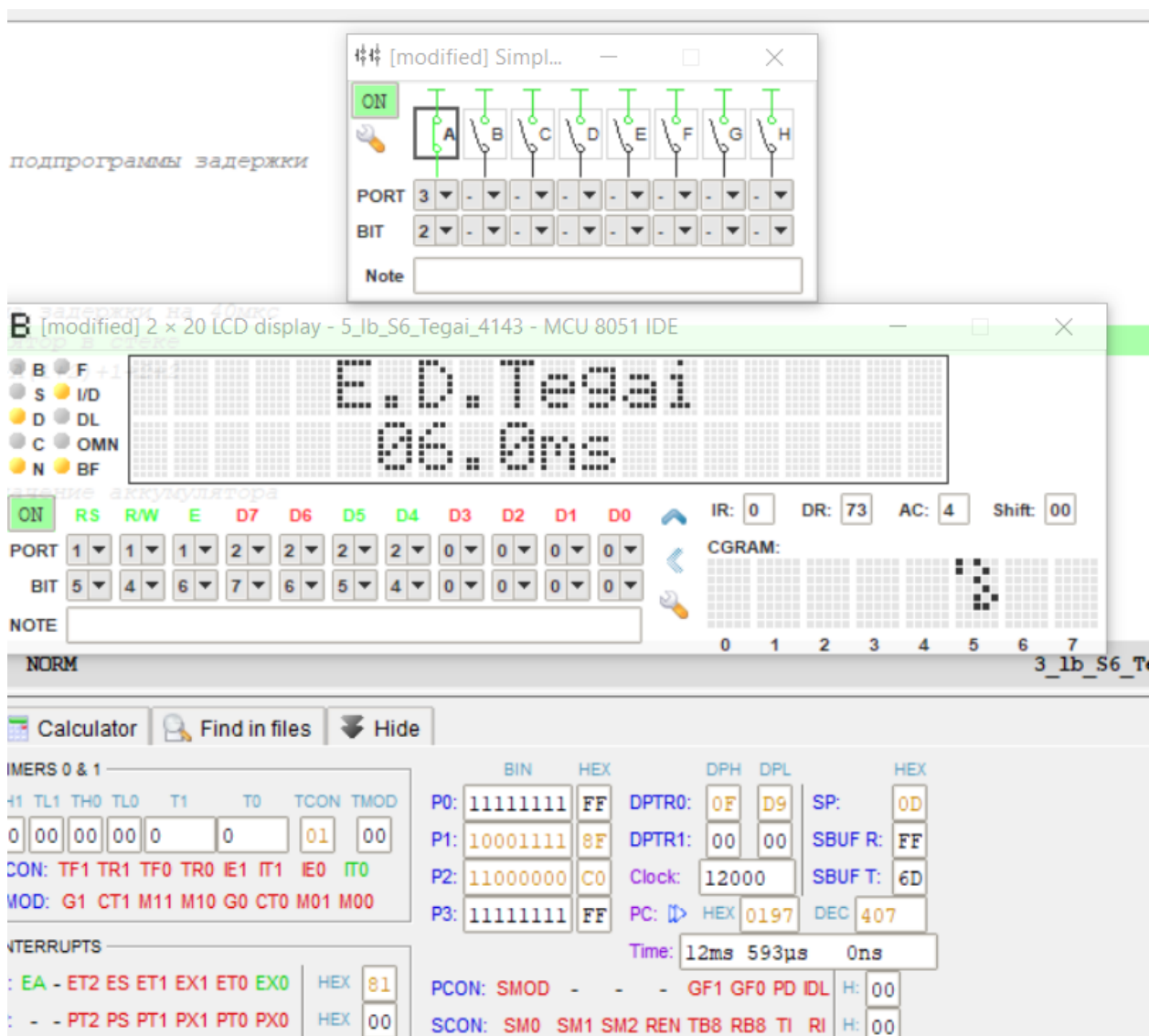


Рисунок 6 – Остановка таймера с выводом максимума

## Вывод

В результате выполнения работы были изучены принципы работы таймеров и системы прерываний микроконтроллера; приобретены навыки разработки микроконтроллерных систем, использующих таймеры.