

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

канд. техн. наук,

должность, уч. степень, звание

подпись, дата

Т.Н. Соловьева

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

РАЗРАБОТКА МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ТАЙМЕРОВ

по курсу: Микропроцессорные системы

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4143

подпись, дата

Д.В. Пономарев

инициалы, фамилия

Санкт-Петербург

2024

Вариант: 4

1. Формулировка задачи и метод ее решения

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

Номер варианта	Таймер	T_{\max} , мс	Старт	Стоп
1	0	20	INT0	INT0
2	1	8	INT0	INT1
3	2	15	INT1	INT0
4	0	20	INT1	INT1

Рисунок 1- Вариант №4

2. Задание по работе

Требуется разработать микроконтроллерную систему «Миллисекундомер», включающую в себя микроконтроллер семейства MCS-51, ЖКИ и одну кнопку.

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.

Листинг кода:

```
Filename: ex3.asm
; Date: 2024/02/25
; File Version: 1
; Author: Ponomarev D.V.
; Company: SUAI
; Description: example 3
; *
;*****
; 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
;*****
; Reset Vector
```

```

;*****
org 0h ; processor reset vector
ajmp start ; go to beginning of program
;*****
org 0013h ; processor reset vector
ajmp int_1
org 000bh ; processor interrupt vector
ajmp tim_0
; MAIN PROGRAM
;*****
org 100h
start:
;*****
lcall indic_init1

indic_init1:
;инициализация ЖКИ
indic_init: mov switch, #0;переключатель уст-ть на команду (RS=0)
mov bte, #38h ;байт – команда
lcall indic_wr ;вызов подпрограммы передачи в ЖКИ
mov bte, #0Fh ;активация всех знакомест
lcall indic_wr
mov bte, #06h ;режим автом. перемещения курсора
lcall indic_wr
mov bte, #83h ;установка адреса первого символа
lcall indic_wr
;Вывод строк
mov switch, #1 ;переключатель – данные (RS=1)
mov dptr, #0fd0h ;адрес, по которому расположены данные
;(см. конец программы)
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, #0DEh, indic_data_wr1
;пока не введены 19 символов 1ой строки
mov switch, #0 ;RS=0 – команда
mov bte, #0C8h ;установка адреса первого символа
lcall indic_wr ;второй строки
mov switch, #1 ;RS=1 - данные
;подпрограмма передачи в ЖКИ

mov TMOD,#00000010b ;таймер 0 в режиме 2 с разрешением
setb EX1 ;разрешение int1 ;////////////////////////

```

```
    setb EA
    jmp main_loop
```

```
indic_wr: mov p2, bte ;передаваемый байт – в P2
    setb p1.7 ;E:=1
    clr p1.6 ;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 p1.7 ;E:=0
    lcall indic_delay
    setb p1.7 ;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
    pop
    pop A ;восстанавливаем значение аккумулятора
    ret
```

```
int_11:
    clr ET0
    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, #1100b ;установка адреса первого символа
```

```

    lcall indic_wr ;второй строки

    clr EX1 ;для нового старта
    setb EX1
    reti

int_1:
clr EX1 ;запрет прерываний от int1
clr IE1
    ;lcall indic_delay
    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 TH0,#156 ;считаем сотни мкс
    mov TL0,#156
    mov hmks,#0
    mov ms,#0
    mov dms,#0
    setb ET0 ;разрешение прерываний от таймера
    mov TCON,#00010101b ;включаем таймер прерывания по фронту
    reti
    ;ППОП таймера 0 (переполнение таймера, прошло 100 мкс)
    ;максимальное время 1с
tim_0:
    mov a, dms
    cjne a, #10b, next
exit:  jb P3.3,int_11
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,#10,exit ;не прошло 100 мс
    mov dms,#0

```

;данные располагаем в памяти программ

org 0FD0h

data: db 'Ponomarev D.V.'

;символов в последовательные ячейки памяти программ

; переход на бесконечный цикл

main_loop:

 sjmp main_loop ; бесконечный цикл

;конец программы

end

3. Разработка программы

Результат работы программы представлен на рисунке 2-3

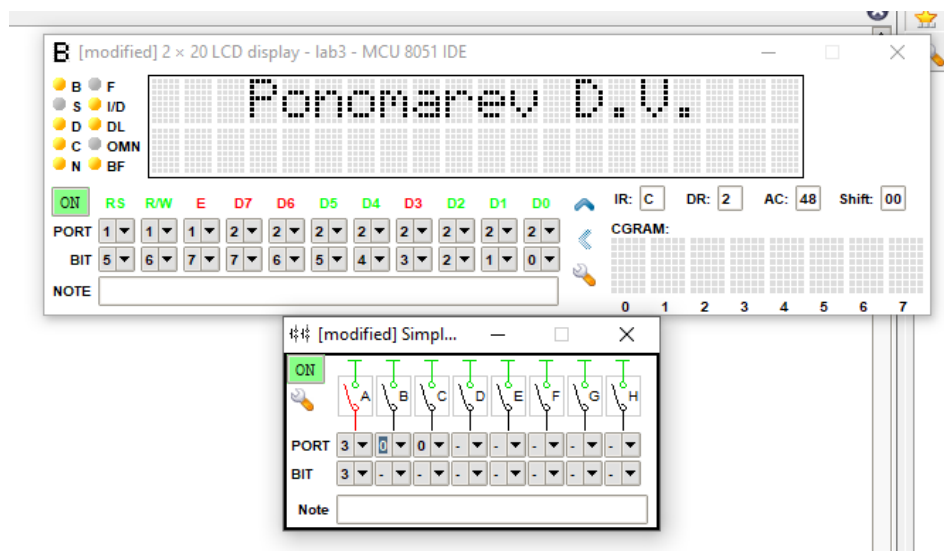


Рисунок 2- Работа программы, до нажатия

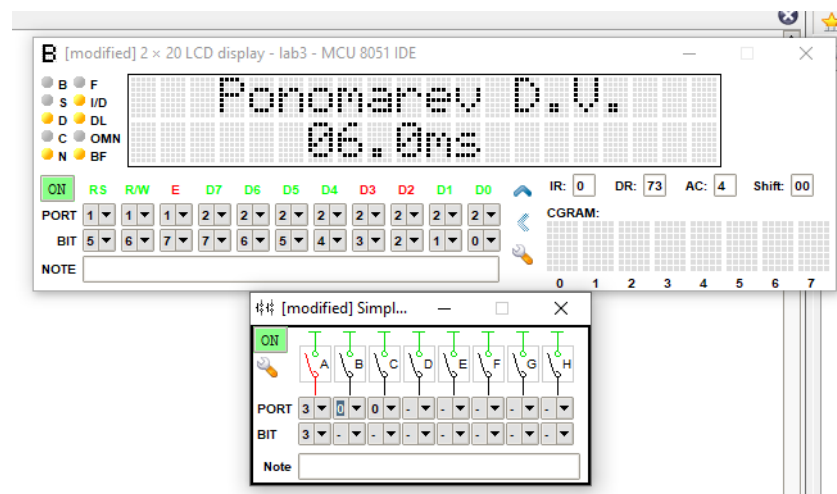


Рисунок 3- Работа программы, после нажатия бмс

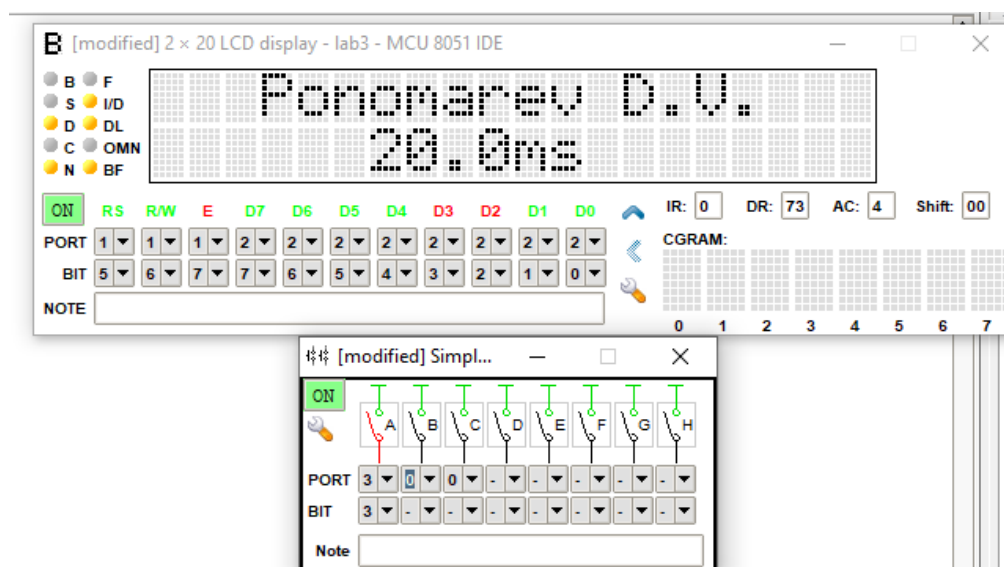


Рисунок 4- Работа программы, после нажатия (максимальная граница)

4. Вывод

Разработал микроконтроллерную систему «Миллисекундомер», включающую в себя микроконтроллер семейства MCS-51, ЖКИ и одну кнопку, а также таймер 0.