

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. В.Г.Шухова)**

Лабораторная работа №2
дисциплина «Программирование микроконтроллеров»
по теме «Написание программы управления сегментным индикатором»

Выполнил: студент группы ВТ-31
Проверил:

Макаров Д.С.
Гольцов Ю. А.

Белгород 2020

Лабораторная работа №2

«Написание программы управления сегментным индикатором»

Цель работы: изучить устройство таймеров микроконтроллеров архитектуры Intel 8051, а так же принцип управления сегментным индикатором.

Вариант 6

Задание: Разработать программу выводящую на сегментный индикатор часы, работающие на таймере.

Ход работы

Схема устройства

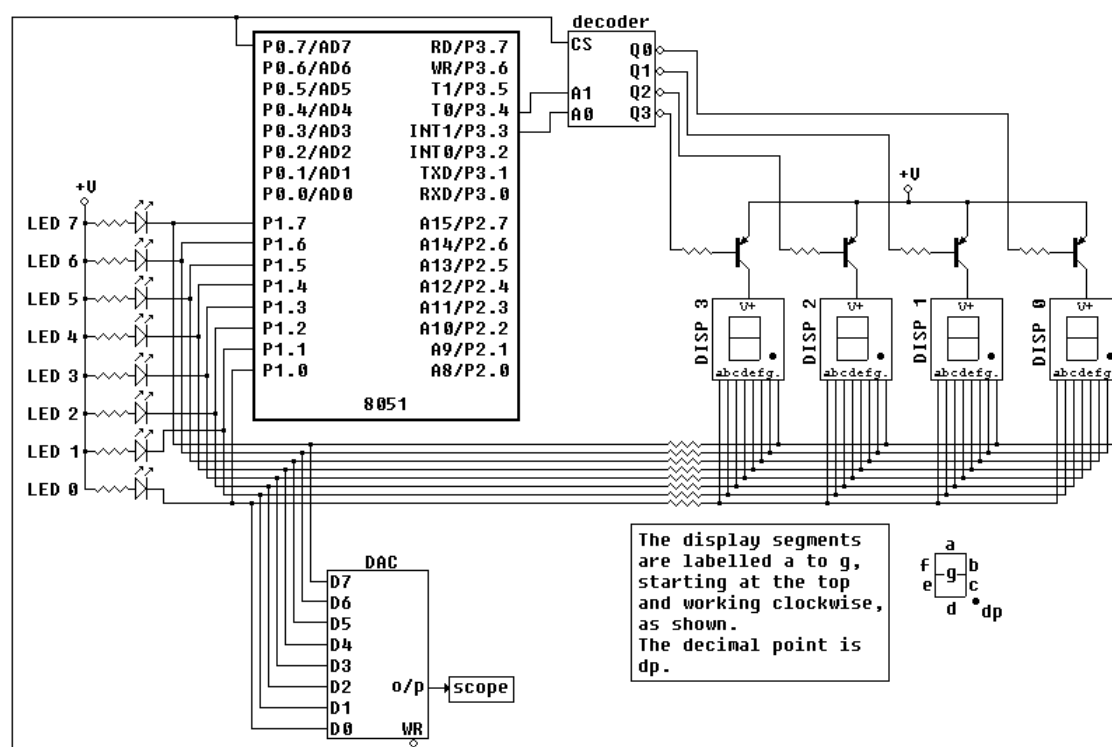


Рис. 1: Схема устройства

Блок схема алгоритма работы

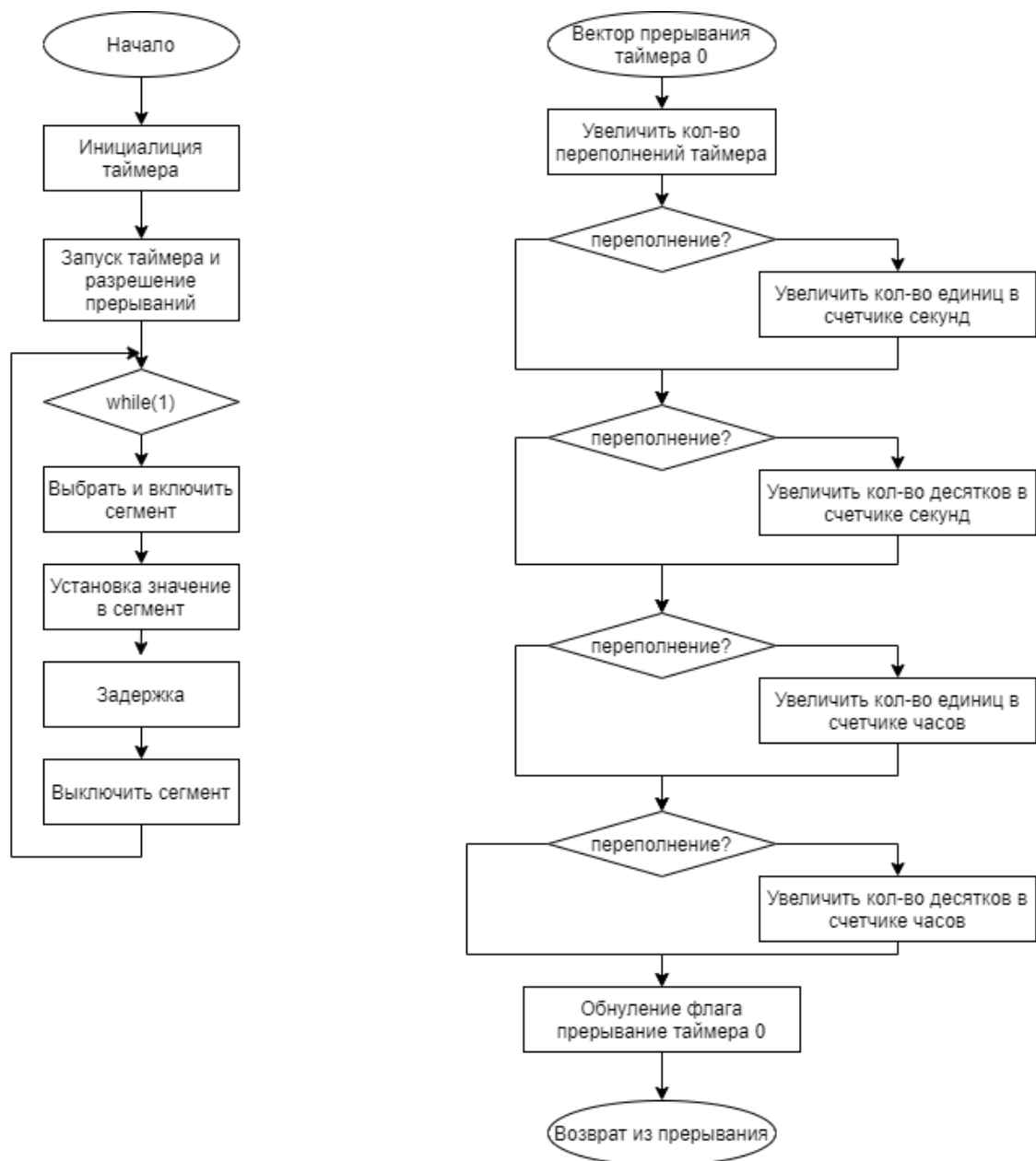


Рис. 2: Блок схема алгоритма работы

Приложение

Содержимое файла lab2.asm

```
org 00h
jmp start

org 0Bh
jmp timer0_interrupt

org 30h
start:
;глобальное разрешение прерываний и прерывание таймера 0
mov ie, #10000010b
;переключение таймера 0 в 1 режим
mov tmod, #1h
;запуск таймера 0
setb tr0
;r7 - количество переполнений таймера
;15 переполнений - 1 секунда
;r6 - счетчик секунд (0-59)
;r5 - счетчик единиц минут (0-9)
;r4 - счетчик десятков (0-6)
;r3 - счетчик единиц часов (0-9)
;r2 - счетчик десятков часов (0-2)
;r0 - флаг точки секунд
loop:
    ;сегмент 1
    clr p3.3
    clr p3.4
    mov a, r5
    call prep_seg
    call wait
    mov p1, #0FFh

    ;сегмент 2
    setb p3.3
    clr p3.4
    mov a, r4
    call prep_seg
    call wait
    mov p1, #0FFh

    ;сегмент 3
    clr p3.3
    setb p3.4
    mov a, r3
    call prep_seg
    call prep_dot
    call wait
    mov p1, #0FFh

    ;сегмент 4
    setb p3.3
    setb p3.4
    mov a, r2
    call prep_seg
    call wait
    mov p1, #0FFh
jmp loop
```

```

prep_dot:
    cjne r0,#0,reset_dot
        setb p1.7
        ret
    reset_dot:
        clr p1.7
        ret

prep_seg:
    cjne a,#9,not9
        mov p1,#090h
        ret
    not9:
    cjne a,#8,not8
        mov p1,#080h
        ret
    not8:
    cjne a,#7,not7
        mov p1,#0B8h
        ret
    not7:
    cjne a,#6,not6
        mov p1,#082h
        ret
    not6:
    cjne a,#5,not5
        mov p1,#092h
        ret
    not5:
    cjne a,#4,not4
        mov p1,#099h
        ret
    not4:
    cjne a,#3,not3
        mov p1,#0B0h
        ret
    not3:
    cjne a,#2,not2
        mov p1,#05Bh
        ret
    not2:
    cjne a,#1,not1
        mov p1,#0F9h
        ret
    not1:
    mov p1,#0C0h
    ret

timer0_interrupt:
    inc r7
    cjne r7,#1,to_sec
    ;cjne r7,#15,to_sec
    mov r7,#0h
    inc r6
    cjne r0,#0,not_dot
        inc r0
        jmp to_sec
    not_dot:

```

```

                                dec r0

to_sec:
cjne r6,#1,to_min0
;cnje r6,#59,to_min0
    mov r6,#0h
    inc r5
to_min0:
cjne r5,#9,to_min1
    mov r5,#0h
    inc r4
to_min1:
cjne r4,#6,to_hr0
    mov r4,0h
    inc r3
to_hr0:
cjne r3,#9,to_hr1
    mov r3,#0h
    inc r2
to_hr1:
cjne r2,#2,to_ret
    mov r2,0h
to_ret:
    clr tf0
    reti

wait:
mov b,#10
wait_loop:
    djnz b,wait_loop
ret

```