

**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 №3
дисциплина «Программирование микроконтроллеров»
по теме «Написание программы управления цифро-аналоговым
преобразователем»

Выполнил: студент группы ВТ-31
Проверил:

Макаров Д.С.
Гольцов Ю. А.

Белгород 2020

Лабораторная работа №3

«Написание программы управления цифро-аналоговым преобразователем»

Цель работы:изучить принцип работы ЦАП и написать программы для генерации сигналов различных форм.

Вариант 6

Задание: Разработать программы для генерации:

- прямоугольного сигнала.
- треугольного сигнала.
- гармонического сигнала

Разработать программу для аналогово-цифрового преобразования методом последовательных приближений.

Ход работы

Блок схема алгоритма работы

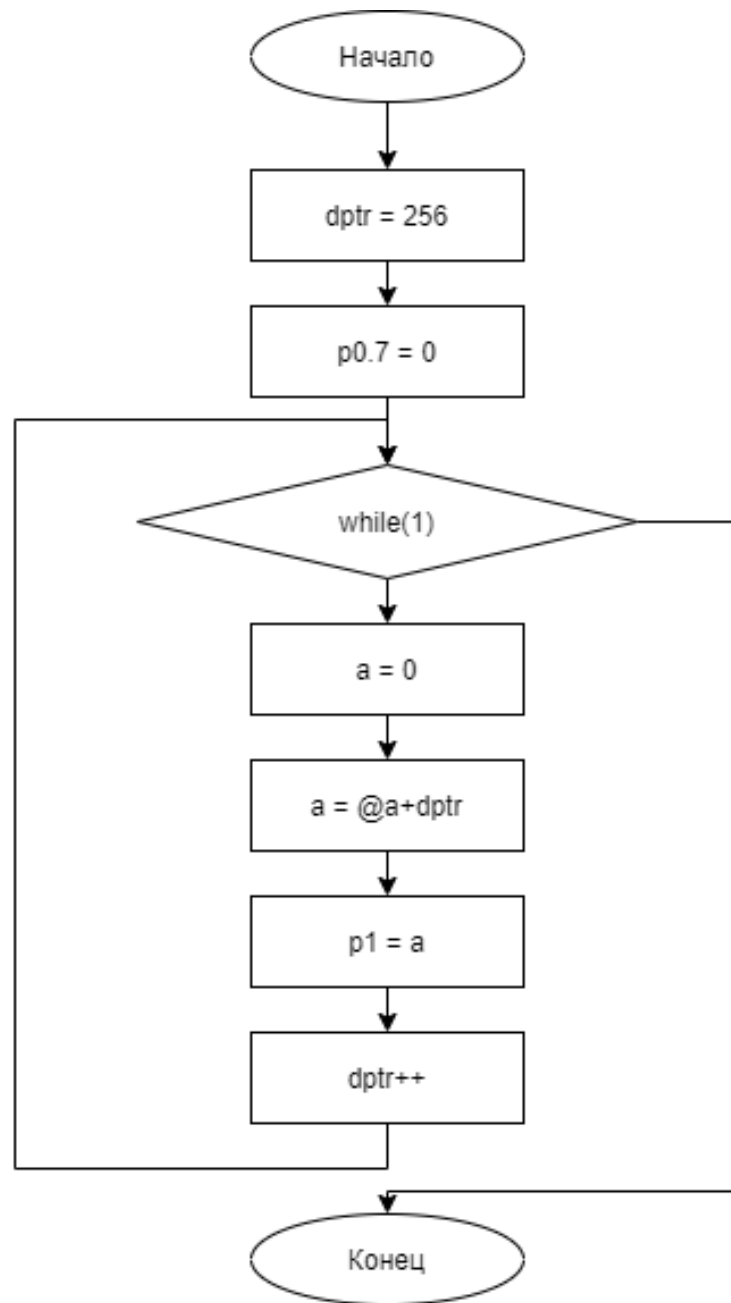


Рис. 1: Блок схема алгоритма вывода аналоговых сигналов



Рис. 2: Блок схема алгоритма реализующего метод последовательного приближения

Приложение

Содержимое файла lab3.asm

```
org 00h

; p1 - вход ЦАП
; p3.7 - выход компаратора

; настройка uart
clr sm0
setb sm1
mov a, pcon
setb acc.7
mov pcon, a
mov tmod, #20h
mov th1, #243
mov tl1, #243
setb tr1

start_conversion:
    clr p0.7
    mov p1, #0b

    check_bit7:
        setb p1.7
        jnb p3.7, reset_bit7
        jmp check_bit6
    reset_bit7:
        clr p1.7
        jmp check_bit6

    check_bit6:
        setb p1.6
        jnb p3.7, reset_bit6
        jmp check_bit5
    reset_bit6:
        clr p1.6
        jmp check_bit5

    check_bit5:
        setb p1.5
        jnb p3.7, reset_bit5
        jmp check_bit4
    reset_bit5:
        clr p1.5
        jmp check_bit4

    check_bit4:
        setb p1.4
        jnb p3.7, reset_bit4
        jmp check_bit3
    reset_bit4:
        clr p1.4
        jmp check_bit3

    check_bit3:
        setb p1.3
        jnb p3.7, reset_bit3
        jmp check_bit2
```

```

        reset_bit3:
            clr p1.3
            jmp check_bit2

check_bit2:
    setb p1.2
    jnb p3.7, reset_bit2
    jmp check_bit1
    reset_bit2:
        clr p1.2
        jmp check_bit1

check_bit1:
    setb p1.1
    jnb p3.7, reset_bit1
    jmp check_bit0
    reset_bit1:
        clr p1.1
        jmp check_bit0

check_bit0:
    setb p1.0
    jnb p3.7, reset_bit0
    jmp end_conversion
    reset_bit0:
        clr p1.0
        jmp end_conversion
    mov b, #1h
end_conversion:

call uart_out

mov b, #1
call wait
setb p0.7
jmp start_conversion

;Входные параметры:
;b -- кол-во циклов ожидания
wait:
    wait_loop:
        djnz b, wait_loop
    ret

;Выводит значение установленное на P1 в десятичной форме в UART
uart_out:
    mov R0, P1
    mov A, R0
    mov B, #10
    div AB
    mov R2, B ; разряд 3
    mov B, #10
    div AB
    mov R1, B ; разряд 2
    add A, #'0'
    mov SBUF, A ; разряд 1 передан
    jnb TI, $
    clr TI
    mov A, R1

```

```

add A, #'0'
mov SBUF, A ; разряд 2 передан
jnb TI, $
    clr TI
mov A, R2
add A, #'0' ; разряд 3 передан
mov SBUF, A
jnb TI, $ ; ожидание пока установится бит TI
    clr TI
mov SBUF, #'\r'
jnb TI, $
    clr TI
mov SBUF, #'\n'
jnb TI, $
    clr TI
ret

```

Содержимое файла lab3SIN.asm

```

org 00h

mov dph, #1
mov dpl, #0
clr p0.7

loop:
    mov a, #0
    movc a, @a+dptr
    mov p1, a
    inc dpl
    jmp loop

org 0100h
db 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h, 0dh, 01bh, 02ch, 041h, 059h, 072h,
→ 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh, 0f6h, 0ebh, 0dbb, 0c8h, 0b1h, 099h,
→ 07fh, 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h, 0dh, 01bh, 02ch, 041h, 059h,
→ 072h, 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh, 0f6h, 0ebh, 0dbb, 0c8h, 0b1h,
→ 099h, 07fh, 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h, 0dh, 01bh, 02ch, 041h,
→ 059h, 072h, 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh, 0f6h, 0ebh, 0dbb, 0c8h,
→ 0b1h, 099h, 07fh, 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h, 0dh, 01bh, 02ch,
→ 041h, 059h, 072h, 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh, 0f6h, 0ebh, 0dbb,
→ 0c8h, 0b1h, 099h, 07fh, 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h, 0dh, 01bh,
→ 02ch, 041h, 059h, 072h, 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh, 0f6h, 0ebh,
→ 0dbb, 0c8h, 0b1h, 099h, 07fh, 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h, 0dh,
→ 01bh, 02ch, 041h, 059h, 072h, 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh, 0f6h,
→ 0ebh, 0dbb, 0c8h, 0b1h, 099h, 07fh, 07fh, 065h, 04dh, 036h, 023h, 013h, 08h, 01h, 00h, 04h,
→ 0dh, 01bh, 02ch, 041h, 059h, 072h, 08ch, 0a5h, 0bdh, 0d2h, 0e3h, 0f1h, 0fah, 0feh, 0fdh,
→ 0f6h, 0ebh, 0dbb, 0c8h, 0b1h, 099h, 07fh

```

Содержимое файла lab3SQUARE.asm

```

org 00h

mov dph, #1
mov dpl, #0
clr p0.7

loop:
    mov a, #0

```

[illegible]

```
org 00h
```

```
loop:
```

```
org 0100h
```

7