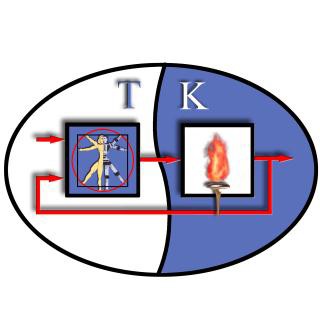
***МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ***



***ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ***

***ВЫСШЕГО ОБРАЗОВАНИЯ***

***«Белгородский Государственный Технологический Университет им. В.Г. Шухова» (БГТУ им. В.Г. Шухова)***

***ИЭИТУС***

***Кафедра Технической Кибернетики***

***Лабораторная работа № 3.1***

***дисциплина: «Микроконтроллеры в РТС»***

***тема: «Использование ЦАП»***

***Выполнил:***

*студент группы МР-41 Рощук Р.Д.*

***Принял:***

*ст. преп. кафедры ТК*

*Гольцов Ю.А.*

*Белгород 2020*

**Цель работы**

Изучить принципы работы с ЦАП в микроконтроллере 8051.

**Задание к работе**

Разработать программу для представления на экране осциллографа, входящего в состав EdSim8051DI, графиков синусоидальной, пилообразной и треугольной формы.

**Выполнение пилы**

В начальный момент выполняется настройка таймера 1 (8бит с автозагрузкой) и разрешается прерывание от таймера 1. Загружаем ноль в вывод P0.7, тем самым включая микросхему ЦАП. После этого идет бесконечный цикл ожидания.

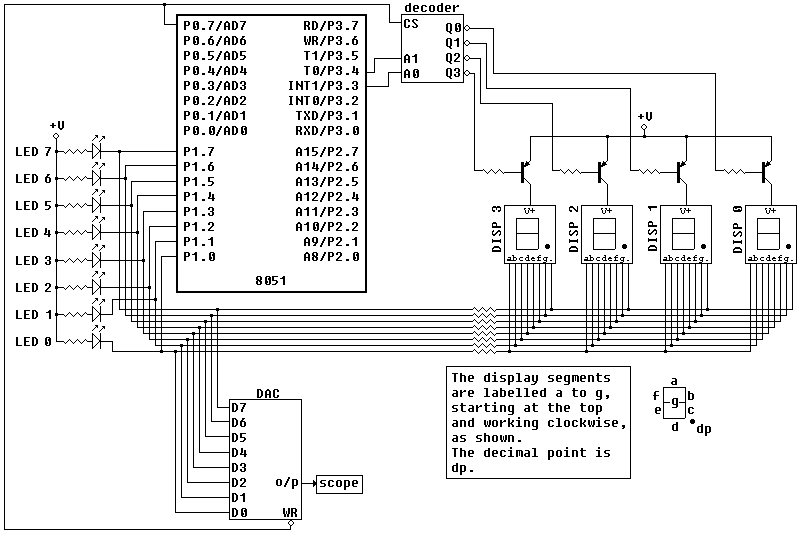


Рис 1. Схема

Срабатывание прерывания таймера 1 будет происходить раз в 6мкс:

2^8-250=6=6h

TH1=0FAh; TL1=0h

В самом прерывании мы загружаем в порт P1 данные из регистра А и затем прибавляем 1 к А. Тем самым мы при каждом вызове прерывания увеличиваем напряжение на выходе ЦАП на 5/256=0,0195 вольта. При переполнении регистра А у нас просто все начинается опять с А=0.

**Выполнение треугольника**

В основном теле программы все аналогично пиле, за исключением присвоения 66h высокого уровня. Этот прямо адресуемый бит будет определять направление движения напряжение на выводе ЦАП.

В прерывании мы выводим в Р1 данный из регистра А. Далее если 66h равен 0, то отнимаем из А единицу, а если равен 1, то прибавляем к А единицу. Затем проверяем не оказалось ли значения 0 или 255 в регистре А. Если оказалось, то надо поменять значение бита 66h чтобы поменять направление движения.

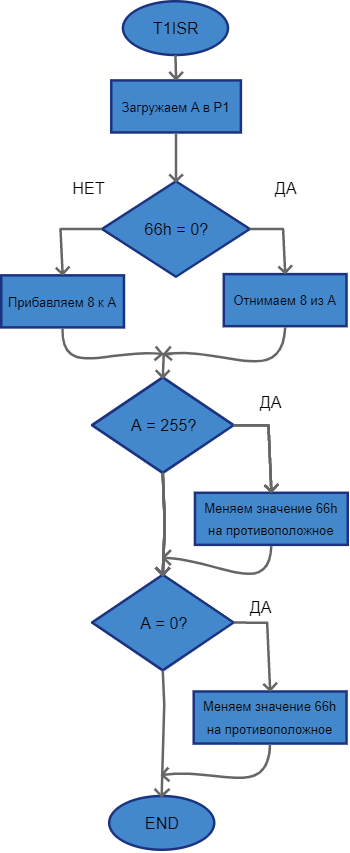


Рис 2. Блок схема треугольника

**Выполнение синуса**

В основном теле программы все аналогично пиле. В прерывании мы будем загружать в Р1 значение напряжения из таблицы синуса:



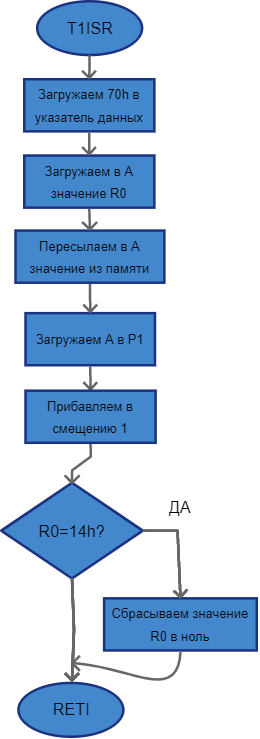
Для хранения мы будем использовать память микроконтроллера с 70h. Загрузка в регистр А осуществляется через указатель данных DPTR (MOVC A, @A + DPTR). В нем хранится значение 70h. Для того чтобы иметь доступ к разным элементам таблица надо задавать смещение в А до пересылки в него значения. Таблица состоит из 20 значений. При каждом вызове прерывания мы будем выводить следующее значение. Когда следующим окажется с смещением необходимо обнулить смещение. 

Рис 2. Блок схема Синуса

**Примеры работы программ**

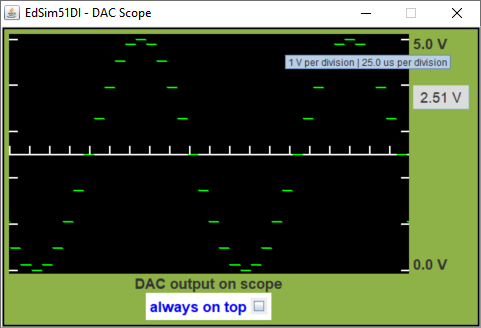


Рис 4. Синус

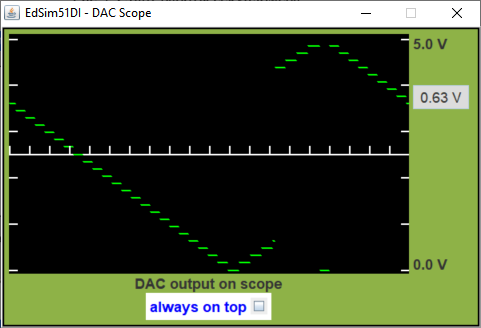


Рис 5. Треугольник

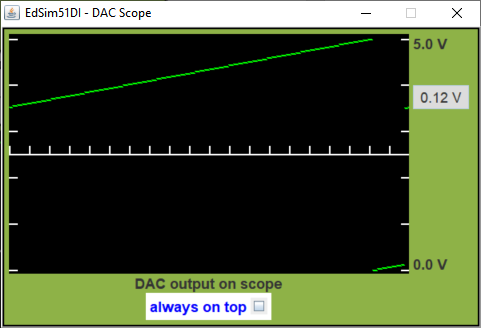


Рис 6. Пила

**Текст программы синус:**

ORG 0h

jmp main

t1ISR:

ORG 1Bh;Указываем где находится обработчик прерывания таймера 1

mov DPL,#70h

mov A, R0

movc A,@A+DPTR

MOV P1,A;Загружаем новое значение

inc R0

mov A, R0

CJNE A,#14h,vihod1 ;Сравнение аккумулятора с константой и переход, если не равно

mov R0,#0h

vihod1:

RETI;Выход из обработчика прерываний

SINUS:

ORG 70h

db 0h

db 6h

db 18h

db 35h

db 58h

db 80h

db 0A7h

db 0CAh

db 0E7h

db 0F9h

db 0FFh

db 0F9h

db 0E7h

db 0CAh

db 0A7h

db 80h

db 58h

db 35h

db 18h

db 6h

ORG 100h

main:

;T1 M1=1 M0=0 для переключения сегментов 8 бит

;T0 M1=0 M0=1 для подсчета времени/16бит

MOV TMOD,#00100001b

mov TH1,#0F3h

;разрешаем прерывание от таймера 1

SETB ET1

;разрешаем выполнение прерываний

SETB EA

CLR p0.7

SETB TR1;запускаем таймер 1

jmp $;бесконечный цикл ожидания

END

**Листинг программы синус:**

MCS-51 MACRO ASSEMBLER SIN 11/05/<0 PAGE 1

DOS 5.0 (038-N) MCS-51 MACRO ASSEMBLER, V2.3

OBJECT MODULE PLACED IN C:\SIM8052\SIN.OBJ

ASSEMBLER INVOKED BY: C:\SIM8052\ASM51.EXE C:\SIM8052\SIN.ASM

LOC OBJ LINE SOURCE

0000 1 ORG 0h

0000 020100 2 jmp main

3

4 t1ISR:

001B 5 ORG 1Bh;Sj`g{b`el cde m`undhrq^ nap`anrwhj opep{b`mh^ r`ilep` 1

001B 758270 6 mov DPL,#70h

001E E8 7 mov A, R0

001F 93 8 movc A,@A+DPTR

0020 F590 9 MOV P1,A;G`cpsf`el mnbne gm`wemhe

0022 08 10 inc R0

0023 E8 11 mov A, R0

0024 B41402 12 CJNE A,#14h,vihod1 ;Qp`bmemhe `jjslsk^rnp` q jnmqr`mrni h oepeund, eqkh me p`bmn

0027 7800 13 mov R0,#0h

14 vihod1:

0029 32 15 RETI;B{und hg nap`anrwhj` opep{b`mhi

16 SINUS:

0070 17 ORG 70h

0070 00 18 db 0h

0071 06 19 db 6h

0072 18 20 db 18h

0073 35 21 db 35h

0074 58 22 db 58h

0075 80 23 db 80h

0076 A7 24 db 0A7h

0077 CA 25 db 0CAh

0078 E7 26 db 0E7h

0079 F9 27 db 0F9h

007A FF 28 db 0FFh

007B F9 29 db 0F9h

007C E7 30 db 0E7h

007D CA 31 db 0CAh

007E A7 32 db 0A7h

007F 80 33 db 80h

0080 58 34 db 58h

0081 35 35 db 35h

0082 18 36 db 18h

0083 06 37 db 6h

0100 38 ORG 100h

39 main:

40 ;T1 M1=1 M0=0 dk^ oepejk~wemh^ qeclemrnb 8 ahr

41 ;T0 M1=0 M0=1 dk^ ondqwer` bpelemh/16ahr

0100 758921 42 MOV TMOD,#00100001b

0103 758DF3 43 mov TH1,#0F3h

44

45 ;p`gpex`el opep{b`mhe nr r`ilep` 1

0106 D2AB 46 SETB ET1

47 ;p`gpex`el b{onkmemhe opep{b`mhi

0108 D2AF 48 SETB EA

010A C287 49 CLR p0.7

010C D28E 50 SETB TR1;g`osqj`el r`ilep 1

MCS-51 MACRO ASSEMBLER SIN 11/05/<0 PAGE 2

LOC OBJ LINE SOURCE

010E 80FE 51 jmp $;aeqjnmewm{i vhjk nfhd`mh^

MCS-51 MACRO ASSEMBLER SIN 11/05/<0 PAGE 3

SYMBOL TABLE LISTING

------ ----- -------

N A M E T Y P E V A L U E A T T R I B U T E S

DPL. . . . D ADDR 0082H A

EA . . . . B ADDR 00A8H.7 A

ET1. . . . B ADDR 00A8H.3 A

MAIN . . . C ADDR 0100H A

P0 . . . . D ADDR 0080H A

P1 . . . . D ADDR 0090H A

SINUS. . . C ADDR 002AH A

T1ISR. . . C ADDR 0003H A

TH1. . . . D ADDR 008DH A

TMOD . . . D ADDR 0089H A

TR1. . . . B ADDR 0088H.6 A

VIHOD1 . . C ADDR 0029H A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND

**Текст программы треугольник:**

ORG 0h

jmp main

t1ISR:

ORG 1Bh;Указываем где находится обработчик прерывания таймера 1

MOV P1,A;Загружаем новое значение

jnb 66h, vniz;Переход, если бит равен нулю

add a,#8h

sjmp Vverh

vniz:

Subb a,#8h

Vverh:

CJNE A,#0FFh,vihod ;Сравнение аккумулятора с константой и переход, если не равно

CPL 66h

vihod:

CJNE A,#0h,vihod1 ;Сравнение аккумулятора с константой и переход, если не равно

CPL 66h

vihod1:

RETI;Выход из обработчика прерываний

ORG 155h

main:

;T1 M1=1 M0=0 для переключения сегментов 8 бит

;T0 M1=0 M0=1 для подсчета времени/16бит

MOV TMOD,#00100001b

mov TH1,#0F9h

;разрешаем прерывание от таймера 1

SETB ET1

;разрешаем выполнение прерываний

SETB EA

CLR p0.7

SETB TR1;запускаем таймер 1

SETB 66h

jmp $;бесконечный цикл ожидания

END

**Текст программы пила:**

ORG 0h

jmp main

t1ISR:

ORG 1Bh;Указываем где находится обработчик прерывания таймера 1

MOV P1,A;Загружаем новое значение

add a,#1h

RETI;Выход из обработчика прерываний

ORG 155h

main:

;T1 M1=1 M0=0 для переключения сегментов 8 бит

;T0 M1=0 M0=1 для подсчета времени/16бит

MOV TMOD,#00100001b

mov TH1,#0FAh

;разрешаем прерывание от таймера 1

SETB ET1

;разрешаем выполнение прерываний

SETB EA

CLR p0.7

SETB TR1;запускаем таймер 1

jmp $;бесконечный цикл ожидания

END