**Введение**

Развитие микроэлектроники и широкое применение её изделий в промышленном производстве, в устройствах и системах управления самыми разнообразными объектами и процессами является в настоящее время одним из основных направлений научно-технического прогресса.

Использование микроэлектронных средств в изделиях производственного и культурно-бытового назначения не только приводит к повышению технико-экономических показателей изделий (стоимости, надёжности, потребляемой мощности, габаритных размеров) и позволяет многократно сократить сроки разработки и отодвинуть сроки "морального старения" изделий, но и придаёт им принципиально новые потребительские качества (расширенные функциональные возможности, модифицируемость, адаптивность и т.д.).

За последние годы в микроэлектронике бурное развитие получило направление, связанное с выпуском однокристальных микроконтроллеров, которые предназначены для "интеллектуализации" оборудования различного назначения. Однокристальные микроконтроллеры представляют собой приборы, конструктивно выполненные в виде БИС и включающие в себя все оставшиеся части "голой" микро-ЭВМ: микропроцессор, память программ и память данных, а также программируемые интерфейсные схемы для связи с внешней средой. К настоящему времени более двух третей мирового рынка микропроцессорных средств составляют именно однокристальные микроконтроллеры.

В данном курсовом проекте необходимо разработать микропроцессорную систему, осуществляющую управление шаговым двигателем.

**1. Описание работы устройства**

Контроллер ШД, реализуемый на базе микроконтроллера серии МК51, представляет собой устройство позволяющее управлять ШД в шаговом режиме, режиме плавного ускорения и торможения.

МК51

Драйвер ШД

# Индикатор

# Клавиатура

Рис.1 Функциональная схема системы

В данное устройство входит клавиатура состоящая из 12 клавиш, индикатор и драйвер ЩД.

При включении питания системы, она начинает опрашивать клавиатуру. После ввода числового значения с клавиатуры система выводит его на индикатор и затем отрабатывает введенное число шагов путем выдачи последовательности импульсов с нарастающей, а затем убывающей частотой на драйвер ШД.

**2. Разработка принципиальной схемы**

Необходимо определить перечень устройств и элементов, которые будут входить в аппаратную часть.

При выборе микроконтроллера руководствуемся тем, что для нашей системы необходим наиболее простой контроллер на базе МК51, выбираем микроконтроллер AT89C51 с 4 килобайтами внутреннего ПЗУ фирмы Atmel, как наиболее подходящий по цене и возможностям.

Микроконтроллер семейства АТ89 фирмы Atmel представляет собой восьмиразрядную однокристальную микроЭВМ с системой команд MCS-51 фирмы Intel. Микроконтроллеры изготавливаются по КМОП (CMOS) технологии и имеют полностью статическую структуру.

Отличительные особенности:

* Совместимость с приборами семейства MCS-51
* Емкость перепрограммируемой Flash памяти: 4 Кбайт, 1000 циклов стирание/запись.
* Диапазон рабочих напряжений от 2,7 В до 6 В
* Полностью статический прибор - диапазон рабочих частот от 0 Гц до 24 Мгц
* Двухуровневая блокировка памяти программ
* ОЗУ емкостью 128 байтов
* 31 программируемых линий ввода/вывода
* 2 16-разрядных таймера/счетчика событий
* Пять источников сигнала прерывания
* Промышленный (-40°С...85°C) и коммерческий (0°C...70°C) диапазоны температур
* 40-выводные корпуса PDIP и SOIC

Описание:

КМОП микроконтроллер АТ89С51, оснащенный Flash программируемым и стираемым ПЗУ, совместим по системе команд и по выводам со стандартными приборами семейства MCS-51. Микроконтроллер содержит 4 Кбайта Flash ПЗУ, 128 байтов ОЗУ, 32 линии ввода/вывода, два 16-разрядных таймера/счетчика событий, полнодуплексный порт (UART), пять векторных двухуровневых прерываний, встроенный прецизионный аналоговый компаратор, встроенные генератор и схему формирования тактовой последовательности. Программирование Flash памяти программ ведется с использованием напряжения 12 В, ее содержимое может быть защищено от несанкционированных записи/считывания. Имеется возможность очистки Flash памяти за одну операцию, возможность считывания встроенного кода идентификации. Потребление в активном режиме на частоте 12 МГц не превышает 15 мА и 5,5 мА при напряжении питания 6 В и 3 В, соответственно.

Назначение выводов:

VCC: напряжение Питания.

GND: общий провод.

Порт 1: Восьми битный квази двунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов.

Порт 2: Восьми битный квази двунаправленный порт, аналогичный Р1; кроме того, выводы этого порта используются для выдачи адресной информации при обращении к внешней памяти программ или данных.

Порт 3: Восьми битный квази двунаправленный порт, аналогичный Р1; кроме того, выводы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, последовательного порта ввода-вывода, контроллера прерываний и внешней памяти программ и данных.

Вывод порта Альтернативная функция

P3.0 RXD (вход последовательного порта)

P3.1 TXD (выход последовательного порта)

P3.2 INT0 (внешнее прерывание)

P3.3 INT1 (внешнее прерывание)

P3.4 T0 (таймер 0 внешний ввод)

P3.5 T1 (таймер 1 внешний ввод)

Порт 0: Восьми битный квази двунаправленный порт ввода/вывода информации: при работе с внешними ПЗУ и ОЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных.

RST: вход сброса. На всех выводах ввода/вывода устанавливается сигнал логической 1, как только RST перейдет в состояние логической 1. Высокий логический уровень на входе RST должен удерживаться в течении двух машинных циклов для надежного сброса устройства.

XTAL1: вход инвертирующего усилителя тактового генератора и вход внешнего тактового сигнала.

XTAL2: выход инвертирующего усилителя генератора.



Рис. 2 Описание выводов.

Характеристики Генератора

XTAL1 и XTAL2 - вход и выход, соответственно, инвертирующего усилителя, который может быть настроен для использования как внутренний генератор, как показано в Рис. 3. Может использоваться кварцевый или керамический резонатор.

Примечание:

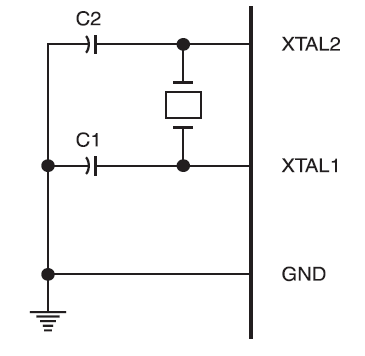


Рис. 3 Схема подключения резонатора

C1, C2 = 30 pF +/- 10 pF для кварцевых резонаторов,

40 pF +/- 10 pF для керамических резонаторов

Предельные допустимые параметры:

Рабочая температура ............................................................ -55°C +125°C

Температура хранения ......................................................... -65°C +150°C

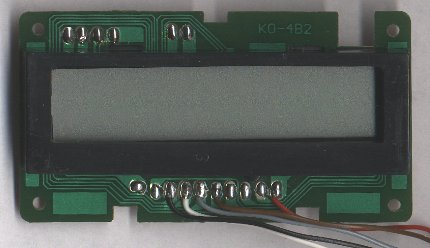
Напряжение на любом выводе относительно земли ............. -1.0В +7.0В

Максимальное напряжение питания ....................................... 6.6В

Выходной ток (лог 0) ..................................................................25.0 мАПодключаем вход RST через конденсатор на +5 В. В качестве конденсатора С6, выбираем конденсатор марки К50-6–50В–1мкФ10%. Для подключения осциллятора выбираем конденсаторы С7, С8 марки КМ-5Б-160В-30 пФ 10%, и кварц РК374 МД-6-ВС (11.059200 МГц).

Выбор индикатора:

Среди 7-сегментных ЖКИ – модулей наибольшее распространение получили модули на основе контроллера HT1611 (или HT1613). Они имеют 10 знакомест и управляются по последовательной шине.



Назначение выводов AL-801LCD

1 12/24 переключение формата времени

2 Vss общий

3 SK тактовая линия шины

4 DI линия данных шины

5 HK переключение часы/индикатор

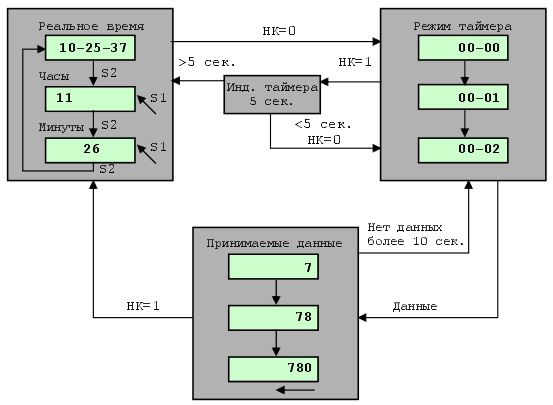
6 S1 установка времени

7 S2 выбор режима установки времени

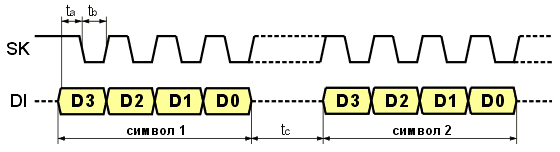
8 TMR сброс таймера

9 Vdd напряжение питания

Блок-схема алгоритма работы индикатора:

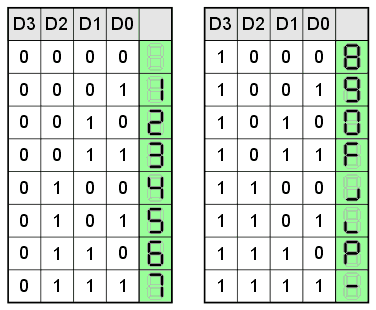


При использовании модуля в микроконтроллерной системе только для отображения загружаемых по последовательной шине символов, требуется соединить вывод HK с общим проводом, а выводы 12/24, S1, S2 и TMR оставить свободными. Временная диаграмма передачи данных по последовательной шине приведена на рисунке, где ta – время установки данных (>1 мкс), tb – время удержания данных (>2 мкс), tc – интервал между символами (>5 мкс).

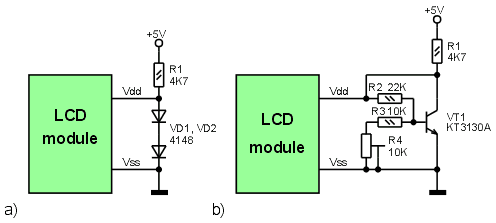


Данные подаются на линию DI и защелкиваются по спаду тактовых импульсов на линии SK. Символы отображаются в крайней правой позиции, уже имеющиеся на индикаторе символы сдвигаются влево. После того, как все необходимые данные переданы, линию SK следует оставить в состоянии низкого логического уровня, чтобы предотвратить автоматический переход модуля в режим отображения значения таймера.

Каждый символ кодируется 4-мя битами, поэтому всего имеется 16 символов.

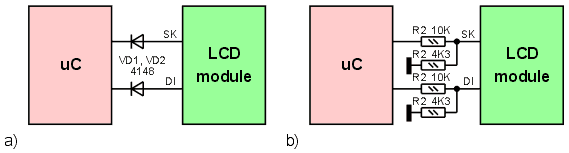


Нужно отметить, что напряжение питания индикатора сильно влияет на контрастность. При низком напряжении контрастность недостаточна, а при большом засвечиваются погашенные сегменты. Оптимум находится в промежутке 1.50 ... 1.65 В. Распространенная схема питания, где в качестве источника образцового напряжения используются диоды в прямом включении (рисунок a), не позволяет получить оптимальную контрастность, так как двух диодов оказывается мало, а трех – много. Тем более, желательно иметь возможность регулировки этого напряжения. Простая схема на одном транзисторе позволяет получить нужное напряжение питания и регулировать его (рисунок b).



Учитывая очень низкий ток потребления индикатора можно обойтись и простым резисторным делителем, если входное напряжение питания постоянно. Описанные схемы питания не являются экономичными и подходят, например, для устройств с сетевым питанием. Система питания автономного устройства может быть очень сложной, и конкретные решения зависят от специфики задачи. Одним из вариантов может быть питание устройства от элемента напряжением 1.5 В, от которого индикатор питается непосредственно. Микроконтроллерная часть устройства питается от того же элемента через повышающий DC-DC преобразователь.

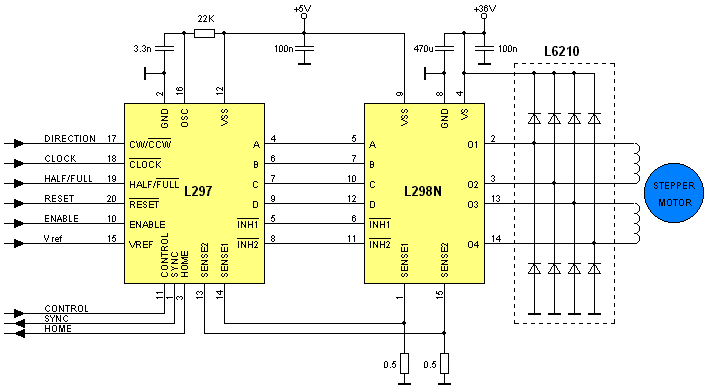
Для согласования логических уровней можно применить разные схемы. Учитывая тот факт, что входы DI и SK имеют внутренние подтягивающие резисторы, можно обойтись просто диодами (рисунок a). Преимущество такого способа заключается в том, что согласование не будет зависеть от напряжения питания микроконтроллера. Однако такой способ имеет и недостаток. Ввиду больших номиналов подтягивающих резисторов уровни на входах будут довольно медленно достигать состояния логической единицы, что потребует значительного снижения скорости обмена. Поэтому предпочтительнее для согласования использовать резисторные делители (рисунок b).



Необходимо отметить, что в течение примерно 2 сек после включения питания модуль не воспринимает данные, передаваемые ему по последовательной шине. Поэтому всегда должна быть задержка между включением питания и началом обмена.

Выбор драйвера ЩД:

Довольно популярной микросхемой, реализующей ШИМ-стабилизацию тока, является L297 фирмы SGS-Thomson. Совместно с микросхемой мостового драйвера L293 или L298 они образуют законченную систему управления для шагового двигателя.



Микросхема L297 сильно разгружает управляющий микроконтроллер, так как от него требуется только тактовая частота CLOCK (частота повторения шагов) и несколько статических сигналов: DIRECTION – направление (сигнал внутренне синхронизирован, переключать можно в любой момент), HALF/FULL – полушаговый/полношаговый режим, RESET – устанавливает фазы в исходное состояние (ABCD = 0101), ENABLE – разрешение работы микросхемы, V ref – опорное напряжение, которое задает пиковую величину тока при ШИМ-регулировании. Кроме того, имеется несколько дополнительных сигналов. Сигнал CONTROL задает режим работы ШИМ-регулятора. При его низком уровне ШИМ-регулирование происходит по выходам INH1, INH2, а при высоком – по выходам ABCD. SYNC – выход внутреннего тактового генератора ШИМ. Он служит для синхронизации работы нескольких микросхем. Также может быть использован как вход при тактировании от внешнего генератора. HOME – сигнал начального положения (ABCD = 0101). Он используется для синхронизации переключения режимов HALF/FULL. В зависимости от момента перехода в полношаговый режим микросхема может работать в режиме с одной включенной фазой или с двумя включенными фазами.

**3. Алгоритм**

Инициализация

Нажата кнопка

Нет

Уменьшить счетчик нажатий на 1

Записать число

в буфер

Вывести число

на индикатор

Нажат enter

Нет

Нажат

Step

Нет

Б

А

Да

Вычисление

перемещения

Выполнить

Freq\_plus и Feq\_minus

А

А

Установка единичного перемещения

Выполнить

Freq\_plus

А

Да

**4. Описание программы**

Данная программа состоит из основной программы 6 подпрограмм:

1. В основной программе мы производим опрос клавиатуры, путем загрузки в порт маски опроса по столбцам и проверяем появление логического нуля на выходах соответствующих строк.

keyboard1:

mov p1,#11101111b

jnb p1.3,n1

jnb p1.2,n4

jnb p1.1,n7

jnb p1.0,enter

mov p1,#11011111b

jnb p1.3,n2

jnb p1.2,n5

jnb p1.1,n8

jnb p1.0,n0

mov p1,#10111111b

jnb p1.3,n3

jnb p1.2,n6

jnb p1.1,n9

jnb p1.0,step

jmp keyboard

1. В процедурах обработки нажатия кнопок мы записываем в буфер набранные числа и выводим их на индикатор. Так же мы оперируем счетчиком нажатий, ограничивая количество набранных цифр до 5.

n1: cjne r4,#4,incr

xch A,r4

jz keyboard

dec A

xch A,r4

mov r2,#1

call LCD

mov r1,#indikator

mov @r1,#1

jmp keyboard

1. Процедура формирования задания на двигатель производится в обработчике кнопки enter. В начале массив из цифр в буфере преобразуется в число хранящееся в паре регистров, путем последовательного умножения в циклах введенной цифры на 10 столько раз, сколько разрядов имеет эта цифра. И последовательное сложение полученных результатов. Далее получены результат делится на 2 чтобы получить треугольную форму задания. Далее вызываются подпрограммы управления частотой.

enter1: push 0h

push 2h

push 6h

push 7h

mov r0,#indikator

mov A,@r0

n10000: mov r2,4

mov B,#10

mul AB

djnz r2,n10000

mov r6,A

mov r7,B

n1000: mov r2,3

mov B,#10

inc r0

mov A,@r0

mul AB

djnz r2,n1000

add A,r6

mov r6,A

mov A,B

add A,r7

mov r7,A

n100: mov r2,2

mov B,#10

inc r0

mov A,@r0

mul AB

djnz r2,n100

add A,r6

mov r6,A

mov A,B

add A,r7

mov r7,A

n10: mov r2,1

mov B,#10

inc r0

mov A,@r0

mul AB

djnz r2,n10

add A,r6

mov r6,A

mov A,B

add A,r7

mov r7,A

inc r0

mov A,@r0

add A,r6

mov r6,A

mov A,r7

mov B,2

div AB

mov r7,B

mov A,r6

mov B,2

div AB

mov r6,B

call freq\_plus

call freq\_minus

pop 0h

pop 2h

pop 6h

pop 7h

ret

1. Процедура вывода на экран.

LCD: push 0h

push 1h

push 2h

mov r1,#4

mov A,r0

swap A

mov r0,A

LDL: mov A,r0

rlc A

jc LD1

LD0: clr p2.0

jmp STR

LD1: setb p2.0

STR: setb p2.1

mov r2,#5

HNG1: djnz r2,HNG1

clr p2.1

mov r2,#10

HNG2: djnz r2,HNG2

djnz r1,LDL

mov r2,#25

HNG3: djnz r2,HNG3

pop 0h

pop 1h

pop 2h

ret

1. Процедура нарастания частоты. Суть данной процедуры состоит в том чтобы установить на ножке порта Р0.0 логическую единицу и держать ее весь промежуток работы таймера. Эта операция выполняется с последующим инкрементированием таймера, что позволяет получить нам нарастающую частоту изменения состояния ножки порта. Так же в подпрограмму встроен счетчик импульсов чтобы отрабатывать задание, получаемое при нажатии кнопки enter. Так же имеется ограничитель максимальной частоты.

freq\_plus: push 0h

push 1h

push 6h

push 7h

mov r0,#0h

mov r1,#0ffh

inc r7

main: call pulse

djnz r6,main

djnz r7,main

jmp en

pulse: setb P0.0

call timer

clr P0.0

call test

ret

timer: mov tmod,#1

mov th0,r0

mov tl0,r1

setb tr0

wait: jbc tf0,ok

sjmp wait

ok: clr tr0

ret

test: cjne r0,#0ffh,dal

ret

dal: inc r0

ret

en: mov A,r0

mov r5,A

pop 0h

pop 1h

pop 6h

pop 7h

ret

1. Процедура уменьшения частоты. Работает аналогично процедуре увеличения частоты с тем лишь отличием, что в данной процедуре значение таймера декрементируется. Так же эта подпрограмма начинает работу именно с той частоты, на которой закончила работу подпрограмма увеличения.

freq\_minus: push 0h

push 1h

push 6h

push 7h

mov A,r5

mov r0,A

mov r1,#0ffh

inc r7

mainm: call pulse

djnz r6,mainm

djnz r7,mainm

jmp en

pulsem: setb P0.0

call timerm

clr P0.0

call testm

ret

timerm: mov tmod,#1

mov th0,r0

mov tl0,r1

setb tr0

waitm: jbc tf0,okm

sjmp waitm

okm: clr tr0

ret

testm: cjne r0,#0h,dalm

ret

dalm: dec r0

ret

enm: pop 0h

pop 1h

pop 6h

pop 7h

ret

**5. Листинг**

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 1

MACRO ASSEMBLER A51 V8.00b

OBJECT MODULE PLACED IN kurs.OBJ

ASSEMBLER INVOKED BY: F:\Keil\C51\BIN\A51.EXE kurs.a51 SET(SMALL) DEBUG EP

LOC OBJ LINE SOURCE

0000 1 org 0

0000 8000 2 jmp init

3

0050 4 temp EQU 50H

0051 5 Cnt EQU 51H

0052 6 Del EQU 52H

0002 7 indikator: db

8

0002 D281 9 init: setb p0.1

0004 D282 10 setb p0.2

0006 C283 11 clr p0.3

0008 7C05 12 mov r4,#5

13

000A 14 keyboard1:

000A 7590EF 15 mov p1,#11101111b

000D 309329 16 jnb p1.3,n1

0010 30925C 17 jnb p1.2,n4

0013 30917F 18 jnb p1.1,n7

0016 30907A 19 jnb p1.0,enter

20

0019 7590DF 21 mov p1,#11011111b

001C 30932C 22 jnb p1.3,n2

001F 30925F 23 jnb p1.2,n5

0022 309172 24 jnb p1.1,n8

0025 309075 25 jnb p1.0,n0

26

0028 7590BF 27 mov p1,#10111111b

002B 30932F 28 jnb p1.3,n3

002E 309272 29 jnb p1.2,n6

0031 30916B 30 jnb p1.1,n9

0034 30906A 31 jnb p1.0,step

32

0037 8060 33 jmp keyboard

34

0039 BC045F 35 n1: cjne r4,#4,incr

003C CC 36 xch A,r4

003D 605A 37 jz keyboard

003F 14 38 dec A

0040 CC 39 xch A,r4

0041 7A01 40 mov r2,#1

0043 31AB 41 call LCD

0045 7902 42 mov r1,#indikator

0047 7701 43 mov @r1,#1

0049 804E 44 jmp keyboard

004B BC044D 45 n2: cjne r4,#4,incr

004E CC 46 xch A,r4

004F 6048 47 jz keyboard

0051 14 48 dec A

0052 CC 49 xch A,r4

0053 7A02 50 mov r2,#2

0055 31AB 51 call LCD

0057 7902 52 mov r1,#indikator

0059 7702 53 mov @r1,#2

005B 803C 54 jmp keyboard

005D BC043B 55 n3: cjne r4,#4,incr

0060 CC 56 xch A,r4

0061 6036 57 jz keyboard

0063 14 58 dec A

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 2

0064 CC 59 xch A,r4

0065 7A03 60 mov r2,#3

0067 31AB 61 call LCD

0069 7902 62 mov r1,#indikator

006B 7703 63 mov @r1,#3

006D 802A 64 jmp keyboard

006F BC0429 65 n4: cjne r4,#4,incr

0072 CC 66 xch A,r4

0073 6024 67 jz keyboard

0075 14 68 dec A

0076 CC 69 xch A,r4

0077 7A04 70 mov r2,#4

0079 31AB 71 call LCD

007B 7902 72 mov r1,#indikator

007D 7704 73 mov @r1,#4

007F 8018 74 jmp keyboard

0081 BC0417 75 n5: cjne r4,#4,incr

0084 CC 76 xch A,r4

0085 6012 77 jz keyboard

0087 14 78 dec A

0088 CC 79 xch A,r4

0089 7A05 80 mov r2,#5

008B 31AB 81 call LCD

008D 7902 82 mov r1,#indikator

008F 7705 83 mov @r1,#5

0091 8006 84 jmp keyboard

85

0093 8073 86 enter: jmp enter1

0095 801E 87 n7: jmp n71

0097 802E 88 n8: jmp n81

0099 010A 89 keyboard: jmp keyboard1

009B 8060 90 incr: jmp incr1

009D 804C 91 n0: jmp n01

009F 8038 92 n9: jmp n91

00A1 805D 93 step: jmp step1

94

00A3 BC04F5 95 n6: cjne r4,#4,incr

00A6 CC 96 xch A,r4

00A7 60F0 97 jz keyboard

00A9 14 98 dec A

00AA CC 99 xch A,r4

00AB 7A06 100 mov r2,#6

00AD 31AB 101 call LCD

00AF 7902 102 mov r1,#indikator

00B1 7706 103 mov @r1,#6

00B3 80E4 104 jmp keyboard

00B5 BC04E3 105 n71: cjne r4,#4,incr

00B8 CC 106 xch A,r4

00B9 60DE 107 jz keyboard

00BB 14 108 dec A

00BC CC 109 xch A,r4

00BD 7A07 110 mov r2,#7

00BF 31AB 111 call LCD

00C1 7902 112 mov r1,#indikator

00C3 7707 113 mov @r1,#7

00C5 80D2 114 jmp keyboard

00C7 BC04D1 115 n81: cjne r4,#4,incr

00CA CC 116 xch A,r4

00CB 60CC 117 jz keyboard

00CD 14 118 dec A

00CE CC 119 xch A,r4

00CF 7A08 120 mov r2,#8

00D1 31AB 121 call LCD

00D3 7902 122 mov r1,#indikator

00D5 7708 123 mov @r1,#8

00D7 80C0 124 jmp keyboard

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 3

00D9 BC04BF 125 n91: cjne r4,#4,incr

00DC CC 126 xch A,r4

00DD 60BA 127 jz keyboard

00DF 14 128 dec A

00E0 CC 129 xch A,r4

00E1 7A09 130 mov r2,#9

00E3 31AB 131 call LCD

00E5 7902 132 mov r1,#indikator

00E7 7709 133 mov @r1,#9

00E9 80AE 134 jmp keyboard

00EB BC04AD 135 n01: cjne r4,#4,incr

00EE CC 136 xch A,r4

00EF 60A8 137 jz keyboard

00F1 14 138 dec A

00F2 CC 139 xch A,r4

00F3 7A00 140 mov r2,#0

00F5 31AB 141 call LCD

00F7 7902 142 mov r1,#indikator

00F9 7700 143 mov @r1,#0

00FB 809C 144 jmp keyboard

00FD 08 145 incr1: inc r0

00FE 8099 146 jmp keyboard

0100 7E01 147 step1: mov r6,#1

0102 7F00 148 mov r7,#0

0104 31D9 149 call freq\_plus

0106 8091 150 jmp keyboard

151

0108 C000 152 enter1: push 0h

010A C002 153 push 2h

010C C006 154 push 6h

010E C007 155 push 7h

0110 7802 156 mov r0,#indikator

0112 E6 157 mov A,@r0

0113 AA04 158 n10000: mov r2,4

0115 75F00A 159 mov B,#10

0118 A4 160 mul AB

0119 DAF8 161 djnz r2,n10000

011B FE 162 mov r6,A

011C AFF0 163 mov r7,B

011E AA03 164 n1000: mov r2,3

0120 75F00A 165 mov B,#10

0123 08 166 inc r0

0124 E6 167 mov A,@r0

0125 A4 168 mul AB

0126 DAF6 169 djnz r2,n1000

0128 2E 170 add A,r6

0129 FE 171 mov r6,A

012A E5F0 172 mov A,B

012C 2F 173 add A,r7

012D FF 174 mov r7,A

012E AA02 175 n100: mov r2,2

0130 75F00A 176 mov B,#10

0133 08 177 inc r0

0134 E6 178 mov A,@r0

0135 A4 179 mul AB

0136 DAF6 180 djnz r2,n100

0138 2E 181 add A,r6

0139 FE 182 mov r6,A

013A E5F0 183 mov A,B

013C 2F 184 add A,r7

013D FF 185 mov r7,A

013E AA01 186 n10: mov r2,1

0140 75F00A 187 mov B,#10

0143 08 188 inc r0

0144 E6 189 mov A,@r0

0145 A4 190 mul AB

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 4

0146 DAF6 191 djnz r2,n10

0148 2E 192 add A,r6

0149 FE 193 mov r6,A

014A E5F0 194 mov A,B

014C 2F 195 add A,r7

014D FF 196 mov r7,A

014E 08 197 inc r0

014F E6 198 mov A,@r0

0150 2E 199 add A,r6

0151 FE 200 mov r6,A

201

0152 EF 202 mov A,r7

0153 8502F0 203 mov B,2

0156 84 204 div AB

0157 AFF0 205 mov r7,B

0159 EE 206 mov A,r6

015A 8502F0 207 mov B,2

015D 84 208 div AB

015E AEF0 209 mov r6,B

210

0160 31D9 211 call freq\_plus

0162 316D 212 call freq\_minus

213

0164 D000 214 pop 0h

0166 D002 215 pop 2h

0168 D006 216 pop 6h

016A D007 217 pop 7h

016C 22 218 ret

219

220

016D C000 221 freq\_minus: push 0h

016F C001 222 push 1h

0171 C006 223 push 6h

0173 C007 224 push 7h

0175 ED 225 mov A,r5

0176 F8 226 mov r0,A

0177 79FF 227 mov r1,#0ffh

0179 0F 228 inc r7

017A 31EE 229 mainm: call pulse

017C DEFC 230 djnz r6,mainm

017E DFFA 231 djnz r7,mainm

0180 410E 232 jmp en

0182 D280 233 pulsem: setb P0.0

0184 318B 234 call timerm

0186 C280 235 clr P0.0

0188 319C 236 call testm

018A 22 237 ret

018B 758901 238 timerm: mov tmod,#1

018E 888C 239 mov th0,r0

0190 898A 240 mov tl0,r1

0192 D28C 241 setb tr0

0194 108D02 242 waitm: jbc tf0,okm

0197 80FB 243 sjmp waitm

0199 C28C 244 okm: clr tr0

019B 22 245 ret

019C B80001 246 testm: cjne r0,#0h,dalm

019F 22 247 ret

01A0 18 248 dalm: dec r0

01A1 22 249 ret

01A2 D000 250 enm: pop 0h

01A4 D001 251 pop 1h

01A6 D006 252 pop 6h

01A8 D007 253 pop 7h

01AA 22 254 ret

255

01AB C000 256 LCD: push 0h

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 5

01AD C001 257 push 1h

01AF C002 258 push 2h

01B1 7904 259 mov r1,#4

01B3 E8 260 mov A,r0

01B4 C4 261 swap A

01B5 F8 262 mov r0,A

01B6 E8 263 LDL: mov A,r0

01B7 33 264 rlc A

01B8 4004 265 jc LD1

01BA C2A0 266 LD0: clr p2.0

01BC 8002 267 jmp STR

01BE D2A0 268 LD1: setb p2.0

01C0 D2A1 269 STR: setb p2.1

01C2 7A05 270 mov r2,#5

01C4 DAFE 271 HNG1: djnz r2,HNG1

01C6 C2A1 272 clr p2.1

01C8 7A0A 273 mov r2,#10

01CA DAFE 274 HNG2: djnz r2,HNG2

01CC D9E8 275 djnz r1,LDL

01CE 7A19 276 mov r2,#25

01D0 DAFE 277 HNG3: djnz r2,HNG3

01D2 D000 278 pop 0h

01D4 D001 279 pop 1h

01D6 D002 280 pop 2h

01D8 22 281 ret

282

01D9 C000 283 freq\_plus: push 0h

01DB C001 284 push 1h

01DD C006 285 push 6h

01DF C007 286 push 7h

01E1 7800 287 mov r0,#0h

01E3 79FF 288 mov r1,#0ffh

01E5 0F 289 inc r7

01E6 31EE 290 main: call pulse

01E8 DEFC 291 djnz r6,main

01EA DFFA 292 djnz r7,main

01EC 8020 293 jmp en

294

01EE D280 295 pulse: setb P0.0

01F0 31F7 296 call timer

01F2 C280 297 clr P0.0

01F4 5108 298 call test

01F6 22 299 ret

01F7 758901 300 timer: mov tmod,#1

01FA 888C 301 mov th0,r0

01FC 898A 302 mov tl0,r1

01FE D28C 303 setb tr0

0200 108D02 304 wait: jbc tf0,ok

0203 80FB 305 sjmp wait

0205 C28C 306 ok: clr tr0

0207 22 307 ret

0208 B8FF01 308 test: cjne r0,#0ffh,dal

020B 22 309 ret

020C 08 310 dal: inc r0

020D 22 311 ret

020E E8 312 en: mov A,r0

020F FD 313 mov r5,A

0210 D000 314 pop 0h

0212 D001 315 pop 1h

0214 D006 316 pop 6h

0216 D007 317 pop 7h

0218 22 318 ret

319

320 END

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 6

SYMBOL TABLE LISTING

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

N A M E T Y P E V A L U E ATTRIBUTES

B. . . . . . . . . D ADDR 00F0H A

CNT. . . . . . . . N NUMB 0051H A

DAL. . . . . . . . C ADDR 020CH A

DALM . . . . . . . C ADDR 01A0H A

DEL. . . . . . . . N NUMB 0052H A

EN . . . . . . . . C ADDR 020EH A

ENM. . . . . . . . C ADDR 01A2H A

ENTER. . . . . . . C ADDR 0093H A

ENTER1 . . . . . . C ADDR 0108H A

FREQ\_MINUS . . . . C ADDR 016DH A

FREQ\_PLUS. . . . . C ADDR 01D9H A

HNG1 . . . . . . . C ADDR 01C4H A

HNG2 . . . . . . . C ADDR 01CAH A

HNG3 . . . . . . . C ADDR 01D0H A

INCR . . . . . . . C ADDR 009BH A

INCR1. . . . . . . C ADDR 00FDH A

INDIKATOR. . . . . C ADDR 0002H A

INIT . . . . . . . C ADDR 0002H A

KEYBOARD . . . . . C ADDR 0099H A

KEYBOARD1. . . . . C ADDR 000AH A

LCD. . . . . . . . C ADDR 01ABH A

LD0. . . . . . . . C ADDR 01BAH A

LD1. . . . . . . . C ADDR 01BEH A

LDL. . . . . . . . C ADDR 01B6H A

MAIN . . . . . . . C ADDR 01E6H A

MAINM. . . . . . . C ADDR 017AH A

N0 . . . . . . . . C ADDR 009DH A

N01. . . . . . . . C ADDR 00EBH A

N1 . . . . . . . . C ADDR 0039H A

N10. . . . . . . . C ADDR 013EH A

N100 . . . . . . . C ADDR 012EH A

N1000. . . . . . . C ADDR 011EH A

N10000 . . . . . . C ADDR 0113H A

N2 . . . . . . . . C ADDR 004BH A

N3 . . . . . . . . C ADDR 005DH A

N4 . . . . . . . . C ADDR 006FH A

N5 . . . . . . . . C ADDR 0081H A

N6 . . . . . . . . C ADDR 00A3H A

N7 . . . . . . . . C ADDR 0095H A

N71. . . . . . . . C ADDR 00B5H A

N8 . . . . . . . . C ADDR 0097H A

N81. . . . . . . . C ADDR 00C7H A

N9 . . . . . . . . C ADDR 009FH A

N91. . . . . . . . C ADDR 00D9H A

OK . . . . . . . . C ADDR 0205H A

OKM. . . . . . . . C ADDR 0199H A

P0 . . . . . . . . D ADDR 0080H A

P1 . . . . . . . . D ADDR 0090H A

P2 . . . . . . . . D ADDR 00A0H A

PULSE. . . . . . . C ADDR 01EEH A

PULSEM . . . . . . C ADDR 0182H A

STEP . . . . . . . C ADDR 00A1H A

STEP1. . . . . . . C ADDR 0100H A

STR. . . . . . . . C ADDR 01C0H A

TEMP . . . . . . . N NUMB 0050H A

TEST . . . . . . . C ADDR 0208H A

TESTM. . . . . . . C ADDR 019CH A

TF0. . . . . . . . B ADDR 0088H.5 A

TH0. . . . . . . . D ADDR 008CH A

TIMER. . . . . . . C ADDR 01F7H A

A51 MACRO ASSEMBLER KURS 06/05/2008 09:31:04 PAGE 7

TIMERM . . . . . . C ADDR 018BH A

TL0. . . . . . . . D ADDR 008AH A

TMOD . . . . . . . D ADDR 0089H A

TR0. . . . . . . . B ADDR 0088H.4 A

WAIT . . . . . . . C ADDR 0200H A

WAITM. . . . . . . C ADDR 0194H A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

1. **Проверка работоспособности**

Так как основной задачей программы является формирование импульсных последовательностей заданного количества импульсов с нарастающей частотой, а затем и убывающей, то можно предположить, что проверкой правильности работы программы могут служить графики этих последовательностей построенные в microVision. Эти графики приведены ниже.

График импульсной последовательности с нарастающей частотой:

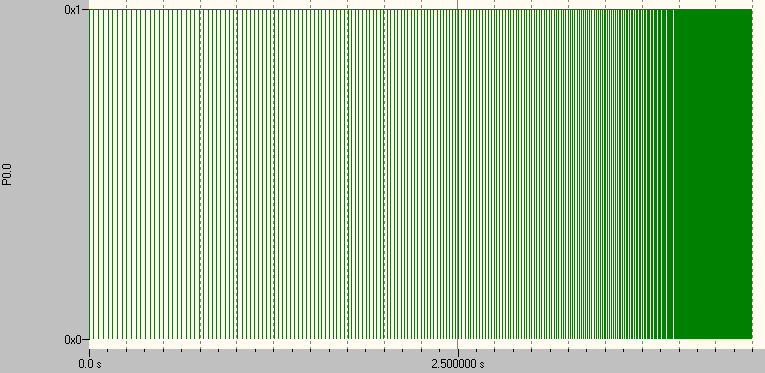
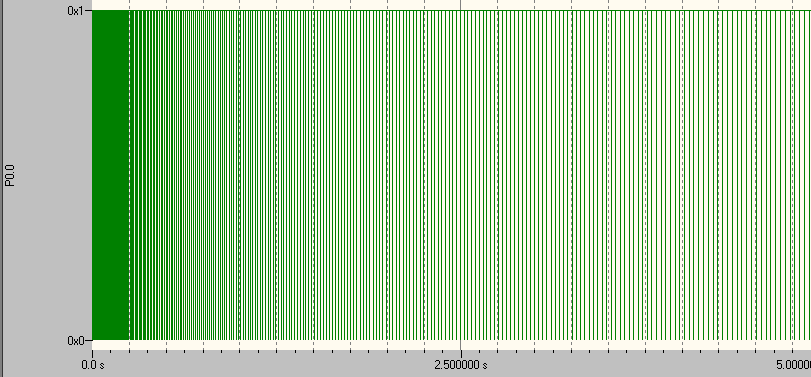


График импульсной последовательности с убывающей частотой:



**7. Список использованной литературы**

1. А.В. Фрунзе. Микроконтроллеры? Это же просто! Т.1. – М.: ООО “ИД СКИМЕН”, 2002. – 336 с.

2. В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. Проектирование цифровых устройств на однокристальных микроконтроллерах. - М.: Энергоатомиздат, 1990. – 224 с.

3. Глобальная сеть Internet