

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 06.11.2024 11:33:51
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d79c5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 11 » 04 2023 г.



МЕТОДЫ СБОРА И АНАЛИЗА БИОМЕДИЦИНСКОЙ ИНФОРМАЦИИ

Методические указания к самостоятельной работе по дисциплине
«Методы сбора и анализа медико-биологической информации» для
студентов направления подготовки 12.03.04 «Биотехнические
системы и технологии»

Курск 2023

УДК 004.93:61

Составитель: С.А. Филист

Рецензент

Доктор технических наук, профессор Р.А. Томакова

Методы сбора и анализа биомедицинской информации:
методические указания к самостоятельной работе / Юго-Зап. гос.
ун-т; сост.: С.А. Филист. - Курск, 2023. - 21 с.

Предназначено для студентов направления подготовки 12.03.04 «Биотехнические системы и технологии» по дисциплине «Методы сбора и анализа медико-биологической информации». Может быть использована аспирантами, обучающимися по научным специальностям 2.3.1. Системный анализ, управление и обработка информации и 2.2.12. Приборы, системы и изделия медицинского назначения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16. Бумага офсетная.

Усл. печ. л. 1,2. Уч.-изд. л. 1,1. Тираж 100 экз. Заказ 6/8.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Цель самостоятельной работы

Изучить программно-аппаратные средства преобразования медико-биологических сигналов в цифровую форму, методы организации файлов данных, получить теоретические и практические навыки проектирования аналоговых интерфейсов и разработки их программного обеспечения, изучить методы сопряжения различных датчиков медико-биологических сигналов с микропроцессорами и микроЭВМ.

1. Перечень модулей программного обеспечения, необходимых для самостоятельной работы

Модульная структура программы WAVE показана на рисунке 1.

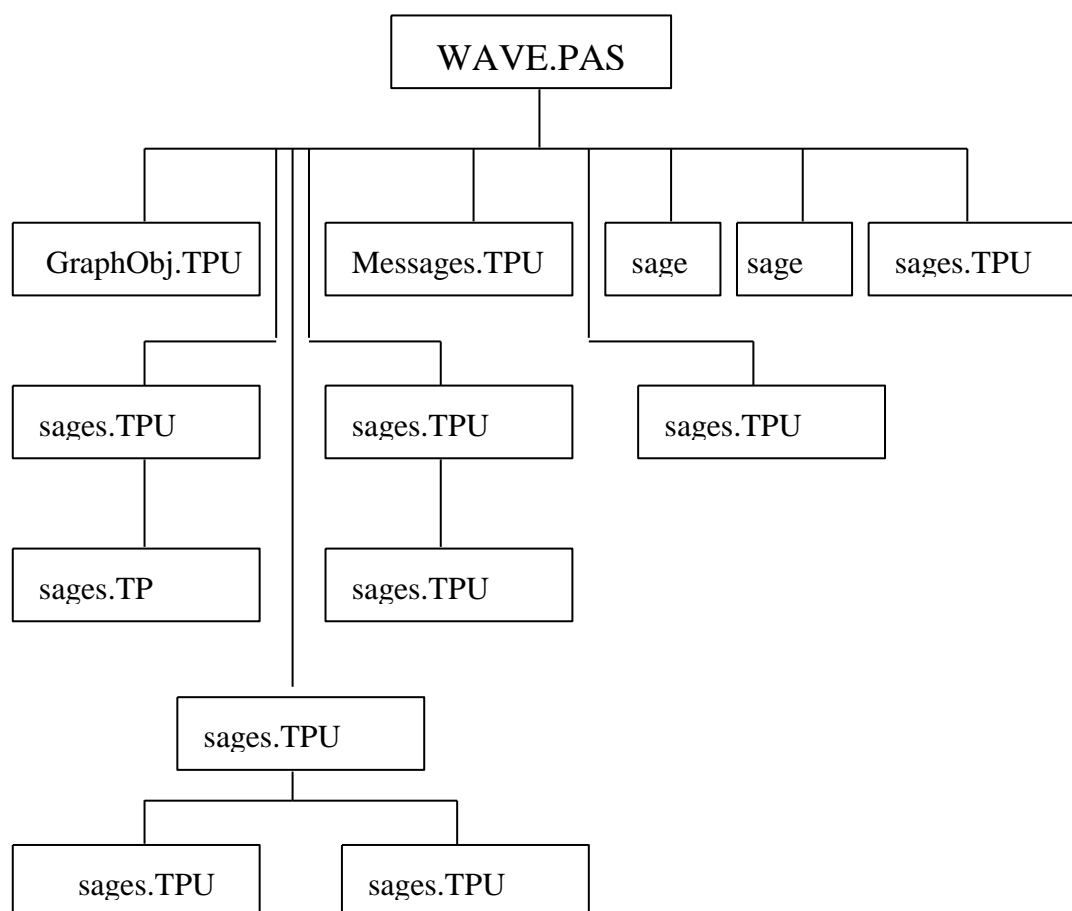


Рисунок 1 – Модульная структура программы WAVE

2. Перечень технических средств, необходимых для самостоятельной работы

- 1) WAVE.PAS;
- 2) GrphObj.PAS;
- 3) GrphMenu.PAS;
- 4) GrphApp.PAS;
- 5) GrphWin.PAS;
- 6) Messages.PAS;
- 7) Supply.ASM.

- Компьютер, совместимый с IBM PC
- Аналоговый интерфейс
- Датчик медико-биологического сигнала или имитатор медико-биологического сигнала

3. Задание на самостоятельную работу

3.1. Изучение работы программы WAVE

Программа WAVE предназначена для дискретизации низкочастотного сигнала (0...20 кГц), создание файлов данных определенной структуры, визуализации получаемых массивов данных и реализации некоторых простейших видов обработки получаемых данных. При работе с программой WAVE с адаптером типа CGA необходимо предварительно загрузить модуль FSU.COM.

При запуске программы WAVE на экране дисплея появляется горизонтальное меню, вид которого показан на рисунок 2.

Файл	Редактор	
Alt-x – конец работы F10 - Меню		Файл > Noname.CRV

Рисунок 2 – Вид горизонтального меню, появляющегося на экране после запуска программы WAVE

Пользователь может выбрать два режима: либо работать с уже полученными данными, либо получить новые данные. И в том, и в другом случае данные загружаются в оперативную память машины.

Загрузить данные в оперативную память машины можно либо с "магнитного диска в режиме "Файл", либо непосредственно от датчика сигнала в режиме "Редактор". Если вы хотите загрузить файл данных с магнитного носителя, то необходимо выбрать в горизонтальном меню режим "Файл", а в вертикальном меню режим "Прочитать...". Выбор осуществляется клавишами "→", "←", "↓", "↑" и "Enter".

После выбора режима "Прочитать..." на экране появляется приглашение указать имя файла, с которым вы будете работать. Отметим, что среда WAVE не позволяет прочитать каталог файлов данных, поэтому их имена нужно либо помнить, либо прочитать каталог в другой среде. Запомните, что программа WAVE создает файлы данных только с расширением CRV, которое можно не указывать. Набрав в окне приглашения имя файла данных, нажмите "Tab" и убедитесь, что загрузка прошла успешно. Вид окна приглашения показан на рисунке 3.

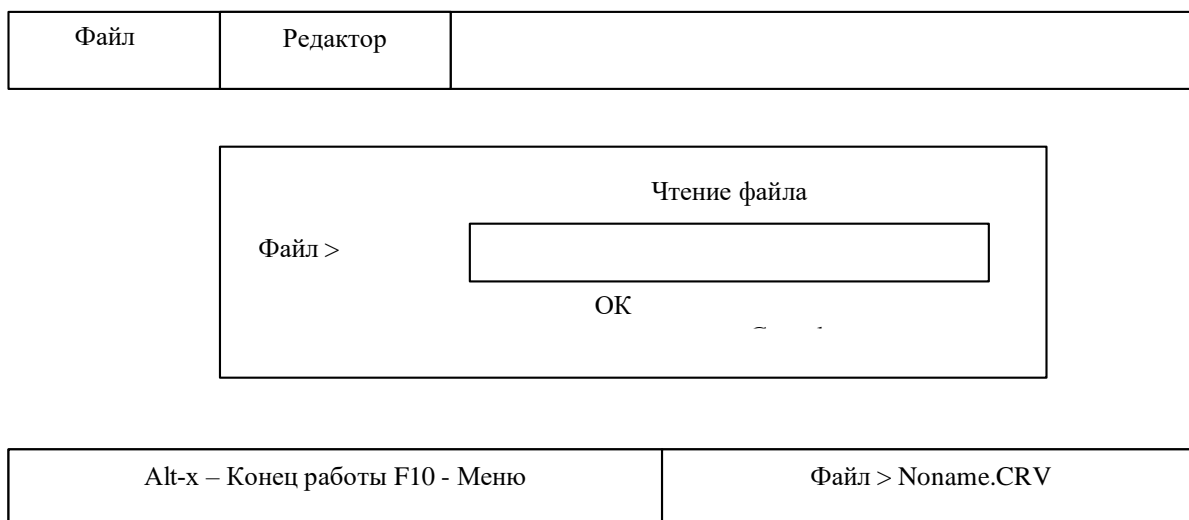


Рисунок 3 – Вид экрана при приглашении к вводу данных с магнитного носителя

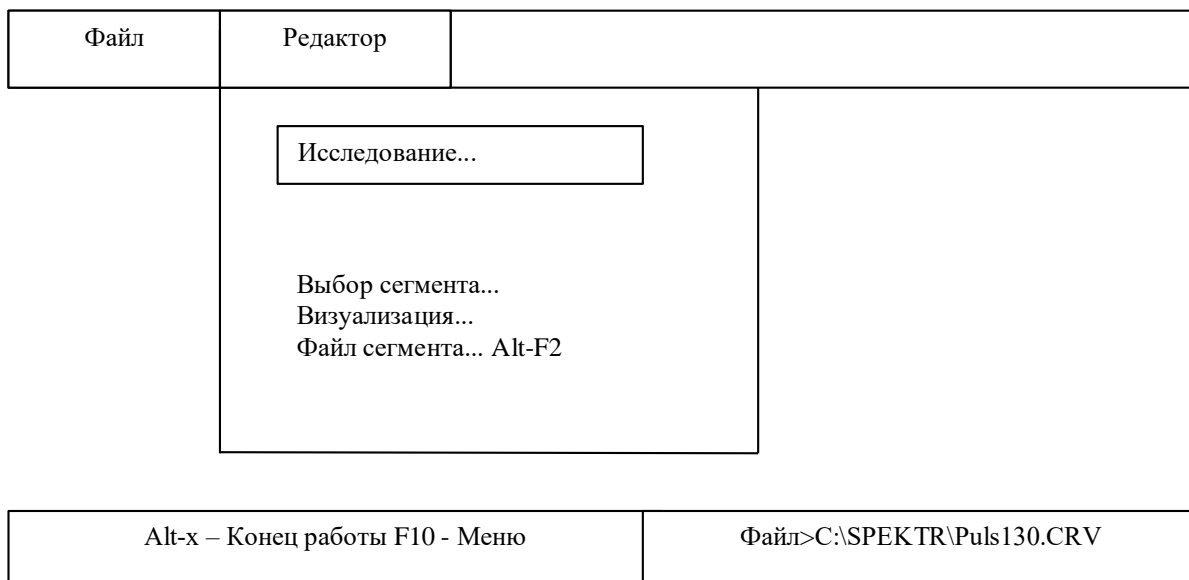


Рисунок 4 – Вид экрана после выбора горизонтального пункта меню "Редактор"

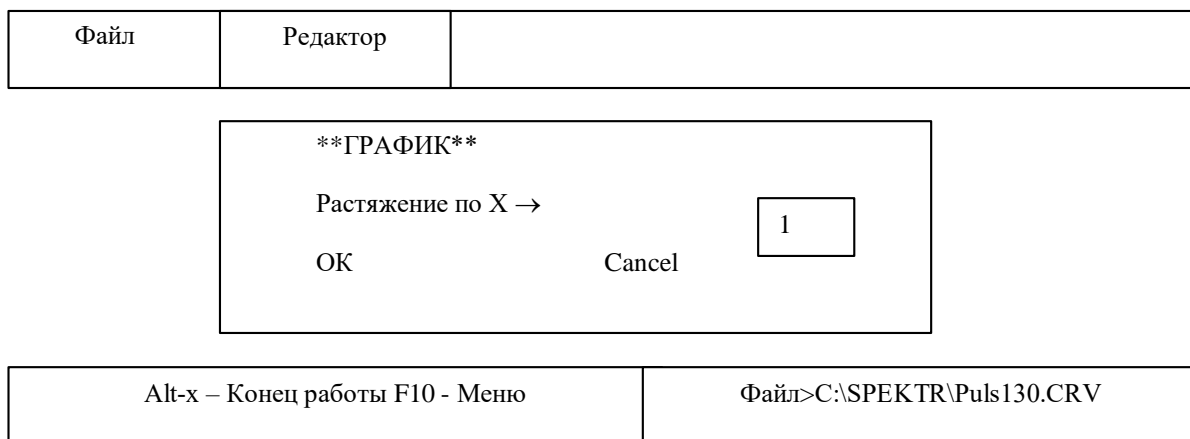


Рисунок 5 – Вид экрана при вводе масштабного коэффициента

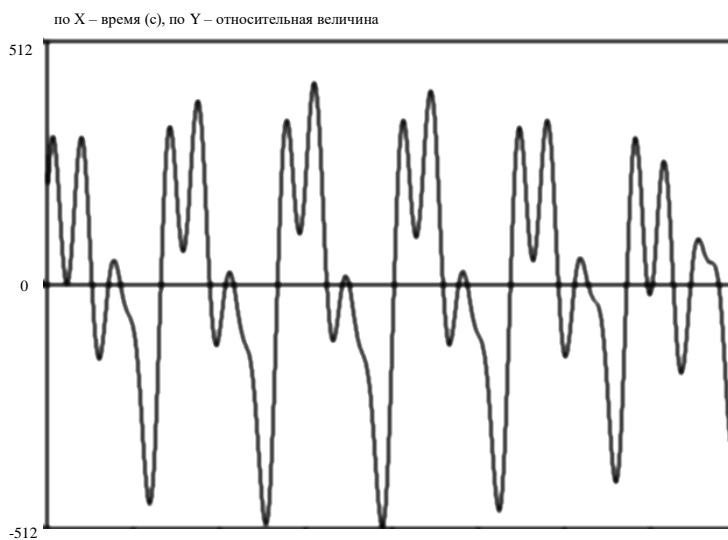


Рисунок 6 – График исследуемого процесса

Если ввод данных прошел успешно, то на экране появляется горизонтальное меню и пользователь может просмотреть данные, выбрав пункт горизонтального меню "Редактор". Вид экрана при выборе этого пункта показан на рисунке 4.

Просмотр данных осуществляется путем выбора пункта "Визуализация" в горизонтальном меню. Так как данные могут выводиться как на экран дисплея, так и на внешнее устройство, например, осциллограф, то на экране появляется еще одно вертикальное меню, в котором необходимо выбрать пункт "Дисплей". После этого на экране появится приглашение провести масштабирование по оси X путем ввода масштабного коэффициента, лежащего от 1 до 9. Вид приглашения показан на рисунке 5. После ввода коэффициента масштабирования и нажатия Enter на экране появится график исследуемого процесса. Возможный вид такого графика показан на рисунке 6.

В режиме "Редактор" имеется возможность выделить фрагмент данных в отдельный файл. Для этого в вертикальном меню режима "Редактор" необходимо выбрать режим "Сегмент".

Вид экрана в режиме "Сегмент" аналогичен виду экрана в режиме "Визуализация", но на графике на оси абсцисс появляется маркер прямоугольной формы, который можно передвигать путем одновременного нажатия клавиш "Ctrl" и "→" при точной установке, либо "Ctrl" и "- →" при грубой установке. Выделение фрагмента (сегмента) происходит путем нажатия клавиши "Enter". Выделенный фрагмент данных может быть записан в отдельный файл путем выбора в вертикальном меню режима "Редактор" пункта "Файл сегмента".

Для того, чтобы получить данные от источника сигнала, необходимо в режиме "Редактор" выбрать пункт "Исследование". Вид экрана после этого пункта показан на рисунке 7. Частота дискретизации выбирается исходя из критерия Найквиста, но не должна быть менее 19Гц, что обусловлено особенностями используемого таймера (см. листинг 1), и выше 20кГц, что обусловлено техническими характеристиками аналогового интерфейса. Следующий параметр, время наблюдения задает время считывания информации в секундах. При этом следует учитывать ограничения, связанные с конечностью оперативной памяти

машины. Максимальное время наблюдения сигнала T_{\max} вычисляется по формуле:

$$T_{\max} = \frac{M_{\max}}{L * f_G},$$

где M_{\max} - объем свободной оперативной памяти, в байтах;

f_G - частота дискретизации, в s^{-1} ;

L - длина слова, в байтах (количество байт, занимаемых одним отсчетом).

В нашем случае $L=2$, то есть отсчет занимает два байта. Расчет необходимой памяти производится автоматически, и при её нехватке раздается звуковой сигнал.

Файл	Редактор	
------	----------	--

Исследование	
Частота дискретизации (Гц)	→
Время наблюдения (с)	→
Номер источника (0...7)	→
Код усиления (0...7)	→
Коэффициент ослабления (1...1024)	→
ОК	Cancel

Alt-x – Конец работы F10 - Меню	Файл > Noname.CRV
---------------------------------	-------------------

Рисунок 7 – Вид экрана при выборе режима "Исследование"

Номер источника сигнала лежит в диапазоне от 0 до 7 и задается пользователем в зависимости от того, от какого источника дискретизируется сигнал.

Если сигнал необходимо усилить, то значение кода усиления задается больше нуля. Если же, наоборот, сигнал должен ослабляться, то вводится коэффициент ослабления, лежащий в пределах от 1 до 1024. Сигнал на выходе аттенюатора определяется по формуле:

$$U_{\text{ВЫХ}} = U_{\text{ВХ}} \left\{ 1 - \frac{K_{\text{осл}}}{1024} \right\}$$

где $U_{\text{ВХ}}$ - сигнал на входе аттенюатора;

$K_{\text{осл}}$ коэффициент ослабления, лежащий в пределах 1...1024.

Ввод параметров режима "Исследование" заканчивается нажатием клавиши "Tab". Если введенные данные устраивают пользователя, то нажимается клавиша "Enter". Если же параметры режима "Исследование" не удовлетворяют пользователя, то, перейдя на "Cancel и нажав "Enter", можно повторить ввод параметров режима "Исследование".

После ввода параметров режима "Исследование" и нажатия клавиши "Enter" происходит считывание данных из выбранного канала. По окончании режима "Исследование" происходит поиск max и min элементов сформированного массива, что необходимо для построения графика исследуемого процесса.

Если данные необходимо записать на магнитный носитель, то в режиме "Файл" выбирается подрежим "Запись" и указывается имя файла, в который помещаются считанные данные.

Листинг 1

```

; supply.asm
; Процедуры, обеспечивающие интерфейс с АЦП и ЦАП

.MODEL TPASCAL
.CODE

ASSUME CS:@CODE
PUBLIC DataIn, DataOut, SetParams
CORR      EQU      512
SetTimer  proc      near
            in       al,61h
            and      al,11111101B    ;запретить динамик
            (бит 1) и
            or       al,1           ;открыть ворота-gate (бит
            0)
            out      61h,al

```

```

                                10
                                mov     al,10110000B    ;установить
режим работы
                                out     43h,al        ;канала 2
                                ret
SetTimer   endp

```

; Procedure DataIn(where:pointer; n:word; counter:word)

```

WHERE EQU DWORD PTR [bp+10]    ;адрес указателей на
блоки
N EQU WORD PTR [bp+8] ;количество блоков памяти
COUNTER EQU WORD PTR [bp+6] ;значение счетчика
для таймера

```

```

DataIn   proc   far
          push   bp
          mov    bp,sp
          mov    bx,N          ;в BX количество блоков
          памяти
          push   ds
          lds   si,WHERE      ;в DS:SI адрес указателей на
          блоки
          les   di,[si]       ;в ES:DI адрес 1-го блока
          памяти
          mov    cx,[si+4]     ;в CX размер блока в
          словах
          add   si,6           ;отн-е смещение адреса след.
          блока
          cld
          call  SetTimer
repeat:
          mov    ax,COUNTER    ; значение счетчика
          out   42h,al         ; младший байт счетчика
          mov    al,ah
          out   42h,al         ; старший байт счетчика

          mov    dx,030Eh

```

```

    in      ax,dx      ;читаем слово из регистра
АЦП...
    stosw                      ;...и пересылаем его в память

    dec    cx          ; уменьшаем кол. свободной
памяти
    cmp    cx,0        ; есть еще память ?
    jnz   count       ; нет - загрузить новый адрес
блока
    dec    bx          ; переходим к следующему
блоку
    cmp    bx,0        ; есть еще блоки ?
    jz    repend      ; если нет - закончить работу
    les   di,[si]      ;в ES:DI адрес след-го блока
памяти
    mov   cx,[si+4]    ;в CX размер блока в
словах
    add   si,6         ;отн-е смещение адреса след.
блока;проверяем не истекло ли время
count:
    in    al,62h      ;считать из порта С
    test  al,00100000B ;проверяем 5-й бит-выход
2-го
                                ;канала
    jz    count       ;повторяем пока низкий
уровень
    jmp   repeat
repend:
    pop   ds
    pop   bp
    ret
DataIn  endp

```

;Procedure DataOut(where:pointer;n:word;counter:word;nda:word)

```

WHERE EQU DWORD PTR [bp+12] ;адрес указателей
на блоки

```

```

N          EQU    WORD PTR [bp+10]    ;количество
          блоков памяти
COUNTER EQU    WORD PTR [bp+8] ;значение счетчика
          для
          ;таймера
NDA       EQU    WORD PTR [bp+6] ;адрес регистра ЦАП

DataOut   proc   far
          push   bp
          mov    bp,sp
          call  SetTimer
          push   ds
          cld

          mov    ax,COUNTER    ; значение счетчика
          out   42h,al          ; младший байт счетчика
          mov    al,ah
          out   42h,al          ; старший байт счетчика

loading:
          mov    di,N            ;в DI количество
          блоков памяти
          les   bx,WHERE        ;в ES:BX адрес указат. на
          блоки
          lds   si,es:[bx]      ;в DS:SI адрес 1-го
          блока памяти
          mov   cx,es:[bx+4]    ;в CX размер блока в
          словах
          add   bx,6            ;отн-е смещение ад-са
          сл-го ;блока

count2:
          in   al,62h           ;читать из порта С
          test  al,00100000B    ;проверяем 5-й бит-
          выход 2-го ;канала
          jz   count2           ;повторяем пока низкий
          уровень

repeat1:
          mov   ax,COUNTER      ;значение счетчика
          out   42h,al          ;младший байт счетчика

```

```

mov     al,ah
out     42h,al           ;старший байт счетчика

mov     dx,NDA           ;регистр ЦАП
lodsw                   ;читаем слово из памяти в
AX
add     ax,CORR          ;корректируем...
out     dx,ax           ;..и посылаем его в
регистр ЦАП

mov     al,0
mov     ah,1             ;проверяем не
нажата ли
                               ;клавиша

int     16h
cmp     al,27            ;это клавиша ESC ?
je      repend1         ;если да завершить работу

dec     cx               ;уменьшаем кол-во
свободной                 ;памяти
                               ;есть еще память ?
cmp     cx,0
jnz    count1           ;нет - загрузить новый
адрес
                               ;блока
dec     di               ;переход. к следующему
блоку
cmp     di,0             ;есть еще блоки ?
jz     loading          ;если нет начинаем
сначала

lds     si,es:[bx]       ;в DS:SI адрес след.
бл. памяти
mov     cx,es:[bx+4]    ;в CX размер блока в
словах
add     bx,6             ;отн-е см-ние ад-са
след-го
                               ;блока

```

```

count1:
    in    al,62h           ; считать из порта С
    test  al,00100000B    ; проверяем 5-й бит-
        выход 2-го
                           ; канала
    jz    count1          ; повторяем пока низкий
                           ; уровень
    jmp   repeat1
repend1:
    pop   ds
    pop   bp
    ret

```

DataOut endp

;Procedure SetParams(source:byte; amplify, attenuate:word)

```

SOURCE      EQU    BYTE PTR [bp+10];№ источника (от 0
        до 7)
AMPLIFY     EQU    WORD  PTR  [bp+8];коэффициент
        усиления
ATTENUATE   EQU    WORD  PTR  [bp+6];коэффициент
        ослабления

```

```

SetParams   proc    far
            push    bp
            mov     bp,sp
            mov     ax,0
            mov     al,SOURCE      ; номер источника
            mov     dx,030Bh      ;      регистр
            измерительных
            out     dx,al          ;преобразователей
            mov     ax,AMPLIFY     ;
            коэффициент усиления
            mov     dx,030Dh      ; регистр коэф.
            усиления
            out     dx,ax
            mov     ax,ATTENUATE  ;
            коэффициент ослабления

```

```

                                mov     dx,030Ch      ;   регистр
аттенюатора
                                out     dx,ax
                                pop     bp
                                ret
SetParams endp

END

```

Листинг 2.

```
PFHeader = ^FHeader;
```

```

FHeader = record                { заголовок файла данных }
    DataType :string[4];        { тип данных REAL или INTG }
    Cols :longint;              { число элементов }
    Lines :byte;                { число строк }
    Frequency :longint;         { частота дискретизации }
    Time :longint;              { время наблюдения }
    Period :longint;            { период наблюдения }
    Pause :longint;             { период паузы }
    Min,Max :integer;           { минимальный и
максимальный }
                                { элементы }
end;                             { всего 30 байт }

```

Работу программы считывания данных в ОЗУ ЭВМ можно изучить по Листингу 1. Формат файла данных, получаемый с помощью этой программы, иллюстрирует Листинг 2. Как видно из него перед размещением данных на диске записывается заголовок длиной в 30 байт. Таким образом, файл данных является нетипизированным и запись и считывание из него ведется с помощью процедур работы с нетипизированными файлами, ознакомиться с которыми можно в [3].

При изучении листинга 2 обратите внимание, что число строк в заголовке файла данных объявлено как переменная типа `byte`, следовательно, при составлении своих программ или при

использовании уже готовых программ, работающих с этими файлами данных, нельзя допускать ситуации, когда число строк превысит величину 255.

3.2. Изучение модуля аналогового интерфейса

Решение задачи рационального построения системы обработки данных может базироваться на использовании аналоговых микропроцессоров (АМП), структура которых формируется на основе следующей цепи преобразования и обработки сигналов:

$$АСВ_{\text{х}} \rightarrow АЦП \rightarrow ЦОС \rightarrow ЦАП \rightarrow АСВ_{\text{вых}},$$

где АСВ_х - входные аналоговые сигналы;

АЦП - аналого-цифровое преобразование;

ЦОС - цифровая обработка сигналов;

ЦАП - цифро-аналоговое преобразование;

АСВ_{вых} - аналоговые сигналы выхода.

Так как в основу АМП положена цифровая обработка сигналов, то системы обработки данных, построенные с применением АМП, обладают высокими эксплуатационными характеристиками.

Примером АМП данного класса может служить ПЭВМ 1030.11 совместно с аналоговым интерфейсом, подключенным к системной шине, структурная схема которого показана на рисунке 8.

Работа представленной структуры аналогового интерфейса управляется программой WAVE посредством пользовательских интерфейсных терминалов, возможные виды которых представлены на рисунках 2-5,7. Пользователь выбирает один из восьми измерительных преобразователей, данные от которых поступают в ОЗУ ЭВМ с последующей возможностью записи на магнитный носитель, или выбирает один из двух выходных каналов, данные от которых поступают к внешнему по отношению к ЭВМ объекту, причем выходная информация представлена в виде аналоговой величины.

Аналоговый интерфейс подключаем к системной шине ПЭВМ. В качестве управляющих сигналов используются следующие сигналы системной шины: XACK, IORD, IOWT.

Ввод и вывод данных в центральном процессоре K1810BM86 может осуществляться двумя способами: с использованием адресного пространства ввода-вывода, и с использованием общего с памятью адресного пространства, то есть с отображением на память.

При первом способе применяются специальные команды IN (ввод) и OUT (вывод), которые обеспечивают передачу данных между аккумуляторами AL или AX и адресуемыми портами. При выполнении этих команд вырабатывается сигнал M/IO=0, который идентифицирует выбор пространства ввода-вывода и в совокупности с сигналами WR и RD позволяет сформировать системные сигналы IOW и IOR для управления операциями записи данных в порт и чтение из порта.

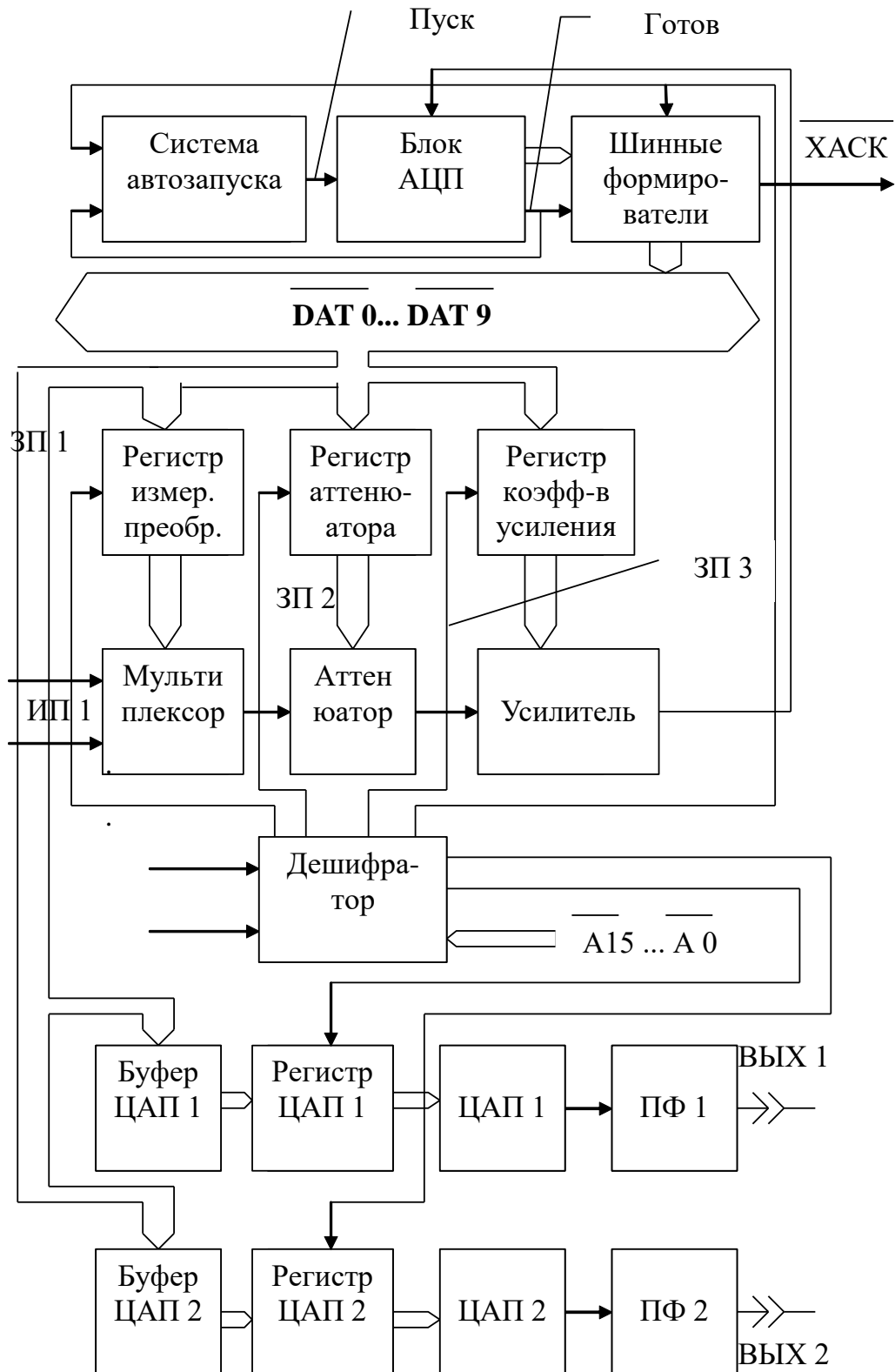


Рисунок 8 –Принципиальная схема

Команды IN и OUT могут использовать прямую адресацию, когда адрес порта содержится в виде константы во втором байте команды, и косвенную адресацию, когда адрес располагается в регистре DX. В первом случае можно адресовать по 256 портов для

ввода и вывода данных. Во втором обеспечивается адресное пространство до 64К 8-битовых портов или до 32К 16-битовых портов. Косвенная адресация позволяет вычислять адреса портов при выполнении программы и удобна при организации вычислительных циклов для обслуживания нескольких портов с помощью одной процедуры.

Восемь ячеек F8...FF в пространстве ввода-вывода зарезервированы для системных целей, и использовать их в прикладных программах не рекомендуется.

При втором способе адреса портов размещаются в общем адресном пространстве, и обращение к ним не отличается от обращения к ячейкам памяти. Это повышает гибкость программирования, так для ввода-вывода можно использовать любую команду с обращением к памяти при любом способе адресации. Так, команда MOV позволяет передать данные между любым общим регистром или ячейкой памяти и портом ввода-вывода, а логические команды AND, OR, XOR и TEST позволяют манипулировать битами в регистре порта. При этом, однако, следует учитывать, что команды с обращением к памяти имеют больший формат и выполняются дольше, чем простые команды IN или OUT. Кроме того, несколько усложняется дешифрирование 20-битового физического адреса порта и сокращается число адресов, которые могут использоваться для чтения ячеек памяти.

Микропроцессор может передавать по шине байт или слово в/из ВУ. Чтобы слово передавалось за один цикл шины, адрес ВУ должен быть четным. Адрес байтового ВУ может быть четным или нечетным, и соответственно порты этих внешних устройств подключаются к линиям младшего и старшего байта шины данных. Для отдельного обращения к этим портам дешифрирование адресов осуществляется с учетом сигналов на линиях ВНЕ (разрешение старшего байта дополнительный адресный выход) и АО.

Схема запуска АЦП построена таким образом, что АЦП находится в перманентном режиме преобразования данных. Этот режим позволяет осуществить схему автозапуска, выполненная на двух одновибраторах DD1, DD2 K155АГ1. Таким образом, в регистре АЦП всегда находится отсчет последних данных с

максимальным запаздыванием, равным времени преобразования АЦП, то есть 30 мкс.

Блок АЦП состоит из АЦП К113ПВ1 и схемы выборки-хранения. Схема выборки-хранения необходима для "защелкивания" аналогового сигнала на время преобразования АЦП и повышает точность преобразования.

Так как аналоговый интерфейс выполняет функции контроллера внешних устройств, то он должен генерировать сигнал ХАСК, который, в свою очередь, генерирует сигнал READY для микропроцессора КР1810ВМ86. Схемы формирования сигнала ХАСК различны для АЦП и регистров (измерительных преобразователей, аттенуаторов, усилителей, ЦАПов). Это связано с тем, что регистры работают в темпе ЭВМ, что не требует дополнительной синхронизации, АЦП работает согласно своему циклу. Кроме того, установка истинных данных в регистре АЦП отстает на 1 мкс от сигнала "Готов" АЦП. В связи с этим сигнал ХАСК генерируется с соответствующим запаздыванием относительно "сигнала "Готов" АЦП, что обеспечивается вторым одновибратором DD2. Так как в регистре АЦП всегда находятся последние данные, то при обращении к нему, то есть по адресу 030В, достаточно заблокировать цепь автозапуска и считать данные по сигналу "Готов" АЦП. Сигнал блокировки приходится на вход ВС DD1. Таким образом, цикл опроса АЦП может быть значительно ниже времени преобразования. Получив сигнал, ХАСК ПЭВМ снимает адрес АЦП и тем самым восстанавливает цепь автозапуска. Отметим, что ПЭВМ имеет схему синхроконтроля, которую реализует второй счетчик таймера КР580ВИ53 (адрес 41Н) и если от порта ввода-вывода или памяти не придет сигнал ХАСК в течение 200мкс, то отрабатывает схема синхроконтроля при наличии сигнала "TE"=1 (разрешение работы таймера).

4. Контрольные вопросы

1. Как определить предельную частоту дискретизации АМП?
2. Как получить с помощью АЦП спектр дифференциального сигнала пульса?
3. Как получите с помощью АМП спектр объемного сигнала пульса?

4. Исследуйте с помощью АМП спектр помех дифференциального сигнала пульса.

5. Исследуйте с помощью АМП спектр помех объемного сигнала пульса.

6. Исследуйте методы получения фонокардиосигнала.

7. Исследуйте с помощью АМП помехи, присутствующие при получении фонокардиосигнала.

8. Определите с помощью АМП спектр фонокардиосигнала.

9. Объясните характер спектра фонокардиосигнала с физиологической точки зрения.

10. Разработайте программу опроса нескольких источников сигнала в реальном масштабе времени.

11. Известно, что преобразование Фурье обратимо. Обратимо ли разложение функции в ряд Фурье? Ответ обоснуйте.

12. Каким образом построена схема запуска АЦП?

13. Перечислите компоненты модульной структура программы WAVE.

14. Какие функции выполняет аналоговый интерфейс?

15. На какой основе формируется структура работа аналоговых микропроцессоров?

5. Список литературы, необходимый для самостоятельной работы

1. Белецкий Я. Энциклопедия языка СИ: Пер. с польск.-М.: Мир, 1992.-687с.

2. Кассим, Кабус Дерхим Али. Компьютерные технологии обработки и анализа биомедицинских сигналов и данных [Текст] : учебное пособие : [для студентов направления подготовки 12.03.04 и 12.04.04 "Биотехнические системы и технологии", аспирантов направ. подготовки 12.06.01 "Фотоника, приборостроение, оптические и биотехнические системы и технологии] / К. Д. А. Кассим, С. А. Филист, А. Ф. Рыбочкин ; Юго-Зап. гос. ун-т. - Курск : ЮЗГУ, 2016. - 290 с. - Библиогр.: с. 288-289. - ISBN 978-5-7681-1159-5 : 410.00 р.

3. Зубов В.С. Программирование на языке TURBO PASCAL (версии 6.0 и 7.0)-М.: Информационно-издательский дом "Филинь", 1997.-304с.

4. Кардиомониторы. Аппаратура непрерывного контроля ЭКГ: Учебное пособие для ВУЗов./А.Л. Барановский, А.М. Колинченко, Л.А. Манило и др. под ред. А.Л. Барановского и А.П. Немирко.-М.: Радио и связь, 1993.-248с.

5. Левшина Е.С., Новицкий П.В. Электрические измерения физических величин: (Измерительные преобразователи). Учеб. пособие для ВУЗов.-Л.: Энергоатомиздат. Ленинградское отделение, 1983.-320с.

6. Новиков Ю.В., Калашников О.А., Гуляев С.Э. Разработка устройств сопряжения для персонального компьютера типа IBM PC. Под общей редакцией Ю.В. Новикова. Практ. пособие.-М.:ЭКОМ.,1997-224 с.

7. Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC; Пер.с англ./Под ред. У.Томкинса, Дж. Уэбстера.-М.: Мир,1992.-592 с.

8. Бобрыкин А.В., Липовецкий Г.В., Литвиновский Г.В. и др. Однокристалльные микроЭВМ. М.: МИКАП, 1994.-400с.