

## Аннотация

Уважаемый пользователь ПК "Вектор-06ц"! Перед тем, как Вы начнете изучение данной документации, хотелось бы об'яснить Вам, для кого и для чего она написана, и какой круг вопросов освещает.

Первое знакомство с азами программирования начинается, как правило, с языка Бейсик. И это, наверное, правильно: на этом этапе пользователю совсем необязательно знать устройство своего компьютера. По мере накопления опыта программирования, а также появления новых инструментальных средств, программист начинает работать на других, более совершенных языках и системах. Но даже на этом уровне, у многих остается чувство "неполной власти" над своим компьютером. Т.к. даже полное знание языка Ассемблера не дает ответа на некоторые вопросы, связанные с программированием внутренней аппаратуры машины. Да и чисто человеческое любопытство остается неудовлетворенным.

Сложность самостоятельного понимания этих вопросов заключается в том, что пользователю-программисту необходимо хорошо знать электронику и отлично разбираться в принципиальной электрической схеме своего компьютера. Поэтому, многие программисты тратят огромное количество времени на выяснение этих тонкостей.

Цель этого описания - избавить программиста от кропотливого труда и дать ему сразу конечный продукт этих изысканий на конкретных примерах программ. Описание будет очень полезно и тем, кто уже имеет опыт работы на других компьютерах, а сейчас стал заниматься "Вектором".

Теперь о формате данного описания: оно представляет собой листинг программ на языке Ассемблера с подробными комментариями в каждой строке. Чтобы ввести его в компьютер, Вам необходимо загрузить МОНИТОР-ОТЛАДЧИК и после его запуска выбрать режим "2". Затем через директиву "R" монитора-отладчика, ввести программу "Редактор-Ассемблер" и, пользуясь им, ввести текст программы без комментариев - т.е. только команды ассемблера. А точки с запятой и весь текст за ним не печатать - т.к. весь текст просто не поместится в память при режиме монитора "2", да и комментарии у Вас уже распечатаны. Для тех, кто пока еще не работал с редактором-ассемблером, в Приложении 1 дана распечатка машинных кодов этой программы, работающая только с монитором (или вообще автономно). Начинается и запускается эта программа с адреса 0100h. Для изменения параметров дана распечатка таблицы меток.

Данную программу можно использовать как учебную - т.е. изменять ее, транслировать и наблюдать работу. Но можно делать на ее базе свою программу, особенно, если Вам нужен минимальный объем (например такую программу, как ТЕСТ ТЕХРПОГОНА - всего 4 блока) или когда объем графики или данных в Вашей программе не позволяет использовать Стандартные Драйверы Устройств.

Опытные программисты могут использовать листинг программы как справку по программированию ПЭВМ "Вектор-06ц".

```
;*****
; Программирование режимов ПЭВМ "Вектор-06ц" на языке Ассемблера.
; Секреты "Вектора"-1992г. Лебедев А.З.
; 400119, Волгоград, А/Я 2684.
;*****
;
; У многих владельцев ПЭВМ "Вектор-06ц" наверняка возникало желание попробовать свои силы в программировании на языке Ассемблера, но это начинание сразу же наталкивалось на необходимость программной установки режимов ПЭВМ: цветов изображения, фона, бордюра, управление скроллингом экрана, выводом звука и опросом клавиатуры.
; Возможно, предлагаемая программа поможет Вам преодолеть этот барьер.
;
; Запуск программы: Сейчас Вы имеете Т Е К С Т программы с комментариями. Прежде чем запустить, его надо оттранслировать в машинные коды, для этого: нажмите клавишу "СТР" и, после появления надписи "АССЕМБЛЕР", нажмите 3. Появится число, показывающее конец машинных кодов этой программы, а начинается она с адреса 1800h. Далее, нажмите клавишу "ВК" и, тем самым, выйдите в МОНИТОР. Запустите программу командой G1800, по окончании она вызовет РЕДАКТОР. В РЕДАКТОРЕ нажмите на клавишу "AP2" и затем "В" - и Вы снова увидите этот текст. Теперь Вы можете изменять текст программы, затем транслировать ее в машинные коды, запускать и наблюдать работу.
; Подпрограммы "Start" и "INIT" пока не трогайте - чтобы их усовершенствовать Вам надо хорошо знать аппаратную часть машины т.к. они и выполняют всю
```

;настройку (инициализацию) ПЭВМ и нужны именно для того, чтобы Ваши программы  
;в машинных кодах могли работать на ПЭВМ "Вектор-06ц".

; Свою программу располагайте с метки "PROGRM:" и, если Вы будете запускать  
;ее из МОНИТОРА, устанавливайте стек 53FFh, чтобы не стереть монитор.

; Для успешной работы Вам необходимо всегда иметь под рукой описание  
;МОНИТОРА-ОТЛАДЧИКА, АССЕМБЛЕРА-РЕДАКТОРА, описание работы ПЭВМ "Вектор-06ц"  
;и, желательно, справочник по программированию БИС 580BV55 и 580BI55.

; Начало программы: Как Вы заметили, многие программы на "Векторе" могут пе-  
;резапускаться после нажатия на клавиши "БЛК" и "СВРОС". После нажатия на эти  
;клавиши счетчик команд процессора указывает на адрес 0000h - то есть именно  
;с этого адреса начнется выполнение программы. Поэтому, если Ваша программа  
;начинается не с нулевого адреса, а Вы хотите иметь ее перезапуск, Вам необ-  
;ходимо установить по адресу 0000h переход на Вашу программу. Кроме того, Вам  
;надо установить переход на обработку прерывания с адреса 0038h на адрес, где  
;начинается программа INIT (о которой будет сказано далее). Если в Вашей сис-  
;теме есть электронный диск (КВАЗИДИСК), или Вы хотите чтобы Ваши программы  
;нормально запускались как на обычном "Векторе", так и на "Векторе" с квази-  
;диском, то Вам необходимо при старте Вашей программы отключить обращение к  
;ОЗУ квазидиска. Эти начальные установки выполняются в участке программы с  
;меткой "Start".

```

;
;      ORG 1800h          ; Определить начало машинных кодов этой программы
Start: DI                ; Запретить прерывания (т.к. пока не ясно куда переход)
      MVI A,0C3h         ; установить переход (код команды JMP) в
      STA 0000h          ; нулевой адрес и
      STA 0038h          ; адрес вызова прерывания.
      LXI H,Progrm       ; Установить адрес перехода по "БЛК","СВРОС"
      SHLD 0001h         ; (сразу после кода команды JMP)
      LXI H,Init         ; Установить адрес перехода на обработку прерывания
      SHLD 0039h         ; (также после JMP)
      XRA A              ; Этой командой проще всего записать в аккумулятор ноль
      OUT 10h            ; и пересылкой кода 00h в порт 10h отключить квазидиск
      EI                 ; Разрешить прерывания, т.к. переходы установлены
      JMP PROGRM         ; Перейти на начало программы
;

```

; Теперь поговорим о прерывании, т.к. для многих это довольно темный вопрос.  
;Микропроцессор 580BM80, которым обладает "Вектор" и ряд других бытовых ком-  
;пьютеров, имеет, так называемый ВХОД запроса прерывания. Если на этот вход  
;будет подан сигнал, то процессор, независимо от того, в каком месте памяти он  
;выполнял программу, сделает переход на один из восьми фиксированных адресов,  
;точнее не переход, а вызов подпрограммы, после выполнения которой он благопо-  
;лучно вернется на тот адрес, с которого произошло П Р Е Р Ы В А Н И Е основ-  
;ной программы. Вас, как программиста, не должно это очень пугать - т.к. у  
;"Вектора" задействован только один такой адрес - 0038h, а вход запроса преры-  
;вания Вам программно недоступен. Так что же вызывает прерывание? Прерывание  
;вызывает схема компьютера, дающая импульсы синхронизации для Вашего дисплея  
;(телевизора). Этот импульс, а, следовательно, и вызов подпрограммы по 0038h  
;адресу происходит 50 раз в секунду, в моменты обратного хода кадровой раз-  
;вертки, т.е. моменты, когда экран вашего дисплея погашен (если Вы думали что  
;экран телевизора светится всегда, Вы ошибались - подвигайте быстро рукой пе-  
;ред экраном и Вы увидите строб-эффект, связанный с миганием экрана).

; Вполне возможно, что у Вас возникнет вопрос - зачем это надо: вызывать  
;подпрограмму, обращение к которой Вы в программе нигде не ставите? Да еще де-  
;лать это так часто, и почему в "темное" для экрана время?

; Ну вообще-то такое прерывание существует во всех профессиональных ЭВМ т.к.  
;это очень удобная вещь для подсчета реального времени. В самом деле: ведь  
;если в программе обработки прерывания организовать счетчик, который будет  
;увеличивать через каждые 50 прерываний - это будет точный счетчик секунд,  
;ну, а значит, и все остальное (время и дата) сделать тоже просто. И эти часы  
;будут работать вместе с Вашей программой, абсолютно не мешая ей!

; Но иногда, такая полезная вещь как прерывание, становится ненужной: при  
;работе со стеклом (можно потерять адрес возврата из прерывания), или когда  
;Вам необходимо "выжать" всю скорость из компьютера. В таких случаях прерыва-  
;ние можно запретить командой DI, а вновь разрешить командой EI.

; А конкретно для "Вектора" прерывание имеет еще большее значение: устано-  
;вить цвет экранных плоскостей и опросить клавиатуру можно лишь во время пре-  
;рывания, иначе будет дергаться изображение, т.к. микросхема, к которой под-  
;ключена клавиатура, управляет и цветом бордюра, и скроллингом (вертикальным

```

;перемещением экрана).
; Чтобы было более понятно, опишем, чем должна заниматься программа обработ-
;ки прерывания:
;
;1). Сохранить все регистры микропроцессора, чтобы прерванная программа могла
;нормально продолжить свою работу.
;
;2). Записать в ОЗУ цветогенератора (порт 0Ch) шестнадцать байт, которые обыч-
;но называют таблицей цветов. При записи надо переключать по порядку адрес ОЗУ
;цветогенератора - адресные шины этого ОЗУ подключены к четырем младшим разря-
;дам порта 02.
;
;3). Опросить клавиатуру (кроме клавиш "УС","СС","РУС/LAT"), которая подключе-
;на в виде матрицы 8x8 к портам 03h и 02h. Причем опрос клавиш производится
;следующим образом: только на один бит порта 03h можно подавать лог."0" и за-
;тем читать состояние линейки из восьми клавиш из порта 02h. Чтобы опросить
;все клавиши надо последовательно подавать лог."0" на все биты порта 03h и по-
;следовательно читать состояние каждой линейки из порта 02h.
;
;4). После записи таблицы цветов и опроса клавиатуры надо установить режимы
;отображения информации на время вывода ее на дисплей (до следующего прерыва -
;ния):
;4.1). Установить на четырех младших битах порта 02h номер цвета (от 0 до 15)
;бордюра.
;4.2). Записать в порт 03h байт, значение которого определяет вертикальное по-
;ложение экрана. Фактически это номер строки, с которой начнется отображение
;информации на экране. Правда, этот номер должен быть записан в порт 03h в ин-
;версном виде. Например: надо получить нормальное положение экрана (не сдвину-
;тое по вертикали), значит в порт 03h надо занести номер 0-й строки, но инвер-
;тированный - т.е. FFh.
;
;5). Записать в третий бит порта 01 сигнал управления индикатором "РУС/LAT" и
;в первый бит порта 01 сигнал управления реле (в некоторых "Векторах" это реле
;отсутствует, т.к. далеко не каждый пользователь подключает двигатель своего
;магнитофона через это реле).
;
;6). Вернуть из стека все регистры микропроцессора, разрешить прерывания и
;выйти из подпрограммы обработки прерываний командой RET, как при выходе из
;обычной подпрограммы.
;
; Для того, чтобы удобно было менять задаваемые в прерывании режимы компью-
;тера, надо под эти параметры отвести соответствующие ячейки памяти и исполь-
;зовать их как системные ячейки из своей основной программы. В самом деле, бу-
;дет значительно удобнее и понятнее, если, например, для изменения цвета ка-
;кой-либо плоскости экрана или цвета бордюра, Вам надо будет лишь изменить
;значение ячейки памяти с соответствующей меткой - "COLR" или "BORDER", совер-
;шенно не думая об адресах портов. Итак, определим эти ячейки:
;
;
Border: DB 15 ;Цвет бордюра (сейчас соответствует цвету 15)
Color: DB 00h ;Цвет фона (задан черный цвет)
;
COLR1: DB 07h ;Цвет плоскости E000-FFFF (задан ярко-красный цвет)
COLR2: DB 38h ;Цвет плоскости C000-DFFF (задан ярко-зеленый цвет)
COLR3: DB 16h ;Цвет совмещения E000-FFFF и C000-DFFF (кирпичный)
COLR4: DB 0C0h ;Цвет плоскости A000-BFFF (задан ярко-синий цвет)
COLR5: DB 86h ;Цвет совмещения A000-BFFF и E000-FFFF (малиновый)
COLR6: DB 20h ;Цвет совмещения A000-BFFF и C000-DFFF (темно-зеленый)
COLR7: DB 0C5h ;Цвет совм. A000-BFFF, C000-DFFF, E000-FFFF (фиолетовый)
COLR8: DB 3Fh ;Цвет плоскости 8000-9FFF (ярко-желтый цвет)
COLR9: DB 04h ;Цвет совмещения 8000-9FFF и E000-FFFF (темно-красный)
COLR10: DB 52h ;Цвет совмещения 8000-9FFF и C000-DFFF (серый)
COLR11: DB 98h ;Цвет совм. 8000-9FFF, C000-DFFF, E000-FFFF (бирюзовый)
COLR12: DB 80h ;Цвет совм. 8000-9FFF, A000-BFFF (темно-синий)
COLR13: DB 24h ;Цвет совм. 8000-9FFF, A000-BFFF, E000-FFFF (темно-желтый)
COLR14: DB 0ffh ;Цвет совм. 8000-9FFF, A000-BFFF, C000-DFFF (ярко-белый)
COLR15: DB 0f7h ;Цвет совм. 8000-9FFF, A000-9FFF, C000-DFFF, E000-FFFF (гол.)
;
;
; Номер байта в таблице цветов - это математический код цвета (т.е какой из
;16-ти байт цветогенератора?), а значение этого байта - это физический код

```

```

;цвета. Физический код цвета кодируется очень просто и Вы можете присвоить лю-
;бой другой цвет, любой из плоскостей или любому из их сочетаний, а также
;цвету фона. Кодировка такая: младшие три бита - яркость красного цвета (на-
;пример 07h - самый яркий красный, а 01h - самый темный), следующие три бита -
;яркость зеленого, два самых старших бита - яркость синего. Смешивая значения
;яркостей (соответствующих им битов) цветов, можно получить до 255 различных
;цветов и оттенков для каждой плоскости и их любых сочетаний.
;
;
Scroll: DB 0FFh          ;ячейка скроллинга экрана (сейчас нормальное положение)
KeyKod: DB 0FFh          ;ячейка в которой будет код нажатой клавиши
;
;
; В данной программе, с целью ее упрощения, реализован опрос только одной
;линейки клавиш, где расположены клавиши курсора (кроме диагональной), а также
;клавиши "BK", "PC", "ЗБ" и "ТАБ". Если ни одна из этих клавиш не нажата, то
;в ячейке "KeyKod" будет 0FFh. Если какая-либо из клавиш этой линейки нажата,
;то в ячейке "KeyKod" будет код этой клавиши, но его не надо путать со стан-
;дартным кодом (ASCII или KOI) - это физический код пересечения одной линии
;порта 03h с одной из 8-ми линий порта 02h. Чтобы было проще проверять в ос-
;новной программе эти коды, надо определить их через соответствующие метки:
;
;
Down: EQU 7Fh            ;Код клавиши "Курсор вниз"
Up: EQU 0DFh             ;Код клавиши "Курсор вверх"
Right: EQU 0BFh          ;Код клавиши "Курсор вправо"
Left: EQU 0EFh           ;Код клавиши "Курсор влево"
Del: EQU 0F7h            ;Код клавиши "ЗБ"
CR: EQU 0FBh             ;Код клавиши "BK"
LF: EQU 0FDh             ;Код клавиши "PC"
TAB: EQU 0FEh            ;Код клавиши "ТАБ"
;
IndRUS: DB 00h           ;Признак вкл.индикатора "РУС/LAT" (00-не горит, 08-горит)
Rele: DB 00h             ;Признак вкл.реле упр. магн. (02-выкл, 00-включено)
;
;=====
;Подпрограмма обработки прерывания.
;=====
;
INIT: PUSH H              ;Сохранить все регистры в стеке
      PUSH B              ;
      PUSH D              ;
      PUSH PSW            ;
      MVI A,88h           ;Установить режимы портов: 02 - вывод, 03 - вывод
      OUT 00              ;мл.4 бита порта 01 - вывод, старшие 4 бита - ввод
      LXI H,COLOR15       ;в пару "HL" - адрес последнего цвета таблицы цветов
      MVI D,16            ;в "D" - счетчик колич. устанавлив. цветов (размер табл)
      MVI E,15            ;в "E" - адрес для записи в ОЗУ цветогенератора
INIT1: MOV A,E             ;установить адрес ОЗУ цвета через
      OUT 02              ;порт 02
      MOV A,M             ;взять в "A" байт (физический код) цвета из таблицы цветов
      OUT 0Ch             ;записать этот байт в ОЗУ цветогенератора и повторить
      OUT 0Ch             ;несколько раз для надежной записи по одному адресу,
      OUT 0Ch             ;т.к. в некоторых "Векорах" наблюдается очень плохая
      OUT 0Ch             ;запись в ОЗУ цветогенератора и лучше подстраховаться.
      OUT 0Ch             ;
      DCX H               ;уменьшить адрес-указатель в таблице цветов
      OUT 0Ch             ;
      DCR E               ;уменьшить адрес для ОЗУ цветогенератора
      OUT 0Ch             ;
      DCR D               ;уменьшить счетчик байт в таблице цветов
      OUT 0Ch             ;
      JNZ INIT1           ;если не вся таблица записана в цв.ген. - продолжать
      MVI A,8Ah           ;установить режимы портов 02-вывод, 03-вывод
      OUT 00              ;мл. 4 бита порта 01 - вывод, старшие 4 бита - ввод
      MVI A,0FEh          ;установить в младшем разряде порта 03 нулевой потенциал
      OUT 03              ;чтобы опросить 1-ую линейку клавиатуры (курсоры и т.д.)
      IN 02               ;взять из порта 02 физич. код нажатия клавиш линейки
      STA KeyKod          ;поместить этот код в ячейку "KeyKod"
      MVI A,88h           ;установить режимы портов 02-вывод, 03-вывод
      OUT 00              ;мл. 4 бита порта 01 - вывод, старшие 4 бита - ввод

```



```

MVI C,5          ;установить время 5 секунд и вызвать подпр.
CALL Time        ;задержки (пауза для показа квадратов)
;
; Переключение цвета фона экрана цветами из установленной таблицы цветов.
;
LXI H,Color      ;установить в "HL" адрес цвета фона
LXI D,COLOR15    ;установить в "DE" адрес последнего цвета таблицы
MVI B,15         ;в "B" - счетчик количества переключаемых цветов
Fon1: LDAX D      ;взять байт цвета из таблицы и
MOV M,A          ;записать по адресу цвета фона экрана
MVI C,01         ;время задержки при переключении = 1 сек
CALL Time        ;вызвать П/П задержки для лучшего восприятия
DCX D            ;перейди к следующему цвету
DCR B            ;все цвета перебраны?
JNZ Fon1         ;если нет - продолжать
MVI M,00         ;вернуть фону прежний черный цвет
;
; Мигание бордюра цветами таблицы
;
MVI A,15         ;установить в "A" номер последнего цвета таблицы
Bord1: STA Border ;записать его в ячейку цвета бордюра
MVI C,01         ;время задержки = 1 сек
CALL Time        ;вызвать П/П задержки
DCR A            ;все цвета таблицы перебраны?
JNZ Bord1        ;если нет - продолжать
MVI A,15         ;вернуть бордюру цвет 15
STA Border       ;
;
;-----
;Управление скроллингом экрана.
;-----
;
;Движение экрана вверх
;
ScUp: LDA Scroll  ;взять номер начальной строки экрана
DCR A            ;уменьшить его
STA Scroll       ;затем снова записать в "Scroll"
HLT              ;Останов процессора до следующего прерывания
HLT              ;(так можно формировать задержки кратные 1/50 с)
ORA A            ;полная прокрутка экрана?
JNZ ScUp         ;если нет - продолжать
;
;Движение экрана вниз
;
ScDown: LDA Scroll ;все аналогично ScUp, только
INR A            ;здесь нужно увеличивать номер
STA Scroll       ;
HLT              ;
HLT              ;
ORA A            ;
JNZ ScDown       ;
;
; Некоторых начинающих программистов может удивить использование команды HLT.
;Да, это еще одна приятная особенность использования прерывания. В тех ПЭВМ,
;где аппаратное (т.е. вызываемое схемой компьютера) прерывание не реализовано,
;эта команда "убила" бы программу и компьютер бы "замер". Здесь же "смерть"
;продлится недолго - до следующего прерывания, т.е. максимум 1/50 секунды.
; С помощью команды HLT можно не только формировать короткие задержки, но и
;организовывать счетчик времени в другом месте программы, а не только в под-
;программе обработки прерывания - например, это использует П/П "Time".
; Но если Вы запретили прерывания на какое-то время командой DI, то после
;этого HLT лучше не делать - выйти из такого "мертвого" останова можно будет
;только по "ВВОДУ" или "СБРОСУ" "Вектора"! После временного запрета прерывания
;обязательно должна следовать команда его разрешения - EI.
;
;Мигание индикатором "РУС/LAT"
;
MVI B,08         ;в регистр "B" - счетчик миганий
Rus: MVI A,08     ;зажечь светодиод индикатора "РУС/LAT" путем записи
STA IndRus       ;в ячейку "IndRus" кода 08 (это как раз 1 в третьем бите)
MVI C,01         ;задержка 1 секунда

```

```

CALL Time          ;
MVI A,00           ;погасить индикатор
STA IndRus         ;
MVI C,01           ;задержка 1 секунда
CALL Time          ;
DCR B              ;количество миганий отработано?
JNZ Rus            ;если НЕТ - продолжать
JMP Voice0         ;иначе - перейти на генерацию звука

```

; Теперь о программировании звука: Из описания ПЭВМ "Вектор-06ц" и некоторых рекламных программ следует, что "Вектор" наделен трехголосным синтезатором звука. Пусть термин "синтезатор" не пугает Вас - это сказано скорее в рекламных целях? на самом деле все гораздо проще. В качестве "синтезатора" использован обычный трехканальный таймер 580ВИ53, который при генерации звука работает как обычный делитель частоты с изменяемым коэффициентом деления.

; Теперь подробнее: имеется три отдельных делителя частоты. входы всех трех делителей соединены вместе и на них, в схеме "Вектора" всегда поступает частота 1.5 МГц (1 500 000 Гц). Выходы всех трех делителей объединены через резисторный смеситель и подключены к динамику и выходу на магнитофон. Чтобы получить нужную Вам частоту звука, необходимо записать в один из делителей число (коэф. деления), чтобы после деления на него 1 500 000 получилась именно эта частота:

Коэффициент деления = 15000000/частота (Гц)

; Эти делители называются каналами - 0-й,1-й,2-й. Для обращения к этим каналам в "Векторе" выделены следующие адреса портов:

0-й канал - порт 0Bh, 1-й канал - порт 0Ah, 2-й канал - порт 09h

; Для управления (выбора канала) служит порт 08h. Приведем пример генерации трехголосного звука, с поочередным включением каналов (голосов) и затем их поочередным выключением. Как и при установке других режимов ПЭВМ, выделим для каждого канала отдельные ячейки памяти, в которые можно будет записывать соответствующие им коэффициенты деления (для удобства записывайте коэфф. деления в десятичном виде, т.е. без "h" в конце числа).

```

;
Sound0: DW 1500      ;коэф.для 0-го канала (1000Гц)
Sound1: DW 0750      ;коэф.для 1-го канала (2000Гц)
Sound2: DW 0500      ;коэф.для 2-го канала (3000Гц)
;
Voice0: MVI A,36h     ;задать режим 3 (мл.байт,ст.байт) для канала 0
        OUT 08        ;записать в регистр режима
        LHLD Sound0   ;взять коэффициент деления для 0-го канала из "Sound0"
        MOV A,L       ;записать сначала младший байт в
        OUT 0Bh       ;делитель 0-го канала
        MOV A,H       ;затем старший байт
        OUT 0Bh       ;в тот же канал

```

; После этого появится один звук (один голос) с частотой 1000 Гц. Для того, чтобы услышать только его, сделаем задержку до появления следующего звука:

```

;
        MVI C,03      ;задержка на 3 секунды
        CALL Time      ;

```

; Аналогично включаются остальные каналы, меняются только адреса портов и значение, которое заносится в порт 08h (регистр режима) для выбора канала.

```

;
Voicel: MVI A,76h     ;включить 1-й канал
        OUT 08        ;
        LHLD Sound1   ;
        MOV A,L       ;
        OUT 0Ah       ;
        MOV A,H       ;
        OUT 0Ah       ;

```

```

;
        MVI C,03      ;задержка до появления следующего звука
        CALL Time      ;

```

```

Voicel: MVI A,0B6h    ;включить 2-й канал
        OUT 08        ;

```

```

    LHL D Sound2      ;
    MOV  A,L          ;
    OUT  09h          ;
    MOV  A,H          ;
    OUT  09h          ;
;
    MVI  C,05          ;задержка, чтобы услышать все 3 голоса
    CALL Time          ;
;
;   Выключение канала производится записью в порт того же значения, которое
;   записывалось для включения данного канала, но без записи в канал коэф.деления.
;
OfVoc0: MVI  A,36h      ;выключение 0-го канала
        OUT  08        ;
;
        MVI  C,03      ;задержка до отключения следующего канала
        CALL Time      ;
;
OfVoc1: MVI  A,76h      ;выключение 1-го канала
        OUT  08        ;
;
        MVI  C,03      ;задержка
        CALL Time      ;
;
OfVoc2: MVI  A,0B6h     ;выключение 0-го канала
        OUT  08        ;
;
;   На этом демонстрационная программа заканчивается и, чтобы вернуться в
;   Редактор-Ассемблер, надо сделать переход на адрес 0100h:
;
        JMP  0100h      ;вернуться в Редактор-Ассемблер
;
;=====
;Используемые подпрограммы:
;=====
;
;Вывод квадрата из 8-ми байт
;
Kvadr0: MVI  C,08        ;в регистр "C" - счетчик байт
Kvadr1: MVI  M,0FFh      ;записать по адресу "HL" все единицы
        INR  L           ;следующий байт для вывода
        DCR  C           ;все 8 байт квадрата записаны в видео ОЗУ?
        JNZ  Kvadr1      ;если НЕТ - продолжать
        RET              ;иначе - выйти из п/п вывода квадратов
;
;Подпрограмма задержки
;
Time:   PUSH B           ;сохранить значение пары "BC" в стеке
Time2:  MVI  B,50        ;в рег. "B" - счетчик прерываний (каждую 1/50 с)
Time1:  HLT              ;ожидать прерывание
        DCR  B           ;секунда обработана? (прошло 50 прерываний?)
        JNZ  Time1       ;если НЕТ - продолжать подсчет прерываний
        DCR  C           ;заданные секунды отработаны?
        JNZ  Time2       ;если НЕТ - продолжать
        POP  B           ;вернуть значение "BC"
        RET              ;и выйти из п/п задержки
;
;   На этом весь об'ем демонстрационной части программы заканчивается. Здесь
;   специально не даны примеры использования клавиш, т.к., согласно описанию,
;   программа после запуска должна отработать и вернуться в Редактор-Ассемблер.
;   Но Вы легко можете проверить работу клавиш, если сами дополните программу
;   существующими модулями, например, такого типа:
;
TstKey: LDA  KeyKod      ;взять код нажатой клавиши и проверить:
        CPI  Up          ;это код клавиши "курсор вверх" ?
        JZ   UpScrn      ;и если ДА - то что-то выполнить (например скрол.вверх)
        CPI  Down        ;это код клавиши "курсор вниз" ?
        JZ   DwScrn      ;если ДА - скроллинг вниз
        CPI  LF           ;была нажата клавиша "ПС" ?
        JZ   0100h       ;тогда вернуться в Редактор-Ассемблер
        JMP  TstKey       ;если ни одна из этих трех клавиш не нажата - пров.снова

```



```

UpScrn: HLT                ;сделать задержку по ожиданию прерывания, чтобы
        HLT                ;экран не двигался слишком быстро
        LDA Scroll         ;взять номер строки начала отображения и
        DCR A              ;уменьшить его
        STA Scroll         ;записать уменьшенный номер в ячейку "Scroll"
        JMP TstKey         ;снова вернуться на опрос клавиатуры
DwScrn: HLT                ;скроллинг вниз аналогично, но
        HLT                ;с увеличением номера строки
        LDA Scroll         ;
        INR A              ;
        STA Scroll         ;
        JMP TstKey         ;
;
;   Надо заметить что клавиши "РУС/LAT", "УС", "СС" не входят в матрицу основной
;клавиатуры 8x8, а подключены отдельно к 7-му, 6-му и 5-му биту порта 01 соот-
;ветственно. Эти биты порта 01 больше ничем не управляют и поэтому опрос этих
;клавиш можно производить в любое время, а не только по прерыванию. Например:
;
        IN 01              ;взять байт из порта 01
        ANI 80h            ;маска для 7-го бита (проверка клавиши "РУС/LAT")
        JZ 0100h           ;вернуться в Редактор-Ассемблер если нажата.
;
;   В тексте основной программы также не оговорены еще некоторые адреса портов
;и значения некоторых битов (чтоб не загромождать описание и сделать его более
;понятным). Приведем их значения для полноты информации.
;
;   4-й бит порта 02 - переключатель формата 256 или 512 точек на экране по
;горизонтали. Если он установлен в лог."0" - то 256 точек и справедливо все, о
;чем говорится в этом описании (это касается цветов). Если лог."1" - то 512
;точек, но при этом на экране могут быть только 4 цвета и будет усложнен вывод
;изображения на экран, т.к. две соседние точки экрана принадлежат разным плос-
;костям. Этот режим не рассматривался в описании с целью упрощения, но его
;легко включить, если установить этот бит в ячейке "Border" т.к. цвет бордюра
;занимает биты 0-3 этой ячейки (от 0 до 15), а в конце программы прерывания
;байт из ячейки "Border" выставляется на линии порта 02.
;
;   4-й бит порта 01 - вход для ввода информации с магнитофона
;   0-й бит порта 01 - выход для вывода информации на магнитофон и через него
;                       же можно выводить программно-формируемый звук без ис-
;                       пользования таймера
;   порт 04h - регистр режима внешнего параллельного адаптера
;   порт 05h - канал "С"
;   порт 06h - канал "В"
;   порт 07h - канал "А"
;
;       Линии каналов А,В,С этого адаптера, выведены через раз'ем "ПУ"
;       На тот же раз'ем выведен канал "0" таймера
;   порт 10h - управление переключением ОЗУ электронного диска (КВАЗИДИСКА)
;   порт 18h - регистр данных контроллера НГМД
;   порт 19h - регистр сектора дорожки контроллера НГМД
;   порт 1Ah - регистр дорожки контроллера НГМД
;   порт 1Bh - регистр команд и статуса контроллера НГМД
;   порт 1Ch - регистр выбора и управления контроллера НГМД
;
;   В конце хотелось бы отметить, что данное описание - это отнюдь не учебник
;по программированию на языке Ассемблера. В нем дан лишь справочный материал
;по программной установке режимов ПЭВМ "Вектор-06ц" с примерами, в которых для
;наглядности использованы простейшие команды Ассемблера. Для изучения языка
;Ассемблера и создания более сложных программ лучше пользоваться следующей ли-
;тературой:
;
; (1) Р.Токхайм
;     "Микропроцессоры", Москва, ЭнергоАтомИздат, 1988
; (2) Л.Левенталь, У.Сэйвилл
;     "Программирование на языке Ассемблера для микропроцессоров 8080 и 8085"
;     перевод А.А.Батнера, Москва, Радио и Связь, 1987
; (3) А.Гуртовцев, С.Гудымено
;     "Программы для микропроцессоров", Минск, Вышэйшая Школа, 1989
; (4) Г.Зеленко, В.Панов, С.Попов
;     "Радиолюбителю о микропроцессорах и микроЭВМ", журналы "Радио" 1982-1983
;
; А также внимательно изучать и пользоваться описанием программ для "Вектора":

```

; "Редактор-Ассемблер" и "Монитор-Отладчик" П/О СчетМаш г.Кишинев.

;

;

;

;

# ПРИЛОЖЕНИЕ 1.

; Таблица машинных кодов программы, приведенной в данном описании.

; (коды можно ввести пользуясь МОНИТОРОМ-ОТЛАДЧИКОМ и затем  
; запустить директивой "G100")

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

Таблица значений символьных меток программы  
(для изменения параметров в машинных кодах)

01C8 Bord1	011C Border	011D Color	011E COLR1	0127 COLR10
0128 COLR11	0129 COLR12	012A COLR13	012B COLR14	012C COLR15
011F COLR2	0120 COLR3	0121 COLR4	0122 COLR5	0123 COLR6
0124 COLR7	0125 COLR8	0126 COLR9	01B8 Fon1	012F IndRus
0131 Init	0140 Init1	012E KeyKod	0267 Kvadr1	0265 Kvadr0
0188 Next	024C OfVoc0	0255 OfVoc1	025E OfVoc2	018B Pro
01F3 Prery	0185 Progrm	0130 Rele	01F5 Rus	01E6 ScDown
012D Scroll1	01D9 ScUp	0210 Sound0	0212 Sound1	0214 Sound2
0100 Start	026F Time	0272 Time1	0270 Time2	0216 Voice0
0228 Voice1	023A Voice2			