БАЙТ

Информационно-рекламный выпуск
для пользователей БПЭВМ «Вектор-06Ц»

Сборник номеров
01 – 33

Редактор сборника: Вячеслав Славинский
Из материалов архива Александра Тимошенко

С.-Петербург
2007

Легальный статус

Оглавление

```
    ВЕКТОР-06Ц                  ВЕКТОР-06Ц                  ВЕКТОР-06Ц
////////////////////////////////////////////////////////////////////////
/                                    *                                 /
/      ИЮНЬ         *****      *      *   *    *****          *       /
/                   *         * *     *  **      *           **       /
/     1991 Г.       ****     *****    * * *      *    ***   * *       /
/                   *   *    *   *    **  *      *            *       /
/      КИРОВ        *****    *   *    *   *      *            *       /
/                                                                     /
/                     ИНФОРМАЦИОННО-РЕКЛАМНЫЙ ВЫПУСК                  /
/                  ДЛЯ ПОЛЬЗОВАТЕЛЕЙ БПЭВМ "ВЕКТОР-06Ц"               /
////////////////////////////////////////////////////////////////////////
```

УЧИТЫВАЯ ВОЗРОСШЕЕ КОЛИЧЕСТВО ВЛАДЕЛЬЦЕВ БПЭВМ "ВЕКТОР-06Ц", А ТАКЖЕ БОЛЬШОЙ ИНТЕРЕС КО ВСЕМУ, ЧТО КАСАЕТСЯ ЭТОГО КОМПЬЮТЕРА, НАШ МАЛЕНЬКИЙ КОЛЛЕКТИВ НАЧИНАЕТ ПУБЛИКАЦИЮ ИНФОРМАЦИОННОГО ЛИСТКА "БАЙТ". В НЕМ БУДУТ ПЕЧАТАТЬСЯ НОВОСТИ , ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ ПО КОМПЬЮТЕРУ, ОБЬЯВЛЕНИЯ, АЛГОРИТМЫ, РЕКЛАМА И.Т.Д.

ПРЕКРАСНО СОЗНАВАЯ , ЧТО НАШЕ ПОЛИГРАФИЧЕСКОЕ ИСПОЛНЕНИЕ ОСТАВЛЯЕТ ЖЕЛАТЬ ЛУЧШЕГО, МЫ ПРИЛОЖИМ ВСЕ УСИЛИЯ, ЧТОБЫ КОМПЕНСИРОВАТЬ ЕГО НЕДОСТАТКИ ИНТЕРЕСНЫМ И ПОЛЕЗНЫМ СОДЕРЖАНИЕМ.

ДЕЛО, КОТОРОЕ МЫ НАЧАЛИ, ЯВЛЯЕТСЯ ДЛЯ НАС НОВЫМ, НЕТ ЕЩЕ НЕОБХОДИМОГО ОПЫТА, И МЫ ОТКРЫТЫ ДЛЯ КОНТАКТОВ С ТЕМИ, КТО МОЖЕТ НАС В ЧЕМ-ТО ПОПРАВИТЬ И ЧТО-ТО ПОДСКАЗАТЬ.

СРЕДИ НАС НЕТ ПРОГРАММИСТОВ-ПРОФЕССИОНАЛОВ. МЫ ТАКИЕ ЖЕ ЛЮБИТЕЛИ КАК И ВЫ, УВАЖАЕМЫЕ НАШИ ЧИТАТЕЛИ. НО МЫ СЧИТАЕМ НЕОБХОДИМЫМ ПОДЕЛИТЬСЯ С ВАМИ ТОЙ ИНФОРМАЦИЕЙ, КОТОРОЙ ОБЛАДАЕМ, СВОИМИ "МАЛЕНЬКИМИ ОТКРЫТИЯМИ" В ОБЛАСТИ ПРОГРАММИРОВАНИЯ. МЫ ХОТИМ ОРГАНИЗОВАТЬ НА СТРАНИЦАХ НАШЕГО ИЗДАНИЯ ДИСКУССИОННЫЙ КЛУБ, ГДЕ КАЖДЫЙ МОЖЕТ ПОДЕЛИТЬСЯ СВОИМ ОПЫТОМ, РАССКАЗАТЬ О РЕЗУЛЬТАТАХ СВОЕГО ПОИСКА , ЗАДАТЬ ИНТЕРЕСУЮЩИЕ ЕГО ВОПРОСЫ, ПОМЕСТИТЬ РЕКЛАМУ ИЛИ ОБЬЯВЛЕНИЕ. НАМ ТАКЖЕ ИНТЕРЕСНО ЗНАТЬ, ЧТОБЫ ВЫ ХОТЕЛИ ЧИТАТЬ В СЛЕДУЮЩИХ ВЫПУСКАХ. НЕ ПОЛЕНИТЕСЬ НАПИСАТЬ СВОИ ВОПРОСЫ, В СЛЕДУЮЩИХ НОМЕРАХ МЫ ПОСТАРАЕМСЯ ОТВЕТИТЬ НА НИХ. ЕСЛИ ВЫ "РАСКОПАЛИ" ЧТО-ТО НЕОБЫЧНОЕ В ИГРАХ И ВООБЩЕ ПРОГРАММАХ (НАПРИМЕР, ЯЧЕЙКУ ЖИЗНИ ИЛИ АДРЕСА РАСПОЛОЖЕНИЯ ПОЛЕЗНОЙ ПОДПРОГРАММЫ), РАЗРАБОТАЛИ КАКИЕ-ЛИБО ТЕХНИЧЕСКИЕ ДОРАБОТКИ ДЛЯ КОМПЬЮТЕРА, ПИШИТЕ, НА КАЖДОЕ ТАКОЕ ПИСЬМО БУДЕТ ДАН ОТВЕТ И В СЛУЧАЕ ОПУБЛИКОВАНИЯ В "БАЙТЕ" ВЫПЛАЧЕН ГОНОРАР.

ПИШИТЕ ПО АДРЕСУ: Г. КИРОВ, А/Я 1248, "БАЙТ".

ЧИТАЙТЕ В СЛЕДУЮЩЕМ НОМЕРЕ: ОСОБЕННОСТИ НАЧАЛЬНОГО ЗАГРУЗЧИКА.ВОЗМОЖЕН ЛИ АВТОЗАПУСК ПРОГРАММ В КОДАХ.ПОСОБИЕ ДЛЯ НАЧИНАЮЩЕГО ХЭККЕРА ИЛИ КАК ВЗЛОМАТЬ ЧУЖУЮ ПРОГРАММУ.ТАБЛИЦА ЦВЕТОВ И ВИДЕО-ОЗУ.

СРЕДИ ПРОГРАММ ДЛЯ ПК "ВЕКТОР-06Ц" ПОЯВИЛИСЬ ИГРЫ, АДАПТИРОВАННЫЕ ДЛЯ ДЖОЙСТИКА , ПОДКЛЮЧАЕМОГО ЧЕРЕЗ РАЗЬЕМ ПУ. ПОДСОЕДИНИВ ДЖОЙСТИК ТАКИМ ОБРАЗОМ, ВЫ ПОЛУЧАЕТЕ РЯД ПРЕИМУЩЕСТВ ПО СРАВНЕНИЮ С ДЖОЙСТИКОМ, ПОДКЛЮЧАЕМЫМ ПАРАЛЛЕЛЬНО КЛАВИАТУРЕ: ВО-ПЕРВЫХ, НЕ НУЖНО ВСКРЫВАТЬ КОМПЬЮТЕР И ПРОИЗВОДИТЬ ВНУТРЕННЮЮ ПАЙКУ, ВО-ВТОРЫХ, ОПРОС ТАКОГО ДЖОЙСТИКА НА ПОРЯДОК ПРОЩЕ И БЫСТРЕЕ. В НАСТОЯЩЕЕ ВРЕМЯ ЦЕНТРОМ "БАЙТ" ПРОИЗВОДИТСЯ ДОРАБОТКА БОЛЬШИНСТВА СУЩЕСТВУЮЩИХ ПРОГРАММ ДЛЯ РАБОТЫ С ДЖОЙСТИКОМ. ПРИ ЭТОМ ЭТИ ПРОГРАММЫ БУДУТ РАБОТАТЬ ОДНОВРЕМЕННО И С КЛАВИАТУРОЙ, И С ДЖОЙСТИКОМ.

НАЗНАЧЕНИЕ ВЫВОДОВ ПУ: А0 - +5В,ОБЩИЙ ПРОВОД; А1 - СТРЕЛКА ВВЕРХ; А2 - СТРЕЛКА ВПРАВО; А3- СТРЕЛКА ВНИЗ; А4 - СТРЕЛКА ВЛЕВО; А5 - ПРОБЕЛ

ЗНАКОГЕНЕРАТОР В БЕЙСИКЕ.

В ОТЛИЧИЕ ОТ РАДИО-86РК, В КОТОРОМ ЗНАКОГЕНЕРАТОР ЗАШИТ В ПЗУ И ПОТОМУ НЕИЗМЕНЕН, ЗНАКОГЕНЕРАТОР НА ВЕКТОРЕ ЗАГРУЖАЕТСЯ ВМЕСТЕ С БЕЙСИКОМ И ДОСТУПЕН ДЛЯ МОДИФИКАЦИИ. ЭТО ДАЕТ МНОЖЕСТВО ИНТЕРЕСНЫХ ВОЗМОЖНОСТЕЙ ДЛЯ ПРОГРАММИСТА: ИЗМЕНЕНИЕ АЛФАВИТА, ВВОД НОВЫХ СИМВОЛОВ, СОЗДАНИЕ ПСЕВДОГРАФИКИ И.Т.Д.

ЗНАКОГЕНЕРАТОР - ЭТО НЕБОЛЬШОЙ БАНК ДАННЫХ, В КОТОРОМ ХРАНЯТСЯ СИМВОЛЫ, ВЫВОДИМЫЕ НА ЭКРАН. В БЕЙСИКЕ ОН РАСПОЛАГАЕТСЯ ПО АДРЕСАМ: 0 - 639. МОЖЕТ ВОЗНИКНУТЬ ВПЕЧАТЛЕНИЕ , ЧТО БЕЙСИК ПЕРЕПРОГРАММИРУЕТ КЛАВИШИ БЛК+СБР, ТАК КАК ОНИ СЛУЖАТ ДЛЯ ЗАПУСКА ПРОГРАММЫ С 0-ГО АДРЕСА, А ИМЕННО ПО ЭТИМ АДРЕСАМ РАСПОЛАГАЕТСЯ ЗНАКОГЕНЕРАТОР. ОДНАКО ИНТЕРПРЕТАТОР ПРОДОЛЖАЕТ БЛАГОПОЛУЧНО РАБОТАТЬ ПОСЛЕ НАЖАТИЯ НА ЭТИ КЛАВИШИ. ДЕЛО В ТОМ, ЧТО РЕАЛЬНО ЗНАКОГЕНЕРАТОР РАСПОЛАГАЕТСЯ ПО АДРЕСАМ 3920Н - 3В9FH, А БЕЙСИК ЭМУЛИРУЕТ (Т.Е. СОЗДАЕТ ВИДИМОСТЬ) ЧЕРЕЗ КОМАНДЫ PEEK И POKE, ЧТО ЗНАКОГЕНЕРАТОР РАСПОЛАГАЕТСЯ С 0-ГО АДРЕСА. СДЕЛАНО ЭТО ДЛЯ УНИФИКАЦИИ РАЗЛИЧНЫХ ВЕРСИЙ БЕЙСИКА.

КАЖДЫЙ СИМВОЛ В ЗНАКОГЕНЕРАТОРЕ ИМЕЕТ КОД. КОД - ЭТО , КАК БЫ, ПОРЯДКОВЫЙ НОМЕР СИМВОЛА, ОН НЕ ИЗМЕНЕН. ПО КОДУ ИНТЕРПРЕТАТОР ОПРЕДЕЛЯЕТ ОБЛАСТЬ ПАМЯТИ, ИЗ КОТОРОЙ НУЖНО ВЗЯТЬ ДАННЫЕ ДЛЯ ВЫВОДА НА ЭКРАН. САМ ЖЕ СИМВОЛ МОЖНО ПЕРЕПРОГРАММИРОВАТЬ КАК УГОДНО. КАЖДЫЙ СИМВОЛ ЗАНИМАЕТ 5 БАЙТ. АДРЕС СИМВОЛА ВЫЧИСЛЯЕТСЯ КАК 5*N, ГДЕ N - КОД СИМВОЛА. ВСЕГО В ЗНАКОГЕНЕРАТОРЕ БЕЙСИКА 128 СИМВОЛОВ. ДЛЯ ПЕРЕПРОГРАММИРОВАНИЯ СИМВОЛА НУЖНО НАРИСОВАТЬ НЕБОЛЬШУЮ ТАБЛИЦУ И ЗАПОЛНИТЬ ЕЕ 0 И 1:

```
  7  6  5  4  3  2  1  0
  ----------------------
  1  1  1  1  1  0  0  0      1   = 2^7+2^6+2^5+2^4+2^3 = 248
  0  0  1  0  0  1  0  0      2   = 2^5+2^2 = 36
  0  0  1  0  0  0  1  0      3   = 2^5+2^1 = 34
  0  0  1  0  0  1  0  0      4   = 2^5+2^2 = 36
  1  1  1  1  1  0  0  0      5   = 2^7+2^6+2^5+2^4+2^3 = 248
  ----------------------
```

СИМВОЛ В ТАБЛИЦЕ ПОВЕРНУТ НА 90 ГРАДУСОВ ВПРАВО, А ПОДПРОГРАММА ВЫВОДА СИМВОЛА НА ЭКРАН РАЗВОРАЧИВАЕТ ЕГО В НОРМАЛЬНОЕ ПОЛОЖЕНИЕ. СДЕЛАНО ЭТО ДЛЯ ЭКОНОМИИ ПАМЯТИ. ТАКИМ ОБРАЗОМ, ЗАПОЛНИВ ТАБЛИЦУ , МЫ ПОЛУЧИМ 5 БАЙТОВ, КОТОРЫЕ И НУЖНО ЗАНЕСТИ В ПАМЯТЬ: POKE 0,248,36,34,36,248. РЕЗУЛЬТАТ МОЖНО ПОСМОТРЕТЬ ОПЕРАТОРОМ PRINT CHR$(0).

КРОМЕ ОБЫЧНЫХ СИМВОЛОВ, ЦИФР И БУКВ СУЩЕСТВУЮТ ЕЩЕ И ПСЕВДОГРАФИЧЕСКИЕ СИМВОЛЫ. ОБЫЧНО ИХ ПОЯВЛЕНИЕ ДОСТИГАЕТСЯ НАЖАТИЕМ КЛАВИШИ УС И ЛЮБОЙ ДРУГОЙ.

НАПРИМЕР: УС+Q - , УС+R - , И.Т.Д.

ЧТОБЫ ПОЛУЧИТЬ СИМВОЛЫ С КОДАМИ ОТ 0 ДО 30, ВОСПОЛЬЗУЙТЕСЬ СЛЕДУЮЩЕЙ ТАБЛИЦЕЙ:

ДЕСЯТ. КОД	00	01	02	03	04	05	06	07	08	09	10	11
УС + КЛАВИША	@	A	B	C	D	E	F	G	H	I	J	K

12	13	14	15	16	17	18	19	20	21	22	23
L	M	N	O	P	Q	R	S	T	U	V	W

24	25	26	27	28	29	30
X	Y	Z	[	\	]	^

УС+S ОЗНАЧАЕТ ОДНОВРЕМЕННОЕ НАЖАТИЕ КЛАВИШ УС И S.

НЕКОТОРЫЕ КОМБИНАЦИИ ЯВЛЯЮТСЯ УПРАВЛЯЮЩИМИ: УС+L ДЕЙСТВУЕТ КАК 'СТРЕЛКА ВЛЕВО-ВВЕРХ', УС+Z - 'СТРЕЛКА ВНИЗ', УС+A - 'F2' И.Т.Д. НЕКОТОРЫЕ КОМБИНАЦИИ ЯВЛЯЮТСЯ ПУСТЫМИ ИЛИ ГЕНЕРИРУЮТ ЗВУК. НАЖАТИЕ КЛАВИШ УС+M ИЛИ УС+J ПОСЛЕ НАБРАННОЙ КОМАНДЫ , НАПРИМЕР 'RUN' , ПЕРЕВОДЯТ ИНТЕРПРЕТАТОР В РЕЖИМ ОЖИДАНИЯ, ПОСЛЕ НАЖАТИЯ ЛЮБОЙ КЛАВИШИ КОМАНДА ВЫПОЛНИТСЯ (В ДАННОМ СЛУЧАЕ ЗАПУСТИТСЯ ПРОГРАММА).

ХОЧЕТСЯ ПРЕДОСТЕРЕЧЬ ЧИТАТЕЛЕЙ ОТ НЕОСТОРОЖНОГО НАЖАТИЯ НА КЛАВИШИ УС+D. В ЭТОМ СЛУЧАЕ ИНТЕРПРЕТАТОР КАК БЫ 'СХОДИТ С УМА' - ВЫДАЕТ НА ЭКРАН СОВСЕМ НЕ ТО, ЧТО ВЫ ХОТИТЕ. И ОСТАЕТСЯ ЛИШЬ ПЕРЕГРУЗИТЬ БЕЙСИК ЗАНОВО. ТАКАЯ СИТУАЦИЯ ЧАСТО ВСТРЕЧАЕТСЯ ПРИ НАБОРЕ ОПЕРАТОРА LINE ПУТЕМ НАЖАТИЯ УС+СС+D . ПРИ НЕПЛОТНОМ НАЖАТИИ НА СС ПОЛУЧАЕМ ОПИСАННУЮ ВЫШЕ РЕАКЦИЮ КОМПЬЮТЕРА. ДЛЯ ИЗБЕЖАНИЯ ТАКИХ НЕПРИЯТНОСТЕЙ СЛЕДУЕТ ПОЛЬЗОВАТЬСЯ КЛАВИШАМИ АР2,Т.

НЕ ВСЕ СИМВОЛЫ МОЖНО ПОЛУЧИТЬ НАЖАТИЕМ КЛАВИШ. НАПРИМЕР, СИМВОЛ С КОДОМ 03, СООТВЕТСТВУЮЩИЙ КЛАВИШЕ 'F4', МОЖНО ПОЛУЧИТЬ, ТОЛЬКО ИСПОЛЬЗУЯ PRINT CHR$(3).

СЛЕДУЮЩАЯ ПРОГРАММА ПОЗВОЛЯЕТ ПРОСМОТРЕТЬ ВСЕ СИМВОЛЫ:

```
 90 FOR N=1 TO 127 : PRINT CHR$(N)
100 IF INKEY$="" THEN 100
110 NEXT N
```

ПРИ РАСПЕЧАТКЕ СИМВОЛОВ С КОДАМИ 8, 24, 31, И.Т.Д. МЫ СИМВОЛОВ НЕ ПОЛУЧИМ. ЭТИ КОДЫ ЯВЛЯЮТСЯ УПРАВЛЯЮЩИМИ ДЛЯ ИНТЕРПРЕТАТОРА И РЕАЛИЗУЮТСЯ СООТВЕТСТВЕННО СДВИГОМ КУРСОРА ВЛЕВО , ВПРАВО НА ОДНУ ПОЗИЦИЮ , СТИРАНИЕМ ЭКРАНА И.Т.Д. ОДНАКО ЗНАКОГЕНЕРАТОР В МЕСТАХ , СООТВЕТСТВУЮЩИХ ЭТИМ КОДАМ, ВСЕ ЖЕ СОДЕРЖИТ СИМВОЛЫ. НАПРИМЕР, В УЧАСТКЕ ПАМЯТИ, СООТВЕТСТВУЮЩЕМ КОДУ 8 (СТРЕЛКА ВЛЕВО), ДЕЙСТВИТЕЛЬНО РАСПОЛАГАЕТСЯ СТРЕЛКА, НАПРАВЛЕННАЯ ВЛЕВО И.Т.Д. В ЭТОМ МОЖНО УБЕДИТЬСЯ С ПОМОЩЬЮ ОПЕРАТОРА PEEK.

ХОЧЕТСЯ ОТМЕТИТЬ ИНТЕРЕСНУЮ ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ СИМВОЛА С КОДОМ 07. КОМАНДА PRINT CHR$(7) ВМЕСТО ВЫВОДА СИМВОЛА ГЕНЕРИРУЕТ ЗВУК. ЭТОТ ЖЕ ЭФФЕКТ

ДОСТИГАЕТСЯ ПРИ НАЖАТИИ НА КЛАВИШИ УС+G. В ОТЛИЧИЕ ОТ ДРУГИХ ЗВУЧАЩИХ КЛАВИШ (НАПРИМЕР, F1) БЕЙСИК ЗАПОМИНАЕТ ЭТОТ СИМВОЛ. ЭТО МОЖНО ИСПОЛЬЗОВАТЬ ДЛЯ ВЫДЕЛЕНИЯ В ТЕКСТЕ ПРОГРАММЫ КАКИХ-ЛИБО ЕЕ ФРАГМЕНТОВ. НЕОБХОДИМО ТОЛЬКО В НУЖНОМ МЕСТЕ (НАПРИМЕР, В НАЧАЛЕ ПРОГРАММНОЙ СТРОКИ) НАЖАТЬ УС+G, РАЗДАСТСЯ СИГНАЛ, КУРСОР ОСТАНЕТСЯ НА ПРЕЖНЕМ МЕСТЕ, НО ИНТЕРПРЕТАТОР ЗАПОМНИТ СИМВОЛ, ЗАТЕМ ВВЕСТИ СТРОКУ В ПАМЯТЬ И 'ВК'. ПРИ ПРОСМОТРЕ ПРОГРАММЫ КОМАНДОЙ LIST ИНТЕРПРЕТАТОР , ДОЙДЯ ДО ЭТОГО МЕСТА ПРИОСТАНОВИТ ЛИСТИНГ, ПРОЗВУЧИТ СИГНАЛ, И ЛИСТИНГ ВОЗОБНОВИТСЯ.

КАЖДЫЙ СИМВОЛ НА ЭКРАН ЗАНИМАЕТ ПРЯМОУГОЛЬНИК РАЗМЕРАМИ 6*8 (X*Y) ТОЧЕК. ШЕСТАЯ ВЕРТИКАЛЬНАЯ ПОЛОСКА - ПУСТАЯ, ОНА СЛУЖИТ ДЛЯ РАЗДЕЛЕНИЯ СИМВОЛОВ ДРУГ ОТ ДРУГА. ДЛЯ ТОГО, ЧТОБЫ ВЫВОДИМЫЙ ТЕКСТ БЫЛ ОТЦЕНТРИРОВАН МОЖНО ВОСПОЛЬЗОВАТЬСЯ ФОРМУЛОЙ: X=(42-L)/2, ГДЕ L - ДЛИНА СТРОКИ. ДЛЯ СИМВОЛОВ УВЕЛИЧЕННОГО ФОРМАТА ФОРМУЛА ПРИМЕТ ВИД: X=(256-L*K*6)/2 , ГДЕ L - ДЛИНА СТРОКИ, K – КОЭФФИЦИЕНТ УВЕЛИЧЕНИЯ СИМВОЛОВ ПО ОСИ X.

ПРИ ВЫВОДЕ ТЕКСТА НЕСКОЛЬКО РАЗ В ОДНО И ТО ЖЕ МЕСТО ПРОИСХОДИТ СКРОЛЛИНГ ЭКРАНА. ПРИ ИСПОЛЬЗОВАНИИ PRINT ...; В ЦИКЛЕ ТАКЖЕ ВОЗНИКАЮТ РАЗЛИЧНЫЕ СБОИ. ЧТОБЫ ЭТОГО НЕ ПРОИСХОДИЛО, СЛЕДУЕТ ВОСПОЛЬЗОВАТЬСЯ ПРОМЕЖУТОЧНЫМ ОПЕРАТОРОМ CUR 0,24 ИЛИ PRINT AT 0,24 "".

А ТЕПЕРЬ НЕСКОЛЬКО ПРИМЕРОВ ИСПОЛЬЗОВАНИЯ ТЕКСТА НА ЭКРАНЕ:

1. "ОБЬЕМНЫЕ БУКВЫ"

```
100 A$="ВЕКТОР-06Ц":FOR N=1 TO 4
110 COLOR N:PLOT 40-N,220-N,2:LINE 3,3,BS
120 PRINT A$:NEXT N
```

2. БУКВЫ С "КАЕМОЧКОЙ"

```
100 A$="ВЕКТОР-06Ц":X=39:Y=221:COLOR 15:CLS
110 FOR I=0 TO 1:FOR J=0 TO 1:PLOT X-2*I,Y-2*J,2
115 LINE 3,3,BS:PRINT A$:NEXT J:NEXT I
120 COLOR 4:PLOT X-1,Y-1,2:LINE 3,3,BS:PRINT A$
```

РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ

/ ПО МАТЕРИАЛАМ ZX-РЕВЮ N1 1991 /

МЫ ЗНАЕМ, ЧТО ОСОБУЮ СЛОЖНОСТЬ ДЛЯ НАЧИНАЮЩИХ ПРЕДСТАВЛЯЕТ РАБОТА С ОПЕРАТОРАМИ БЕЙСИКА READ И DATA, НЕ ГОВОРЯ УЖЕ О RESTORE. ПРОФЕССИОНАЛЫ, ЕСЛИ ВСПОМНЯТ СВОИ ПЕРВЫЕ ШАГИ, КОНЕЧНО С НАМИ СОГЛАСЯТСЯ. ПРИЧИН ЗДЕСЬ ДВЕ. ВО-ПЕРВЫХ, РАБОТА ЭТИХ ОПЕРАТОРОВ НЕОЧЕВИДНА, ТО ЕСТЬ ТРЕБУЕТ СПЕЦИАЛЬНЫХ РАЗЬЯСНЕНИЙ, А ВО-ВТОРЫХ , В БЕЙСИКЕ ЕСТЬ ЗАМЕЧАТЕЛЬНЫЙ ОПЕРАТОР INPUT , КОТОРЫЙ ПОЗВОЛЯЕТ УДОБНО ОРГАНИЗОВЫВАТЬ ВВОД ДАННЫХ, И НАЧИНАЮЩИЕ ПРЕДПОЧИТАЮТ ПОЛЬЗОВАТЬСЯ ИМ.

СЕГОДНЯ МЫ ПРЕДЛАГАЕМ ВАШЕМУ ВНИМАНИЮ НЕСЛОЖНУЮ ПРОГРАММУ , ПОЗВОЛЯЮЩУЮ ВОСПРОИЗВЕСТИ НА КОМПЬЮТЕРЕ НЕСКОЛЬКО ПРОСТЫХ МЕЛОДИЙ. РАБОТА ЗВУКОВОГО ДИНАМИКА ЗДЕСЬ ОСНОВЫВАЕТСЯ НА ИСПОЛЬЗОВАНИИ ОПЕРАТОРА BEEP M,N. ПАРАМЕТР M ОЗНАЧАЕТ ПРОДОЛЖИТЕЛЬНОСТЬ ЗВУКОВОГО СИГНАЛА В СЕКУНДАХ , А ПАРАМЕТР N - ВЫСОТУ ТОНА.

ЭТИ ПАРАМЕТРЫ ВВОДЯТСЯ С ПОМОЩЬЮ ОПЕРАТОРОВ READ И DATA. ОНИ ХРАНЯТСЯ В СТРОКАХ DATA И СЧИТЫВАЮТСЯ ОТТУДА ОПЕРАТОРОМ READ. ОПЕРАТОРЫ READ И DATA РАБОТАЮТ В ПАРЕ. КОГДА ПРОГРАММА ВСТРЕЧАЕТ ОЧЕРЕДНОЙ ОПЕРАТОР READ, НАПРИМЕР , READ Q,R,S ,ОНА ОБРАЩАЕТСЯ К БЛИЖАЙШЕМУ НЕИСПОЛЬЗОВАННОМУ ОПЕРАТОРУ DATA И ВВОДИТ ИЗ НЕГО СТОЛЬКО ЧИСЕЛ, СКОЛЬКО НУЖНО ОПЕРАТОРУ READ, В НАШЕМ СЛУЧАЕ - ТРИ (Q,R И S). ЕСЛИ В СТРОКЕ DATA СТОЛЬКИХ ЧИСЕЛ НЕТ, ТО ПРОГРАММА

ИДЕТ К СЛЕДУЮЩЕМУ БЛОКУ DATA И ЧИТАЕТ ДАННЫЕ ИЗ НЕГО. ПОСЛЕ ТОГО, КАК ДАННЫЕ ПРОЧИТАНЫ , ОНИ СЧИТАЮТСЯ ИСПОЛЬЗОВАННЫМИ И БОЛЬШЕ К НИМ ТАК ПРОСТО НЕ ВЕРНУТЬСЯ.

ПОСЛЕ СОВМЕСТНОЙ РАБОТЫ READ И DATA ВОЗМОЖНЫ ТРИ ВАРИАНТА. ПЕРВЫЙ, КОГДА СКОЛЬКО ДАННЫХ ПОТРЕБОВАЛОСЬ ДЛЯ READ, СТОЛЬКО И НАШЛОСЬ В DATA. ЭТО НОРМАЛЬНЫЙ ВАРИАНТ РАБОТЫ, ВСЕ В ПОРЯДКЕ. БЫВАЕТ , ЧТО ОПЕРАТОРЫ READ УЖЕ ВЗЯЛИ ВСЕ, ЧТО ИМ НАДО, А В СТРОКАХ DATA ЕЩЕ НЕ ВСЕ ДАННЫЕ ИСПОЛЬЗОВАНЫ. ЭТО ВОЗМОЖНО, НИКАКОЙ ОШИБКИ ЗДЕСЬ НЕТ. ХУЖЕ, КОГДА READ ПРОСИТ КАКИЕ-ТО ДАННЫЕ, А СТРОКИ DATA УЖЕ ИСЧЕРПАНЫ. В ЭТОМ СЛУЧАЕ ПОЯВЛЯЕТСЯ СООБЩЕНИЕ ОБ ОШИБКЕ. СКОРЕЕ ВСЕГО ПРИЧИНА ОШИБКИ В ТОМ, ЧТО ВЫ ПО НЕВНИМАТЕЛЬНОСТИ ПРОПУСТИЛИ КАКОЕ-ТО ЧИСЛО, НАБИРАЯ СТРОКУ DATA.

ЕСЛИ ВЫ ПОДУМАЛИ, ЧТО ДАННЫЕ В СТРОКАХ DATA МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ТОЛЬКО РАЗ И ПОТОМ ОНИ НЕДОСТИЖИМЫ, ТО ЭТО НЕ СОВСЕМ ТАК. ДЕЛО МОЖЕТ ПОПРАВИТЬ ОПЕРАТОР RESTORE NNNN, ГДЕ NNNN - НОМЕР СТРОКИ, НАПРИМЕР RESTORE 2500. ЭТОТ ОПЕРАТОР "ВОССТАНАВЛИВАЕТ" СТРОКИ DATA, НАЧИНАЯ СО СТРОКИ NNNN, ПОСЛЕ ЧЕГО СЛЕДУЮЩИЙ ОПЕРАТОР READ БУДЕТ ЧИТАТЬ ДАННЫЕ ИЗ ЭТОЙ СТРОКИ И ПОСЛЕДУЮЩИХ И.Т.Д.

С ПОМОЩЬЮ ОПЕРАТОРА RESTORE МОЖНО НЕ ТОЛЬКО "ВОССТАНАВЛИВАТЬ" СТРОКИ DATA, НО С ТЕМ ЖЕ УСПЕХОМ И ВООБЩЕ МАНИПУЛИРОВАТЬ ДАННЫМИ , ОПРЕДЕЛЯЯ КАКАЯ СТРОКА КОГДА БУДЕТ ВВОДИТЬСЯ.

ПРЕДЛАГАЕМАЯ ПРОГРАММА ДАСТ ВАМ ПРЕДСТАВЛЕНИЕ О ТОМ, КАК ВСЕ ЭТО ПРОИСХОДИТ. ВВЕДИТЕ ПРОГРАММУ И НАЖМИТЕ RUN И ВК. ВЫ МОЖЕТЕ ОСТАНОВИТЬ ПРОГРАММУ - УС+С.

```
  10 CLS:SCREEN0,0,0,128,16,208,6,134,22,54,0,197,34,192,2,152,82,173:
     COLOR 7,0,8:SCREEN 2,15:REM ВЫСТАВИЛИ ЦВЕТА ЭКРАНА
  40 REM МЕНЮ
  50 RESTORE 301
  55 FOR N=1 TO 4:READ A$:REM ПРОЧИТАЛИ ИЗ СТРОКИ DATA НАЗВАНИЯ МЕЛОДИЙ
  60 PRINT AT1,15-2*N N;"=";A$:REM РАСПЕЧАТАЛИ ИХ НА ЭКРАНЕ
  70 NEXT N
  90 INPUT P:REM ЗАПРОС МЕЛОДИИ, КОТОРУЮ ВЫ ХОТИТЕ ПРОСЛУШАТЬ
 100 IF P<1 OR P>4 THEN CLS:GOTO 40
 101 REM В ПРОГРАММЕ ВСЕГО 4 МЕЛОДИИ И, ЕСЛИ ВВЕДЕН НОМЕР, КОТОРОГО НЕТ,
         ВОЗВРАТ В МЕНЮ
 110 IF P<>INT(P) THEN CLS:GOTO 40
 111 REM ЕСЛИ ВВЕДЕНО НЕ ЦЕЛОЕ ЧИСЛО, ТО ТОЖЕ ВОЗВРАТ В МЕНЮ
 130 CLS
 140 INPUT"СКОЛЬКО РАЗ ПОВТОРИТЬ";V
 150 V=INT(V)
 160 IF V<1 THEN 40
 170 ON P GOTO 175,176,177,178
 175 RESTORE 301:GOTO 190:REM ОЧЕРЕДНОЙ READ УСТАНАВЛИВАЕТСЯ НА НАЗВАНИЕ
 176 RESTORE 302:GOTO 190:REM ВЫБРАННОЙ МЕЛОДИИ
 177 RESTORE 303:GOTO 190
 178 RESTORE 304
 190 READ A$
 191 REM ТЕПЕРЬ РАССЧИТАЕМ ВЕЛИЧИНУ 'K', НЕОБХОДИМУЮ ДЛЯ ЦЕНТРИРОВАНИЯ
         НАЗВАНИЯ МЕЛОДИИ НА ЭКРАНЕ.
 210 K=INT((42-LEN(A$))/2)
 220 PRINT AT K-2,11"- "A$" -"
 260 ON P GOTO 265,266,267,268
 265 RESTORE 401:GOTO 275
 266 RESTORE 402:GOTO 275
 267 RESTORE 403:GOTO 275
 268 RESTORE 404
 275 READ Q,R,S:REM ВВЕЛИ ПАРАМЕТРЫ МЕЛОДИИ: Q-КОЛИЧЕСТВО НОТ, R-РИТМ,
    S-ПАУЗА МЕЖДУ ПОВТОРАМИ. ДАЛЕЕ ИДЕТ САМА МЕЛОДИЯ.
 276 FOR M=1 TO V
 277 ON P GOTO 278,279,280,281
```

```
278 RESTORE 500:GOTO 282
279 RESTORE 600:GOTO 282
280 RESTORE 700:GOTO 282
281 RESTORE 800
282 FOR N=1 TO Q
283 READ A,B:BEEP A/R,B:NEXT N
286 PAUSE S:NEXT M:RUN
300 REM ------------------------
301 DATA "МЕЛОДИЯ 1"
302 DATA "МЕЛОДИЯ 2"
303 DATA "МЕЛОДИЯ 3"
304 DATA "МЕЛОДИЯ 4"
400 REM ------------------------
401 DATA 23,6,16
402 DATA 72,4,75
403 DATA 38,3,1
404 DATA 33,3,33
490 REM ----- МЕЛОДИЯ 1 --------
500 DATA 2,0,2,0,2,4,2,7
501 DATA 6,12,10,9,2,9,2,5
502 DATA 2,7,2,9,16,7,2,0
503 DATA 2,0,2,4,1,7,1,7
504 DATA 6,7,10,2,2,4,2,5
505 DATA 2,4,2,2,14,0
590 REM ----- МЕЛОДИЯ 2 --------
600 DATA 2,2,2,7,3,7,1,7
601 DATA 1,7,3,11,2,7,2,9
602 DATA 3,9,1,9,1,9,3,12
603 DATA 2,9,2,11,2,9,2,7
604 DATA 2,14,2,12,2,11,2,11
605 DATA 3,9,1,9,4,9,1,2
606 DATA 1,2,2,7,2,7,2,7
607 DATA 1,7,3,11,2,7,2,9
608 DATA 3,9,1,9,1,9,3,12
609 DATA 1,9,1,9,1,11,3,14
610 DATA 2,12,1,11,3,14,2,12
611 DATA 3,11,1,7,2,9,4,7
612 DATA 2,2,3,7,1,7,2,7
613 DATA 1,7,3,11,2,7,3,9
614 DATA 1,9,2,9,1,9,3,12
615 DATA 1,9,1,9,1,11,3,14
616 DATA 2,12,1,11,3,14,2,12
617 DATA 3,11,1,7,2,9,4,7
690 REM ----- МЕЛОДИЯ 3 -----
700 DATA 1,0,1,0,2,5,2,5
701 DATA 2,9,1,0,1,9,2,10
702 DATA 2,10,2,14,1,10,1,10
703 DATA 2,9,2,9,2,12,1,9
704 DATA 1,5,2,7,2,7,2,12
705 DATA 2,0,2,5,1,5,1,5
706 DATA 2,9,1,12,1,9,2,10
707 DATA 2,10,2,14,2,10,2,9
708 DATA 1,9,1,9,2,0,2,9
709 DATA 2,7,2,5
790 REM ----- МЕЛОДИЯ 4 ------
800 DATA 2,7,2,11,2,11,6,11
801 DATA 2,7,2,9,1,11,1,9
802 DATA 6,7,2,7,2,11,2,14
803 DATA 6,14,2,9,2,11,8,14
804 DATA 2,14,2,16,2,14,6,11
805 DATA 1,9,1,7,2,14,2,11
806 DATA 6,2,1,2,1,2,2,4
807 DATA 2,7,6,9,2,2,2,4,6,7
```

ИТОГИ КОНКУРСА НА СОЗДАНИЕ БЫТОВЫХ ПЭВМ

(ПО МАТЕРИАЛАМ "ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЕ ПРИМЕНЕНИЕ",1990,8)

В ПРОШЛОМ ГОДУ ГОСУДАРСТВЕННЫЙ КОМИТЕТ ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ И ИНФОРМАТИКЕ, ЦК ВЛКСМ , ЦЕНТРАЛЬНЫЙ СОВЕТ ВОИР И ЦК ПРОФСОЮЗА РАБОЧИХ РАДИОЭЛЕКТРОННОЙ ПРОМЫШЛЕННОСТИ ПРОВЕЛИ ВСЕСОЮЗНЫЙ КОНКУРС НА СОЗДАНИЕ ПЕРСПЕКТИВНЫХ МОДЕЛЕЙ БЫТОВЫХ ПЕРСОНАЛЬНЫХ ЭВМ. ИТАК , СРЕДИ ПРОЧИХ ДОСТОИНСТВ МОДЕЛЬ ДОЛЖНА БЫТЬ ВЫСОКОНАДЕЖНОЙ, ДЕШЕВОЙ В ИЗГОТОВЛЕНИИ, ИМЕТЬ ОТЕЧЕСТВЕННУЮ ЭЛЕМЕНТНУЮ БАЗУ И ТОЛЬКО СЕРИЙНУЮ.

НА КОНКУРС БЫЛО ПРЕДОСТАВЛЕНО БОЛЕЕ ДВАДЦАТИ ЗАКОНЧЕННЫХ РАЗРАБОТОК, ВЫПОЛНЕНЫХ КАК ПРОФЕССИОНАЛЬНЫМИ КОНСТРУКТОРАМИ, ТАК И ЭНТУЗИАСТАМИ-УМЕЛЬЦАМИ.НЕДАВНО БЫЛИ ПОДВЕДЕНЫ ИТОГИ. НА ПЕРВОМ МЕСТЕ ОКАЗАЛАСЬ 16-РАЗРЯДНАЯ "ПЕРСОНАЛКА" МК-88 , РАЗРАБОТАННАЯ ИНСТИТУТОМ КИБЕРНЕТИКИ АН УССР СОВМЕСТНО С МИНСКИМ ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ. НА ВТОРОМ - СРАЗУ ДВЕ МАШИНЫ: 8-РАЗРЯДНАЯ "ВЕКТОР-06Ц" КИШИНЕВСКОГО ПО "СЧЕТМАШ" И МС-1502, РАЗРАБОТАННАЯ В КИЕВЕ НА ПО "МИКРОПРОЦЕССОР" (ТОЖЕ 16-РАЗРЯДНАЯ). ТРЕТЬЕЙ ПРЕМИЕЙ НАГРАЖДЕНЫ ПЭВМ "ЭЛЬФ" МОСКОВСКОГО НИИСЧЕТМАША И "ГУРУ" ЧЕЛЯБИНСКОГО ПО "ЭЛЕКТРОНМАШ".

У КАЖДОЙ ПЕРСОНАЛКИ - СВОИ ОТЛИЧИТЕЛЬНЫЕ ЧЕРТЫ И ХАРАКТЕРИСТИКИ, НО КАКИЕ КОНКРЕТНО? И ЕЩЕ. КТО, ГДЕ И СКОЛЬКО БУДЕТ ИХ ПРОИЗВОДИТЬ? КОГДА И ПО КАКИМ ЦЕНАМ ПРОДАВАТЬ? ВОТ ЧТО РАССКАЗАЛ ПО "ВЕКТОРУ" ОДИН ИЗ РАЗРАБОТЧИКОВ ДОНАТ ТЕМИРАЗОВ, ПЕРЕШЕДШИЙ НА РАБОТУ В НИИ СЧЕТМАШ:

- ГЛАВНОЕ ОТЛИЧИЕ "ВЕКТОРА-06Ц" ОТ , СКАЖЕМ , ХОРОШО ИЗВЕСТНЫХ КОМПЬЮТЕРОВ "МИКРОША", "ЛЬВОВ ПК-01", БК-0010 И ДРУГИХ, КОТОРЫЕ ИНОГДА ПОЯВЛЯЮТСЯ В ПРОДАЖЕ , - ОТОБРАЖЕНИЕ МНОГОЦВЕТНЫХ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ. ЧИСЛО ЦВЕТОВ В ПАЛИТРЕ УВЕЛИЧЕНО В НЕСКОЛЬКО РАЗ, А ЧИСЛО ОДНОВРЕМЕННО ОТОБРАЖАЕМЫХ ЦВЕТОВ - В ЧЕТЫРЕ РАЗА. РАЗРЕШАЮЩАЯ СПОСОБНОСТЬ ЭКРАНА В "ВЕКТОРЕ" ПОВЫШЕНА В ДВА РАЗА. БЫСТРОДЕЙСТВИЕ КОМПЬЮТЕРА И ОБЬЕМ ПАМЯТИ УВЕЛИЧЕНЫ В ПОЛТОРА-ДВА РАЗА, А САМ КОМПЬЮТЕР СНАБЖЕН ТРЕМЯ МУЗЫКАЛЬНЫМИ КАНАЛАМИ И СПЕЦИАЛЬНЫМ КАНАЛОМ ДЛЯ ШУМОВЫХ ЭФФЕКТОВ.К "ВЕКТОРУ" МОЖНО ПОДКЛЮЧАТЬ ПЕЧАТАЮЩЕЕ УСТРОЙСТВО, ЭЛЕКТРОННЫЙ ДИСК ЕМКОСТЬЮ 256 КБАЙТ, НАКОПИТЕЛЬ НА ГИБКОМ МАГНИТНОМ ДИСКЕ. В КИШИНЕВЕ, МИНСКЕ, АСТРАХАНИ, КИРОВЕ И ВОЛЖСКОМ УЖЕ ОСВОЕНО СЕРИЙНОЕ ИЗГОТОВЛЕНИЕ "ВЕКТОРА-06Ц". ПЕРВАЯ ПАРТИЯ ПОСТУПИЛА В ПРОДАЖУ В КИШИНЕВЕ ЕЩЕ В ПРОШЛОМ ГОДУ. А В ЭТОМ ГОДУ В ТОРГОВУЮ СЕТЬ ПОСТУПИТ ПРИМЕРНО 20 ТЫСЯЧ БЫТОВЫХ ПЕРСОНАЛОК "ВЕКТОР-06Ц". ПРОДАЖНАЯ ЦЕНА КОМПЬЮТЕРА ЗАВИСИТ ОТ ЗАВОДА-ИЗГОТОВИТЕЛЯ И НАХОДИТСЯ В ПРЕДЕЛАХ 750 - 850 РУБЛЕЙ. ТЕХНИЧЕСКИЙ УРОВЕНЬ СООТВЕТСТВУЕТ МИРОВОМУ СТАНДАРТУ. НЕДОСТАТКОМ МОЖНО СЧИТАТЬ ИСПОЛЬЗОВАНИЕ ОТНОСИТЕЛЬНО ДЕШЕВОГО, НО МОРАЛЬНО УСТАРЕВШЕГО МИКРОПРОЦЕССОРА КР580ВМ80А. НАЧАТА РАЗРАБОТКА

НОВОЙ МОДЕЛИ КОМПЬЮТЕРА "ВЕКТОР-1608Ц". ОН БУДЕТ ИМЕТЬ ДВА ПРОЦЕССОРА - 8 И 16-РАЗРЯДНЫЕ, ЗНАЧИТЕЛЬНО БОЛЬШИЙ ОБЬЕМ ПАМЯТИ, НАКОПИТЕЛИ НА ГИБКИХ МАГНИТНЫХ ДИСКАХ, МНОГОЦВЕТНЫЙ (БОЛЕЕ 20 ТЫС. ЦВЕТОВ) ГРАФИЧЕСКИЙ АДАПТЕР И РАБОТАТЬ В СРЕДЕ СТАНДАРТНЫХ ДЛЯ ПРОФЕССИОНАЛЬНЫХ ПЭВМ ОПЕРАЦИОННЫХ СИСТЕМ. КОМПЬЮТЕР СОВМЕСТИМ С ЕС-1840, ЕС-1841 И IBM/PC-XT И ПРЕДНАЗНАЧЕН ДЛЯ ПОЛЬЗОВАТЕЛЕЙ, ЗАНИМАЮЩИХСЯ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТЬЮ ПО РАЗРАБОТКЕ ПРИКЛАДНЫХ ПРОГРАММ В ДОМАШНИХ УСЛОВИЯХ. СЕРИЙНОЕ ПРОИЗВОДСТВО НОВОЙ МАШИНЫ ПЛАНИРУЕТСЯ НАЧАТЬ С 1992 ГОДА.

ЕСЛИ ВЫ ХОТИТЕ ПРИОБРЕСТИ ПРОГРАММЫ ДЛЯ БЫТОВЫХ КОМПЬЮТЕРОВ "ВЕКТОР- 06Ц", "БК-0010" , "СИНКЛЕР", ВЫ МОЖЕТЕ СДЕЛАТЬ ЗАКАЗ ПО АДРЕСУ: Г. КИРОВ,
УЛ. КАРЛА МАРКСА, 126 , ЦЕНТР "БАЙТ" , ПРИЕМНЫЙ ПУНКТ "УНИСОН". А ИНОГО-
РОДНИЕ МОГУТ ЗАКАЗАТЬ ПРОГРАММЫ И КАТАЛОГИ ПО АДРЕСУ: Г. КИРОВ А/Я 1248
ЗУБКОВУ АЛЕКСАНДРУ НИКОЛАЕВИЧУ. В "УНИСОНЕ" ВЫ МОЖЕТЕ ТАКЖЕ ПРИОБРЕСТИ
СХЕМЫ И ЧЕРТЕЖИ ПЕЧАТНЫХ ПЛАТ КВАЗИДИСКА НА 256 КБТ И КОНТРОЛЛЕРА ДИСКО-
ВОДА , ПРОГРАММЫ , РАБОТАЮЩИЕ В ФОРМАТЕ МДОС И ДОКУМЕНТАЦИЮ НА МДОС.
ИМЕЕТСЯ ТАКЖЕ РАСПЕЧАТКА ИСХОДНОГО ТЕКСТА НАЧАЛЬНОГО ЗАГРУЗЧИКА , НАПИ-
САННОГО НА АССЕМБЛЕРЕ.

ОСОБЕННОСТИ НАЧАЛЬНОГО ЗАГРУЗЧИКА

НАЧАЛЬНЫЙ ЗАГРУЗЧИК ЗАШИТ В ПЗУ И ОБЕСПЕЧИВАЕТ ЗАГРУЗКУ ПРОГРАММ В ФОРМАТЕ ROM. ПРИ НАЖАТИИ НА ВВОД+БЛК ПРОИСХОДИТ ПОДКЛЮЧЕНИЕ ПЗУ ПО АДРЕСАМ 0Н-1FFH. ПРИЧЕМ ПРОЦЕССОР СЧИТЫВАЕТ КОМАНДЫ С ПЗУ,А ДАННЫЕ ЗАПИСЫВАЕТ В ОЗУ. ПОСЛЕ НАЖАТИЯ НА БЛК+СБР ПЗУ ОТКЛЮЧАЕТСЯ И ЗАПУСКАЕТСЯ ПРОГРАММА С 0-ГО АДРЕСА. ЗАГРУЗЧИК РИСУЕТ КАРТУ ЗАГРУЗКИ ВО ВТОРОЙ ЭКРАННОЙ ПЛОСКОСТИ ПО АДРЕСАМ: 0С000Н-0DFFFH И ОТМЕЧАЕТ НА НЕЙ ЗАГРУЖЕННЫЕ БЛОКИ. КАЖДЫЙ БЛОК (256 БАЙТ) СОСТОИТ ИЗ ВОСЬМИ БЛОЧКОВ ПО 32 БАЙТА.ИМЕННО ЭТИ БЛОЧКИ И ОТМЕЧАЮТСЯ НА КАРТЕ ЗАГРУЗКИ ПОЛОСКАМИ. КОГДА ЗАГРУЗЧИК СЧИТЫВАЕТ ЗАГОЛОВОК НОВОГО БЛОКА ОН ПРОВЕРЯЕТ, ВСЕ ЛИ ВОСЕМЬ ПОЛОСОК ПРЕДЫДУЩЕГО БЛОКА ЕСТЬ НА ЭКРАНЕ, А ТАКЖЕ ПОРЯДКОВЫЙ НОМЕР НОВОГО БЛОКА. ЕСЛИ ЧТО-ТО НЕ В ПОРЯДКЕ (БЫЛ СБОЙ), ПРОИСХОДИТ ПЕРЕЗАПУСК ЗАГРУЗЧИКА.

ПРИ ПЕРЕЗАПУСКЕ ЗАГРУЗЧИК СТИРАЕТ ВСЕ ОЗУ, В ТОМ ЧИСЛЕ И ЭКРАННОЕ, И ЗАНОВО РИСУЕТ КАРТУ ЗАГРУЗКИ.ЕСЛИ ПРИ НАЖАТИИ НА ВВОД+БЛК УДЕРЖИВАТЬ КЛАВИШУ УС, ТО ЗАГРУЗЧИК НЕ СТАНЕТ ОБНУЛЯТЬ ОЗУ И СОХРАНИТ ЕГО,НО КАРТУ ЗАГРУЗКИ ВСЕ ЖЕ НАРИСУЕТ. ЭТО ОЧЕНЬ УДОБНО ДЛЯ МАНИПУЛИРОВАНИЯ РАЗЛИЧНЫМИ ПРОГРАММАМИ.ВЫ МОЖЕТЕ ЗАГРУЗИТЬ ОДНОВРЕМЕННО В ПАМЯТЬ НЕСКОЛЬКО ПРОГРАММ В КОДАХ,ЛИШЬ БЫ ОНИ НЕ ЗАТИРАЛИ ДРУГ ДРУГА. ВЕСЬ ПРОЦЕСС СВОДИТСЯ К СЛЕДУЮЩИМ ОПЕРАЦИЯМ: ГРУЗИТЕ ОБЫЧНЫМ СПОСОБОМ ПРОГРАММУ, НАЖИМАЕТЕ ВВОД+БЛК+УС, ОТПУСКАЕТЕ ВВОД+БЛК,УДЕРЖИВАЯ УС, ОТПУСКАЕТЕ УС, ГРУЗИТЕ ДРУГУЮ ПРОГРАММУ И.Т.Д. КРОМЕ ТОГО ПОЛЕЗНО УДЕРЖИВАТЬ УС ВО ВРЕМЯ ЗАГРУЗКИ, ЕСЛИ ПРИ ЗАГРУЗКЕ ВОЗМОЖНЫ СБОИ (ПОМЕХИ В СЕТИ, ДЕФЕКТ ПЛЕНКИ И.Т.Д.).ПРИ ОШИБКЕ ВСЕ РАНЕЕ ЗАГРУЖЕННЫЕ ПРОГРАММЫ ОСТАНУТСЯ В ПАМЯТИ И ВАМ НУЖНО БУДЕТ ПЕРЕЗАГРУЗИТЬ ТОЛЬКО ПОСЛЕДНЮЮ.ЕСЛИ НА КАРТЕ ЗАГРУЗКИ НА МЕСТЕ ПОСЛЕДНЕГО БЛОКА УЖЕ ЕСТЬ МЕТКИ, ТО В ОЗУ ЗАГРУЗИТСЯ ТОЛЬКО ПЕРВЫЕ 32 БАЙТА, ПОСЛЕ ЧЕГО НАЧНЕТ МИГАТЬ ИНДИКАТОР РУС/ЛАТ , СООБЩАЯ ОБ ОКОНЧАНИИ ЗАГРУЗКИ. ПОЭТОМУ, ЕСЛИ ВЫ НЕ ХОТИТЕ ТЕРЯТЬ ИНФОРМАЦИЮ,ТО УЧАСТОК КАРТЫ НА МЕСТЕ ПОСЛЕДНЕГО БЛОКА ПРОГРАММЫ ДОЛЖЕН БЫТЬ ЧИСТЫМ. КСТАТИ, ИМЕННО ТАКИМ ОБРАЗОМ МОЖНО ПОСМОТРЕТЬ УЖЕ ЗАГРУЖЕННЫЙ 0-ОЙ БЛОК, ИСКЛЮЧАЯ ПЕРВЫЕ 32 БАЙТА.

ЕЩЕ ОДНОЙ ОСОБЕННОСТЬЮ НАЧАЛЬНОГО ЗАГРУЗЧИКА, ОТЛИЧАЮЩЕЙ ЕГО ОТ COPY, ЯВЛЯЕТСЯ ТО,ЧТО ОН ГРУЗИТ ИНФОРМАЦИЮ ТУДА,ГДЕ ОНА ДОЛЖНА РЕАЛЬНО РАСПОЛАГАТЬСЯ. ТОГДА КАК COPY ГРУЗИТ ВСЕ ПРОГРАММЫ В БУФЕР С 800Н-ГО АДРЕСА.ЕСЛИ ВАША ПРОГРАММА ДОЛЖНА ГРУЗИТСЯ В ЭКРАННУЮ ОБЛАСТЬ, ТО ИМЕННО ТУДА ОНА И ЗАГРУЗИТСЯ. В СВЯЗИ С ЭТИМ ВОЗМОЖНА ЗАЩИТА ПРОГРАММ ОТ ПЕРЕЗАПИСИ ЧЕРЕЗ COPY, КОТОРАЯ ПРИМЕНЯЕТСЯ В ПРОГРАММЕ В ПРОГРАММЕ SPY'S DEMISE. СДЕЛАНА ОНА ТАК. ПЕРВЫЙ БЛОК ГРУЗИТСЯ В ЭКРАННУЮ ОБЛАСТЬ НЕПОСРЕДСТВЕННО НА КАРТУ НАЧАЛЬНОГО ЗАГРУЗЧИКА. ПРИЧЕМ ЭТОТ БЛОК СОДЕРЖИТ ТОЛЬКО 7 УЧАСТКОВ ПО 32 БАЙТА, 8-ГО УЧАСТКА НЕТ А МЕТКА 8-ГО УЧАСТКА ВОЗНИКАЕТ ПРИ ЗАГРУЗКЕ 7-ГО УЧАСТКА,КОТОРЫЙ СОДЕРЖИТ ЕЕ В СЕБЕ (МЕТКА КОДИРУЕТСЯ БАЙТОМ 7ЕН).ТО ЕСТЬ ПРОГРАММА САМА ПОДПРАВЛЯЕТ СВОЮ ЗАГРУЗКУ (КОНЕЧНО, ЕСЛИ ОНА ГРУЗИЛАСЬ В НАЧАЛЬНЫЙ ЗАГРУЗЧИК). ЕСЛИ ЖЕ ЭТА ПРОГРАММА ГРУЗИТСЯ В COPY,ТО ОНА УЖЕ НЕ КОРРЕКТИРУЕТ ЭКРАН, ТАК КАК ГРУЗИТСЯ С 800Н-ГО АДРЕСА. ПОЭТОМУ ОШИБКА ГЕНЕРИРУЕТСЯ СРАЗУ И ПРОИСХОДИТ ПЕРЕЗАПУСК COPY (В ОТЛИЧИИ ОТ ЗАЩИТЫ ПО ИМЕНИ, КОГДА ОШИБКА ГЕНЕРИРУЕТСЯ ТОЛЬКО ПОСЛЕ ЗАГРУЗКИ ВСЕЙ ПРОГРАММЫ).

ТЕПЕРЬ РАССМОТРИМ СТЕК НАЧАЛЬНОГО ЗАГРУЗЧИКА И БУФЕР ЗАГРУЗКИ. БУФЕР - ЭТО БОЛЬШОЙ ПРЯМОУГОЛЬНИК В ПРАВОМ ВЕРХНЕМ УГЛУ. ЕГО НАЧАЛЬНЫЙ АДРЕС - 0DED0H, ЗАТЕМ ИДЕТ НАРАЩИВАНИЕ ВВЕРХ. В БУФЕР ГРУЗИТСЯ БЛОЧКИ ПО 35 БАЙТ, ИЗ КОТОРЫХ 0-ОЙ БАЙТ ОПРЕДЕЛЯЕТ, ЧТО ЭТА ИНФОРМАЦИЯ - ЗАГОЛОВОК (ЕСЛИ ОН РАВЕН 0) ИЛИ МЛАДШИЙ БАЙТ АДРЕСА ЗАГРУЗКИ (ЕСЛИ ОН НЕ РАВЕН 0), ПРИ ЭТОМ МЛАДШИЙ БАЙТ АДРЕСА ОПРЕДЕЛЯЕТСЯ КАК ЗНАЧЕНИЕ БАЙТА,УМНОЖЕННОЕ НА 32, 1-ЫЙ БАЙТ - КОНТРОЛЬНАЯ СУММА ТЕКУЩЕГО БЛОКА, СЛЕДУЮЩИЕ 32 БАЙТА - ИНФОРМАЦИОННЫЕ, 34 - Й БАЙТ - КОНТРОЛЬНАЯ СУММА, ДЕЛЕННАЯ ПОПОЛАМ. У ЗАГОЛОВКА НУЛЕВОГО БЛОКА ПРОГРАММЫ 31 - Й БАЙТ ОПРЕДЕЛЯЕТ СТАРШИЙ БАЙТ АДРЕСА ЗАГРУЗКИ (С КАКОГО БЛОКА БУДЕТ ГРУЗИТЬСЯ ПРОГРАММА), 32-ОЙ И 33-Й СОДЕРЖАТ ЧИСЛО ЗАГРУЖАЕМЫХ БЛОКОВ (ИНФОР-

МАЦИЯ ДУБЛИРУЕТСЯ ДЛЯ ПОМЕХОЗАЩИЩЕННОСТИ). ПОСЛЕ ПРОВЕРКИ КОНТРОЛЬНОЙ СУММЫ ИНФОРМАЦИОННЫЕ БАЙТЫ ПЕРЕПИСЫВАЮТСЯ В ОЗУ.

ВЕРХУШКА СТЕКА (МАЛЕНЬКИЙ КВАДРАТИК РЯДОМ) РАСПОЛАГАЕТСЯ ПО АДРЕСУ 0DCF0H. В НЕМ ХРАНЯТСЯ ПРОМЕЖУТОЧНЫЕ ДАННЫЕ И АДРЕСА ВОЗВРАТА ИЗ ПОДПРОГРАММ.ИЗ ВСЕХ ПРОЧИХ УКАЖЕМ ОСНОВНЫЕ: 0DCEEH-0DCEFH - АДРЕС ВОЗВРАТА ИЗ ПОДПРОГРАММЫ LOADF (008DH, LOADF - ПОДПРОГРАММА ЗАГРУЗКИ ФАЙЛА), 0DCEDH - ЧИСЛО ОСТАВШИХСЯ ДЛЯ ЗАГРУЗКИ БЛОКОВ, 0DCECH - ТЕКУЩАЯ КОНТРОЛЬНАЯ СУММА, 0DCEBH - СТАРШИЙ БАЙТ АДРЕСА ЗАГРУЗКИ, 0DCEAH - МЛАДШИЙ БАЙТ АДРЕСА ЗАГРУЗКИ. ЗАГРУЗЧИК РАБОТАЕТ СЛЕДУЮЩИМ ОБРАЗОМ: УСТАНАВЛИВАЕТ ЦВЕТА, НАЧАЛЬНЫЕ ДАННЫЕ, СТИРАЕТ ОЗУ,ПЕРЕДАЕТ УПРАВЛЕНИЕ ПОДПРОГРАММЕ LOADF, LOADF ОПРЕДЕЛЯЕТ КОНСТАНТУ ЧТЕНИЯ, ЗАПИСЫВАЕТ ЕЕ В ЯЧЕЙКУ ПО АДРЕСУ 0DEF6H, ЛОВИТ 0-ОЙ БЛОК, ЗАТЕМ НАЧИНАЕТ ГРУЗИТЬ БЛОКИ.ПРИ ЭТОМ ЛОЖИТ DE - ЧИСЛО БЛОКОВ И КОНТРОЛЬНАЯ СУММА И BC - АДРЕС ЗАГРУЗКИ В СТЕК, ЗАГРУЖАЕТ 35 БАЙТ, ПЕРЕЗАПИСЫВАЕТ ИНФОРМАЦИОННЫЕ БАЙТЫ В ОЗУ, ВОССТАНАВЛИВАЕТ DE И BC И.Т.Д. ДО ОКОНЧАНИЯ ЗАГРУЗКИ. ЗАТЕМ УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ОСНОВНОЙ ПРОГРАММЕ,НАЧИНАЕТ МИГАТЬ РУС/ЛАТ, В СТЕКЕ ОТОБРАЖАЕТСЯ СОСТОЯНИЕ СЧЕТЧИКА ВРЕМЕНИ И.Т.Д.

ИЗ ВЫШЕПЕРЕЧИСЛЕННЫХ ОСОБЕННОСТЕЙ НАЧАЛЬНОГО ЗАГРУЗЧИКА МОЖНО СДЕЛАТЬ ВЫВОД О ВОЗМОЖНОСТИ АВТОЗАПУСКА ПРОГРАММ В КОДАХ. ДЕЙСТВИТЕЛЬНО, ЗАГРУЖАЯ ИНФОРМАЦИЮ ПРЯМО НА ЭКРАН, МЫ МОЖЕМ ЗАМЕНИТЬ АДРЕС ВОЗВРАТА ИЗ LOADF НА СВОЙ АДРЕС ЗАПУСКА,ЧИСЛО ОСТАВШИХСЯ БЛОКОВ УСТАНОВИТЬ В 1 И ВОССТАНОВИТЬ МЕТКИ БЛОКА НА КАРТЕ ЗАГРУЗКИ.ЗАГРУЗЧИК ОБНАРУЖИТ, ЧТО ВСЕ БЛОКИ ЗАГРУЖЕНЫ, МЕТКИ НА ЭКРАНЕ ЕСТЬ И ПО КОМАНДЕ RET ПЕРЕДАСТ УПРАВЛЕНИЕ НА НАШУ ПРОГРАММУ. ПРИ ЭТОМ ПРОГРАММА АВТОЗАПУСТИТСЯ,НО ПЗУ НЕ ОТКЛЮЧИТСЯ, А ОСТАНЕТСЯ ПО АДРЕСАМ 0H - 1FFH. ПРИ ЖЕЛАНИИ МОЖНО ЭТУ ОБЛАСТЬ ПЕРЕПИСАТЬ ПОВЫШЕ И ЧЕРЕЗ УЖЕ ОПИСАННЫЙ СПОСОБ (С ИСПОЛЬЗОВАНИЕМ УС) ПРОСМОТРЕТЬ В МОНИТОРЕ (ПРАКТИЧЕСКИ МЫ СЧИТАЕМ СОДЕРЖИМОЕ ПЗУ ЧИСТО ПРОГРАММНЫМ ПУТЕМ).

ПРИ ЗАГРУЗКЕ ПРОГРАММЫ УКАЗАННЫМ СПОСОБОМ ИМЕЮТСЯ НЕКОТОРЫЕ ОГРАНИЧЕНИЯ: ВО-ПЕРВЫХ, НЕ ДОСТУПНЫ АДРЕСА 0H-1FFH И ,СЛЕДОВАТЕЛЬНО, НЕЛЬЗЯ ПЕРЕЗАПУСТИТЬ ПРОГРАММУ ЧЕРЕЗ 0-ОЙ АДРЕС (ЗАРАБОТАЕТ НАЧАЛЬНЫЙ ЗАГРУЗЧИК), ВО-ВТОРЫХ, ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ НЕ СМОЖЕТ РАБОТАТЬ,ТАК КАК ЕЙ ТРЕБУЕТСЯ ТОЧКА ВХОДА ПО АДРЕСУ 38H, А ТАМ РАСПОЛАГАЕТСЯ ПОДПРОГРАММА ОБРАБОТКИ КОНСТАНТЫ ЧТЕНИЯ. ОДНАКО, ЕСЛИ ВЫ КАК СЛЕДУЕТ ПОДУМАЕТЕ, ТО СМОЖЕТЕ ОБОЙТИ И ЭТИ ПРЕПЯТСТВИЯ.

НАДЕЕМСЯ, ЧТО ПРИВЕДЕННАЯ ИНФОРМАЦИЯ О НАЧАЛЬНОМ ЗАГРУЗЧИКЕ ОКАЖЕТСЯ ИНТЕРЕСНОЙ И ПОЛЕЗНОЙ ДЛЯ ВАС. ЖДЕМ ВАШИХ ОТКЛИКОВ НА СТАТЬЮ.

ОТМЫЧКА ДЛЯ ВЗЛОМЩИКА ИЛИ ПОСОБИЕ ДЛЯ НАЧИНАЮЩЕГО ХЭККЕРА

НАСТОЯЩАЯ СТАТЬЯ ПУБЛИКУЕТСЯ В ПОМОЩЬ ТОЙ ЧАСТИ ЛЮБИТЕЛЕЙ 'ВЕКТОР-06Ц', КОТОРЫЕ НЕ ХОТЯТ ОСТАНАВЛИВАТЬСЯ НА ПОЛЬЗОВАНИИ ПРОГРАММАМИ И ПРОГРАММИРОВАНИИ НА БЕЙСИКЕ, А ЖЕЛАЮТ ПОБЛИЖЕ ОЗНАКОМИТСЯ С АССЕМБЛЕРОМ.

ДЕВИЗ ХЭККЕРОВ:"ВЗЛАМЫВАТЬ ЛЕГЧЕ, ЧЕМ ДЕЛАТЬ!". ОДНАКО, ЕСЛИ У ВАС МАЛО ТЕРПЕНИЯ И НАСТОЙЧИВОСТИ, ЛУЧШЕ И НЕ БЕРИТЕСЬ ЗА ЭТО ДЕЛО.

ОГРОМНУЮ ПОМОЩЬ ВЗЛОМЩИКУ ЧУЖИХ ПРОГРАММ ОКАЗЫВАЮТ COPY, МОНИТОР И НАЧАЛЬНЫЙ ЗАГРУЗЧИК.НАЧАЛЬНЫЙ ЗАГРУЗЧИК ЗАМЕЧАТЕЛЕН ТЕМ, ЧТО ДАЕТ ВОЗМОЖНОСТЬ ЗАГРУЖАТЬ В ПАМЯТЬ СРАЗУ НЕСКОЛЬКО ПРОГРАММ. КАК ЭТО ДЕЛАЕТСЯ РАССКАЗАНО В ПРЕДЫДУЩЕЙ СТАТЬЕ. ТО ЕСТЬ МЫ МОЖЕМ ЗАГРУЗИТЬ МОНИТОР, ЗАПУСТИТЬ ЕГО, ЧЕРЕЗ УС+ВВОД+БЛК ЗАГРУЗИТЬ ЛЮБУЮ ПРОГРАММУ В КОДАХ (КОНЕЧНО, ЕСЛИ ОНА НЕ ЗАТИРАЕТ 0-ОЙ БЛОК И САМ МОНИТОР),НАЖАТЬ БЛК+СБР И ПОСМОТРЕТЬ ЕЕ В МОНИТОРЕ. ДЛЯ ПРОФЕССИОНАЛЬНОЙ РАБОТЫ ЛУЧШЕ ИМЕТЬ ДВА ВАРИАНТА МОНИТОРА,КОТОРЫЕ ГРУЗЯТСЯ СРАЗУ В ТЕ ОБЛАСТИ ПАМЯТИ, В КОТОРЫХ РАБОТАЮТ. ДЛЯ ЭТОГО ЗАГРУЗИТЕ МОНИТОР, ВЫБЕРИТЕ ОБЛАСТЬ РАСПОЛОЖЕНИЯ <1>, КОМАНДОЙ O>94,C,0 ВЫГРУЗИТЕ ПЕРВЫЙ КУСОК, КОМАНДОЙ O>E0,20,0 - ВТОРОЙ И КОМАНДОЙ O>0,1,0 ВЫГРУЗИТЕ 0-ОЙ БЛОК,В КОТОРОМ НАХОДЯТСЯ ТОЧКИ ВХОДА В МОНИТОР. ЗАТЕМ СНОВА ЗАГРУЗИТЕ МОНИТОР, ВЫБЕРИТЕ ОБЛАСТЬ РАСПОЛОЖЕНИЯ <2>, КОМАНДОЙ O>54,2C,0 ВЫГРУЗИТЕ МОНИТОР НА ЛЕНТУ,КОМАН-

ДОЙ O>0,1,0 ВЫГРУЗИТЕ 0-ОЙ БЛОК. ТЕПЕРЬ У ВАС ЕСТЬ МОНИТОР, КОТОРЫЙ ГРУЗИТСЯ СРАЗУ ЖЕ ПО СВОИМ РАБОЧИМ АДРЕСАМ, НЕ ЗАТИРАЯ ПРОГРАММУ, КОТОРАЯ НАХОДИТСЯ В ОЗУ С АДРЕСА 100H И ВЫШЕ.В ДАЛЬНЕЙШЕМ МЫ БУДЕМ ПОДРАЗУМЕВАТЬ РАБОТУ ИМЕННО С ТАКИМ МОНИТОРОМ.

В МОНИТОРЕ ЕСТЬ ПРЕКРАСНЫЕ ДИРЕКТИВЫ D,L,F,A,S,Q И ДРУГИЕ. ДИРЕКТИВОЙ D УДОБНО ПРОСМАТРИВАТЬ ПРОГРАММЫ В ПОИСКАХ ТЕКСТОВЫХ СООБЩЕНИЙ, ТАК КАК ОНА ОТОБРАЖАЕТ СОДЕРЖИМОЕ ОЗУ В ШЕСТНАДЦАТИРИЧНЫХ КОДАХ И СИМВОЛАХ ASCII. ПРИ ПРОСМОТРЕ, СПРАВА ВИДНЫ ВСЕ СООБЩЕНИЯ, ИМЕЮЩИЕ КАКОЙ-ТО СМЫСЛ, А ТАКЖЕ МУЗЫКА, ЕСЛИ ПРОГРАММА НАПИСАНА НА ДРАЙВЕРАХ.ДИРЕКТИВА L ДИЗАССЕМБЛИРУЕТ ТЕКСТ ПРОГРАММЫ И ВЫ МОЖЕТЕ С ЕЕ ПОМОЩЬЮ ОТСЛЕДИТЬ ВСЮ ЛОГИКУ.ОДНАКО БУДЬТЕ ВНИМАТЕЛЬНЫ, ТАК КАК ДИРЕКТИВА L НЕ ОТЛИЧАЕТ КОМАНД ОТ ДАННЫХ И ДИЗАССЕМБЛИРУЕТ ВСЕ ПОДРЯД. ДИРЕКТИВА Q СЛУЖИТ ДЛЯ ПОИСКА ШАБЛОНА В ОБЛАСТИ ПАМЯТИ. С ЕЕ ПОМОЩЬЮ ВЫ МОЖЕТЕ НАЙТИ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ,КОДОВ ИЛИ ИНТЕРЕСУЮЩУЮ ВАС КОМАНДУ АССЕМБЛЕРА. НАПРИМЕР, ВЫ ХОТИТЕ УЗНАТЬ В КАКОМ МЕСТЕ ПРОГРАММЫ ВЫВОДИТСЯ ТЕКСТОВОЕ СООБЩЕНИЕ "GAME OVER". КОМАНДОЙ Q100, КОНЕЦ, ВК, "GAME OVER", ВК ВЫ ПОЛУЧАЕТЕ АДРЕС С КОТОРОГО НАЧИНАЕТСЯ ЭТО СООБЩЕНИЕ: 4500H. ДИРЕКТИВОЙ D4500 ВЫ МОЖЕТЕ УБЕДИТЬСЯ В ПРАВИЛЬНОСТИ ПОИСКА. ЕСЛИ ПРОГРАММА НАПИСАНА НА ДРАЙВЕРАХ, ТО СООБЩЕНИЕ ДОЛЖНО КОНЧАТЬСЯ НУЛЕВЫМ БАЙТОМ. ТЕПЕРЬ ВАМ НУЖНО ОПРЕДЕЛИТЬ НАЧАЛО СООБЩЕНИЯ.ДЕЛАЙТЕ ЭТО ТАКЖЕ ДИРЕКТИВОЙ D: D44F0. ИЩИТЕ НАЧАЛО, ПРИЧЕМ НЕ ЗАБЫВАЙТЕ , ЧТО КОДЫ 1FH,1BH, 0AH, 0DH И НЕКОТОРЫЕ ДРУГИЕ ЯВЛЯЮТСЯ УПРАВЛЯЮЩИМИ ПРИ ПЕЧАТИ И ТАКЖЕ МОГУТ ВХОДИТЬ В СОСТАВ СООБЩЕНИЯ. ДОПУСТИМ ВЫ НАШЛИ НАЧАЛО ПО АДРЕСУ 44F6H. ТЕПЕРЬ ДИРЕКТИВОЙ Q100, КОНЕЦ, ВК, 'LXI H,44F6H, ВК ВЫ СМОЖЕТЕ ОПРЕДЕЛИТЬ МЕСТО , В КОТОРОМ ПРОИЗВОДИТСЯ ВЫВОД СООБЩЕНИЯ "GAME OVER" НА ЭКРАН.

ДЛЯ ПОИСКА КАРТИНОК ИЛИ ЗНАКОГЕНЕРАТОРА МОЖНО ВОСПОЛЬЗОВАТЬСЯ ДИРЕКТИВОЙ M:M100,1FFF,A000 - ВЫВОД СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ С 100 ПО 1FFFH НА ЭКРАН. НЕ ВЫВОДИТЕ СЛИШКОМ БОЛЬШОЙ ОБЬЕМ ИНФОРМАЦИИ - ВЫ СОТРЕТЕ МОНИТОР. И НЕ ЗАБЫВАЙТЕ,ЧТО ЭТОТ СПОСОБ ГОДИТСЯ ТОЛЬКО ДЛЯ БАЙТОВЫХ, НЕ СЖАТЫХ КАРТИНОК, ВЫВОДЯЩИХСЯ НА ЭКРАН СНИЗУ-ВВЕРХ,СЛЕВА-НАПРАВО ИЛИ СВЕРХУ-ВНИЗ, СЛЕВА-НАПРАВО. ДЛЯ ПОИСКА РИСУНКОВ В ФОРМАТЕ PUT И GET ЭТОТ СПОСОБ ТАКЖЕ НЕ ГОДИТСЯ.

ВОТ ВЫ НАШЛИ ЗНАКОГЕНЕРАТОР И КОНЕЧНО ЗАХОТЕЛИ ИЗМЕНИТЬ ЕГО. ТУТ ВАМ ПРИГОДИТСЯ ДИРЕКТИВА S. ИЗМЕНЯЯ ПОБАЙТНО СИМВОЛ ИЛИ КАРТИНКУ И КОНТРОЛИРУЯ СЕБЯ ВЫВОДОМ ЕЕ НА ЭКРАН, ВЫ СМОЖЕТЕ ДОБИТСЯ НУЖНЫХ РЕЗУЛЬТАТОВ.

ЗАТЕМ ВАМ ЗАХОТЕЛОСЬ УВЕЛИЧИТЬ ЧИСЛО ЖИЗНЕЙ В ИГРЕ.ДЛЯ ЭТОГО ВАМ НУЖНО НАЙТИ ЯЧЕЙКУ ЖИЗНИ И ТО МЕСТО, ГДЕ ОНА ЗАГРУЖАЕТСЯ КАКИМ-ТО ЗНАЧЕНИЕМ. ВЫ ЗНАЕТЕ,ЧТО ЖИЗНЕЙ ВСЕГО 3 И МОЖЕТЕ ПОИСКАТЬ В ПАМЯТИ ЯЧЕЙКУ С ТАКИМ СОДЕРЖИМЫМ. НО ЛУЧШЕ ЭТОГО НЕ ДЕЛАТЬ, ТАК КАК МОНИТОР ВЫБРОСИТ ВАМ СЛИШКОМ МНОГО АДРЕСОВ ГОРАЗДО ПРАКТИЧНЕЕ ПОИСКАТЬ КОМАНДУ MVI A,3 ИЛИ MVI M,3. ТАКИХ КОМБИНАЦИЙ УЖЕ ГОРАЗДО МЕНЬШЕ И ИХ МОЖНО ПЕРЕБРАТЬ И ПРОАНАЛИЗИРОВАТЬ.САМЫМ РАДИКАЛЬНЫМ ЯВЛЯЕТСЯ НЕ УВЕЛИЧЕНИЕ ЖИЗНЕЙ, А ВЫРЕЗАНИЕ ТОГО МЕСТА, В КОТОРОМ ПРОИСХОДИТ УМЕНЬШЕНИЕ ЯЧЕЙКИ ЖИЗНИ, ТО ЕСТЬ "ОБРЕТЕНИЕ БЕССМЕРТИЯ". НО ЭТА ЗАДАЧА УЖЕ ПОТРУДНЕЕ.

В НЕКОТОРЫХ СЛУЧАЯХ ВОЗМОЖНО ПРИМЕНЕНИЕ ПОШАГОВОЙ ТРАНСЛЯЦИИ (ДИРЕКТИВА T) И ДРУГИХ ОТЛАДОЧНЫХ СРЕДСТВ ДЛЯ ЛУЧШЕГО ПОНИМАНИЯ СУТИ ПОДПРОГРАММ. ОДНАКО ЗАПУСК ТОЙ ИЛИ ИНОЙ ПОДПРОГРАММЫ ВСЕГДА ГРОЗИТ РАЗРУШЕНИЕМ МОНИТОРА.
ЧТО ДЕЛАТЬ, ЕСЛИ ПРОГРАММА ГРУЗИТСЯ С 0-ГО БЛОКА? ТОГДА СЛЕДУЕТ ВОСПОЛЬЗОВАТЬСЯ COPY. COPY ГРУЗИТ ПРОГРАММЫ В БУФЕР С 800H-ГО АДРЕСА. ЗАГРУЗИВ ПРОГРАММУ COPY, ВЫ ЧЕРЕЗ УС+ВВОД+БЛК ЗАГРУЖАЕТЕ МОНИТОР ПО ОБЛАСТИ <1> И МОЖЕТЕ РАБОТАТЬ С ПРОГРАММОЙ СЛЕДУЕТ ЛИШЬ УЧИТЫВАТЬ, ЧТО КО ВСЕМ АДРЕСАМ ПЕРЕХОДОВ В ПРОГРАММЕ СЛЕДУЕТ ДОБАВИТЬ 800H. ЕСЛИ ПРОГРАММА ИДЕТ С ЗАЩИТОЙ ПО ИМЕНИ - ВОСПОЛЬЗУЙТЕСЬ COPY-HELP.

ПРИ АНАЛИЗЕ ПРОГРАММ НУЖНО ВСЕГДА ИМЕТЬ В ВИДУ, ЧТО В НИХ ДЕЙСТВУЕТ ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ. ОНА ЗАНИМАЕТСЯ УСТАНОВКОЙ ЦВЕТОВ, ОПРОСОМ КЛАВИАТУРЫ, СКРОЛЛИНГОМ, МУЗЫКОЙ И.Т.Д. ТОЧКА ВХОДА В ЭТУ ПОДПРОГРАММУ РАСПОЛАГАЕТСЯ ПО АДРЕСУ 38H (RST 7). ОПРЕДЕЛИТЬ АДРЕС РАСПОЛОЖЕНИЯ ПОДПРОГРАММЫ ПРЕРЫВАНИЙ МОЖНО ПУТЕМ ПОИСКА ЦЕПОЧКИ MVI A,C3 STA 38 LXI H,ТОЧКА ВХОДА

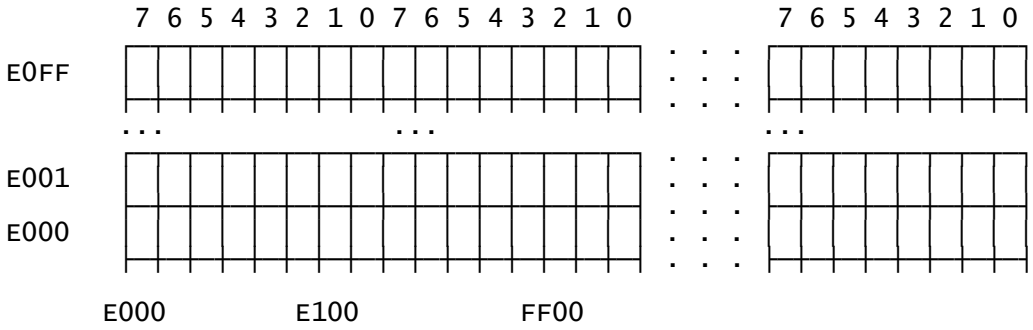
SHLD 39. РАЗОБРАВШИСЬ, НАПРИМЕР, С ПОДПРОГРАММОЙ ПРЕРЫВАНИЙ В БЕЙСИКЕ, ВЫ СМОЖЕТЕ УЗНАТЬ АДРЕСА ТАБЛИЦЫ ЦВЕТОВ, ЦВЕТА БОРДЮРА, ТАЙМЕРА, СКРОЛЛИНГА, КОНСТАНТ ЧТЕНИЯ И ЗАПИСИ НА МАГНИТОФОН И.Т.Д.

АНАЛИЗ ЧУЖИХ ПРОГРАММ - ДЕЛО ОЧЕНЬ УВЛЕКАТЕЛЬНОЕ.НАЧАВ, ВЫ ПОСТЕПЕННО ОБРЕТЕТЕ ВКУС К ЭТОМУ ДЕЛУ. КРОМЕ ТОГО - ЭТО НАСУЩНАЯ НЕОБХОДИМОСТЬ, ТАК КАК РАЗРАБОТЧИКИ, НАПРИМЕР, ТОГО ЖЕ БЕЙСИКА НЕ ПОЗАБОТИЛИСЬ СОСТАВИТЬ БОЛЕЕ ПОДРОБНУЮ ДОКУМЕНТАЦИЮ ПО СТРУКТУРЕ БЕЙСИКА, ОСТАВИВ ЕЕ СОСТАВЛЕНИЕ НА ДОЛЮ ПОЛЬЗОВАТЕЛЕЙ. ЕСЛИ ВЫ РАСКОПАЕТЕ ЧТО-ТО, ПОЛЕЗНОЕ НАПИШИТЕ В НАШ "БАЙТ" - МНОГИЕ ЧИТАТЕЛИ БУДУТ ВАМ БЛАГОДАРНЫ.

ТАБЛИЦА ЦВЕТОВ И ВИДЕО-ОЗУ.

У ПОЛЬЗОВАТЕЛЕЙ "ВЕКТОРА-06Ц", РАБОТАЮЩИХ С БЕЙСИКОМ, ЧАСТО ВОЗНИКАЮТ ВОПРОСЫ, СВЯЗАННЫЕ С ВИДЕО-ОЗУ И ТАБЛИЦЕЙ ЦВЕТОВ. В ЭТОЙ СТАТЬЕ МЫ ПОПЫТАЕМСЯ ОТВЕТИТЬ НА ЧАСТЬ ИЗ НИХ. КАК ИЗВЕСТНО, ОСНОВНЫМ ОТЛИЧИЕМ "ВЕКТОРА-06Ц" ОТ ДРУГИХ ПЕРСОНАЛЬНЫХ ЭВМ ЯВЛЯЕТСЯ ВОЗМОЖНОСТЬ РАБОТЫ С МНОГОЦВЕТНОЙ ГРАФИКОЙ. ПОЛЬЗОВАТЕЛЮ ПРЕДОСТАВЛЯЕТСЯ 256 ЦВЕТОВ И ОТТЕНКОВ (ФИЗИЧЕСКИЕ ЦВЕТА), ИЗ КОТОРЫХ ОН ОДНОВРЕМЕННО МОЖЕТ ИСПОЛЬЗОВАТЬ ТОЛЬКО 16 (МАТЕМАТИЧЕСКИЕ ЦВЕТА). "ВЕКТОР-06Ц" ИМЕЕТ ЧЕТЫРЕ ЭКРАННЫХ ПЛОСКОСТИ ПО 8 КБТ, ЛЕЖАЩИХ ДРУГ НАД ДРУГОМ. ЭТИ ПЛОСКОСТИ ЯВЛЯЮТСЯ ТАКОЙ ЖЕ ПАМЯТЬЮ, КАК И ОСТАЛЬНОЕ ОЗУ, ТОЛЬКО С ТОЙ ОСОБЕННОСТЬЮ, ЧТО ОНА СОВМЕСТНО ИСПОЛЬЗУЕТСЯ ПРОЦЕССОРОМ И ВИДЕОКОНТРОЛЛЕРОМ. ПРИЧЕМ, ЧТОБЫ НЕ ВОЗНИКАЛО ПОМЕХ В ВИДЕ "СНЕГА", ПРИОРИТЕТ ПРИ ОБРАЩЕНИИ К ВИДЕО-ОЗУ ПРЕДОСТАВЛЯЕТСЯ ВИДЕОКОНТРОЛЛЕРУ. АДРЕСА ПЛОСКОСТЕЙ ДАНЫ В ОПИСАНИИ БЕЙСИКА.

РАССМОТРИМ СТРУКТУРУ ЭТИХ ПЛОСКОСТЕЙ НА ПРИМЕРЕ 1-ОЙ ПЛОСКОСТИ.



ВЕСЬ ЭКРАН РАЗБИВАЕТСЯ НА 32 СТОЛБИКА ПО 8 ТОЧЕК (БИТОВ). КАЖДАЯ ЯЧЕЙКА ВИДЕО ОЗУ ИМЕЕТ АДРЕС, СОСТОЯЩИЙ ИЗ СТАРШЕГО И МЛАДШЕГО БАЙТОВ. СТАРШИЙ БАЙТ ОПРЕДЕЛЯЕТ ПЛОСКОСТЬ И СТОЛБИК, А МЛАДШИЙ - ГОРИЗОНТАЛЬНУЮ СТРОКУ. НАПРИМЕР, 0E003H - 1-АЯ ПЛОСКОСТЬ , НУЛЕВОЙ СТОЛБИК, 3-ЬЯ СТРОКА. ЕСЛИ КАКОЙ-ТО БИТ ЯЧЕЙКИ ВИДЕО-ОЗУ УСТАНОВЛЕН В 1, ТО НА ЭКРАНЕ ЕСТЬ ТОЧКА, В 0 - ТОЧКИ НЕТ. ТАК 7-0Й БИТ ЯЧЕЙКИ 0E003H СООТВЕТСТВУЕТ ТОЧКЕ С КООРДИНАТАМИ (0,3).ОПЕРАТОР POKE &E003,128 ВЫЗОВЕТ ПОЯВЛЕНИЕ ТОЧКИ ЦВЕТА 1 С КООРДИНАТАМИ (0,3). ЕСЛИ МЫ ТЕПЕРЬ БУДЕМ НАРАЩИВАТЬ АДРЕС НА 1,ТО ТОЧКА БУДЕТ ПОДНИМАТЬСЯ ВВЕРХ ПО НУЛЕВОМУ СТОЛБИКУ, А ЗАТЕМ, ДОЙДЯ ДО ВЕРХА ЭКРАНА, ОКАЖЕТСЯ ВНИЗУ 1-ГО СТОЛБИКА И.Т.Д.

НАПРИМЕР, 10 SCREEN 2,15:CLS:SCREEN 0,0,255
```
      20 HH=&E000( ИЛИ HH=57344):M=128
      30 FOR I=1 TO 8192: POKE HH,M: HH=HH+1: NEXT I
```
КАЖДАЯ ТОЧКА НА ЭКРАНЕ ИМЕЕТ СВОЙ МАТЕМАТИЧЕСКИЙ ЦВЕТ, КОТОРЫЙ КОДИРУЕТСЯ ЧЕТЫРЬМЯ БИТАМИ. ЭТИ БИТЫ ХРАНЯТСЯ СООТВЕТСТВЕННО В 1-ОЙ, 2-ОЙ, 3-ЕЙ И 4-ОЙ ПЛОСКОСТЯХ. ОТ ЧИСЛА ПЛОСКОСТЕЙ, ИСПОЛЬЗУЕМЫХ ДЛЯ ХРАНЕНИЯ КАРТИНКИ, ЗАВИСИТ КОЛИЧЕСТВО ЦВЕТОВ, ИЗ КОТОРЫХ МОЖЕТ СОСТОЯТЬ ИЗОБРАЖЕНИЕ. НАПРИМЕР, ДЛЯ

3-Х ПЛОСКОСТЕЙ - 8 ЦВЕТОВ, ДЛЯ 2-Х - 4 ЦВЕТА, ДЛЯ 1-ОЙ - 2 ЦВЕТА.

16 ОДНОВРЕМЕННО ИСПОЛЬЗУЕМЫХ ЦВЕТОВ ХРАНЯТСЯ В ТАБЛИЦЕ ЦВЕТОВ,КОТОРАЯ УСТАНАВЛИВАЕТ СООТВЕТСТВИЕ МЕЖДУ МАТЕМАТИЧЕСКИМИ И ФИЗИЧЕСКИМИ ЦВЕТАМИ.

0-ОЙ ЦВЕТ ОПРЕДЕЛЯЕТ ЦВЕТ ФОНА, 1-ЫЙ - ЦВЕТ 1-ОЙ ПЛОСКОСТИ,2-ОЙ - ЦВЕТ 2-ОЙ ПЛОСКОСТИ, 4-ЫЙ - ЦВЕТ 3-ЕЙ ПЛОСКОСТИ, 8-ОЙ - ЦВЕТ 4-ОЙ ПЛОСКОСТИ. НАПРИМЕР, ЕСЛИ МЫ УСТАНОВИМ ЛЮБОЙ БИТ В 3-ЕЙ ПЛОСКОСТИ В 1, ТО ПОЛУЧИМ НА ЭКРАНЕ ТОЧКУ С 4-М МАТЕМАТИЧЕСКИМ ЦВЕТОМ. ПРИ НАЛОЖЕНИИ КАРТИНОК, НАХОДЯЩИХСЯ В РАЗНЫХ ПЛОСКОСТЯХ, ВОЗНИКАЕТ НОВЫЙ ЦВЕТ, КОТОРЫЙ ОПРЕДЕЛЯЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМ. ЕСЛИ НАКЛАДЫВАЮТСЯ 1-АЯ И 2-АЯ ПЛОСКОСТИ, ТО ВОЗНИКАЕТ 1+2=3-ИЙ МАТЕМАТИЧЕСКИЙ ЦВЕТ. ЕСЛИ 2-АЯ И 4-АЯ, ТО 2+8=10-Й МАТЕМАТИЧЕСКИЙ ЦВЕТ. ЕСЛИ 2-АЯ,3-ЬЯ И 4-АЯ, ТО 2+4+8=14-Й МАТЕМАТИЧЕСКИЙ ЦВЕТ И.Т.Д.

НАПРИМЕР:
```
10 SCREEN 2,15: CLS: SCREEN 0,0,255,192,54
20 SCREEN 2,1: COLOR1: PLOT 0,0,1: LINE 100,100,BF
30 SCREEN 2,2: COLOR2: PLOT 50,50,1: LINE 150,150,BF
40 SCREEN 2,15
```

ЕСЛИ У ВАС В 1-ОЙ ПЛОСКОСТИ НАХОДИТСЯ ФОНОВАЯ КАРТИНКА, А ВО 2-ОЙ ПЛОСКОСТИ ПОДВИЖНЫЙ СПРАЙТ,И ВАМ НУЖНО, ЧТОБЫ СПРАЙТ "ЗАХОДИЛ" ПОД НЕПОДВИЖНУЮ КАРТИНКУ, ТО ДОСТАТОЧНО УСТАНОВИТЬ 3-ИЙ МАТЕМАТИЧЕСКИЙ ЦВЕТ (ЦВЕТ ИХ НАЛОЖЕНИЯ) РАВНЫМ 1-МУ МАТЕМАТИЧЕСКОМУ ЦВЕТУ:

```
10 SCREEN 2,15: CLS: SCREEN 0,0,255,192,255: DIM A(33)
20 COLOR2: PLOT1,1,1: LINE14,14,BF: PLOT1,1,2: GET16,16,ADDR(A(0))
30 COLOR1: PLOT 0,0,1: LINE 100,100,BF: SCREEN 2,2
40 FOR Y=200 TO 0 STEP -1:PUT10,Y,ADDR(A(0)),2: NEXT Y
50 GOTO 40
```

А ЕСЛИ ВЫ ХОТИТЕ, ЧТОБЫ СПРАЙТ БЫЛ НАВЕРХУ, ТО 3-ИЙ ЦВЕТ УСТАНОВИТЕ РАВНЫМ 2-МУ МАТЕМАТИЧЕСКОМУ ЦВЕТУ.

ЕСЛИ ВАМ НУЖНО УВЕЛИЧИТЬ ОЗУ ЗА СЧЕТ, СКАЖЕМ, 4-ОЙ ЭКРАННОЙ ПЛОСКОСТИ, ТО КОМАНД SCREEN 2,7 И HIMEM &A000 НЕДОСТАТОЧНО , МОГУТ ВОЗНИКНУТЬ НЕПРИЯТНЫЕ ЭФФЕКТЫ. ПОЭТОМУ ЕЩЕ НЕОБХОДИМО ПЕРЕУСТАНОВИТЬ ТАБЛИЦУ ЦВЕТОВ ТАКИМ ОБРАЗОМ, ЧТОБЫ НАЛИЧИЕ ИЛИ ОТСУТСТВИЕ ИНФОРМАЦИИ В 4-ОЙ ПЛОСКОСТИ НЕ ВЛИЯЛО НА СОДЕРЖИМОЕ ЭКРАНА. ТО ЕСТЬ УСТАНОВИТЬ 8-ОЙ ЦВЕТ, РАВНЫМ 0-МУ, 9-Й - 1-МУ,10-Й - 2-МУ, ... , 15-ЫЙ - 7-МУ.

СОДЕРЖИМОЕ ТАБЛИЦЫ ЦВЕТОВ МОЖНО ПРОГРАММНО МЕНЯТЬ ПО СВОЕМУ УСМОТРЕНИЮ ОПЕРАТОРОМ SCREEN0. ЭТО ПОЗВОЛЯЕТ ПОЛУЧИТЬ РЯД КРАСИВЫХ ВИДЕОЭФФЕКТОВ. ВО МНОГИХ ПРОГРАММАХ ИСПОЛЬЗУЕТСЯ ЭФФЕКТ ВНЕЗАПНО ПОЯВЛЯЮЩЕЙСЯ КАРТИНКИ. ДЕЛАЕТСЯ ЭТО ТАК. СНАЧАЛА ВСЕ ЦВЕТА В ТАБЛИЦЕ ЦВЕТОВ УСТАНАВЛИВАЮТСЯ РАВНЫМИ ЦВЕТУ ФОНА. ЗАТЕМ, КАК ОБЫЧНО, РИСУЕТСЯ КАРТИНКА, НО ОНА НЕ ВИДНА НА ЭКРАНЕ. ПОСЛЕ ЭТОГО ЦВЕТА В ТАБЛИЦЕ ПЕРЕУСТАНАВЛИВАЮТСЯ В ТЕ ЦВЕТА, КОТОРЫЕ ВАМ НУЖНЫ, И КАРТИНКА "ПРОЯВЛЯЕТСЯ".

ТАКЖЕ ЧАСТО С ПОМОЩЬЮ СМЕНЫ ТАБЛИЦЫ ЦВЕТОВ ДОБИВАЮТСЯ ИЛЛЮЗИИ ДВИЖЕНИЯ.НАПРИМЕР:

```
10 CLS: SCREEN 2,15: DIM C(15): GOTO30
20 DATA 64,128,16,208,6,134,22,54,0,197,34,192,2,152,82,173
30 FOR N=0 TO 15: READ C(N): NEXT N  ; СЧИТЫВАЕМ МАССИВ ЦВЕТОВ
40 GOSUB 100 ;УСТАНАВЛИВАЕМ ТАБЛИЦУ ЦВЕТОВ
50 FOR N=1 TO 15:COLORN: PLOT 20+5*N,20+5*N,1: LINE 40+5*N,40+5*N,BF:NEXT N
60 CC=C(1): FOR N=1 TO 14: C(N)=C(N+1): NEXT N: C(15)=CC
70 GOSUB 100: GOTO 60; ПЕРЕУСТАНАВЛИВАЕМ ЦВЕТА
100 SCREEN 0,0,C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),C(9),
           C(10),C(11),C(12),C(13),C(14),C(15)
110 RETURN
```

ДЛЯ ТЕХ, КТО ИНТЕРЕСУЕТСЯ КОДАМИ, ДОБАВИМ СЛЕДУЮЩЕЕ. КОДИРОВКА ТАБЛИЦЫ ЦВЕТОВ ПРОИЗВОДИТСЯ ПОРТАМИ 02 И 0С ВО ВРЕМЯ ПРЕРЫВАНИЯ. ПРИЧЕМ, В ПОРТ 02 ЗАНОСИТСЯ МАТЕМАТИЧЕСКИЙ ЦВЕТ,А В ПОРТ 0С - ФИЗИЧЕСКИЙ. В ОСТАЛЬНОЕ ВРЕМЯ ПОРТ

02 СОДЕРЖИТ МАТЕМАТИЧЕСКИЙ ЦВЕТ БОРДЮРА. ПРИВЕДЕМ ПРИМЕР.

```
 TABC: DB 0,192,54,7,255,84,193,64,16,22,34,82,2,208,6,173
 UST:  LXI H,TABC+15
       LXI B,100FH
 UST1: XRA A
       OUT 02
       EI
       HLT
       MOV A,C
       OUT 2    ;ЗАПИСЬ МАТЕМ. ЦВЕТА В ПОРТ 2
       MOV A,M
       OUT 0CH  ;ЗАПИСЬ ФИЗИЧ. ЦВЕТА В ПОРТ 0С
       DCR C
       DCX H
       DCR B
       JNZ UST1
       MVI A,4
       OUT 2    ;ЗАПИСЬ МАТЕМ. ЦВЕТА БОРДЮРА
       RET
```


ПОДКЛЮЧЕНИЕ ПЭВМ "ВЕКТОР-06Ц" К ТЕЛЕВИЗОРУ

В ТЕЛЕВИЗОРАХ, ВЫПУСКАЕМЫХ В НАСТОЯЩЕЕ ВРЕМЯ , БЛОК ЦВЕТНОСТИ СОБРАН НА М/С К174ХА17 (TDA3501), TDA3505. В ЭТИХ МИКРОСХЕМАХ ПРЕДУСМОТРЕН СПЕЦИАЛЬНЫЙ RGB-ВХОД, ПЕРЕКЛЮЧАЕМЫЙ ЭЛЕКТРИЧЕСКИ. ДЛЯ ПРАВИЛЬНОГО ВОСПРОИЗВЕДЕНИЯ ЦВЕТОВ НА ЭКРАНЕ СИГНАЛЫ, ПОДАВАЕМЫЕ ОТ КОМПЬЮТЕРА, НЕОБХОДИМО ИНВЕРТИРОВАТЬ ПО ПРИВЕДЕННОЙ СХЕМЕ. СХЕМА ПРОСТА И ПРИ ПРАВИЛЬНОМ МОНТАЖЕ НЕ ТРЕБУЕТ НАСТРОЙКИ. МОДУЛИ А2 И А3 АНАЛОГИЧНЫ А1. РЕЗИСТОРЫ, ОБОЗНАЧЕННЫЕ ПУНКТИРОМ, НЕОБХОДИМО ВЫПАЯТЬ. ДЛЯ ПЕРЕКЛЮЧЕНИЯ ТЕЛЕВИЗОРА ПРИ РАБОТЕ С КОМПЬЮТЕРОМ ВАМ ПОТРЕБУЕТСЯ ТУМБЛЕР С ДВУМЯ ПЕРЕКЛЮЧАЮЩИМИ ГРУППАМИ.

АНАЛОГИЧНЫЕ ИНВЕРТОРЫ НЕОБХОДИМЫ ПРИ ПОДКЛЮЧЕНИИ К МОНИТОРУ "ЭЛЕКТРОНИКА".

ЧИТАЙТЕ В СЛЕДУЮЩЕМ ВЫПУСКЕ: ЗАЩИТА ПРОГРАММ НА БЕЙСИКЕ И В КОДАХ. ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ В ДРАЙВЕРАХ УСТРОЙСТВ.СТАНДАРТНЫЕ ПОДПРОГРАММЫ МОНИТОРА.

"БАЙТ-3" ИНФОРМАЦИОННО-РЕКЛАМНЫЙ ВЫПУСК
ДЛЯ ПОЛЬЗОВАТЕЛЕЙ БПЭВМ "ВЕКТОР-06Ц"
август 1991г.

1. ИЗ ПИСЕМ ЧИТАТЕЛЕЙ

ПРИ НАЖАТИИ НА КЛАВИШИ УС+D ИНТЕРПРИТАТОРBASIC НЕ СХОДИТ С УМА-А ПЕЧАТАЕТ СЖАТЫЙ СИМВОЛ ТОЛЬКО ПОСЛЕ ТОГО-КАК БУДЕТ ЕЩЕ +' РАЗ НАЖАТА ЛЮБАЯ КЛАВИША. КОГДА Я ЭТО ВЫЯСНИЛ. ТО НАШЕЛ СПОСОБ ИСПРАВИТЬ ПРОГРАММУ.ДЛЯ ЭТОГО +' РАЗ НУЖНО НАЖАТЬ КЛАВИШУ УС+D.И ЕЩЕ ИНТЕРПРИТАТОР МОЖЕТ-СВИХНУТЬСЯ-ПРИ ИСПОЛЬЗОВАНИИ В ОПЕРАТОРЕ PRINT ОПЕРАТОРА НОМЕ-ЕСЛИ ОН ТАМ СЛУЧАЙНО ОКАЖЕТСЯ.

2. А ЗНАЕТЕ ЛИ ВЫ?

КАК ИГРАТЬ В "WARP & WARP".ПО ЭКРАНУ ВАС ПРЕСЛЕДУЮТ ЧУДОВИЩА.ОНИ ПОСТЕПЕНО ИЗМЕНЯЮТ СВОЙ ЦВЕТ-СТАНОВЯСЬ ТО ЖЕЛТЫМИ-ТО КРАСНЫМИ-ТО РОЗОВЫМИ. ЗА КАЖДОГО ЖЕЛТОГО ДАЕТСЯ с_ ОЧКОВ-ЗА КРАСНОГ_##_-ЗА РОЗОВОГО#+'_.ЕСЛИ ВЫ УБЬЕТЕ ПОДРЯД ' ЖЕЛТЫХ-ТО ВАМ ПРЕДОСТАВЛЯЕТСЯ ШАНС УБИТЬ ЗЕЛЕНУЮ ЖАБУ И ПОЛУЧИТЬ '__ ОЧКОВ.

ЕСЛИ # ' КРАСНЫХ-ТО ПОЯВИТСЯ ОСЬМИНОГ(1000 ОЧКОВ),ЕСЛИ#ТРЕХ РОЗОВЫХ-ТО ПОЯВЯТСЯ КУСАЧКИ (2000 ОЧКОВ).ОБРАТИТЕ ВНИМАНИЕ-ЧТО УБИВАТЬ НУЖНО ТРЕХ ОБНОЦВЕТНЫХ ПОДРЯД.ПРИ ПРАВИЛЬНОЙ ТАКТИКЕ "УБИВАТЬ ТОЛЬКО РОЗОВЫХ┐ - ВЫ СМОЖЕТЕ НАБРИТЬ ТОЛЬКО ЗА ПЕРВЫЙ УРОВЕНЬ#15000 ОЧКОВ.КРОМЕ ТОГО В ИГРЕ ЕСТЬ ДОПОЛНИТЕЛЬНАЯ КЛАВИША # АР_ # ПАУЗА.

ПЕСТОВ С.Е.,КИРОВ.

В ИГРЕ PUSHER-M МОЖНО ПЕРЕЙТИ НА ЛЮБОЙ ПРОЙДЕННЫЙ ВАМИ ЛАБИРИНТ-ЗНАЯ ЕГО НАЗВАНИЕ .ОНО ПИШЕТСЯ ВВЕРХУ ЭКРАНА)!ДЛЯ ЭТОГО НУЖНО ВО ВРЕМЯ ИГРЫ НАЖАТЬ ОДИН РАЗ КЛАВИШУ"ПРОБЕЛ".ПРИ НАЖАТИИ ВТОРОЙ РАЗ БОЧКИ НА УРОВНЕ УСТАНОВЯТСЯ В ПЕРВОНАЧАЛЬНОЕ ПОЛОЖЕНИЕ). ПОСЛЕ ЭТОГО ПОД ПЕРВОЙ БУКВОЙ НАЗВАНИЯ УРОВНЯ ПОЯВИТСЯ "КУРСОР", КОТОРЫЙ МОЖНО ПЕРЕДВИГАТЬ ВЛЕВО ИЛИ ВПРАВО .КЛАВИШИ:"СТРЕЛКА ВЛЕВО" И "СТРЕЛКА ВПРАВО").ПРИ НАЖАТИИ НА КЛАВИШИ"СТРЕЛКА ВВЕРХ"ИЛИ "СТРЕЛКА ВНИЗ" БУКВЫ НАД КУРСОРОМ БУДУТ МЕНЯТЬСЯ ОНИ РАСПОЛОЖЕНЫ ПО АЛФАВИТУ И НАЖАТИЕ ЭТИХ КЛАВИШ ПЕРЕБИРАЕТ ИХ СООТВЕТСТВЕННО ВПЕРЕД ИЛИ НАЗАД.

ИЗМЕНИВ ТАКИМ ОБРАЗОМ НАЗВАНИЕ И НАЖАВ КЛАВИШУ "ПРОБЕЛ", ВЫ ПЕРЕЙДЕТЕ НА ДАННЫЙ ЛАБИРИНТ.БУДЬТЕ ВНИМАТЕЛЬНЫ-ТАК КАК ПРИ НЕПРАВИЛЬНОМ НАБОРЕ "ПРИ ОШИБКЕ В НАЗВАНИИ┐ ПРОГРАММА ПЕРЕЙДЕТ НА НАЧАЛО.

СПИСОК НАЗВАНИЙ ЛАБИРИНТОВ В ИГРЕ "PUSHER-M":

```
+"НАЧАЛО          с"КРЕСТ         +_"КОМПЬЮТЕР
_"КИРПИЧНЫЙ ЗАВАЛ  +"УСПИД        ++"ТРИ СУНДУКА
'"СУНДУЧНЫЕ РЯДЫ   _"ЛАБИРИНТ     +_"ВАДЦАТЬ А РИНАДЦАТЬ
_"КАМЕРА ТРИ       #"КАРАКАТАЦА   +'"КОРИДОРИСТЫЙ
'"ПИРАМИДЫ
```

ТМК SOFT Г.КИРОВ

3. ЗАЩИТА ПРОГРАММ ОТ ТИРАЖИРОВАНИЯ

С ТЕХ ПОР- КАК ПОЯВИЛИСЬ КОМЕРЧЕСКИЕ ПРОГРАММЫ- ПОЯВИЛИСЬ И НЕПРЕРЫВНО СОВЕРШЕНСТВУЮТСЯ МЕТОДЫ ИХ ЗАЩИТЫ ОТ НЕЗАКОННОГО ТИРАЖИРОВАНИЯ.ПОЛЬЗОВАТЕЛИ СТРЕМЯТСЯ ПОЛУЧИТЬ КОПИИ ПРОГРАММ БЕСПЛАТНО ИЛИ ПО ЦЕНЕ НАМНОГО МЕНЬШЕ ОФИЦИАЛЬНОЙ.А ОРГАНИЗАЦИИ-КОТОРЫЕ ТИРАЖИРУЮТ АВТОРСКИЕ ПРОГРАММЫ- ПРЕПЯТСТВУЮТ ЭТОМУ НАВЕШИВАЯ НА НИХ ЗАЩИТЫ.АНАЛОГИЧНО ОБСТОИТ ДЕЛО И С "ВЕКТОРОМ". ПОДАВЛЯЮЩАЯ ЧАСТЬ ЛЮБИТЕЛЕЙ "ВЕКТОРА-06Ц" ПЕРЕПИСЫВАЕТ ПРОГРАММЫ У ДРУЗЕЙ- ЗНАКОМЫХ И.Т.Д.-НЕ ОПАСАЯСЬ НЕ КАКИХ ВИРУСОВ- ЧЕР-

ВЕЙ И Т.П. В ЭТОЙ СТАТЬЕ МЫ ОПИШЕМ НЕСКОЛЬКО МЕР ЗАЩИТЫ. КОНЕЧНО- ПОКА РАНО ГОВОРИТЬ ОБ ЭФФЕКТИВНОЙ ЗАЩИТЕ ПРОГРАММ-ТАК КАК БОЛЬШИНСТВО ПОЛЬЗОВАТЕЛЕЙ НЕ ИМЕЮТ ДИСКОВОДОВ,НО ОПРЕДЕЛЕННЫЕ МЕТОДЫ БЕЗОПАСНОСТИ МОЖНО ПРЕТВОРИТЬ В ЖИЗНЬ И В КОНФИГУРАЦИИ С МАГНИТОФОНОМ. НАЧНЕМ С КОДОВЫХ ПРОГРАММ.

ПРЕДВАРИТЕЛЬНО ВВЕДЕМ НЕБОЛЬШУЮ СЛАССИФИКАЦИЮ.БУДЕМ ВСЕ МЕТОДЫ ЗАЩИТЫ ПОДРАЗДЕЛЯТЬ НА "ВНЕШНИЕ" И"ВНУТРЕННИЕ".ВНЕШНЯЯ ЗАЩИТА НЕ ДАЕТ ПЕРЕПИСАТЬ ПРОГРАММУ И НАПРАВЛЕНА ПРОТИВ ТЕХ ПРОГРАММ С ПОМОЩЬЮ КОТОРЫХ МОЖНО ОСУЩЕСТВИТЬ КОПИРОВАНИЕ (В НАШЕМ СЛУЧАЕ ЭТО COPY И МОНИТОР). ВНУТРЕННЯ ЗЩИТА СТАРАЕТСЯ ОБНАРУЖИТЬ ФАКТ КОПИРОВАНИЯ И- В СЛУЧАЕ УСПЕХА- КАК-ТО НЕГАТИВНО РЕАГИРУЕТ.

ТЕПЕРЬ ПРИСТУПИМ НЕПОСРЕДСТВЕННО К ОПИСАНИЮ ЗАЩИТ.

ЗАЩИТА ПО ИМЕНИ

ТИПИЧНАЯ ВНЕШНЯЯ ЗАЩИТА.ОСНОВАНА НА ВНУТРЕННЕЙ СТРУКТУРЕ ПРОГРАММЫ COPY. ДЕЛО В ТОМ- ЧТО ЭТО ПРОГРАММА ПОСЛЕ ЗАГРУЗКИ КОПИРУЕМОГО ФАЙЛА И ПЕРЕД ВЫВДОМ ЕГО НА МАГНИТОФОН ДЕЛАЕТ ПРОВЕРКУ СИМВОЛОВ ЕГО ИМЕНИ.ЕСЛИ ОЧЕРЕДНОЙ СИМВОЛ НЕ УДОВЛЕТВОРЯЕТ ОПРЕДЕЛЕННЫМ ТРЕБОВАНИЯМ-ТО ЗАГОРАЕТСЯ-НЕЛЬЗЯ"И ПРОГРАММА СБРАСЫВАЕТСЯ.ДЛЯ ПОСТАНОВКИ ТАКОЙ ЗАЩИТЫ ПРОГРАММЫ ДОСТАТОЧНО ПЕРЕПИСАТЬ ЧЕРЕЗ COPY - 3 ИЛИ ВЫГРУЗИТЬ ИЗ МОНИТОРА ДИРЕКТИВОЙ "O" ПРЕДВАРИТЕЛЬНО ЗАДАВ ИМЯ ПРОГРАММЫ ПРОПИСНЫМИ БУКВАМИ" СНЯТИЕ ЗАЩИТЫ ОСУЩЕСТВЛЯЕТСЯ ДОСТАТОЧНО ПРОСТЫМИ СРЕДСТВАМИ" В НАЧАЛЕ -ПИРАТЫ- ПОЛЬЗОВАЛИСЬ МОНИТОРОМ" ОНИ ЗАГРУЖАЛИ В ОЗУ МОНИТОР- ЗАПУСКАЛИ ЕГО- ЗАДАВАЛИ ОБЛАСТЬ РАСПОЛОЖЕНИЯ (1), НАЖИМАЛИ УС+ВВОД+БЛК, ЗАГРУЖАЛИ ПРОГРАММУ:НАЖАВ БЛК+СБР,ПЕРЕХОДИЛИ В МОНИТОР И ВЫГРУЖАЛИ ПРОГРАММУ ДИРЕКТИВОЙ "O". ОБНАКО ТАКОЙ СПОСОБ НЕ ГОДИЛСЯ ДЛЯ СНЯТИЯ ЗАЩИТЫ С ПР#ММ С _#М БЛОКОМ" В НАСТОЯЩЕЕ ВРЕМЯ СУЩЕСТВУЕТ ПРОГРАММА COPY-HELP ВНЕШНЕ ИДЕНТИЧНАЯ COPY -V2.1,КОТОРАЯ ПОЗВОЛЯЕТ КОПИРОВАТЬ ВСЕ ПРОГРАММЫ С ЗАЩИТОЙ ПО ИМЕНИ" В ЭТОЙ ПРОГРАММЕ ЗАТЕРТЫ ВСЕ ПЕРЕХОДЫ НА АДРЕС 4C6H, ПО КОТОРОМУ ПРОИСХОДИТ УСТАНОВКА ПРИЗНАКОВ ЗАПРЕТА ДЛЯ ТИРАЖИРОВАНИЯ"В ИТОГЕ ЗАЩИТА ПО ИМЕНИ ОКАЗАЛАСЬ ПОЛНОСТЬЮ НЕЙТРАЛИЗОВАННОЙ.

ЗАЩИТА ПО КОНСТАНТЕ ЧТЕНИЯ

ОТНОСИТСЯ К ВНУТРЕННИМ ЗАЩИТАМ"ДЕЛО В ТОМ-ЧТО НАЧАЛЬНЫЙ ЗАГРУЗЧИК ХРАНИТ В Я;ЧЕЙКЕ ПО АДРЕСУ 0DEF6H КОНСТАНТУ ЧТЕНИЯ-В СООТВЕТСТВИИ С КОТОРОЙ И ЗАГРУЖАЕТ ИНФОРМАЦИЮ В ОЗУ" АВТОР ПРОГРАММЫ МОЖЕТ ТИРАЖИРОВАТЬ ЕЕ НА НЕБОЛЬШОЙ СКОРОСТИ И ПРЕДУСМОТРЕТЬ В САМОЙ ПРОГРАММЕ ПРОВЕРКУ ЭТОЙ КОНСТАНТЫ ЧТЕНИЯ"ЛЮБОЙ -ПИРАТ--ПОЛУЧИВ ПРОГРАММУ ТУТ ЖЕ ПЕРЕПИШЕТ ЕЕ НА ПОВЫШЕННОЙ СКОРОСТИ-НЕ ЖЕЛАЯ КАЖДЫЙ РАЗ ЖДАТЬ ПО +_ МИН"ЗАГРУЗКИ ПРОГРАММЫ"ПОСЛЕ ЗАПУСКА ПРОГРАММ ВКЛЮЧАЕТСЯ ПРОВЕРКА КОНСТАНТЫ ЧТЕНИЯ И ЕСЛИ ОНА ВЫХОДИТ ЗА РАМКИ ЕСТЕСТВЕННЫХ КОЛЕБАНИЙ- ПРОГРАММА СБРАСЫВАЕТСЯ"ПРЕДЕЛЫ КОЛЕБАНИЙ НЕОБХОДИМЫ ИЗ-ТОГО-ЧТО ЛЕНТОПРОТЯЖНЫЕ МЕХАНИЗМЫ У РАЗНЫХ МАГНИТОФОНОВ ОТЛИЧАЮТСЯ ПО СКОРОСТИ ПЕРЕМОТКИ ЛЕНТЫ.

СИНКЛЕР-ЗАЩИТА

ОТНОСИТСЯ К ВНЕШНИМ ЗАЩИТАМ. ПРИМЕНЯЕТСЯ В НАСТОЯЩЕЕ ВРЕМЯ КИШИНЕВСКИМ ЦЕНТРОМ -КОМПЬЮТЕР--НА ТИРАЖИРУЕМЫХ ПРОГРАММАХ. АВТОР-ФИЛЛИПОВ- ОДИН ИЗ АВТОРОВ MOONBUGS,НАЗЫВАЕТСЯ ТАК ПОТОМУ ЧТО ОДИН ИЗ ВАРИАНТОВ ЗАЩИТЫ ВОСПРОИЗВОДИТ ВНЕШНИЕ ЭФФЕКТЫ ЗАГРУЗКИ ПРОГРАММ В СИНКЛЕРЕ .ПО ЭКРАНУ ИДЕТ ПИЛОТ ТОН. ЗАЩИТА БОЛЕЕ СЛОЖНАЯ-ЧЕМ ЗАЩИТА ПО ИМЕНИ-НО НЕ НАСТОЛЬКО ЧТОБЫ СРЕДНИЙ УМЕЛЫЙ ПРОГРАММИСТ ЕЕ НЕ СНЯЛ.

СУТЬ ЗАЩИТЫ В ТОМ-ЧТО САМА ПРОГРАММА ЗАПИСАНА В ФОРМАТЕ-КОТОРЫЙ НЕ БЕРУТ НИКАКИЕ СУЩЕСТВУЮЩИЕ КОПИРУЮЩИЕ УТИЛИТЫ.ЭТО НЕ БЕЙСИК-ФОРМАТ-НЕ ФОРМАТ МОНИТОРА- НЕ ROM-ФОРМАТ И НЕ ФОРМАТ АССЕМБЛЕРА-РЕДАКТОРА. ЭТО НОВЫЙ ФОРМАТ-ПОЭТОМУ И КОПИРОВАНИЕ ЭТОГО ФОРМАТА БЕЗ ВСПОМОГАТЕЛЬНЫХ ПРОГРАММ

НЕВОЗМОЖНО.

ПРОЦЕСС ЗАГРУЗКИ ПРОГРАММ В КОМПЬЮТЕР СЛЕДУЮЩИЙ:СНАЧАЛА ГРУЗИТСЯ НУЛЕВОЙ БЛОК-ЗАГРУЗЧИК В ФОРМАТЕ ROM.ЗАТЕМ МИГАЕТ ИНДИКАТОР РУС/ЛАТ, ВЫ ЗАПУСКАЕТЕ ЗАГРУЗЧИК КЛАВИШАМИ БЛК+СБР. ЗАТЕМ ГРУЗИТСЯ ЗАСТАВКА В ФОРМАТЕ НОВОГО ЗАГРУЗЧИКА"ПРИ ЭТОМ ЭКРАН ОСТАЕТСЯ ЛИБО ТЕМНЫМ- ЛИБО ПО БОРДЮРУ ИДЕТ ПИЛОТ#ТОН-ЛИБО СТИРАЮТСЯ БАЙТЫ#МЕТКИ- ОПРЕДЕЛЯЮЩИЕ СКОЛЬКО БЛОКОВ ЕЩЕ ЗАГРУЗИТЬ (ЭТО ВАРИАНТ ПРИМЕНЯЕТСЯ В "Карандаше" и "Жизни"). ЗАТЕМ ЗАСТАВКА РИСУЕТСЯ НА ЭКРАНЕ И НАЧИНАЕТСЯ ЗАГРУЗКА САМОЙ ПРОГРАММЫ"ПОСЛЕ ЧЕГО ЛИБО ЗАПУСКАЕТСЯ ПРОГРАММА ЛИБО ДОЗАГРУЖАЕТСЯ ЕЩЕ БЛОЧЕК-НА ЭКРАНЕ ПИШЕТСЯ -ПРОГРАММА ЗАГРУЖЕНА-ўИ ВЫ МОЖЕТЕ С НЕЙ РАБОТАТЬ"

СЛАБОСТЬ ЗАЩИТЫ В ТОМ-ЧТО САМ ЗАГРУЗЧИК (КСТАТИ ОН ИДЕТ С ЗАЩИТОЙ ПО СТРУКТУРЕ"ЭТО ПОЗВОЛЯЕТ ЕГО ПОСМОТРЕТЬ В МОНИТОРЕ И ПОЛНОСТЬЮ РАЗОБРАТЬСЯ В ЕГО АЛГОРИТМЕ"ПРИ ЭТОМ ОКАЗЫВАЕТСЯ- ЧТО ЗАГРУЗКА ПРОГРАММЫ ПРОИСХОДИТ БЛОКАМИ ПО _'с БАЙТ-НАЧИНАЯ С АДРЕСА 100Н. ПОСЛЕ ЗАГРУЗКИ УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ПРОГРАММЕ ЛИБО JMP100,ЛИБО CALL100,ИЛИ LXI H,100PCHL. ЕСЛИ ЭТО ЗАСТАВКА" ТО ОНА ВОЗВРАЩАЕТ УПРАВЛЕНИЕ ЗАГРУЗЧИКУ- ЕСЛИ ЭТО ПРОГРАММА- ТО ОНА НАЧИНАЕТ РАБОТАТЬ"-ПИРАТУ- ДАЖЕ НЕ НУЖНО ВДАВАТЬСЯ В СУТЬ РАБОТЫ ЗАГРУЗЧИКА" ЕМУ НУЖНО ОПРЕДЕОИТЬ ТОЧКУ ВХОДА В ПРОГРАММУ 100Н # ЫЙ АДРЕС- ПОСЛЕ ЧЕГО ОН МЕНЯЕТ JMP100 НА ПУСТОЙ ЦИКЛ МЕТКА: JMP МЕТКА И ВЫГРУЖАЕТ ИЗМЕНЕННЫЙ ЗАГРУЗЧИК ДИРЕКТИВОЙ "О".ТЕПЕРЬ ЗАГРУЗЧИК ПОСЛЕ ВВОДА ПРОГРАММЫ НЕСМОЖЕТ ПЕРЕДАТЬ ЕЙ УПРАВЛЕНИЕ-А ЗАВИСНЕТ В ПУСТОМ ЦИКЛЕ"ТОГДА-ПИРАТ-ЗАГРУЖАЕТ МОНИТОР И ВЫГРУЖАЕТ ИЗ НЕГО ПРОГРАММУ В ФОРМАТЕ НАЧАЛЬНОГО ЗАГРУЗЧИКА С +#ГО БЛОКА И ДЕЛО СДЕЛАНО- ЗАЩИТА СНЯТА.

ЗАЩИТА ПО КЛЮЧУ

ВНЕШНЯЯ ЗАЩИТА.ПОЯВЛЯЛАСЬ КОЕ-ГДЕ В КИРОВЕ.СУТЬ ЗАЩИТЫ:ПРОГРАММА ЗАКОДИРОВАНА В НЕРАБОЧУЮ ФОРМУ.ВЫ ЕЕ ВЫГРУЖАЕТЕ-ВКЛЮЧАЕТСЯ ЗАГРУЗЧИК-КОТОРЫЙ ЛОВИТ КЛЮЧ-ИДУЩИЙ ПОСЛЕ ПРОГРАММЫ-ПО ЭТОМУ КЛЮЧУ ДЕКОДИРОВЩИК РАСШИФРОВЫВАЕТ ПРОГРАММУ И ЗАПУСКАЕТ.ТРУДНОСТЬ СОСТОИТ В КОПИРОВАНИИ КЛЮЧА-КОТОРЫЙ ЗАПИСАН В ОСОБОМ ФОРМАТЕ.АНАЛОГИЧНО СИНКЛЕР-ЗАЩИТЕ.СЛАБОСТЬ ЗАЩИТЫ ТОЖЕ АНАЛОГИЧНАЯ.САМ ДЕКОДИРОВЩИК ДОЛЖЕН БЫТЬ НЕ ЗАКОДИРОВАН-Т. Е.ЕГО МОЖНО ПОСМОТРЕТЬ В МОНИТОРЕ И РАЗОБРАТЬСЯ-КАК ОН РАБОТАЕТ.А ЗАТЕМ ПРИМЕНИТЬ ПРИЕМ АНАЛОГИЧНЫЙ СИНКЛЕР-ЗАЩИТЕ-ПУСТОЙ ЦИКЛ).

ЗАЩИТА ПРОГРАММ НА БЕЙСИКЕ

ВО ПЕРВЫХ-О ТОМ КАК ЗАЩИТИТЬ СВОЮ ПРОГРАММУ ОТ ПРОСМОТРА"ДЛЯ ПРОГРАММ НА БЕЙСИКЕ"ВОПРОС ЗАЩИТЫ ПРОГРАММ ОТ ПРОСМОТРА СТОИТ БОЛЕЕ ОСТРО- ЧЕМ ДЛЯ ПРОГРАММВ В КОДАХ" ПРАКТИЧЕСКИ ЛЮБОЙ МОЖЕТ ПЕРЕПИСАТЬ ВАШУ ПРОГРАММУ И ПРОСМОТРЕТЬ ЕЕ"ОДНАКО МОЖНО ПРЕДЛОЖИТЬ РЯД ПРИЕМОВ-ЕСЛИ И НЕ СТОПРОЦЕНТНО ЗАЩИЩАЮЩИХ ВАШУ ПРОГРАММУ-ТО ПОКРАЙНЕЙ МЕРЕ-ЗАТРУДНЯЮЩИХ НЕ САНКЦИОНИРОВАННЫЙ ДОСТУП К НЕЙ"ДЛЯ НАЧАЛА МОЖНО ВОСПОЛЬЗОВАТЬСЯ ТЕМ- ЧТО-ЕСЛИ ЧИСЛО СИМВОЛОВ ПРИ ВЫВОДЕ ПРОГРАММНОЙ ШТРОКИ НА ЭКРАН ПРЕВЫШАЕТ +__- ТО ВЫДАЕТСЯ ОШИБКА"ЭТО ЧАСТО БЫВАЕТ ЕСЛИ ВЫ ПЕРЕНУМЕРОВАЛИ СТРОКИ НА БОЛЬШИЕ НОМЕРА-А СТРОКИ В ПРОГРАММЕ УЖЕ БЫЛИ БЛИЗКИ К +__ СИМВОЛАМ"ТОГДА ПРИ ЛИСТИНГЕ ПОСЛЕ ОЧЕРЕДНОЙ ПЕРЕГРУЖЕННОЙ СТРОКИ-БУДЕТ ВЫДАВТЬСЯ-ПЕРЕПОЛНЕНИЕ ОШИБКА"И ЛИСТИНГ БУДЕТ ПРЕКРАЩАТЬСЯ"ТОГО ЖЕ ЭФФЕКТА МОЖНО ДОСТИЧЬ-ЕСЛИ ПОСЛЕ НОМЕРА СТРОКИ УДАЛЯТЬ ПРОБЕЛ-А В КОНЦЕ СТРОКИ ВСТАВЛЯТЬ: REM АААААА...А ДО КОНЦА СТРОКИ (ВМЕСТО СИМВОЛА А МОЖНО ИСПОЛЬЗОВАТЬ ЗВУЧАЩИЙ СИМВОЛ УС+G, КОТОРЫЙ МЕСТА НА ЭКРАНЕ НЕ ЗАНИМАЕТ- НО ПРИ РАСПЕЧАТКЕ НАЧИНАЕТ ПИЩАТЬ-СИГНАЛИЗИРУЯ О НЕСАНКЦИОНИРОВАННОМ ДОСТУПЕ).ПРИ ЭТОМ ВАШУ ПРОГРАММУ СМОЖЕТ ПРОСМОТРЕТЬ ТОЛЬКО ОЧЕНЬ УПОРНЫЙ -ПИРАТ--КОТОРОМУ ДЛЯ ПРОСМОТРА ПРОГРАММЫ ПРИДЕТСЯ ВВОДИТ LIST NNNN СТОЛЬКО#ЖЕ РАЗ- СКОЛЬКО СТРОК В ПРОГРАММЕ"КРОМЕ ТОГО-МОЖНО ВОСПОЛЬЗОВАТЬСЯ УПРАВЛЯЮЩИМИ ПРИ ПЕЧАТИ СИМВОЛАМИ"КОД 31 =1FH - СТИРАЕТ ЭКРАН-А КОД 25=19Н-СМЕЩАЕТ КУРСОР ВВЕРХ НА СТРОКУ"ПРЕДПОЛОЖИМ ВАШЕМУ ВНИМАНИЮ СЛЕДУЮЩИЙ ПОРЯДОК ДЕЙСТВИЙ"

У ВАС ИМЕЕТСЯ ПРОГРАММ-КОТОРУЮ НУЖНО ЗАЩИТИТЬ ОТ ПРОСМОТРА- ВЫ НАБИВАЕТЕ В КОНЦЕ КАЖДОЙ СТРОКИЁREM А, ЗАТЕМ НАБИРАЕТЕ НЕБОЛЬШУЮ ПРОГРАММУ- И ЗАПУСКАЕТЕ ЕЕ RUN 10000:

```
10000 H=17153
10002 IF PEEK(H)=0 AND PEEK(H+1)=0 THEN DELETE 10000,10004
10003 IF PEEK(H)=142 AND PEEK(H+1)<>0 THEN POKE H+1,31
10004 H=H+1:GOTO 10002
```

10000-ЗАДАЕМ НАЧАЛЬНЫЙ АДРЕС ПРОГРАММЫ НА БЕЙСИКЕ
10002-ПОИСК КОНЦА ПРОГРАММЫ
10003-ПОИСК СЛУЖЕБНОГО СЛОВА REM И ЗАНЕСЕНИЕ В СЛЕДУЮЩУЮ ЗА НИМ ЯЧЕЙКУ
КОДА СТИРАНИЯ ЭКРАНА-31.
10004-НАРАЩИВАНИЕ АДРЕСА И ПЕРЕХОД К СТРОКЕ 10002.

ИЗ-ЗА МЕДЛИТЕЛЬНОСТИ БЕЙСИКА ВАМ ПРИДЕТЬСЯ НЕМНОГО ПОДОЖДАТЬ.ЗАТО ТЕПЕР ПРИ ПРОСМОТРЕ ПРОГРАММЫ КОМАНДОЙ LIST ПРОГРАММНЫЕ СТРОКИ БУДУТ ПЕЧАТАТЬСЯ НА ЭКРАНЕ И ТУТ-ЖЕ СТИРАТЬСЯ.УЛОВИТЬ О ЧЕМ В НИХ ИДЕТ РЕЧЬ БУДЕТ ДОВОЛЬНО ПРОБЛЕМАТИЧНО.
ЕСЛИ ЖЕ ВМЕСТО КОДА 31 ВСТАВИТЬ КОД 25,ТО СЛЕДУЮЩАЯ ПРОГРАММНАЯ СТРОКА БУДЕТ ПЕЧАТАТЬСЯ НА МЕСТЕ ПОСЛЕДНЕЙ ПРЕДЫДУЩЕЙ ПРОГРАММНОЙ СТРОКИ.

ТЕПЕРЬ О ТОМ КАК ЗАЩИТИТЬ ПРОГРАММУ ОТ КОПИРОВАНИЯ"СДЕЛАТЬ ЭТО БУДЕТ НАМНОГО ТРУДННЕ-ЧЕМ В КОДАХ-НО ОДНА ИДЕЯ У АВТОРОВ СТАТЬИ ЕСТЬ-И МЫ С ВАМИ ЕЮ ПОДЕЛИМСЯ.

МЫ ПРЕДЛОЖИМ ВАШЕМУ ВНИМАНИЮ ТИПИЧНУЮ ВНУТРЕННЮЮ ЗАЩИТУ-КОТОРАЯ ХОТЬ И НЕ СПАСАЕТ ОТ КОПИРОВАНИЯ-НО НЕ ДАЕТ ПРОГРАММЕ РАБОТАТЬ ПОСЛЕ ЭТОГО.ДЕЛО В ТОМ-ЧТО ПРИ ЗАГРУЗКЕ ПРОГРАММЫ КОМАНДАМИ CLOAD и BLOAD ЕЕ ИМЯ ПОМЕЩАЕТСЯ В БУФЕР ПО АДРЕСУ 3F08H.ТУТ-ЖЕ ВОЗНИКАЕТ МЫСЛЬ О ПРОВЕРКЕ ЭТОГО БУФЕРА ПОСЛЕ ЗАПУСКА ПРОГРАММЫ И-ЕСЛИ ИМЯ НЕ СОВСЕМ ТО(ТО ЕСТЬ КТО#ТО ИЗМЕНЯЛ ЕГО-А СДЕЛАТЬ ЭТО МОЖНО ЛИШЬ ПРИ КОПИРОВАНИИ ПРОГРАММ┐ -ТОГДА СЛЕДУЕТ НЕГАТИВНАЯ РЕАКЦИЯ.МОЖНО СДЕЛАТЬ ТАК-ЧТО ПРОГРАММА ИЗНАЧАЛЬНО НЕ РАБОТОСПОСОБНА- А ИСПРАВЛЯЕТСЯ ЛИШЬ ПОСЛЕ УСПЕШНОЙ ПРОВЕРКИ ИМЕНИ И Т.Д.

ВСТАЮТ ДВЕ ПРОБЛЕМЫ :
1.ЗАДАТЬ ИМЯ ТРУДНОЕ ДЛЯ ВОСПРОИЗВЕДЕНИЯ;
2.ПРОВЕРИТЬ ЕГО.

ПРОБЛЕМА ПЕРВАЯ:САМОЕ ПРОСТОЕ-ЗАДАТЬ В ИМЕНИ N-ОЕ КОЛ-ВО НЕВИДИМЫХ, НО ЗВУЧАЩИХ СИМВОЛОВ УС+G (КОД07).ПРИ ЗАГРУЗКЕ ПРОГРАММЫ ТОЛЬКО ПРОФЕССИОНАЛЬНЫЙ МУЗЫКАНТ СМОЖЕТ ОПРЕДЕЛИТЬ-КАКОЕ КОЛИЧЕСТВО СИГНАЛОВ ВЫ ЗАДАЛИ. А ЕСЛИ ОНА К ТОМУ ЖЕ ЧЕРЕДУЮТСЯ С ОБЫЧНЫМИИМВОЛАМИ-ТО ЭТО ПОЧТИ НЕВОЗМОЖНО("ПОЧТИ"ОСТАВИМ НА ДОЛЮ СУПЕРМЕНОВ).БОЛЕЕ СЛОЖНЫЕ СПОСОБЫ ПОДРАЗУМЕВАЮТ ХОРОШЕЕ ЗНАНИЕ СТРУКТУРЫ БЕЙСИКА- ПОЭТОМУ МЫ ИХ ОПУСКАЕМ.

ПРОБЛЕМА ВТОРАЯ:ОПЕРАТОРЫ POKE и PEEK ЗДЕСЬ НЕ ПОМОГУТ-ТАК КАК-САМ ИНТЕРПРИТАТОР ПРЕДОХРАНЯЕТ СЕБЯ ОТ ПРОСМОТРА И ИЗМЕНЕНИЯ" ВЫХОД ТОЛЬКО В ПРОГРАММЕ В МАШИННЫХ КОДАХ-ДЛЯ КОТОРОЙ ТАКИХ ТРУДНОСТЕЙ НЕ СУЩЕСТВУЕТ. ВЫ ПЕРЕДАЕТЕ ЕЙ УПРАВЛЕНИЕ USR (ADDR),ОНА СРАВНИВАЕТ СОДЕРЖИМОЕ БУФЕРА С ШАБЛОНОМ В ПАМЯТИ И ЛИБО ВОЗВРАЩАЕТ УПРАВЛЕНИЕ(ЕСЛИ ВСЕ НОРМАЛЬНО),ЛИБО ПОРТИТ БЕЙСИК"ДЛЯ ЛУЧШЕЙ ЗАЩИТЫ-ШАБЛОН ЗАДАВАЙТЕ РАВНЫМ ИСХОДНОМУ ИМЕНИ С ДОБАВЛЕНИЕМ ИЛИ С ВЫЧИТАНИЕМ КАКОЙ-ТО КОНСТАНТЫ.

БОЛЕЕ СИЛЬНАЯ ЗАЩИТА ВОЗМОЖНА ТОЛЬКО ПРИ БОЛЕЕ ТЩАТЕЛЬНОМ ИЗУЧЕНИИ РАБОТЫ И СТРУКТУРЫ ИНТЕРПРИТАТОРА.

В ЗАКЛЮЧЕНИИ: СЕЙЧАС НА ЗАПАДЕ ГОСПОДСТВУЕТ ТЕНДЕНЦИЯ НЕ К НАВЕШИВАНИЮ ЗАЩИТ-А К СОПРОВОЖДЕНИЮ ПРОГРАММНОГО ПРОДУКТА РАЗВИТЫМ СЕРВИСОМ: ОПИСАНИЯ-ДОКУМЕНТАЦИЯ-ПОСТАВКА НОВЫХ ВЕРСИЙ И Т.Д.

"БАЙТ-4"
ИНФОРМАЦИОННО-РЕКЛАМНЫЙ ВЫПУСК ДЛЯ ПОЛЬЗОВАТЕЛЕЙ БПЭВМ "ВЕКТОР-06Ц"
СЕНТЯБРЬ 1991 Г.

ПРЕРЫВАНИЯ, ПОРТЫ И ДРУГОЕ

В ЭТОЙ СТАТЬЕ МЫ ХОТИМ ПРЕДЛОЖИТЬ ВАМ ТЕКСТ ПОДПРОГРАММЫ ОБРАБОТКИ ПРЕРЫВАНИЙ В ДРАЙВЕРАХ УСТРОЙСТВ С ПОДРОБНЫМИ КОММЕНТАРИЯМИ. НО В НАЧАЛЕ НЕСКОЛЬКО СЛОВ О ПРЕРЫВАНИЯХ И ПОРТАХ.ПОНЯТИЕ И МЕХАНИЗМ ПРЕРЫВАНИЯ БЫЛИ ВВЕДЕНЫ ДЛЯ ОРГАНИЗАЦИИ ВЗАИМОДЕЙСТВИЯ ПРОЦЕССОРА С ВНЕШНИМИ УСТРОЙСТВАМИ. В ЭВМ "ВЕКТОР-06Ц"ПРЕРЫВАНИЙ ВОСЕМЬ: RST0-RST7.ПРИ ИНИЦИАЛИЗАЦИИ ПРЕРЫВАНИЯ ВАПОЛНЕНИЕ ОСНОВНОЙ ПРОГРАММЫ ПРИОСТАНАВЛИВАЕТСЯ И УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ НА СООТВЕТСТВУЮЩУЮ ПОДПРОГРАММУ ОБРАБОТКИ ПРЕРЫВАНИЙ. ЗАТЕМ ПОСЛЕ КОМАНДЫ RET УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ОБРАТНО НА ОСНОВНУЮ ПРОГРАММУ.В ТАБЛИЦЕ + ПРИВЕЕНЫ АДРЕСА–НА КОТОРЫЕ ПЕРЕДАЕТСЯ УПРАВЛЕНИЕ ПРИ ВЫПОЛНЕНИИ КОМАНД RST N:

RST	I 0	I 1	I 2	I 3	I 4	I 5	I 6	I 7 I
ТОЧКА ВХОДА	I 0H	I 8H	I 10H	I 18H	I 20H	I 28H	I 30H	I 38H I

ПО АДРЕСАМ.УКАЗАННЫМ В ТАБЛИЦЕ ОБЫЧНО РАСПОЛАГАЕТСЯЁ JMP NNNN, ГДЕ NNNN -АДРЕС ПОДПРОГРАММЫ ОБРАБОТКИ ДАННОГО ПРЕРЫВАНИЯ.

ЭТО ТАК НАЗЫВАЕМЫЕ ПРОГРАММНЫЕ ПРЕРЫВАНИЯ– КОТОРЫЕ ГЕНЕРИРУЮТСЯ САМИМ ПРОГРАММИСТОМ.ОНИ МАЛО ЧЕМ ОТЛИЧАЮТСЯ ОТ ВЫЗОВА ПОДПРОГРАММЫ КОМАНДОЙ CALL ADR. НО– ВО-ПЕРВЫХ– ЗАНИМАЮТ ВСЕГО ОДИН БАЙТ– А– ВО-ВТОРЫХ УДОБНЕЕ.

ПРИМЕЧАТЕЛЬНЕЕ ВСЕГО ПРЕРЫВАНИЯ ПО 38Н-МУ АДРЕСУ– ТАК КАК ОНО МОЖЕТ БЫТЬ И ПРОГРАММНЫМ И АППАРАТНЫМ.ОНО РАЗРЕШАЕТСЯ КОМАНДОЙ EI, А ЗАПРЕЩАЕТСЯ КОМАНДОЙ DI. КОМАНДА HLT ОСТАНАВЛИВАЕТ РАБОТУ ПРОЦЕССОРА И ЖДЕТ–ПОКА НЕ ПРОИЗОЙДЕТ ЭТО ПРЕРЫВАНИЕ.ДАННЫЕ ПРЕРЫВАНИЯ ГЕНЕРИРУЮТСЯ ЧЕРЕЗ КАЖДЫЕ 20 МС ВО ВРЕМЯ РЕГЕНЕРАЦИИ ЭКРАНА. ЭТО ОЧЕНЬ УДОБНО ДЛЯ КОНТРОЛЯ ВРЕМЕНИ– Т.К. КАЖДЫЕ 50 ОБРАЩЕНИЙ ПО 38Н АДРЕСУ СООТВЕТСТВУЮТ 1 СЕКУНДЕ.ВО ВРЕМЯ ПРЕРЫВАНИЙ СКАНИРУЮТ КЛАВИАТУРУ–ПЕРЕПРОГРАММИРУЮТ ТАБЛИЦУ ЦВЕТОВ– ОСУЩЕСТВЛЯЮТ ВЫВОД МУЗЫКИ И Т.Д.ПРЕРЫВАНИЕ ПО 38-МУ АДРЕСУ ЗАДЕЙСТВОВАНО В БОЛЬШИНСТВЕ ПРОГРАММ.В ДАЛЬНЕЙШЕМ РЕЧЬ ИДЕТ ИМЕННО О НЕМ.

ТЕПЕРИ О ПОРТАХ. ПОРТЫ РАЗБИТЫ НА ЧЕТВЕРКИ– В КОТОРЫХ ПЕРВЫЙ ПОРТ ЯВЛЯЕТСЯ ПОРТОМ УПРАВЛЯЮЩЕГО СЛОВАЁ0Н,Н,2Н, 3Н, 4Н ... ПОРТЫ 8Н-0ВН -ЭТО ПОРТЫ ТАЙМЕРА–ЧЕРЕЗ НИХ ОРГАНИЗУЕТСЯ ВЫВОД МУЗЫКИ. ПОРТЫ 4Н-7Н - ПОРТЫ ПАРАЛЛЕЛЬНОГО ИНТЕРФЕЙЦА–СЛУЖАТ ДЛЯ ПОДКЛЮЧЕНИЯ ПРИНТЕРА.ПОРТЫ 0Н-3Н-ДЛЯ ПРОГРАММНОГО ДОСТУПА К МГ–КЛАВИАТУРЕ–СДВИГОВЫМ РЕГИСТРАМ–СЧЕТЧИКУ АДРЕСА ТЕКУЩЕЙ СТРОКИ ИЗОБРАЖЕНИЯ–ДИНАМИЧЕСКОЙ ГОЛОВКЕ И Т.Д.

ОПИШЕМ ПОДРОБНЕЕ ЧАСТО ИСПОЛЬЗУЕМЫЕ ПОРТЫ.

УСТАНОВКА ТАБЛИЦЫ ЦВЕТОВ

ПРОИЗВОДИТСЯ ПОРТАМИ 02Н И 0СН ВО ВРЕМЯ ПРЕРЫВАНИЯ.ПРИ ЭТОМ В ПОРТ 00Н ЗАПИСЫВАЕТСЯ УПРАВЛЯЮЩЕЕ СЛОВО 88Н.КОДИРОВКА ПРОИЗВОДИТСЯ ЗАПИСЬЮ В ПОРТ 02Н МАТЕМАТИЧЕСКОГО ЦВЕТА–А В ПОРТ 0СН -СООТВЕТСТВУЮЩЕГО ФИЗИЧЕЧКОГО.

УСТАНОВКА ЦВЕТА БОРДЮРА–ПЕРЕКЛЮЧЕНИЕ РЕЖИМОВ 512*256 И 256*256 ТОЧЕК

ВСЕ ВЫШЕПЕРЕЧИСЛЕННОЕ ПРОИЗВОДИТСЯ ПОРТОМ 02Н (УПРАВЛЯЮЩЕЕ СЛОВО В 00Н -М ПОРТУ -88Н) МЕЖДУ ПРЕРЫВАНИЯМИ.ПРИ ЭТОМЁ

```
 7 6 5 4 3 2 1 0
       I I     I
       I -------
       I    I
       I    -------МАТЕМАТИЧЕСКИЙ ЦВЕТ БОРДЮРА
       ------------0- 256*256;1- 512*256
```

СКРОЛЛИНГ ЭКРАНА

ПРОИЗВОДИТСЯ ПОРТОМ 03Н МЕЖДУ ПРЕРЫВАНИЯМИ (УПРАВЛЯЮЩЕЕ СЛОВО В 00Н-88Н).ПРИ ЭТОМ В НЕМ ДОЛЖЕН НАХОДИТЬСЯ НОМЕР ГОРИЗОНТАЛЬНОЙ СТРОКИ–ОТОБРАЖАЕМОЙ В САМОМ ВЕРХУ ЭКРАНА. В ОБЫЧНОМ СОСТОЯНИИ ТАМ НАХОДИТСЯ 0FFH (255-Я СТРОКА ВЕРХУ).

ПОРТ 01Н

7-КЛАВИША РУС/ЛАТ
6-КЛАВИША УС
5-КЛАВИША СС
4-ВХОД С МГ
3-СВЕТОДИОД РУС/ЛАТ
2
1-РЕЛЕ МГ
0-ВЫХОД НА МГ И ДИНАМИЧЕСКУЮ ГОЛОВКУ

ОПРОС КЛАВИАТУРЫ

ПРОИЗВОДИТСЯ ПОРТАМИ 02Н И 03Н ВО ВРЕМЯ ПРЕРЫВАНИЙ,В ПОРТУ 00Н-8АН).

7	6	5	4	3	2	1	0	ПОРТ	03Н:
80Н	40Н	20Н	10Н	8Н	4Н	2Н	1Н	ПОРТ	02Н:
Ь	П	Х	Ю	8	0	СТРЕЛ	ТАБ	1Н	0
Х	Р	Н	@	(		ВЛ-ВВ			
Ы	Я	И	А	9	1	СТР	ПС	2Н	1
Y	Q	I	A	)	!				
З	Р	Й	Б		2	АР2	ВК	4Н	2
Z	R	J	B	*	"				
Ш	С	К	Ц	;	3	F1	ЗБ	8Н	3
{	S	K	C	+	#				
Э	Т	Л	Д	,	4	F2	СТРЕЛ	10Н:	4
I	T	L	D	<	$		ВЛЕВО:		
Щ	У	М	Е	-	5	F3	СТРЕЛ	20Н:	5
}	U	M	E	=	%		ВВЕРХ:		
Ч	Ж	Н	Ф	.	6	F4	СТРЕЛ	40Н:	6
^	V	N	F	>	&		ВПРАВ:		
ПРОБ.	В	О	Г	/	7	F5	СТРЕЛ	80Н:	7
	W	O	G	?	'		ВНИЗ		

КЛАВИАТУРА ПРЕДСТАВЛЯЕТ СОБОЙ МАТРИЦУ 8*8.СТОЛБИКИ В МАТРИЦЕ СООТВЕТСТВУЮТ БИТАМ ПОРТА 03Н, СТРОЧКИ -БИТАМ ПОРТА 02Н.ДЛЯ ОПРЕДЕЛЕНИЯ- НАЖАТА ЛИ КЛАВИША-ВЫ ДОЛЖНЫ ПОДАТЬ СИГНАЛ НА СООТВЕТСТВУЮЩИЙ СТОЛБИК ЧЕРЕЗ ПОРТ 03Н,А ЗАТЕМ СОСЧИТАТЬ ПОРТ 02Н И ОПРЕДЕЛИТЬ-КАКАЯ СТРОЧКА БЫЛА ВОЗБУЖДЕНА.ПРИ ЗТОМ ЧИСЛО-КОТОРОЕ ВЫ ПОДАЕТЕ+СЧИТЫВАЕТЕ РАССМАТРИВАЕТСЯ ПОБИТНО. ЕСЛИ БИТ УСТАНОВЛЕН В 0,ТО СООТВЕТСТВУЮЩИЙ+АЯ СТОЛБИК+СТРОЧКА ВОЗБУЖДАЕТСЯ+ВОЗБУЖДЕНА.НАПРИМЕР-ВАМ НЦЖНО ОПРЕДЕЛИТЬ НАЖАТА ЛИ КЛАВИША "ПРОБЕЛ".ВЫ ЗАГРУЖАЕТЕ В ПОРТ 03Н ЧИСЛО 7FH=01111111B.ВОЗБУЖДАЯ ЛЕВЫЙ КРАЙНИЙ СТОЛБЕЦ.ЗАТЕМ СЧИТЫВАЕТЕ ИЗ ПОРТА 02Н ЧИСЛО-И ЕСЛИ СТАРШИЙ БИТ ЭТОГО ЧИСЛА УСТАНОВЛЕН В 0,ТО КЛАВИША "ПРОБЕЛ" НАЖАТА- ЕСЛИ НЕТ+ТО НЕ НАЖАТА.

ДЛИТЕЛЬНОСТЬ ПОЛУПЕРИОДОВ ДЛЯ ЧЕТНОГО ЧИСЛА N РАВНА TCLK*N/2,A ДЛЯ НЕЧЕТНОГО N ДЛИТЕЛЬНОСТЬ ПОЛОЖИТЕЛЬНОГО ПОЛУПЕРИОДА РАВНА TCLK*N/2, A ОТРИЦАТЕЛЬНОГО - TCLK*(N-1)/2.

РЕЖИМ 4 (ПРОГРАММНОГО ФОРМИРОВАНИЯ ОДИНОЧНОГО СТРОБА) НА ВЫХОДЕ КАНАЛА ФОРМИРУЕТСЯ ИМПУЛЬС ОТРИЦАТЕЛЬНОЙ ПОЛЯРНОСТИ ДЛИТЕЛЬНОСТЬ T=TCLK ПОСЛЕ ОТСЧЕТА ЧИСЛА- ЗАГРУЖЕННОГО В СЧЕТЧИК.

РЕЖИМ 5 (АППАРАТНОГО ФОРМИРОВАНИЯ ОДИНОЧНОГО СТРОБА) НА ВЫВОДЕ КАНАЛА ФОРМИРУЕТСЯ ИМПУЛЬС ОТРИЦАТЕЛЬНОЙ ПОЛЯРНОСТИ ДЛИТЕЛЬНОСТЬЮ T=TCLK ПОСЛЕ ОТСЧЕТА ЧИСЛА- ЗАГРУЖЕННОГО В СЧЕТЧИК.

ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ В ДРАЙВЕРАХ УСТРОЙСТВ

```
0038 JMP 0BD2 ;ТОЧКА ВХОДА
0BD2 PUSH PSW ;ПОМЕСТИМ РЕГИСТРЫ
0BD3 PUSH B   ;В СТЕК
0BD4 PUSH D
0BD5 PUSH H
0BD6 LXI D,100F;ЗАДАЕМ СЧЕТЧИК
               ;ЦВЕТОВ И НАЧ.ЦВЕТ
0BD9 LXI H,0BD1;АДРЕС КОНЦА
               ;ТАБЛИЦЫ ЦВЕТОВ
0BDC MOV A,E   ;ЗАДАЕМ МАТЕМ.ЦВЕТ
0BDD OUT 2
0BDF MOV A,M   ;ЗАДАЕМ ФИЗ.ЦВЕТ
0BE0 OUT 0C
0BE2 DCR C
0BE3 OUT 0C
0BE5 DCX H     ;СМЕЩЕНИЕ ПО ТАБЛ.
0BE6 OUT 0C
0BE8 DCR E     ;СМЕЩ. ПО МАТЕМ.ЦВ.
0BE9 DCR D     ;ВСЕ 16 ЦВЕТОВ?
0BEA OUT 0C
0BEC JNZ 0BDC  ;ЕСЛИ НЕТ-ТО СНАЧАЛА
0BEF LHLD 1A88 ;УВЕЛИЧИВАЕМ ТАЙМЕР
0BF2 INX H     ;НА ЕДИНИЦУ
0BF3 SHLD 1A88
0C29 LI 0800     ;ЗАДЕЙСТВОВАННЫЙ
0C2C MOV A,B     ;СТОЛБЕЦ
0C2D OUT 03
0C2F IN 02
0C31 CPI FF
0C33 JNZ 0C4E    ;ПЕРЕХОД-ЕСЛИ
0C36 MOV A,E     ;НАШЛИ
0C37 ADD D
0C38 MOV E,A
0C39 MOV A,B
0C3A RLC
0C3B MOV B,A
0BF6 MVI A,8A ;ЗАЛАЕМ УПР.СЛОВО
0BF8 OUT 00   ;ДЛЯ СКАН-НИЯ КЛАВ.
0BFA LDA 0D59
0BFD OUT 01
0BFF IN 01      ;НАЖАТА РУС/ЛАТ?
0C01 ANI 80
0C03 LXI H,1C1A ;ЯЧЕЙКА-ПРИЗНАК
0C06 JNZ 0C1A ;ПЕРЕХОД-ЕСЛИ НЕТ
0C09 MOV A,M
0C0A ORA A
0C0B JZ 0C1C
0C0E MVI M,00
0C10 LXI H,0D56
0C13 MOV A,M    ;УСТАНАВЛИВАЕМ
0C14 XRI 08     ;ПРИЗНАК РУС/ЛАТ
0C16 MOV A,M
0C17 JMP 0C1C
0C1A MVI M,FF   ;ХОТЬ ОДНА
0C1C XRA A    ;КЛАВИША НАЖАТА ?
0C1D OUT 03
0C1F IN 02
+C21 CPI FF
0C23 JZ 0C40 ;НЕТ.УХОДИМ НА 0C40
0C26 LXI B,E08  ;ИЩЕМ +-Й
0C9A RET     ;ПЕРЕХОД НА 0C91
0C9B ANI 20
0C9D MOV A,E
0C9E JNZ 0C98
0CA1 JMP 0C96
0CA4 MOV A,B
0CA5 ANI 40
0CA7 JNZ 0CAE
0CAA MOV A,E
0CAB ANI 1F     ;КОДЫ 00-1F
0CAD RET
0CAE MOV A,B
```

```
0C3C DCR C                          0CAF ANI 20
0C3D JNZ 0C2D                       0CB1 MOV B,A    ;CC -?
0C40 CALL 0D14   ;ОБРАЩАЕМСЯ К П+П  0CB2 LDA 0D56   ;РУС/ЛАТ -?
            ўВОССТВНОВЛЕНИЯ ПОРТОВ  +CB5 ANI 08
0C43 XRA A    ;ПРИЗНАК–ЧТО КЛАВИШИ  0CB7 ORA B
0C44 STA 1C16    ;НЕ НАЖАТЫ         0CB8 MOV A,E
0C47 CMA                            0CB9 JPO 0CBE
0C48 STA 1C19                       0CBC ADI 20  КОДЫ 40-5E,60-7E
0C4B JMP 0D35    ;ПЕРЕХОД НА МУЗЫКУ 0CC0 RET     ;ПЕРЕХОД НА 0CD1
0C4E RAR     ўЦИКЛ ПРЕДВ.ОБРАБОТКИ  0CD1 LXI H,0D35 ;АДРЕС ПЕРЕХОДА В
0C4F JNZ 0C56   ;КИ РЕЗУЛЬТАТОВ СКА- 0CD4 PUSH H  ;СТЕК–В А-КОД КЛАВИШИ
0C52 INR E       ;НИРОВАНИЯ         0CD6 LDA 1C19
0C53 JMP 0C4E                       0CD9 CPI FF
0C56 CALL 0D14  ;П+П ВОССТАНОВЛЕНИЯ 0CD8 JZ 0CF6
                 ;ПОРТОВ            0CDE CMP B
0C59 LXI H,0CD1 ;ПОМЕСТИМ АДРЕС     0CDF RNZ
                 ;ВОЗВРАТА В СТЕК   0CE0 LXI H,1C16
0C5C PUSH H                         0CE3 MOV A,M
0C5D IN 01       ;НАЖАТЫ СС ИЛИ УС? 0CE4 CPI 32
0C5F ANI 60                         0CE6 JZ 0CEB
0C61 JNZ 0C66                       0CE9 INR M
0C64 MVI A,60                       0CEA RET     ;ПЕРЕХОД НА 0D35
0C66 MOV B,A                        0CEB LXI H,1C15
0C67 MOV A,E                        0CEE MOV A,M
0C68 CPI 3F      ;ПРОБЕЛ –          0CEF CPI 0A
0C6A MVT A,20                       0CF1 JNZ 0CE9
0C6C RZ       ;А ДА - ПЕРЕХОД НА 0CD1  0CF4 MVI M,0
0C6D MOV A,E                        0CF6 MOV A,B ;ПОМЕЩАЕМ КОД КЛАВИШИ
0C6E CPI 10                         0CF7 STA 1C19
0C70 JNZ 0C85                       0CFA LXI H,1C1B ;СЧЕТЧИК ВВЕДЕННЫХ
0C73 LXI H,0CC1 ўКЛАВИШИ С КОДАМИ:  0CFD MOV A,M    ;КОДОВ (00-0F)
0CC1 09,0A,0D,7F,08,19,18,1A,0C,1F, 0CFE CPI 10
1B,00,01,02,03,04                   0D00 RNC ;ПЕРЕХОД НА 0D35 ЕСЛИ
0C76 MVI D,0                                 ;БУФЕР ЗАПОЛНЕН
0C78 DAD D                          0D01 INR M   ;УВЕЛИЧИВАЕМ СЧЕТЧИК
0C79 MOV A,M                        0D02 LDA 1C1C ;МЕСТО РАСПОЛОЖЕНИЯ
0C7A CPI 7F      ;ЗАБОЙ –               ;ОЧЕРЕДНОГО ВВОДИМОГО СИМВОЛА
0C7C RNZ      ;НЕТ. ПЕРЕХОД НА 0CD1 0D05 MOV E,A
0C7D MOV A,B     ;CC-?              0D06 MVI D,0
0C7E ANI 20                         0D08 LXI H,1C1E ;НАЧАЛО БУФЕРА
0C80 MVI A,5F    ;ДА - СИМВОЛ " "   0D0B DAD D
0C82 RZ                             0D0C INR A
0C83 MOV A,M     ;НЕТ - "ЗБ"        0D0D ANI 0F
0C84 RET         ;ПЕРЕХОД НА 0CD1   0D0F STA 1C1C
0C85 CPI 20                         0D12 MOV M,B
0C87 JNZ 0CA4                       0D13 RET         ;ПЕРЕХОД НА 0D35
0C8A CPI 1C                         0D14 LDA 0D56   ;П+П ВОССТАНОВЛЕНИЯ
0C8C MOV A,B                                        ;ПОРТОВ
0C8D JNZ 0C9B                       0D17 ANI 09     ;РУС/ЛАТ+БИТ НА МГ
0C90 ANI 20                         0D19 MOV H,A
0C92 MOV A,E     ;CC-?              0D1A LDA 0D59   ;РЕЛЕ МГ
0C93 JZ 0C98                        0D1D ANI 02
0C96 ADI 10 ;СИМВОЛЫ С КОДАМИ 20-3F 0D1F ADD H
0C98 ADI 10                         0D20 MOV H,A
0D21 MVI A,88    ;ВОССТ. УПР.СЛОВО  0E48 PUSH H
0D23 OUT 00                         0E49 PUSH H
0D25 MOV A,H                        0E4A MOV A,C
0D26 OUT 01                         0E4B LXI D,0080 ;ВЫЧ.НАЧАЛО НОТ
0D28 LDA 1A8A    ;СКРОЛЛИНГ         0E4E LXI H,1A0F
```

```
0D2B OUT 03                          0E51 DAD D
0D2D LDA 0D57    ;ЦВЕТ БОРДЮРА       0E52 DCR A
0D30 ANI 0F                          0E53 JP 0E51
0D32 OUT 02                          0E56 ROR D
0D34 RET                             0E57 LDAX D  ;ПОРЯД.НОМЕР НОТЫ
0D35 LDA 1A8B    ;ПРИЗНАК ЗВУЧАНИЯ   0E58 MOV E,A
                 ;КАНАЛОВ            0E59 MVI D,00
0D38 RRC                             0E5B DAD D
0D39 RRC                             0E5C MOV A,M    ;ИЗВЛЕКАЕМ НОТУ
0D3A RRC                             0E5D ORA A
0D3B LXI B,0002                      0E5E XCHG     ;B DE - АДРЕС НОТЫ
0D3E RLC         ;КАНАЛ ЗВУЧИТ ?     0E5F POP H    ;B HL - АДРЕС СЧЕТЧИКА
0D3F PUSH PSW                        0E60 JZ 0E1A  ;ЕСЛИ 0,ТО ВЫКЛ.КАНАЛ
0D40 CC 0E0A     ;ДА                 0E63 CPI 4C   ;"L" - ?
0D43 POP PSW                         0E65 JNZ 0E85
0D44 DCR C                           0E68 INR M    ;СЧЕТЧИК+1
0D45 JP 0D3E                         0E69 INX D ;СМЕЩ. НА СЛЕД.НОТУ

+D48 POP H       ;ВОССТ. РЕГИСТРЫ    0E6A LDAX D
0D49 POP D                           0E6B CPI 30     ;"0"-?
0D4A POP B                           0E6D JZ 0E7B    ;ЦИФРЫ -?
0D4B POP PSW                         0E70 CPI 3A     ;":" -?
0D4C EI          ;РАЗР.ПРЕРЫВАНИЙ    0E72 JNZ 0E7B
0D49 RET         ;ВОЗВРАТ К П+П      0E75 CALL 0F6E ;ДВУХЗНАЧН.ЧИСЛО?
0E0A LXI H,0DD3  ;ДЛИТ.ТЕК.НОТЫ      0E78 JMP 0E7D
0E0D DAD B                           0E7B MVI A,04 ;L=4 ПО УМОЛЧИНИЮ
0E0E DCR M                           0E7D PUSH H
0E0F RNZ         ;НОТА ЕЩЕ ИГРАЕТ    0E7E LXI H,0DD0 ;ОБЩАЯ ДЛИТ.НОТ
0E10 LXI H,1A8C  ;СЧЕТЧИК КОЛ.НОТ    0E81 DAD B
0E13 DAD B                           0E82 MOV M,A
0E14 MOV A,M                         0E83 POP H
0E15 CPI 80                          0E84 LDAX D
0E17 JZ 0E48     ;ЕСЛИ МЕНЬШЕ 128,TO 0E85 CPI 4      ;"0" -?
ИДЕМ ДАЛЬШЕ- ИНАЧЕ ВЫКЛ.КАНАЛА      0E87 JNZ 0EA3
0E1A MVI M,00                        0E8A INR M      ;СЧЕТЧИК+1
0E1C LXI H,0DD6  ;ОКТАВА             0E8B INX D
0E1F DAD B                           0E8C LDAX D
0E20 MVT M,05 ;4-Я ОКТАВА ПО УМОЛЧ.  0E8D CPI 30     ;"0" -?
0E22 LXI H,0DD0  ;ДЛИТЕЛЬНОСТЬ       0E8F JC 0EA3    ;ЦИФРЫ?
0E25 DAD B                           0E92 CPI 39     ;"9" -?
0E26 MVT M,04    ;4 ПО УМОЛЧАНИЮ     0E94 JNZ 0EA3
0E28 PUSH B                          0E97 PUSH H
0E29 MVI A,7F                        0E98 LXI H,0DD6 ;ОКТАВА
0E2B RLC                             0E9B DAD B
0E2C DCR D                           0E9C     2F
0E2D JP 0E2B                         0E9E MOV M,A
0E30 LXI H,1A8B  ;СТИРАЕМ ПРИЗНАК    0E9F POP H
0E33 ANA M       ;ЗВУЧАНИЯ КАНАЛА    0EA0 INR M      СЧЕТЧИК +1
0E34 MOV M,A                         0EA1 INX D
0E35 POP B                           0EA2 LDAX D
0E36 LXI H,0DD3  ;ДЛИТ.ТЕК.НОТЫ=1    0EA3 INR M
0E39 DAD B                           0EA4 INX D
0E3A MVI M,1                         0EA5 PUSH H
0E3C MOV A,C                         0EA6 PUSH D
0E3D INR A                           0EA7 LXI H,0EE5 ;АДРЕС
0E3E CMA                             0EAA PUSH H     ;ПЕРЕХОДА
0E3F RRC                             0EAB LXI H,0DDB ;АДРЕС КОНКР.ЗНАЧ.
                                         ;НОТ
0E40 RRC                             ;BD,D2,2B,B3,1C,A9,9E,9F,A9,
```

```
                                    ;96,34,8E,39
0E41 ANI C0      ;ВЫКЛ.КАНАЛА       ;86 BO 7E,94,77,DE,70,88,
                                    ;6A,8E,64,E9,5E,
0E43 ORI 30                         ;95,59
0E45 OUT 08                    0EAE LXI D,0004 ;СМЕЩЕНИЕ
0E47 RET                       0EB1 CPI 43     ;"C" -?
0EB3 RZ      ;ПЕРЕХОД НА 0EE5  0F0A DAD B                0F5B MOV A,C
0EB4 DAD D                     0F0B MOV A,M                0F5C INR A
0EB5 CPI 44     ;"D" -?        0F0C POP H                  0F5D CMA
0EB7 RZ                        0F0D PUSH B                 0F5E RRC
0EB8 DAD D                     0F0E MOV C,A                0F5F RRC
0EB9 CPI 45     ;"E" -?        0F0F DCR C                  0F60 ANI C0
0EBB RZ                        0F10 JZ 0F1D                0F62 ORI 36
0EBC INX H                     0F13 MOV A,H                0F64 OUT 08
0EBD INX H                     0F14 ORA A                  0F66 MOV A,L
0EBE CPI 46     ;"F" -?        0F15 RAR                    0F67 OUT 09
0EC0 RZ                        0F16 MOV H,A                0F69 MOV A,H
0EC1 DAD D                     0F17 MOV A,L                0F6A OUT 09
0EC2 CPI 47     ;"G" -?        0F18 RAR                    0F6C POP PSW
0EC4 RZ                        0F19 MOV L,A                0F6D RET
0EC5 DAD D                     0F1A JMP 0F0F               0F6E SUI 30
0EC6 CPI 41     ;"A" -?        0F1D POP B                  0F70 INR M
0EC8 RZ                        0F1E CALL 0F51              0F71 INX D
0EC9 DAD D                     0F21 POP H                  0F72 PUSH B
0ECA CPI42      ;"B" -?        0F22 LDAX D                 0F73 MOV C,A
0ECC RZ                        0F23 CPI 30      ;"0" -?    0F74 LDAX D
0ECD POP H                     0F25 JC 0F33                0F75 CPI 30
0ECE POP D                     0F28 CPI A       ;":" -?    0F77 JC 0F8B
0ECF POP H                     0F2A JNC 0F33               0F7A CPI 3A
0ED0 CPI 50     ;"P" -?        0F2D CALL 0F6E   ;ДВУХЗН.   0F7C JNC 0F8B
0ED2 JZ 0EDF                   0F30 JMP 0F3A   ;ЧИСЛО -?   0F7F SUI 30
0ED5 CPI 52     ;"R" -?        0F33 PUSH H                 0F81 MOV B,A
0ED7 JNZ 0E36   ;ВЫКЛЮЧЕНИЕ    0F34 LXI H,0DD0    ;ОБЩ.    0F82 MOV A,C
0EDA MVI M,00   ;КАНАЛА        0F37 DAD B    ;ДЛИТЕЛЬН.    0F83 ADD A
0EDC JMP 0E36   ;ПОВТОРЕНИЕ    0F38 MOV A,M                0F84 ADD A
0EDF CALL 0E3C  ;ВЫКЛ.КАНАЛА   0F39 POP H                  0F85 ADD C
0EEF JMP 0F22                  0F3A PUSH H                 0F86 ADD A
0EE5 POP D                     0F3B LXI H,0DD3    ;ДЛИТ.   0F87 ADD B
0EE6 LDAX D                    0F3E DAD B    ;ДАНН.НОТЫ    0F88 MOV C,A
0EE7 CPI 23     ;"#" -?        0F3F MOV M,A                0F89 INX D
0EE9 INX H                     0F40 LDAX D                 0F8A INR M
0EEA INX H                     0F41 CPI 2E   ;"." -?       0F8B MOV A,C
0EEB JZ 0EFC                   0F43 MOV A,M                0F8E RLC
0EEE CPI 2B     ;"+" -?        0F44 JNZ 0F4F               0F8F MOV B,A
0EF0 JZ 0EFC                   0F47 ADD A    ;УВЕЛ.ДЛИТ.   0F90 MVI A,80
0EF3 CPI 2D     ;"-" -?        0F48 ADD M    ;В 1.5 РАЗА   0F92 RLC
0EF5 DCX H                     0F49 ORA A                  0F93 MOV C,A
0EF6 DCX H                     0F4A RAR                    0F94 MOV A,B
0EF7 JNZ 0F00                  0F4B MOV M,A                0F95 RLC
0EFA DCX H                     0F4C POP H                  0F96 MOV B,A
0EFB DCX H                     FF4D INR M                  0F97 MOV A,C
0EFC XTHL                      0F4E RET                    0F98 JNC 0F92
0EFD INR M                     0F4F POP H                  0F9B POP B
0EFE XTHL                      0F50 RET                    0F9C RET
0EFF INX D                     0F51 PUSH PSW
0F00 SHLD 0F04  ;ЗАП. КОНКР.   0F52 MVI A,59
0F03 LHLD 0000  ;ЗНАЧ.         0F54 ADD C
0F06 PUSH H                    0F55 STA 0F68
0F07 LXI H,0DD6 ;ОКТАВА        0F58 STA 0F6B
```

СТАНДАРТНЫЕ ПОДПРОГРАММЫ МОНИТОРА

	ПОДПРОГРАММЫ МОНИТОРА	АДРЕС	ВХОДНЫЕ ПАРАМЕТРЫ	ВЫХОДНЫЕ ПАРАМЕТРЫ
1	ВЫВОД СИМВОЛА С КЛАВИАТУРЫ	7803H F803H	НЕТ	A - ВВЕДЕННЫЙ КОД
2	ВВОД БАЙТА С МАГНИТОФОНА	7806H F806H	A=FF - С ПОИСКОМ СИНХРОБАЙТА A=08 - БЕЗ ПОИСКА СИНХРОБАЙТА	A - ВВЕДЕННЫЙ БАЙТ
3	ВЫВОД СИМВОЛА НА ЭКРАН	7809H F809H	C - ВЫВОДИМЫЙ СИМВОЛ	НЕТ

БАЙТ-5 ИНФОРМАЦИОННО-РЕКЛАМНЫЙ ВЫПУСК
ДЛЯ ПОЛЬЗОВАТЕЛЕЙ БПЭВМ "ВЕКТОР-06Ц"
ОКТЯБРЬ 1991г.

КАК НАПИСАТЬ ИГРОВУЮ ПРОГРАММУ НА БЕЙСИКЕ

ВЫ ПРИОБРЕЛИ КОМПЬЮТЕР–ОСВОИЛИ ЕГО И–ГЛЯДЯ НА ИНТЕРЕСНЫЕ–УВЛЕКАТЕЛЬНЫЕ ИГРЫ–ВАМ ЗАХОТЕЛОСЬ СОЗДАТЬ НЕЧТО ПОДОБНОЕ САМОМУ.ВЫ ПОКА НЕ ВЛАДЕЕТЕ АССЕМБЛЕРОМ И КОДОВЫЕ ПРОГРАММЫ ВАМ НЕ ПОД СИЛУ– НО НЕСЛОЖНУЮ ИГРУ НА ДЕМОКРАТИЧНОМ И ДРУЖЕСКОМ БЕЙСИКЕ ВЫ НАПИСАТЬ СМОЖЕТЕ.

АВТОР НЕ СТАВИТ ЦЕЛЬЮ ДАТЬ ИСЧЕРПЫВАЮЩИЕ РЕКОМЕНДАЦИИ И ПОДРОБНЫЕ ПРИМЕРЫ.В СТАТЬЕ ДАЮТСЯ ОБЩИЕ НАПРАВЛЕНИЯ И НЕКОТОРЫЕ ПРИЕМЫ ПРОГРАММИРОВАНИЯ. С ЧЕГО НАЧАТЬ –

СНАЧАЛА ВАМ НУЖНО ПРИДУМАТЬ СЮЖЕТ ИГРЫ–ВЫБРАТЬ ГЛАВНОГО ГЕРОЯ И ТЕХ–С КЕМ ОН БУДЕТ БОРОТЬСЯ.ОПРЕДЕЛИТЬ ЦЕЛЬ–КОТОРУЮ НУЖНО ДОСТИЧЬ– И ТРУДНОСТИ–КОТОРЫЕ НУЖНО ПРЕОДОЛЕТЬ.ИДЕИ ВЫ МОЖЕТЕ БРАТЬ ИЗ КНИГ–СКАЗОК–ФАНТАСТИКИ ИЛИ ИЗ ЖИЗНИ. ГЛАВНЫЙ ГЕРОЙ МОЖЕТ БЫТЬ ЧЕЛОВЕЧКОМ–ЖИВОТНЫМ–РОБОТОМ И Т.Д. НЕКОТРЫЕ ПЕРСОНАЖИ МОГУТ ЕМУ ПОМОГАТЬ (ДРУЗЬЯ) ДРУГИЕ - МЕШАТЬ (ВРАГИ). И ВРАГИ И ДРУЗЬЯ МОГУТ БЫТЬ ПАССИВНЫМИ (СКАЖЕМ ОЗЕРО) ИЛИ АКТИВНЫМИ (САМОЛЕТ ПРОТИВНИКА). ЦЕЛЬ ИГРЫ МОЖЕТ ЗАКЛЮЧАТЬСЯ В НАХОЖДЕНИИ ВЫХОДА ИЗ ЛАБИРИНТА–В УНИЧТОЖЕНИИ ВРАГОВ И Т.Д.

ПОСЛЕ СОСТАВЛЕНИЯ СЦЕНАРИЯ ВАМ НУЖНО ПРОДУМАТЬ АЛГОРИТМ ПЕРЕДВИЖЕНИЯ ПЕРСОНАЖЕЙ.НЕКОТРЫЕ МОГУТ ДВИГАТЬСЯ ВО ВСЕХ НАПРАВЛЕНИЯХ–ДРУГИЕ ВВЕРХ-ВНИЗ ИЛИ ВПРАВО-ВЛЕВО–КТО-ТО МОЖЕТ СТРЕЛЯТЬ–КТО-ТО-НЕТ И Т.Д.ЗАТЕМ ВЫ МОЖЕТЕ ПРИСТУПИТЬ К СОЗДАНИЮ СПРАЙТОВ И КАРТИНОК.ПОД СПРАЙТОМ ПОНИМАЕТСЯ ДВИЖУЩИЙСЯ ОБЬЕКТ–ПОД ФОНОВОЙ КАРТИНКОЙ-ДЕКОРАЦИИ(ГОРЫ–НЕБО–МОРЕ–ЭТАЖИ ЛАБИРИНТА).ДЛЯ СОЗДАНИЯ СПРАЙТОВ И ДЕКОРАЦИЙ ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ЛЮБЫЕ ГРАФИЧЕСКИЕ РЕДАКТОРЫ ИЛИ ЖЕ РИСОВАТЬ ИХ САМИ ИЗ ГРАФИЧЕСИХ ПРИМИТИВОВ–ИСПОЛЬЗУЯ КОМАНДЫ PLOT,LINE, CIRCLE,COLOR, PAINT.ЕСЛИ ВЫ ОРИЕНТИРУЕТЕСЬ НА ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОВ PUT И GET,ТО ВАМ НУЖНО БУДЕТ ПОЛУЧИТЬ ПРОГРАММНЫЕ СТРОКИ НА БЕЙСИКЕ СОСТОЯЩИЕ ИЗ КОМАНД PLOT,LINE И Т.Д.В ВИДЕ ПОДПРОГРАММ.ЭТИ ПОДПРОГРАММЫ БУДУТ РИСОВАТЬ КАРТИНКИ–КОТОРЫЕ ЗАТЕМ БУДУТ ЗАПОМИНАТЬСЯ ОПЕРАТОРОМ GET.СПРАЙТ ДОЛЖЕН ИМЕТЬ НЕБОЛЬШИЕ РАЗМЕРЫ–ИНАЧЕ ОН БУДЕТ ДОЛГО ВЫВОДИТЬСЯ.ОПТИМАЛЬНО-НЕ БОЛЬШЕ ЧЕМ 16*16 ТОЧЕК.В ГОТОВОЙ ПРОГРАММЕ СНАЧАЛА БУДУТ РИСОВАТЬСЯ СПРАЙТЫ–ЗАТЕМ ЗАПОМИНАТЬСЯ ОПЕРАТОРОМ GET И В ДАЛЬНЕЙШЕМ ВЫВОДИТЬСЯ ОПЕРАТОРОМ PUT.ЧТОБЫ НА ЭКРАНЕ НЕ БЫЛ ВИДЕН ПРЦЕСС РИСОВАНИЯ СПРАЙТОВЮМОЖНО СНАЧАЛА ПЕРЕУСТАНОВИТЬ ТАБЛИЦУ ЦВЕТОВ:SCREEN 0,0,0...0,НАРИСОВАТЬ СПРАЙТЫ–ЗАПОМНИТЬ ИХ–СТЕРЕТЬ ЭКРАН И ВОССТАНОВИТЬ ТАБЛИЦУ ЦВЕТОВ.ВАЖНОЕ МЕСТО В СОЗДАНИИ ДИНАМИЧЕСКОЙ ИГРОВОЙ ПРОГРАММЫ ЗАНИМАЕТ ОРГАНИЗАЦИЯ ДВИЖЕНИЯ СТПРАЙТОВ. СПРАЙТ МОЖЕТ ПЕРЕМЕЩАТЬСЯ ДВУМЯ СПОСОБАМИЁ

1.ЕСЛИ БОЛЬШОЕ(+ ТОЧКИ И БОЛЕЕ)–ТО ДВИЖЕНИЕ ОРГАНИЗУЕТСЯ ПО СХЕМЕЁ СПРАЙТ ВЫВОДИТСЯ–СТИРАЕТСЯ ПЕРЕСЧИТЫВАЮТСЯ КООРДИНАТЫ–СНОВА ВЫВОДИТСЯ И Т.Д.ПРИ ЭТОМ СПОСОБЕ ДВИЖЕНИЕ ОРГАНИЗУЕТСЯ ПО БОЛОЙ ПЛОЩАДИ –НО КАРТИНКА БУДЕТ МИГАТЬ.

2.ЕСЛИ ПЕРЕМЕЩЕНИЕ МАЛОЕ(+.., ТОЧКИ)–ТО СПРАЙТ МОЖНО ЗАПОМНИТЬ С ПУСТОЙ КАЕМОЧКОЙ–ТОЛЩИНА КОТОРОЙ БУДЕТ ЗАВИСЕТЬ ОТ ВЕЛИЧИНЫ СМЕЩЕНИЯ. ТО ЕСТЬ ЗАПОМНЕННАЯ КАРТИНКА БУДЕТ БОЛЬШЕ–ЧЕМ ЕЕ ВИДИМАЯ ЧАСТЬ– ТЕПЕРЬ ПРИ СМЕЩЕНИИ СПРАЙТА ЕГО НЕ НУЖНО БУДЕТ ПРЕДВАРИТЕЛЬНО СТИРАТЬ–ТАК КАК ПРИ ВЫВОДЕ ЕГО В НОВОЕ МЕСТО НЕВИДИМАЯ И ВИДИМАЯ ЧАСТИ БУДУТ СТИРАТЬ ИЗОБРАЖЕНИЕ– И СХЕМА ПЕРЕМЕЩЕНИЙ БУДЕТ СЛЕДУЮЩЕЙЁВЫВОД СПРАЙТА–ПЕРЕСЧЕТ КООРДИНАТ–ВЫВОД СПРАЙТА И Т.Д.

ИЗОБРАЖЕНИЕ БУДЕТ ВЫВОДИТЬСЯ БЫСТРЕЕ И НЕ БУДЕТ МИГАТЬ.ОДНАКО ДЛЯ ПЕРЕМЕЩЕНИЯ НА ЗНАЧИТЕЛЬНЫЕ РАССТОЯНИЯ ПОНАДОБИТСЯ БОЛШЕ ВРЕМЕНИ.ПРИ ОРГАНИЗАЦИИ ДВИЖЕНИЯ СПРАЙТОВ ВОЗНИКАЕТ ПРОБЛЕМА СОХРАННОСТИ ДРУГИХ

КАРТИНОК.ЭТУ ПРОБЛЕМУ МОЖНО РЕШИТЬ ТРЕМЯ СПОСОБАМИ:
 1 СПОСОБ.ПЕРЕД ВЫВОДОМ СПРАЙТА ОПЕРАТОРОМ GET ВЫ ЗАПОМИНАЕТЕ ТОТ УЧАСТОК ИЗОБРАЖЕНИЯ-КУДА БУДЕТ ВЫВОДТЬСЯ СПРАЙТ.ЗАТЕМ ВЫВОДИТЕ СПРАЙТ-ЗАТЕМ ВОССТАНАВЛИВАЕТЕ ИЗОБРАЖЕНИЕ-ПЕРЕСЧИТЫВАЕТЕ КООРДИНАТЫ И ПОВТОРЯЕТЕ ЦИКЛ ЗАНОВО.НЕДОСТАТОК ЭТОГО СПОСОБА-НИЗКОЕ БЫСТРОДЕЙ-СТВИЕ.
 2 СПОСОБ.СПРАЙТ ВЫ ПЕРЕМЕЩАЕТЕ НЕ ВО ВСЕХ ПЛОСКОСТЯХ-А В ОДНОЙ ИЛИ В ДВУХ-И ПРИ ЕГО ПЕРЕМЕЩЕНИИ ЗАПРЕЩАЕТЕ ИЗМЕНЕНИЕ В ДРУГИХ ПЛОСКОС-ТЯХ.НАПРИМЕР-ВЫ НАРИСОВАЛИ СПРАЙТ +-М МАТЕМАТИЧЕСКИМ ЦВЕТОМ-ТО ЕСТЬ ОН БУДЕТ ПЕРЕМЕЩАТЬСЯ ПО +-ОЙ ВИДЕОПЛОСКОСТИ.ДЛЯ ЕГО ПЕРЕМЕЩЕНИЯ ВЫ ДАЕТЕ КОМАНДУ:SCREEN 2,1:PUT X,Y,ADDR(A(0)),2:SCREEN 2,15.ДОСТОИН-СТВО-БОЛЕЕ ВЫСОКОЕ БЫСТРОДЕЙСТВИЕ(НЕ НУЖНО ПРЕДВАРИТЕЛЬНО ЗАПОМИНАТЬ УЧАСТОК ЭКРАНА-КРОМЕ ТОГО ОПЕРАТОР PUT ВЫВОДИТ КАРТИНКУ ПО ТОЧКАМ-А ПОДПРОГРАММА РИСОВАНИЯ ТОЧКИ РАБОТАЕТ БЫСТРЕЕ-ЕСЛИ ДЛЯ ИЗМЕНЕНИЯ РАЗРЕШЕНА ТОЛЬКО ОДНА ПЛОСКОСТЬ.НЕДОСТАТОК-МАЛОЕ КОЛ-ВО ЦВЕТОВ ИЗ КОТОРЫХ СОСТОИТ СПРАЙТ.
 3 СПОСОБ.САМЫЙ ПРОСТОЙ.МОЖНО ПЕРЕМЕЩАТЬ СПРАЙТЫ НА ЧИСТОМ ПОЛЕ БЕЗ ДЕКОРАЦИЙ-ТОГДА ПРОБЛЕМ С СОХРАННОСТЬЮ ИЗОБРАЖЕНИЯ НЕ ВОЗНИКАЕТ.
   КРОМЕ ЭТОГО НУЖНО ВЫБРАТЬ СПОСОБ ВЗАИМОДЕЙСТВИЯ ИГРОКА С ГЛАВНЫМ ПЕРСОНАЖЕМ.ЕСТЬ ДВА ВАРИАТНА:1).НАЖАЛ КНОПКУ-ПЕРЕДВИНУЛ КАРТИНКУ; 2)НАЖАЛ КНОПКУ-ИЗМЕНИЛ НАПРАВЛЕНИЕ ДВИЖЕНИЯ КАРТИНКИ(ТО ЕСТЬ КАР-ТИНКА ВСЕ ВРЕМЯ ДВИЖЕТСЯ).

БЛОК УПРАВЛЕНИЯ ДЛЯ ПЕРВОГО ВАРИАНТА:

```
100 V=ASC(INKEY$):IF V=255 THEN 130
110 IF V=8 THNE 150
120 ON V-23 GOTO 170, 190, 210
130 ...:REM П+П ПЕРЕДВИЖЕНИЯ ВРАГОВ
135 GOTO 100
140 REM П+П СДВИГА ВЛЕВО
150 ...:GOTO 130
160 REM П+П+ СДВИГА ВПРАВО
170 ...:GOTO 130
180 REM П+П СДВИГА ВВЕРХ
190 ...:GOTO 130
200 REM П+П СДВИГА ВНИЗ
210 ...:GOTO 130
```

БЛОК УПРАВЛЕНИЯ ДЛЯ ВТОРОГО ВАРИАНТАЁ

```
A) 100 V=ASC(INKEY$):IF V=255THENV=V1
   110 IF V=8 THEN 150
   120 ON V-23 GOTO 170,190,210
   130 V=V1: GOTO 110
   140 REM П+П СДВИГА ВЛЕВО
   150 ...:GOTO 220
   160 REM П+П СДВИГА ВПРАВО
   170 ...:GOTO 220
   180 REM П+П СДВИГА ВВЕРХ
   190 ...:GOTO 220
   200 REM П+П СДВИГА ВНИЗ
   210 ...:GOTO 220
   220 V1=V:...:REM П+П ПЕРЕМЕЩЕНИЯ ВРАГОВ
   230 GOTO 100
B) 100 V=ASC(INKEY$):IF V=255 THEN 150
   110 IF V=8 THEN DX=-1:DY=0:GOTO 150
   120 IF V=24 THEN DX=1:DY=0:GOTO 150
   130 IF V=25 THEN DY=1:DX=0:GOTO 150
   140 IF V=26 THEN DY=-1:DX=0
   150 X=X+DX:Y=Y+DY:PUT X,Y,ADDR(A(0)),1
```

```
   160 REM П+П ДВИЖЕНИЯ ВРАГОВ
   +++ GOTO 100
```
   ОПРЕДЕЛИТЬ НАЛОЖЕНИЕ КАРТИНОК МОЖНО ТРЕМЯ СПОСОБАМИ(ТО ЕСТЬ УЗНАТЬ КОГДА ГЛАВНЫЙ ГЕРЙ ПОЙМАЛ КОГО-ТО ИЛИ ЕГО КТО-ТО ПОЙМАЛ)Ё

 1)СРАВНЕНИЕ КООРДИНАТ.САМЫЙ ПРОСТОЙ И РАСПРОСТРАНЕННЫЙ СПОСОБ.ПРИ ЭТОМ КАРТИНКИ РАССМАТРИВАЮТСЯ КАК ПРЯМОУГОЛЬНИКИ.И ЕСЛИ ЭТИ ПРЯМОУГОЛЬНИКИ НАКЛАДЫВАЮТСЯ ДРУГ НА ДРУГА–ДАЖЕ ЕСЛИ ВИДИМЫЕ ЧАСТИ КАСАЮТСЯ ДРУГ ДРУГА–ТО СЧИТАЕТСЯ–ЧТО НАЛОЖЕНИЕ КАРТИНОК ПРОИЗОШЛО–ВЕСЬ АЛГОРИТМ СВОДИТСЯ К СЛЕДУЮЩИМ ПРОВЕРКАМ–ПУСТЬ X,Y-КООРДИНАТЫ ПЕРВОГО ПЕРСОНАЖА;X1,Y1-ВТОРОГО;DX,DY,DX1,DY1-РАЗМЕРЫ СООТВЕТСТВЕННО ПЕРВОГО И ВТОРОГО ПО X И Y.ЕСЛИ (X<X1<X+DX ИЛИ X1<X<X1+DX1) И (Y<Y1<Y+DY ИЛИ Y1<Y<Y1+DY1),ТО НАЛОЖЕНИЕ ПРОИЗОШЛО.

 2) ФУНКЦИЕЙ POINT(1) МОЖНО ОПРЕДЕЛИТЬ МАТЕМАТИЧЕСКИЙ ЦВЕТ ТОЧКИ ПО НАПРВЛЕНИЮ ДВИЖЕНИЯ.ЕСЛИ ЭТОТ ЦВЕТ-ЦВЕТ ФОНА–ТО МОЖНО ДВИГАТЬСЯ. ИНАЧЕ-ТАМ ЧТО-ТО НАХОДИТСЯ.

 3) ЕСЛИ ДВА СПРАЙТА НАХОДЯТСЯ В РАЗНЫХ ПЛОСКОСТЯХ–ТО НАЛОЖЕНИЕ МОЖНО ОПРЕДЕЛИТЬ ЛОГИЧЕСКИМ УМНОЖЕНИЕМ (AND) ЭКРАННЫХ БАЙТОВ–НАХОДЯЩИХСЯ ДРУГ НАД ДРУГОМ.ПРИ НУЛЕВОМ РЕЗУЛЬТАТЕ-НАЛОЖЕНИЯ НЕТ–ПРИ НЕНУЛЕВОМ-ЕСТЬ.

  ВЫБРАВ ИЗ ПЕРЕЧИСЛЕННЫХ ВЫШЕ ВАРИАНТОВ ТЕ–КОТОРЫЕ ВАМ БОЛЬШЕ НРАВЯТСЯ–ИЛИ ПРИДУМАВ ЧТО-ТО СВОЕ–ВЫ НАЧИНАЕТЕ ПИСАТЬ ПРОГРАММУ.НАЧАТЬ СЛЕДУЕТ С ОПИСАНИЯ ИГРОВОГО АЛГОРИТМА.ЗАСТАВКУ ВЫ НАРИСУЕТЕ ПОТОМ. ВСЯ ПРОГРАММА БУДЕТ СОСТОЯТЬ ИЗ НЕСКОЛЬКИХ БЛОКОВ:БЛОК ЗАСТАВОК(ПРАВИЛ–ВЫБОРА УРОВНЯ И Т.Д.)–БЛОК РИСОВАНИЯ ДВИЖУЩИХСЯ КАРТИНОК И ЗАПОМИНАНИЯ ИХ ОПЕРАТОРОМ GET,БЛОК ВЫВОДА ДЕКОРАЦИЙ–БЛОК УПРАВЛЕНИЯ ДВИЖЕНИЕМ ПЕРСОНАЖЕЙ–БЛОК УПРАВЛЕНИЯ ГЛАВНЫМ ГЕРОЕМ И БЛОК КОНЦОВКИ.

   ДЛЯ НАЧАЛА НАПИШИТЕ БЛОК УПРАВЛЕНИЯ ГЛАВНЫМ ГЕРОЕМ–ДОБЕЙТЕСЬ–ЧТОБЫ ОН ПЕРЕМЕЩАЛСЯ В ЗАДАННОМ УЧАСТКЕ ЭКРАНА.ОТКЛАКАЛСЯ НА КЛАВИШИ УПРАВЛЕНИЯ–НЕ ВЫХОДИЛ ЗА ПРЕДЕЛЫ ИГРОВОЙ ЗОНЫ.ЗАТЕМ НАПИШИТЕ БЛОК УПРАВЛЕНИЯ ДВИЖЕНИЕМ ДРУГИХ ПЕРСОНАЖЕЙ.ДЛЯ НАЧАЛА ПУСТЬ ОНИ ПРОСТО ДВИЖУТСЯ ПО ЗАДАННОМУ АЛГОРИТМУ.ЗАТЕМ ВВЕДДИТЕ ПРОВЕРКИ НА СТОЛКНОВЕНИЕ ИХ С ГЛАВНЫМ ГЕРОЕМ И РЕАКЦИИ НА ЭТО(ПЕРЕСЧЕТ ОЧКОВ–ЖИЗНЕЙ–ТОПЛИВА И Т.Д.).ТУТ УЖЕ МОЖНО ПОДКЛЮЧИТЬ БЛОК КОНЦОВКИ(ЕСЛИ ГЛАВНЫЙ ГЕРОЙ ВЫПОЛНИЛ ЗАДАНИЕ ИЛИ КОНЧИЛИСЬ ЖИЗНИ).ПОСЛЕ ЭТОГО МОЖНО ДОПИСАТЬ ЗАСТАВКУ–ВВЕСТИ ИНСТРУКЦИЮ–ВНЕШНЕЕ ГРАФИЧЕСКОЕ И МУЗЫКАЛЬНОЕ ОФОРМЛЕНИЕ И Т.Д.

МАШИННЫЕ ПРОГРАММЫ В БЕЙСИКЕ

   БЕЙСИК-ИНТЕРПРИТАТОР-ОЧЕНЬ МЕДЛЕННЫЙ ЯЗЫК.ОН ПОСЛЕДОВАТЕЛЬНО ЧИТАЕТ ОПЕРАТОРЫ В ПРОГРАММНОЙ СТРОКЕ И ВЫПОЛНЯЕТ ИХ.НА ЭТО УХОДИТ МНОГО ВРЕМЕНИ.И–ЕСЛИ ВАМ НУЖНО БЫСТРО ВЫПОЛНИТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ.ТО ВЫХОД-ТОЛЬКО В МАШИННЫЕ ПОДПРОГРАММЫ.МАШИНААЯ П+П-ЭТО П+П–НАПИСАННАЯ НА АССЕМБЛЕРЕ И ОТТРАНСЛИРОВАННАЯ В КОДЫ.ОБРАТИТЬСЯ К НЕЙ МОЖНО ИЗ БЕЙСИКА ИСПОЛЬЗУЯ ОПЕРАТОР USR(...).К ПОДПРОГРАММЕ ПРЕДЬЯВЛЯЮТСЯ ДВА ТРЕБОВАНИЯЁ +.ОНА ДОЛЖНА В НАЧАЛЕ РАБОТЫ СОХРАНЯТЬ ВСЕ РЕГИСТРЫ В СТЕКЕ.А ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ-ВОССТАНАВЛИВАТЬ ИХ–Ќ.П+П ДОЛНА ЗАКАНЧИВАТЬСЯ КОМАНДОЙ RET (КОД &C9).ПЕРЕДАЧА ДАННЫХ МЕЖДУ ПРОГРАММОЙ НА БЕЙСИКЕ И МАШИННОЙ П+П ОСУЩЕСТВЛЯЕТСЯ ОПЕРАТОРОМИ PEEK И POKE ЧЕРЕЗ ВЫБРАННЫЕ САМИМ ПРОГРАММИСТОМ ЯЧЕЙКИ.ЧТОБЫ ГАРАНТИРОВАТЬ П+П ОТ ПОРЧИ ИНТЕРПРИТАТОРОМ НУЖНО ЗАДАТЬ ГРАНИЦУ ОБЛАСТИ–ДОСТУПНОЙ ИНТЕРПРИТАТОРУ– ОПЕРАТОРОМ HIMEM.

   МАШИННАЯ П+П МОЖЕТ ЗАГРУЖАТЬСЯ В ОЗУ ОПЕРАТОРОМ BLOAD С МАГНИТОФОНА ИЛИ ЖЕ ЗАПИСЫВАТЬСЯ ИЗ САМОЙ ПРОГРАММЫ ОПЕРЕТОРОМ POKE.СЛЕДУЕТ УЧИТЫВАТЬ–ЧТО ПОКА ПИСАТЬ И ОТЛАЖИВАТЬ МАШИНЫЫЕ П+П УДОБНЕЕ ВСЕГО В МОНИТОРЕ–ПОЛЬЗУЯСЬ РЕДАКТОРОМ-АССЕМБЛЕРОМ–А ЗАТЕМ ВЫГРУЖАТЬ ИХ НА ЛЕНТУ В ФОРМАТЕ МОНИТОРА (W) И ЗАГРУЖАТЬ В ОЗУ КОМАНДОЙ BLOAD (ФОРМАТ МОНИТОРА И ДВОИЧНЫЙ ФОРМАТ В БЕЙСИКЕ СОВПАДАЮТ).

   НЕБОЛЬШОЙ ПРИМЕР:

```
  10 CLS:GOTO 60
  20 DATA &F5,&E5,&D5,&C5,&2A,&25,&70,&EB,&2A,&27,&70,&46
  30 DATA &23,&4E,&23,&C5,&D5,&7E,&12,&23,&14,&05,&C2,&11,&70
  40 DATA &D1,&C1,&13,&0D,&C2,&0F,&70,&C1,&D1,&E1,&F1,&C9,&80
  5O DATA &A0,&29,&70,3,3,255,255,255,255,255,255,255,255,255
  60 M=&7000:FOR I=1 TO 255: READ X: POKE M,X: M=M+1: NEXT I
  70 X=USR (&7000)
```

ПРОГРАММА ВЫВОДИТ КРАСНЫЙ ПРЯМОУГОЛЬНИК РАЗМЕРАМИ 24*3 ТОЧКИ.ТЕКСТ МАШИННОЙ П+П НА АССЕМБЛЕРЕ СЛЕДУЮЩИЙЁ

```
7000 PUSH PSW    700F PUSH B     701C DCR C
7001 PUSH H      7010 PUSH D     701D JNZ 700F
7002 PUSH D      7011 MOV A,M    7020 POP B
7003 PUSH B      7012 STAX D     7021 POP D
7004 LHLD 7025   7013 INX H      7022 POP H
7007 XCHG        7014 INR D      7023 POP RSW
7008 LHLD 7027   7015 DCR B      7024 RET
700B MOV B,M     7016 JNZ 7011
700C INX H       7019 POP D
700D MOV C,M     701A POP B
700E INX H       701B INX D
```

ТЕПЕРЬ ДЛЯ ТЕХ–КТО УЖЕ ПРОБОВАЛ ИСПОЛЬЗОВАТЬ МАШИННЫЕ П+П НА БЕЙСИКЕ.НА БЕЙСИКЕ С ИСПОЛЬЗОВАНИЕМ МАШИННЫХ П+П МОЖНО ПИСАТЬ ПРОГРАММЫ ПРИБЛИЖАЮЩИЕСЯ ПО УРОВНЮ К КОДОВЫМ ’ СХЕМА ПРОСТАЯЁ СЛОЖНЫЕ ВЫЧИСЛЕНИЯ И УЧАСТКИ–НЕ ТРЕБУЮЩИЕ СКОРОСТИ–ПИШИТЕ НА БЕЙСИКЕ–А СКОРОСТНЫЕ--В КОДАХ.

НО ВОТ ПРОБЛЕМАЁМАШИННЫЕ П+П И ДАННЫЕ–ЗАПИСЫВАЕМЫЕ В ОЗУ ЧЕРЕЗ POKE ЗАНИМАЮТ БОЛЬШОЙ ОБЬЕМ ПАМЯТИ(Ќ Ќ-, РАЗА БОЛЬШЕ ОТ РЕАЛЬНОГО ОБЬЕМА ПОДРОГРАММЫ).СУДИТЕ САМИ.П+П–СКАЖЕМ–ИЗ ДЕСЯТИ БАЙТ ЗАНИМАЕТ ЭТИ ДЕСЯТЬ БАЙТ–ПЛЮС ОТ ДВУХ ДО ЧЕТЫРЕХ СИМОЛОВ НА КАЖДЫЙ БАЙТ П+П В СТРОКЕ DATA. А ПОДГРУЗКА ЧЕРЕЗ BLOAD НЕ УДОБНА ТЕМ–ЧТО НУЖНО ЗАГРУЖАТЬ ДВА ДВА ФАЙЛА С МАГНИТОФОНА.КРОМЕ ТОГО И ТАМ И ТУТ ВОЗНИКАЕТ ПРОБЛЕМА ЗАЩИТЫ П+П ОТ ИЗМЕНЕНИЯ ИНТЕРПРИТАТОРОМ.

МЫ ПРЕДЛОЖИМ СПОСОБ–УДОБНЫЙ ВО ВСЕХ ОТНОШЕНИЯХ.ЗАГРУЗИТЕ БЕЙСИК–НАБЕРИТЕ В НЕМ ПРОГРАММЫ.ЛЮБУЮ–ТОЛЬКО +-АЯ СТРОКА ДОЛЖНА ИМЕТЬ ВИДЁ 1 REM A,A...ДО КОНЦА ПРОГРАММНОЙ СТРОКИ.ЗАТЕМ ЧЕРЕЗ УС+ВВОД+БЛК ЗАГРУЗИТЕ МОНИТОР–УЖЕ ЗАПИСАННЫЙ ПО РАБОЧЕЙ ОБЛАСТИ <1>(СМ."БАЙТ-2"), И ЗАПУСТИТЕ ЕГО БЛК+СБР.С АДРЕСА 4301Н РАСПОЛАГАЕТСЯ ТЕКСТ ПРОГРАММЫ НА БЕЙСИКЕ.ПРОГРАММА КОДИРУЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМЁВ НАЧАЛЕ КАЖДОЙ СТРОКИ ДВА БАЙТА-АДРЕС СЛЕДУЮЩЕЙ СТРОКИ(НЕ ЗАБУДЮТЕ–ЧТО СНАЧАЛА ИДЕТ МЛАДШИЙ БАЙТ–ЗАТЕМ СТАРШИЙ)–СЛЕДУЮЩИЕ ДВА БАЙТА-НОМЕР ДАННОЙ ПРОГРАММНОЙ СТРОКИ.ЗАТЕМ ИДУТ ОПЕРАТОРЫ (СМ.ТАБЛИЦУ ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ ОПЕРАТОРОВ В ОПИСАНИИ БЕЙСИКА).ДАННЫЕ–ЗНАКИ ПУНКТУАЦИИ И Т.Д. КОНЧАЕТСЯ СТРОКА НУЛЕВЫМ БАЙТОМ-00Н,ВСЯ ПРОГРАММА КОНЧАЕТСЯ ДВУМЯ НУЛЕВЫМИ БАЙТАМИ-00Н,00Н.В НАШЕМ СЛУЧАЕ В ПЕРВОЙ ПРОГРАММНОЙ СТРОКЕ БУДЕТ КОД 8ЕН(СЛУЖЕБНОЕ СЛОВО REM),20Н,(ПРОБЕЛ),40Н (БУКВА А)...41Н–++Н ВМЕСТО БУКВ А МЫ МОЖЕМ ЗАБИТЬ П+П– Т.Е.ДАННЫЕ.ТЕПЕР СНОВА НАЖМЕМ УС ВВОД БЛК И НАЧНЕМ ГРУЗИТЬ БЕЙСИК.ДОСТАТОЧНО ЗАГРУЗИТЬ НУЛЕВОЙ БЛОК И НАЖАТЬ БЛК+СБР.ТЕПЕРЬ МЫ СНОВА ПЕРЕШЛИ В ИНТЕРПРИТАТОР.МАШИННАЯ П+П ОКАЗАЛАСЬ ВКЛЮЧЕННОЙ В ТЕКСТ ПРОГРАММЫ НА БЕЙСИКЕ.ЕЕ НЕ НАДО ЗАЩАТЬ ОТ ИЗМЕНЕНИЯ–ОНА НАХОДИТСЯ ПО СВОИМ РАБОЧИМ АДРЕСАМ.ЕСЛИ НЕ ХВАТИТ ОДНОЙ ПРОГРАММНОЙ СТРОКИ–МОЖНО ИСПОЛЬЗОВАТЬ НЕСКОЛЬКО–ТАКИМ ЖЕ ОБРАЗОМ–РАЗМЕЩАЯ В НИХ П+П.НАПИСАВ И ЗАШИВ БАЗОВЫЕ П-П И ДАННЫЕ МОЖНО ПРИСТУПИТЬ К НАПИСАНИЮ ВНЕШНЕГО УПРАВЛЕНИЯ И ДРУГИХ ДЕЙСТВИЙ В ВИДЕ ПРОГРАММ НА БЕЙСИКЕ–НАЧИНАЯ–СКАЖЕМ С СОТОЙ СТРОКИ.СТРОЧКИ НА БЕЙСИКЕ ВЫ МОЖЕТЕ МЕНЯТЬ–РЕДАКТИРОВАТЬ И Т.Д.–НЕ ТРОГАЯ СТРОКИ С МАШИННАМИ П+П.ПРИ ЗАПУСКЕ ПРОГРАММ ИНТЕРПРИТАТОР–ВСТРЕЧАЯ СЛУЖЕБНОЕ СЛОВО REM (КОММЕНТАРИИ).ПЕРЕХОДИТ К ДРУГОЙ СТРОКЕ–ТО ЕСТЬ НАРУШЕНИЯ РАБОТЫ НЕ ПРОИСХОДИТ.МОЖНО ВООБЩЕ СДЕЛАТЬ ТАК.ЧТОБЫ БЕЙСИК НЕ ПОПАДАЛ НА СТРО-

КИ С ЗАШИТЫМИ ДАННЫМИ.ДЛЯ ЭТОГО НУЖНО НАБИТЬ СТРОКУ: 0 GOTO 100.
  ВЫ МОЖЕТЕ ЗАПИСАТЬ ПРОГРАММУ НА МЛ В СОСТАВЕ ОСНОВНОЙ ПРОГРАММЫ.
ТО ЕСТЬ П+П ЗАНИМАЮТ МЕСТА ИМЕННО СТОЛЬКО-СКОЛЬКО ОНИ ДОЛЖНЫ ЗАНИ-
МАТЬ-И НЕ НУЖНО ЗАГРУЖАТЬ ДВА ФАЙЛА.В КАЧЕСТВЕ БАЗОВЫХ МАШИННЫХ П+П
МОЖНО ИСПОЛЬЗОВАТЬ П+П ИЗ SPRITE ИЛИ МОДУЛЯ-+ (ПРИЛАГАЕТСЯ К АНИМА-
ТОРУ).ЭТИ П+П БЫСТРО ВЫВОДЯТ КАРТИНКИ ИЗ ОЗУ НА ЭКРАН.МОЖНО САМОМУ
СОЗДАТЬ НЕОБХОДИМЫЙ БАЗОВЫЙ НАБОР.ИСПОЛЬЗОВАНИЕ МАШИННЫХ П+П СТАНЕТ
ОСОБЕННО ЭФФЕКТИВНЫМ-ЕСЛИ ПОТИХОНЬКУ ДАВАТЬ АДРЕСА И ВХОДНЫЕ+ВЫХОД-
НЫЕ ДАННЫЕ ПОЛЕЗНЫХ П+П БЕЙСИКА.ПОКА ЖЕ ДАЕМ АДРЕС САМОЙ НУЖНОЙ -
С АДРЕСА 2CFCH РАСПОЛАГАЕТСЯ П+П ВВОДА КОДА НАЖАТОЙ КЛАВИШИ БЕЗ
ОЖИДАНИЯ НАЖАТИЯЁ В РЕГИСТРЕ А ПРИ ВЫХОДЕ НАХОДИТСЯ КОД НАЖАТОЙ
КЛАВИШИ (АНАЛОГИЧНО ИСПОЛЬЗОВАНИЮ ASC(INKEY$)).П+П СОДЕРЖИМОГО
ОСТАЛЬНЫХ РЕГИСТРОВ НЕ НАРУШАЕТ.

СТАНДАРТНЫЕ ПОДПРОГРАММЫ ВВОДА/ВЫВОДА В ФОРМАТЕ МОНИТОРА

ФИЗИЧЕСКИЙ ФОРМАТ: 0 0...256раз...0 E6 D2 D2 D2 D2 ИМЯ ФАЙЛА
0 0...256 раз..0 E6 СТАРШИЙ БАЙТ АДРЕСА НАЧАЛА МЛАДШИЙ БАЙТ АДРЕСА НАЧАЛА СТАРШИЙ БАЙТ АДРЕСА КОНЦА МЛАДШИЙ БАЙТ АДРЕСА КОНЦА.. БАЙТЫ ПРОГРАММЫ...БАЙТ-КОНТРОЛЬНАЯ СУММА

```
ORG 9000H
IMON PUSH B
PUSH D
PUSH H
DI
PUSH H
LXI H,IM+1
RUSH H
MVI E,4
CALL IMON7
CALL IMON13
POP H
POP D
MVI H,0
IMON1: CALL IMON15
CPI 0
JZ IMON3
CPI 255
JZ IMON3
DCR H
JNZ IMON1
```

```
IMON2:MVI A,1
ANA A
JMP IMON 6
IMON3:MVI H,14H
MOV L,A
IMON 4:CALL IMON15
CMP L
JNZ IMON2
DCR H
JNZ IMON4
MVI A,255
CALL IMON11
MOV H,B
MOV L,C
SHLD NACH
CALL IMON10
MOV H,B
MOV L,C
SHLD KON
XCHG
LDA NACH
CMA
```

```
MOV L,A
LDA NACH+1
CMA
MOV H,A
INX H
DAD D
XCHG
LHLD NACH
MVI B,0
IMON5: CALL IMON15
MOV M,A
ADD B
MOV B,A
MOV A,D
ORA E
DCX D
INX H
JNZ IMON5
CALL IMON12
MOV A,C
CMP B
JNC IMON2
```

```
HRA A
IMON6: POP H
POP D
POP B
EI
RET
IMON7:MOV D,E
IMON8:MOV E,D
MVI A,255
IMON9: CALL IBUTE
CPI 0D2H
JNZ IMON8
DCR E
```

```
JNZ BYTE3
IN 01
ANI 10H
MOV A,E
MOV A,D
ORA A
JP BUTE5
MOV A,C
CPI 0E6H
JNZ BUTE4
HRA A
STA PRINV
JMP BYTE6
```

```
CALL OMON4
INX H
JNZ OMON2
MOV A,B
CALL OBYTE
POP H
POP D
POP B
EI
RET
OMON3:MOV A,H
CALL OBYTE
MOV A,L
```

```
OBUTE1:
MOV A,D
RLC
MOV D,A
MVI A,1
HRA D
ANI 01
OUT 0
CALL OBUTE2
MVI A,0
HRA D
ANI 1
OUT 0
```

```
RZ                  BYTE4:CPI 19H   CALL OBYTE      CALL OBYTE2
MVI A,8             JNZ BYTE2       OMON4:MOV A,H   DCR C
JMP IMON9           MVI A,255       CMP D           JNZ OBYTE1
IMON10:MVI A,8      STA PRINV       RNZ             POP PSW
IMON11:CALL IBYTE   BYTE6:MVI D,9   MOV A,E         POP D
MOV B,A             BYTE5:DCR D     CMP L           POP B
IMON12:CALL IMON15  JNZ BYTE2       RET             RET
MOV C,A             LDA PRINV       OMON5:XRA A     OBYTE2:
RET                 XRA C           MVI E,1         PUSH PSW
IMON13:MVI E,3      POP D           CALL OMON8      LDA CONZAR
IMON14:CALL IMON15  POP B           MVI A,0E6H      MOV B,A
MOV M,A             RET             CALL OBYTE      POP PSW
INX H               CONR:DB 4BH     MVI A,0D2H      OBYTE3:
ORA A               PRINV: DB 0     MVI E,4         DCR B
JNZ IMON13                          CALL OMON10     JNZ OBYTE3
MVI M,'" '          OMON PUSH B     OMON6: INX H    RET
DCR E               PUSH D          MOV A,M         CONZAP:
JNZ IMON14          PUSH H          CPI '" '        DB 32H
RET                 DI              JZ OMON7        IM: DB '"ЦЕН
IMON15:MVI A,8      LXI H,IM        CALL OBYTE      ТР БАЙТ" ',0
JMP IBYTE           CALL OMON5      JMP OMON6
IBYTE:PUSH B        MVI B,0         OMON7:INX H     П/П ВВОДА-
MVI C,0             CALL OBYTE      MVI E,3         OMON.П+П ВЗЯТЫ
MOV D,A             DCR B           CALL OMON10     ПРОГРАММЫ
IN 01               XTHL            RET             " КАРАНДАШ ".
ANI 10H             XTHL            OMON8:MVI B,0
MOV E,A             JNZ OMON1       OMON9:CALL OBYTE
BYTE2:IN 01         MVI A,0E6H      DCR B
ANI 10H             CALL OBYTE      JNZ OMON9
CMP E               LHLD NACH       DCR E
JZ BYTE2            CALL OMON3      JNZ OMON8
RLC                 XCHG            RET
RLC                 LHLD KON        OMON10:CALL OBYTE
RLC                 XCHG            DCR E
RLC                 XCHG            JNZ OMON10
MOV A,C             CALL OMON3      RET
RAL                 LHLD KON        NACH :DW 100H
MOV C,A             XCHG            KON:DW 3FFH
PUSH PSW            LHLD NACH       OBYTE:PUSH B
LDA CONR            OMON2:MOV A,M   PUSH D
MOV B,A             CALL OBYTE      PUSH RSW
POP PSW             ADD B           MOV D,A
BYTE3:DCR B         MOV B,A         MVI C,8
```

А ЗНАЕТЕ ЛИ ВЫ ?

XYZOLOG

В ИГРЕ "HYZOLOG" ТАК ЖЕ СУЩЕСТВУЕТ ВОЗМОЖНОСТЬ СМЕНЫ УРОВНЯ (КЛАВИШИ УС, СС и З),ПРИЧЕМ–ЕСЛИ ВЫ НАЖМЕТЕ ЭТИ КЛАВИШИ–КОГДА ВАС ЗАДЕЛА "ЗВЕЗДА"-НО ВЫ ЕЩЕ НЕ РАСПАЛИСЬ–ТО ПРОИСХОДИТ СМЕНА УРОВНЯ С СОХРАНЕНИЕМ ПОПЫТКИ.

ТМК SOFT Г.КИРОВ-1991

HELICOPTER

В ИГРЕ "HELICOPTER" :

1. КЛАВИША "ПРОБЕЛ"-СБРОС БОМБЫ (ПРИЧЕМ–ЕСЛИ ВЫ ПОПАЛИ В ЦЕЛЬ–ТО ЗАПАС БОМБ НЕ УМЕНЬШАЕТСЯ).

2. КЛАВИША "ТАБ"-СТРЕЛЬБА ИЗ ПУЛЕМЕТА (ПОПАДАНИЕ ВО ВРАЖЕСКИЙ ВЕРТОЛЕТ ЗАСТАВЛЯЕТ ЕГО СНИЖАТЬСЯ.ОН ТЕРЯЕТ НА НЕБОЛЬШОЙ ПРОМЕЖУТОК ВРЕМЕНИ УПРАВЛЕНИЕ–А ЕСЛИ ОН НАХОДИТСЯ БЛИЗКО ОТ ЗЕМЛИ ТО ОН РАЗБИВАЕТСЯ).
,. КЛАВИША "АР2"-ОСТАНОВКА ИГРЫ (ПОВТОРНОЕ НАЖАТИЕ ПРОДОЛЖАЕТ ИГРУ).
4. КЛАВИША "F4"-КОНЕЦ ИГРЫ–ВЫХОД НА ЗАСТАВКУ(ПЕРЕЗАПУСТИТЬ ИГРУ НАЖАТИЕМ КЛАВИШ БЛК+СБР НЕ СОВЕТУЕМ ТАК КАК ОНА "ЗАВИСАЕТ").
5. КЛАВИША "СТРЕЛКА ВЛЕВО-ВВЕРХ"-УМЕНЬШАЕТ СКОРОСТЬ ПОЛЕТА (ПРИ НАЖАТИИ НЕСКОЛЬКО РАЗ ДО НУЖНОГО МИНИМУМА).
6. КЛАВИША "СТР"-УВЕЛИЧИВАЕТ СКОРОСТЬ ПОЛЕТА (ПРИ НАЖАТИИ НЕСКОЛЬКО РАЗ ДО НУЖНОГО МАКСИМУМА).
7. ВЕРТОЛЕТ УПРАВЛЯЕТСЯ КЛАВИШАМИ КУРСОРА ПРИЧЕМ НАЖАТИЕ НА ЭТИ КЛАВИШИ ВЫЗЫВАЕТ ДВИЖЕНИЕ ВЕРТОЛЕТА В СООТВЕТСТВУЮЩУЮ СТОРОНУ–ОСТАНОВИТЬ КОТОРОЕ МОЖЕТ ОДНОКРАТНОЕ НАЖАТИЕ КЛАВИШИ С ПРОТИВОПОЛОЖНЫМ НАПРАВЛЕНИЕМ.

ТМК SOFT Г.КИРОВ-1991

THE RANNER MAN

В ИГРЕ "THE RANNER MAN" :
1. КЛАВИША "F2"-КОНЕЦ ИГРЫ–ПЕРЕХОД НА ЗАСТАВКУ.
2. КЛАВИША "СТРЕЛКА ВЛЕВО-ВВЕРХ"-ПРЫЖОК ВЛЕВО.
3. КЛАВМША "СТР"-ПРЫЖОК ВПРАВО.
4. КЛАВИШИ КУРСОРА-УПРАВЛЕНИЕ ДВИЖЕНИЕМ.
5. КЛАВИША "ПРОБЕЛ"-СТОП.
6. КЛАВИША "ТАБ"-ИСЧЕЗНОВЕНИЕ ЧЕЛОВЕЧКА(ПРИ ЭТОМ ОН СТОИТ НА МЕСТЕ)–А ПОВТОРНОЕ НАЖАТИЕ "ПРОЯВЛЯЕТ" ЕГО.ЭТО НУЖНО ДЛЯ ПРОХОЖДЕНИЯ ПОСЛЕДНЕГО УРОВНЯ (ТАК КАК ТАМ НЕЛЬЗЯ ПРЫГАТЬ И МЕШАЕТ "ЧЕРЕПОК")–ТАК ЖЕ НА ЭТОМ ПОДУРОВНЕ МОЖНО "ПРОДОЛБИТЬ" МОСТИК ДЛЯ ПОСЛЕДНЕГО ПРЫЖКА НА ЛЕСТНИЦУ–НАЖАТИЕМ КЛАВИШИ "ВНИЗ"–ПОТОМ МОЖНО ПРЫГАТЬ.

ТМК SOFT Г.КИРОВ-1991

ПРИ ЗАПУСКЕ ПРОГРАММ "МОНИТОР-ОТЛАДЧИК", "БЕЙСИК-V2.5","ПОЛЕТ", "ШТУРМ","XONIX"–"ТЕТРИС" НАЖАТИЕМ "БЛК+СБР" ПОПРОБУЙТЕ ОДНОВРЕМЕННО УДЕРЖИВАТЬ КЛАВИШУ "А".ТОГДА НА ЗАСТАВКЕ СООТВЕТСТВЕННОЙ ПРОГРАММЫ ВЫСВЕТИТСЯ ФАМИЛИЯ АВТОРОВ.

Программное обеспечение для ком-
пьютера "ВЕКТОР-06Ц": игровые,
учебные и прикладные программы Вы
можете заказать по адресу: г.Киров
ул. Карла-Маркса, 126,центр "БАЙТ"
приемный пункт "Унисон". А иного-
родние могут сделать заказ по ад-
ресу: г.Киров, 610006, а/я 1248,
Зубкову Александру Николаевичу.

Если у Вас есть вопросы и проблемы, связанные с "ВЕКТОРом-06Ц", предложения и обьявления, материалы для статьи или заметки, напишите нам. Мы сумеем Вам помочь. Наш адрес: г. Киров, 610006, а/я 1248, центр "Байт".

ПОДКЛЮЧЕНИЕ ПК "ВЕКТОР-06Ц" К ТЕЛЕВИЗОРУ

Известно много случаев,когда при подключении ПК "Вектор-06ц" к ТЛВ ТЛВ выходит из строя. Обычно "сгорает" одна из микросхем: К174УК1 (МСА660) или К174АФ5. Микросхемы "сгорают"из-за напряжения порядка 100В на RGB-выходе компьютера. Напряжение образуется в результате не совсем удачного построения импульсного блока питания. Это напряжение падает до нуля в момент соединения "корпуса" ТЛВ с "корпусом" компьютера. Исходя из этого, предлагаем два простых пути сохранения телевизора:

1. Соединение ТЛВ и ПК производить при выключенном ПК.
2. Доработать шнур, соединяющий ПК и ТЛВ, следующим образом:
   - разобрать на шнуре штекер и припаять небольшой кусочек провода между контактом 4 и одной из металлических половинок штекера(произвести доработку обоих штекеров);
   - в RGB-гнезде, которое вы встроили в свой телевизор, кусочком провода с помощью пайки соедините контакт 4 и корпус гнезда.

Мы советуем выполнить эти рекомендации,т.к. "сгорающие" микросхемы очень дефицитны.

МАЛЕНЬКИЕ БУКВЫ В РЕДАКТОРЕ-АССЕМБЛЕРЕ

У многих пользователей БПЭВМ "Вектор-06Ц" возникают проблемы при попытке создавать тексты, пользуясь редактором-ассемблером,входящим в состав базового программного обеспечения компьютера. При попытке записать на магнитную ленту текст, содержащий большие и маленькие буквы, запись происходит до тех пор, пока не встретится маленькая буква (с кодом больше 127). Ниже будет приведен способ, позволяющий обойти это ограничение, но сначала одно очень важное замечание. дело в том, что после запуска редактор настраивается на работу с набором символов КОИ-7. Этот код позволяет использовать символы с кодами 0...127, т.е. можно использовать только большие буквы русского и латинского алфавитов. Следовательно, разработнный для создания текстов программ, ассемблер-редактор не рассчитан на использование маленьких букв. Маленькие буквы, которые удается вводить при использовании редактора - результат недостаточной проработанности редактора и Монитора-отладчика,разработчиками которого являются,кстати, небезызвестные нам г-да Темиразов & Соколов.

Столкнувшись с невозможностью записать на магнитофон набранный текст, большинство умельцев прибегает к помощи Монитора-отладчика, выгружая в его формате область памяти,содержащей необходимые данные При этом мало кто задумывается: а в каком коде набран текст, доставивший мне столько хлопот ?

Действительно, код этот представляет собой нечто весьма несуразное и не имеющее аналогов в мире, во всяком случае цивилизованном. Вы можете возразить мне: какая разница, в каком коде набран текст,ведь на экран он выводится нормально ? Дело в том, что проблемы начнутся при попытке напечатать текст на принтере.

Поэтому я искренне советую вам перейти на использование кода КОИ-8 Тексты, набранные в этом коде, хорошо печатаются на принтере, особенно на "МС 6313". (Но принтер - это отдельная проблема.)

А сейчас желающие работать в КОИ-8,изменять число символов в строке, записывать и читать программы,содержащие маленькие буквы должны набрать следующую программу:

```
00C0: 3E 2F 32 06 0B 3E 4E 32 E9 0B 21 D9 00 22 07 01
00D0: 21 DB 00 22 27 01 C3 F6 00 CD 03 F8 FE 80 D8 FE
00E0: C0 D0 E6 7F C9 01 00 00 7E FE FF C8 81 4F 3E 00
00F0: 88 47 23 C3 E8 00 21 E5 00 22 0E 09 22 CE 09 7F
```

Теперь выгрузите ее на магнитную ленту вместе с ассемблером-редактором. Первый запуск программы после загрузки осуществляется командой GC0, последующие - GC0 или G100.

По адресу 00C6 содержится количество символов в строке - Вы можете изменить его.

После того, как основные проблемы решены,возникает вопрос:как быть с текстами, набранными в коде КОИ-7 ? Воспользуйтесь такой вот программой:

```
0040: 21 00 28 7E FE FF CA 00 00 CD 51 00 77 23 C3 43
0050: 00 FE 60 DA 67 00 FE 80 DA 68 00 FE C0 DA 6D 00
0060: FE E0 DA 72 00 D6 20 C9 F6 80 C3 67 00 E6 7F C3
0070: 67 00 D6 60 C3 67 00 00 00 00 00 00 00 00 00 00
```

г.Вятка А. Кочуров

КАК НА БЕЙСИКЕ СЫГРАТЬ МЕЛОДИЮ БОЛЬШЕ 127 СИМВОЛОВ

В 4 номере "Байта" нашел просьбу откликнуться на вопрос читателей, как сыграть на Бейсике мелодию, занимающую более 127 символов. Сделать, на мой взгляд, это довольно легко.

На "Векторе" я программирую музыку таким вот, весьма простым, способом.Речь пойдет о программировании одного канала,т.к. для остальных - все аналогично.

```
10 CLEAR 5000          резервирую память для
               строковых переменных в зави-
               симости от об'ема музыки

20 A$="L805ABP.....C16E#"  пишу строку, величину к-рой
               можно менять по собствен-
               ному усмотрению, оборвав там,
               где это более удобно. естест-
               венно, она д.б. меньше 127
               символов.
```

```
 30 PLAY A$

 40 IF PEEK(772)<>0 THEN 40 проверяю, все ли сыграно.

 50 A$="L16O2BPAP....E#P"   пишу очередную часть мелодии

 60 PLAY A$

 70 IF PEEK(772)<>0 THEN 70 проверяю, все ли сыграно

                     и т.д.
```

Таким образом, можно напихать в программу сколько угодно длинную мелодию, пока хватит памяти.

Некоторая сложность возникает в том,что при переходе от одного кусочка к другому во время исполнения музыки появляется небольшая пауза, примерно в 1/16 или 1/32 долю, заметная на слух только специалисту. Избежать этого можно следующим образом.

Строку, с предложением на музыкальном макроязыке, стараться закончить на паузе. Например:

```
 10 A$="L2O5APGPD#PCP"   , и из последней паузы
```

просто вычесть эту 1/16 или 1/32, кому как нравится. строка будет иметь такой вид:

```
 10 A$="L2O5APGPD#PCPP4P8P16"
       или       - если вычтена 1/16
 10 A$="L2O5APGPD#PCP4.P16" (пауза вместо 8/16
                стала 7/16)

 10 A$="L2O5APGPD#PCP4P8P16P32"
       или       - если вычли 1/32
 10 A$="L2O5APGPD#PCP4.P16."    (пауза вместо 16/32
                стала 15/32)
```

Сбивания темпа между двумя кусочками теперь нет. Хуже, если по каким-то причинам вам не удается закончить строку паузой,либо в вашей мелодии вообще нет пауз. Тогда подобную операцию придется производить с любой нотой, отчего общее ее восприятие слегка пострадает (она будет как бы раздроблена на мелкие части), хотя, повторяю, что не посвященный в такие нюансы вряд ли что-нибудь заметит.

В случае работы с несколькими каналами необходимо заканчивать стро ки с их записью синхронно по времени,т.е. если в одном канале у вас записано в целых 1/4 доли, то и в остальных д.б. столько же (или не больше). В последнем случае удаление лишней доли необходимо произвести только в самом длинном по времени звучания канале.

г.Киров — Потапов Д.А.

Примечание редакции "Байт":

Байт по адресу 772 имеет следующий формат:

```
     7 6 5 4 3 2 1 0
     0 0 0 0 0 X X X         0 - не звучит
           | | |         1 - звучит
           | | +-- 1 КАНАЛ
           | |
           | +---- 2 КАНАЛ
           |
           +------ 3 КАНАЛ
```

В Драйверах устройств имеется такая же ячйка по адресу 1A8BH.

РУССКИЕ БУКВЫ В EDASM

В текстовом редакторе пакета EDASM в результате ошибочного определения конца файла становится невозможным правильный вывод на ленту текстов, содержащих символы с кодами больше 127, т.к. авторы проверяют только наличие в байте знакового (старшего) бита. Предлагаемые исправления в коде редактора позволяют обойти этот недостаток:

```
975: 3C  INR A
976: C8  RZ
977: 3D  DCR A
978: 81  ADD C
979: 4F  MOV C,A
97A: 88  ADC B
97B: 91  SUB C
```

П.Желнов, г.Отрадный

Цикл статей по программированию на языке Паскаль

С этого номера "Байта" мы начинаем цикл статей о программировании на языке ЛС-Паскаль.

Примечание редакции "Байт":
Программисты, занимающиеся изучением языка ЛС-Паскаль ! Если вы что-то обнаружили (интересные особенности,ошибки и т.д.) или составили свои процедуры или функции, необходимые в работе, напишите нам по адресу: 610006, г.Киров, а/я 1248, "Байт" (с пометкой Паскаль). Самое интересное будет опубликовано!

TMK Soft ; г.Киров-1991

Часть I.

НОВЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ "ЛС-ПАСКАЛЬ v2.1"

1. Назначение модуля "HELP".

Каждый, кому приходилось приходилось писать программы на языке ЛС-Паскаль, ощущал отсутствие (или присутствие в форме,затрудняющей

затрудняющей отладку) ряда необходимых процедур и функций,имеющихся в Бейсике.
Модуль "Help1V27" (см. Приложение 1) представляет собой библиотеку процедур и функций, предназначенных для использования при написании программ на языке ЛС-Паскаль v2.1 для БПЭВМ "Вектор-06Ц". Использование этого модуля облегчает создание программ.
В состав "Help.Pas v(1.27)" входят следующие подпрограммы:

Процедуры:

1. CLS - очистка экрана, перемещение курсора в левый верхний угол экрана.
2. HOME - перемещение курсора в левый верхний угол экрана.
3. CUR(X,Y) - прямая адресация курсора в знакоместо и и строку.
4. PAUSE(S,D) - пауза в выполнении программы в секундах и долях екунды.
5. RANDOM - перебор значений RND.
6. POKE(A,D) - запись числа в ячейку памяти.
7. OUT(P,D) - запись числа в порт.
8. SCREN0(16 параметров)- программирование таблицы цветов.
9. SCREN3(8 параметров) - задание маски вывода.
10. TEXT(6 параметров) - --//--
11. WRTIME - вывод показания таймера.
12. SPRITE(4 параметра) - вывод спрайтов.

Функции:

1. INP(P) - чтение числа из порта.
2. SGN(P) - определение знака числа.
3. ABS(P) - определение абсолютной величины аргумента.
4. PEEK(A) - чтение числа из ячейки памяти.
5. POINT(X,Y) - цвет точки.
6. SQR(P) - вычисление квадратного корня.
7. FACT(P) - вычисление факториала.

2. Условия применения.

Для разработки программного обеспечения на языке ЛС-Паскаль с использованием модуля "Help.Pas v(1.27)" необходима следующая минимальная конфигурация технических средств:

- БПЭВМ "Вектор-06Ц";
- монитор или ТВ-приемник;
- бытовой кассетный магнитофон.

а также программное обеспечение:

файлы:

1. PAS1.ROM (система ЛС-Паскаль v2.1);
2. PAS2.ROM (КомПас);
3. PASLIB.MON (библиотека подпрограмм);
4. HELP1v27.PAS (модуль);
5. SPRITE.MON (п/п к PROC SPRITE);
6. MONITOR2v5.ROM (Монитор-отладчик);

и для более эффективного составления программ желательно иметь:

- FONTED1v1.BAS (редактор знакогенератора 1.1;Лапушнер,Кишинев)
- МузРед-1.BAS (Дашковский, Черновцы);
- SPRITEDESIGN.BAS (спрайт-редактор; Луппов, Киров);
- Карандаш.ROM (графический редактор; Смирнов, Ростов-на-Дону);
- Редактор палитры v1.2.BAS (Кишинев);
- Редактор масок v1.1 (Кишинев),

а также описания всех перечисленных выше программ.

3. Подключение модуля "Help" к программе пользователя.

В редактор системы ЛС-Паскаль директивой LOAD грузят модуль "Help" (если он был набран ранее и сохранен на МЛ; если отсутствует, то вводят с клавиатуры). Директивой INSERT#1 вводят описания переменных, констант и т.д. Затем директивой APPEND вводят свои процедуры и функции, а также тело программы.

P.S.Внимание ! При составлении программы пользователь(программист) может убрать (уничтожить) неиспользуемые процедуры или функции модуля командой DELETE#, т.к. они не взаимосвязаны (то есть работают автономно, не обращаясь друг к другу).

Внимание !!! В программе пользователя не использовать переменные: P,P0,P1,P2,P3,P4,P5,P6,P7,P8,P9,PA,PB,PC,PD,PE,PF,а также массив P.

4. Описание процедур и функций модуля "Help v(1.27)".

PROC CLS

Назначение: Очистка экрана, перемещение курсора в левый верхний угол экрана.
Входные параметры: Нет
Действие: Очистка экрана (в соответствии с режимом SPLAN, разрешенные экранные плоскости).
Побочное действие: а). Установка режима вывода UNIT(1);
б). перевод строки;
в). выход из режима вывода увеличенных символов.
Пример: 22 CLS;

PROC HOME

Назначение: Перемещение курсора в левый верхний угол экрана.
Входные параметры: Нет
действие: Перемещение курсора в левый верхний угол экрана.
Побочное действие: а). Установка режима вывода UNIT(1);
б). перевод строки;
в). выход из режима вывода увеличенных символов;
г). "антискроллинг" экрана (против смещения экрана при многократном выводе текста и графической информации в одно и то же место экра-

на).
Пример: 23 HOME;

PROC CUR(X,Y)

Назначение: Прямая адресация курсора.
Входные параметры: X - знакоместо (по горизонтали)
X= от 0 до 41
Y - строка (по вертикали)
Y= от 0 до 24
Нумерация снизу-вверх, слева-направо.
Действие: Прямая адресация курсора в знакоместо и строку.
Побочное действие: а). Режим вывода UNIT(1);
б). перевод строки (отмена вывода увеличенных символов).
Пример: 114 CUR(4,11);

PROC PAUSE(S,D)

Назначение: "Пустой" цикл, задержка в выполнении программы.
Входные параметры: S - пауза целых секунд
S= от 0 до 128
D - пауза в долях секунды
D= от 0.02 до 0.99
Действие: Задержка в выполнении программы (более удобна, чем использование "пустого" цикла).
Побочное действие: Режим вывода UNIT(1).
Пример: 13 PAUSE(0,2) - пауза в 0.02 с.
14 PAUSE(0,70)- пауза в 0.7 с.

PROC RANDOM

Назнчение: Перебор значений RND.
Входные параметры: Нет
Действие: Перебор значений RND, т.к. при запуске программы распределение RND одинаково. (Более случайные,не повторяемые значения RND).
Побочное действие: Нет
Пример: 32 RANDOM;

PROC POKE(A,D)

Назначение: Запись числа в ячейку памяти.
Входные параметры: A - адрес ячейки памяти
A= от 0 до 65535
D - число
D= от 0 до 65535 8
Действие: Запись числа в ячейку памяти. Если число меньше 255, то оно записывается в одну ячейку. Число больше 255 записывается следующим образом:в ячей ку A - младший байт, в ячейку A+1 - старший байт Это удобно для передачи данных кодовым п/п и т.д

```
Побочное действие: Нет
Пример:        41 POKE(13000,40000);
```

PROC SCREN0(P0,P1,P2,P3,P4,P5,P6,P7,P8,P9,PA,PB,PC,PD,PE,PF)

```
Назначение:    Установка физических цветов.
Входные параметры: P0 -  0 цвет (цвет фона)
           P1 -  1 цвет
           ...........
           PE - 14 цвет
           PF - 15 цвет
           Внимание! В приложении даны два варианта  проце-
           дур: для режима 1 (16 цветов) и режима 2 (8 цве-
           тов).
Действие:      Установка физических (таблица  цветогенератора).
           Эта процедура более удобна при отладке  программ
           (на отладчике), чем встроенная процедура SCOLOR.
Пример:        11   SCREEN0 (0,128,16,208,7,134,22,63,0,197,34,
            192,2,152,82,255);
           Устанавливает цвета Бейсика.
```

PROC SCREN3(P0,P1,P2,P3,P4,P5,P6,P7)

```
Назначение:    Установка маски вывода.
Входные параметры: P0 - 1 байт маски
           ................
           P7 - 8 байт маски
Действие:      Установка маски вывода для прямоугольников и си-
           мволов увеличенного размера. (Более  удобна  для
           отладки программ  на  отладчике, чем  встроенная
           процедура MASC).
Побочное действие: Нет
Пример:        57 SCREN3(254,254,254,0,240,240,240,0);
           Устанавливает маску "кирпичи".
```

PROC OUT(P,D)

```
Назначение:    Вывод числа в порт.
Входные параметры: P - номер порта
           D - число (от 0 до 255)
Действие:      Вывод числа D в порт P.
Побочное действие: Нет
Пример:        11 OUT(3,136);
```

Продолжение в следующем "Байте".

А ЗНАЕТЕ ЛИ ВЫ ?

(или кое-что о секретах программ).

"КИРАШ"

В игре "Кираш" очень трудно пройти все 26 раундов, имея всего 3 попытки. И хотя вышла версия игры с 96 жизнями ("Кираш-М"), все равно очень многие не могут пройти всю игру.

Для установки "вечной" жизни нужно загрузить программу в Монитор-отладчик (см."Байт-2" и "Байт-4") и изменить ее содержимое по таблице изменений.

Таблица изменений (по рабочим адресам).

адрес	старое значение	новое значение
2350H	CAH \	00H (NOP)
2351H	FAH --> (JZ 2AFAH)	00H (NOP)
2352H	2AH /	00H (NOP)

Если же вы хотите только задать большое количество попыток, то их количество (в шестнадцатиричном виде) хранится в ячейке 1E18H (рабочий адрес). Изменения проводятся слдующим образом:

```
#S1E18

1E18 03 #230 (десятичное значение) <ВК>
1E19 <F4>
```

Затем измененный вариант программы выгружают на ленту (в формате ROM) директивой "O>".

P.S. При увеличении числа попыток, имйте ввиду, что максимальное число попыток 255.

TMK Soft ; Киров-1991

"ПОЛЕТ"

В игре "Полет" при нажатии клавиши <ВК> происходит появление маркера-указателя готовности смены карты, т.е. можно менять карты, не уничтожив нужное количество техники протвника.

Всего в игре 8 карт (0-7). Из них 4 основные (0-3), а остальные (4-7) аналогичны остальным, но вам мешают ракеты. Клавиша включения ракет "R",при этом на основных картах (0-3), повторное нажатие клавиши "R" выключает их.

TMK Soft ; Киров-1991

ОСОБЕННОСТИ "МОНИТОРА-ОТЛАДЧИКА"

1. При запуске Монитора-отладчика (или Монитора "Super Monster v3.5") бывает случайное двойное нажатие клавиш СБР и БЛК. При этом Монитор остается по адресам загрузки (с 0000H или 0100H),но он работоспособен. Для распределения по стандартным рабочим адресам (область 1 или 2) нужно запустить его с адреса 2D00H (G2D00 <ВК>), при этом выводится меню распределения.

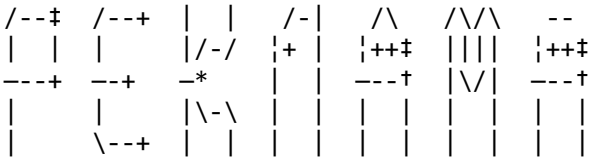
2. В Мониторе и программах, функционирующих в нем (с условием, что они используют его подпрограммы), возможна приостановка вывода текстовой информации - клавиши: УС и S, возобновление вывода - УС и Q.

3. Клавиша <F5> включает режим прописных букв, а повторное нажатие выключает его.

4. Клавиши УС и P включают режим дублированного вывода (все, что выводится на дисплей,одновременно дубируется на принтере),а повторное нажатие выключает его.

TMK Soft ; Киров-1991

+--+
| Если у Вас есть вопросы и проблемы, связанные с "ВЕКТОРом-06Ц", |
|предложения и объявления, материалы для статьи или заметки, напи- |
|ќите нам. Мы сумеем Вам помочь. Наш адрес: г. Киров, 610006, |
|а/я 1248, центр "Байт". |
+--+



Всем пользователям "ВЕКТОРа-06Ц", работающим с магнитофоном, центр "БАЙТ" предлагает новую копирующую утилиту "COPY-N4", предназначенную для копирования программ в кодах (формат ROM).

"COPY-N4" выдает имя программы, позволяет редактировать его, запускает программу на выполнение, позволяет возвращаться из режима "ЗАПИСЬ" в режим "ЖДУ!", а самое главное - помогает восстанавливать копии плохого качества. Дело в том, что при сбое загрузки копир не сбрасывает программу, а продолжает грузить дальше. Сбойный участок на экране остается пустым и не заполняется информацией. В любой момент Вы можете перемотать пленку и продолжать грузить с любого места программы. Для удобства копир выдает номер текущего блока. Копир поможет в тех ситуациях, когда ошибка "плавающая". Например, помеха от сети, загрязненная или старая головка и т.д. Кроме того, если сбой происходит в одном и том же месте, но это место попадает либо на не особо важную информацию (графика), либо на не используемый участок (отдельные подпрограммы в ДРАЙВЕРАХ или знакогенератор), то "COPY-N4" Вам также пригодится. Цена программы - всего 10 рублей. Заказать ее Вы можете в нашем центре. Ждем Ваших заказов!

::
Редакция "БАЙТа" поздравляет всех своих читателей с новым 1992 годом
и желает всяческих удач в наступающем году!!!
::

ОСОБЕННОСТИ "МОНИТОРА-ОТЛАДЧИКА"

1. При запуске Монитора-отладчика (или Монитора "Super Monster v3.5") бывает случайное двойное нажатие клавиш СБР и БЛК. При этом Монитор остается по адресам загрузки (с 0000H или 0100H),но он работоспособен. Для распределения по стандартным рабочим адресам (область 1 или 2) нужно запустить его с адреса 2D00H (G2D00 <ВК>),

при этом выводится меню распределения.
 2. В Мониторе и программах, функционирующих в нем (с условием, что они используют его подпрограммы), возможна приостановка вывода текстовой информации - клавиши: УС и S, возобновление вывода - УС и Q.
 3. Клавиша <F5> включает режим прописных букв,а повторное нажатие 3 выключает его.
 4. Клавиши УС и P включают режим дублированного вывода (все, что выводится на дисплей,одновременно дубируется на принтере),а повторное нажатие выключает его.

TMK Soft ; Киров-1991

Программы COPY

Сейчас у пользователей БПЭВМ "Вектор-06Ц" появилось множество различных копиров (COPY-BLOCK; COPY v(2.5); COPY-Z; COPY v(2P); DCOPY; COPY1 и др.).
Обычно многие пользователи не знают возможности имеющихся копиров, их назначение, особенности использования.
Мне хочется выделить в одну группу основные копиры, которые желательно иметь всем пользователям БПЭВМ:

- COPY v(3.0) - для установки защиты "имя";
- COPY-HELP - для снятия защит "имя";
- COPY v(N.3) - для просмотра и изменения имени программы,ее запуска;
- COPY-S v(2.1) - для ускоренной переписи программ и переписи программ, не поддающихся копированию.

Остальные копиры я советую стереть, т.к. все они в основном повторяют друг друга (в разных вариациях).

COPY-HELP

Занимает 8 блоков (2 кБт). После запуска высвечивает "COPY-HELP ЧТЕНИЕ".
Не реагирует на запрещенные символы в имени, ошибки.
Следует внимательно следить за загрузкой программы,т.к. при ошибке копир переходит в режим готовности к записи,а программа загрузилась не до конца.

COPY v(3.0) - (COPY 3)

Занимает 9 блоков.После загрузки высвечивает "COPY v(3.0) ЧТЕНИЕ"
Выгружает программы с защитой "имя" (для программ с 1-го блока добавляет 0-й информационный блок).
Реагирует при загрузке на неверное имя, ошибки.
Внимание! Выгрузка возможна только один раз,т.к. во второй раз выгружается неработоспособная программа.

COPY v(N.3) - (COPY N3)

Занимает 11 блоков. После загрузки высвечивает "COPY v(N.3) ЧТЕНИЕ".
Позволяет узнать имя загружаемой программы, изменить его перед записью, прервать процесс записи, запустить программу из копира. Реагирует на неверное имя и ошибки чтения.

Правила использования.

После успешной загрузки вверху экрана высветится:

```
УС WRITE
СС NAME
```

Для изменения имени нажмите "СС", после чего Вы можете изменять буквы имени (по одной) клавишами "УС" и "СС" (перебор знакогенератора),а клавишей "РУС/ЛАТ" закрепляете букву. Учтите,что опрос клавиш очень быстрый и вы вначале будете "проскакивать", поэтому нужно кратковременное нажатие клавиши "РУС/ЛАТ".
Далее клавишей "УС" выйдите в режим ожидания записи (при этом "УС" и "СС" - увеличение или уменьшение скорости, наблюдается по шкале в левом верхнем углу экрана; "РУС/ЛАТ" - начало записи).
Во время записи: клавиша "РУС/ЛАТ" в нажатом состоянии - однократная запись (без дублирования блоков), а также "УС" - запуск программы из COPY, а "СС" - прерывание записи и выход в режим ожидания. Имейте ввиду, что во время записи клавиши "УС", "СС" и "РУС/ЛАТ" сканируются в перерывах между записью блоков.

COPY-S v(2.1)

Занимает 1 блок (без инструкции) или 11 блоков с инструкцией.После загрузки высвечивает "(C) CopyS".
Позволяет копировать программы в любом формате(ROM,DOS,PAS,BAS,MON и т.д.). Копиру безразлично наличие защит и порядок записи программ "COPY-S" записывает дубликат кассеты, и хотя в паузах между программами записывается шум и треск, это не влияет на качество записи и чтения программ. Для использования необходимо 2 магнитофона.

P.S. Описанные здесь копиры Вы можете приобрести в центре "Байт".

TMK Soft ; г.Киров-1991

МОНИТОР "SUPER MONSTER v(3.5)

У многих пользователей возникают вопросы по использованию Монитора "Super Monster v(3.5)".
После запуска Монитор высвечивает информацию-подсказку об областях распределения и ждет нажатия клавиш "1", "2" или "3". При нажатии на клавиши "1" или "2" Монитор распределяется как в стандартной версии, а при нажатии клавиши "3" Монитор распределяется в 1 область, при этом с адреса 9100H по 93FFH расположен загрузчик программ в формате ROM.
После распределения желательно очистить память директивой F0,FF,0, т.к. с 0000H по 00FFH расположен "мусор".

Запуск начального загрузчика:

```
B<вк>   - для загрузки программ по рабочему адресу(будте внимательны, не затрите Монитор !!!).
BM<вк>  - для загрузки программ со смещением 100H (например,с 0 блока).
```

После этого экран очистится, появится "карта загрузки" и в левом верхнем углу надпись: "BOOT" (если B) или "BOOT & MOVE" (если BM). После загрузки (или при ошибке чтения) происходит очистка экрана и

выход в Монитор. 5
Внимание! Следите за процессом загрузки,т.к. при ошибке происходит выход в Монитор (с очисткой экрана) и неизвестно, полностью ли загрузилась программа.

TMK Soft ; г.Киров-1991

ДЛЯ ТЕХ, КТО ПРОГРАММИРУЕТ НА АССЕМБЛЕРЕ (ИВ-1)

Программа двоичного умножения

Способ умножения в столбик, начиная со старшего разряда, со сдвигом влево.

```
 MPL: LXI H,0    ;Регистры: B  - счетчик
      MVI B,8    ;       DE - первый множитель
 L1:  DAD H      ;       A  - второй множитель
      RAL    ;       HL - прозведение
      JNC DEC    ;
      DAD D      ;Вызов подпрограммы: CALL MPL
      ACI 0
 DEC: DCR B
      JNZ L1
      RET
```

При использовании программы следует учитывать, что точное произведение однобайтного числа на двухбайтное в общем случае должно занимать 3 байта. Т.е. при умножении больших чисел эта программа дает округленный результат.

ПОДКЛЮЧЕНИЕ ПРИНТЕРА "CONSUL-260" (ИВ-4)

Принтер "CONSUL-260" и другие модели можно подключить так,как описано в журнале "Радио" N12 за 1989 год,используя тот же программный драйвер.При этом необходимо вывод БЛК/В1 разъема X1 соединить с общим проводом, резисторы R52, R53, R54 заменить на 10 кОм.В Мониторе для 1 области в ячейках FC32H, FC33H поставить адрес,с которого оттранслирован драйвер.

Юшкетов В.Д., г.Слободской

НЕКОТОРЫЕ ТОНКОСТИ РАБОТЫ С РЕДАКТОРОМ-АССЕМБЛЕРОМ (ИВ-4)

При работе с редактором-ассемблером могут возникнуть некоторые трудности, если редактируемый текст занимает в памяти адреса больше 5000H. При этом программа начинает ассемблироваться с ошибками, а в тексте при каждом повторном запуске редактора-ассемблера оказывается "мусор". Происходит это из-за того,что при запуске программы директивой G100 всем регистрам микропроцессора присваивается определенное значение.В частности регистр стека SP по умолчанию принимает значение 5000H. После этого в область с 5000H и ниже начинает записываться содержимое стека, стирая текст программы,если он там находится. 6
Чтобы этого не происходило,необходимо: сразу после того как Вы загрузили с ленты редактор-ассемблер, ввести команду Монитора XS, по-

сле чего набрать новое значение верхушки стека - E6EEH (если Монитор занимает адреса 9400H-9FFFH, E000H-FFFFH) или 66EEH (если Монитор занимает адреса 5400H-7FFFH). Теперь при каждом запуске редактора-ассемблера, а также при запуске Вашей отассемблированой программы регистру стека будет присваиваться адрес области, специально отведенной в Мониторе под хранение данных стека.

Флейдерман И., г.Кишинев

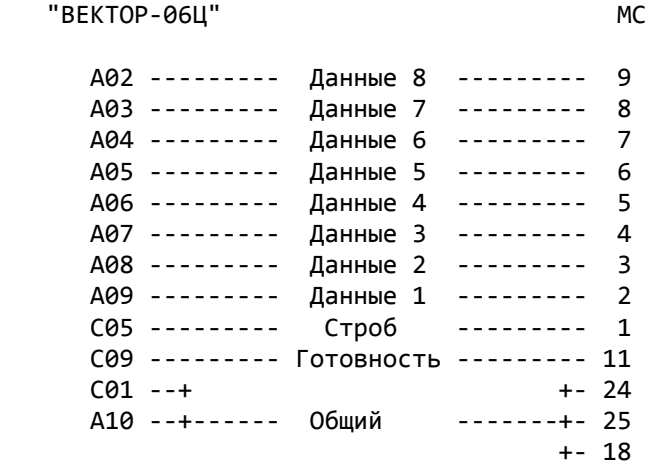
УСТРАНЕНИЕ СБОЕВ ОЗУ

Хочу рассказать как устранить сбои ОЗУ. Хотя "Вектор" устойчив в отношении сбоев, но они всетаки дают о себе знать,особенно на больших программах.На экране вылезает "мусор",а Бейсик сообщает об ошибке в какой-либо строке, написанной совершенно верно. Все дело в корректности подводки к микросхемам памяти питания. В частности,если микросхемы 1,3 и 4-го рядов имеют несколько контактов с шиной питания +5В, то микросхемы 2-го ряда - только один контакт.Это увеличивает вероятность сбоя ОЗУ микросхем 2-го ряда. Проложив монтажным проводом несколько дублирующих шин к +5В, удалось полностью устранить сбои !!!

г.Киров Нагаев А.А.

ПОДКЛЮЧЕНИЕ ПРИНТЕРА МС-6312 (ИВ-6)

Хочу поделиться опытом подключения принтера МС-6312. Со стороны БПЭВМ желательно установить перемычку между контактами С01 и А10, а со стороны принтера - между контактами 24 и 25. Необходимо использовать 11-жильный кабель, распаянный по следующей схеме:



Кроме того,со стороны принтера обязательна установка перемычки между контактами 18 и 24 (или 25).Кабель,изготовленный таким образом, обеспечивает обмен информацией между БПЭВМ и принтером по интерфейсу ИРПР-М. Печатающее устройство "Электроника МС-6312" относится к типу "Роботрон", поэтому не требует инициализации оператором SCREEN 6,0, т.к. устанавливается автоматически при запуске Бейсика или Монитора. Поскольку в применяемой версии Бейсика используется кодиро-7 вка символов КОИ-7 НО/1 ,т.е. совмщенный русско-латинский алфавит, прописные буквы, сразу после включения принтера необходимо установить набор символов КОИ-7 НО/1 с помощью оператора:

```
LPRINT CHR$(27)"R"CHR$(2)
```

Дальнейшее управление принтером осуществляется согласно описаний Бейсика и принтера.

А.Ирецкий, г.Санкт-Петербург

УЛУЧШЕНИЕ РАБОТЫ КЛАВИАТУРЫ

Рано или поздно владельцы БПЭВМ "Вектор-06Ц" сталкиваются с проблемой клавиатуры: дребезг, отказ в работе отдельных клавиш. Дело в том,что контакты клавиатуры изготовлены из разных материалов: олова и алюминия. Эти металлы не совместимы между собой,при их совместной работе наблюдается интенсивная электрохимическая коррозия, что и приводит к отказам клавиатуры.

Чтобы избавиться от этого неприятного явления, необходимо заменить алюминиевую фольгу на подвижном контакте на медную, приклеив медные кружочки клеем "Момент".

Правда найти тонкую медную фольгу достаточно сложно, а при использовании толстой фольги клавиши отказывают при попадании малейшей пылинки под контакты. Чтобы этого избежать, необходимо каким-либо заостренным предметом (гвоздем) сделать в кружочке два выступа так, чтобы во время работы контакты замыкались не всей пластиной,а этими выступами.

Указанные переделки значительно повышают надежность работы клавиатуры.

Набор медных контактов для ремонта клавиатуры Вы можете приобрести если обратитесь в наш центр.

Напольский С.

BASIC I (ИВ-7)

Клавиши, используемые для редактирования строк

```
  []  - перемещение курсора на символ вправо
  []  - перемещение курсора на символ влево
  []  - перемещение курсора на слово влево
  СТР - перемещение курсора на слово вправо
  []  - перемещение курсора в начало строки
  []  - перемещение курсора в конец строки
  ЗБ  - удаление символа слева от курсора
  F2  - удаление символа под курсором
```

Использование клавиши УС

УС Нажатие клавиши УС приостанавливает процесс выполнения программы на BASICe (или вывод строк программы по LIST), а ПРОБЕЛ или любой символьной клавиши - продолжает рабо-8 ту интерпретатора. Благодаря этому можно осуществить пошаговый режим выполнения программы (или построчный вывод листинга), для чего, удерживая УС, нажимать ПРОБЕЛ. Каждое нажатие ПРОБЕЛ будет вызывать выполнение одного оператора, после которого интерпретатор переходит в режим ожидания до следующего нажатия ПРОБЕЛ.

УС+Ц(^C) Нажатие клавиши Ц при нажатой клавише УС приводит к прерыванию выполнения программы (или вывод листинга) и вы-

ходу в непосредственный режим. Настоятельно рекомендуем применять для останова программы ^C, а не СБР+БЛК, т.к. сброс изменяет некоторые установки (например, HIMEM) и может вызвать другие нежелательные последствия.

УС+E(^E) Прерывает программу так же,как и ^C, но дополнительно выдает на экран номер строки, которая выполняется в момент прерывания. Использование ^E весьма полезно при отладке и редактировании программ, т.к. позволяет легко определить, какие строки отвечают за выполнение того или иного куска программы. Для этого достаточно запустить программу, дождаться интересующего момента и нажать ^E.

При использовании УС, ^C, ^E следует помнить,что интерпретатор опрашивает клавишу УС только в промежутках между выполнениями операторов. Поэтому приостанов или прерывание не происходит во время выполнения медленных операторов, таких, как PAUSE, PAINT и т.п. Также УС, ^C, ^E не прерывают звучания уже запущенного оператора PLAY.Если после нажатия ^C (или ^E) программа остановилась,но не появилось приглашение "=>" и курсор (и надпись с номером прерванной строки в случае ^E),значит, в программе имелись изменения цветов и (или) экранных плоскостей. Чтобы сделать курсор видимым, надо вслепую дать все или какие-то из следующих команд:

```
SCREEN 2,15
COLOR 7,0
SCREEN 0,7,34
```

Оператор SCREEN 3

Оператор SCREEN с режимом 3 устанавливает маску, которая влияет на результат выполнения оператора LINE с параметрами BF и BS (закрашенные прямоугольники и символы увеличенного размера).

Формат оператора:

```
SCREEN 3,M0[,M1[,M2...[,M7]]]
```

где M0...M7 - выражения в диапозоне 0...255.

Весь экран "ВЕКТОР-06Ц" условно разбит на прямоугольники 8*8 точек Каждый прямоугольник - это 8 байт. Нижний байт каждого прямоугольника маскируется параметром M0, второй снизу - M1 и т.д. Каждый бит маскирующего байта разрешает (1) или запрещает (0) менять цвет в этой точке. Рассмотрим действие оператора SCREEN 3 на примере:

```
10 CLS
20 COLOR 6,0,0
30 SCREEN 3,247,247,247,0,127,127,127,0
40 PLOT 10,0,1
50 LINE 245,200,BF  Продолжение следует
```

+-----------------------------------+
| Программное обеспечение для ком- |
| пьютера "ВЕКТОР-06Ц": игровые, |
| учебные и прикладные программы Вы |
| можете заказать по адресу: г.Киров |
| ул. Карла-Маркса, 126,центр "БАЙТ" |
| приемный пункт "Унисон". А иного- |
| родние могут сделать заказ по ад- |
| ресу: г.Киров, 610006, а/я 1248, |
| Зубкову Александру Николаевичу. |
+-----------------------------------+

+---+
| Если у Вас есть вопросы и проблемы, связанные с "ВЕКТОРом-06Ц", |
|предложения и объявления, материалы для статьи или заметки, напи- |
|шите нам. Мы сумеем Вам помочь. Наш адрес: г. Киров, 610006, |
|а/я 1248, центр "Байт". |
+---+

PUTUP

Всем известна игра "Putup". И, наверное, наскучили уже 15 этажей, которые Вы прошли, помогая маленькому белому существу найти ключ. Не отчаивайтесь! При желании Вы сможете изменить этажи,как Вам угодно...

В игре "Putup" блоки этажей располагаются с адреса 1C8EH, каждый этаж 11 байт "в высоту" и 16 - "в длину", занимает 176 байт и записывается в блоке последовательно, байт за байтом, слева направо и сверху вниз.

Адреса (шестнадцатиричные) начала этажей:

1 этаж - 1C8E	6 этаж - 1FFE	11 этаж - 236E
2 этаж - 1D3E	7 этаж - 20AE	12 этаж - 241E
3 этаж - 1DEE	8 этаж - 215E	13 этаж - 24CE
4 этаж - 1EDE	9 этаж - 220E	14 этаж - 257E
5 этаж - 1F4E	10 этаж - 22BE	15 этаж - 262E

Обозначения в блоках этажей:

01 - пусто	06 - огонь	0B - яблоко
02 - стена	07 - ключ	0C - ананасик
03 - ограничительё	08 - замок	0D - виноград
04 - "вопрос"	0D - тайный выход	0E - жизнь
05 - тайник	0A - вишенки	0F - "вертолет"(графика)

11 - "мышь" (существо с глазами и красным хвостиком) графикой.

В сектор, где стоит ограничитель, может попасть только игрок и самое страшное существо.

Но в блоках этажей не указано, где появляются существа и выход.Эта информация содержится с адреса 1C16H в блоках координат. Координаты каждого этажа занимают 8 байт и записываются в таком порядке:

1 байт - горизонтальная координата выхода
2 байт - вертикальная координата выхода
3 байт - код первого существа

4 байт - координата появления его по горизонтали
5 байт - координата появления его по вертикали
6 байт - код второго существа
7 байт - координата появления его по горизонтали
8 байт - координата появления его по вертикали

Коды существ:

01 и 02 - "рогатый" (синее существо)
03 и 04 - "мышь" (синее существо с глазами и красным хвостиком)
05 - 08 - "вертолетик"

Самое страшное - коричневое существо - появляется, если в одном месте задать появление кодов 09 и 0A. Впрочем, если появится только код 09, беспокоиться не нужно - коричневое существо будет неподвижно, но не менее опасно.

Адреса (шестнадцатиричные) блоков координат:

1 этаж - 1C16	6 этаж - 1C3E	11 этаж - 1C66
2 этаж - 1C1E	7 этаж - 1C46	12 этаж - 1C6E
3 этаж - 1C26	8 этаж - 1C4E	13 этаж - 1C76
4 этаж - 1C2E	9 этаж - 1C56	14 этаж - 1C7E
5 этаж - 1C36	10 этаж - 1C5E	15 этаж - 1C86

Расчет координат: отчет координат из левого верхнего угла, который имеет координату 00. Соседняя клетка имеет координату 02. Таким образом, координаты по горизонтали будут (слева направо): 00, 02, 04, 06, 08, 0A, 0C, 0E, 10, 12, 14, 16, 18, 1A, 1C, 1E и по вертикали (сверху вниз): 00, 02, 04, 06, 08, 0A, 0C, 0E, 10, 12, 14. Принцип же расчета аналогичен тому, который используется школьниками в игре "Морской бой", только в данном случае поле 16*11 и другие обозначения координат.
Успеха в проектировании новых этажей!
Напоследок информация для тех, кто захочет изменить и заставку.Заставка располагается с адреса 0976H, коды там аналогичны тем, что используются в блоках этажей. Надпись "PUTUP", написанная ягодками, располагается с адреса 0A06H. Обозначения там: 9B - ягодка,20 - пустое место.

SimBroth company, г.Киров, 1991

ДЛЯ ТЕХ, КТО ПРОГРАММИРУЕТ НА АССЕМБЛЕРЕ

Подпрограмма десятичного сложения с выводом суммы на экран

Подпрограмма является специализированной и ориентирована для вывода счета, времени и т.д.

```
                                3
SUMM: ADD M    ;складываем          RAC: LXI H,CIFR ;задаем
               ;начало области, в кото-
               ;рой находятся образцы
               ;дес. цифр

   DAA  ;десят. коррекция      PUSH H
   MOV M,A  ;запоминаем результат      PUSH PSW
   PUSH PSW ;выводим результат        ANI 15     ;выделя-
                                                 ;ем мл.
                                                 ;разряд
```

```
    PUSH H   ;на экран          CALL RAC1  ;выводим
                                           ;его на
                                           ;экран
    CALL RAC                          POP PSW
    POP H                             POP H
    POP PSW                           RRC  ;выделяем ст.
    RNC  ;возврат, если результат     RRC  ;дес. разряд
         ;уложился в один байт        RRC
  SUMM1:DCX H   ;переходим на следующие RRC
    MOV A,M  ;два разряда             ANI 15
    ADI 1    ;добавляем единицу       JMP RAC1 ;выводим
                                              ;его на
                                              ;экран
    DAA  ;десят. коррекция         RAC1:JZ RAC2  ;раачиты-
                                           ;ваем адр.
                                           ;дес.цифры
    MOV M,A  ;запоминаем результата     RLC  ;умножаем
    PUSH PSW ;вывод результата          RLC  ;на 8
    PUSH H                               RLC
    CALL RAC                             MOV C,A  ;добавляем
    POP H                                MVI B,0  ;смещение
    POP PSW                              DAD B    ;относите-
                                                  ;льно ба-
                                                  ;зового
                                                  ;адреса
    JC SUMM1 ;если результат опять не RAC2:PUSH D   ;выводим
         ;уложился в один байт, то     CALL VC1F;цифру на
                                              ;экран
         ;переходим на SUMM1           POP D
    RET                           DCR D;подготавли-
                                        ;ваем регистры
  VC1F: MVI C,8  ;п/п вывода            RET  ;для вывода
  VC1F1:MOV A,M  ;образа дес. цифры          ;след.цифры
    STAX D   ;на экран
    INX H
    INX D
    DCR C
    JNZ VC1F1
    RET
```

При обращении к п/п в регистрах должна содержаться следующая информация: A - что прибавить, HL - адрес ячеек, которые складываются с A, DE - адрес вывода на экран. По адресу CIFR должна находиться информация о том, что выводить на экран. На каждую цифру отводится по 8 байт, сначала идет 0, 1 и т.д. ... до 9.

Например, по адресу SCORE у Вас находится результат игры, занимающий, скажем, шесть десятичных цифр: SCORE:DB 00H,00H,50H. И Вам нужно добавить туда десятичное число 25. Тогда Вы используете следующую последовательность команд: MVI A,25H LXI H,SCORE+2 LXI D,0CDD0H 4
CALL SUMM...

Подпрограмма выводит два десятичных разряда, если был перенос, то следующие два разряда и т.д. Если выводимое число занимает только один разряд из пары, то впереди выводится незначащий ноль.

BASIC

С адреса 2D65ш находится подрограмма обработки прерываний.

```
 2D55H, 2D64H        - таблица цветов (16 байт)
 3C20H, 3C21H, 3C22H - младший, средний и старший байты таймера
 3C23H           - ячейка скроллинга
 2EF1H           - математич. цвет бордюра
 3C24H           - признак звучания каналов
 2572H           - подпрограмма  рисования точки, PLOT. В (B)-
          координата Y, (C) - координата X, (A) - ре-
          жим (0 - стирание точки, 1 - установка точ-
          ки, 2 - установка граф. курсора)
 26BDH           - п/п рисования линии, в (A) - координата  Y,
          в (C) - X
 25BFH           - п/п рисования прямоугольника, в (A) - коор-
          дината Y, в (C) - X
 25ECH           - п/п рисования прямоугольника с закраской, в
          (A) - Y, в (C) - X
 24E0H           - п/п COLOR, в (A) - матем. цвет. Предварите-
          льно требуется следующая последовательность
          команд: PUSH PSW  ANI F0  STA 3DB0  POP PSW
          ANI 0F  STA 2EF2  MOV E,A  CALL 24E0.
 19A8H           - п/п PGET, при выходе в (A) - матем.цвет те-
          кущей точки
```

Приведенные адреса подпрограмм предназначаются для программистов, использующих Бейсик с машинными подпрограммами. Подпрограммы не сохраняют содержимого регистров, это остается на долю программиста.

ДЛЯ ТЕХ, КТО ПРОГРАММИРУЕТ НА АССЕМБЛЕРЕ

П/п ввода/вывода в формате эмулятора РК и Монитора-1200

Уважаемые читатели, на этот раз мы предлагаем Вашему вниманию стандартные п/п ввода/вывода в формате эмулятора РК-86.
Физический формат записи на МГ: 256 нулевых байт, Е6, ст.байт адреса начала, мл. байт адреса начала, ст.байт адреса конца, мл.байт адреса конца, информационные байты, Е6, ст.байт КС, мл.байт КС.

Тот же самый алгоритм ввода/вывода используется в формате Монитора 1200. Только следует внести следующие изменения: убрать строчку СшANGE: MVI A,0F8ш и СшANGE2: MVI A,0E6ш CALL OBYTE. Т.е. перед кс не должен идти синхробайт 0E6H.

```
;П/П LOADPK-86  LEN2: INR D       POP B              5
LOAD: DI          IN 01           RET
 LXI H,0          ANI 10H        SR: MOV A,H
 MVI D,40H        CMP E           SUB D
LOAD1: CALL LEN   JZ LEN2         RNZ
 PUSH D         LEN3: INR D       MOV A,L
 MVI D,0          IN 01           SUB E
 MOV E,A          ANI 10H         RET
 DAD D            CMP E       VSP: LHLD KON
 POP D            JNZ LEN3        XCHG
 DCR D            MOV A,D         LHLD NACH
```

```
 JNZ LOAD1       POP D            LXI B,0
 DAD H           RET              RET
 DAD H          BYTE: MVI A,8    PKS: PUSH H
 MOV A,H        BYTE1: PUSH B     ADD C
 RRC             PUSH D           MOV C,A
 ANI 7FH         MVI C,0          PUSH PSW
 ADD H           MOV D,A          CALL SR
 STA CONR        IN 01            JZ PKS1
 MVI A,0F8H      ANI 10H          POP PSW
 CALL BYTE1      MOV E,A          MOV A,B
 STA NACH+1     BYTE2: IN 01      ADC M
 CALL BYTE       ANI 10H          MOV B,A
 STA NACH        CMP E       PKS2: MOV H,B
 CALL BYTE       JZ BYTE2         MOV L,C
 STA KON+1       RLC              SHLD KS
 CALL BYTE       RLC              POP H
 STA KON         RLC              CALL SR
 CALL VSP        RLC              INX H
LOAD3: CALL BYTE MOV A,C          RET
 MOV M,A         RAL         PKS1: POP PSW
 CALL PKS        MOV C,A          JMP PKS2
 JNZ LOAD3       PUSH PSW
CHANGE: MVI A,0F8H   LDA CONR        ;П/П SAVEPK-86
 CALL BYTE1      MOV B,A          DI
 MOV D,A         POP PSW          CALL SAVE12
 CALL BYTE      BYTE3: DCR B      LHLD NACH
 MOV E,A         JNZ BYTE3        PUSH H
 LHLD KS         IN 01            CALL SAVE2
 CALL SR         ANI 10H          LHLD KON
 RZ              MOV E,A          CALL SAVE2
 MVI A,1         MOV A,D          XCHG
 STA ERROR       ORA A            POP H
 RET             JP BYTE5         LXI B,0
KS: DW 0         MOV A,C         SAVE3: MOV A,M
NACH: DW 0       CPI 0E6H         CALL OBYTE
KON: DW 0        JNZ BYTE4        CALL SAVE14
ERROR: DB 0      XRA A            JNZ SAVE3
PRINV: DB 0      STA PRINV       CHANGE2: MVI A,0E6H
CONR: DB 4BH     JMP BYTE6        CALL OBYTE
LEN: PUSH D     BYTE4: CPI 19H    MOV H,B
 MVI D,0         JNZ BYTE2        MOV L,C
 IN 01           MVI A,255        CALL SAVE2
 ANI 10H         STA PRINV        EI
 MOV E,A        BYTE6: MVI D,9    RET
LEN1: IN 01     BYTE5: DCR D     SAVE12: XRA A
 ANI 10H         JNZ BYTE2        MOV B,A
 CMP E           LDA PRINV       SAVE13: CALL OBYTE
 JZ LEN1         XRA C            DCR B
 MOV E,A         POP D            JNZ SAVE13
 MVI A,0E6H      CALL VSP         PUSH D          CALL OBYTE2  6
 JMP OBYTE       INX H            PUSH PSW        DCR C
SAVE14: PUSH H   RET              MOV D,A         JNZ OBYTE1
 ADD C        SAVE15: POP PSW     MVI C,8         POP PSW
 MOV C,A         JMP SAVE16      OBYTE1: MOV A,D  POP D
 PUSH PSW        SAVE2: MOV A,H   RLC          POP B
 CALL SR         CALL OBYTE       MOV D,A         RET
 JZ SAVE15       MOV A,L          MVI A,1      OBYTE2: PUSH PSW
 POP PSW         JMP OBYTE        XRA D           LDA CONZAP
 MOV A,B      SR: MOV A,H         ANI 1           MOV B,A
```

```
  ADC M           SUB D           OUT 0           POP PSW
  MOV B,A         RNZ             CALL OBYTE2  OBYTE3: DCR B
 SAVE16: MOV H,B  MOV A,L         MVI A,0         JNZ OBYTE3
  MOV L,C         SUB E           XRA D           RET
  SHLD KS        RET              ANI 1        CONZAP: DB 46H
  POP H        OBYTE: PUSH B      OUT 0
```

ФИЗИЧЕСКИЙ ФОРМАТ ЗАПИСИ НА МЛ
В ФОРМАТЕ ЗАГРУЗЧИКА

Вначале передается преамбула: 4 серии блоков (25 нулей и 25 код 55H).

Затем передаются информационные блоки.Каждый блок начинается с серии: 16 нулей, 4 байта - 55H, 1 байт - 0E6H, 4 байта - 0, 30 байт - заголовок блока (включает идентификатор "NODISC00", имя файла и расширение - 11 байт, дата - 6 байт, начальный адрес загрузки программы, количество блоков, остаток блоков для передачи), 1 байт - контрольная сумма.

Каждый передаваемый блок состоит из 8 подблоков,каждый из которых, в свою очередь, имеет свой собственный заголовок: 4 байта - 0, 1 байт - 0E6H,1 байт - счетчик (при двойной записи для первого блока 80...87, для второго - 88...8F), 1 байт - контрольная сумма заголовка блока.Затем следуют 32 байта собственно информации и 1 байт контрольная сумма.

```
  ;п/п записи в формате ROM          EI                DCR D
   ORG 8000H              ZAP6:  JMP ZAP6          JNZ NBYTE
   MVI A,88H              BLOK:  PUSH B            RET
   OUT 0                         PUSH H       BYTE:  PUSH B
   MVI A,0C9H                    PUSH D                 PUSH D
   STA 38H                       MVI D,10H              PUSH PSW
   DI                            XRA A             MOV D,A
  ZAP: LXI H,SHAPKA              CALL NBYTE             MVI A,0
   LXI SP,9300H                  MVI D,4                ANI 0EH
   PUSH B                        MVI A,55H              MOV E,A
   PUSH D                        CALL NBYTE             MVI C,8
   PUSH H                        MVI A,0E6H   BYTE1: MOV A,D
   MVI E,0                       CALL BYTE              RLC
   MVI B,1BH                     XRA A             MOV D,A
  ZAP1: MOV A,M                  MVI D,4                MVI A,1
   ADD E                         CALL NBYTE             XRA D
   MOV E,A                       MVI D,1DH              ANI 01
   INX H                  BLOK1:       MOV A,M        ORA E
   DCR B                         CALL BYTE              OUT 1
   JNZ ZAP1                      INX H             CALL ZAD
   MOV B,M                       DCR D             MVI A,0
   INX H                         JNZ BLOK1              XRA D
   ADD M                         MOV A,C                ANI 1
   MOV E,A                       CALL BYTE              ORA E
7
   MOV C,M                       LDA M1                 OUT 1
   INX H                         ADD C             CALL ZAD
   MOV D,M                       CALL BYTE              DCR C
   MOV A,B                       STA M2                 JNZ BYTE1
   ADD D                         POP D             POP PSW
   DCX H                         MVI H,8                POP D
   DCX H                  BLOK3:       MVI L,20H      POP B
   MOV M,A                       PUSH D                 RET
   ADD E                         XRA A             ZAD:   PUSH PSW
```

```
   STA M1                          MVI D,4           LDA CONZAP
   MVI H,4                         CALL NBYTE              MOV B,A
  ZAP4: MVI L,19H                  POP D             POP PSW
  ZAP2: XRA A                      MVI A,0E6H        ZAD1:  DCR B
   CALL BYTE                       CALL BYTE               JNZ ZAD1
   DCR L                           LDA M2                  RET
   JNZ ZAP2                        ADD B             SHAPKA:      DB
'NODISC000
   MVI L,19H                       MOV C,A               30591COPY
  ZAP3: MVI A,55H                  MOV A,B                    V4 COM',0
   CALL BYTE                       CALL BYTE         DB 0,40H,1BH,0,0
   DCR L                           LDA M2                  |   |
   JNZ ZAP3                        CALL BYTE               |   +- число
   DCR H                  BLOK2:       LDAX D                          |
блоков
   JNZ ZAP4                        INR E                        |
   POP H                           CALL BYTE               +--- нач.блок
   PUSH B                          ADD C
   MOV D,B                         MOV C,A
   MVI E,0                         DCR L             M1:    DB 0
  ZAP5: MVI B,80H                  JNZ BLOK2         M2:    DB 0
   CALL BLOK                       MOV A,C           CONZAP:      DB 20H
   MVI E,0                         CALL BYTE
   MVI B,88H                       INR B
   CALL BLOK                       DCR H
   INR D                           JNZ BLOK3
   DCR C                           POP H
   JNZ ZAP5                        POP B
   POP B                           RET
                        NBYTE: CALL BYTE
```

ДОРАБОТКА ДЖОЙСТИКА =ПУ=

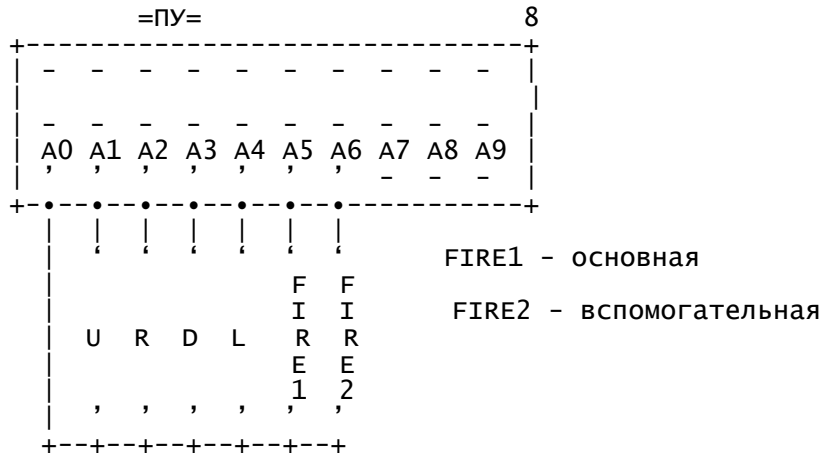
Многие пользователи БПЭВМ "Вектор-06Ц" уже подключили джойстик =ПУ=.

Его отличия от имеющихся типов джойстиков (Джойстик С, Джойстик П и аппаратный джойстик): простота подключения,не требующая переделки (вскрытия) компьютера; простота опроса; возможность работы с программами без переключения: джойстик или клавиатура, т.к. сканируется и клавиатура и джойстик (с преимуществом у клавиатуры).

Однако, у джостика есть один недостаток: во многих программах необходимы 2 клавиши "FIRE" (пробел).

Этот недостаток легко устранить, если доработать подключение джойстика.

Теперь подключение джойстика следует производить по следующей схеме (сравните со схемой в "Байте-3"):



Для ранее подключивших джойстик доработка потребует минимум усилий т.к. во всех, имеющихся в продаже,джойстиков имеются 2 клавиши "FIRE".

Все программы полностью совместимы с доработкой (при необходимости в них производится и опрос клавиши "FIRE2").

И в заключение для программистов:

Опрос джойстика производится через порт 07 [IN 07, INP(07)].

При этом: бит 2 - клавиша "FIRE2" (вспомогательная)
бит 3 - клавиша "FIRE1" (основная)
бит 4 - клавиша "влево"
бит 5 - клавиша "вниз"
бит 6 - клавиша "вправо"
бит 7 - клавиша "вверх"

Если бит равен 1, то клавиша нажата, если 0, то нет.

Приведенная ниже п/п предназначена для программистов, программирующих на Ассемблере. Обращение по CALL JOY. П/п не изменяет значения регистров, выходной параметр: в аккумуляторе (А) - код. Занимает 17 байт памяти.

```
JOY: PUSH B  ; сохраняем рег. пару BC          Выходной
     XRA A   ; обнуляем аккумулятор            параметр:
     OUT 07  ; обнуляем порт 07             в (A)-код
     IN 07   ; считываем число из порта 07
     MOV B,A ; сохраняем его в регистре B      бит: 0-"CC"
     IN 01   ; считываем число из порта 01      1-"УС"
     ANI 60H ; выделяем биты клавиш "УС" и "СС"     2-"FIRE2"
     RAL     ; передвигаем их на 3 разряда влево    3-"FIRE1"
     RAL                        4-"."
     RAL                        5-"№"
     ORA B   ; "накладываем" регистр B на аккумулятор   6-"¶"
     POP B   ; восстанавливаем рег. пару BC         7-"ё"
     RET     ; возврат из п/п
```

P.S. В центре "Байт" Вы можете приобрести программы,адаптированные под джойстик =ПУ= (Joystick & Keyboard). Они отмечены в каталоге - (J&K).

TMK Soft;г.Киров-1991

+-----------------------------------+
| Программное обеспечение для ком- |
| пьютера "ВЕКТОР-06Ц": игровые, |
| учебные и прикладные программы Вы |
| можете заказать по адресу: г.Киров |
| ул. Карла-Маркса, 126,центр "БАЙТ" |
| приемный пункт "Унисон". А иного- |
| родние могут сделать заказ по ад- |
| ресу: г.Киров, 610006, а/я 1248, |
| Зубкову Александру Николаевичу. |
+-----------------------------------+

+---+
| Если у Вас есть вопросы и проблемы, связанные с "ВЕКТОРом-06Ц", |
|предложения и объявления, материалы для статьи или заметки, напи- |
|Ќите нам. Мы сумеем Вам помочь. НаЌ адрес: г. Киров, 610006, |
|а/я 1248, центр "Байт". |
+---+

ОБЪЯВЛЕНИЕ

В связи с ростом налогов и затрат мы вынуждены поднять цены на все услуги, оказываемые центром.
На программы - в 2 раза. Кассеты (90 минут) - 50 руб. Почтовые расходы - 30 % от стоимости заказа.

Центр "БАЙТ"

ОБЪЯВЛЕНИЕ

Авторы статей и заметок, печатаемых в инфо-выпусках будут получать бесплатное программное обеспечение, обьем которого будет соответствовать обьему и содержанию статей.

Центр "БАЙТ"

НЕСКОЛЬКО ЗАМЕЧАНИЙ ПО ПОВОДУ "RETEX"

Приблизительно год назад у пользователей "Вектора-06Ц" появился текстовый редактор "Retex". Редактор имеет наплохие характеристики: небольшой объем, возможность создавать тексты до 4 кБт, полный знакогенератор, большие и легко различимые символы (8*12 точек), длина строки до 255 символов,возможность поиска и дублирования информации
Вместе с тем у редактора имеется и ряд недостатков. Во-первых,если Вы хотите воспользоваться управляющими клавишами, то Вам нужно каж-

дый раз возвращаться в режим: JCUKEN+ЛАТИНСКИЕ+БОЛЬШИЕ, а уже затем 2
нажимать АР2, символ. В частности,сразу же после загрузки нужно нажать: УС+СС, УС+РУС/LAT, УС+ПС и убедиться в том,что включен нужный режим, нажав "R". Если печатается большая латинская "R", то все нормально, если - нет, то значит какой-то из режимов переключился дважды. Во-вторых,если Вы хотите загрузить какой-то текстовый файл, то нужно предварительно набрать его имя путем АР2,N. Курсор окажется в правом верхнем углу. После набора имени лишние символы нужно заменить пробелами и нажать "ВК". Теперь можно приступать и к загрузке. Нажмите АР2,L. В левом нижнем углу появится ":SKIP".Включите МГ, внизу экрана появится имя файла. Если это имя не соответствует заданному, то загрузки не происходит, редактор ищет следующее имя. Если нужное имя найдено,то загорается ":LOAD" и начинается загрузка файла.В-третьих, для получения псевдографики нужно нажать АР2 и затем три цифры, т.е. чтобы получить один символ нужно нажать четыре клавиши. Это несколько многовато.

И в-четвертых, в редакторе применена нестандартная подпрограмма ввода байта. Она не проверяет вводимый файл на инверсию. Логика построена следующим образом: если файл в прямом коде, то он грузится, если же в инверсном,то нет.Однако во всех магнитофонах имеются усилители, причем большинство из них инвертируют сигнал. Если при записи на ленту сигнал был проинвертирован, а при считывании с ленты был проинвертирован опять, то никаких проблем не возникает. Если же инверсия была один раз, то "Retex" уже не будет считывать файл.Также возникают затруднения при работе с файлом, записанном на другом магнитофоне.

Чтобы исправить "Retex" нужно:

1. Загрузить "Retex" в Монитор (см. "Байт-2") через УС+ВВОД+БЛК.
2. Внести следующие изменения директивой "A адрес":

```
а). 0114 LXI H,0153   б). 0153 RZ         в). 1225 JMP 0003
0117 LXI D,0003       0154 CPI 19
011A MVI B,1A         0156 JNZ F21F
011C MOV A,M          0159 LDA 1C
011D STAX D       015C DCR A
011E INX H        015D ANI 1
011F INX D        015F STA 1C
0120 DCR B        0162 JZ 0018
0121 JNZ 011C         0165 DCR A
0124 JMP E200         0166 MVI A,3F
              0168 STA F256
              016B RET
              016C LXI B,0
```

3. Выгрузить исправленный вариант:

```
#ORETEX.COM
#O>1,24,0
```

Смысл исправлений в следующем. К подпрограмме поиска синхробайта (0E6H) мы добавляем кусок поиска инверсии синхробайта (19H). Если найден код 0E6H, то изменений не производится, т.к. файл грузится в прямом коде.

Если найден код 19H и это было в нечетный раз (1-ый, 3-ий ...), то по адресу 0F256H стирается команда СМС, что обеспечивает инверсию считываемой информации. Если же это было в четный раз (2-ой, 4-ый и т.д.),то команда СМС восстанавливается.

После внесения вышеописанных исправлений проблем с загрузкой текста у Вас больше не появится. Если же Вы не хотите сами заниматься переделкой, то можете заказать редактор "Retex" в центре "Байт".Те- 3
перь мы тиражируем только исправленную версию.

Центр "Байт"

ВНИМАНИЕ! НОВИНКА! ПРОГРАММА "Copy v(JPAS.1)"

С этого года центром "Байт" начато тиражирование копировщика программ в ROM, который при копировании программ написанных на Паскале или "Драйверах Устройств" пристыковывает к программам блок опроса джойстика =ПУ=. После копирования через этот копировщик программы работают от Джойстика =ПУ= или Клавиатуры. Этот копировщик позволит пользователям "ВЕКТОРа-06Ц", которые имеют программы на Паскале или Драйверах адаптировать их под джойстик не затрачивая дополнительных средств и усилий. К программе прилагается инструкция (в ROM).

TMK Soft, г.Киров-1992

ДЛЯ ПОЛЬЗОВАТЕЛЕЙ МИКРОДОС

Как выяснилось в последнее время, утилита SAVEMON.COM, предназначенная для вывода на МЛ файлов в формате Монитора, оказалась с ошибкой.Если начальный адрес выводимой программы имеет нулевой младший байт, то ошибки не происходит (например: 100, 3000 и т.д.). Если же младший байт начального адреса не равен 0, то программа выгружается с ошибкой. Суть ошибки в следующем.Утилита при расчете адреса конца использует только старшие байты, а младший всегда устанавливает в 0FFH. Если младший байт начального адреса - 0,то все нормально,если же нет, то длина задаваемая начальным и конечным адресами не будет соответствовать реальной длине программы. При загрузке утилита LOADMON.COM выдает ошибку, так как получит неверную контрольную сумму. Надо заметить, что сама программа все-таки выгружается правильно, неверен только адрес конца.Для исправления утилиты предлагаем следующее:

1. Загрузить в SID утилиту SAVEMON.COM:

```
C>SID SAVEMON.COM
```

2. Перебить команды с помощью директивы A:

```
#A033A

старое значение         новое значение

 033A ADD D          033A MOV H,A
 033B MOV H,A        033B MVI L,FF
 033C MVI L,FF       033D DAD D
```

3. Задать имя:

```
#ISAVEMON.COM
```

4. Записать измененную программу на диск:

```
    #W100,4FF
```

Теперь утилита будет работать правильно.
Кому не хочется перебить несколько команд,можно обратиться в центр "Байт". Теперь мы тиражируем исправленную утилиту.

Центр "Байт"

BASIC 2.5 (Продолжение)

Эта программа моментально рисует на экране кирпичную стену.Посмотрим, как определились значения параметров:

```
 ########     - 00000000 - M7=0
 #            - 01111111 - M6=127
 #            - 01111111 - M5=127
 #            - 01111111 - M4=127
 ########     - 00000000 - M3=0
 #            - 11110111 - M2=247
 #            - 11110111 - M1=247
 #            - 11110111 - M0=247
```

Оператор SCREEN 3 можно использовать для быстрого рисования небольших сложных изображений.

Использование 16-ричных значений

Для ввода и вывода значений в 16-ричном виде в BASICe v2.5 используются символы "&" и "@".Их применение проиллюстрируем на примерах:

```
+-------------------  Здесь мы вводим 16-ричное число 20H, а полу-
| 10 INPUT A:PRINT A  чаем его 10-чный эквивалент 32.
| RUN
| ?&20
| 32

+-------------------  Здесь мы записываем, начиная с адреса 8000H,
| POKE &8000,&FF,&F5  коды FFH,F5H.
|

+-------------------  Здесь десятичное число выводится в 16-ричном
| PRINT @255          представлении.
- FF

+-------------------  В этом примере показано выполнение смешанных
| 10 A=&FE            действий с выводом результата в 16-ричном
| 20 B=2              виде.
| 30 PRINT @(A+B)
| RUN
| 100
```

? ВОПРОСЫ ?

-Почему в Редакторе-Ассемблере не действует функция печати текста ?
-Какие программы для компьютера ВЕКТОР-06ц подходят к ВЕКТОРу-Старт 1200, на какие критерии ориентироваться в выборе программ ?

ОШИБКИ В BASICe 2.5 (ИВ-7)

1. Неверно работает оператор INPUT в случае ввода символьной строки. Если введенная строка или ее часть совпадает с одним из зарезервированных слов BASICa, она возвращается в виде кода внутреннего представления этого слова.
Например, попробуйте ввести следующюю небольшую программу:

```
10 INPUT A$
20 PRINT A$
```

Запустив программу, введите какое-нибудь зарезервированное слово, например, RENUM. Вместо него оператором PRINT будет напечатана совершенная чепуха. Хотя эту ошибку обойти совершенно невозможно,ввод таких строк бывает нужен разве что в обучающих программах.
Кроме того,оператор INPUT также неверно возвращает коды большинства специальных символов:

Ввод	Возврат	Ввод	Возврат
-	%	"	ПУСТО
/	'	&	I
:	ПУСТО	=	,
^	(	*	&
@	S	>	+
,	ПУСТО	<	-
+	$		

2. Как написано в документации, для печати крупным шрифтом достаточны оператор PLOT в режиме 2, LINE в режиме BS и PRINT.Однако такая последовательность может нормально работать 24 раза, после чего происходит скроллинг экрана на строку вверх. Чтобы избежать этого, надо вставлять в программу CUR 1,1.
3. Некорректно работает функция ADDR для переменных, числовых и символьных массивов. В случае символьных переменных выдается неверный адрес, использовать который дальше нельзя. Это серьезный недостаток BASICa v2.5, т.к. теряется возможность записи символьных данных на магнитную ленту.
Правда, с помощью некоторых программных ухищрений это все же можно сделать. Для этого необходимо в программе оператором HIMEM зарезервировать область памяти необходимой величины, а перед записью символьных данных перенести их побайтно в зарезервированную область в известном Вам порядке.
К примеру, надо записать на ленту символьный массив A$, состоящий из 10 элементов. Вот фрагмент программы для его записи на ленту.

```
10 HIMEM &74FF
№№№
100 A=&7500
110 FOR I=0 TO 9
120 W$=A$(I)
130 POKE A,ASC(W$):A=A+1
140 IF LEN(W$)=1 THEN 160
150 W$=RIGHT$(W$,LEN(W$)-1):GOTO 130
160 POKE A,0:A=A+1:NEXT
```

```
  170 BSAVE"DATA",&7500,A                                          6
```

Строка 100 задает начальный адрес зарезервированной области, в строке 120 очередная символьная строка копируется во вспомогательную символьную переменную W$.

4. Оператор RENUM часто работает некорректно, особенно в длинных программах: после перенумерации в операторах передачи управления (GOTO, GOSUB, ON..., IF...THEN) появляются несуществующие строки.За этим надо следить, просматривая программу после перенумерации. Обычно эти несуществующие номера очень велики (больше 65000), поэтому заметить их нетрудно. Более того, если при перенумеровании задан, к примеру, шаг 10,то эти несуществующие номера еще более заметны,т.к. не будут кратны 10.

В статье использованы материалы И.Шиповского, В.Князева, В.Краева

BASIC II

Раздел для начинающих

Бейсик шаг за шагом

```
 10 PRINT"*";   Попробуйте вместо точки с запятой  ввести  запятую
 20 GOTO 10     или вообще ничего не вводить.
```

```
 10 FOR A=1 TO 1050  Экран может содержать не более 1050 символов,
 20 PRINT"*";        т.к. 42*25=1050.
 30 NEXT A
```

```
 40 PAUSE 3          Добавим к предыдущей программе эту, выре-
 50 FOR S=5 TO 16    жем на экране окно. Отметим, что в строке
 60 PRINT AT 10,S;"    " 60 - 12 пробелов.
 70 NEXT S
```

```
 10 C=INT(15*RND(1))+1   Компьютер случайным образом выбирает цвет
 20 COLOR C,0,C      бордюра и цвет знаков от 1 до 15.
 30 PRINT"ВЕКТОР-06Ц";
 40 GOTO 10
```

```
 10 X=INT(256*RND(1))    Заполняем экран разноцветными точками.Ко-
 20 Y=INT(256*RND(1))    ординаты точек генерируются случайным об-
 30 C=INT(16*RND(1))     разом командой RND(1).
 40 COLOR C
 50 PLOT X,Y,1
 60 GOTO 10
```

```
 20 C=INT(16*RND(1))     Мозаика из разноцветных квадратов.
 30 COLOR C
 40 PRINT CHR$(127);
 50 GOTO 20
```

Если вы хотите, чтобы программа остановилась сразу после заполнения экрана, то допишите к этой программе следующие строки:

```
 10 N=0
 45 IF N=1050 THEN STOP
 47 N=N+1
```

```
 10 COLOR 15,5,0        Рисуем "лесенку", цвет  бордюра  и
 20 CLS             экрана можно выбрать любой.
 30 X=1
 40 FOR L=0 TO 24
 50 FOR C=1 TO 15
 60 COLOR C
 70 PRINT AT C+X,L;CHR$(127);
 80 NEXT C
 90 X=X+1
 100 NEXT L
```

```
 10 CLS              Заполнение экрана случайным обра-
 20 FOR N=1 TO 1000          зом,случайными символами (конфет-
 30 C=INT(RND(1)*16)         ти). Программа остановится  после
 40 X=INT(42*RND(1))         того, как выведет  на  экран 1000
 50 Y=INT(25*RND(1))         символов.
 60 A$=CHR$(INT(RND(1)*95)+32)
 70 COLOR C
 80 PRINT AT X,Y;A$
 90 NEXT N
```

```
 10 FOR F=0 TO 40   Бегущая буква. Буква пробегает только одну
 20 PRINT AT F,10;"A"   строку 10. Попробуйте заставить букву про-
 30 PAUSE 3         бежать другую строчку или несколько строк.
 35 BEEP .01,12
```

```
  40 PRINT AT F,10;" "
  50 NEXT F
```

```
  10 FOR C=0 TO 255  Горизонтальные цветные полосы.
  20 P$=" 32 пробела "
  30 COLOR C
  40 PRINT P$
  50 NEXT C
```

```
  10 FOR N=0 TO 15   Вертикальные цветные полосы.
  20 FOR C=0 TO 15
  30 COLOR N*16+C
  40 PRINT" ";
  50 NEXT C
  60 NEXT N
```

```
  10 FOR F=1 TO 30     Звук и цвет.
  20 Z=INT(RND(1)*25)-12
  30 C=INT(RND(1)*16)
  40 BEEP .1,Z
  50 COLOR 15,0,C
  60 NEXT F
```

```
  10 FOR X=12 TO 36 Звуки.
  20 BEEP .01,X-24
  30 BEEP .01,24-X
  40 NEXT X
```

```
  10 P=ASC(INKEY$)        Клавиатура - муз. инструмент.
  20 IF P=255 THEN 10
  30 BEEP .04,INT(P*3/16)-12
  40 GOTO 10
```

(Продолжение следует)

НОВИНКИ

(цифры в скобках означают оценку игровой программы по 10-ти балльной шкале)

Copy-Z2, Copy-N4, Rbase.COM, Super Calc-2.COM, Word Master.COM, Word Star.COM, Run.COM, BT.COM, Дизассемблер.MON, Basic-M.ROM, Attr.COM, L80.COM, Lib.COM, Load.COM, M80.COM, Stat.COM, Hdir.COM, Эл.голос.ROM, ЦМУ-1,2.ROM, Basic-1.3.ROM, Putup-3,3.1 (10).ROM, Lode runner (9).ROM, Ниньзя.BAS (6), DEASH STAR.BAS (6), Фруктовая машина.BAS (7), Step & Jump.ROM (7), Runner.ROM (4), Space Kommander.ROM (6), Popcorn.ROM (5), Замок.ROM (5), Сокровища.ROM (4), Рэмбо.ROM (6), Каратэ.ROM (5), Шарик.ROM (5), Водолаз.ROM (5), Вор.ROM (5), Охота.ROM (5), Тир-3.ROM (6).

BASIC II
Раздел для начинающих

Бейсик шаг за шагом

```
  10 FOR N=85 TO 0 STEP -2  Имитатор прыгающего шарика.
  20 BEEP .01,5
  30 FOR A=N TO 0 STEP -1
  40 NEXT A:NEXT N

  10 FOR N=1 TO 25
  20 P=INT(RND(1)*40)-35
  30 BEEP .05,P
```

```
  40 BEEP .05,P+7
  50 BEEP .05,P+4
  60 NEXT N
```

Далее идут программы, основой которых является исполнение операторов PLOT, LINE, CIRCLE.

```
  10 COLOR 4           Рамка.
  20 PLOT 50,50,1
  30 LINE 205,205,BF
  40 COLOR 0
  50 PLOT 70,70,1
  60 LINE 185,185,BF
  70 COLOR 7
```

```
  10 FOR F=0 TO 31             Рамка.
  20 PRINT AT F,0;CHR$(127)
  30 PRINT AT F,21;CHR$(127)
  40 NEXT F
  50 FOR F=0 TO 21
  60 PRINT AT 0,F;CHR$(127)
  70 PRINT AT 31,F;CHR$(127)
  80 NEXT F
```

```
  10 R=INT(RND(1)*120)    Цветные окружности.
  20 C=INT(RND(1)*16)
  30 CIRCLE 128,128,R
  40 GOTO 10
```

```
  10 X1=INT(RND(1)*120)   Вложенные прямоугольники.
  20 Y1=X1/2
  30 X2=255-X1
  40 Y2=255-Y1
  50 PLOT X1,Y1,1
  60 LINE X2,Y2,B
  70 GOTO 10
```

```
  10 FOR X=0 TO 124 STEP 4   Колодец.
  20 PLOT X,X,1
  30 LINE 255-X,255-X,B
  40 NEXT X
```

```
  10 FOR Y=0 TO 255 STEP 4     Лучи.
  20 PLOT 255,255,1:LINE 0,Y
  30 PLOT 0,255,1
  40 LINE 255,Y
  50 NEXT Y
  60 FOR X=0 TO 255 STEP 4
  70 PLOT 255,255,1:LINE X,255
  80 PLOT 0,255,1:LINE X,0
  90 NEXT X
```

```
  10 FOR X=0 TO 255 STEP 4       Косая штриховка.
  20 PLOT X,0,1:LINE 255,255-X
  30 PLOT 0,Y,1:LINE 255-X,255
  40 NEXT X
```

BASIC III

Маленькие хитрости

Если у Вас еще небольшой опыт работы с "ВЕКТОРом-06Ц", то эти маленькие хитрости могут Вам доставить удовольствие. Считайте их головоломками. Давайте сделаем так. Сначала попробуйте угадать, что

будет напечатано по команде PRINT в приведенных примерах. Потом проверьте себя на компьютере и попробуйте объяснить почему так происходит. Если не сумеете - не беда. Ниже мы дадим объяснение и расскажем как все это можно применять в программах.

```
1.10 A=5:B=8
  20 X=10
  30 X=X+(A>B)
  40 PRINT X
```

```
2.10 A=15:B=8
  20 X=10
  30 X=X+(A>B)
  40 PRINT X
```

```
3.10 A=15:B=8
  20 X=A=B
  30 PRINT X
```

```
4.10 X-0
  20 X=NOT X
  30 PRINT X
```

```
5.10 X=1
  20 X=NOT X
  30 PRINT X
```

```
6.10 X=10
  20 X=NOTNOTX
  30 PRINT X
```

```
7.10 A=5:B=8
  20 X=(A=B)=1
  30 PRINT X
```

```
8.10 A=5:B=8
  20 X=(A=B)=0
  30 PRINT X
```

Все приведенные примеры используют особенность "ВЕКТОРа-06Ц" выполнять логические вычисления. Сейчас мы подробно разберем пути использования этих способностей компьютера в Ваших программах.
Рассмотрим, например, такой фрагмент какой-либо программы:

```
100 IF INKEY$="8" THEN X=X+1
110 IF INKEY$="5" THEN X=X-1
```

Такая логика часто встречается, когда Вы хотите, например, чтобы при нажатии клавиши "8" объект двигался по экрану вправо, а при нажатии клавиши "5" - влево. Занимает такой фрагмент 38 байт и далеко не оптимален.
Рассмотрим другой пример:

```
100 A$=INKEY$:X=X+(A$="5")-(A$="8")
```

Это совершенно то же самое, но занимает 32 байта и, самое главное, намного быстрее. Если же сделать так:

```
100 A=ASC(INKEY$):X=X+(A=53)-(A=56)   ,
```

то можно еще сократить объем программы и ускорить ее выполнение. Вся хитрость в том,что те выражения,которые записаны внутри скобок, являются логическими.Если результат логического отношения-"истина", то логическое выражение имеет значение 1, а если "ложь" - 0.
Т.к. функция INKEY$ не может считывать две клавиши одновременно, то либо была нажата клавиша "8", либо "5", либо ни одна из них.Тогда либо первая скобка равна -1, либо вторая, либо ни одна.Фактически мы в этом примере прибавим к числовой переменной X результат логического выражения. Надо сказать, что не всякий язык программирования позволил бы такую вольность, да и Бейсик не всякий, но Бейсик "ВЕКТОРа" позволяет.
Продолжим с нашим примером. При изображении на экране движущегося объекта бывает важно так ограничить его перемещение,чтобы он не вышел за пределы экрана. Обычно это делают так:

```
120 IF X>31 THEN X=31
130 IF X<0 THEN X=0
```

Все можно сделать лучше и проще путем:

```
100 A$=INKEY$:X=X-(A$="8"AND X<31)+(A$="5"AND X>0)
```

или

```
100 A=ASC(INKEY$):X=X-(A=56 AND X<31)+(A=53 AND X>0)
```

Другой пример:

```
10 X=A=B
```

На первый взгляд в этой записи мало смысла,но это только на первый взгляд. Здесь ясно видны два этапа равенства,а вот означают они совершенно разные вещи. Первый - оператор присваивания, т.е. он присваивает переменной X результат логического выражения A=B. А второй этап равенства - это логическое отношение "Равно" и, тем самым, является частью логического выражения A=B,которое может быть либо истинным, либо ложным и, соответственно, иметь в качестве результата либо 0, либо -1. Вся запись заняла 11 байт, в то время как его традиционный эквивалент имел бы 30 байт:

```
10 IF A=B THEN X=-1
20 IF A<>B THEN X=0
```

А если надо, чтобы при A=B, X равнялся бы десяти, тогда так:

```
10 X=-(A=B)*10
```

В программах такую систему иногда используют для организации подсчета очков. Например, Вам надо, чтобы при правильном ответе (при нажатии заданной клавиши) к счету прибавлялось бы 50 очков:

```
10 S=S-(INKEY$="A")*50
```

Очень часто в программах для создания мультипликационных эффектов используют быстрое чередование двух каких-то символов,например псевдографики. Первый, допустим, изображает бабочку со сложенными крыльями, а второй - с расправленными. Удобнее всего для этого пользоваться оператором NOT.

```
510 A=0
520 A=NOT A
530 PRINT AT 10,15;CHR$(144+A)
540 IF INKEY$<>"" THEN RETURN
550 GOTO 520
```

Цикл между операторами 520 и 550 будет продолжаться до тех пор,пока Вы не нажмете какую-либо клавишу и не выйдите через строку 540 в вызывающую подпрограмму.

Следующий пример хоть и не дает такой ощутимой экономии памяти,как большинство предыдущих, зато демонстрирует оригинальный прием организации переключателей в программе и,вообще, открывает доступ новым идеям:

```
10 IF R=1 THEN PRINT"A"
20 IF R=2 THEN PRINT"B"
30 IF R=3 THEN PRINT"C"
```

Можно записать так:

```
  10 PRINT CHR$((65 AND R=1)+(66 AND R=2)+(67 AND R=3))
```

 Кроме того, можно использовать оригинальное действие функции NOT. Выражение NOT X будет равно -X-1. Т.е. NOT 0=-1, NOT 1=-2 и т.д.

Март БАЙТ-10
КИРОВ информационно-рекламный выпуск для
1992 г. пользователей БПВМ «ВЕКТОР-06Ц»

Центр «Байт» предлагает документацию для пользователей ПК «ВЕКТОР-06Ц»!

1. Бейсик-Корвет — 20 руб.
2. Паскаль-компилятор — 20 руб.
3. Драйверы устройств — 20 руб.
4. Байт-91 (сборник статей за 1991 г.).

Документация отпечатана типографским способом.

Программное обеспечение для компьютера «ВЕКТОР-06Ц»: игровые, учебные и прикладные программы Вы можете заказать по адресу: г. Киров, ул. К. Маркса, 126, центр «БАЙТ», приемный пункт «Унисон». А иногородние могут сделать заказ по адресу: г. Киров, 610006, а/я 1248, Зубкову Александру Николаевичу.

Если у Вас есть вопросы и проблемы, связанные с «ВЕКТОРом-06Ц», предложения и объявления, материалы для статьи или заметки, напишите нам. Мы сумеем Вам помочь. Наш адрес: г. Киров, 610006, а/я 1248, центр «Байт».

ВНИМАНИЕ! НОВИНКА! ПРОГРАММА «COPY v (JPAS. 1)»

С этого года центром «Байт» начато тиражирование копировщика программ в ROM, который при копировании программ, написанных на Паскале или «Драйверах устройств», пристыковывает к программам блок опроса джойстика -ПУ,-. После копирования через этот копировщик программы работают от джойстика -ПУ- или Клавиатуры. Этот копировщик позволит пользователям «ВЕКТОРа-06Ц», которые имеют программы на Паскале или Драйверах, адаптировать их под джойстик,, не затрачивая дополнительных средств и усилий. К программе прилагается инструкция (в формате ROM). Если при эксплуатации этого копировщика у Вас возникнут какие-либо проблемы или вопросы, просим написать в наш центр «Байт» (с пометкой «COPY-FD» и вложить пустой конверт с обратным адресом для ответа).

ТМК Soft, г. Киров—1992.

ВНИМАНИЕ! НОВИНКА! ПРОГРАММА «COPY v (FORDOS)»

С этого года центром «Байт» начато тиражирование копировщика программ в ROM, который при копировании адаптирует программы для запуска из МикроДОС. При работе в МикроДОС пользователи получают неоспоримые удобства и преимущества. Но до сих пор запуск некоторых программ из МикроДОСа был затруднен (программы, начинающиеся не с 1-го блока, т. к. при запуске программы МикроДОС гру_ зит ее с квазидиска (дисковода) в область транзитных программ [ОТП], начинающуюся с адреса 01 ООН и запускает). Этот недостаток устраняет «COPY v (FD)», который при копировании адаптирует программы под 0100Н адрес. Приобретя этот копировщик, Вы затратите минимум усилий и средств для переработки программ для МикроДОС. К программе прилагается инструкция (в формате ROM). Если при эксплуатации этого копировщика у Вас возникнут какие-либо проблемы или вопросы, просим написать в центр «Байт» (с пометкой «COPY-FD» и вложить пустой конверт с обратным адресом для ответа).

ТМК Soft, г. Киров —1992.

BASIC II

Раздел для начинающих. Бейсик шаг за шагом (продолжение).

```
10 CLS: COLOR7                  Эта программа позволит Вам
20 PLOT 128, 128, 1             самим рисовать на экране.
30 X=4
40 V=ASC(INKEY$): IF V=255 THEN 40
50 IF V=8 THEN LINE STEP -X,0
60 IF V=24 THEN LINE STEP X,0
70 IF V=25 THEN LINE STEP 0,Y
80 IF V = 26 THEN LINE STEP 0, -Y
90 GOTO 40
```

```
10 FOR F = 32 TO 127            Вывод символьного набора.
20 PRINT CHR$(F);" ";
30 NEXT F
```

```
10 INPUT A                      Вывод символа по его номеру.
20 PRINT A, CHR$(A)
30 GOTO 10
```

```
10 FOR A=32 TO 127              Вывод таблицы кодов.
20 PRINT A, CHR$(A)
30 NEXT A
```

```
10 INPUT "X="; X                Увеличенные символы.
20 INPUT "Y="; Y
30 PLOT 50, 50, 2
40 INPUT "СИМВОЛ=";A$
50 LINE X, Y, BS:CLS
60 PRINT A$
70 GOTO 10
```

Зачерчивание экрана линиями:

```
10 COLOR 7, 0, 2
20 FOR I=0 TO 50
```

Построение вертикальных линий:

```
30 X=INT(RND(1)*256)
40 PLOT X, 0, 1:LINE X, 255
150 NEXT I
```

Добавив следующую часть программы, мы будем строить и горизонтальные линии:

```
50 Y=INT(RND(1)*256)
60 PLOT 0, Y, 1 : LINE 255,Y
```

```
10 B=INT(RND(1)*30)             Прыгающий НЛО.
20 A=INT(RND(1)*18)
30 PRINT AT B+1, A; "O"
40 PRINT AT B, A+1; "<8>"
50 PRINT AT B+1, A+2; "H"
60 PAUSE 10
70 CLS
80 PRINT AT B+1, A+1; "O"
90 PRINT AT B, A+2; "<8>"
100 PRINT AT B+1, A+3; "H"
110 PAUSE 10
120 CLS
130 GOTO 10
```

Попробуйте после строки 70 ввести:

```
75 GOTO 10
```

```
10 A$=" "                        Узор.
20 FOR X=1 TO 7
30 As=A$+CHR$(INT(RND(1)*14+129)
40 NEXT X
50 COLOR INT(RND(1)*16, INT(RND(1)*7), 0
60 FOR N = 0 TO 87
70 PRINT A$
80 NEXT N
```

```
10 CLS                           Мозаика.
20 H=16
30 V=11                          Попробуйте присвоить
40 X=INT(RND (1)*3–1)            строке 70 номер 35,
50 Y=INT(RND(1)*3–1)             а строку 70 убрать
60 COLOR INT(RND(1)*7)
70 FOR Z=1 TO 20
80 PRINT AT H,V;CHR$(143)
90 H=H+X
100 V=V+Y
110 IF H<0 THEN H=31
120 IF H>31 THEN H = 0
130 IF V<0 THEN V=21
140 IF V>21 THEN V = 0
150 NEXT Z
160 GOTO 20
```

Далее все программы — «звучащие», это варианты звучания оператора BEEP.

```
10 FOR Y=–2 TO 4                 Звуки.
20 FOR X = 0 TO 6                Измените .02 на .2 или .002
30 BEEP .02, X+Y
40 NEXT X:NEXT Y
```

```
10 FOR X = –12 TO 12             Полный диапазон звуков
20 BEEP .02, X                   Вашего компьютера.
30 NEXT X
```

```
10 FOR A=0.01 TO .1 STEP .01
20 FOR B=12 TO –12 STEP –4
30 BEEP .15–A, B
40 NEXT B:NEXT A
```

```
10 FOR N=1 TO 10                 Имитатор сирены. Обратите
20 BEEP .15, 5                   внимание на разницу в наборе
30 BEEP 0.15, 0                  первой цифры в 20 и 30 стро-
40 PAUSE 3                       ках (варианты эквивалентны).
50 NEXT N
```

Далее в печатном выпуске повторяется содержимое «Маленьких хитростей» из выпуска №9, повтор в настоящем издании опущен. — *прим. ред.*

Операционная система CP/M

В настоящее время на «ВЕКТОРе-06Ц» с использованием квазидиска или дисковода применяется операционная система МикроДОС. МикроДОС является адаптированным вариантом операционной системы CP/M и имеет с ней много схожего. Так как парк компьютеров с квазидиском и дисководом

расширяется, мы решили дать вводную статью по системе CP/M с некоторыми комментариями и примечаниями, применительно к МикроДОС. Статья в основном дана по книге «Компьютеры» под ред. Хелмса, перевод с английского. Итак...

Существует операционная система, фактически ставшая стандартной для микрокомпьютеров. Это система CP/M, разработанная в 1975 г. фирмой Digital Research, Inc. Система CP/M используется в вычислительных системах на базе микропроцессоров Z80, 8080 и 8085, а после переделки ее на 16-разрядный вариант может применяться и для микропроцессоров 8086.

Поскольку система CP/M является первой универсальной операционной системой, созданной для микропроцессоров, ее создание ликвидировало, наконец, этот явный пробел и привело к широкому ее рас. пространеиию. В течение нескольких лет система CP/M была единственной универсальной системой и быстро приобрела статус стандарта. В то же время она подверглась ряду модификаций и усовершенствований. Несмотря на большую популярность, эта система обладает рядом недостатков.

Тем не менее, поскольку на многих микропроцессорах используется система CP/M, охарактеризуем се более подробно. Дальнейшее изложение можно рассматривать как общее введение в микрокомпьютерные операционные системы, поскольку концепции CP/M во многом свойственны и другим операционным системам.

Введение в CP/M

Система CP/M представляет собой микрокомпьютерную операционную систему, предназначенную для обслуживания одного пользователя в однозадачном режиме. Как правило, система CP/M располага,-ется в старших адресах памяти и делится на три основных модуля: BIOS (Basic Input/Output System) — базовую систему ввода/вывода, BDOS (Basic Disk Operating System) — базовую дисковую операционную систему и CCP (Console Command Processor) — процессор команд, поступающих с консоли. Память, используемая задачей пользователя, называемая TPA (Transient Program Area), размещается с. младших адресов доступного адресного пространства. Рассмотрим каждую из частей поподробнее.

Модуль BIOS представляет собой аппаратно-зависимый модуль, выполняющий функции интерфейса между аппаратурой микрокомпьютерной системы и BDOS. Он является единственным модулем, зависящим от спецификаций аппаратуры и содержит подсистемы для ввода/вывода символьной информации и для управления дисками. Этот модуль обеспечивает такие функции, как определение адреса для прямого доступа к памяти, размещение дорожек и секторов и управление чтением-записью данных.

(Продолжение следует).

ДЛЯ ПОЛЬЗОВАТЕЛЕЙ МИКРОДОС

Ошибки в системе

При различных операциях обмена с диском могут возникнуть ошибки. При этом на дисплей выдается сообщение следующего формата:

```
ОШИБКА БДОС НА X: [имя файла] тип ошибки
OP=NN   PC=AAAA
[ST=SS  T=TT  S=YY]
[ИГНОРИРОВАТЬ (Y/N)],
```

где тип ошибки: - сбой диска;
- выбор (выбор диска, не поддерживаемого БСВВ);
- только для чтения;
- системный;
- неявное имя;
- файл есть.

NN - указывает десятичный номер операции БДОС, при которой произошла ошибка

AAAA - указывает адрес команды,куда возвращается управление после использования операции БДОС

SS - байт состояния диска

TT - номер дорожки

YY - номер сектора

В МикроДОС инеются два режима обработки ошибок:

ПО УМОЛЧАНИЮ (устанавливается после загрузки или "горячего старта") - управление возвращается вызывающей программе только при отсутствии ошибок, иначе вызывающая программа прерывается, выводится сообщение об ошибке и производится "горячий старт".

В режиме ВОЗВРАТ ОБРАБОТКИ ОШИБОК при обнаружении ошибки управление возвращается вызывающей программе с признаком ошибки выполнения.

Байт состояния для ошибок:

```
 0 - все нормально
 1 - удаленная запись
 2 - ошибка КС (CRC) данных
 4 - ошибка поиска дорожки
 8 - ошибка адреса
 A - ошибка CRC заголовка сектора
 E - не найден адресный маркер
 F - не найден маркер данных
10 - сбой в аппаратуре
20 - защита записи
40 - ошибка записи
80 - диск не готов
```

"ПЛАНЕТА ПТИЦ"

Игра "Планета птиц" при инициализации своей работы перемещается на 400H адресов вперед.
В ячейках:

04B2H - текущее значение очков в HEX-виде
04B3H

04B4H - текущее значение топлива в HEX-виде
04B5H

Установка этих ячеек производится так:

```
05EB LXI  H,001E
05EE SHLD 04B2
05F1 LXI  H,00FA
05F4 SHLD 04B4
```

Но так как в игре есть смещение (при инициализации), то при загрузке этот код находится в ячейках 05EBH+400H=09EBH.
Поэтому,чтобы в начале игры топливо устанавливалось в 9999(270FH), Вам нужно записать в ячейку 09F2H значение 0FH, а в ячейку 09F3H - значение 27H.
Для установки очков в 80 (50H) Вам надо записать в ячейку 09ECH значение 50H.
Но даже сделав все эти действия, добиться победы в игре довольно сложно, так как быстро исчезают набранные очки за пропуск "врагов". Это производится кодом:

```
0BD5 LHLD 04B2
0BD8 DCX H
0BD9 DCX H
0BDA DCX H
0BDB DCX H
0BDC DCX H
0BDD SHLD 04B2
```

Так как программа смещается, то при загрузке этот код находится в ячейках, начиная с 0BD5H+400H=0FD5H.
Для того,чтобы за пропуск "врагов" очки не вычитались не по 5,а по 4, надо заменить в любой из ячеек от 0FD8H до 0FDCH значение 2BH на 00H.
Если Вы замените значение всех ячеек на 00H, то очки вообще не будут вычитаться.
Если Вы вместо 00H поставите 23H, то очки не только не будут вычитаться, а будут прибавляться за пропуск "врагов".
Все эти изменения можно компановать так, как Вам нравиться, создавая игру на любой вкус.

Саттаров Виктор, г.Киров-1992

STEP & JUMP

В игре, управляя человечком и перепрыгивая с квадрата на квадрат, нужно дойти до конца уровня. Всего уровней 15.
Если же человечек на краю квадрата или над пропастью, то он погибает. Прыгать человечек может через один квадрат. На некоторых квадратах есть рисунок и надпись. Вот их обозначение:

DEATH - если сюда ступить, то человечек погибнет
SPEED - увеличивается скорость передвижения
BONUS - прибавляется 10 очков
JUMP - Вы можете один раз перепрыгнуть на неограниченное расстояние (в пределах экрана) до первого попавшегося квадрата. Если же не встретится ни одного квадрата, то человечек погибает.
LIFE - прибавляется одна жизнь
FLY   - смена уровня

STAGE END - конец уровня

За взятие любого такого квадрата, кроме STAGE END, еще прибавляется 5 очков.
По экрану иногда движется монстр, который может убить, столкнувшись с Вами. За проход одного уровня прибавляется 514 очков. Так что Вам будет очень трудно пройти все уровни, не обретя бессмертия.
Для этого следует сделать следующее:
1). Сдвинуть программу с помощью команды Монитора "M",так как сама программа располагается с 800H адреса (рабочие адреса программы находятся с 100H по 3585H адреса). С 100H по 700H находится заставка центра "Байт", а с 700H по 800H надпись "УСПИД".
Пример: M800,5000,100
2). Программа зашифрована в нерабочую форму. С адреса 3570H располагается дешифровщик программы. Нужно записать по адресу 3583H JMP F815. А после запустить дешифровщик.После того, как Монитор напечатает цифру 0B, программа будет расшифрована. Теперь программу можно будет просмотреть через команду "L".
3). Сейчас нужно изменить содержимое ячеек 184H и 33FH на 00H:

```
  было     стало
184 DCR A 184 NOP
33F INR A 33F NOP
```

Записать по ячейкам 188H-18AH 00H:

```
  было     стало
188 JZ 9F4    188 NOP
      189 NOP
      18A NOP
```

4). И самое последнее. Чтобы программа в процессе работы снова не зашифровала себя, измените содержимое ячеек 100H-102H через команду "A":

```
  было     стало
100 JMP 3570  100 JMP C90
```

Теперь запишите программу: O>1,35,0
Тем, кто не интересуется кодами, скажу, что "S"+"T" (когда картинка еще не двигается) смена уровня.

P.S. Если у кого-то программа "STEP AND JUMP" без заставки центра "Байт", то нужно начать со 2-го пункта.

Пантелеев Р.Е., Киров

ROCK MAN

Если Вы записали игру "Rock Man", но не знаете как играть, вот ее инструкция:

Управляя человечком и собирая алмазы, нужно пройти 31 этаж.Во время игры Вам будут мешать "жители" этажей - бабочки, шарики, квадраты. При столкновении с ними Вы взрываетесь и теряете жизнь.
Вот основные элементы этажа:

1). Зеленая стенка - человечек может пройти через нее, а "жители" этажей нет. Там, где человечек прошел, стенка исчезает.
2). Камни - человечек может двигать один камень нажатием клавиш "ЗБ"+"</>". Если камень находится на краю, то он падает. Если человечек стоит под падающим камнем, то он погибает, как и все другие "жители" этажа.
3). Алмазы - с помощью их человечек набирает очки. Один алмаз - одно очко.
4). Красная стенка - непроходима для человека.Но если камень упадет на "жителя" этажа,находящегося около стенки,то он "взрывается", после чего часть стены рядом с ним пропадает.
5). Мигающая стенка - проходя через нее, камни превращаются в алмазы, а алмазы - в камни.
6). Разрастающаяся стенка - через некоторое время, если она огорожена со всех сторон камнями или стенками (кроме зеленой),она превращается в алмазы. если не огорожена - в камни.
7). Выход на следующий этаж (похож на игральный кубик) - открывается после мерцания фона.

"Жители" этажа:

1). Квадрат - двигается по периметру свободного пространства.
2). Бабочка - передвигается, как квадрат.
3). Шарик - передвигается в произвольном направлении, отскакивая от препятствий.

При попадании камня (алмаза) в бабочку или шарик, они превращаются в 9 алмазов.

В этой игре есть редактор этажей.Вход в него осуществляется из заставки нажатием клавиш "УС+СС+РУС/LAT". Перемещения курсора производится стрелками. выбор элемента этажа производится нажатием клавиш "ЗБ"+">/<". Ннажимая несколько раз эти клавиши, Вы можете выбирать любой элемент этажа. Ставятся элементы нажатием клавиши "ЗБ". Переход на следующий этаж: "ЗБ"+"^/V". Стрелка вверх листает этажи назад, стрелка вниз - вперед.
Выход из редактора или игры "РУС/LAT".
И в конце скажу: во время игры есть возможность смены этажа с помощью клавиш "УС"+"РУС/LAT".

Зубарев, г. Киров

СТАНДАРТНЫЕ ПОДПРОГРАММЫ ВВОДА-ВЫВОДА В ФОРМАТЕ EDASM.COM

Физический формат вывода на МЛ: 64 раза по 55H, 64 раза по 0, 64 раза по 55H, 64 раза по 0, 5 раз по 0E6H, имя программы, 256 раз по 0, 0E6H, инвертированный младший байт длины файла, инвертированный старший байт длины файла, байты программы, 0FFH, младший байт

КС, старший байт КС.

```
;П/П ЗАГРУЗКИ В ФОРМАТЕ EDASM
ORG 100H
I: DI
 XRA A   ;обнуляем признак ошибки
 STA ERR
 CALL I4 ;ввод заголовка
 PUSH H
I1:CALL IBYTE
 MOV B,A
 LDA FLAG ;выбор режима
 INR A    ;"LOAD" или "VERIFY"
 JNZ I2
 MOV A,B
 CMP M
 JNZ ERROR
I2:MOV M,B
 INR B    ;поиск конца файла (0FFH)
 INX H
 JNZ I1
 DCX H
 MVI A,8
 CALL I10 ;ввод кс
 POP H
 PUSH B
 CALL KS  ;подсчет кс
 XTHL
 MOV D,B
 MOV E,C
 CALL SRAV
I3:JNZ ERROR;если кс1 и кс2
 POP H    ;не равны, то оНибка
 EI
 RET
;п/п ввода заголовка
I4:CALL I6  ;поиск нужного имени
 JNZ I4
 CALL I9  ;ввод длины текста
 LHLD NACH
 LDA FLAG ;выбор режима "LOAD"
 DCR A    ;или "MERGE"
 JM I5
 LHLD NACH1
I5:XCHG      ;проверяем, войдет ли
 LXI H,KON ;загружаемый файл
 DAD B
 XCHG
 CALL SRAV
 RC
ERROR:POP H ;генерируем ошибку
 MVI A,1
 STA ERR
 EI
 RET
;п/п сравнения HL и DE
SRAV:MOV A,H
 CMP D
 RNZ
  ;загрузки имени тек.файла
I6:LXI H,NAME2
CALL I11;ищем 5 синхробайтов
I7:CALL IBYTE
 MOV M,A
 ORA A
 INX H
 JNZ I7
 LXI H,NAME2;сверяем с за-
 LXI D,NAME1;данным именем
I8:LDAX D
 ORA A
 RZ
 CMP M
 INX H
 INX D
 JZ I8
 RET
I9:MVI A,0FFH
I10:CALL IBYTE1
 MOV C,A
 CALL IBYTE
 MOV B,A
 RET
;п/п поиска 5 синхробайтов
I11:MVI B,4
 MVI A,0FFH
I12:CALL IBYTE1
 CPI 0E6H
 JNZ I11
 DCR B
 MVI A,8
 JNZ I12
 RET
;п/п подсчета кс
KS:LXI B,0
KS1:MOV A,M
 CPI 0FFH
 RZ
 ADD C
 MOV C,A
 MVI A,0
 ADC B
 MOV B,A
 INX H
 JMP KS1
;п/п ввода байта
IBYTE:MVI A,8
IBYTE1:PUSH B
 PUSH D
 MVI C,0
 MOV D,A
 IN 01
 ANI 10H
 MOV E,A
IBYTE2:IN 1
 ANI 10H
```

```
 MOV A,L                          CMP E
 CMP E                            JZ IBYTE2
 RET                              RLC


 RLC
 RLC
 RLC
 MOV A,C
 RAL
 MOV C,A
 PUSH PSW
 LDA CONR
 MOV B,A
 POP PSW
IBYTE3:DCR B
 JNZ IBYTE3
 IN 1
 ANI 10H
 MOV E,A
 MOV A,D
 ORA A
 JP IBYTE5
 MOV A,C
 CPI 0E6H
 JNZ IBYTE4
 XRA A
 STA PRINV
 JMP IBYTE6
IBYTE4:CPI 19
 JNZ IBYTE2
 MVI A,255
 STA PRINV
IBYTE6:MVI D,9
IBYTE5:DCR D
 JNZ IBYTE2
 LDA PRINV
 XRA C
 POP D
 POP B
 RET
CONR:DB 4BH;константа чтения
PRINV:DB 0 ;признак инверсии канала
FLAG:DB 1;режим:0FFH-"VERIFY",0-"LOAD",1-"MERGE"
NACH:DW 2000H;адрес начала текстового буфера
NACH1:DW 4000H;адрес,определяющий место загрузки при "MERGE"
ERR:DB 0;признак ошибки (1-ошибка)
KON:EQU 8000H;макс. адрес конца текста
NAME1:DB 'LOADASM',0 ;шаблон имени для загрузки/выгрузки
NAME2:DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;вводимое имя с МЛ
;П/П ВЫГРУЗКИ ФАЙЛА В ФОРМАТЕ EDASM
O:DI
 LHLD NACH
 CALL KS   ;подсчет кс
 CALL O8   ;определение конца текста
 PUSH B
 PUSH D
 CALL O1   ;вывод преамбулы
 XTHL
```

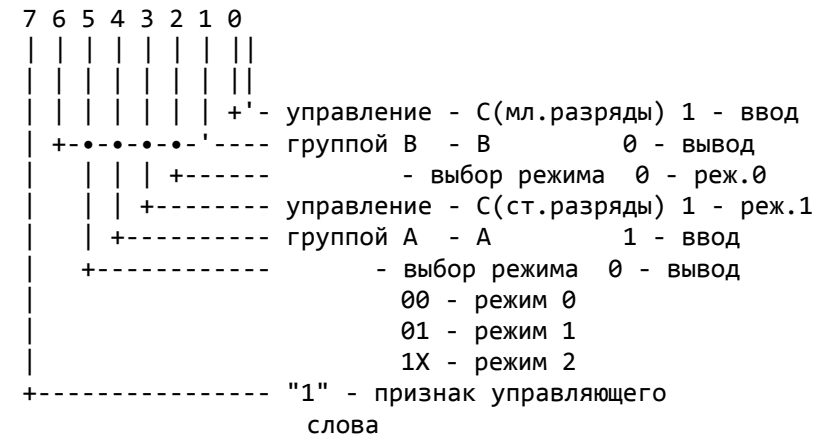
```
 XCHG
 LXI H,NAME1
 CALL O5   ;вывод 5 синхробайт и имени
 CALL O3   ;вывод нулевых байтов
 POP D
 LHLD NACH ;определяем длину текста


 MOV A,H
 CMA
 MOV H,A
 MOV A,L
 CMA
 MOV L,A
 INR L
 DAD D
 MVI A,0E6H;выводим инвертированную
 CALL OBYTE;длину на МЛ
 MOV A,L
 CMA
 CALL OBYTE
 MOV A,H
 CMA
 LHLD NACH ;выводим текст
 CALL O7
 POP B
 MOV A,C   ;выводим кс
 CALL OBYTE
 MOV A,B
 CALL OBYTE
 EI
 RET
;п/п вывода преамбулы
O1:MVI D,4
 XRA A
O2:MVI E,40H
 XRI 55H
 CALL O4
 DCR D
 JNZ O2
 RET
;п/п вывода 256 нулевых байт
O3:XRA A
 MOV E,A
O4:CALL OBYTE
 DCR E
 JNZ O4
 RET
;п/п вывода 5 синхробайт и инф.байт
O5:MVI A,0E6H
 MVI B,4
O6:CALL OBYTE
 DCR B
 JNZ O6
O7:CALL OBYTE
 CALL SRAV
 MOV A,M
 INX H
 JNZ O7

;п/п вывода байта на МЛ
OBYTE:PUSH B
 PUSH D
 PUSH PSW
 MOV D,A
 MVI C,8
OBYTE1:MOV A,D
 RLC
 MOV D,A
 MVI A,1
 XRA D
 ANI 1
 OUT 0
 CALL OBYTE2
 MVI A,0
 XRA D
 ANI 1
 OUT 0
 CALL OBYTE2
 DCR C
 JNZ OBYTE1
 POP PSW
 POP D
 POP B
 RET
OBYTE2:PUSH PSW
 LDA CONZAP
 MOV B,A
 POP PSW
OBYTE3:DCR B
 JNZ OBYTE3
 RET
CONZAP:DB 32H;константа
              ;записи
```

```
 JMP OBYTE
;п/п определения конца текста
O8:LXI D,NAME1
O9:LDAX D
 ORA A
 INX D
 JNZ O9
 DCX D
 RET
 END
```

ПАРАЛЛЕЛЬНЫЙ ИНТЕРФЕЙС

В "Векторе-06Ц" в качестве параллельного интерфейса используется микросхема КР580ВВ55. БИС имеет 24 линии ввода/вывода,которые можно запрограммировать около 100 различными способами. Программирование производится записью в регистр управляющего слова (порт 04) специальной информации.Имеются также три канала ввода/вывода:А (порт 7), В (порт 06) и С (порт 05).Каналы А и В - восьмиразрядные, С - может делиться на два 4-х разрядных канала, которые могут использоваться как для передачи данных,так и для обмена управляющими сигналами для А и В.

Управляющее слово имеет следующий формат:



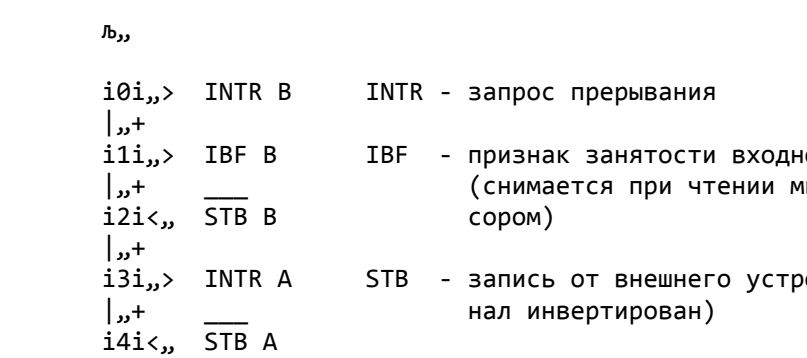
Например: режим 0, А - ввод,
В - вывод, С (мл. разряды) - ввод,
С (ст. разряды) - вывод
Упр. слово: 10010001В

Режим 0 - простой ввод/вывод
Имеем два 8-ми разрядных и два 4-х разрядных канала,каждый из которых может быть входным и выходным, выходы с запоминанием, входы без запоминания.

Режим 1 - клавиатурный ввод/вывод или асинхронный обмен с использованием клавиатурных сигналов.

Имеем две группы А и В. Кканалы А и В могут быть входными и выходными,каждому придается по три канала С для обмена управляющими сигналами. Два оставшихся разряда канала С могут использоваться как отдельный двухразрядный канал ввода/вывода.

Канал С - "Ввод"



```
 |„+           „>   - на ВУ
 i5i„>  IBF A
 |„+                <„   - с ВУ
 i6i<„> ввод/вывод
 |„+
 i7i<„>
 +„+
```

Канал С - "Вывод"

```
 Љ„
                ___
 i0i„>  INTR B        OBF - выходной буфер полон
 |„+    ___           ___
 i1i„>  OBF B         IBF - вырабатывается ВУ после имени, при
 |„+    ___                 этом происходит сброс сигнала OBF
 i2i<„  ACK B
 |„+
 i3i„>  INTR A
 |„+
 i4i<„> ввод/вывод
 |„+
 i5i<„>
 |„+   ___
 i6i<„ ACK A
 |„+   ___
 i7i„> OBF A
 +„+
```

Режим 2

Работает в этом режиме только группа А, которая становится двунаправленной. Каналу А придается 5 линий канала С. Канал В может работать в режиме 0 или 1.Все входы и выходы у канала А с запоминанием.

Канал С - "Ввод" - аналогично режиму 1, только сигналы 6 и 7, работающие на ввод/вывод,заменяются на сигналы ACK A и OBF A соответственно.

Информацию из канала С можно получить обычным чтением порта С и при необходимости выделить и проанализировать нужные разряды.

Имеется возможность программного управления отдельными разрядами порта С при помощи второго типа управляющего слова.

Формат управляющего слова:

```
            7 6 5 4 3 2 1 0
 Упр. Слово ----------------
            -’-’-’-’-’-’-’--+
            | | | | | | | -‘- 0 - сброс
            +-‘-+     1        - установка
                               "0"X X X номер
                               разряда
```

Параллельный интерфейс используется для обмена с внешними устройствами, в частности с принтером, джойстиком и т.д.

ИНСТРУКЦИЯ К ПРОГРАММЕ "ЭЛЕКТРОННЫЕ КУБИКИ"

Программа "Электронные кубики" предназначена для приобретения навыков построения устройств на базе цифровых микросхем и моделирования их работы.

Для построения схем в программе "зашита" библиотека различных эле-

ментов, которые вызываются на экран путем нажатия соответствующих клавиш компьютера.

Перемещение элементов в требуемое место экрана производится клавишами: вверх, вниз, влево, вправо. Установка - клавишей "ВК".

Кроме десяти типов микросхем, в библиотеку входят элементы соединительных шин, позволяющие нарисовать все соединения в задуманном Вами устройстве.Для моделирования работы схемы предусмотрены восемь пронумерованных ключей, которые подключаются к входам проектируемых устройств. Для индикации состояния шин моделируемой схемы (соответствие логической единице или логическому нулю) служит "светодиод", вызываемый на экран клавишей ":" и устанавливаемыйна те шины, работа которых Вас интересует.

Режим моделирования включается в программе клавишей "F2". Исследование работы устройства происходит путем "подачи сигнала" на входы схемы через установленные Вами ключи (подаче сигнала с уровнем логической единицы соответствует однократное нажатие клавиши с номером ключа, подаче логического нуля - повторное нажатие). Состояние интересующих Вас шин отслеживается по "свечению" или отсутствию "свечения" индикаторов, установленных на этих шинах.

Программа обеспечивает вывод составленной схемы на принтер - клавиша "ПС", запись на магнитофон - клавиши "F3" и "F4", чтение с магнитофона - клавиша "F1".

Кроме того имеется 10 различных экранов, в каждый из которых можно записать свою схему.

В.Лагунов, Вожгалы

СХЕМА ПОДКЛЮЧЕНИЯ ПРИНТЕРА МС6312 К ПЭВМ "ВЕКТОР-06Ц"

"Вектор-06Ц"

МС6312 сигнал	контакт		разъем =ПУ= контакт
DATA0(д0)	2	+---+	A09
DATA1(д1)	3	+---+	A08
DATA2(д2)	4	+---+	A07
DATA3(д3)	5	+---+	A06
DATA4(д4)	6	+---+	A05
DATA5(д5)	7	+---+	A04
DATA6(д6)	8	+---+	A03
DATA7(д7)	9	+---+	A02
STROBE(стр)	1	+---+	C05
INIT (зам)	11	+---+	C09
BUSY (mun)	18	+---+	C01,A10
OY (общ)	24	+---+	C01,A10
OY (общ)	25	+---+	C01,A10

Работа УВИП поддерживается программно. Драйверы печати входят в состав стандартного программного обеспечения (интерпретатора BASIC, текстового редактора RETEX, графического редактора "Карандаш"),а так же в состав других прикладных и системных программ.

Управление работой осуществляется сериями ESCAPE последовательностей.ESCAPE последовательность состоит из кода ESCAPE 27 и последующих за ним буквенно-цифровых знаков или символов. (Список команд приведен в инструкции по эксплуатации).

При работе со стандартным BASIC управление производится операторами: LLIST, LPRINT, SCREEN 7,N. Пример: LPRINT CHR$(27); CHR$(15) - включается уплотненный режим.

При работе с RETEX, при включении УВИП, необходимо установить кодировку символов согласно таблицы 9 инструкции.Для этого, при нажатых клавишах

"Линия" и "ПС", включить питание. Ввод ESCAPE последовательностей осуществляется при помощи клавиши "AP2" (ввод символа по его цифровому коду).Пример: AP2G - устанавливается режим двойного удара.(Для ввода команды нужно нажать: AP2-0-2-7-AP2-0-7-1).

В.Лагунов, Вожгалы

Внимание !!!
Только в центре "Байт" !!!
Новинка !!!

Kesha N' Company presents:
PUTUP-LE

Эта программа для Вас,любители прекрасной игры "Putup" ! Кроме новых этажей она содержит встроенный редактор этажей. А, значит, Вы сами сможете построить какие угодно лабиринты этажей, а потом проходить их вместе с друзьями.

Построенные этажи Вы сможете записать на МЛ и затем догружать их в любые версии программы "Putup".

БЛИЦ-ИНФОРМАЦИЯ

1) Для взломщиков.

В "Байте-2" было рассказано о том,как выгружать Монитор для того, чтобы с его помощью взламывать программы в кодах. Также было указано, что из-за особенности начального загрузчика участок карты загрузчика на месте последнего загружаемого блока должен быть чистым, иначе загрузятся только первые 32 байта. Чтобы обойти это препятствие, нужно выгрузить каждый участок Монитора длиной на 1 блок больше. Лишний блок будет фиктивным, то есть не содержащим полезной информации. Но зато теперь полезная информация загрузится полностью, и можно уже не следить за чистотой экрана.

2) В "Байте-8" была приведена подпрограмма выгрузки файла в формате ROM. Кроме указанных управляющих байтов в имени есть еще один, определяющий смещение в блоках.

```
    SHAPKA: DB 'NODISC00030591COPYV4 COM',0

            DB 0,40H,1BH,0C1H,0
                  |   |   |
                  |   |   +---- смещение в блоках
                  |   |
                  |   +-------- число блоков
                  |
                  +------------ начальный блок
```

Начальный блок складывается со смещением, результат будет определять адрес загрузки. В данном примере информация будет выгружаться с адреса 4000H, а загружаться с адреса 4000H+0C100H=100H.

3). В "Байте-7" была напечатана подпрограмма двоичного умножения однобайтового числа на двухбайтовое. В статье отмечалось, что такое умножение должно давать в результате трехбайтовое число, и указывалось, что подпрограмма при больших числах дает округленный результат.На самом деле старший (3-ий) байт результата при выходе из подпрограммы сохраняется в регистре A.Этому способствует команда ACI 0

4). Начальный загрузчик при вводе программ, записанных с высокой скоростью, часто не успевает схватывать их начало.При этом динамично меняющийся стек на экране неподвижно застывает. Для того, чтобы загрузчик всегда схватывал начало программы нужно:

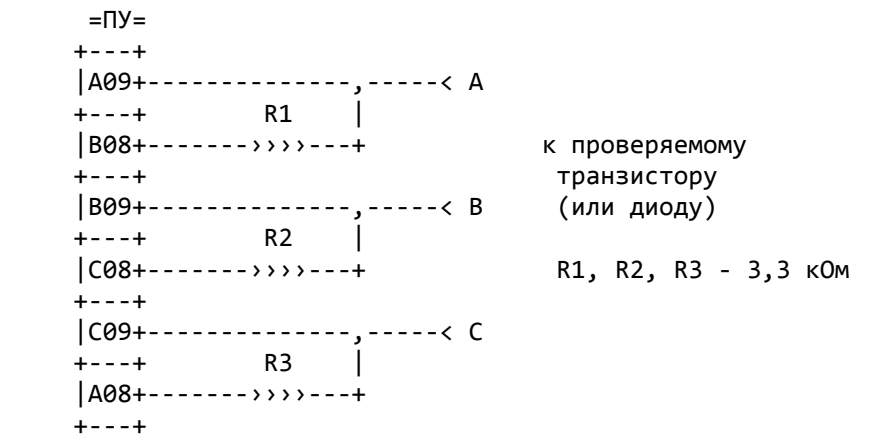
1. Включить МГ на воспроизведение
2. Нажать УС+ВВОД+БЛК
3. Как только начнется программа (но не раньше !) отпустить ВВОД и БЛК, удерживая УС
4. Оотпустить УС

При некотором навыке Вы всегда сможете начать загрузку программы с первого раза.

ИНСТРУКЦИЯ К ПРОГРАММЕ PNP

Вам предлагается простая программа для проверки полупроводниковых приборов. С ее помощью Вы сможете определить базу биполярного транзистора, анод или катод диода,а также выявить робитый или испорченный p-n переход.

Ппрежде чем работать с программой необходимо собрать простую схему (см. рис.) и подключить ее к 30-контактному разъему "ПУ" на задней панели компьютера.



После запуска программы в середине экрана появится изображение:

```
+-,-,-+
|A|B|C|
+-+-+-+
|X|X|X|
+-+-+-+
```

Это означает, что к разъему ничего не подключено.

Затем следует подключить транзистор или диод в любом порядке к контактам A B C. Возможны следующие варианты вывода сообщений:

```
+-,-,-+
|A|B|C| - исправный транзистор структуры p-n-p,
+-+-+-+   база которого подключена к контакту B
|P|N|P|
+-+-+-+

+-,-,-+
|A|B|C| - исправный транзистор структуры p-n-p,
+-+-+-+   база которого подключена к контакту C
|N|N|P|
+-+-+-+

+-,-,-+
|A|B|C| - транзистор с пробитым n-p переходом
+-+-+-+
|P|O|O|
+-+-+-+

+-,-,-+
|A|B|C| - транзистор  с неисправным p-n перехо-
+-+-+-+   дом или диод, подлюченный анодом к B,
|N|P|X|   а катодом к A
+-+-+-+
```

Если при подключении прибора во всех разрядах высвечивается "X", значит, прибор неисправен.

Если необходимо найти коллектор или эммитер транзистора, его определяют каким-нибудь другим способом. Обычно вывод коллектора соединен с корпусом, а в транзисторах с пластмассовым корпусом обычно располагается средним.

КОДЫ КОМАНД МИКРОПРОЦЕССОРА КР580ИК80А

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NOP	LXI B,&	STAX B	INX B	INR B	DCR B	MVI B,#	RLC	–	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C,#	RRC	0
1	–	LXI D,&	STAX D	INX D	INR D	DCR D	MVI D,#	RAL	–	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E,#	RAR	1
2	–	LXI H,&	SHLD *	INX H	INR H	DCR H	MVI H,#	DAA	–	DAD H	LHLD *	DCX H	INR L	DCR L	MVI L,#	CMA	2
3	–	LXI SP,&	STA *	INX SP	INR M	DCR M	MVI M,#	STC	–	DAD SP	LDA *	DCX SP	INR A	DCR A	MVI A,#	CMC	3
4	MOV B,B	MOV B,C	MOV B,D	MOV D,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A	4
5	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A	5
6	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A	6
7	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A	7
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A	8
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A	9
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A	A
B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A	B
C	RNZ	POP B	JNZ *	JMP *	CNZ *	PUSH B	ADI #	RST 0	RZ	RET	JZ *	–	CZ *	CALL *	ACI #	RST 1	C
D	RNC	POP D	JNC *	OUT N	CNC *	PUSH D	SUI #	RST 2	RC	–	JC *	IN N	CC *	–	SBI #	RST 3	D
E	RPO	POP H	JPO *	XTHL	CPO *	PUSH H	ANI #	RST 4	RPE	PCHL	JPE *	XCHG	CPE *	–	XRI #	RST 5	E
F	RP	POP PSW	JP *	DI	CP *	PUSH PSW	ORI #	RST 6	RM	SPHL	JM *	EI	CM *	–	CPI #	RST 7	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

N - номер порта ввода/вывода
& - двухбайтовый операнд - D16
* - двухбайтовый операнд - ADR соответствует
# - однобайтовый операнд - D8
#D | E | H | L | M | A |

Пример: команда STAX D имеет код операции 12.
код операции CA
команде JZ ADR.

" Б А Й Т - 1 6 "

Киров, 1993
информационно-рекламный выпуск
центра "БАЙТ"
для пользователей ПК "Вектор-06Ц"

+++
+ Наш адрес: г. Киров, Октябрьский пр-т, 118, м-н "Дом Радио" +
+ Адрес для писем" 610006, г.Киров, а/я 1248, Зубкову А.Н. +
+++

СЖАТИЕ ДАННЫХ

В этой статье мы продолжаем разговор о сжатии данных,который мы начали в предыдущих выпусках "Байта". Речь идет о сжатии графических образов (картинок). Идея сжатия проста. Переводим прямоугольную картинку в одномерный массив.Ищем в массиве цепочки одинаковых элементов. Найдя такую цепочку, заменяем ее на три элемента: признак сжатых данных, что сжато (образец), сколько раз.

Если исходные цепочки длинные, то сжатие очень эффективно. Легко увидеть и следующее: если длина цепочки равна 2 , то сжатие производить невыгодно.

Имеется множество различных модификаций этого общего метода. Они отличаются выбором признака сжатия и организацией сжатых данных. Кроме того перевод картинки в одномерный массив и сжатие можно производить как в два этапа, так и одновременно. Размер элементов в массиве может быть произвольным: восемь бит (байт) , четыре бита (полубайт), и т.д. Следует заметить, что переходя к битовым модификациям, мы выигрываем в эффективности сжатия, но проигрываем в скорости распаковки,кроме того возрастает трудоемкость написания упаковщика/ распаковщика. Нельзя забывать, что распаковщик должен знать, когда закончить распаковку. Для этого нужно знать или длину исходной картинки или длину сжатой.

Итак ,рассмотрим несколько конкретных модификаций данного метода.

А. Логика алгоритма полностью аналогична выше рассмотренному общему методу. Организация элементов в массиве - байтовая. Особенность: весь массив просматривается и определяется байт, который не встречается в массиве вообще. Этот байт запоминается в начале сжатой картинки и используется затем как признак сжатия. Допустим исходный массив был: 7,0,0,0,0,0,5,5,5,5,8,... и байт, равный 1, не встречался вообще, тогда сжатый массив будет выглядеть:

```
1 , 7 , 1 , 0 , 5 , 1 , 5 , 4 , 8 ,...
I       I   I   I
I       I   I   +-- сколько раз
I       I   +------ образец
I       +---------- признак сжатия
+------------------ байт,использующийся как признак сжатия.
```

Если все байты встречаются в исходном массиве, то можно либо найти сочетание двух байтов, не встречающееся в массиве, либо хранить адреса наиболее редко встречающегося байта отдельно, вычеркнуть его из массива (заменить на другой байт) и использовать

в качестве признака сжатия.
 Данная модификация гарантирует от увеличения обьема сжатых данных при обилии цепочек одинаковых элементов с длиной, равной 2, т.к. признак сжатия мы ставим только тогда, когда нам выгодно это сжатие производить. Однако возникают вышеописанные сложности при определении байта - признака сжатия.

 B. Модификация также байтовая и часто используется программистами на "Синклере". Исходный массив заменяется цепочками двух типов.
 1-ый тип: длина цепочки, последовательность байтов;
 2-ой тип: длина цепочки, байт-образец.
 Оба типа различаются старшим битом длины цепочки. Если он установлен в 0, то далее следует цепочка неупакованных байтов, соответствующей длины. Если старший бит длины устновлен в 1, то оставшиеся семь бит определяют длину цепочки из одинаковых элементов, равных байту-образцу.
 Например, исходный массив:

 7,0,0,0,0,0,5,5,5,5,8,9,3,.....

 Сжатый массив:

```
 1 , 7 , -5 , 0 , -4 , 5 , 3 , 8 , 9 , 3 , .......,0
 I        I   I                                    I
 I        I   +-- байт-образец     признак конца --+
 I        +------ признак/длина сжатых данных
 +--------------- признак/длина несжатых данных
```

 Распаковка прекращается, когда обнаруживаем байт - признак/длину, равный 0.

 C. Наиболее простая, универсальная и часто встречающаяся модификация.
 Логика алгоритма такова. Просматриваем исходный массив,если находим два и более одинаковых элемента рядом, то заменяем их на цепочку: этот же элемент, этот же элемент, длина исходной цепочки. Остальные элементы переписываем без изменений.Признаком сжатия являются два одинаковых элемента рядом.
 Например, исходный массив: 7,0,0,0,0,0,5,5,5,5,8,....
 Сжатый массив: 7 , 0 , 0 , 4 , 5 , 5 , 3 , 8 , ...
```
                    I   I   I
признак сжатия -----+---+   +- длина цепочки
```

 То, что длина цепочки на 1 меньше реальной обьясняется логикой алгоритма при распаковке. Первый байт-признак рассматривается сам по себе, а вот следующий уже сигнализирует о сжатых данных.
 В качестве примера приведем тексты упаковщика и распаковщика на языке Ассемблера для процессора КР580.

```
; упаковщик
; входные данные: BC - начало исх. массива, HL - конец исх. мас
; сива + 1, DE - начало буфера для сжатой информации.
UP: SHLD KON
    INX D    ;оставляем место для длины сжатой картинки
    INX D
    PUSH D
    XCHG
UP1: MVI D,0 ; D - счетчик одинаковых байтов
```

```
    LDAX B   ; первый байт переписываем по умолчанию
    MOV M,A
    MOV E,A  ; E - значение предыдущего байта
    INX H
UP2: INX B   ; смещаемся на следующий байт
    CALL SRAV; достигли конца исх. текста?
    JZ UP5   ; тогда на завершение алгоритма
    LDAX B   ; берем очередной байт
    CMP E    ; сравниваем с предыдущим
    JNZ UP3  ; равны?
    INR D    ; да, тогда увеличиваем счетчик
    JNZ UP2  ; если нет переполнения разр.сетки, продолжаем цикл
    JMP UP7  ; есть переполнение - разрядности не хватает
UP3: DCR D   ; проверяем была ли перед этим цепочка один.байтов
    INR D
    JZ UP4   ; нет
UP7: MOV M,E ; записываем предыд.байт и длину цепочки
    INX H    ; в сжатый файл
    MOV M,D
    INX H
    MVI D,0  ; обнуляем счетчик один. байтов
UP4: MOV M,A ; записываем текущий байт в сжатый файл
    INX H    ; делаем очередной шаг
    MOV E,A  ; обновляем байт-образец
    JMP UP2  ; повторяем цикл
UP5: DCR D   ; проверяем счетчик - длину цепочки
    INR D
    JZ UP6   ; длина равна 0
    MOV M,E  ; иначе записываем байт-образец и длину цепочки
    INX H
    MOV M,D
UP6: SHLD KBUF ; запоминаем адрес конц сжатого файла
    POP D    ; определяем длину сжатого файла и запоминаем ее
    INX H    ; в первых двух байтах сжатого файла.
    MOV A,L
    SUB E
    MOV L,A
    MOV A,H
    SBB D
    DCX D
    STAX D
    DCX D
    MOV A,L
    STAX D
    RET
; п/п определения конца исх. текста
SRAV: PUSH H
     LHLD KON
     MOV A,H
     CMP B
     JNZ SRAV1
     MOV A,L
     CMP C
SRAV1: POP H
     RET
KON: DW 0
KBUF: DW 0

; распаковщик
```

```
; вход.данные: HL - начало упакованных данных,
; BC - начало распакованных данных.
; выход.данные: BC - конец распак. данных + 1
RAS:   MOV E,M      ; извлекаем длину сжатой картинки
       INX H
       MOV D,M
       INX H
       PUSH D
       MOV A,M      ; достаем первый байт по умолчанию
RAS1:  MOV E,A      ; и переписываем его в расп.файл
       STAX B
       INX B
RAS3:  INX H  ; делаем шаг
       XTHL    ; конец файла?
         DCX H
       MOV A,L
       ORA H
       XTHL
       JZ RAS4      ; да
       MOV A,M      ; тек. байт совпадает с предыдущим?
       CMP E
       JNZ RAS1; нет - просто переписываем
       INX H  ; да - достаем длину сжатой цепочки
       MOV D,M
       MOV A,E      ; записываем байт-образец заданное число раз
RAS2:  STAX B
       INX B
       DCR D
       JNZ RAS2
       JMP RAS3
RAS4:  POP D  ; конец программы
       RET
```

D. Имеется еще одна модификация, очень похожая на предыдущую. Так называемый "разностный алгоритм", часто используемый в графических редакторах на IBM. При его работе исходный массив переводится в промежуточную фазу следующим образом: первый элемент переписывается без изменений, вместо второго записывается разность между вторым и первым, вместо третьего - разность между третьим и вторым и т.д. Т.е. вместо текущего элемента записывается разность между текущим и предыдущим элементами. Теперь вместо цепочек из одинаковых элементов, мы получили цепочки из нулей. Следующим этапом является замена таких цепочек двумя элементами: 0, длина цепочки.

Пример, исх.массив: 7,0,0,0,0,0,5,5,5,5,8,...
пром.массив: 7,-7,0,0,0,0,5,0,0,0,3,...
сжат.массив: 7,-7,0,4,5,0,3,3,...

Распаковка идет в обратном порядке. Текущий элемент определяется как сумма текущего элемента сжатого массива и предыдущего элемента распакованного массива. Если встречается 0, то предыдущий элемент распакованного массива повторяется заданное число раз.

Естественно в реальном алгоритме построение всего промежуточного массива не производится, т.к. нас в каждый момент времени интересует только текущая разность между текущим и предыдущим элементами массива, а не все сразу.

Луппов Г.Б., Киров

ПРОГРАММА "ШРИФТ"

Многие компьютерные игры удивляют нас своими оригинальными надписями. На экране появляются наклонные, утолщенные, готические, рисованные буквы, иероглифы и другие символы. Это делает игровую программу более привлекательной.

В том, что разработка шрифта - занятие длительное и кропотливое, никого убеждать не нужно. Данная программа позволяет облегчить его. Сделать это можно следующим образом: вначале извлечь интересные шрифты из готовых программ, а затем использовать их для собственных целей. Кроме того программа знакомит с основами программирования на языке Ассемблер. Программа работает совместно с "Монитором",используя некоторые его подпрограммы.

Для начала нужно оттранслировать данный текст программы согласно описания "Ассемблер-редактор". Затем директивой Монитора M1800,194E,9200 переместить программу по рабочим адресам (начальный адрес - 9200). Нажав клавиши УС+БЛК+СБР загрузить любую программу.

Снова войти в Монитор и командой G9200 запустить программу "Шрифт", изменяя адреса исследуемой программы клавишами управления, следить за окном в нижней части экрана. Окно содержит набор знаков,формируемых в данный момент знакогенератором. Почти всегда это будут какие-то случайные формы. Программа "Шрифт" поможет найти место размещения знакогенератора в выбранной игровой программе. Поиск осуществляется с помощью клавиш управления.

Когда в окне начнут появляться какие-нибудь осмысленные наборы знаков, следует с помощью клавиш совместить форму, соответствующую букве алфавита "А" с левым краем окна.Потом следует запомнить адрес начала знакогенератора, выведенный в левом верхнем углу экрана,выйти в Монитор (БЛК+СБР) и записать набор знаков на магнитную ленту командой W.

Теперь всякий раз, когда мы захотим использовать выбранный шрифт в собственной программе, нужно будет загрузить его по адресу, с которого расположен знакогенератор.

Теперь все появляющиеся на экране надписи и даже листинг программы будут написаны новым шрифтом.

Исходный текст программы:

```
      ORG 9200H  ;начальный рабочий адрес программы
      LXI H,TITR ;оформление
      CALL 0F818H;экрана
      LHLD KEY   ;запись нач.адреса в HL
      JMP START  ;переход на начало программы
TITR: DB 1FH,18H,18H,18H,18H,18H,'<=АДРЕС',1BH,59H,20H,50H
      DB 'П Р О Г Р А М М А',1BH,59H,22H,51H,'= Ш Р И Ф Т ='
      DB 1BH,59H,28H,3FH,'КЛАВИШИ',1BH,59H,29H,3FH
      DB 'УПРАВЛЕНИЯ:'
      DB 1BH,59H,2BH,3FH,'<- УМЕНЬШЕНИЕ АДРЕСА',1BH,59H,2CH
      DB 3FH,'-> УВЕЛИЧЕНИЕ АДРЕСА',1BH,59H,2DH,3FH
      DB 'ТАБ - БЫСТРО/МЕДЛЕННО',1BH,48H,0

; подпрограмма опроса клавиатуры

START: CALL 0F81BH ;опрос клавиатуры
       CPI 8            ;клавиша "<-" ?
       JZ LEW           ;подпрограмма сдвига влево
```

```
       CPI 18H          ;клав. "->" ?
       JZ PRAW          ;подпрограмма сдвига вправо
       CPI 9            ;клавиша "ТАБ" ?
       CZ TURBO         ;подпрограмма ускоренного сдвига
       JMP START        ;сканирование клавиатуры

; подпрограмма сдвига вправо

PRAW: LXI D,0C810H ;начальный адрес экрана
      LDA TUR
      CPI 0
      CNZ MP
      INX H
      SHLD KEY     ;запомним указатель адреса
      CALL ZNAK
      LHLD KEY
      JMP DISPL

; ускоренное перемещение указателя вправо

MP:  MVI C,9
MP1: INX H
     DCR C
     JNZ MP1
     RET

; подпрограмма сдвига влево

LEW: LXI D,0C810H  ;нач. адрес экрана
     LDA TUR
     CPI 0
     CNZ ML
     DCX H
     SHLD KEY          ;запомним указатель адреса
     CALL ZNAK
     LHLD KEY
     JMP DISPL

; ускоренное перемещение указателя влево

ML:  MVI C,9
ML1: DCX H
     DCR C
     JNZ ML1
     RET

; подпрограмма вывода знакоместа на экран

ZNAK:MVI B,10H
DATA:MOV A,M
     MVI C,8
SDW: RLC               ;сдвиг побайтно
     CC ZAP1           ;выводим квадрат
     CNC ZAP2          ;пусто
     DCR C
     JNZ SDW           ;след. бит
     INX H             ;след. ячейка
     INR D
     MVI E,10H
```

```
      DCR B             ;уменьшаем счетчик
      JNZ DATA          ;след. байт
      RET
 ZAP1:PUSH B
      XCHG
      MVI C,7           ;счетчик
 ZAP11:MVI M,0FEH
      INR L
      DCR C
      JNZ ZAP11
      MVI M,0
      INR L
      XCHG
      POP B
      RET
 ZAP2:PUSH B
      XCHG
      MVI C,8
 ZAP21:MVI M,0
      INR L
      DCR C
      JNZ ZAP21
      XCHG
      POP B
      RET

 ; подпрограмма вывода адреса на дисплей

 DISPL:LDA KEY+1
       CALL 0F815H
       LDA KEY
       CALL 0F815H
       MVI C,0DH
       CALL 0F809H
       JMP START

 ;установка/снятие признака режима TURBO

 TURBO:LDA TUR
       CMA
       STA TUR
       RET

 KEY:  DW 100H
 TUR:  DB 0

 ;конец исходного текста
```

После трансляции программы и перевода ее по рабочему адресу рекомендуется выгрузить ее директивами ІШРИФТ и W9200,934E на магнитную ленту и использовать в формате Монитора.

Для начала можно просмотреть знакогенератор "Монитора-отладчика". Он находится с адреса F300H. Двигаться по ячейкам памяти можно в любом направлении, т.е. при движении влево после адреса 0000H будет адрес FFFFH и т.д.

Лебедев Н.Г.,п.Вожгалы

А знаете ли вы ?

"SPACE DEMON"

Ваша задача управляя космическим истребителем выжить, отбивая атаки чудовищ и другой твари. В ходе игры появляются птицы, которые несут в своих лапах дополнительное оружие или призовые очки

Shot - Усиление огневой силы.
Magic - Образования защитного шара, который вращается вокруг корабля и уничтожает всех врагов попавщихся ему на пути.
Score - Призовые 500 очков
* При гибели все ранее набранное оружие отбирается.
* При наборе 10000 очков прибавляется жизнь.

Всего в игре шесть зон.

1. Good Luck (практика)
2. Danger Zone
3. Crazy Mytants
4. Space Snake
5. War Zone
6. Space Demon

P.S. Проходя последню зону вы должны долго упорно стрелят в космического демона и только тогда добьетесь победы.
Кл. "CTP"-пауза. Кл. "ПРОБЕЛ"- отмена паузы.

"POLY GAME"

Эта игра включает в себя четыре известные вам игры: Tetris, Columns,Пентикс,Крестики.Можно выбрать любую из игр. Для этого вам нужно нажать клавишу "AP2" и вверху появиться меню в котором и можно выбрать нужную вам игру. Также можно играть вдвоем. Клавиши первого игрока стрелки, второго "Ш","Щ","Z" и кл. "ПРОБЕЛ". При игре вам мещает жучек, которого не стоит придавлять фигурой, потому что это кончится плачевно. Но можно играть и без него.

Так же в игре возможно:

* выбор уровня сложности
* выбор подсказки
* 1 или 2 игрока
* выбор игры
* выбор помех
* усложнение игры и другое.

При игре вдвоем собранные полные вами строки переходят другому, только в "разбитом" виде. Это повышает трудность и интерес к игре такой вариантт возможен только при игре в "tetris" в других случаях этого не происходит.
"Пентикс"- это игра аналогична игре "tetris" только более сложными фигурами.

Назаров А.Е. г.Киров 1993 г.

" Б А Й Т - 1 7 "

Киров, 1993
информационно-рекламный выпуск
центра "БАЙТ"
для пользователей ПК "Вектор-06Ц"

+++
+ Наш адрес: г. Киров, Октябрьский пр-т, 118, м-н "Дом Радио" +
+ Адрес для писем" 610006, г.Киров, а/я 1248, Зубкову А.Н. +
+++

Новая аппаратная разработка для "Вектора"!

П Р О Г Р А М М И Р У Е М Ы Й Г Е Н Е Р А Т О Р
(М У З Ы К А Л Ь Н Ы Й С Т Е Р Е О - С О П Р О Ц Е С С О Р)
Д Л Я Б П Э В М "В Е К Т О Р - 0 6 Ц"
А У - 3 - 8 9 1 0

Применение: БПЭВМ "Вектор-06Ц" (адаптация музыки с "ZX-SPECTRUM 128 кбт")

(c) Design By Саттаров В.

Технические характеристики:

3 независимых музыкальных канала (стерео музыка) + генератор "белого шума"
2 (или 1) параллельных порта ввода/вывода
16 градаций амплитуды
16 форм волного пакета
Основная частота - 1.7734 Мгц
Диапазон воспроизводимых частот: от 27 Гц до 11 Кгц
Частоты волнового пакета: от 0.1 Гц до 6 Кгц
Частота генератора шума: от 3 Кгц до 110 Кгц

Программирование ПГЗ АУ-3-8910

ПГЗ является регистро-ориентированным генератором звуков. Его функции выполняются посредством 16 внутренних регистров. Номер регистра задается 4 младшими разрядами при подаче команды "фиксация адреса" и остается действительным до получения команды о смене этого адреса.
В таблице приведены функции регистров и допустимые значения для этих регистров.

N регистра	назначение или содержимое	значение
0, 2, 4	нижние 8 бит частоты голосов A, B, C	0 - 255
1, 3, 5	верхние 4 бита частоты голосов A, B, C	0 - 15
6	управление частотой генератора шума	0 - 31
8, 9, 10	управление амплитудой каналов A, B, C	0 - 15
11	нижние 8 бит управления периодом пакета	0 - 255
12	верхние 8 бит управления периодом пакета	0 - 255

13	выбор формы волнового пакета	0 - 15
14, 15	регистры портов ввода/вывода	0 - 255

Основным при работе ПГЗ является регистр 7. Его главное назначения - определять, какие каналы должны участвовать в образовании звука и определять направление обмена портов ввода/вывода.

Его структура показана на рис.1. Ноль соответствует включению определенной позиции, а единица - выключению.

7	6	5	4	3	2	1	0
порт B	порт A	шум C	шум B	шум A	тон C	тон B	тон A
упр.ввод/вывод	выбор канала для шума				выбор канала для тона		

Для портов "1" соответствует режиму вывода информации.

Рис.1. Регистр смешивания и выбора канала

OUT 15H - задает номер регистра
OUT 14H - задает значение этого регистра

Частота генератора = 1625000/(16*((256*R1)+R0))
Частота шума = 1625000/(16*R6)

Амплитуда: R0, R9, R10: бит 0-3 - эффективное значение
бит 4 - модуляция (1- активно)
Период волнового пакета P=1625000/(256*((256*R12)+R11)

Состояние регистра R13 определяет форму звучания сигнала. Его значения показаны в таблице. Бит 3 этого регистра определяет включен эффект или нет. (0 соответствует выключению эффекта).

Уровень звучания	Значение
0	одно снижение, затем выключение
1	одно нарастание, затем выключение
2	одно снижение, затем поддержка
3	одно нарастание, затем поддержка
4	повторящееся снижение
5	повторящееся нарастание
6	повторяющееся нарастание-снижение
7	повторяющееся снижение-нарастание

Внимание! В настоящее время заканчивается работа над музыкальным редактором, поддерживающим стерео-процессор AY-3-8910, который позволяет писать музыку даже непрофессионалам (расставляя ноты на нотном листе), редактор полностью использует все возможности стерео-процессора. Конечным итогом работы редактора является исполняемый модуль в формат ROM, который можно вставить в любую программу. Также ведется разработка нескольких музыкальных DEMO-версий.

СЖАТИЕ ДАННЫХ МЕТОДОМ ХАФФМАНА

(печатается по статье "Как работает упаковщик", из журнала "Компьютер-пресс",№6 за 1991 год.)

Алгоритм Хаффмана (Huffman Enkoding), или кодирование символами переменной длины, предлагает отказаться от обычного представления файлов в виде последовательностей 8-ми битовых символов. Главный недостаток такого способа записи файлов состоит в том, что в нем не делается различий между кодированием символов, с разной частотой втречающихся в файлах. Так, наиболее частые в английском языке E или I, имеют ту же длину,что и относительно редкие Z,X или Q. Код переменной длины позволяет записывать наиболее часто встречающиеся символы всего лишь несколькими битами, в то время, как редкие символы и фразы будут записаны более длинными битовыми строками. Простейший способ кодирования информации символами переменной длины, осуществляется при помощи таблицы соответствий (translation table). Например, анализируя любой английский текст , можно установить, что буква E встречается гораздо чаще, чем Z, а X и Q относятся к наименее часто встречающимся. Таким образом, используя таблицы соответствий,мы можем закодировать каждую букву E меньшим числом бит, используя более длинный код для более редких символов. Хотя такой прием пригоден для текстов любого типа , большинство практических программ вычисляют свою таблицу соответствий для каждого нового документа. Таким образом исключаются случайные отклонения каждого текста от среднестатистического и устраняются связанные с этим потери эффективности сжатия.

Попробуем описать алгоритм кодирования Хаффмана на наглядном примере. В качестве кодируемого текста возьмем предложение: This is a simple example of Huffman encoding.

Первая задача состоит в подсчете числа повторений каждой буквы алфавита в исходном тексте (рис.2). Далее мы должны удалить из списка те буквы, которые ни разу не встретились в нашем тексте и построить дерево кодирования (снизу вверх). Лучше всего предварительно отсортировать получившийся список в порядке убывания частоты повторений символов. Построение дерева мы начнем от самого нижнего символа в списке. Частоты двух наиболее редко встречающихся символов просуммируем и результат запишем в узле дерева, как показано на рис.3. Исходные частоты стали теперь детьми новой суммарной частоты. Если имеется более двух символов с минимальной частотой повторений, то нужно образовывать из них самостоятельные пары. Если имеется нечетное число наименьших значений частоты - будем обьединять непарную частоту со следующей наименьшей частотой (рис.3). По мере продолжения процесса построения дети становятся внуками, правнуками и так до тех пор, пока все дерево не будет закончено. Полное количество символов исходного текста будет представлено в вершине полученного дерева. Таким образом можно осуществлять эффективную проверку правильности построения дерева кодирования (рис.4).

Необходимо сказать, что та структура дерева, которую мы создали является оптимальной с точки зрения минимизации кодирования.То есть с помощью таблицы кодирования (или префиксного кода), полученной на основе анализа структуры дерева, можно наиболее эффективно считать исходный текст. (Это можно доказать математически, но мы здесь ограничимся лишь констатацией). Окончательный код Хаффмана для каждого символа исходного текста можно получить, добавив к каждой ветви дерева определнный двоичный атрибут. Все левые ветви, исходящие из всех узлов, мы можем помечать "1", а все правые ветви - "0". Таким образом код каждой буквы мы сможем полу-

чить, перемещаясь по ветвям дерева от вершины к интересующей нас букве и записывая по порядку двоичные атрибуты ветвей.

Результаты сведем в таблицу, которую будем использовать в качестве "словаря" при кодировании и декодировании. Например , слово this теперь будет кодироваться последовательностью 000010 0110 1100 0111. Общая длина кода уменьшилась при этом с 32 бит до 17 бит. Для всего предложения коэффициент сжатия составляет 175/352, что дает 0.497.

При разработке реальных программ сжатия данных составление словаря кодирования оказывается несколько более сложным, чем это может показаться из приведенной здесь диаграммы. Кроме того, при кодировании и декодировании информации возникают проблемы , связанные с переменной длиной получаемого кода. Для решения этих задач используется специальные технические приемы, которые, однако , не представляют самостоятельного интереса, поскольку не влияют на эффективность сжатия.

На основе данного алгоритма работают архиваторы для "Вектора" - Архив-1.2 и Архив-1.3, которые Вы можете заказать в нашем центре.

This is a simple example of Huffman enkoding.

Рис.1 Исходный текст для кодирования.

Символ.	Пробел	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Част.	7	3	0	1	1	4	3	1	2	4	0	0	2	3	3	2	2	0	0	3	1	1	0	0	1	0	0

Рис.2.Таблица частот каждого символа.

```
Пробел 7
E      4
I      4
A      3
F      3
M      3
N      3
S      3
H      2
L      2
O      2
P      2-----4
C      1--2 /
D      1-/
G      1--2--4
T      1-/  /
U      1--2/
X      1-/
```

Рис.3.Символы с наименьшими частотами группируются в пары и суммируются.

```
Пробел 7-1----15-1--27--
     E 4-1-8-0/     /   \
     I 4-0/        /     \
     A 3-1-6-1-12-0       \
     F 3-0/    /           \
     M 3-1-6-0/             \
     N 3-0/                  \
     S 3-1-5-1-9-1-------17-0-44
     H 2-0/    /       /-/
     L 2-1-4-0/       /
     O 2-0/          0
     P 2-1-----4\   /
     C 1-1-2-0/  1-8
     D 1-0/       /
     G 1-1-2-1\  0
     T 1-0/    4/
     U 1-1-2-0/
     X 1-0/
```

Рис.4.Когда дерево частот построено
число в вершине равно количеству символов в исх.тексте.

Получается таблица кодирования:

Пробел	1 1 1	L	0 1 0 1
E	1 1 0 1	O	0 1 0 0
I	1 1 0 0	P	0 0 1 1
A	1 0 1 1	C	0 0 1 0 1
F	1 0 1 0	D	0 0 1 0 0
M	1 0 0 1	G	0 0 0 1 1
N	1 0 0 0	T	0 0 0 1 0
S	0 1 1 1	U	0 0 0 0 1
H	0 1 1 0	X	0 0 0 0 0

Рис.5.Код Хаффмана, используемый для кодирования и декодирования исходного пакета.

"BOLD/1 and BOLD/2"

Если Вы записали игру "BOLD/1" или "BOLD/2",но не знаете как играть, вот ее инструкция:

Управляя человечком и собирая алмазы, нужно пройти уровни. Во время игры вам будут мешать бабочки и квадраты. При столкновении с ними вы теряете жизнь.

Вот основные элементы этажа:

1). Затушованная стенка - человечек может пройти через нее, а квадрат и бабочка нет. Там где человечек прошел, стенка исчезает.

2). Камни - человечек может двигать один камень нажатием клавиш "<-" и "->". Если человечек стоит под падающим камнем, то он погибает, как и все другие недруги.

3). Алмазы - с помощь их человечек набирает очки, за каждый алмаз дается столько очков, сколько указано в левом верхнем углу экрана. В зависимости от этажа и уровня сложности разная ценность у алмазов. Так же при взятии нужного количества алмазов (указывается в левом верхем углу) открывается выход для перехода на следующий уровень.
Если человечек стоит под падающим камнем, то он погибает, как и все другие недруги.

4). Кирпичная стенка - непроходима для человека. Но если камень упадет на квадрат или бабочку, находяшуюся около стенки, то он "взрывается", после чего часть стены рядом с ним пропадает.

5). Мигаюшая стенка - прходя через нее, камни превращаются в алмазы, а алмазы - в камни.

6). Разратающаяся жидкость-через некоторое время,если она огорожена со всех сторон (камнями,алмазами,другими стенками(кроме затушованной стенки)),она превращается в алмазы,если она не огорожена-в камни. При столкновении бабочки с жидкостью бабочка превращается в алмазы, а при столкновении квадрата происходит "взрыв".

7). Выход на следующий уровень (похож на игральный кубик), который открывается после взятия нужного количества алмазов.

Пр попадании камня(алмаза):
а). в бабочку она превращается в 9 алмазов.

б). в квадрат происходит "взрыв", камни попавшие под "взрыв" исчезают.

P.S. В игре возможен выбор этажа клавишами "<-" и "->".
Клавишами "вверх" и "вниз" можно установить сложность уровня с увеличением сложности меняется расположение камней и уменьшается время,отведенное на прохождение данного уровня. По истечению времени теряется одна жизнь.

Клавиша "УС" - самоуничтожение.
Клавиша "ВК" - начало игры.
Клавиша "ТАБ" + стрелка (в нужном направлении) можно стоя на месте можно взять алмаз или толкнуть камень "вслепую".

Назаров А.Е. г.Киров 1993 г.

А знаете ли вы ?

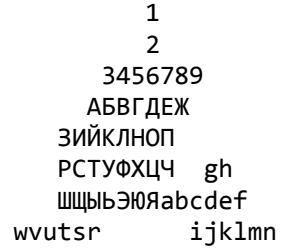
"LIBERATOR"

Управляя танком вы должны поразить цели противника, произвести разведку местности и захватить штаб.
В процессе игры вам нужно пополнять запас горючего и боеприпасов. Кл. "AP2"-выводит приборный шит,в котором выдается сообщения: о повреждениях во время боя, о запасах горючего и боекомплекте.
Кл. "F1"-вкл/выкл паузы.
Кл. "F3"-замена танка
Кл. "F5"-останов танка (тормоз)

"JET SET WILLI-1"

Вы должны помоч Вилли собрать все предметы в доме, тогда Мария пустит его себе в постель. Играть в эту игру вам поможет карта.

Карта игры "JET SET WILLI-1"



Условные обозначения под каждой буквой стоит название этажа.

1-Нет хода	Т-Плавател.бассейн	n-Нет хода
2-Башенные часы	У-Инд.смоковница	o-Вход в Гадсе
3-Номен луни	Ф-Ком.кошмаров	p-Под дорогой
4-На крыше	Х-1-ая комната	q-Корень дерева
5-Перепрыги гантель	Ц-Капелька	r-Забытое аббатство
6-Над кольями	Ч-Опоры ванной ст.	s-Винный погреб
7-Уверял,что пр.это видел	Ш-Черный ход	t-Мастерская
8-Спаси Эсмиральду	Щ-Забытая лестница	u-Берег
9-Вершина дома	Ы-Холодильник	v-Яхта
А-Теплица	Ь-Запад кухни	w-Нос яхты
Б-Под крышей	Э-Кухня	

```
В-Чердак                     Ю-Главная лестница
Г-Док.Джокс в это не повер. Я-Западная зала
Д-безымянный                 a-Восточная зала
Е-Святилище                  b-Холл
Ж-Аварийный генератор        c-Парадная дверь
З-Около западной спальни     d-В ветвях под дорогой
И-Левое крыло крыши          e-Внутри великого дерева
Й-Оранжерея                  f-Кукушкино гнездо
К-Часть дерева               g-Снаржи в ветвях
Л-Хозяйка спальни            h-Вершина великого дерева
Н-Верхняя комната            i-Надежная охрана
О-Ванная                     j-Дорога
П-На пол пути к вост.стене   k-У подножия великого дерева
Р-Западная спальня           l-Под великим деревом
С-Левое крыло                m-Мост
```

Игра "Life" ("Жизнь")

Инструкция.

1. Набор поля

   1). Управление
   маркером: клавиши управления курсором
   окном: УС + <стрелка вверх> - вверх
           УС + <стрелка вниз> - вниз
           УС + <-   - влево
           УС + ->   - вправо
   дисплеем: СС + <стрелка вверх> - вверх
           СС + <стрелка вниз> - вниз
           СС + <-   - влево
           СС + ->   - вправо

   2). Установка точек
   стрелка влево-вверх - зажечь точку
   СТР   - погасить точку

   3). Память
   Писать в память - [П]
   Читать из памяти - [Ч]
   P.S. При отсутствии надпись: "Нет данных"
   Писать на МЛ - [УС]+[П]
   Читать с МЛ - [УС]+[Ч]
   P.S. При записи или чтении вводится имя 1-8 букв. Исправле-
        ния - [ЗБ], конец - [ВК].
   При чтении нажатие клавиши СС или ошибка при чтении вызывают
   надпись "ОШИБКА" и выход в режим построения. При считывании
   программы, записанной в форматах ROM, BAS, MON и др.,выдает-
   ся надпись "НЕ LIFE ФАЙЛ" и переход в режим построения.

   4). Дополнения
     [С]   - вкл/выкл сетку
   [УС]+[Т] - стереть игровое поле
     [Т]   - стереть окно
     [Х]   - показать номер хода
     [Ф]   - показать имя файла

```
  5). Начало игры
      <пробел> - с окном
       УС+СТР  - без окна
       СС+СТР  - в окне

2. Макросы

  1). Информация
     Макрос - это набор команд в памяти ЭВМ. Его можно вызвать на-
     жатием одной или более клавиш. Есть два типа макросов:
           Готовые        - 1
           Макросы игрока - 2
                Готовые макросы
       0 - блок              8 - средний
       1 - бадья             9 - тяжелый
       2 - улей              A - ружье
       3 - пруд              B - пожиратель
       4 - лодка             C - паровоз
       5 - корабль           D - мигалка
       6 - планер            E - светофор
       7 - легкий            F - тумблер
      Все названия из книги М. Гарднера (см. раздел литература).
                  Макросы игрока
    За клавишей F3 и до AP2 в памяти ЭВМ запоминаются нажатые  иг-
    роком клавиши.

  2). Вызов макроса
      Макрос игрока (простой):  F4+[П]     - вызов макроса игрока
      Макрос игрока (сложный):  F5+[П]     - зеркальный вызов  ма-
                                             кроса игрока
      Готовый макрос (простой): F1+[П]+[Н] - вызов  готового  мак-
                                             роса
      Готовый макрос (сложный): F2+[П]+[Н] - зеркальный вызов  го-
                                             тового макроса
      P.S. Сразу после вызова  макроса  его можно убрать  нажатием
           клавиши ЗБ.
      [П] - поворот (в градусах): <стрелка вниз>   - 0
                                  <стрелка вправо> - 90
                                  <стрелка вниз>   - 180
                                  <стрелка влево>  - 270
      [Н] - номер макроса (см. список макросов).

  3). Запись макроса игрока \  смотри
  4). Список макросов       /  Информация

3. Остановка игры

   Игра Life, переход в режим набора поля - УС или СС.

4. Дополнения
  1). Автор игры - Джон Хортон Конуэй (Америка), Октябрь 1970 г.
                   John Horton Conway (Scientific American),
                   October, 1970
  2). Copyright
      (C) 1990
      (C) Филиппов Е. В., г. Кишинев, т. 52-80-27
  3). Литература
      М. Гарднер "Крестики-нолики", М:Мир, 1988
```

P.S. 1. В режиме "Инструкция" клавиша F4 - выход в режим построения.
2. В режиме построения: надпись "X:073 Y:127" - координаты курсора (0<=X или Y<=255).
3. При перезапуске (СБР и БЛК) очистки игровог о поля не происходит, но при нажатии УС+СБР и БЛК происходит очистка игрового поля.

TMK Soft - 1992

" Б А Й Т - 1 8 "

Киров, 1993
информационно-рекламный выпуск
центра "БАЙТ"
для пользователей ПК "Вектор-06Ц"

+++
+ Наш адрес: г. Киров, Октябрьский пр-т, 118, м-н "Дом Радио" +
+ Адрес для писем" 610006, г.Киров, а/я 1248, Зубкову А.Н. +
+++

РЕСТАРТ MDOS ЧЕРЕЗ СБР+БЛК

Многие пользователи MDOS (независимо от версии MDOS1 или MDOS2, т.к. в данном случае это безразлично) страдают от того, что их ОС не перезапускается при нажатии СБР+БЛК. Это возникает из-за несоответствия между MDOS и схемой квази-диска (в дальнейшем КД).]ри одновременном нажатии на клавиши БЛК и СБР сигнал СБРОС с системной шины АС41 (по схеме компьютера) выключает КД, т.к. сбрасывает в 0 D7,D8,D10 (по схеме КД). Поэтому при рестарте необходимо включить КД перед тем, как сделать переход по стартовому адресу (ячейки 0001 и 0002), иначе переход будет произведен не в память КД, а в экранное ОЗУ, что приводит к "зависанию" компьютера. Поэтому по адресу 0000 надо вместо кода 0С3Н записывать код 0F7H (RST 6), что при рестарте приведет к переходу по адресу 0030H и включению КД.

В MDOS вектора переходов устанавливаются следующим фрагментом.

для MDOS2:	для MDOS1:
D365 MVI A,C3	DB65 MVI A,C3
D367]TA 0000	DB67 STA 0000
D36A LXI H,B903	DB6A LXI H,C103
D36D SHLD 0001	DB6D SHLD 0001
D370 STA 0005	DB70 STA 0005
D373 LXI H,B900	DB73 LXI H,C100
D376 SHLD 0006	DB76 SHLD 0006

Но для нас это безразлично, т.к. MDOS самонастраивающаяся система и мы будем менять не настроенный код, который идентичен для обеих версий MDOS. Вот его текст при загрузке в SID с адреса 100H (для этого наберите командную строку A>SID OS.COM<ВК>):

```
1E65 MVI A,C3
1E67 STA 0000
1E6A LXI H,0203
1E6D SHLD 0001
1E70 STA 0005
1E73 LXI H,0200
1E76 SHLD 0006
```

Заменим его на следующий код:

```
1E65 MVI A,C3
```

```
    1E67 STA 0005
    1E6A LXI H,0203
    1E6D SHLD 0001
    1E70 LXI H,0200
    1E73 SHLD 0006
    1E76 MOV H,L    ;L=0  из предыстории
    1E77 MVI M,F7   ;HL=0
```

Т. к. перестраиваемый байт с адреса 1E75H передвинулся на адрес 1E72H, проводим соответствующие изменения в массиве настройки, а очнее меняем 04 по адресу 266EH на 20H.

Кому мое обьяснение покажется не ясным может просто сменить значения ячеек памяти через директиву S SIDa в соответствии с таблицей:

Адрес ячейки	Старое значение	Новое значение
1E68	00	05
1E70	32	21
1E71	05	00
1E72	00	02
1E73	21	22
1E74	00	06
1E75	02	00
1E76	22	65
1E77	06	36
1E78	00	F7
266E	04	20

Виктор Саттаров, Киров-1992

Обзор программ-упаковщиков для БПЭВМ "Вектор-06ц"

Аннотация.

Уважаемые господа пользователи, вашему вниманию предлагается статья, в которой я попытался вкратце, но доступным языком описать все имеющиеся на данный момент у меня "упаковщики", "прессы", "архивы" и аналогичные программы, уменьшающие об'ем программ,а так же оценить особенности и эффективнось работы каждой из них. Заранее приношу извинения за не совсем полный приводимый список программ, т.к. не все авторы соглашаютя предоставлять свои произведения на всеобщее народное обсуждение и использование, а некоторые версии еще не пришли ко мне (например, COMDEC или DeltCompressor).

Уважаемые господа программисты,авторы упаковщиков! Ваши программы приобретут большую известность, если описания их будут включены в один из таких обзорных выпусков. Такая реклама не будет вам ничего стоить, но расширит круг пользователей ваших программ и принесет вам известность в кругах "векторных" программистов не только Санкт-Петербурга, но и других городов страны. Вы можете высылать свои разработки и предложения по адресу :

189630 г.Санкт-Петербург - Колпино-5 ул.Тверская 50-23 Мершиеву Ф.А.
Мой телефон (812) 463-38-27.

По Вашему желанию, Ваши авторские произведения будут (или не будут) тиражироваться на взаимовыгодных условиях.

* * *

При исследованиях в качестве "подопытного кролика" была использована программа "Тест устройств" (базовое программное обеспечение). Ее начальный об'ем - 64 блока (здесь и далее размеры файлов указаны только в HEX-виде).

* * *

АРХИВ 1:2 Луппов Г.Б. Киров 1992

Впервые этот упаковщик был представлен центром "Байт" в составе авторского выпуска(А-1) программ Г.Б.Луппова в прошлом году. Он выполнен в виде копира со стандартной картой загрузки, определяет имя загружаемого файла (здесь и далее имеются в виду файлы в формате ROM), хотя копировщиком совсем не является. По окончании загрузки определяет, не был ли данный файл им уже сжат. Если да, то происходит автоматическая распаковка, в противном случае-автоматическая упаковка, при этом архиватор выводит об'ем программы до и после обработки.
Для знатоков можно добавить, что алгоритм упаковщика построен на методе оптимального статистического кодирования Хаффмана.
Сжатый модуль грузится в память со 0100h.
Эксперимент показал следующее: "Тесты устройств" были сжаты с 64 до 5B блоков. На упаковку было затрачено 30сек., на распаковку при запуске сжатого модуля - 2мин.10сек. Медлительность распаковщика - главный недостаток этой программы.
Внимание! В версии "Архив 1:3" учтен этот недостаток. Скорость распаковки увеличилась в 13 раз (с 2 мин 10 сек до 9 сек)!

Pack-1 Желнов П.А. Отрадный 1992

Выполнен в виде сервисного копировщика. Автоматически упаковывает загруженный файл (если тот не был сжат ранее тем же упаковщиком) , выдавая начальный и конечный об'ем программы.После упаковки сжатый модуль будет загружаться в память с адреса 0000h (что не всегда желательно), так что, если Вы хотите без проблем запускать затем такую программу из ДОС, используйте программу R0.COM (Желнов П.А.) или PS.COM (Калашников Б.О.). Подробное описание Pack-1 Вы найдете в файле PACK1.DOC, написанном самим автором упаковщика.
Затраты времени на упаковку (здесь и далее речь идет о "Тестах устройств")- 8сек., на распаковку при запуске - 2сек. Сжатый модуль - 5C блоков.
Чуть позже автором была написана обратная программа-распаковщик - UnPack-1 , она работает как копировщик (описание в PACK1.DOC), и лишь при нажатии [F1] производит распаковку модуля.

Unpack Авдеюк Н.В. Волгоград 27.09.1992

Представляет из себя исходный ассемблерный текст простых упаковщика и распаковщика, расчитанных на работу в МОНИТОРе. Имеет большие ограничения сжимаемого модуля, т.к. производит упаковку без "наложения" и нужно самому смотреть, где отвести область памяти для упаковываемого(распаковываемого) массива, да так, чтоб они не наложились друг на дружку, да МОНИТОР не затерли...
Данная прогграмма носит скорее обучающий, нежели прикладной характер и может оказаться полезной для начинающего программиста.

Упаковщики фирмы SeS, написанные в Петербурге за 1992 год :

Все упаковщики этой фирмы расчитаны на работу в МОНИТОРе. После упаковки сжатый модуль загружается в память с адреса 0000h. Описание каждого из них имеется в программе SeS-Text-2. Теперь отдельно о каждом из них :

Compressor-1

Сжал подопытный файл до 5D блоков, на упаковку затратил 6 сек., на распаковку при запуске сжатого модуля - 4 сек.

Quick-Compressor v(1)

В отличии от предыдущего, позволяет сжимать программы, работающие как с адреса 0100h, так и с 0000h (по выбору).

Сжал тоже до 5D блоков, но на упаковку затратил 16сек., а на распаковку - около секунды.

Quick-Compressor v(2)

Позволяет устанавливать рабочую область декомпрессора, адрес старта и гасить при распаковке экран (по желанию пользователя). По эффективности компрессии полностью аналогичен первой версии.

Версии программы PRESS :

(Первоначально разработана О.О.Бобковым, Санкт-Петербург)

Описание начинаю с версии 2.1, т.к. предыдущие версии не являлись полноценными. Сжатию подлежат лишь файлы, работающие с адреса 0100h !

Press v(2.1) Бобков О.О. Апрель 1992

Работает в МОНИТОРе. Процесс упаковки отображается в динамике, в виде двух растущих столбиков (старый и новый об'ем) и побайтного отображения сжимаемого участка (как при загрузке в начальный загрузчик "Вектора"). Это прибавляет программе привлекательности,тем более, что упаковка происходит отнюдь не мнгновенно, как в других упаковщиках. Время сжатия нашего "теста"-1мин. 13сек., однако теперь упакованный модуль занимает 4С блоков (!). Распаковщик разворачивает программу за 1.5сек.

Пользователям, использующим в работе эту или другие версии PRESSa, необходимо помнить о том, что по окончании упаковки область памяти от конца сжатого модуля до начала PRESSa очищается (в данной версии- до 83FFh), так что лучше не размещать в этой области другие служебные программы.

Press v(3.0) Бобков О.О. Октябрь 1992

Существует в двух вариантах: для работы в МОНИТОРе и для работы в дисковом отладчике типа XSID. Для обоих вариантов размер сжимаемого файла не должен превышать 33.5Кб, а работоспособность сжатого модуля размером более 29Кб не гарантируется. В связи с этим не рекомендую сжимать файлы, уже сжатые (это не касается перечисленных выше упаковщиков,кроме PRESS; их сжатые модули тоже эффективно сжимаются "сверху" еще раз. Не стоит лишь забывать, что с каждой упаковкой "сверху" общее время распаковки увеличивается) другими упаковщиками или закодированные (подобно кодировке в играх Г.Б.Луппова).

Упаковал "тест" до 48 блоков за 3мин.33сек. Распаковка - 1.8сек.

Press v(3.1) Бобков О.О. и Перлин В.А. Ноябрь 1992

Это первая (или одна из первых-другие мне пока не известны) ДИСКОВАЯ версия упаковщика ROM-COM-файлов для "Вектора" (именно так, ибо чисто дисковые, машинно не ориентированные CP/M-овские упаков-

щики появились значительно раньше на других компьютерах. Примером таких программ можно назвать упаковщик, входящий в состав оболочки NSWEEP; о нем будет сказано ниже).
 Работать способна на "Векторе" в системах любой конфигурации, в любых версиях BIOS. По окончании упаковки записывает сжатый модуль со старым именем на текущий диск,а исходному файлу присваивает расширение BAK. Эффективность упаковки равноценна версии 3.0.

Press v(3.2) Бобков О.О. Ноябрь-декабрь 1992

 Несколько изменен алгоритм архивации, за счет чего уменьшилось время упаковки ("тест" был сжат до 48 блоков за 2 минуты 30сек.). Появилась возможность прерывать процесс упаковки (УС/С). Если файл уже был сжат ранее этим упаковщиком, то об этом будет выдано соответствующее сообщение. Эффективность сжатия та же, что и в 3.1, но скорость распаковки увеличена (в нашем случае-1.4 сек.).

 Скоро ждите следующие версии : для ДОС и для МЛ (в виде копировщика).

* * *

LZ77.COM Луппов Г.Б. Январь 1993

 Алгоритм упаковки аналогичен "Press v(3.1)". Эффективность сжатия та же. Обрабатывает файлы любого размера. Создает самораспаковывающиеся файлы. Адаптирует их с 100H адреса. Позволяет гибко выбирать параметры сжатия, что позволяет найти оптимальный вариант. Программа Fatax была сжата с 32 кбт до 16.

 Говоря обо всех версиях, вместе взятых, стоит отметить,что результатом работы упаковщика является упакованный модуль, а не программа. Как правило, в начале каждого такого модуля находится распаковщик; его размер колеблется в пределах FFh, за исключением "АРХИВА 1:2"(его распаковщик занимает около четырех блоков) и "Unpack"(в его модулях нет распаковщика).
 Проблему распаковки модуля без его запуска иногда легко решить, используя собственный распаковщик модуля. Сравнительно просто можно развернуть в МОНИТОРе, а тем более-в ДОС программу, упакованную в одном из PRESSов, правильно распределив память и вовремя перехватив передачу управления от распаковщика программе. Это не сложно, попробуйте.

* * *

Об упаковщиках, работающих только в МикроДОС (CP/M).

 На сегодняшний день мне известен только один такой архиватор, он входит в состав сервисной оболочки для работы с файлами NSWEEP. Отличен своей повышенной эффективностью при сжатии текстовых файлов.

NSWEEP v2.05 Daiv Rand Edmonton, Alberta 04.11.1984

 Нажав при работе в NSWEEP клавишу [Q], Вы можете упаковать или распаковать файл (файлы), отмеченный командой [T]. Распаковка не должна производиться на текущий диск в текущий USER, поэтому система попросит ввести диск и USER(по умолчанию 0), куда произвести распаковку.
 Если Вам не известно, запакован отмеченный файл или нет, нажмите

после [Q] клавишу [R](Reverse), будет выявлено, запакован ли файл,
и произведено обратное действие (распаковка либо упаковка).
 Подробное опиисание на английском и краткое - на русском языке име-
ется.

NSWEEP v1.2   Daiv Rand   Edmonton, Alberta 11.10.1985

 Эта версия тоже снабжена своим упаковщиком, который, однако не хо-
чет работать в BIOS 3.1. В отличии от v2.05, здесь все на немецком
языке.
                    *   *   *

 Уважаемые господа программисты! Не ленитесь, снабжайте свои архи-
ваторы дезархиваторами и подробными описаниями! Лучше лишний часик
посидеть, чем заставить рядовых хаккеров лишний раз разбираться в
ваших произведениях.
                    Успехов вам !
                    *   *   *
                                Федор Мершиев
                        08.02.1993 * Санкт-Петербург * Колпино

            Печать экрана 512*256.

```
;*************************************************************************
;Эта подпрограмма позволяет получать твердую копию экрана при работе в режиме*
;512x256 тип печатающего устройства - EPSON (CM 6337). Драйвер предназначен  *
;для ОС МикроДОС, но после минимальных переделок будет работать и в других   *
;системах.                                                                    *
; (c) Усков И.М.    Волгоград          1992-1993г                            *
;*************************************************************************
;
            PUBLIC PSCR
;
PSCR:
            DI
            LXI   H,0
            DAD   SP
            SHLD  STK
            LXI   SP,STACK
            XRA   A
            OUT   10H
            ;
            MVI   C,1BH
            CALL  0F80FH ;по этому адресу расположена п/п вывода байта
                         ;на принтер. Она очень простая - убедитесь сами
            MVI   C,'3'
            CALL  0F80FH
            MVI   C,24
            CALL  0F80FH
            ;
            LXI   H,GRBUF
            SHLD  PBUF
            MVI   A,32
            STA   LINE
            MVI   H,0C0H
```

```
            LDA   0FFABH ;здесь в МикроДОСе хранится значение засылаемое
            MOV   L,A    ;в регистр скроллинга, т.е номер байта, с кото-
                         ;рого контроллер дисплея сканирует экран.
PS1:
            PUSH  H
            ;
            LXI   D,WORK
            MVI   C,8
            CALL  MOVER
            MOV   A,H
            SUI   20H
            MOV   H,A
            MOV   A,L
            MVI   C,8
            ADD   C
            MOV   L,A
            CALL  MOVER
            ;
            LHLD  PBUF
            MVI   C,8
B16:
            LXI   D,WORK
            CALL  CARR
            INX   H
            CALL  CARR
            INX   H
            DCR   C
            JNZ   B16
            SHLD  PBUF
            ;
            POP   H
            INR   H
            MVI   A,0E0H
            CMP   H
            JNZ   PS1
            ;
            PUSH  H
            MVI   C,1BH
            CALL  0F80FH
            MVI   C,'*'
            CALL  0F80FH
            LDA   REG
            MOV   C,A
            CALL  0F80FH
            MVI   C,0
            CALL  0F80FH
            MVI   C,2
            CALL  0F80FH
            ;
            LXI   H,GRBUF
            SHLD  PBUF
            LXI   D,512
PRINT:
            MOV   A,M
INV:
            XRI   000H   ;если нужны белые символы на черном фоне,
                         ;замените 000 на 0FF.
            MOV   C,A
            CALL  0F80FH
```

```
            INX   H
            DCX   D
            MOV   A,D
            ORA   E
            JNZ   PRINT
            MVI   C,0DH
            CALL  0F80FH
            MVI   C,0AH
            CALL  0F80FH
            ;
            POP   H
            MVI   H,0C0H
            MOV   A,L
            SUI   8
            MOV   L,A
            LDA   LINE
            DCR   A
            STA   LINE
            JNZ   PS1
            ;
            MVI   A,23H
            OUT   10H
            LHLD  STK
            SPHL
            EI
            RET
            ;
CARR:
            MVI   B,8
CARR1:
            LDAX  D
            RAL
            STAX  D
            MOV   A,M
            RAL
            MOV   M,A
            INX   D
            DCR   B
            JNZ   CARR1
            RET
;
MOVER:
            MOV   A,M
            STAX  D
            INX   D
            DCR   L
            DCR   C
            JNZ   MOVER
            RET
            ;
PBUF:       DW    0000
STK:        DW    0000
LINE:       DB    32
WORK:       DS    16
REG:        DB    6     ;это номер графического режима
GRBUF:      DS    512
STACK       EQU   GRBUF+530
            END
```

" Б А Й Т - 1 9 "

Киров, 1993
информационно-рекламный выпуск
центра "БАЙТ"
для пользователей ПК "Вектор-06Ц"

+++
+ Наш адрес: г. Киров, Октябрьский пр-т, 118, м-н "Дом Радио" +
+ Адрес для писем" 610006, г.Киров, а/я 1248, Зубкову А.Н. +
+++

Для любителей Ассемблера

```
            TITLE  KDSWAP
;
;Этот файл содержит две подпрограммы, которые помогут вам организовать
;обмен между квазидиском и экранными плоскостями не прибегая к функци-
;ям МикроДОС. Таким образом Вы можете создавать игры с выходом в ДОС,
;подгружать уровни с диска и т.п.
;Подпрограммы получены путем переработки соответствующих подпрограмм
;дискового бейсика v 2.5 Усковым И.М.
;
;      ПОДПРОГРАММА TESTSPACE
;
;
;ЭТА ПОДПРОГРАММА ПРОВЕРЯЕТ, СВОБОДЕН ЛИ 4-й БАНК КВАЗИДКА И, ЕСЛИ НЕТ,
;ВЫХОДИТ В ДОС ПЕЧАТАЯ НА ЭКРАНЕ ПРЕДУПРЕЖДЕНИЕ "КВАЗИДИСК !!!".
;В ЭТОМ СЛУЧАЕ НЕОБХОДИМО УДАЛИТЬ ФАЙЛЫ С КВАЗИДИСКА.
;
;В Н И М А Н И Е !
;ЭТО ЖЕ ПРЕДУПРЕЖДЕНИЕ МОЖЕТ ПОЯВИТЬСЯ И ПРИ СБОЯХ НА ДИСКЕ А !
;ПОМНИТЕ ОБ ЭТОМ.
;
;ПЕРЕД ИСПОЛЬЗОВАНИЕМ RAMSWAP ЖЕЛАТЕЛЬНО ИСПОЛЬЗОВАТЬ ЭТУ ПОДПРОГРАММУ
;ВО ИЗБЕЖАНИЕ ПОРЧИ ФАЙЛОВ НА КВАЗИДИСКЕ
;
       PUBLIC TESTSPACE
;
TESTSPACE:
       MVI    C,019H
       CALL   BDOS
       PUSH   PSW
       MVI    C,018H
       CALL   BDOS
       CPI    004H
       MVI    E,000H
       JC     VER1
VER2:  MVI    E,002H
VER1:  MVI    C,00EH
       CALL   BDOS
       ORA    A
       JNZ    SPICBAD
       MVI    C,01FH
       CALL   BDOS
       LXI    D,00005H
```

```
       DAD    D
       MOV    E,M
       INX    H
       MOV    D,M
       XCHG
       SHLD   @MEM1
       MOV    A,H
       ORA    A
       JNZ    VER2
       MVI    C,01BH
       CALL   BDOS
       INX    H
       MOV    A,L
       ORA    H
       JZ     SPICBAD
       LXI    B,00019H
       DAD    B
       LXI    D,@MEM2
       MVI    C,004H
@LOOP1:      LDAX   D
       INX    D
       ANA    M
       INX    H
       JNZ    SPICBAD
       DCR    C
       JNZ    @LOOP1
       LHLD   @MEM1
       LXI    B,0FFE8H
       DAD    B
       XCHG
       MOV    M,D
       DCX    H
       MOV    M,E
       POP    PSW
       MOV    E,A
       MVI    C,00EH
       CALL   BDOS
       ORA    A
       JNZ    SPICBAD
;
       RET
;
SPICBAD:
       LXI    D,BADDSK
       MVI    C,009H
       CALL   BDOS
       RST    0
;
@MEM1: DW 0000
BADDSK:
       DB 0DH,0AH,'К',7,'В',7,'А',7,'З',7,'И',7,'Д',7,'И',7,'С',7
       DB 'К',7,'!!!',7,'$'
@MEM2:
       DB 0FH,0FFH,0FFH,0F0H,00
;
;
;            ПОДПРОГРАММА RAMSWAP
;
;ЭТА ПОДПРОГРАММА ОБМЕНИВАЕТ ЭКРАННУЮ ПЛОСКОСТЬ, НОМЕР КОТОРОЙ УКАЗАН В
```

```
;РЕГИСТРЕ "B" НА 8-ми КИЛОБАЙТОВЫЙ БЛОК 4-го (ПОСЛЕДНЕГО) БАНКА КВАЗИДИСКА,
;НОМЕР КОТОРОГО (0-3) ЗАДАЕТСЯ В РЕГИСТРЕ "C"
;
;СОДЕРЖИМОЕ РЕГИСТРА "B":       ЭКРАННАЯ ПЛОСКОСТЬ:
;                  0                  8000H-9FFFH
;                  1                  A000H-BFFFH
;                  2                  C000H-DFFFH
;                  3                  E000H-FFFFH
;
;      ПРИ ОБМЕНЕ С ПЛОСКОСТЬЮ 3 НЕ ЗАБУДЬ ЗАПРЕТИТЬ ПРЕРЫВАНИЯ !
;
;      ПОДПРОГРАММА СОХРАНЯЕТ УКАЗАТЕЛЬ СТЕКА И НЕ СОХРАНЯЕТ РЕГИСТРЫ
;
;      ПО ВЫХОДУ ПОДПРОГРАММА ПОСЫЛАЕТ 23H В ПОРТ 10H Т.Е. ДЛЯ МикроДОС
;
;      ПОДПРОГРАММА "ВЫТАЩЕНА" ИЗ BASIC-ДИСКа Усковым И.М. 11/12/92
;
       PUBLIC RAMSWAP
;
RAMSWAP:
       LXI    H,00000H
       DAD    SP
       SHLD   MEMSTACK     ;Сохранить текущий указатель стека
       MOV    A,B
       ANI    003H
       LXI    H,TABSCR
       ADD    L
       MOV    L,A
       MOV    D,M
       MVI    E,000H       ;в DE - адрес выбранной плоскости
       MOV    A,C
       ANI    003H
       ADD    A
       LXI    H,TABMEM
       ADD    L
       MOV    L,A
       MOV    A,M
       STA    BLOCKCHK
       INX    H
       MOV    H,M
       MOV    L,E
       SPHL
       MVI    A,010H
       OUT    010H         ;работаем с четвертым блоком в режиме "стек"
       LXI    B,00040H     ; 8 килобайт это 40h блоков по 128 байт
       LXI    H,TABCHECK
       SHLD   LDR+1
       XCHG
NEXTBLK:                   ;обмен 128 байт квазидиск - плоскость
       POP    D
       MOV    A,M
       MOV    M,E
       MOV    E,A
       ADD    B
       MOV    B,A
       INX    H
       MOV    A,M
       MOV    M,D
       MOV    D,A
```

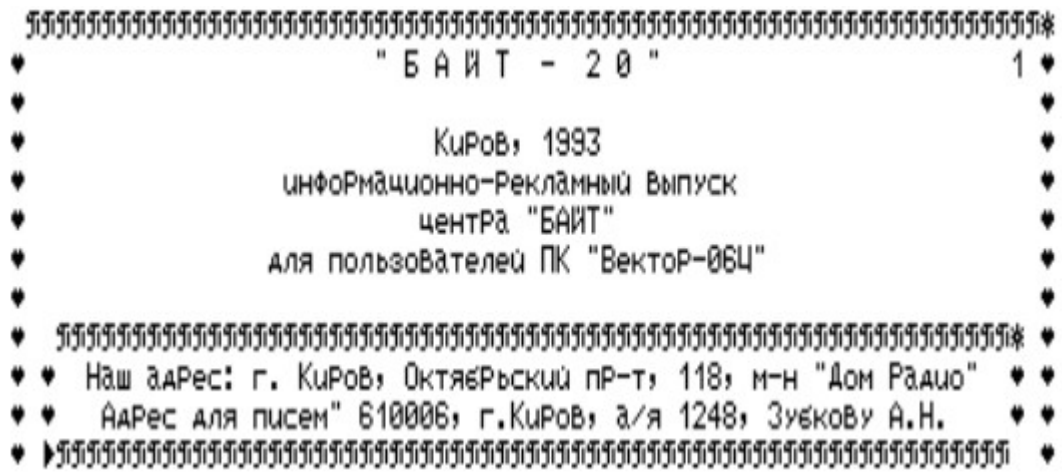
```
       ADD    B
       MOV    B,A
       INX    H
       PUSH   D
       POP    D
       MOV    A,L
       ANI    07FH
       JNZ    NEXTBLK
       XCHG
LDR:
       LXI    H,000        ; НА САМОМ ДЕЛЕ ЗДЕСЬ НЕ 0 ! (см ниже)
       MOV    M,B          ; записать контрольную сумму для 128 байт в
       INX    H            ; таблицу и увеличить указатель на 1
       SHLD   LDR+1
       XCHG
       MOV    B,A
       DCR    C            ; все блоки обработанны ?
       JNZ    NEXTBLK            ; нет, продолжать
       LHLD   BLOCKCHK
       DAD    H
       DAD    H
       DAD    H
       DAD    H
       SPHL
       MVI    A,03FH       ; нулевой банк в двух режимах
       OUT    010H
       LXI    H,TABCHECK   ; далее запись в ЭД контрольных сумм
       MVI    B,040H
NEXTCHCK:
       MOV    D,M
       INX    H
       MOV    E,D
       PUSH   D
       DCR    B
       JNZ    NEXTCHCK
       MVI    A,023H
       OUT    010H
       LHLD   MEMSTACK     ; восстановить стек
       SPHL
       RET
;
; ТАБЛИЦЫ:
TABSCR:
       DB 80H,0A0H,0C0H,0E0H
TABMEM:
       DB 0ECH,000,0E4H,020H,0DCH,040H,0D4H,060H
BLOCKCHK:
       DB 000,0FH
TABCHECK:
       DS     40H    ; контрольные суммы
MEMSTACK:
       DW 0000            ; здесь хранится адрес стека
@MEM3:
       DW 0000
@MEM4:
       DW 0000
BDOS   EQU    00005H
       END
```

```
;УПРАЖНЕНИЕ 1

;НЕПОСРЕДСТВЕННАЯ ЗАГРУЗКА РЕГИСТРОВ
НАЧ:    MVI  A,12     ;ЗАГРУЗИТЬ ДЕСЯТИЧНОЕ ЧИСЛО
        MVI  C,0CH    ;ЗАГРУЗИТЬ ШЕСТНАДЦАРИТИЧНОЕ ЧИСЛО
        LXI  H,110FH  ;ЗАГРУЗИТЬ ДАННЫЕ В ПАРУ РЕГИСТРОВ
        LXI  D,MET1   ;ЗАГРУЗИТЬ АДРЕС В ПАРУ РЕГ. (DE)
;ПЕРЕМЕШЕНИЕ ИЗ РЕГИСТРА В РЕГИСТР
        MOV  C,E      ;В (C) ИЗ (E)
        MOV  B,D      ;В (B) ИЗ (D)
        MOV  A,H      ;В (A) ИЗ (H)
        XCHG          ;ОБМЕНИВАЕТ СОДЕРЖИМОЕ (DE) И (HL)
;ПРЯМАЯ ЗАГРУЗКА ИЗ ПАМЯТИ
        LHLD MET2     ;ЗАГРУЗИТЬ (HL) ИЗ ЯЧЕЕК MET2,MET2+1
        LDA  MET2     ;ЗАГРУЗИТЬ (A) ИЗ ЯЧЕЙКИ MET2
;КОСВЕННАЯ ЗАГРУЗКА АККУМУЛЯТОРА ИЗ ПАМЯТИ
        MOV  A,M      ;АДРЕС ЯЧЕЙКИ В ПАРЕ РЕГИСТРОВ (HL)
        LDAX D        ;АДРЕС ЯЧЕЙКИ В ПАРЕ РЕГИСТРОВ (DE)
        LDAX B        ;АДРЕС ЯЧЕЙКИ В ПАРЕ РЕГИСТРОВ (BC)
        JMP  НАЧ
MET1:   DB   0C3H     ;В ЯЧЕЙКУ MET1 ПОМЕЩАЕМ ЧИСЛО 0C3H
MET2:   DW   MET1     ;В MET2,MET2+1 АДРЕС ЯЧЕЙКИ MET1

;УПРАЖНЕНИЕ 2

;ЗАПОМИНАНИЕ И ЗАГРУЗКА РЕГИСТРОВ ИЗ ПАМЯТИ
НАЧ:    MVI  A,75H    ;ЗАГРУЗИТЬ ДАННЫЕ
        STA  MET1     ;ПРЯМОЕ ЗАПОМИНАНИЕ (A) В MET1
        LXI  H,3FA5H  ;ЗАГРУЗИТЬ ДАННЫЕ
        SHLD MET2     ;ПРЯМОЕ ЗАПОМИНАНИЕ (L) В MET2, (H) В MET2+1
;КОСВЕННОЕ ЗАПОМИНАНИЕ
        LXI  H,MET1   ;ЗАГРУЗИТЬ АДРЕС В (HL)
        MVI  C,35H    ;ЗАГРУЗИТЬ ДАННЫЕ В РЕГИСТР (C)
        MOV  M,C      ;ЗАПОМНИТЬ РЕГИСТР (C) В MET1
        MOV  A,M      ;ЗАГРУЗИТЬ (A) ИЗ ПАМЯТИ
        MVI  M,0FEH   ;ЗАПОМНИТЬ ДАННЫЕ В ПАМЯТИ
        MOV  A,M      ;ЗАГРУЗИТЬ (A) ИЗ ПАМЯТИ
        XCHG          ;ПЕРЕСЛАТЬ АДРЕС ИЗ (HL) В (DE)
        STAX D        ;ЗАПОМНИТЬ (A) ПО АДРЕСУ ИЗ (DE)
        MOV  C,E      ;ЗАГРУЗИТЬ АДРЕС В (BC
        MOV  B,D      ;
        MOV  A,H      ;КОСВЕННОЕ ЗАПОМИНАНИЕ РЕГИСТРА (H)
        STAX B        ;ПО АДРЕСУ ИЗ ПАРЫ РЕГИСТРОВ (BC)
        JMP  НАЧ
MET1:   DS   1        ;РЕЗЕРВИРУЕМ 1 БАЙТ
MET2:   DS   2        ;РЕЗЕРВИРУЕМ 1 СЛОВО

;УПРАЖНЕНИЕ 3

;АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ
НАЧ:    MVI  A,0C2H   ;ЗАГРУЗИТЬ ДАННЫЕ В (A)
        LXI  B,6E12H  ;ЗАГРУЗИТЬ ДАННЫЕ В (BC)
        ADD  B        ;(A) <= (A) + (B)
        ADC  C        ;(A) <= (A) + (C) + ФЛАГ ПЕРЕНОСА
        LXI  H,12F5H  ;ЗАГРУЗИТЬ ДАННЫЕ ДЛЯ
        LXI  D,200EH  ;ШЕСТНАДЦАРИТИЧНОГО СЛОЖЕНИЯ
        DAD  D        ;(HL) <= (HL) + (DE)
```

```
        DAD  B         ;(HL) <= (HL) + (BC)
        MVI  A,38H     ;ДЕСЯТИЧНОЕ СЛОЖЕНИЕ
        ADI  15H       ;(A) <= 38H + 15H
        DAA            ;КОРРЕКТИРОВАТЬ В ДЕСЯТИЧНЫЙ ВИД
;ШЕСТНАДЦАРИТИЧНОЕ ВЫЧИТАНИЕ (HL) <= (HL) - (DE)
        MOV  A,L       ;ВЫЧИТАЕМ МЛАДШИЕ БАЙТЫ L=L-E
        SUB  E         ;ЕСЛИ (E) > (L), ТО ФЛАГ ПЕРЕНОСА=1
        MOV  L,A       ;ЗАПОМИНАЕМ МЛАДШИЙ БАЙТ РЕЗУЛЬТАТА
        MOV  A,H       ;ВЫЧИТАЕМ СТАРШИЕ БАЙТЫ С УЧЕТОМ ПЕРЕНОСА
        SBB  D         ;(A) <= (A) - (D) - ФЛАГ ПЕРЕНОСА
        MOV  H,A       ;ЗАПОМИНАЕМ СТАРШИЙ БАЙТ РЕЗУЛЬТАТА
        JMP  НАЧ

;УПРАЖНЕНИЕ 4

;ЛОГИЧЕСКИЕ ОПЕРАЦИИ
НАЧ:    MVI  A,95H     ;ЗАГРУЗИТЬ ДАННЫЕ
        ORI  20H       ;ЛОГИЧЕСКОЕ "ИЛИ"
        XRI  63H       ;ИСКЛЮЧАЮЩЕЕ "ИЛИ"
        RAL            ;СДВИГ ВЛЕВО ЧЕРЕС ПЕРЕНОС
        ANI  0F0H      ;ЛОГИЧЕСКОЕ "И"
        RLC            ;ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО
        XRI  42H       ;ИСКЛЮЧАЮЩЕЕ "ИЛИ"
        RRC            ;ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО
        CMA            ;ИНВЕРТИРОВАТЬ (A)
        RAR            ;СДВИГ ВПРАВО ЧЕРЕЗ ПЕРЕНОС
        MOV  C,A       ;ЗАГРУЗИТЬ РЕГИСТР (C)
        STC            ;УСТАНОВИТЬ ФЛАГ ПЕРЕНОСА CY=1
        XRA  A         ;ОЧИСТИТЬ ФЛАГ ПЕРЕНОСА CY=0
        CMC            ;ИНВЕРТИРОВАТЬ ФЛАГ ПЕРЕНОСА
        XRA  C         ;ЛОГИЧЕСКАЯ ОПЕРАЦИЯ С РЕГИСТРОМ
        LXI  H,МЕТ1    ;ЗАГРУЗИТЬ АДРЕС
        ORA  M         ;ЛОГИЧЕСКАЯ ОПЕРАЦИЯ С ПАМЯТЬЮ
        JMP  НАЧ
МЕТ1:   DB   0F0H      ;ОПРЕДЕЛИТЬ БАЙТ В ПАМЯТИ

;УПРАЖНЕНИЕ 5

;ОПЕРАЦИИ СО СТЕКОМ
НАЧ:    LXI  SP,75F0H ;ЗАГРУЗИТЬ УКАЗАТЕЛЬ СТЕКА
        LXI  B,8513H  ;ЗАГРУЗИТЬ ДАННЫЕ В (BC)
        PUSH B         ;ЗАПОМНИТЬ (BC) В СТЕКЕ
;УКАЗАТЕЛЬ СТЕКА УМЕНЬШИЛСЯ НА 2
        POP  D         ;ЗАГРУЗИТЬ (DE) ИЗ СТЕКА
;УКАЗАТЕЛЬ СТЕКА УВЕЛИЧИЛСЯ НА 2
        MOV  H,D       ;ПОДГОТОВИТЬ ДАННЫЕ ДЛЯ
        PUSH B         ;ДЕМОНСТРАЦИИ КОМАНДЫ XTHL
        XTHL           ;ОБМЕН (HL) И ВЕРШИНЫ СТЕКА
        XTHL           ;ОБМЕН (HL) И ВЕРШИНЫ СТЕКА
        POP  B         ;ВОССТАНОВИТЬ СОСТОЯНИЕ СТЕКА
        MOV  A,B       ;ЗАГРУЗИТЬ (A) ДЛЯ НАГЛЯДНОСТИ
;ВОЗМОЖНОСТЬ ЗАГРУЗИТЬ РЕГИСТР ФЛАЖКОВ "F"
        PUSH PSW       ;(A) И (F) В СТЕК
        POP  H         ;(H) <= (A), (L) <= (F)
        LXI  H,0       ;ПОДГОТОВИТЬ ПАРУ РЕГИСТРОВ (HL)
        DAD  SP        ;В (HL) ПОЛУЧИТЬ АДРЕС СТЕКА
        LXI  SP,0      ;ОЧИСТИТЬ РЕГИСТР СТЕКА
        SPHL           ;ЗАГРУЗКА АДРЕСА СТЕКА ИЗ (HL)
        JMP  НАЧ
```

```
;УПРАЖНЕНИЕ 6

;ВЕТВЛЕНИЕ ПРОГРАММЫ
НАЧ:    MVI  B,32H    ;ЗАГРУЗИТЬ ДАННЫЕ В (B)
M1:     ADI  85H      ;ГЕНЕРИРУЕМ ЧИСЛА В АККУМУЛЯТОРЕ
        CPI  20H      ;(A) БОЛЬШЕ 20H ?
        JC   M1       ;ЕСЛИ МЕНЬШЕ,ТО ИДТИ НА M1
        CPI  40H      ;(A) БОЛЬШЕ 40H ?
        JNC  M1       ;ЕСЛИ НЕ МЕНЬШЕ,ТО ИДТИ НА M1
        CMP  B        ;(A) РАВНО (B) ?
        JNZ  M1       ;ЕСЛИ НЕ РАВНО,ТО ИДТИ НА M1
M2:     MOV  C,A      ;СОХРАНИТЬ (A) В (C)
        ANI  4        ;ВЫДЕЛИТЬ РАЗРЯД 00000100B
        JNZ  M3       ;ЕСЛИ НЕ НУЛЬ,ТО ИДТИ НА M3
        ORI  4        ;УСТАНОВИТЬ РАЗРЯД В 1
        JMP  M2       ;ПОВТОРИТЬ ПРОВЕРКУ РАЗРЯДА
M3:     ORA  A        ;УСТАНОВИТЬ ФЛАГИ ПО АККУМУЛЯТОРУ
        JP   M6       ;ПРИМЕР ПЕРЕХОДА ПО ПЛЮСУ
M6:     INR  B        ;УСТАНОВИТЬ ФЛАГИ ПО РЕГИСТРУ
        DCR  B        ;СОХРАНИВ ЕГО ЗНАЧЕНИЕ
        JM   M7       ;ПРИМЕР ПЕРЕХОДА ПО МИНУСУ
M7:     XRA  A        ;ОЧИСТИТЬ АККУМУЛЯТОР
        JMP  НАЧ

;УПРАЖНЕНИЕ 7

;ОРГАНИЗАЦИЯ ЦИКЛОВ (I)
;ПРИМЕР ЦИКЛА ДО 256 ПОВТОРЕНИЙ
НАЧ:    MVI  C,5      ;УСТАНОВИТЬ СЧЕТЧИК ЦИКЛОВ
M1:     ADI  1        ;ПРИМЕР ТЕЛА ЦИКЛА
        DCR  C        ;УМЕНЬШИТЬ СЧЕТЧИК НА 1
        JNZ  M1       ;ПОВТОРИТЬ,ЕСЛИ (C) НЕ НУЛЬ
;ПРИМЕР ЦИКЛА: 16-РАЗРЯДНЫЙ СЧЕТЧИК
        LXI  B,0004   ;УСТАНОВИТЬ 16-РАЗРЯДНЫЙ СЧЕТЧИК
M2:     ADI  2        ;ПРИМЕР ТЕЛА ЦИКЛА
        DCX  B        ;УМЕНЬШИТЬ СЧЕТЧИК НА 1
        MOV  A,B      ;ПРОВЕРИТЬ 16-РАЗРЯДНЫЙ СЧЕТЧИК
        ORA  C        ;НА НУЛЬ
        JNZ  M2       ;ПОВТОРИТЬ,ЕСЛИ (BC) НЕ НУЛЬ
;ПРИМЕР ВЛОЖЕННОГО ЦИКЛА
        MVI  B,2      ;УСТАНОВИТЬ ВНЕШНИЙ СЧЕТЧИК
M3:     MVI  C,3      ;УСТАНОВИТЬ ВНУТРЕННИЙ СЧЕТЧИК
M4:     SUI  1        ;ПРИМЕР ТЕЛА ЦИКЛА
        DCR  C        ;УМЕНЬШИТЬ ВНУТРЕННИЙ СЧЕТЧИК
        JNZ  M4       ;ПОВТОРИТЬ,ЕСЛИ НЕ НУЛЬ
        JMP  НАЧ

;УПРАЖНЕНИЕ 8

;ОРГАНИЗАЦИЯ ЦИКЛОВ (II)
;ПРИМЕР ЦИКЛА ОТ (HL) ДО (DE)-1
НАЧ:    LXI  H,1100H  ;НАЧАЛЬНОЕ ЗНАЧЕНИЕ (HL)
        LXI  D,1105H  ;КОНЕЧНОЕ ЗНАЧЕНИЕ (DE)
M1:     MOV  A,M      ;ПРИМЕР ТЕЛА ЦИКЛА
        INX  H        ;УВЕЛИЧИТЬ ТЕКУЩЕЕ (HL)
        MOV  A,L      ;СРАВНИТЬ МЛАДШИЕ БАЙТЫ
        CMP  E        ;НА СОВПАДЕНИЕ
        JNZ  M1       ;ЕСЛИ НЕ РАВНЫ,ТО ИДТИ НА M1
```

```
        MOV  A,H      ;СРАВНИТЬ СТАРШИЕ БАЙТЫ
        CMP  D        ;НА СОВПАДЕНИЕ
        JNZ  M1       ;ЕСЛИ НЕ РАВНЫ,ТО ИДТИ НА M1
;ПРИМЕР ЦИКЛА ПРИ ОБРАБОТКЕ ТАБЛИЦЫ
;1 ЭЛЕМЕНТ ТАБЛИЦЫ - 1 БАЙТ,НЕ БОЛЕЕ 256 БАЙТ
        LXI  H,НАЧ    ;АДРЕС НАЧАЛА ТАБЛИЦЫ
M2:     MOV  A,M      ;ПРИМЕР ТЕЛА ЦИКЛА
        MOV  A,L      ;ПРОВЕРИТЬ НА КОНЕЦ ТАБЛИЦЫ
        CPI  НАЧ+6    ;АДРЕС ПОСЛЕДНЕГО ЭЛЕМЕНТА
        INX  H        ;АДРЕС СЛЕДУЮЩГО ЭЛЕМЕНТА
        JNZ  M2       ;ЕСЛИ НЕТ,ТО ОБРАБОТАТЬ
        JMP  НАЧ

;УПРАЖНЕНИЕ 9

;РАБОТА С ПОДПРОГРАММАМИ (I)
НАЧ:    CALL SB1      ;ВЫПОЛНЕНИЕ ПОДПРОГРАММЫ БЕЗ ПАРАМЕТРОВ
        LXI  H,2222H  ;ЗАГРУЗИТЬ 1 ПАРАМЕРТ
        LXI  D,5555H  ;ЗАГРУЗИТЬ 2 ПАРАМЕТР
        CALL SB2      ;ПЕРЕДАЧА ПАРАМЕТРОВ ЧЕРЕЗ РЕГ.
        CALL SB2      ;ДЕМОНСТРАЦИЯ ВЫХОДА ПО ФЛАГУ
        CALL SB3      ;ДЕМОНСТРАЦИЯ СОХРАНЕНИЯ РЕГ.
        JMP  НАЧ
SB1:    MVI  A,12     ;ПРИМЕР ТЕЛА ПОДПРОГРАММЫ
        RET           ;БЕЗУСЛОВНЫЙ ВЫХОД ИЗ ПОДПРОГРАММЫ
SB2:    MOV  A,D      ;СРАВНИТЬ (D) С (H)
        CMP  H        ;
        RC            ;ВЫХОД,ЕСЛИ (D) < (H)
        DAD  D        ;(HL) <= (HL) + (DE)
        RET
SB3:    PUSH H        ;СОХРАНИТЬ (HL)
        PUSH D        ;СОХРАНИТЬ (DE)
        ANA  A        ;ПРИМЕР ТЕЛА ПОДПРОГРАММЫ
        POP  D        ;ВОССТАНОВИТЬ (DE)
        POP  H        ;ВОССТАНОВИТЬ (HL)
        RET           ;ВЫХОД

;УПРАЖНЕНИЕ 10
;РАБОТА С ПОДПРОГРАММАМИ (II)
;ПЕРЕДАЧА ПАРАМЕТРОВ ПОДПРОГРАММЕ ЧЕРЕЗ ОЗУ
НАЧ:    LXI  H,1111H  ;ВЗЯТЬ ПЕРВЫЙ ПАРАМЕТР
        SHLD PR       ;ЗАПИСАТЬ В ПАМЯТЬ
        LXI  H,2222H  ;ВЗЯТЬ ВТОРОЙ ПАРАМЕТР
        SHLD PR+2     ;ЗАПИСАТЬ В ПАМЯТЬ
        ADI  1        ;ОРГАНИЗОВАТЬ В (A) СЧЕТЧИК
        CPI  2        ;ДЛЯ ДЕМОНСТРАЦИИ КОМАНД CC И CZ
        PUSH PSW      ;СОХРАНИТЬ (A) И ФЛАГИ
        CC   SB1      ;ВЫПОЛНИТЬ ПОДПРОГРАММУ,ЕСЛИ (A)=0 ИЛИ 1
        POP  PSW      ;ВОССТАНОВИТЬ (A) И ФЛАГИ
        CNC  SB1      ;ВЫПОЛНИТЬ ПОДПРОГРАММУ,ЕСЛИ (A) >=1
        CPI  3        ;СРАВНИТЬ СЧЕТЧИК С 3
        CZ   SB1      ;ВЫПОЛНИТЬ ПОДРОГРАММУ,ЕСЛИ СОВПАЛО
        ANI  3        ;ОГРАНИЧИТЬ СЧЕТЧИК
        JMP  НАЧ
SB1:    LHLD PR+2     ;ВЗЯТЬ ВТОРОЙ ПАРАМЕТР
        XCHG          ;ПЕРЕСЛАТЬ В (DE)
        LHLD PR       ;ВЗЯТЬ ПЕРВЫЙ ПАРАМЕТР В (HL)
        RET
PR:     DS   4        ;РЕЗЕРВИРУЕМ ОЗУ ПОД ПАРАМЕТРЫ
```



ROBBO

В игре, управляя роботом, нужно собрать заданное количество болтов, причем на каждом этаже их число различно. Можно двигать решетки и камни, на некоторых этажах можно проедать стенки.Всего 18 уровней.

Имея всего 3 жизни, сложно пройти всю игру.Для того, чтобы сделать бессмертие необходимо загрузить программу в Монитор и немного изменить, перебить несколько значений в ячейках с 9F2 по 9F5.

Наберите следующее:

```
#S9F2 <ВК>
9F2 3D 3F <ВК>
9F3 CA 3F <ВК>
9F4 B5 3F <ВК>
9F5 08 3F <ВК>
9F6 32 <F4>
```

Затем измененный вариант программы нужно выгрузить командой "O>1,42,0", предварительно задав ее имя.

Меринов Сергей, FMS, 1993 г.

PESHERA

Автор Пересадов О.

Управляя человечком, нужно, спасаясь от чертей, собрать все клады на каждом этапе. Игра очень динамичная, и поэтому с пятью жизнями Вы много не поиграете.

Бессмертие достигается незначительным изменением в программе.Для этого, естественно, нужно загрузить ее в Монитор и директивой "S" "исправить" содержимое всего одной ячейки.

```
#S46C <ВК>
46C 35 00 <ВК>
46D 7E <F4>
```

Затем нужно выгрузить программу директивой "O>" или,если хотите, сразу запустить ее.

Клавиша <ТАБ>-стрельба влево

Клавиша <ПС>-стрельба вправо

Клавиша <ЗБ>-выбор фона игры.

BOLDER-2

Все вы знаете эту красивую и интересную игру, одну из самых лучших игр на ВЕКТОРе. Это продолжение первой версии. Наверное, вам было очень досадно, когда вы, пройдя большую половину игры, вдруг обнаруживали, что кончились все пять жизней и приходиться начинать игру сначала.Не отчаивайтесь, после того, как вы прочитаете, что написано ниже, вы сможете пройти всю игру от начала до конца, не боясь того, что у вас кончатся жизни.
К сожалению, программа закодирована и запакована, но бессмертие в ней сделать все-таки можно. Для этого, как всегда, загрузите игру в Монитор-отладчик и измените содержимое всего одной ячейки директивой "S".

```
#SF2C <BK>
F2C AD 90 <BK>
F2D A3 <F4>
```

После этого вы можете спокойно выгрузить на магнитофон директивой "O>" и запустить. Спокойной вам игры!

Для тех, у кого есть распакованная версия программы скажу следующее:

```
5DF LXI D,0628  ; в ячейку FE3 заноситься
5E2 MOV A,D     ; количество жизней,
5E3 STA FE3     ; равное шести.

DC6 DCR A       ; уменьшение  жизней.
DC7 STA FE3     ;

5CC LDA 0028    ; начальная установка
5CF SUI C2      ; номера
5D1 STA 85C     ; этажа.
```

Но, будьте внимтельны, в игре всего 10 этажей.
Успехов вам!

Меринов Сергей, FMS, 1993 г.

BOLDER-M

Продолжение BOLDERa-2. В игру включены новые персонажи, которые делают эту игру еще интереснее, красивее и умнее. Игра тоже запакована, как и ее предшественница, но бессмертие сделать можно,конечно, хотя изменить придется содержимое не только одной ячейки. Программу, как вы уже догадались,нужно загрузить в Монитор и немного поработать директивой "S", хотя, я думаю,вам это будет только в радость.

```
#S10CF <BK>
10CF 5B 66 <BK>
```

```
  10D0 8B <F4>
  #S1144 <BK>
  1144 F0 66 <BK>
  1145 DE 66 <BK>
  1146 71 66 <BK>
  1147 E2 <F4>
```

А у тех, у кого имееться распакованная версия игру, у меня есть информация.

```
  6A9 LDA 0038 ; установка
  6AC SUI C2   ; номера
  6AE JMP 6D2  ; этажа.
     ......    ;
  6D2 STA 1A1  ;

  6DB MVI A,F7 ; устанавливается количество
  6DD STA 100  ; жизней, равное шести.
  6E0 SUI F1   ;
  6E2 STA 1E0  ;-ячейка жизней

  F94 DCR A    ; уменьшение
  F95 STA 01E0 ; жизней.
```

Но, если вы в программе измените хоть один байт,то она у вас повиснет. Это происходит потому, что с адрессов 10C8 и FF0 находятся подпрограммы,которые подсчитывают контрольную сумму всей программы, первая с адреса 1C48 по адрес 6440, а вторая с адреса 271 по 1BF7. Нейтрализовать подпрограммы можно описанным ниже способом:

```
  #S1112 <BK>
  1112 C2 00 <BK> ; \
  1113 26 00 <BK> ;  JNZ 1926
  1114 19 00 <BK> ; /

  и

  #S1009 <BK>
  1009 C2 00 <BK> ; \
  100A 3D 00 <BK> ;  JNZ 173D
  100Б 17 00 <BK> ; /
```

Удачи вам!

Меринов Сергей, FMS, 1993 г.

А знаете ли вы?

S P A C E

Игра, открывающая серию игр "Звездные войны". В далеком космосе идет война 2-х цивилизаций. Вражеская база регулярно выпускает ко-

рабли убийцы. Их задача - стереть вас в порошок. Но поскольку ваш корабль им не уничтожить, они атакуют вашу базу и перехватывают транспорты с помощью, обрекая вас на энергетический голод. Вы можете выбрать любую таткику - охранять свою базу, быстро атаковать вражескую, или продвигаясь к вражеской, перехватывать тарелки главное победить. Но управление мощным космическим крейсером, сами понимаете, дело сложное.
Полная имитация 3-хмерного пространства. Инструкция: Маршевый двигатель - F1/F5, ориентационные двигатели - курсор,защита - <Я>, ракеты - <Ч>, лазер - <С>.

R A N G E R

Вам, известному диверсанту, поручается уничтожить вражеские огневые точки. Для этого вы готовите закладки (3шт.) из тех предметов, которые сочтете нужными, в тылу врага. На самолете вас доставят на место. При полете не зевайте и сбрасывайте припасы в удобных местах, а затем прыгайте сами. После приземления действуйте быстро, т.к. через неколько минут за вами прибудет вертолет и вы должны быть к тому времени в условленном месте (коричневый квадрат). Вражеская земля напичкана минами и прочесывается пулями.
Управление оружием на клавишах: <Я> - автомат, <Ч> - гранатомет, <С> - граната, <М> - фугас, <АР2> - карта.

S N A K E

Змея проделывает ходы, а мангусты ловят ее по этим ходам. С'есть можно только "нерадивого" мангуста (красн.цвета).

S P A C E S A F A R Y

Путешествие в космосе. На зеленых планетах водятся гнусные насекомые. Надо найти эти планеты, уничтожив охрану, приземлиться, на поверхности планеты найти вход в пещеру,избегая контактов с метеоритами, смерчами и минами, и лишь затем ловко набросить сеть на паразита. Он тоже не безобиден - точно брызжет ядовитой слюной. Семь попаданий - и оболочка коробля рассыпается в пыль.

P U S H

Призраки продят по игровому полю.Единственный способ избавится от них (и перейти на следующий уровень) -придавить их камнями. Будьте внимательны, камней не намного больше, чем привидений. Толкать их можно при помощи кл. <РУС/LAT> и "стрелок".

T O W E R

Вы попали в заповедник старых замков,в которых полно сокровищ. Но стоители этих замков очень хитры, и постарались максимально затруднить путь тем, кто захочет поживиться сокровищами. Одичавшие обитатели, хищные птицы, ловушки, лифты...

Назаров А.Е. г.Киров 1993г.

```
            ************
            * T O W E R *
            ************
```

Управляя главным героем этой игры, вам необходимо собрать все камни, находящиеся на башне, после сбора которых вы сможете перейти на следующий уровень.

Во время игры вам будут мешать птицы и шарики. Взяв зонтик вы сможете защититься от птиц. Кроме того ваша скорость движения будет постепенно уменьшаться, но взяв маленькую'птичку'скорость восстановится.

В башне есть тунели,войдя в один из них вы появляетесь в другом. На некоторых уровнях камни расположены высоко и вам их не достать,значит вам нужно искать лестницу. Включить или выкл.музыку клавиша'AP-2'.

```
            ******************
            *C Y B E R N O I D*
            ******************
```

В этой игре вам нужно пройти пять уровней, каждый из них имеет девять комнат.Во время игры вам будут мешать роботы, столкнувшись с которыми вы теряете жизнь, но против них у вас есть ракеты 'ЗБ', а также защита, которую вы можете взять во время игры.

Кроме того у робота имеется еще два вида оружия - это: 'AP2'невидимость и 'СТР'-шарики, которыми можно уничтожить "улей", выбрав оружие, нажмите 'ПС' для его запуска.

В процессе игры вы имеете возможность пополнить запас оружия, взяв контейнер с ним.

Если перед взятием контейнера вы нажмете 'AP2' или 'СТР', то у вас пополнится запас невидимости или шариков соответственно.

В заставке, нажав цифру 3-"PASSWORD" и набрав название уровня, вы начнете с него играть !

Названия уровней:1-"START"
2-"ROOM"
3-"KISS"
4-"LEFT"
5-"CLEAR"

```
            <=============>
            < R A N G E R >
            <=============>
```

В начале игры вы попадаете на склад, котором выбираете себе оружие.

Виды оружия:1-обойма с патронами
2-мед.аптечка
3-ракета
4-ручная граната
5-часовая мина

Если вас ранили и у вас есть аптечка, то нажав клавишу 'F1' жизнь восстанавливается, если же аптечки нет, вы погибаете.

Ракета необходима для уничтожения дота. Клавиша - "^".
Ручной гранатой можно уничтожить препятствие на пути. Клавиша - "S".
Если у вас кончился запас ракет и есть мина, то подойдя к доту нажмите клавишу "M" и подальше отойдите, через несколько секунд дот взорвется.
Стрельба из автомата - клавиша "Q".
Затем наведите стрелку на 'EXIT' и нажмите пробел. В верху экрана появляется самолет, с которого вам предстоит сбросить три тюка с запасом оружия (только не на болото), взяв которые у вас восстанавливается то количество оружия, которое вы выбрали на складе. Затем спрыгнуть самому. Прыжок и сброс тюков осуществляется клавишей-"ПРОБЕЛ". После удачного прыжка в левом углу появляется табличка, показывающая запас оружия.Нажав клавишу-"AP2", вы появляетесь на поле боя.
Добравшись до дота, уничтожьте его, а затем идите на взлетное поле, где вскоре вас заберет вертолет.
После каждой удачно завершенной миссии вас повысят в звании.

```
#############
# R O B B O #
#############
```

Загрузите игру в Монитор SM, найдите ячейку с адресом 0827, в ней должно быть записано 'MVI A,03', исправьте ее на 'MVI A,F0'. После этого наберите O>1,M,0 ВК и выгрузите программу. M-количество блоков.

ПРОГРАММА

```
========================
=Ц В Е Т О  М У З Ы К А=
========================
```

Вашему вниманию предлагается простая программа в формате Редактора-ассемблера, которая переводит звуковые сигналы, поступающие с магнитофона в цветные полосы на экране!

```
 DI
 MVI A,89H
 OUT 00
 LXI H,8000H
M1: MVI M,00
 INX H
 MOV A,H
 ORA L
 JNZ M1
M3: MVI C,04
M2: IN 01
 RLC
 DCR C
 JNZ M2
 OUT 02
 JMP M3
```

Г.Киров Шилин Д.В.

РЕСТАРТ MDOS ЧЕРЕЗ СБР+БЛК

Многие пользователи MDOS (независимо от версии MDOS1 или MDOS2, т.к. в данном случае это безразлично) страдают от того, что их ОС не перезапускается при нажатии СБР+БЛК. Это возникает из-за несоответствия между MDOS и схемой квази-диска (в дальнейшем КД). При одновременном нажатии на клавиши БЛК и СБР сигнал СБРОС с системной шины АС41 (по схеме компьютера) выключает КД, т.к. сбрасывает в 0 D7,D8,D10 (по схеме КД). Поэтому при рестарте необходимо включить КД перед тем,как сделать переход по стартовому адресу (ячейки 0001 и 0002), иначе переход будет произведен не в память КД,а в экранное ОЗУ,что приводит к "зависанию" компьютера. Поэтому по адресу 0000 надо вместо кода 0С3Н записывать код 0F7H (RST 6), что при рестарте приведет к переходу по адресу 0030Н и включению КД.
В MDOS вектора переходов устанавливаются следующим фрагментом.

для MDOS2:	для MDOS1:
D365 MVI A,C3	DB65 MVI A,C3
D367]TA 0000	DB67 STA 0000
D36A LXI H,B903	DB6A LXI H,C103
D36D SHLD 0001	DB6D SHLD 0001
D370 STA 0005	DB70 STA 0005
D373 LXI H,B900	DB73 LXI H,C100
D376 SHLD 0006	DB76 SHLD 0006

Но для нас это безразлично,т.к. MDOS самонастраивающаяся система и мы будем менять не настроенный код, который идентичен для обеих версий MDOS. Вот его текст при загрузке в SID с адреса 100Н (для этого наберите командную строку A>SID OS.COM<ВК>):

```
1E65 MVI A,C3
1E67 STA 0000
1E6A LXI H,0203
1E6D SHLD 0001
1E70 STA 0005
1E73 LXI H,0200
1E76 SHLD 0006
```

Заменим его на следующий код:

```
1E65 MVI A,C3
1E67 STA 0005
1E6A LXI H,0203
1E6D SHLD 0001
1E70 LXI H,0200
1E73 SHLD 0006
1E76 MOV H,L    ;L=0  из предыстории
1E77 MVI M,F7   ;HL=0
```

Т. к. перестраиваемый байт с адреса 1E75H передвинулся на адрес 1E72H, проводим соответствующие изменения в массиве настройки, а точнее меняем 04 по адресу 266EH на 20H.

Кому мое обьяснение покажется неясным может просто сменить значения ячеек памяти через директиву S SIDa в соответствии с таблицей:

Адрес ячейки	Старое значение	Новое значение
1E68	00	05
1E70	32	21
1E71	05	00
1E72	00	02
1E73	21	22
1E74	00	06
1E75	02	00
1E76	22	65
1E77	06	36
1E78	00	F7
266E	04	20

Виктор Саттаров, Киров-1992

Внимание ! Новинка !

В фирме "Байт" разработан новый загрузчик - "Load Byte". Загрузчик позволяет осуществлять загрузку программ с магнитофона и дисковода. Для пользователей магнитофонной конфигурации "Вектора-06Ц" введен ряд сервисных удобств:

- вывод имени загружаемого файла на экран;
- пометка начального и конечного блоков на экране;
- возможность загрузки сбойных программ как в Copy-N4 (при сбое программа не сбрасывается, а продолжает грузиться. Ленту можно проматывать вперед-назад и повторно грузить сбойные участки). Причем загрузчик сам определит, когда программа загрузится без ошибок ;
- специальный маркер указывает на экране на текущий загружаемый блок;
- более надежное схватывание начала программы по сравнению с предшествующими версиями;
- возможность загрузки программы прямо с середины с выводом имени загружаемой программы на экран. Это значительно облегчит поиск нужной программы и ее ввод тем, кто пользуется магнитофонами без счетчиков;
- автозапуск загружаемой программы;
- действие клавиши УС сохраняется.

Загрузчик зашит в м/с ПЗУ РФ5 объемом 2 кбт. Прилагается инструкция по замене м/с ПЗУ. Стоимость загрузчика _____ руб. по предоплате телеграфным (почтовым) переводом. При оплате наложенным платежем стоимость возрастает на 35% .

В следующем номере "Байта" - описание м/с таймера, с помощью которого на "Векторе" реализуются музыка и звуковые эффекты.

" Б А Й Т - 2 1, 2 2 "

Киров, 1994
информационно-рекламный выпуск
центра "БАЙТ"
для пользователей ПК "Вектор-06Ц"

++
+ Наш адрес: г. Киров, Октябрьский пр-т, 118, м-н "Дом Радио" +
+ Адрес для писем" 610006, г.Киров, а/я 1248, Зубкову А.Н. +
++

НЕСКОЛЬКО СЛОВ О БЕССМЕРТИИ.

Эта статья рассказывает о способах установления бессмертия в играх. Следует отметить, что описанные здесь способы наиболее часто встречаются в играх до 91 г.,а за 92-93 - достаточно редко, т.к. программисты стараются как можно лучше защитить свои программы от несанкционированного доступа. Для успешной работы вам следует иметь как минимум один вид Монитора, который загружается сразу по своим рабочим адресам. Нужно загрузить обычный Монитор-отл., после нажатия на клавиши БЛК+СБР, выбрать область расположения "1", а затем записать на пленку три составные части Монитора : O>0,1,0 ; O>94,0C,0 ; O>E0,20,0. Загружать такой Монитор нужно через клавиши УС+ВВОД+БЛК, а после загрузки трех частей нажать на клавиши БЛК+СБР. Следует так же отметить, что Монитор лучше подзагружать в игры, что позволит вам раскодировать закодированную игру или же распаковать ее.Описанный выше тип отладчика хорош тем, что загружаясь сразу по своим рабочим адресам, он не затирает собой программу, которая расположена с адреса 100h (1 блок) и выше. Но будьте осторожны: этот способ не подействует, если в игре нельзя нажимать на клавиши БЛК+СБР или если игра загружается с нулевого блока. После загрузки игры вы, убедившись , что она перезапускается при нажатии на клавиши БЛК+СБР,можете смело подзагружать в нее Монитор. Сейчас я в общих чертах расскажу, какими способами в играх можно делать бессмертие. Например, известно, что жизней в игре пять. Теперь следует поискать ячейку с таким содержимым : MVI A,05 с помощью дерективы Q: Q<нач.адр.>,<кон.адр.> <ВК> :'MVI A,5 <ВК> Компьютер может выдать как 1-2 адреса, так и 10-15. Для удобства, если выданных адресов больше 5,следует записать адреса на листок бумаги, чтобы в дальнейшем работать с ними. Если же ни одного адреса не выведется на экран, то следует поискать другое значение, например, MVI M,05, а затем работать уже с этими значениями. Затем следует с помощью директивы L<адрес> просмотреть те адреса, которые вывелись нам при поиске описаной выше ячейки. Обращать внимание следует только на те адреса, где после MVI A,05 стоит команда STA<адрес>. То, что стоит после STA следует выписать на бумагу. После этого опять же с помощью директивы Q следует поискать следущую последовательность : DCR A и STA <адрес>, разделяя эти команды знаком "!" (более подробную информацию смотрите в описании Монитора-отладчика). Например, DCR A!STA 1234. Так вы поочередно перебираете все адреса, которые находились после команды STA. Если после поиска такой последовательности вам выдался какой нибудь адрес, то скорее именно он и есть то, что вы искали. Тогда по этому адресу вместо команды DCR A следует поставить NOP. Но следует так же обратить ваше внимание на то, что после такой последовательности должен стоять переход JNZ<адрес> или JZ<адрес>.

Если стоит команда JZ<адрес>, то посмотрев с помощью директивы D этот самый адрес, можно увидеть сообщение "GAME OVER".Если стоит команда JNZ<адрес>,то посмотрев опять директивой D адрес,который идет непосредственно после команды перехода,можно так же увидеть то же сообщение. Если же после такого поиска вам не выдалось ни одного адреса, то нужно поочередно просматривать все команды "DCR A" в программе, опять же с помощью команды Q. Но DCR-ов в программе может быть очень много. В этом случае нужно воспользоваться директивой M. С помощью этой команды можно просмотреть графику в программе.Формат команды следующий : M <нач.адр.>,<кон.адр.>,A000.Но не следует вводить слишком большие адреса - это может привести к уничтожению Монитора и уничтожению программы. Разность между начальным и конечным адресами не должна привышать 2000h (1 строчка). Например, "M100,2000,A000"; M4000,6000,A000 и т.д. При выполнении этой команды экран заполняется графическими символами.Начиная слево направо идет графический эквивалент тех столбиков или адресов,которые вы ввели.При этом самый левый граф.столбик соответствует адресам 0000h-00FFh, следущий - адресам 100h-200h и т.д. Следует обратить внимание на те столбцы, где находится графическая информация. Если в этом же столбце находится и один из адресов,которые выдались вам при поиске команды DCR A, то этот адрес можно сразу отбросить от рассмотрения. Таким образом нужно просмотреть все адреса с этой командой.Нужно замечать те DCR-ы, после которых стоят команды JNZ или JZ.Если и это не поможет, то есть еще один способ. В Ассемблере имеется команда CPI<число>.Она сравнивает два числа (которые находятся в аккумуляторе и непосредственно после команды). Некоторые авторы программ делают вот как : сравнивают количество жизней в данный момент с их конечным количеством. Если в результате сравнения получился 0,то следует конец игры.С помощью команды Q следует поискать следущее: CPI 05. После этой команды должен стоять адрес перехода:JNZ<адрес> или JZ<адрес>.Эта команда используется в программах в тех случаях,когда жизни в игре не вычитаются,а прибавляются (например, как в игре EAGLE SNEST прибавляются ранения). Ниже приводятся игровые программы, в которых бессмертие было установлено с помощью выше описанных способов. Во все эти игры был подзагружен Монитор, описанный в начале статьи.

EXOLON.

Известно, что жизней в игре девять. Поищем ячейку с таким содержимым: #Q100,8000 <BK> :'MVI A,9 <BK> *2021 *6F38 #_ После просмотра этих адресов дерективой L, мы видим, что после команды MVI A,09 идет STA 2029 (в первом случае) и STA 3EF1 (во втором).Сейчас следует опять с помощью дерективы Q поискать следущее: DCR A!STA 2029 и DCR A!STA 3EF1.В первом случае вам не выдадется ни одного адреса, а во втором - один. Этот адрес - 41D6. Именно в это место взамен команды DCR A нужно поставить NOP.

EAGLE SNEST.

В этой игре ведется счет патронам и ранениям.Вам нужно нейтрализовать эти две вещи (сделать патроны и ранения бесконечными). Подзагружаем в игру Монитор и ищем адрес, в котором число патронов устанавливается в восемь: Q100,6000 <BK> :'MVI A,8 <BK> Вам выдадуться следущ. адреса: 3E5,413,41B,449,509,D4C,E3A,18F3, 1911,1932,1953. После адресов D4C,18F3,1911,1932,1953 стоит команда STA 0D9F, а после адреса E3A - STA 0DE1. Будем искать следующие последовательности: DCR A!STA D9F и DCR A!STA DE1. #Q100,6000 <BK>

:'DCR A!STA D9F <BK> *13B1 #Q100,6000 <BK> :'DCR A!STA DE1 <BK> #_ По этому адресу нужно поставить вместо DCR A - NOP. Теперь ранения: их нужно всего девять - и конец игры. Они прибавляются, потому ищем следущее: #Q100,6000 <BK> :'CPI 09 <BK> *108A *10CE *4676 #_ После команды по адресу 108A стоит переход - JZ<адрес>. Директивой D смотрим этот самый адрес. Читая правый столбик, вы видите сообщение "Игра окончена".Значит это и есть та самая ячейка, которая нам нужна. Поэтому вместо команды JZ нужно поставить NOP.

Навалон А.В., Шилин Д.В., Киров.

SHOP TOUR.

В этой игре вам нужно заработать как можно больше денег, разъезжая по биржам СНГ . В начале вы имеете 100 долларов и машину на ней вам предстоит зарабатывать себе на жизнь и обгонять конкурентов, которые используют любые способы, чтобы навредить вам. Начинаете вы свое путешествие из столицы нашей родины - Москвы. Когда на экране вашему вниманию представится карта, вы с помощью указателя можете посмотреть цены на биржах в любом городе страны наведя указатель на любой нужный вам городок. Ваша задача,покупая подешевле товары в одном городе продавать их подороже в другом. Когда вы накопите достаточно много денег, то на биржи вас будут сопровождать конкуренты . Ни в коем случае не давайте им шанс добраться до биржи первее вас,т.к. в случае если они каким-то чудом сумеют это сделать , то могут упасть цены как раз на тот товар, который вы везете на продажу,что может привести к убыткам. Если же это случилось,то внизу экрана на факсе вам выдадется соответствующее сообщение о том, что упали цены на такой-то товар. В любом случае число конкурентов уменьшится,если ваш кошелек потерпит довольно ощутимые убытки. Машина - довольно хорошая вещь, но лишь при том условии, если ей есть на чем ехать, т.е. бензин. В любом городе вы можете купить последний за 2 доллара.Если же он кончится прямо в дороге,то ничего не поделаешь - придется покупать его у очень предприимчевых бизнесменов - за 20 долларов (если у вас имеются деньги). Если денег нет, то нужно играть сначала. Покупка товаров производится при нажатии на клавиши УС+"влево". Продажа - УС+"вправо" (вместо клавиши УС можно нажимать на клавиши "CC" или РУС/LAT).Вот вы накупили товаров и едете на другую биржу, чтобы продать их там за более высокую цену.Когда вы едете по дороге, то если вы находитесь у города, то справа от проезжей части появится табличка с названием данного городка. Когда эта табличка дойдет до края экрана, нужно нажать клавишу "пробел". Тогда вы окажитесь на бирже в этом городе,где и можете произвести все нужные вам сделки. Несколько полезных советов: 1) В начале игры лучше вести торговлю только между двумя городами.Когда вы накопите достаточно много денег, на них можно купить бензин, чтобы затем не беспокоиться о нем, а только затем расширять свою деятельность. 2) Если вы повернули не на ту дорогу или случайно проехали город то используйте клавиши СТР или "влево-вверх" - ваша машина будет развернута в противоположную сторону. 3) Если конкурентов довольно много , то их довольно сложно обогнать, т.к. они движутся с большой скоростью и их не заносит на дорогах. В этом случае можно нажать на клавишу "СТР" и уменьшить свою скорость, подождать, пока они вас обгонят, развернуться еще раз и продолжить путь в нужном направлении.

LAND.

Графика и звук в этой игре отнюдь не делают ее привлекательней и интересней.Перед началом игры вы можете выбрать скорость, на которой собираетесь играть.Управление главным героем осуществляется клавишами курсора, "СТР" и "влево-вверх" - пробивание стенки. Так же в процессе игры можно выбрать комнату, нажав на клавиши 1,2,...,9,0. В этой игре сразу установлена бесконечная жизнь,так что вы можете играть сколко вам захочется.Но скорее вы сбросите эту игру из своего окна вместе с компьютером.

DIAMOND-1, -2.

В игре имеется возможность смены уровня и увеличения количества жизней. Уровень,с которого вы будете играть, можно выбрать перед началом игры, нажав на клавиши "X"+"Y". Количество жизней можно увеличить в самой игре, нажав на клавиши "РУС/LAT"+"Z". Первая часть игры имеет 10 уровней, вторая - 15. Следует так же сказать,что жизней в игре не может быть больше восьмидесяти.Если их станет больше, то происходит переигровка.

ROCKFORD.

Наиболее удачная версия игры из серии "BOULDER DUSH". Она адаптирована с компьютера IBM . Игрушка отличается от других аналогичных ей тем,что она имеет превосходную графику,быстроту движений, отличный звук, что делает ее интересной и не надоедливой. В игре вы управляете маленьким поваренком,который должен собрать определенное количество яблок на каждом уровне игры. В начале вы имеете пятнадцать жизней , за каждый пройденный уровень вам прибавляется еще одна.Дойти же до последнего уровня довольно сложно. Но в игре имеется возможность смены уровня. Для этого нужно нажать клавиши "УС+"СС"+"РУС/LAT". При этом вам прибавится очередная жизнь и вы окажитесь на следущем уровне. Но будье внимательны - уровней всего десять. При нажатии на клавишу ВК игра переходит в режим ожидания и ждет пока не будут нажаты клавишы: F1 -"продолжить"/F2 -"переиграть".

FLIGHT.

Тип этой игры - "стрелялка".Ваша задача - убить как можно больше иноземных сушеств. Вы можете пополнить запас горючего,взяв бочонок с ним. Так же в процессе игры по полю движется камень, который, задев вас, отнимает одну жизнь.За попадание в иноземное существо у вас увеличиваются очки.Но за один выстрел вычитается 2 очка. За очередные 200 очков прибавляется жизнь. Но основной задачей является набор 1000 очков, раньше, чем вы успеете доехать до горы. Иначе игра закончится. Для этого (на первом уровне) стреляйте все подряд, даже топливо. Но открывать огонь следует очень экономно.На более старших уровнях топливо, наоборот, следует каждый раз брать. Так же не стреляйте в чудовищ, когда они будут совсем близко от вас.

КЛАД-1.

В этой игре вам нужно собрать как можно больше кладов, при этом не попадаясь злобным черепкам. В кладе вы можете найти звездочку оружие протов черепков, а так же ключ- он дает вамм право перейти на следущий уровень игры. Жизнь же дается только за 20000 очков, а набрать их практически невозможно. В начале вы имеете три жизни. Это довольно мало, т.к. уровней в игре двадцать, и следущий гораз-

до сложнее предыдущего. Чтобы увеличить число попыток в игре , нужно в нее подзагрузить Монитор.Затем с помощью деректива A<адрес> нужно изменить содержимое ячейки 3A1 : #A3A1 <ВК> 3A1 MVI A,F0 <ВК> 3A3 <F4> Затем можно выгрузить на пленку 3-й столбик, чтобы затем по мере надобности загрузить его в игру через клавиши УС+ВВОД+БЛК : O>3,1,0 <ВК> Такой способ можно использовать в любой игре, которая не залетает при нажатии на клавиши УС+ВВОД+БЛК, затем БЛК+СБР.

Навалон А.В., Киров.

****
* E X O L O N *
****

В ходе исследования соседней галактики была обнаружена пригодная для жизни планета EXOLON .Вскоре она была заселена, и мир долгое время царил там. Но вот однажды связь с планетой была потеряна. На помощь была выслана экспедиция из трех патрульных кораблей. Приближаясь к планете, корабли неожиданно были атакованны мощным огмем НЛО. Не успев опомниться, вы с ужасом замечаете уходящие в глубины космоса обломки двух кораблей экспедиции. Еще находясь в полушоке,вы даете бортовому компьютеру команду экстренной посадки. Приземлившись и выйдя из корабля, вы видите, что он изрядно разбит вражеским огнем.Видя, что ремонт - пустая трата времени, вы,захватив немного еды,автомат с запасом в 99 патронов и 10 ракет, отправляетесь навстречу судьбе... Целью этой игры является уничтожение всего,что встретится на вашем пути.Уничтожать необходимо движущихся врагов и встречающиеся на вашем пути препятствия.Вы же можете стрелять в снаряды недвижущихся транспортеров, стоящих на дороге. Необходимо быть очень осторожным, в радио управляемую ракету не стреляйте, а уничтожте ракетой ее пульт управления. На вашем пути будут попадаться мины и подъемники,одно прикосновение к которым приводит вас к гибели. Обязательно собирайте контейнеры с боеприпасами.Во время игры вы увидите телепортационные камеры, зайдя в одну из которых и нажав "вверх",вы появляетесь в другой.В одной из комнат уровня вы увидите кабину со сквозным проходом,зайдя в которую и нажав "вверх" вы получите новое оружие и скафандр - в нем вам будут не страшны мины и подъемники.Всего в игре 5 уровней, каждый уровень состоит из 25 зон. Пройдя один уровень вам дается возможность заработать очки, после появления таблицы чисел, напротив них начнет бегать стрелка,нажав РУС/LAT она остановится напротив какого-либо числа

Советы начинающим:
---

1) Прыжки совершайте как можно точнее. 2) Вертикальный силовой барьер будет разрушен после продолжитель ной стрельбы. 3) Старайтесь идти всегда по верху. 4) Не задерживайтесь долго в одной комнате - это приведет вас к гибели. 5) Не старайтесь уничтожить все, так как для вас это может плохо кончиться.

SABOTAGE.

Вы-диверсант,находящийся в тылу врага.Вам было дано задание уничтожить вражескую подземную базу, заложив бомбу цепного действия Успешно выполнив данное вам поручение,вы дожны найти выход с базы. Во время игры вам будут встречаться охранники,с которыми вам придется драться, а так же агрессивно настроенные против вас собаки.

Столкновений с ними лучше избегать.Для этих и других целей можно пользоваться лифтами.Он управляется с помощью клавиш вверх или вниз. Для более быстрого прохождения игры старайтесь сначала выйти на самый верх, затем двигаться до конца вправо, а после этого идти все время вправо и вниз. В игре можно восстановить энергию до изначального уровня.В режиме "пауза" (нажать "РУС/LAT"), при нажатии на клавиши ПС+ВК+ЗБ, ваша энергия восстановиться до изначального уровня.

Шилин Д.В., Киров.

И Н Ф О Р М А Ц И Я

Таймер КР580ВИ53

Микросхема программируемого таймера КР580ВИ53 предназначена для задания временных интервалов в МПС и может использоваться в следующих стандартных режимах работы: счетчик внешних событий, программируемый ждущий мультивибратор, делитель частоты, генератор меандра, программно и аппаратно запускаемый строб.

БИС КР580ВИ53 содержит три независимых канала. В каждом канале есть регистр управляющего слова, шестнадцатиразрядный программируемый счетчик, работающий на вычитание в двоичном или двоично-десятичном коде, двухбайтный буферный регистр, в который по специальной команде переписывается текущий код счетчика. Программирование каждого канала таймера сводится к следующим операциям:

запись в регистр управляющего слова индивидуального слова управления канала;

запись в шестнадцатиразрядный программируемый счетчик необходимого кода пересчета.

Обратим внимание читателя на основные особенности программирования таймера.

1. Запись индивидуального слова управлния канала происходит по единому для всех каналов адресу (A0=1, A1=1). Указание конкретного канала, к которому относится упавляющее слово, содержится в самом управляющем слове.

2. Режим обращения к шестнадцатиразрядному программируемому счетчику определяется разрядами D4, D5 упавляющего слова. Однако необходимо обязательно завершить цикл обращения к счетчику полностью, т. е. если запрограммировано два обращения, то однократное обращение вызовет неправильную работу канала. Если разряды управляющего слова D4, D5 - 0, то при записи такого управляющего слова содержимое шеснадцатиразрядного счетчика "защелкивается" в буферном регистре. Таким образом, можно получать мгновенное (в момент записи) значение содержимого счетчика.

3. Обращение к адресам начальных счетчиков таймера как при начальной загрузке, так и в процессе работы канала, может происходить в любой последовательности.

4. Считывание регистра управляющего слова запрещено.

Адресация регистров таймера приведена в табл. 1.

Адресация регистров таймера — Таблица 1

A1	A0	Название регистра	A1	A0	Название регистра
0	0	Счетчик канала 0	1	0	Счетчик канала 2
0	1	Счетчик канала 1	1	1	Регистр управляющего слова

Опишем назначение разрядов управляющего слова канала. Разряд D0 в нулевом состоянии устаналивает режим двоичного счета, а в единичном состоянии - режим двично-десятичного счета.

Разряды D3, D2, D0 определяют режим работы канала. Каждый канал может работать в одном из шести режимов (табл.2).

Кодирование режимов — Таблица 2

D3	D2	D1	Ном. реж.	Название режима	D3	D2	D1	Ном. реж.	Название режима
0	0	0	0	Генерация программир. задержанного перепада 0 -> 1	X	1	1	3	Программируемый делитель частоты выходным сигналом типа меандр
0	0	1	1	Программируемый одновибратор	1	0	0	4	Программно-запускаемый задержан. строб
X	1	0	2	Программируемый делитель частоты	1	0	1	5	Аппаратно-запускаемый задержан. строб

Разряды D5, D4 устанавливают режимы обращения к старшему и младшему байтам счетчиков канала (табл.3).

Кодирование разрядов D5, D4 — Таблица 3

D5	D4	Функция
0	0	Перепись текущего кода счетчика канала в буферный регистр канала
1	0	Обращение к старшему байту счетчика
0	1	Обращение к младшему байту счетчика
1	1	Последовательное обращение сначала к младшему, затем к старшему байтам счетчика канала

Разряды D7, D6 - разряды косвенной адресации к трем регистрам управляющего слова, т. е. они определяют, в какой из регистров управляющего слова, какого канала будут записаны остальные управляющие разряды D5-D0 (табл.4).

Таблица 4

Кодирование разрядов D7, D6

D7	D6	Номер канала
0	0	0
0	1	1
1	0	2
1	1	-

Практическую релизацию операций инициализации или настройки таймера рассмотрим на примерах. Во всех примерах примем, что адрес счетчика канала 0 равен PORT53, все остальные адреса будем записывать в виде PORT53+i, где i - номер канала.

Пример 1. Подпрограмма начальной установки канала 0 (режим 5) имеет вид:

```
TIME:   MVI     A,1AH   ;Канал 0, режим 5
        OUT     PORT53+3
```

```
        MVI     A,13H
        OUT     PORT53+3
        RET
```

1AH=00011010B - режим двоичного счета, режим 5, канал 0, обращение к младшему байту.

13H - число, записываемое в младший байт счетчика.

Если в счетчик необходимо загрузить двухбайтное число 1513H, то подпрограмма будет выглядеть следующим образом:

```
TIME:   MVI     A,3AH
        OUT     PORT53+3
        MVI     A,13H
        OUT     PORT53+3
        MVI     A,15H
        OUT     PORT53
        RET
```

Пример 2. Подпрограмма начальной установки таймера:

канал 0, режим 5, в счетчик загрузить 1513H;
канал 1, режим 1, в счетчик загрузить 06H;
канал 2, режим 5, в счетчик загрузить 0BH имеет вид:

```
TIME3:  MVI     A,3AH
        OUT     PORT53+3
        MVI     A,52H
        OUT     PORT53+3
        MVI     A,9AH
        OUT     PORT53+3
        MVI     A,13H
        OUT     PORT53
        MVI     A,15H
        OUT     PORT53
        MVI     A,06H
        OUT     PORT53+1
        MVI     A,0BH
        OUT     PORT53+2
        RET
```

52H=01010010B - режим двоичного счета, режим 1, канал 1, обращение к младшему байту;

9AH=10011010B - режим двоичного счета, режим 5, канал 2, обращение к младшему байту.

Иногда по текущему значению числа в счетчике канала необходимо принимать какое-то решение в программе. Существует два способа считывания микропроцессором числа, содержащегося в счетчике. При первом способе на время выполнения обычной операции "Чтение счетчика" внешней логикой должна быть запрещена подача синхронизирующих импульсов на данный канал таймера.

В этом случае считывание осуществляется одной или двумя командами в зависимости от разрядов D4, D5 ранее записанного управляющего слова. Причем при первом считывнии получаем младший байт счетчика (D4=D5=1), при втором - старший байт счетчика.

Пример 3. Подпрограмма чтения содержимого счетчика канала 0 (D4=D5=1):

```
TIME4:  MVI     A,01H   ;Программное отключение
        OUT     OTKL    ;синхроимпульсов таймера
        IN      PORT53  ;Получение младшего байта
        MOV     L,A     ;счетчика
        IN      PORT53  ;Получение старшего байта
        MOV     H,A     ;счетчика
        RET
```

Второй способ считывания содержимого счетчика канала не нарушает выполнение счета в канале.

Пример 4. Подпрограмма чтения содержиого счетчика канала 0 без нарушения счета в канале:

```
TIME5:  MVI     A,XX00XXXXB
        OUT     PORT53+3
        IN      PORT53
        MOV     L,A
        IN      PORT53
        MOV     H,A
        RET
```

XX00XXXXB - управляющее слово, записываемое в регистр управляющего слова, причем D4=D5=0, а вместо X указываются те значения битов, которые были записаны при инициализации канала 0.

(взято из книги В.Г.Майоров, А.И.Гаврилов "ПРАКТИЧЕСКИЙ КУРС ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ")

Генерация случайных чисел с помощью микросхемы таймера

Поскольку регистр счетчика канала таймера перезагружается снова и снова данным числом (а в промежутках идет счет вниз до 0), выберите в качестве загружаемого в счетчик значисло, равное требуемому диапазону случайных чисел.

Лучше всего использовать режим 3 канала микросхемы таймера. Сначала установите для счетчика желаемый диапазон случайных чисел. Затем, чтобы получить из канала случайное число, надо подать команду командному регистру микросхемы таймера перенести текущее значение счетчика в регистр "задвижки", для чего надо сбросить биты D4 и D5. Этот перенос в регистр задвижки не мешает продолжающемуся счету. Затем установите биты D4 и D5 командного регистра, чтобы процессор мог читать из регистра задвижки. После этого две инструкции IN дадут сначала младший, а затем старший байт. Наконец, восстановите первоначальное значение регистра задвижки, чтобы счет продолжался в пределах указанного диапазона времени.

Создание звуковых эффектов

Звуковые эффекты обычно достигаются непрерывным изменением частоты тона. Все должно быть сконстуировано из чистых музыкальных тонов, а это значит, что эффект дисторции звука должен достигаться за счет такого быстрого изменения тона, что ухо не успевает разделить тона. Например, душераздирающее "чирикание" может быть получено при быстром переключении между одним и тем же тоном, отстоящим на несколько октав.

При изменении частоты всего на несколько герц получаем вибрацию.

Другой метод заключается во вложении плавно меняющихся тонов внутрь последовательности, которая сама "гуляет" по частотам вверх и вниз. Этот метод применяется во многих играх с лабиринтами.

```
100 FOR I=1 TO 10                число повторений
110 FOR J=1 TO 6                 число разных октав
120 PLAY "mbl64t255O=j;ba#ag#gf#feb#de#cc#dd#eff#gg#aa#b"
130 NEXT J
140 NEXT I
```

Ассемблер кроме всего прочего позволяет генерировать нечистые тона, когда интервал, в течение которого динамик включен, не равен интервалу, в течение которого он выключен. Такое нарушение симметрии может приводить к жужжащим и брякающим звукам. Когда отношение этих интервалов составляет, скажем, 50 к 1, получаем жужжание. Если увеличить отношение еще в 10 - 20 раз, то жужжание переходит в отдельные брякающие звуки. Этот эффект достигается в ВЕКТОРЕ при помощи нулевого бита порта 01H, отвечающего за вывод на МГ.

Итак, звук можно генерировать двумя способами: с помощью микросхемы таймера и с помощью порта 01H. Ассемблер позволяет соединить два способа генерации звука, что создает имитацию одновременной генерации двух разных звуков. Интерференция этих двух сигналов приводит к сложной форме звуковой волны. Каждый из двух звуков имеет меньшую громкость, поэтому в результате получается скорее жужжание, чем два разных голоса. Этот прием реально полезен только для создания звуковых эффектов.

Цифровой синтез музыкальной шкалы

Не вдаваясь в подпробности строгого математического анализа предложенного метода, покажем на конкретном примере его практическую направленность. Установим коэффициенты деления делителей частоты каждой ступени шкалы частот самой верхней октавы в соответствии с выражением

$$Kj = (10\square\square(2^{(1/12)})^{-j}, \qquad (1)$$

где j - порядковый номер ступени шкалы частот;
n - целое число, определяющее требуемую точность формирования.

Таблица 5

j	(2^(1/12))^-j	Kj	fj	Значения по музыкальной шкале, Гц	Наименов. тона
1	2	3	4	5	6
0	1,0	10000	440	440	ЛЯ [1]
1	0,9438743	9438	466,2		ЛЯ [1]
2	0,8908987	8908	493,93	494	СИ [1]
3	0,8408904	8408	523,31	523	ДО [2]
4	0,7937011	7937	554,36		ДО [2]
5	0,7491540	7491	587,97	588	РЕ [2]
6	0,7071070	7071	622,25		РЕ [2]
7	0,6674204	6674	659,27	660	МИ [2]

```
 8 |  0,6299609  |   6299 | 698,52 |    698    | ФА   [2]
 9 |  0,5946038  |   5946 | 739,99 |           | ФА   [2]
10 |  0,5612314  |   5612 | 784,03 |    784    | СОЛЬ [2]
11 |  0,5297317  |   5297 | 830,65 |           | СОЛЬ [2]
12 |  0,500000   |   5000 | 880    |    880    | ЛЯ   [2]
---'-------------'--------'--------'-----------'-----------
------------------------------------------------------------
```

Частота следования импульсов на выходе j-го делителя будет равна:

$$fj = \frac{fr}{(10\square\square 2^{(1/12)})^{-j}} \qquad (2)$$

где fr - частота на выходе высокостабильного генератора.

В качестве опорной частоты, соответствующей j=0, примем частоту тока Ля первой октавы, имеющую точное целочисленное значение fo=440 Гц.

Ограничивая требуемую точность формирования четырьмя значащими цифрами после запятой (m=4), получим (в МГц):

fr=fo□10^4=4,4.

Значения частот ступеней второй октавы в порядке возростания, вычисленные в соответствии с выражением (2), приведены в табл. 1.

В графе 5 даны округленные значения частот музыкальных тонов.

Частоты на октаву выше можно получить, удваивая эти значения, на две октавы выше - еще раз удваивая частоты. И наоборот, частоты на октаву ниже равны приблизительно половине этих значений (хорошо настроенное пианино точно не следует арифметическим интервалам).

Микропроцессор КР580ВМ80А

Система команд микропроцессора КР580ВМ80А — Таблица 6

команда	код операции	число байтов	число циклов	число тактов	признак рез-та S	Z	AC	P	CY
ACI DATA	11001110	2	2	7	+!	+!	+!	+!	+
ADC R/M	10001R/M	1	1/2	4/7	+!	+!	+!	+!	+
ADD R/M	10000R/M	1	1/2	4/7	+!	+!	+!	+!	+
ADI DATA	11000110	2	2	7	+!	+!	+!	+!	+
ANA R/M	10100R/M	1	1/2	4/7	+!	+!	U!	+!	0
ANI DATA	11100110	2	2	7	+!	+!	U!	+!	0
CALL ADDR	11001101	3	5	17	-!	-!	-!	-!	-
Ccnd ADDR	11CND100	3	3/5	11/17	-!	-!	-!	-!	-
CMA	00101111	1	1	4	-!	-!	-!	-!	-
CMC	00111111	1	1	4	-!	-!	-!	-!	+
CMP R/M	10111R/M	1	1/2	4/7	+!	+!	+!	+!	+
CPI DATA	11111110	2	2	7	+!	+!	+!	+!	+
DAA	00100111	1	1	4	+!	+!	+!	+!	+
DAD RS	00RS1001	1	3	10	-!	-!	-!	-!	+
DCR R/M	00R/M101	1	1/3	5/10	+!	+!	+!	+!	-
DCX RS	00RS1011	1	1	5	-!	-!	-!	-!	-
POP RP	11RP0001	1	3	10	-!	-!	-!	-!	-
POP PSW	11110001	1	3	10	+!	+!	+!	+!	+
PUSH RP	11RP0101	1	3	11	-!	-!	-!	-!	-

RAL	00010111	1	1	4	-	-	-	-	+
RAR	00011111	1	1	4	-	-	-	-	+
RET	11001001	1	3	10	-	-	-	-	-
Rcnd	11CND000	1	1/3	5/11	-	-	-	-	-
RLC	00000111	1	1	4	-	-	-	-	+
RRC	00001111	1	1	4	-	-	-	-	+
RST NUM	11NUM111	1	3	11	-	-	-	-	-
SBB R/M	10011R/M	1	1/2	4/7	+	+	+	+	+
SBI DATA	11011110	2	2	7	+	+	+	+	+
SHLD ADDR	00100010	3	5	16	-	-	-	-	-
SPHL	11111001	1	1	5	-	-	-	-	-
STA ADDR	00110010	3	4	13	-	-	-	-	-
STAX R	000R0010	1	2	7	-	-	-	-	-
STC	00110111	1	1	4	-	-	-	-	1
SUB R/M	10010R/M	1	1/2	4/7	+	+	+	+	+
SUI DATA	11010110	2	2	7	+	+	+	+	+
XCHG	11101011	1	1	4	-	-	-	-	-
XRA R/M	10101R/M	1	1/2	4/7	+	+	0	+	0
XRI DATA	11101110	2	2	7	+	+	0	+	0
XTHL	11100011	1	5	18	-	-	-	-	-
DI	11110011	1	1	4	-	-	-	-	-
EI	11111011	1	1	4	-	-	-	-	-
HLT	01110110	1	1	7	-	-	-	-	-
IN PORT	11011011	2	3	10	-	-	-	-	-
INR R/M	00R/M100	1	1/3	5 10	+	+	+	+	-
INX RS	00RS0011	1	1	5	-	-	-	-	-
JMP ADDR	11000011	3	3	10	-	-	-	-	-
Jcnd ADDR	11CND010	3	3	10	-	-	-	-	-
LDA ADDR	00111010	3	4	13	-	-	-	-	-
LDAX R	000R1010	1	2	7	-	-	-	-	-
LHLD ADDR	00101010	3	5	16	-	-	-	-	-
LXI RS,DATA	00RS0001	3	3	10	-	-	-	-	-
MOV R/M,R/M	01R/MR/M	1	1/2	5/7	-	-	-	-	-
MVI R/M,DAT	00R/M110	2	2/3	7/10	-	-	-	-	-
NOP	00000000	1	1	4	-	-	-	-	-
ORA R/M	10110R/M	1	1/2	4/7	+	+	0	+	0
ORI DATA	11110110	2	2	7	+	+	0	+	0
OUT PORT	11010011	2	3	10	-	-	-	-	-
PCHL	11101001	1	1	5	-	-	-	-	-

Примечания:

Ccnd обозначает группу команд CNZ,CZ,CNC,CC,CPO,CPE,CP,CM
Rcnd обозначает группу команд RNZ,RZ,RNC,RC,RPO,RPE,RP,RM
Jcnd обозначает группу команд JNZ,JZ,JNC,JC,JPO,JPE,JP,JM

Состояние признака результата указывается следующим образом: "+" - признак устанавливается в 1 или 0 в зависимости от результата выполнения команды; "-" - признак не изменяется; U признак не определен.

В команде MOV R/M,R/M первый и второй операнды не должны определять имя ячейки памяти одновременно.

```
        +-----+ +-----+ +  -  + +--+--+     +-----+ +-----+
        +       +     + +    -+    +              +       +
        +-----+ +-----+ +  -  +    +    --- +-----+    ---+
        +     + +     + +-    +    +        +             +
        +-----+ +     + +     +    +        +------ +-----+
           информационно-рекламный выпуск центра "БАЙТ"
                 для пользователей ПК "Вектор-06Ц"
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +   Адрес для писем" 610006, г.Киров, а/я 1248, Зубкову А.Н.   +
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

НЕБОЛЬШОЕ ВСТУПЛЕНИЕ!

Разработчики аппатного обеспечения для "Вектора" не стоят на месте. Уже разработаны квазидиск, контроллер дисковода, джойстики. А в этом номере мы расскажем о новых разработках: 2-версиях муз.процессора и новом ПЗУ для компьютера.

Бессменный редактор выпуска "БАЙТ" - Луппов Г.Б.

1. Музыкальный сопроцессор

Музыкальный сопроцессор представляет собой плату с разъемами для подключения к компьютеру и для подключения магнитофона или усилителя. Разработан он был по аналогии с музпроцессором от "Пентагона-128" и предоставляет программисту широкие возможности по созданию музыкального сопровождения на высоком качественном уровне. Кроме того возможна адаптация уже существующей музыки с "Пентагона" на "Вектор-06Ц". Музыкальный процессор значительно расширит и обогатит возможности Вашего компьютера. Пока имеется небольшой минус - небольшое число программ, поддерживающих музпроцессор, но это дело времени. Кто решится приобрести музпроцессор сейчас, тот вложит деньги в будущее своего компьютера! Итак слово авторам.

1.1. Музыкальный процессор "Sound Tracker".
Разработчик Саттаров Виктор

1.1.1. Технические характеристики

3 независимых музыкальных канала (стерео музыка) + генератор шума;
2 (или 1) параллельных порта ввода/вывода; 16 градаций амплитуды;
16 форм волного пакета; Основная частота - 1.7734 Мгц;
Диапазон воспроизводимых частот: от 27 Гц до 11 Кгц
Частоты волнового пакета: от 0.1 Гц до 6 Кгц
Частота генератора шума: от 3 Кгц до 110 Кгц

1.1.2. Программирование ПГЗ АУ-3-8910

ПГЗ является регистро-ориентированным генератором звуков. Его функции выполняются посредством 16 внутренних регистров. Номер регистра задается 4 младшими разрядами при подаче команды "фиксация адреса" и остается действительным до получения команды о смене этого адреса.

В таблице приведены функции регистров и допустимые значения для этих регистров.

N регистра	назначение или содержимое	значение
0, 2, 4	нижние 8 бит частоты голосов A, B, C	0 - 255
1, 3, 5	верхние 4 бита частоты голосов A, B, C	0 - 15
6	управление частотой генератора шума	0 - 31

8, 9, 10	управление амплитудой каналов A, B, C	0 - 15
11	нижние 8 бит управления периодом пакета	0 - 255
12	верхние 8 бит управления периодом пакета	0 - 255
13	выбор формы волнового пакета	0 - 15
14, 15	регистры портов ввода/вывода	0 - 255

Основным при работе ПГЗ является регистр 7. Его главное назначения - определять, какие каналы должны участвовать в образовании звука и определять направление обмена портов ввода/вывода.

Его структура показана на рис.1. Ноль соответствует включению определенной позиции, а единица - выключению.

7	6	5	4	3	2	1	0
порт B	порт A:	шум C	шум B	шум A	тон C	тон B	тон A
упр.ввод/вывод	выбор канала для шума				выбор канала для тона		

Для портов "1" соответствует режиму вывода информации.

Рис.1. Регистр смешивания и выбора канала.

```
OUT 15H - задает номер регистра
OUT 14H - задает значение этого регистра

Частота генератора = 1773400/(16*((256*R1)+R0))
Частота шума       = 1773400/(16*R6)

Амплитуда: R0, R9, R10: бит 0-3 - эффективное значение
                        бит 4   - модуляция (1- активно)
Период волнового пакета P=1773400/(256*((256*R12)+R11)
```

Состояние регистра R13 определяет форму звучания сигнала. Его значения показаны в таблице. Бит 3 этого регистра определяет включен эффект или нет. (0 соответствует выключению эффекта).

Уровень звучания	Значение
0	одно снижение, затем выключение
1	одно нарастание, затем выключение
2	одно снижение, затем поддержка
3	одно нарастание, затем поддержка
4	повторящееся снижение
5	повторящееся нарастание
6	повторяющееся нарастание-снижение
7	повторяющееся снижение-нарастание

2.2. Музыкальный процессор "R-SOUND".
Разработчик - Руслан Костиневич.

222120 Беларусь,
г.Борисов, ул.Гагарина
д.67 кв.157
тел. (01777) 50-146

Подключение музыкального сопроцессора AY-3-8910 или YM2149F к "Вектору-06Ц"

В последнее время все больше разных моделей компьютеров стали оснащаться музыкальными сопроцессорами AY-3-8910/12 или YM2149F, позволяющими получать три канала звука, шума, программную регулировку амплитуды выходного сигнала, 8 режимов амплитудного вибрато, смешение звуковой частоты и шума в одном канале, управлвение частотой шума, амплитудного вибрато, атакой и затуханием звука.

В нашем "Векторе-06Ц" уже стоит таймер КР580ВИ53, вырабатывающий три канала звука, но ни о какой регулировке амплитуды выходного сигнала, шумах и прочих эффектах говорить не приходится. Были попытки некоторых программистов при помощи программных ухищрений добиться регулировки громкости и шума, но я не могу их назвать успешными, ибо "рожденный пользать, летать не может" (это о ВИ53-й). И хотя в Кишиневском центре "Компьютер" уже придумали способ замены в "Векторе-06Ц" микропроцессора КР580ВМ80А на более мощный Z80, дающий возможность работы как с обширным программным обеспечением для ZX-SPECTRUM, так и с программами для "Вектора-06Ц", тема сопряжения музыкального сопроцессора осталась "за бортом".

Именно неудовлетворенность существующим положением дел и побудила меня заняться темой подключения сопроцессора к "Вектору-06Ц". В результате изучения схем компьютеров, к которым сопроцессор уже был подключен, выяснилось, что "прицепить" программируемый генератор звука без серьезного вмешательства в схему "Вектора" из-за отсутствия некоторых необходимых для этого сигналов и несовпадения по времени других, довольно проблематично. Поэтому у меня созрело решение "навесить" сопроцессор на внешний порт ввода/вывода D27 (КР580ВВ55), подключенный интерфейсными шинами к гнезду XS5 (разъем "ПУ" компьютера), обеспечив тем самым простоту решения задачи.

И хотя во время написания статьи мне стало известно, что похожая разработка уже осуществлена в г.Кирове, я все-же решил сообщить о своей, так как она хоть и требует некоторого несущественного вмешательства в схему компьютера, но обладает несомненным преимуществом, о котором будет сказано чуть ниже.

Разработанная мной схема (рис.1) не "конфликтует" ни с одним из известных периферийных устройств массового применения (джойстик "П", джойстик "УСПИД", принтер), и позволяет в полной мере производить обмен данными между компьютером и сопроцессором, осуществлять программный и аппаратный (нажатие "ВВОД"+"БЛК") сброс ПГЗ, а также пользоваться звуковыми возможностями "родной" ВИ53-й через совмещенный аудиовыход. В результате на "Векторе-06Ц" можно играть довольно серьезные музыкальные партии, одновременно используя 6 каналов вывода звука, что является существенным плюсом по сравнению с кировской разработкой. Теперь вам вовсе не обязательно переключать шнуры из одного гнезда в другое всякий раз, когда захочется поиграть в игру, в которой обращение к программируемому генератору звука не предусмотрено и для генерации звука используется внутренний таймер ВИ53, и наоборот, когда вы желаете насладиться чудесным псевдостереофоническим звучанием музыки и игровыми спец-эффектами, вырабатываемыми сопроцессором.

Плата сопряжения с установленным на ней сопроцессором имеет 2 разъема: для подключения к гнезду "ПУ" компьютера и обычное пятиконтактное гнездо для соединения аналоговых выходов сопроцессора и таймера со стереофническим усилителем. В качестве инвертора сигнала RESET мной применена микросхема К555 ЛА3, но возможно применение любого другого имеющегося у вас инвертора, рисунок печатной платы (рис.2) придется естественно изменить. Что же касается доработки "Вектора", то вам необходимо с 7 ножки счетчика D35 (см.принципиальную схему компьютера) тактовую частоту 1.5 МГц перебросить проводом на контакт В10 разъема

"ПУ", (подпорный резистор R11 лучше не выпаивать), и с конденсатора C58 (выход на магнитофон) пустить провод на контакт B01 разъема, предварительно обрезав старую связь.

Программирование сопроцессора заключается в изменении содержимого 16 внутренних регистров, которые отвечают за высоту тона в каждом канале, частоту шума, амплитуду, эффекты и др. функции. Для того, чтобы записать данные в текущий регистр необходимо выполнить следующую последовательность команд:

```
        MVI A,4         ; Записываем управляющее слово
        OUT 5           ; и адресуем в 5 порт, сопроцессор
        MVI A,DATA      ; готов к приему данных, записываем
        OUT 6           ; данные и адресуем в 6 порт, данные
                        ; приняты.
```

Выбор регистра, в котором нужно поменять содержимое или считать из него данные осуществляется такой последовательностью команд:

```
        MVI A,6         ; Управляющее слово выбора регистра  - в 5
        OUT 5           ; порт, а номер регистра - в 6 порт. Регистр
        MVI A,NREG      ; выбран и становится текущим.
        OUT 6           ;
```

Считывание данных из текущего регистра производится командами:

```
        MVI A,2         ; Управляющее слово считывания данных - в
        OUT 5           ; 5 порт, и читаем
        IN 6            ; содержимое регистра из 6 порта.
```

Программный сброс микросхемы осуществляется еще проще:

```
        MVI A,8         ; Сопроцессор прекращает свою работу и
        OUT 5           ; обнуляет все регистры.
```

К сожалению невозможно в рамках одной небольшой статьи подробно описать функции регистров сопроцессора, и какие данные необходимо в них засылать, чтобы вызвать музыкальный тон, шум, изменение амплитуды сигнала и звуковых эффектов. Поэтому мной написана брошюра, освещающая все вопросы программирования музыкального сопроцессора применительно к "Вектору-06Ц", описаны регистры и работа с ними. Желающие могут получить готовые платы сопряжения и ее, выслав в мой адрес заявку, чистый конверт с марками и своим обратным адресом. Некоторые программисты уже работают над программами, использующими AY-3-8910 в работе, и довольно успешно. Сейчас, когда "Вектор-06Ц" оснастили процессоро Z80, станет гораздо легче адаптировать программы с "SINCLAIR"- совместимых компьютеров типа "Пентагон-128", в которых музыкальный сопроцессор "стоит" с 1984 года и за 10 лет наработано невообразимое количество демонстрационных, игровых программ и довольно мощных музыкальных редакторов.

Господа программисты! Если вы заинтересовались программированием сопроцессора, пожалуйста пишите, вам гарантирую ответ и всемерную поддержку в первую очередь.

В заключение хочу поблагодарить Филатова Олега (г.Минск) за неоценимую помощь консультационного характера при выяснении функций регистров сопроцессора, другие сведения и поддержку проекта.

1.3. Сравнительный анализ обеих разработок.

Так как есть две схемы музыкального процессора, то имеется и проблема выбора: как с пользой потратить свои деньги. Предлагаем вниманию небольшую табличку сравнения сопроцессоров по основным параметрам:

Таблица 1.3.

Параметры сравнения	Sound Tracker	R-SOUND
1. Основная музыкальная м/с	AY-3-8910	AY-3-8910
2. Аналог	YM2149F	YM2149F
3. Внешний вид	отлаженная пла-	плата без му-

	та со всеми м/с	зыкальной м/с
	без корпуса	без корпуса
4. Наличие разъемов	имеются	имеются
5. Готовность к работе (сразу после покупки)	готова	требуется м/с AY-3-8910
6. Доработка компьютера	не нужна	нужна
7. Подключается к разъему:	ВУ	ПУ
8. Конфликты с другим оборудованием	отсутствуют	отсутствуют
9. Наличие схемы и чертежа печатной платы	нет	имеются
10. Совмещен ли звуковой выход муз.процессора и таймера	да	да
11. Тактовая частота	1.7734 МГц	1.5 МГц
12. Совместимость с музыкой, написанной на "Пентагоне-128"	полностью	не полностью
13. Наличие программного обеспечения на май 1994 г.	имеется - тест - 5 демонстрашек - player снятой с "Пентагона" музыки - практически готов редактор	имеется - тест
14. Автор разработки находится	Ведется дальнейшая разработка в России	Ведется дальнейшая разработка в Беларусии

Небольшой комментарий по поводу таблицы. Пункты 4,5,6 и 8 определяют "сервисность" разработки, т.е. требуются ли от пользователя какие-то знания и усилия для подключения сопроцессора. "Sound Tracker" выгодно отличается тем, что для его подключения нужно просто вставить плату в разъем ВУ и подсоединить шнур к магнитофону, после этого можно запускать тест или другую программу. Если же разъем ВУ занят контроллером или квазидиском, то возможны два варианта: а) вилка; б) отпаять от платы сопроцессора разъем ВУ и подпаять провода паралельно на уже существующий разъем ВУ. Последнее относится и к "R-SOUND" (только для ПУ).

Пункт 9 интересен тем, кто разбирается в схемах, хочет изготовить плату самостоятельно или же озабочен возможным ремонтом. По этому пункту впереди "R-SOUND". Надо лишь заметить, что платка муз.процессора такая маленькая и на ней располагается такое небольшое количество деталей, что ломаться почти нечему.

Руслан Костиневич отмечает как главное достоинство своей схемы - совмещенный выход с муз.процессора и таймера. Однако это не так. Виктор Саттаров предусмотрел такую возможность в "Sound Tracker". Для это нужно один проводок выхода на магнитофон (не "землю" конечно) перебросить на плату муз.процессора в точку отмеченную как "BEEP".

Пункты 11 и 12 говорят о том, что "Sound Tracker" по тактовой частоте совпадает с соответствующими разработками для "Пентагона". Следовательно можно просто брать блоки музыкальных данных с Пентагона и кидать их в муз.процессор на Векторе - музыка пойдет один к одному. Именно из-за этого, кстати, на плате Виктора Саттарова больше корпусов. Руслан Костиневич пожертвовал полной совместимостью своей разработки, сократив при этом число корпусов м/с.

По пункту 13 можно судить о реальной программной поддержке на сегодняшний день обеих разработок. Так как "Sound Tracker" появился раньше, то и программ для него больше. Кроме того, он уже начал активно поку-

паться. Руслан Костиневич может попытаться за счет удешевления стоимости и мощной рекламной кампании отвоевать часть рынка. Рекламная компания действительно им ведется: это и статьи в популярные издания и скидка при покупке или бесплатное предоставление платы разработчикам программного обеспечения.

Пункт 14 введен из-за того, что российским пользователям не все равно откуда получать заказанный сопроцессор. Беларусь - другая республика, а значит более высокие почтовые тарифы, проблемы с пересылкой денег, более долгие сроки выполнения заказа.

Окончательный выбор вы сделаете, приобретя тот или иной музыкальный процессор. На сегодня вопрос стоит "что именно приобрести", а не "стоит ли вообще приобретать", так как музыкальный процессор - будущее вашего компьютера уже сейчас!

2. Новое ПЗУ для "Вектора-06Ц"

2.1. ПЗУ "Load Byte".

В фирме "Байт" разработан новый загрузчик - "Load Byte". Загрузчик позволяет осуществлять загрузку программ с магнитофона и дисковода. Для пользователей магнитофонной конфигурации "Вектора-06Ц" введен ряд сервисных удобств:

- вывод имени загружаемого файла на экран;
- пометка начального и конечного блоков на экране;
- возможность загрузки сбойных программ как в Copy-N4 (при сбое программа не сбрасывается, а продолжает грузиться. Ленту можно проматывать вперед-назад и повторно грузить сбойные участки). Причем загрузчик сам определит, когда программа загрузится без ошибок ;
- специальный маркер указывает на экране на текущий загружаемый блок;
- возможность загрузки программы прямо с середины с выводом имени загружаемой программы на экран. Это значительно облегчит поиск нужной программы и ее ввод тем, кто пользуется магнитофонами без счетчиков;
- действие клавиши УС сохраняется.

Загрузчик зашит в м/с ПЗУ РФ5 объемом 2 кбт. Прилагается инструкция по замене м/с ПЗУ.

2.2. Внешнее ППЗУ.

Практически готово внешнее ППЗУ объемом 64 кбт. Разработчик - Виктор Саттаров. Внешне ППЗУ выглядит в виде готовой к работе платы, вставляемой в разъем ВУ вашего компьютера. Плата имеет ответный разъем ВУ, через который можно подключать другие внешние устройства (контроллер, квазидиск, муз.процессор). Плата при включении компьютера перехватывает управление и вы можете выбрать любую из прошитых в ППЗУ программ. Содержимое ППЗУ может формироваться по желанию заказчика. Применение ППЗУ резко облегчит работу на компьютере. Такие популярные программы, как Бейсик, Монитор-Отладчик, операционная система, различные копиры, будут всегда под рукой.

Вначале хорошая новость для всех клиентов фирмы "Байт": теперь вы вместе с заказом будете получать бесплатно самый свежий выпуск информационного выпуска "Байт" в виде файла! Вы будете в курсе всех последних новостей по "Вектору", а также узнаете немало интересного о программах и программировании! Уточним, что этой льготой будут пользоваться только те, кто сделал заказ на программное обеспечение. Работаем под девизом "Нашим клиентам - бесплатное информационное сопровождение"!

Напоминаем наш почтовый адрес: 610031, г.Киров, а/я 2629, Луппову Григорию Борисовичу.

Для тех, кто перешел на IBM, Вы можете заказывать программы для IBM и по почте. Каталог - бесплатно (в виде файла). Нужно только оплатить стоимость пересылки (500 руб.) и стоимость носителя (кассета/дискета). Каталог состоит из трех частей: полноценные игрушки (EGA/VGA), CGA-игрушки и системно-прикладные программы. Каталог можно записать на носитель в формате "Вектора"(в виде ROM-файлов или в виде текстовых файлов) или на носитель в формате IBM.

МУЗЫКАЛЬНЫЕ ПЛАТЫ ДЛЯ ВЕКТОРА

Для тех, кто не слышал, напоминаем, что в городе Кирове впервые была разработана музыкальная плата "Sound Tracker", аналог музыкальной платы компьютеров типа "Синклер", которая внесла новые возможности в обработке звука на "Векторе". Потрясающая стерео-музыка, возможность простого переноса наработанных музыкальных файлов с "Синклера" на "Вектор" привлекли внимание как многочисленных пользователей, так и разработчиков программно-аппаратного обеспечения. В настоящее время плата активно приобретается всеми, кто хочет идти в ногу со временем. Появились и аналогичные разработки: вначале в г.Борисове (Беларусь), позднее в г.Омске. Сегодня мы представляем принципиальные схемы кировской и омской разработок. Надеемся, что вы заинтересуетесь расширением возможностей вашего компьютера и тоже приобретете что-нибудь в этом духе.

На рис.1 представлена схема кировской разработки. В этой схеме возможны следующие изменения:

1. R12=R14 влияют на соотношение громкости между AY и стандартным выходом "Вектора". Если вы считаете, что стандартный звук компьютера громче, чем звук от AY - увеличьте эти сопротивления.

2. Если вам кажется, что звук с AY очень тихий можно заменить R13=R11 и R9=R10, соблюдая при этом соотношение:

$$\frac{R11=R13}{R10=R9} = \frac{1}{2}$$

3. На плате не установлены C1=C2, так как у подавляющего большинства усилителей они стоят на входе. Но возможно вам самим захочется их установить. При этом с емкостью придется поэкспериментировать, подбирая ее от 0.1 мкф до 10 мкф.

На рис.2 представлена схема альтернативной разработки из г.Омска фирмы Spase Corporation. Данная схема является полным функциональным аналогом сопроцессора, разработанного Виктором Саттаровым.

Руслан Костиневич из г.Борисова, также продолжает работу по модернизации своей разработки - музыкалной платы "R-Sound". Он разработал плату "R-Sound-2", которая теперь полностью совместима с музыкальными отрывками, снятыми со "Спектрума-128". Плата выпускается промышленным способом, с металлизацией отверстий, и потому трудна для самостоятельного повторения. Теперь слово автору.

ЕЩЕ РАЗ О СОПРОЦЕССОРАХ

Прочитав в ИРВ "Байт-23" довольно объективную оценку разработок по подключению музыкального сопроцессора к "Вектору", я согласился с некоторыми выводами, сделанными в нем, в частности с тем, что ни одна из описанных разработок не является идеальной, недостатки есть и у "Sound Tracker" и у "R-Sound", хоть и не одинаковые.

С целью создания наилучшего варианта подключения муз.процессора к компьютеру мной была разработана новая плата "R-Sound 2", в которой устранены все недостатки, присущие как плате "R-Sound", так и "Sound Tracker". Для большей наглядности лучше всего представить результаты сравнения разработок в виде таблицы (см. таб.1).

Плата "R-Sound 2", как и предыдущая разработка устанавливается в гнездо "ПУ", имеет собственный генератор тактовой частоты, и обеспечивает полное соответствие музыкального тона с разработками для "ZX Spectrum-128". Простота схемы и небольшое количество деталей обеспечивают высокую надежность, а наличие на плате залуженных отверстий для второго гнезда "ПУ" позволяет пользователю (если он работает с принтерами или джойстиками) припаять розетку разъема на предусмотренное для нее место, или еще проще, припаять провода, идущие от джойстика или принтера к контактам на плате, полностью соответствующим контактам разъема "ПУ". Очевидно, что проделать такие операции намного проще с разъемом "ПУ", нежели с разъемом "ВУ". Желающим автор предлагает комплект для самостоятельной сборки, состоящий из платы, разъема "ПУ", панельки под музпроцессор, схемы и документации по сборке и наладке. Микросхема музпроцессора высылается по дополнительной заявке.

В заключение привожу цитату из статьи редактора "Байта": "На сегодняшний день вопрос стоит "что именно приобрести", а не "стоит ли вообще приобретать", так как музыкальный процессор - будущее Вашего компьютера уже сейчас!". Думаю, ознакомившись с этой статьей Вы будете знать, что именно приобрести.

Костиневич Р.Л.
тел. (01777) 50-149

Таблица 1.

Параметры оценки	Sound Tracker	R-Sound	R-Sound 2
1. Тип сопроцессора	AY-3-8910	AY-3-8910	AY-3-8910
2. Аналог	YM2149F	YM2149F	YM2149F
3. Техническое исполнение	Отлаженная плата со всеми м/с без корп.	Отлаженная плата со всеми м/с без корп.	Отлаженная плата со всеми м/с без корп.
4. Размеры		45*80 мм	53*68 мм
5. Наличие разъемов	Имеются	Имеются	Имеются
6. Доработка ПК	Не нужна	Нужна	Не нужна
7. Подключение к	ВУ	ПУ	ПУ
8. Программные конфликты с др.оборудованием	Отсутствуют	Отсутствуют	Отсутствуют
9. Аппаратные конфликты с др.оборудованием	Имеются	Имеются	Отсутствуют
10. Наличие схемы и чертежа печатной платы	Имеются	Имеются	Имеются
11. Совмещен ли аудиовыход сопроцессора и таймера (ВИ53)	Да	Да	Да
12. Тактовая частота	1.7734 МГц	1.5 МГц	1.7734 МГц
13. Совместимость с музыкой с "Пентагона-128"	полная	не полная	полная
14. Наличие места для установки гнезда ВУ или ПУ	нет	нет	есть
15. Наличие ПО	- тест - 5 демо - player Ведется дальнейшая разр.	- тест, 1 демо + все ПО для "Sound Tracker"	
16. Автор разработки	В России	В Беларуссии	В Беларуссии
17. Стоимость доставки оплачивает	Входит в стоимость заказа	Продавец	Продавец
18. Цена	27 тыс.руб. сентябрь 1994	15 тыс.руб. июль 1994	18 тыс.руб июль 1994

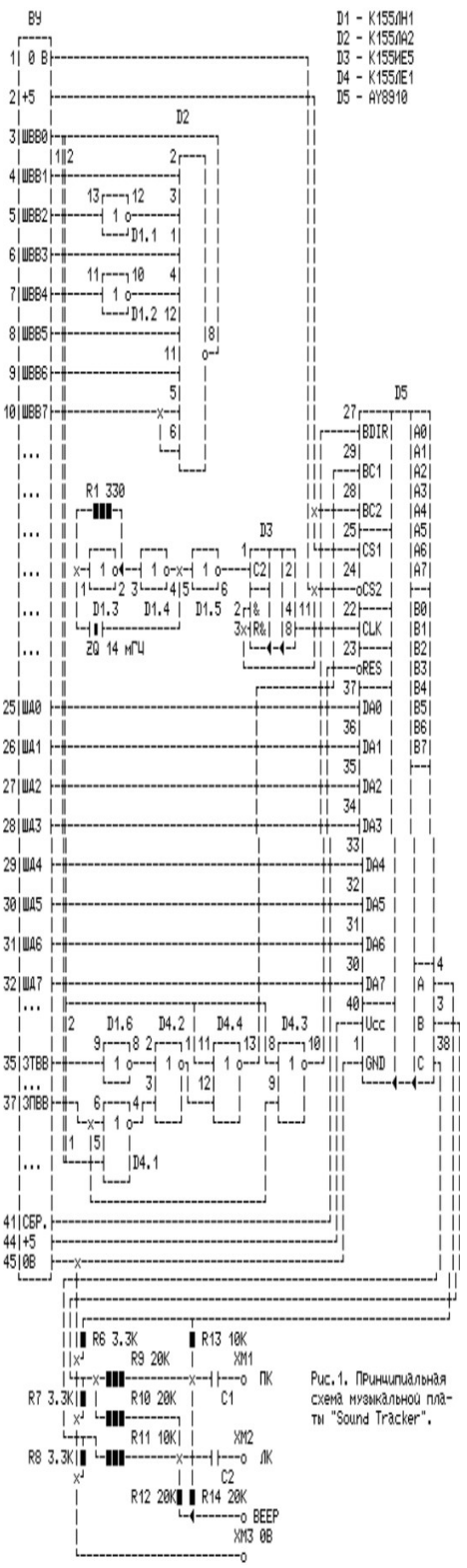


Рис.1. Принципиальная схема музыкальной платы "Sound Tracker".

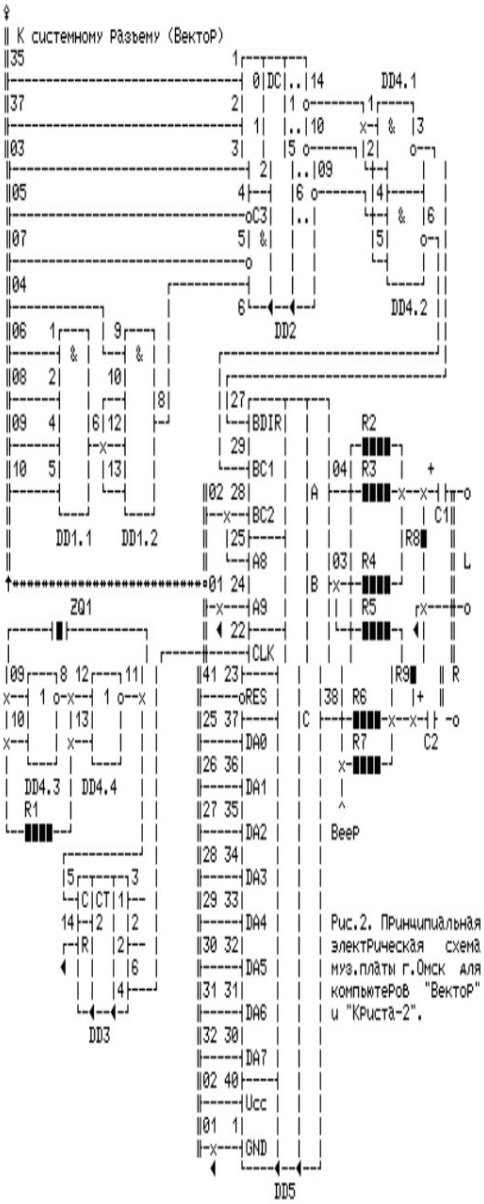


Рис.2. Принципиальная электрическая схема муз.платы г.Омск для компьютеров "Вектор" и "Криста-2".

DD1 К555ЛИ6
DD2 К555ИА7
DD3 К555ИЕ7
DD4 К555ЛА3
DD5 AY-3-8910 (YM-2149F)

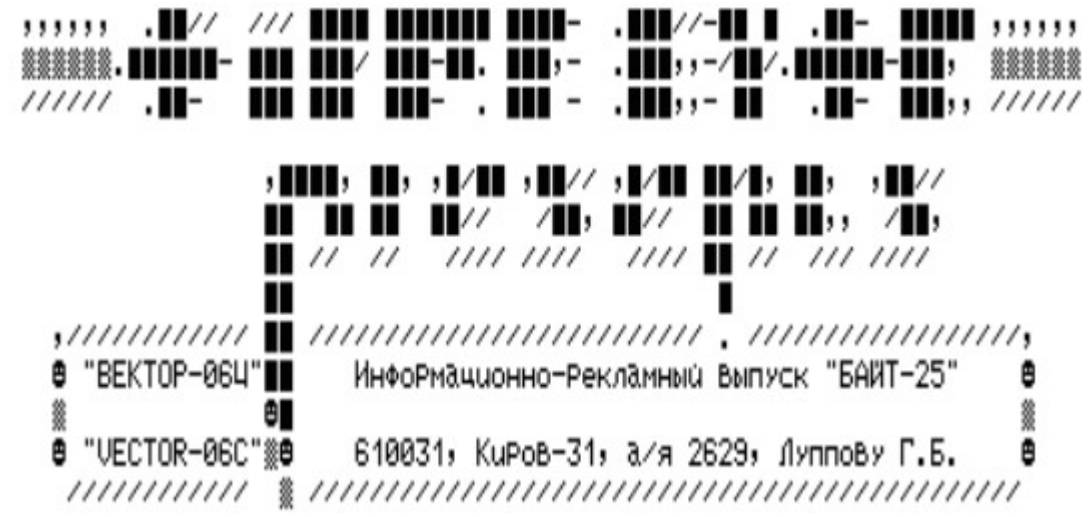
R1 330 R6 5.1 k C1,C2 10.0x10в
R2 4.7 k R7 4.7 k
R3 5.1 k R8 10 k ZQ1 14000 кГЧ
R4 10 k R9 10 k
R5 10 k

Соответствие сигналов "Криста-2" и "Вектор-06Ц"

Вектор	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
Криста	38	39	37	36	35	34	33	32	31	30	38	26	22	24	27	25	23	29	09	38

Вектор	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Криста	03	21	28	38	15	13	11	10	12	14	16	17	38	19	05	20	01	08	18	07

Вектор	41	42	43	44	45
Криста	04	38	06	39	38



Гуменюк Н.В. Почта наших читателей Саяногорск

Здравствуйте, центр "Байт"!

Пишет Вам один из пользователей и поклонников ПК "Вектор-06Ц". Прежде всего хочу выразить признательность за создание ИРВ "Байт" для пользователей "Вектора". Я, и думаю многие пользователи, несомненно Вам за это очень благодарны. Немного о себе. Компьютерами занимаюсь примерно четыре года. Первые два года работал на "БК", "МС", "ZX SPECTRUM", потом на "Векторе". В данный момент занимаюсь усиленным изучением ассемблера, а так же составлением программ на Basic v2.5, с применением процедур на кодах. Но к сожалению знания ассемблера у меня весьма ограничены. В большинстве случаев приходиться обходиться командами Бейсика. За несколько лет работы с Бейсиком я достиг определенных результатов и у меня возникло желание поделиться с начинающими пользователями некоторыми моими знаниями. Многие скептически относятся к программам, написанным на Бейсике, но я смело могу заверить, что они ошибаются. При правильном подходе даже на Бейсике можно написать более менее полезную программу. Я в частности могу показать это на своем примере. Я со своим братом заканчиваем программу на Бейсике "ARTIST" (графический редактор). Программа имеет удобное меню в виде оконного интерфейса с "падающими" окнами нескольких степеней вложения. Имеется четырнадцать основных режимов, в виде графических пиктограмм.

1. Создание рамки - выбор рамок для будущего рисунка.
2. Калькулятор - производит `+, -, *, /, кв.корень, %`
3. Деловая графика - отображение процентного соотношения в виде цветных объемных столбиков.
4. Окна - инверсия окна, переворот окна, увеличение окна, копирование окна, перенос окна, растягивание окна.
5. База данных - набор готовых геометрических фигур, построение трехмерной графики, манипуляция трехмерными объектами, создание фона.
6. Знакогенератор - редактирование знакогенератора, загрузка, отгрузка.
7. Магнитофон - отгрузка, загрузка картинки.
8. Принтер - распечатка на принтере.
9. Карандаш - рисование отрезков, лучей, ломанных линий, точек, прямоугольников, окружностей. Выбор нескольких

типов кистей, напыление, заливка и др.
10.Текст - вывод текстовых надписей различных размеров. Спец.режимы: автоцентрирование, планирование, печать сверху вниз, по диагонали и др.
11.Экран - инверсия экрана, стирание экрана, просмотр.
12.Маска - выбор маски из 16 представленных, редактирование маски.
13.Цвета - выбор цвета из 16 представленных.
14.END - конец работы, восстановление основных параметров Бейсика, выход в Бейсик.

Даже не умеющий рисовать, не будет расстроен, т.к. программа практически сама помогает нарисовать рисунок. Конечно, такое возможно сделать лишь при самой строжайшей экономии памяти, и тщательном планировании будущей программы.

В одном из первых выпусков ИРВ "Байт" была напечатана хорошая статья "Как написать игровую программу на Бейсике". Хочу предложить материал с немного другим уклоном. Т.к. игры на Бейсике получаются не очень серьезны, то лучше использовать его под программы, которые могут принести реальную пользу, для пользователей. Пусть хоть небольшую. Я имею в виду прикладные программы (графические редакторы, базы данных, текстовые редакторы и др.). Это моя точка зрения. В ниже приведенной статье, я попробовал дать то, через что я уже прошел сам. Так сказать информация для размышления.

Прикладные программы на Бейсике или несколько советов для начинающих

В одном из номеров ИРВ "Байт" была опубликована статья для начинающих программистов "Как написать игровую программу на Бейсике". В данной статье хочу продолжить разговор на тему "с чего начать". Будут даны общие направления для тех, кто хочет попробовать написать прикладную программу, в частности, графический редактор.

Для начала советую задать себе несколько вопросов:

1. Будет ли ваша будущая программа приносить реальную пользу?
2. Не будет ли она повторять уже имеющиеся программы?
3. Внесет ли ваша программа что-нибудь новое, или она будет похожа на все остальные программы?
4. Будет ли ваша программа дружелюбна к возможному пользователю, а так же проста в обращении?
5. Хватит ли ваших знаний, чтобы осуществить все то, что вы задумали?
6. Хватит ли сил у Бейсика, чтобы выдать все то, что Вы решили сделать?

Сначала точно наметьте цель, а затем пробуйте ее осуществить в своей программе. Ищите самый простой и короткий путь. Точно рассчитывайте свои знания, их должно хватать для достижения намеченной цели. Старайтесь как можно тщательнее планировать свою будущую программу. Советую завести небольшую тетрадь, в которой можно делать все расчеты, пометки, структурные схемы, эскизы и наброски.

Когда, в зависимости от поставленной задачи, вы решите, что вам нужно, то можно приступать к планированию будущей программы. Допустим, вы решили попробовать сделать графический редактор.

Различают три основных режима работы графических редакторов: создание и редактирование на экране компьютера картинок, спрайтов и наборов символов. Наличие всех трех режимов необязательно, но возможность редактирования всей картинки - отличительный признак графического редактора как такового. Исходя из этого, вы должны продумать, чем и как рисовать и редактировать картинку, спрайт, знакогенератор и т.д. Рассмотрим первый случай, т.е. рисование полноэкранной картинки. Самое ми-

нимальное, что должен уметь ваш графический редактор - это рисовать простейшие графические примитивы: отрезки, прямоугольники, окружности. Конечно, скорее всего вам захочется сделать и многое другое. Возникает мысль о создании определенного меню, где все режимы в наглядной форме будут выведены на экран. Каждый редим в этом меню можно выделить небольшой пиктограммой. Например, если пиктограмма соответствует режиму рисования окружностей, то и картинка или надпись на ней должна наглядно выражать это. Меню можно разместить в верхней части экрана, можно в нижней - это ваше дело. Главное, чтобы оно было максимально удобным. Постарайтесь, чтобы меню было небольших размеров, т.к. оно затирает часть экрана, т.е. часть рисуемой картинки. Есть один простой выход - это запоминание через оператор GET части изображения, которое находится под меню. Но смотрите, это повлечет за собой определенный расход памяти Бейсика. Иногда можно пожертвовать частью экрана, т.е. делать меню намертво. В целях экономии памяти, меню можно не рисовать из программы, а сделать подгружаемым в Видео ОЗУ. Когда вы нарисовали меню, следует позаботиться об активизации того или иного режима из вашего меню. Соответствующая подпрограмма должна перемещать какой-нибудь маркер, а также производить перерасчет координат этого маркера или наращивать, либо уменьшать произвольную переменную. Это нужно для оператора ON (переменная) GOTO (номера строк, по которым находится тот или иной режим). К примеру у вас десять основных режимов. Тогда выбранная переменная будет изменяться от 1 до 10, в зависимости от того, в каком направлении движется маркер по меню. Таким способом можно легко сделать выбор нужного режима. Постарайтесь, чтобы вся работа с выбором какого-нибудь режима из меню осуществлялась всего тремя клавишами. Напрмер, если меню горизонтального типа, можно выбрать: стрелки вправо, влево и пробел. Стрелки - перемещение маркера, пробел - активизация нужного режима.

Теперь приступим к написанию режимов. Обычно при активизации режима на экране появляется маркер в виде стрелки, квадратика, окружности и т.п. На этом этапе можно сэкономить немного памяти. Например, использовать всего один маркер, причем взять его точно таким же, как и в меню. Особенно, если маркер был сделан в виде спрайта, незатирающего изображение. В самом начале сделайте маркер в виде стрелочки и используйте его всегда, когда возникнет необходимость. Не стоит заново прописывать программу для передвижения главного маркера, возьмите ту, которая передвигает маркер по вашему меню. Чтобы это было можно сделать, позаботьтесь об этом заранее. Для более быстрого перемещения спрайта маркера делайте его не более, чем 8*8 точек. Следует позаботиться и о шаге перемещения. Желательно сделать его двухрежимным. Первый - смещение на одну точку, а второй - на 10-15. Второй нужен для быстрого перемещения маркера из одного угла экрана в другой. Теперь к блоку перемещения маркера пристыкуйте блок ограничения границ экрана, опрос клавиши изменения шага, выхода в меню, а также активизации уже выбранного режима (это может быть установка точки, центра будущей окружности и т.д.). Сейчас кратко рассмотрим реализацию некоторых режимов.

1. Проведение отрезков.

Вы перемещаете маркер по экрану, соответственно изменяются две условные переменные X и Y. Во всех наших примерах X и Y - это координаты заостренного кончика стрелки - маркера. Пробел - активизация (условная фиксация) маркера. Вот вы решили провести отрезок. Поставили маркер в нужное место и нажали пробел. Переменные X и Y зафиксировали начало будущего отрезка. Сохраним значения X и Y, воспользовавшись дополнительными переменными A и B. Таким образом, A=X, B=Y. Теперь оператором GOTO опять запустим программу движения маркера. Ставим маркер в нужное место и нажимаем пробел. Фиксируются уже измененные X и Y - координаты конца линии. Осталось самое простое - это запустить строку:

PLOT A,B,1:LINE X,Y
Отрезок проведен. Повторяем все сначала.

2. Рисование окружностей.
Ставим маркер в нужное место и нажимаем пробел. Переменные X и Y зафиксировали центр будущей окружности. Сохраним значения A=X, B=Y. Опять перемещаем маркер, теперь уже для определения радиуса окружности. Нажимаем пробел. После этого переменные A и B - это координаты центра, X и Y - это граница окружности. Надо высчитать радиус. Воспользуемся формулой, которая определит расстояние от A,B до X,Y.
R=SQR((A-X)^2+(B-Y)^2)
Выполняем строку: CIRCLE A,B,R.

3. Рисование прямоугольников.
Все аналогично режиму "проведение отрезков".
A,B - нижний левый угол прямоугольника.
X,Y - верхний правый угол прямоугольника.
PLOT A,B,1: LINE X,Y,B

4. Ставим точку.
Самый простой режим.
PLOT X,Y,1 , где X,Y - координаты маркера.

5. Выбор цвета.
Режим, который определяет текущий цвет. Т.е., если выбрали целеный цвет, то и все другие режимы будут использовать зеленый. Все отрезки, окружности и т.д. будут рисоваться зеленым цветом. Этот режим можно сделать методом перебора цветов. В меню нарисуйте прямоугольник (закрашенный), он будет отображать текущий цвет. При нажатии на определенные клавиши, этот прямоугольник будет поочередно менять цвет. Таким образом, можно будет видеть, какой в данный момент времени установлен. Не забудьте при переборе цвета изменять переменную, которую затем нужно будет подставлять в оператор COLOR при активации других режимов. Этот метод удобен тем, что отпадает надобность в показе всех цветов одновременно. Цветов может быть и 8, и 16, а меню по своим размерам может быть очень маленьким.
Выбор цвета можно организовать в виде "горячих" клавиш, т.е. в блоке управления главным маркером вы опрашиваете еще и клавиши смены цвета. Можно задействовать ряд цифровых клавиш. Это удобно тем, что не надо лишний раз входить в меню для смены текущего цвета. Нажал клавишу - цвет сменился. Не удобно тем, что если у вас используется 16 цветов, то и опрашивать надо 16 клавиш, что замедляет скорость движения главного маркера.

6. Заливка.
Как сделать этот режим, вы наверное догадались сами. Закраска производится текущим цветом, через оператор PAINT. Вся проблема в определении границ закраски. Эту проблему можно решить несколькими путями. Самый простой - добавить новый режим в меню, где будет определяться текущий цвет границы закраски. Можно попробовать определять значение границы закраски через оператор POINT. Относительно центра начала закраски (X, Y) сдвигать в одном из направлений мнимую точку, проверяя ее через POINT. Если будет встречен любой цвет, не равный цвету фона, то он автоматически становится цветом границы закраски.
Пример:
```
10 X=X+1
20 PLOT X,Y,2
30 G=POINT(1)
40 GOTO 10
```

Мнимую точку сдвигаем вправо, относительно координат главного маркера - X и Y. Переменная G будет принимать значения встреченного цвета. Есть и более ухищренные способы определения границ закраски, но о них в другой раз.

После того, как вы научились делать столь простые режимы графического редактора, можно подумать и о реализации более мощных процедур рисования и редактирования экранных картинок. Любой уважающий себя графический редактор имеет мощную систему окон. Вы задаете произвольного размера окно, а потом проделываете с ним все, что захотите - инвертирование, скроллинг, зеркальное отображение, увеличение, переворот, копирование, перенос и даже растягивание картинки в окне. На Бейсике все это возможно и проверено на практике. Причем все это можно сделать, не применяя процедур, написанных в машинных кодах. Ограничение накладывается лишь на размеры задаваемого окна. Его делают одноразмерным для всех перечисленных режимов.

Большое внимание нужно уделить режиму "текст", т.к. вывод надписей при создании картинок, заставок бывает просто необходим. При желании можно осуществить такие экзотические режимы, которые существуют лишь в лучших графических редакторах на ПК "ZX SPECTRUM". Это редакторы "ART STUDIO" и "ARTIST2". Вот некоторые из них: режим "Thicken" (утолщение) и режим "Outline" (обводка). Допустим, вы нарисовали отрезок. Первый режим утолщит его в два раза, а второй аккуратно обведет. Это может быть и не просто отрезок, а картинка любой степени сложности. На Бейсике это сделать легко. Поэтому не стоит останавливаться на достигнутом, пробуйте, экспериментируйте и все получится.

Экзотические режимы для графических редакторов на Бейсике

В данной статье хочу привести несколько программ, которые продемонстрируют достаточно интересные режимы для графических редакторов, написанных на Бейсике. Если кто-то из начинающих пользователей захочет попробовать сделать графический редактор, то ниже приведенные программы могут быть очень полезны. В статье будут рассмотрены и такие режимы, которые отсутствуют даже у таких редакторов как "Карандаш" и "Рембрандт".

Любой уважающий себя редактор обычно имеет мощную систему окон. Пользователь может задать окно произвольного размера, навести его на нужный фрагмент картинки, а затем произвести определенное преобразование с этим фрагментом. Например, перевернуть фрагмент картинки, попавший в окно, либо увеличить его, инвертировать и многое другое. Для редакторов, написанных на Бейсике, окно делают обычно одноразмерным, так как сделать редактор, где окно можно задавать произвольных размеров достаточно сложно. Особенно, если используются все 16 цветов, то есть свободная память Бейсика ограничена 15.5 кбт. Хотя 15.5 кбт не так уж и много, но если все хорошо продумать, то и этого вполне достаточно, чтобы сделать приличный редактор, даже если и ограничиться системой окон одного размера. Надо сказать, что большинство очень сильных и сложных режимов, использующих окно, делаются по принципу сканирования изображения из окна в массив, равный размеру окна. Через оператор POINT определяется математический цвет каждой точки изображения в окне и затем помещается в массив. Когда он заполняется, определенная программа делает преобразование с числами массива. После того, как массив будет обработан, изображение считывается из него опять на экран, но уже в измененной форме. Правда использование массива хоть и облегчает работу программисту, но ведет к проигрышу в быстродействии программы. Поэтому, если есть возможность обойтись без массива, его не используют, то есть все преобразования делают прямо в окне. Если это по каким-то причинам невозможно, то изображение из окна копируется в меню,

там проходит преобразование, а потом уже возвращается обратно. Либо изображение сканируется, преобразуется и перекидывается одновременно с копией в окно. Большинство приведенных программ именно на этот метод и рассчитаны. Взять, к примеру, режим увеличения фрагмента картинки. Может произойти такая ситуация, когда увеличенное изображение начнет накладываться на окно, откуда происходит сканирование. Если окно, содержащее фрагмент картинки испорчено, то что же тогда увеличивать?Поэтому надо использовать один из вариантов:

1. Перед увеличением сделать копию изображения из окна в безопасное место, например, в меню. После чего сканирование изображения можно проводить и с копии.
2. Выводить увеличенное изображение в меню, тогда и картинка не пострадает и окно.

Выше сказанное справедливо и для некоторых других режимов графических редакторов.

Продолжение статьи - в следующем номере

|Самойлов С.А. Вечная жизнь в программах Элиста|

```
           Predator        - по адресу 0635H записать 00H
           Color Ball      - по адресу 0B2FH записать 00H
           Гротоход        - по адресу 1060H записать 00H
                           - по адресу 1064H записать 00H
                           - по адресу 1065H записать 00H
                           - по адресу 1066H записать 00H
           Eric            - по адресу 0CF2H записать 00H
           Diamond Country - по адресу 089FH записать 00H
                           - по адресу 08A0H записать 00H
           Blaster         - по адресу 2E94H записать 00H
```

продолжение. начало в номере 25

Гуменюк Н.В. — Почта наших читателей — Саяногорск

Для начала рассмотрим режим "MAGNIFY" (увеличение) части картинки. Это один из самых важных режимов, т.к. рисование мелких деталей картинки очень утомляет глаза, да и рисовать значительно труднее, поэтому режим увеличения просто необходим. Наберите небольшую программу, запустите ее и посмотрите как будет увеличено окно размером 11*11 точек.

```
10 CLS: COLOR 15,0
20 X=50: Y=5-: Z=101: H=101
30 PLOT 50,50,1: LINE 60,60,B: PLOT 53,52,2:
   LINE 1,1,BS: PRINT "S": PAINT 51,51,4,15
40 COLOR 10: FOR A=100 TO 210 STEP 10: PLOT A,100,1:
   LINE A,210: PLOT 100,A,1: LINE 210,A: NEXT A
50 FOR I=1 TO 121
60 PLOT X,Y,2: IF POINT(1)>0 THEN COLOR POINT(1):
   PLOT Z,H,1: LINE Z+8,H+8,BF
70 X=X+1: Z=Z+10: IF X=61 THEN X=50:Y=Y+1: Z=101: H=H+10
80 NEXT I
```

Комментарий:

```
20 X,Y - нижний левый угол окна.
   Z,H - нижний левый угол первого прямоугольника.
         Из этих прямоугольников и строится увеличенное изображение.
30 Рисуем графику в окне
40 Устанавливаем цвет разделительной сетки. Сетка применяется для наг-
   лядного разделения прямоугольников в увеличенном фрагменте картинки
50 Задаем цикл: сколько точек нужно увеличить (в нашем случае
   11*11=121)
60 Через оператор POINT узнаем математический цвет точки в окне. Меня-
   ем цвет: COLOR POINT(1). Рисуем закрашенный прямоугольник:
   PLOT Z,H,1: LINE Z+8,H+8,BF. Этот прямоугольник и есть увеличенная
   точка.
70 Проводим смещение вправо. Как в реальном окне, так и в увеличенном.
   Узнаем цвет точки, которая стоит правее и т.д. Пока не будет увели-
   чен весь нижний ряд в заданном окне, то есть 11 точек.
   Если ряд пройден, то поднимаемся на ряд выше и т.д., пока все изоб-
   ражение в окне 11*11 точек не будет увеличено.
80 Повторяем цикл.
```

В строку 30 можно вставить любую графику, используя все 16 цветов. Все, что попадает в окно 11*11 точек, будет увеличено. Теперь, если

кто хочет стать обладателем микроредактора, может набить дополнительные строки 90-220. Это дополнение позволит редактировать изображение в увеличенном окне всеми 16 цветами. Клавиши управления: стрелки - перемещение маркера, ВК - перебор цветов, ПРОБЕЛ - установить точку.

```
 90 X=50: Y=50: Z=100: H=100: C=15
100 GOSUB 200: GOSUB 220
110 A$=INKEY$
120 IF A$=CHR$(24) AND Z<200 THEN GOSUB 190: Z=Z+10; X=X+1:
    GOSUB 200
130 IF A$=CHR$(8) AND Z>100 THEN GOSUB 190: Z=Z-10: X=X-1:
    GOSUB 200
140 IF A$=CHR$(25) AND H<200 THEN GOSUB 190: H=H+10: Y=Y+1:
    GOSUB 200
150 IF A$=CHR$(26) AND H>100 THEN GOSUB 190: H=H-10: Y=Y-1:
    GOSUB 200
160 IF A$=CHR$(13) THEN C=C+1: GOSUB 210
170 IF A$=" " THEN COLOR C: PLOT Z+1,H+1,1: LINE Z+9,H+9,BF:
    PLOT X,Y,1
180 GOTO 110
190 COLOR 10: PLOT Z,H,1: LINE Z+10,H+10,B: RETURN
200 COLOR 15: PLOT Z,H,1: LINE Z+10,H+10,B: RETURN
210 IF C=16 THEN C=0
220 COLOR C: PLOT 150,50,1: LINE 160,60,BF: RETURN
```

Комментарий:

```
90 X,Y - координаты по которым будут ставиться точки в реальном окне,
         то есть в окне 11*11 точек.
   Z,H - координаты нижнего левого угла маркера, которым вы управляете
   C   - текущий цвет.
100  Рисуем маркер и выводим закрашенный прямоугольник, он определяет
     текущий цвет. Для этого вызываем подпрограмму.
110-150  Управление маркером, смещение координат в реальном окне, по
         которым ставится точка.
160  Перебор цвета. Вызываем соответствующую п/п.
170  Рисуем закрашенный прямоугольник, то есть увеличенную точку, в
     увеличенном окне и точку в реальном окне. Все это происходит,
     когда вы нажимаете клавишу "ПРОБЕЛ".
180  Отправляемся на строку 110 для повтора
190  Подпрограмма. Тушим маркер, таким образом восстанавливаем разде-
     лительную сетку.
200  Подпрограмма рисования маркера.
210-220  Подпрограмма для перебора цвета.
```

Если у кого-то есть желание, то эту программу можно усовершенствовать, изменить, переменить клавиши управления и т.д. Блок управления можно заменить любым другим более эффективным блоком, но в данный момент важно понятие работы программы, а не высокая степень эффективности. Для использования программы в своем редакторе нужно сделать все необходимые координаты перемещаемыми. Это переменные X и Y, в нашем примере они равны 50. Когда вы напишите блок управления окном, то переменные X и Y должны определять нижний левый угол этого окна. Для изменения размеров окна уделите внимание строке 70. Там проверяется, когда пройден весь ряд, а так же шаг смещения по увеличенному и реальному окнам. Думаю, со всем этим проблем не будет.

Один совет. Не стоит увлекаться большим масштабом увеличения и размером окна, так как увеличенное изображение окна выводится прямо поверх рисуемой картинки, то есть портит ее. Поэтому перед выводом увеличенного изображения предварительно оператором GET запоминают часть кар-

тинки, которая будет испорчена, а затем, когда режим увеличения перестает быть нужным, эту часть картинки восстанавливают оператором PUT. Так вот, если масштаб увеличения и окно будут большими, то будет запорчен большой участок рисуемой картинки, для сохранения которого придется использовать очень большой массив, что приведет к огромному расходу памяти. Вывод увеличенного изображения можно сделать в меню, если оно, конечно, позволяет это сделать. Оптимальнее всего использовать окно размером не более 20*20 и масштаб увеличения в 4-6 раз. Для экономии памяти массив, используемый для сохранения от порчи картинки, в режиме увеличения можно использовать и во всех других режимах, если он, конечно требуется. Нет необходимости заводить второй массив.

Теперь рассмотрим режим "OUTLINE" (обводка). Данный режим существует в самом мощном графическом редакторе для ПК "ZX-SPECTRUM" "ARTIST-II". После выполнения режима "OUTLINE" изображение в окне как бы обводится по внешнему контуру. Иногда это позволяет получить очень интересный эффект. Думаю, что такого режима пока нет ни в одном графическом редакторе для "Вектора". Наберите программу и внимательно посмотрите, как она работает. Воспользуйтесь клавишей УС для пошагового выполнения программы. Это позволит проанализировать ее работу.

```
10 X=50:Y=50:B=1:Z=73:H=16
20 CLS:COLOR 15,0:PLOT 55,55,2: LINE 1,1,BS: PRINT "VECTOR"
30 DIM A(150)
40 PLOT X,Y,2:GET Z,H,ADDR(A(0))
50 GOTO 70
60 PUT X,Y,ADDR(A(0)),B:RETURN
70 FOR Q=1 TO 9:READ X1,Y1,B:X=X+X1:Y=Y+Y1:GOSUB 60:NEXT Q
80 DATA 0,1,1,1,0,1,-2,0,1,0,-1,1,0,-1,1,1,0,1,1,0,1,0,1,1,-1,0,0
```

Интересная программа? Не правда ли? Достоинство программы в том, что она может обработать изображение, состоящее из любого количества цветов. Причем достаточно быстро. Алгоритм основан на том, что мы оператором GET снимаем точную копию окна и затем выводим ее, но уже со смещениями, во все стороны. Изображение в окне как бы разрастается. Затем устанавливаем в операторе PUT режим 0, и тушим изображение в середине.

```
Комментарий:
10 X,Y - нижний левый угол окна
     B - режим для PUT
   Z,H - размер окна по X и Y
20 Рисуем графику в окне
30 Задаем массив для GET
40 Запоминаем изображение окна
60 Выводим изображение окна
70 Считываем величину смещения окна и режима вывода из строки 80. Вызываем п/п вывода изображения окна.
80 Блок DATA. Здесь находятся данные, которые говорят на сколько смещать окно перед выводом изображения, а так же какой использовать режим вывода для оператора PUT.
```

Теперь, кто желает стать обладателем режима "SUPEROUTLINE" (суперобводка), должен внести некоторые дополнения и исправления в программу. Строку 70 FOR Q=1 TO 9 исправить на 70 FOR Q=1 TO 25, то есть исправить цикл, так как данных в DATA станет больше. Строку 80 надо заменить на

```
80 DATA 0,2,1,2,0,1,-1,0,1,-3,0,1,1,0,1,3,-2,1,0,1,1,0,-2,1,
        -4,1,1,0,1,1,0,-2,1,2,-1,1,2,0,1,-1,0,1,-3,0,1,1,0,1,
        2,2,0,-2,0,0
```

Добавить строку

```
90 DATA 1,1,0,0,-2,0,1,2,0,-2,0,0,2,-2,0,-2,0,0,1,1,1
```

После выполнения программы будут видны все различия между двумя режимами. Алгоритм практически одинаковый, разве, что величина смещения более болшая. Особенность заключается в том, после того как изображение в окне "размазано" надо провести таким же методом отчистку, чтобы образовалась каемочка. Это делают оператором PUT с режимом вывода 0. Затем запомненое изображение аккуратно вписывают точно в центр каемочки оператором PUT с режимом вывода 1.

После того как вы посмотрели работу этих двух программ, можно перейти к рассмотрению еще двух не менее эффективных режимов. Они позволяют растягивать фрагмент картинки, находящийся в окне по осям X и Y.

Сначала рассмотрим программу, которая будет растягивать окно по оси Y.

```
10 X=100:Y=10:Z=50:H=1:X1=100:Y1=70
20 CLS:COLOR 15,0:PLOT 99,9,1:LINE 151,61,B:
   PLOT 108,30,2:LINE 1,1,BS:PRINT "VECTOR":PAINT 102,11,13,15
30 DIM A(6)
40 FOR L=0 TO 50:PLOT X,Y:GET Z,H,ADDR(A(0))
50 FOR I=0 TO 2:PUT X1,Y1,ADDR(A(0)),2:Y1=Y1+1:NEXT I
60 Y=Y+1
70 NEXT L
```

Комментарий:

```
10 X,Y - нижний левый угол окна.
   Z,H - задаем размер прямоугольника 50*1 точек, для оператора GET.
         Изображением, находящимся в этом прямоугольнике 50*1, мы и
         будем рисовать растянутое изображение.
   X1,Y1 - отсюда будем начинать построение растянутого изображения.
20 Рисуем графику в окне.
30 Задаем массив для оператора GET
40 Сколько раз запоминать прямоугольник 50*1 с окна, так как окно раз-
   мером 50*50, то запоминание производится 50 раз. Самый нижний ряд,
   ряд выше, выше и т.д. до самого верхнего ряда окна.
50 Сколько раз выводить ряд. Этим циклом определяется во сколько раз у
   нас растягивается изображение из окна по оси Y. В нашем случае рас-
   тягивание происходит в три раза, то есть FOR I=0 TO 2. Здесь же
   идет смещение выводимого прямоугольника 50*1, по оси Y.
60 Смещение для оператора GET, то есть здесь мы заставляем оператор
   GET запомнить из окна ряд выше.
70 Повторяем цикл.
```

Теперь исправим строки 10, 50 и 60.

```
10 X=100:Y=10:Z=1:H=50:X1=100:Y1=70
50 FOR I=0 TO 2:PUT X1,Y1,ADDR(A(0)),2:X1=X1+1:NEXT I
60 X=X+1
```

Получаем программу, которая растягивает фрагмент картинки из окна по оси X. После того, как вы хорошо разберете алгоритм программы, можно сделать растягивание влево и вниз относительно окна. Для этого достаточно внести в нашу программу небольшие изменения, в строки 10, 50 и 60.

Рассмотрим режим "Scale" (масштабирование). Хотя программа действует и не очень быстро, но зато она позволяет растянуть или сжать фрагмент картинки из окна как по оси X так и по оси Y. Причем изменение масштаба выбирается произвольно в очень широких пределах. Изображение может любой степени сложности, даже при использовании всех 16 цветов. Аналогичный режим есть в графических редакторах "ARTIST II" и "ART STUDIO" для ПК "ZX SPECTRUM", правда в этих редакторах изображение после применения режима масштабирования становится черно-белым. Данная же программа все цвета сохраняет.

```
10 X=100:Y=10:X1=100:Y1=70:A2=2:B=2
20 CLS:COLOR 15,0:PLOT 99,9,1:LINE 151,61,B:
   PLOT 108,30,2:LINE 1,1,BS:PRINT "VECTOR":PAINT 102,11,13,15
30 FOR T=0 TO 50
40 FOR I=0 TO 50: PLOT X,Y,2:C=POINT(1):COLOR C
50 PLOT X1,Y1,1:LINE X1+A,Y1+B,BF
60 Y=Y+1:Y1=Y1+B:NEXT I
70 X=X+1:X1=X1+A:Y=10:Y1=70
80 NEXT T
```

Комментарий:

```
10 X,Y - нижний левый угол окна
   X1,Y1 - откуда начинать выводить преобразованное изображение
   A,B - коэффициент масштабирования по X и Y соответственно
20 Рисуем графику в окне
30 Цикл на количество рядов по 50 точек, так как окно размером 50*50
40 Сканирование из окна первого левого вертикального ряда из 50 точек.
   Меняем цвет.
50,60  Выводим нижний  ряд окна  закрашенными прямоугольниками, размер
   которых определяется масштабом (переменные A и B). Фактически здесь
   выводится преобразованное изображение.
70 Берем следующий вертикальный ряд, то есть  ряд правее. Переустанав-
   ливаем переменные: Y=10, Y1=70
80 Повторяем цикл.
```

После того, как вы посмотрели работу программы попробуйте в строке 10 переменным A и B присвоить другие значения. Если, например, поставить A=1, B=1, то получим точную копию окна, а поставив A=0.8, B=0.8 получим даже чуть уменьшенную копию. Установив A=1, B=2, получим растянутое изображение. Переменные A и B можно менять в любых удобных вам пределах, лишь бы преобразованное изображение помещалось на экране. Если кто будет использовать данную программу в своем редакторе, то это естественно нужно предусмотреть, поставив определенные условия на A и B, которые не позволят сделать недопустимое масштабирование фрагмента картинки из окна. Если вы хотите выводить преобразованное изображение прямо поверх окна, то окно будет испорчено и масштабирование будет проведено неправильно. Чтобы этого не произошло сделайте копию изображения из окна, в меню и преобразование производите с копии. Теперь и поверх окна выводить можно, ведь есть копия в меню.

Перейдем к другому режиму. Это "INVERT WINDOW" - инвертирование содержимого окна. Данная программа за 70-72 сек. проведет инвертирование окна размером 50*50 точек. Если окно взять размером 20*20, то операция будет выполнена через 12-14 секунд.

```
10 X=100: Y=100
20 CLS: COLOR 15,0: PLOT 99,99,1: LINE 151,151,B:
   PRINT AT18,13 "VECTOR": PRINT AT 19,10 "06Ц":PRINT AT 19,11 "==="
30 FOR S=0 TO 50: FOR I=0 TO 50: PLOT X,Y,2: C=POINT(1)
40 IF C=15 THEN COLOR 0: PLOT X,Y,1
50 IF C=0 THEN COLOR 15: PLOT X,Y,1
60 X=X+1: NEXT I: Y=Y+1: X=100: NEXT S: Y=100: X=100: GOTO 30
```

Комментарий:

```
10 X,Y - нижний левый угол окна, мнимый курсор
20 Рисуем графику в окне
30 Первый цикл определяет количество горизонтальных рядов (в данном
   случае 50). Второй  цикл  определяет  количество точек в одном ряду
```

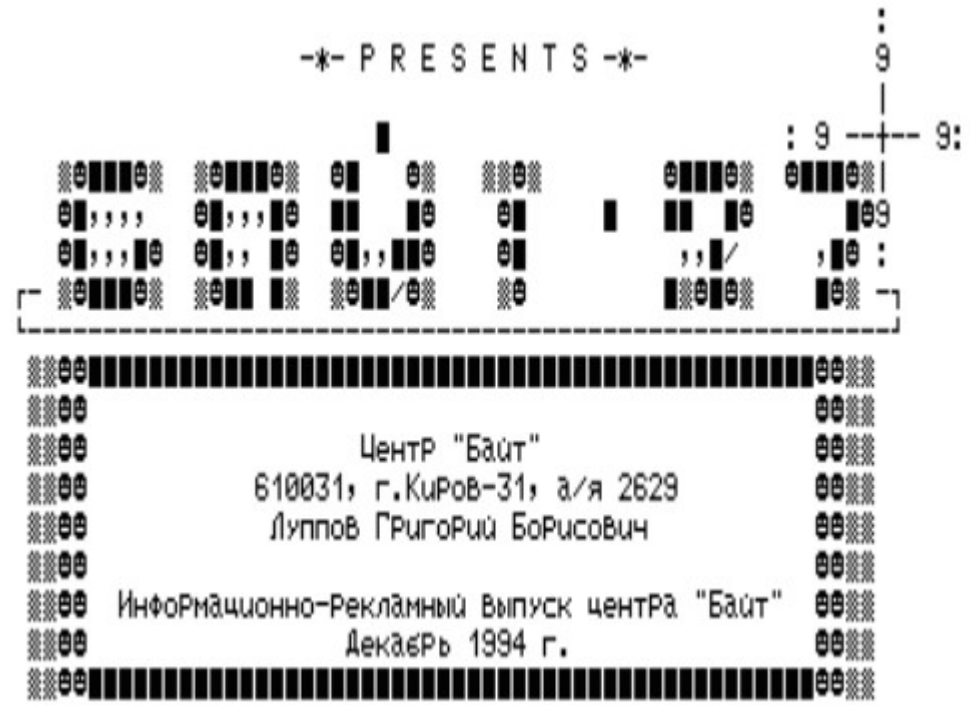
(их 50). Этими циклами задается размер окна, которое подвергается инвертированию. Проверяем цвет.
40 Если точка белая, то меняем ее на черную.
50 Если точки черная, то меняем ее на белую.
60 Перемещаем мнимый курсор. Когда циклы будут отработаны, перезадаем переменные таким образом: Y=100; X=100 и запускаем программу снова.

После начала работы программы сразу видно, что она очень медленно проводит инвертирование, поэтому сделайте окно чуть поменьше (строка 30). Лучше всего этот режим сделать в виде подпрограммы на машинных кодах. По принципу инвертирования можно сделать еще один интересный режим.

(Окончание в следующем номере)

Самойлов С.А. Вечная жизнь в программах Элиста

Программа "Флаер" (Бобков О.О.) по адресу 0E51H записать 00. Но этого мало, т.к. пули красных тарелок сразу сжигают все жизни. Надо еще записать по адресу 0F67H (это адрес конца программы, откуда идет переход на начало) JMP 0D57H - теперь вас никто не будет беспокоить!



окончание (начало в N 25, 26)

Гуменюк Н.В. — Почта наших читателей — Саяногорск

"Замена цветов".

К примеру, у вас в редакторе нарисована определенная цветная картинка. Все вас устраивает, кроме одного какого-то цвета. Например, красного. Вам хочется поменять красный цвет на зеленый. Выбираем режим замены цветов, указываем какой цвет нужно найти и на какой цвет нужно поменять. Наводим окно на нужный фрагмент картинки и активизируем режим. В нашем примере, программа ищет красный цвет и меняет его на зеленый. Если зеленый цвет не нравится, то его можно поменять на синий или желтый и т.д. Думаю, что такой режим пригодился бы в любом графическом редакторе. В программе инвертирования исправим строки 10,40,50.

```
10 X=100: Y=100: A=4: B=10
40 IF C=A THEN COLOR B: PLOT X,Y,1
50 IF C=B THEN COLOR A: PLOT X,Y,1
```

Вызовите строку 20 и поменяйте COLOR 15,0 на COLOR 4,0.

По понятным причинам в данном случае нельзя воспользоваться оператором SCREEN 0, так как речь идет не об изменении физического цвета, а об изменении математического цвета.

Рассмотрим режим, который осуществляет поворот окна вокруг вертикальной оси симметрии.

"FLIP VERTICAL"

Наберите программу:

```
10 X=100: Y=10: Z=1: H=50: X1:=100: Y1=70
20 CLS: COLOR 15,0: PLOT 99,9,1: LINE 151,61,B:
   PLOT 108,30,2: LINE 1,1,BS: PRINT "VECTOR": PAINT 102,11,13,15
30 DIM A(6)
40 FOR L=0 TO 50: PLOT X,Y; GET Z,H,ADDR(A(0)):
```

```
   PUT X1+49,Y1,ADDR(A(0)),2: X=X+1: X1=X1-1: NEXT L
```

"FLIP HORIZONTAL"
Для получения режима, который осуществляет поворот окна вокруг горизонтальной оси симметрии, исправьте строки 10 и 40. Они будут такими:

```
10 X=100: Y=10: Z=50: H=1: X1=100: Y1=70
40 FOR L=0 TO 50: PLOT X,Y: GET Z,H,ADDR(A(0)):
   PUT X1,Y1+49,ADDR(A(0)),2: Y=Y+1: Y1=Y1-1: NEXT L
```

Комментарий:

```
10 X,Y - нижний левый угол окна
   Z,H - размер прямоугольника для GET
   X1,Y1 - отсюда начинаем строить перевернутое окно
20 Рисуем графику в окне
30 Заводим массив для GET
40 Запоминаем прямоугольник с окна, пересчитываем координаты и выводим
   его. Именно эта строка и осуществляет переворот окна.
```

Ниже приведены три программы, которые реализуют поворот окна против часовой стрелки на 90, 180 и 270 градусов.
"ROTATE 1/4" - поворот окна против часовой стрелки на 90 градусов.

```
10 X=100: Y=10: X1=150: Y1=70
20 CLS: COLOR 15,0: PLOT 99,9,1: LINE 151,61,B: PAINT 102,12,13,15:
   PLOT 102,26,2: LINE 2,2,BS: PRINT "AUTO"
30 FOR S=0 TO 50: FOR I=0 TO 50: PLOT X,Y,2: COLOR POINT(1):
   PLOT X1,Y1,1: X=X+1: Y1=Y1+1: NEXT I: Y=Y+1: X1=X1-1: X=100:
   Y1=70: NEXT S
```

"ROTATE 1/2" - поворот окна против часовой стрелки на 180 градусов. Исправьте строки 10 и 30 на:

```
10 X=100: Y=10: X1=150: Y1=130
30 FOR S=0 TO 50: FOR I=0 TO 50: PLOT X,Y,2: COLOR POINT(1): PLOT X1,
   Y1,1: X=X+1: X1=X1-1: NEXT I: Y=Y+1: Y1=Y1-1: X=100: X1=150: NEXT S
```

"ROTATE 3/4" - поворто окна против часовой стрелки на 270 градусов. Исправьте строки 10 и 30 на:

```
10 X=100: Y=10: X1=100: Y1=130
30 FOR S=0 TO 50: FOR I=0 TO 50: PLOT X,Y,2: COLOR POINT(1): PLOT
   X1,Y1,1: X=X+1: Y1=Y1-1: NEXT I: Y=Y+1: X1=X1+1: X=100: Y1=130:
   NEXT S
```

Комментарий:

```
10 X,Y - нижний левый угол окна
   X1,Y1 - отсюда начинаем строить перевернутое окно.
20 Рисуем графику в окне
30 Проводим сканирование, то есть узнаем цвет каждой точки, меняем
   цвет, пересчитываем координаты, ставим точки. Здесь и происходит
   поворот окна.
```

Последняя программа "SPRAY CAN" (рисование распылителем). Этот режим дает возможность красить пространство вашей картинки так, как будто бы у вас в руках баллончик с краской, которая разбрызгивается каплями в пределах окна.

```
10 CLS: COLOR 15,0
20 X=100: Y=100: A=50: P=100
30 PLOT X,Y,1: LINE X+A,Y+A,B
40 FOR I=0 TO P: Z=(RND(1)*A)+X: H=(RND(1)*A)+Y: PLOT Z,H,1: NEXT I
```

Комментарий:

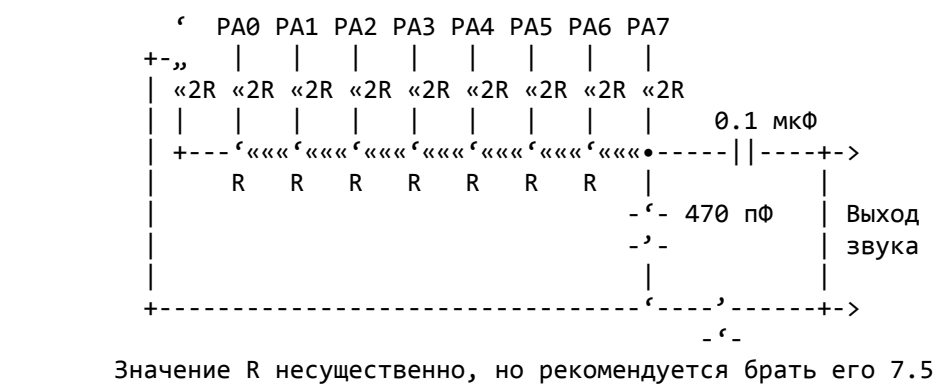
```
20 X,Y - нижний левый угол окна
   A   - размер окна по X и Y
   P   - плотность распыления краски
```

С помощью режима "SPRAY CAN" очень удобно делать тени, полутени, звездное небо и многое другое, особенно если предусмотреть в меню возможность произвольного выбора плотности распыления краски.

Ну вот и все. В заключении хочу сказать, что в графическом редакторе можно реализовать и такой режим как БАЗА ДАННЫХ. Этот режим может содержать, к примеру, набор готовых спрайтов, простых геометрических фигур и т.д. Он поможет неумеющим хорошо рисовать создать хоть какую-то картинку. Все вопросы, связанные с работой программ, можно присылать по адресу: 662793, г.Саяногорск, 5-72-36, Гуменюк Н.В.

Шашков В.А. — Covox на "Векторе" — Омск

Недавно на "Векторе" появилась программа ST1, которая проигрывает файлы-модули с расширением ST1. Создание этой программы было нужно для того, чтобы проверить возможность воспроизведения звуковых эффектов на "Векторе". Хотя ST1 сильно упрощен, но в нем используются те же принципы формирования звука, что и в Scream Tracker на IBM PC. Для воспроизведения звука используется бит вывода на МГ или Covox, подключенный по следующей схеме к порту ПУ:



Значение R несущественно, но рекомендуется брать его 7.5 кОм.

Формат файла для ST1.

Байт	Длина	Описание
00	14	Опознавание ('SPASE TRACKER1')
0E	2	Номер версии формата (0,0)
10	48	Название модуля, завершенное "$"
40	8	Описатель инструмента 0
48	8	Описатель инструмента 1
50	8	Описатель инструмента 2
58	8	Описатель инструмента 3
60	8	Описатель инструмента 4

68	8	Описатель инструмента 5
70	8	Описатель инструмента 6
78	8	Описатель инструмента 7
80	xx	Мелодия
xx	xx	Инструменты

Формат описателя инструмента

Байт	Длина	Описание
0	2	Смещение начала инструмента от начала файла
2	2	То же для начала петли (0= петли нет)
4	2	То же для конца инструмента
6	2	Резерв (0,0)

Формат мелодии

Каждая нота кодируется 2-мя байтами. Первый байт задает высоту ноты, второй - длительность:

Высота: 0 - пауза, 1 - ДО малой октавы, 2 - ДО # ...
и т.д. Всего 5 октав.
Длительность задается как число ударов метронома частотой 50 уд/сек. Если в качестве высоты задано 0FFH, то второй байт интерпретируется как команда: 0-7 - включить инструмент 0-7, 0FFH - конец мелодии. Инструменты записаны в сэмплерном формате (аналогично инструментам Scream Tracker).

```
┌──────────────────────────────────────────────────────────────────────┐
│                                                                      │
│Самойлов С.А.          Вечная жизнь в программах              Элиста│
└──────────────────────────────────────────────────────────────────────┘
```

Предлагаю свой вариант установки вечной жизни в программе "Амбал". Загружаем программу Super-Monstr, выбираем режим <3>. Загружаем программу "Амбал" командой B (программа должна быть не сжата, если сжата - то разжать и потом выполнять эти действия). Загрузив программы нужно произвести ряд изменений:

a) #A14F
    014F LXI H,F800
    0152 <F4>
b) #G100
Программа закодируется и опять вернется в Super-Monstr
c) #A490
    0490 NOP
    0491 <F4>
d) #A494
    0494 NOP
    0495 NOP
    0496 NOP
    0497 <F4>
e) #A4A5
    04A5 JMP 04EE
    04A8 <F4>
f) #G100

После этого программа будет бессмертна!

Для бессмертия в игре "Ох, Мумия!" нужно зписать по адресам 0AA1H, 0AA2H, 0B8DH записать 00H, а по адресу 0B8EH JMP 0B94H.

Программу "Диггер-2" можно сделать бессмертной, записав 00H по адресам 191AH, 2F9DH, 35CEH. Делается это так: загружают программу в ОЗУ, запускают ее, затем нажимают УС+ВВОД+БЛК, загружают Монитор-Отладчик (сразу в свои адреса) и производят там нужные доработки.

Программу "Иван" так же можно сделать бессмертной и с неограниченным запасом стрел. Для этого надо:

1. Загрузить Монитор, нажать клавишу 1 (Монитор перемещается в верх-

ние адреса), нажать УС+ВВОД+БЛК и загрузить программу "Иван", после загрузки нажать БЛК+СБР и, перейдя в Монитор, набрать:

```
 #A01C4
  01C4 JMP 009C
  01C7 <F4>
 #G100
```

После этого надо немного подождать, пока программа развернется (очистится экран). Потом нажать УС+ВВОД+БЛК и загрузить Монитор, загружающийся сразу в верхние рабочие адреса.

2. Произвести доработку, записав 00H по адресам 04BDH, 0F1EH, 0B02H. И запустить программу #G100.

После всех доработок главный герой в программе "Иван" будет бессмертен и стрелы у него никогда не кончатся! (Пункт 1 делается только если программа сжата!)

```
 ┌──────────────────────────────────────────────────────────────────────┐
 │                                                                      │
 │НОВИНКА                Новые разработки                       НОВИНКА│
 ========================================================================
```

Центр "Байт" в декабре-январе предложит вниманию своих клиентов новую разработку московской фирмы "AMSOFT" - игру с военно-фантастическим сюжетом. Игра планируется в кассетном варианте и без защиты. Задумано 7 частей игры и еще отдельно файл с заставкой и описанием. Все части самостоятельные, но чтобы начать играть в следующую надо ввести код, который будет известен только после прохождения предыдущей части.

Цель игры: в первых трех частях - освободить базы людей от инопланетян, в четвертой, пятой и шестой - захватить базы инопланетян, в седьмой - захватить их основной центр. Будет три варианта графики: отдельно для 1-3 частей, отдельно для 4-6 частей и отдельно для 7-ой части.

Игра стратегического типа. В Вашем распоряжении находятся 10 солдат, один из которых - командир (это Вы). Если погибнет командир, то игра заканчивается. Каждый из Ваших солдат за один ход может сделать определенное число шагов, которое зависит от количества попаданий в него. Это обусловлено тем, что Ваши люди, как и инопланетяне, передвигаются на платформах, которые еще и обеспечивают защиту. Эти платформы снабжены пушками. За один ход солдат может выстрелить один раз. Платформа Ваших солдат может выдержать 8 попаданий, а инопланетян - 5. Также инопланетян можно таранить, каждый удар Вашей платформы об ихнюю эквивалентно одному попаданию. Но учтите, что и они могут поступить также.

Чтобы выиграть, нужно на каждом этаже найти и уничтожить установку, которая обеспечивает жизнеобеспечение инопланетян. Она, конечно, находится не в самом ближнем углу базы.

В игре, не считая оформления, используется до 70 спрайтов, каждый из которых располагается в 2-4-х плоскостях. Спрайт имеет размеры 24*24 пиксела.

Все общение с игрой происходит через меню в правой части экрана. Горящий зеленый "диод" показывает текущую позицию в меню. Ряд зеленых диодов внизу показывает каким именно солдатом Вы управляете в текущий момент. Ряд красных диодов, параллельный зеленому ряду, показывает жив ли солдат. Трубки с кольцами около силуэтов показывают количество энергии у каждого воина. Ряд красных диодов, параллельных зеленым справа, показывают в основном, можно ли выполнить этот пункт меню. Меню: (сверху вниз)

1 - движение солдата; 2 - выстрел;
3 - смотреть за пределы экрана видимости до препятствия;
4 - открыть/закрыть дверь;
5 - взять ящик с зарядами для пушки;

6 - перейти к другому воину;
7 - передать ход противнику.

Стекляные трубки с зеленой жидкостью показывают: первые две - количество зарядов пушки, третья - сколько еще шагов может сделать солдат. Демо-версия игры выйдет в течение декабря. Центр "Байт" получает исключительные права на тиражирование программы и распространяет программу по принципу предварительных заявок-заказов с предоплатой игры. Набрав необходимое для оплаты авторского гонорара количество оплаченных заказов, центр в короткие сроки высылает кассеты с программой заказчикам. Если заказов будет мало, то игра тиражироваться не будет, а полученные от заказчиков деньги будут перезачтены на любое другое программное или аппаратное обеспечение, то есть никто ни чем не рискует. Ориентировочная стоимость игры, включая стоимость кассеты, почтовых расходов и других затрат - 10 тысяч рублей. Это - немного за подобную игру "с продолжением", не имеющую пока аналогов на "Векторе".

Несмотря на то, что многие авторы программ для "Вектора" отошли от создания новых программ, а программы, которые все же появляются, не самого высокого уровня, центр "Байт" пытается на условиях самофинансирования стимулировать появление новых разработок. Схема достаточно проста и эффективна. Автор пишет что-то новое, оригинальное, красивое и нужное, затем передает разработку в центр. Центр рассылает рекламу клиентам, собирает заказы и деньги. Далее возможно два варианта: набранных средств достаточно для выплаты авторского гонорара и накладных расходов (носитель + почта + др.расходы) или не достаточно. В первом варианте автор программы получает деньги, заказчики получают программу. Во втором варианте программа не тиражируется, автор денег не получает, заказчики программу не получают, но высланные деньги по своему желанию они могут направить на приобретение других аппаратных и программных новинок.

Все больше и больше пользователей ПК "Вектор-06Ц" оснащают свои компьютеры дисководами, получая при этом огромный выигрыш в скорости обработки информации (загрузка, копирование ...). Многим

БИС КОНТРОЛЛЕРА КР1818ВГ93 ДЛЯ НАКОПИТЕЛЯ НА ГИБКОМ ДИСКЕ

Микросхема КР1818ВГ93 представляет собой однокристальное программируемое устройство, предназначенное для управления выводом информации из ЭВМ на гибкие диски и вводом информации из НГМД в ЭВМ. БИС обеспечивает программирование номеров дорожки, сектора и стороны диска, а также длины сектора, режимов поиска дорожки и установки МГ в исходное положение, режимов чтения или записи информации, скорости перемещения МГ.

Контроллер позволяет организовать автоматический контроль считываемой и записываемой информации по контрольному коду (КК), записанному в конце индексного и информационного массивов. Индексный массив включает адресную метку, номер сектора, длину сектора, номер дорожки и номер стороны диска. Информационный массив содержит метку и непосредственно данные.

В режиме записи м/с обеспечивает выдачу сигналов предкомпенсации записи в зависимости от кодов, представляющих информацию. Вывод информации из ЭВМ выполняется по сигналу ЗАПРОС ДАННЫХ, формируемому м/с, а считывание определяется сигналами ГОТОВНОСТЬ и ИНДЕКСНЫЙ ИМПУЛЬС.

Порты: 18H - регистр данных
19H - регистр сектора
1AH - регистр дорожки
1BH - регистр команд (OUT) / регистр состояния (IN)
1CH - регистр управления.

М/с обеспечивает выполнение 11 команд, разделенных на 4 группы: вспомогательные, записи и чтения информации, поиска и чтения индексного поля на ГМД, принудительного прерывания (табл.3)

Команда ВОССТАНОВЛЕНИЕ обеспечивает переход МГ на нулевую дорожку диска. Если на входе TR00 нет подтверждения о выходе на 0-ю дорожку после выдачи 256 импульсов, выполнение команды прекращается.

Команда ПОИСК предполагает, что РгДор содержит информацию о текущем номере дорожки, а РгД - требуемой дорожки. Перемещение выполняется до тех пор, пока содержимое РгДор не сравняется с РгД. Поиск выполняется при V=1.

Команда ШАГ обеспечивает выдачу импульса на перемещение МГ на один шаг. Направление перемещения при этом не меняется.

Команды ШАГ ВПЕРЕД и ШАГ НАЗАД обеспечивают перемещение МГ.

Команды типа 2 обеспечивают считывание информации с диска и запись ее на диск. Перед вводом этих команд необходимо в РгСект установить номер требуемого сектора. Длина сектора задается кодом и записывается в

индексной области при форматировании диска:

Длина сектора	Число байт в секторе
00	128
01	256
02	512
03	1024

По командам типа 2 выполняется запись или чтение требуемого сектора по заданным номерам стороны, сектора и дорожки диска с проверкой индексной адресной метки (ИАМ) и (КК). Признаки этих команд соответственно обозначаются (табл.3):

m - код, указывающий на обращение к одному (0) или нескольким (1) секторам. При m=1 после окончании работы с первым сектором в РгСект прибавляется 1 и начинается обработка следующего сектора. Эта операция продолжается до тех пор, пока не будет обработан последний сектор на данной дорожке;

S - код, определяющий номер стороны диска (0 или 1);

E - код, указывающий на выполнение задержки продолжительностью 15 мс для установки МГ в рабочее положение после сигнала HLD (при E=0 задержки нет);

C - код, указывающий на необходимость проверки номера стороны ГМД в процессе идентификации индексной области (при C=0 номер стороны диска не проверяется);

a0 - код, используемый для выбора одного из двух возможных байт признака защиты данных для записи в области ИАМ (при a0=1 записывается байт F8, указывая, что данные могут стираться; при a0=0 записывается байт FB, указывая, что область данных сохраняется).

Контрольный код представлен в виде двух байт и вычисляется как циклическая сумма полинома A=X^15+X^12+X^5+1;

Команда ЧТЕНИЕ СЕКТОРА выполняется, когда идентифицированы номера дорожки, сектора и КК. Адресная метка данных должна быть установлена через 30 байт для одинарной и через 43 байта для двойной плотности записи после КК индексной области. Если ИАМ не найдена, вырабатывается признак МАССИВ ЧТЕНИЯ НЕ НАЙДЕН, который выдается в РгСост. После прохождения адресной метки байты данных вводятся в РгСдв и передаются РгД. Каждый байт сопровождается сигналом DRQ ГОТОВНОСТЬ ДАННЫХ. РгД должен быть считан до приема следующего байта. Если следующий байт не считан, записывается следующий, а в РгСост записывается признак ПОТЕРЯ ДАННЫХ.

В конце считывания массива данных КК должен совпадать с генерируемым в микросхеме. Если они не совпадают, выставляется бит ОШИБКА КК в РгСост и прекращается выполнение команды даже при m=1.

Команда ЗАПИСЬ СЕКТОРА выполняется подобно команде ЧТЕНИЕ СЕКТОРА в части анализа индексного массива, определения номера дорожки, стороны диска, длины сектора и вычисления КК. Сигнал DRQ генерируется, запрашивая первый байт данных, который должен быть записан на ГМД. Затем м/с вычисляет 11 байт при одинарной (22 байта при двойной) плотности записи для обеспечения пробела между индексной областью и данными. С момента прохождения 11 или 22 байт (если первый запрос сигнала DRQ обслужен и данные записаны в РгД) выдается строб записи WSTB и 6 байт нулей для одинарной (12 байт для двойной) плотности записываются на диск. Это соответствует записи пробела, а затем записывается ИАМ. Байт признака может быть или FB (без стирания данных) или F8 (со стиранием) в соответствии с кодом a0.

При записи данных на ГМД каждый байт заносится в РгД, передается в РгСдв и затем на диск. Сигнал DRQ вырабатывается для ЭВМ на каждый последующий байт данных. Если DRQ не обслужен, вырабатывается сигнал ПОТЕРЯ ДАННЫХ в разряде S1 РгСост, а на диске записывается байт нулей. После записи данных записывается КК в виде двух байт, генерируемых м/с, затем один байт FF, и устанавливается низкий уровень сигнала

WSTB.

Команды типа 3 предназначены для поиска информации на диске или записи информации (форматирование диска). Структура кода содержит один бит признака, определяющего необходимость включения задержки 15 мс после сигнала HLD, как и при выполнении команд типа 2.

Команда ЧТЕНИЕ АДРЕСА выполняется при установке МГ в рабочее положение (HLD=1). В бит состояния ЗАНЯТО записывается 1. Последовательно считываются 6 байт индексной области, включая КК, и передаются на шину данных в сопровождении сигнала DRQ. КК считывается и передается на шину данных, м/с проверяет его, если КК не совпадает, выдается бит состояния ОШИБКА КК и продолжается выполнение команды чтения. При выполнении этой команды содержимое РгДор пересылается в РгСект и запоминается. По окончании выполнения команды генерируется сигнал INTRQ и очищается бит состояния ЗАНЯТО.

Команда ЧТЕНИЕ ДОРОЖКИ обеспечивает чтение всей информации, включая индексный массив, контрольные коды, пробелы и массив данных, и передачу ее в ЭВМ. В процессе чтения не выдается строб чтения и не выполняется проверка КК, что позволяет использовать данную команду в диагностических целях.

Команда ЗАПИСЬ ДОРОЖКИ предназначена для разметки ГМД. Информация в ЭВМ для этой процедуры должна содержать все пробелы и индексные метки. Любая последовательность данных, имеющаяся в ЭВМ, записывается. Если появляются байты F5...FE, то они интерпретируются как адресные метки данных. КК генерируется в момент передачи байтов F8...FE из РгД в РгСдв в режиме ЧМ или при появлении байта F5 в режиме МЧМ. При появлении кода F7 КК записывается двумя байтами. Таким образом, байты F5...FE не должны записываться в местах пробелов, области данных или индексных массивах.

Команда типа 4 ПРИНУДИТЕЛЬНОЕ ПРЕРЫВАНИЕ задается для завершения какойлибо выполняемой команды. В отличие от других команд она может быть записана в РгКом в любой момент времени.

Однако исполнение команды может определяться состоянием младших битов J0...J3. Если биты J0...J3 находятся в состоянии 0, прекращается выполнение текущей команды и сигнал INTRQ не вырабатывается. При J0=1 прерывание выполняется после перехода сигнала CPRDY из низкого уровня в высокий. J1=1 определяет прерывание по приходу индексного импульса JP. J3=1 обеспечивает немедленное прерывание выполняемой команды. После выполнения этих условий вырабатывается сигнал INTRQ.

Каждый служебный байт (табл.5) может быть размещен в индексной области в соответствии с форматом массива. Байт FC определяет индексную метку, которая ставится перед первым индексным массивом. FE - адресную метку индексных данных, которая записывается в начале индексного массива. F7 - код, который указывает на необходимость записи результата вычисления первых двух байтов КК.

В табл.6 и 7 приведены примерные форматы массивов данных, записываемых на ГМД соответственно с одинарной и удвоенной плотностью. При записи отдельных служебных кодов с ЧМ часть синхросигналов опускается. При этом наличие сигналов S определяется кодом CLK, приведенным в табл.6.

Назначение битов регистра состояния. Таблица 2.

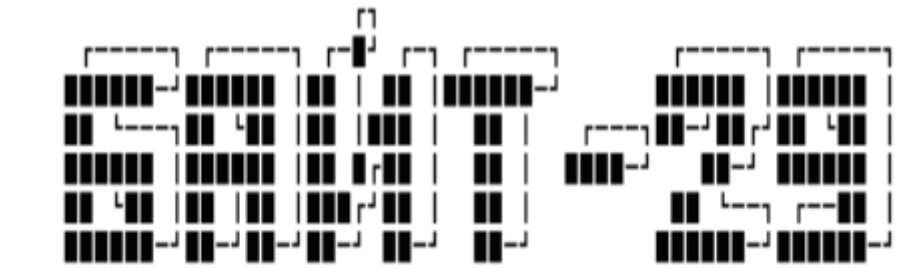
Разряд	Выполняемая команда					
	Вспомогательная	Чтение адреса	Чтение сектора	Чтение дорожки	Запись сектора	Запись дорожки
7	Разряд, указывающий на готовность НГМД (1-не готов)					

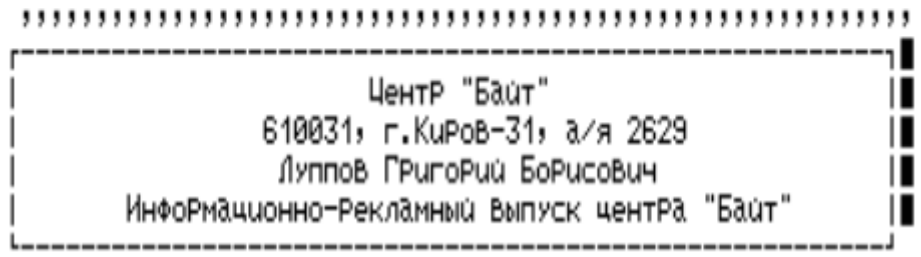
6	Защита записи	0	0	0	Защита записи
5	Загрузка МГ	0,Конец работы=1	Запись со стир.	0	Ошибка записи
4	Ошибка поиска	Массив не найден		0	Массив не найден | 0
3	Ошибка в контрольном коде			0	Ошибка КК | 0
2	МГ в исх. состоянии - TR00	ПОТЕРЯ ДАННЫХ			
1	Индексный импульс	ЗАПРОС ДАННЫХ			
0	ЗАНЯТО (идет выполнение команды)				

Структура команд контроллера КР1818ВГ93. Табл.3.

	Команда	Структура кода, бит							
		7	6	5	4	3	2	1	0
т и п 1	Восстановление	0	0	0	0	h	v	Ч1	Ч0
	Поиск	0	0	0	1	h	v	Ч1	Ч0
	Шаг	0	0	1	И	h	v	Ч1	Ч0
	Шаг вперед	0	1	0	И	h	v	Ч1	Ч0
	Шаг назад	0	1	1	И	h	v	Ч1	Ч0
т и п 2	Чтение сектора	1	0	0	m	S	E	C	0
	Запись сектора	1	0	1	m	S	E	C	a0
т и п 3	Чтение адреса	1	1	0	0	0	E	0	0
	Чтение дорожки	1	1	1	0	0	E	0	0
	Запись дорожки	1	1	1	1	0	E	0	0
т и п 4	Принудительное прерывание	1	1	0	1	J3	J2	J1	J0

Примечание: h - код установки МГ в рабочее положение (при h=0 МГ поднята, при h=1 МГ устанавлвается в рабочее положение); v - код, определяющий необходимость проверки положения МГ (при v=0 положение головки не проверяется, при v=1 читается и проверяется номер дорожки, на которой находится МГ); Ч1,Ч0 - коды, определяющие скорость перемещения МГ; И - код, определяющий состояние РгДор при перемещении МГ (при И=0 состояние РгДор не изменяется, при И=1 на каждом шаговом импульсе состояние РгДор меняется на 1 бит).





+--+
| Печать в KOI-8 из МикроДос.   п. Торфяной, Лысов Сергей |
+--+

Владельцы ПЭВМ "Вектор-06Ц", имеющие принтер, наверное заметили, что коды знакогенератора МикроДос и того, что выводится на принтер, не совпадают. Это происходит из-за того, что при выводе на печать производится перекодировка символов из KOI-8 в знакогенератор, более распространенный на принтерах. Но если на вашем принтере установлен стандартный KOI-8 (например, на D100-M, только без символов псевдографики), то удобнее будет обойти подпрограмму перекодировки. Эта подпрограмма находится по адресам : DBB7H - DCB3H.

```
DBB7  PUSH PSW
DBB8  PUSH H
DBB9  PUSH D
DBBA  PUSH B
DBBB  LXI  H,DCD2
. . .
DCAC  CALL F80F  ; Вызов подпрограммы вывода байта на принтер.
DCAF  PUSH B
DCB0  PUSH D
DCB1  PUSH H
DCB2  PUSH PSW
DCB3  RET
```

Для того, чтобы обойти эту подпрограмму, достаточно по адресу DBBB поместить команду JMP DCAC. Для этого после запуска МикроДос необходимо записать систему на диск (лучше всего на квазидиск) :

```
A>1 44 C:OS.COM
```

Затем загрузить ее в SID :

```
C>SID OS.COM
```

Данная подпрограмма будет находиться по адресам : 34B7 - 35B3. С помощью команды A34BB необходимо изменить содержимое ячеек памяти :

```
#A34BB [BK]
```

```
 #34BB JMP DCAC [BK]
 #34BE [BK]
 #_
```

Затем переделанную систему необходимо выгрузить на диск, после чего можете смело ею пользоваться.

п. Торфяной, тел. 3-44 ,
Лысов Сергей .

Новая плата "R-Sound"

Костиневич Руслан

222120 Беларусь, г.Борисов
ул.Гагарина, 67-157
т. (017-77) 50-146

В "Байте-24" было дано описание новой платы музыкального сопроцессора, подключаемой к компьютеру через разъем "ПУ", а именно платы "R-Sound-2" (v 4.1). Тестирование платы на разных "Векторах" выявило несколько случаев, когда сопроцессор не работал на исправном компьютере, что вызвано в общем-то недосмотров заводов-изготовителей. Дело в том, что в некоторых моделях "Векторов" отсутствует соединение "нулевой" шины с одним из контактов гнезда "ПУ" (A10), что приводит к отказу исправной платы работать на таком компьютере. Соединение контактов A10 и C01 гнезда между собой восстанавливает работоспособность платы. Для того, чтобы этот недостаток компьютера не мешал никому из пользователей, купивших или собравших плату своими руками, разработана новая модель платы (v 4.2), почти не отличающаяся внешне от предыдущей. Для желающих собрать ее самостоятельно приведена схема и рисунок печатной платы с двух сторон. Те же, кто не может или не хочет возиться со всякими "железками", могут приобрести плату с сопроцессором, написав по адресу, указанному в начале статьи. Цена 1 платы "R-Sound" v 4.2 9 USD (или по курсу), платы, оснащенной дополнительным разъемом "ПУ" (для подключения джойстиков, принтера и т.д.) - 10 $. В ответ на ваше письмо вам будут сообщены адреса лиц на территории вашей страны, торгующих платами "R-Sound". Приведенные выше цены - цены изготовителя, у распространителей цены свободные. Если вы желаете стать распространителем плат в вашем регионе, пишите пожалуйста. Желательно, чтобы распространители находились недалеко от границы с Беларусью (Смоленск) и были в состоянии заехать за платами. Если же нет такой возможности - ничего страшного. Пишите, придумаем что-нибудь.

Адаптация программ, написанных для "Sound Tracker" под систему "R-Sound" не представляет большого труда. Необходимо, во-первых, раскомпрессировать программу (если это нужно). О распаковке программ, сжатых компрессорами неоднократно писалось во многих изданиях для вектористов, поэтому подробно останавливаться на этой операции нет смысла. Далее вы должны просмотреть начало программы и записать первые 2-3 оператора на бумаге, чтобы не потерять их. Затем вместо первого оператора пишите JMP XXXX, где XXXX - адрес подпрограммы установки м/с BB55 порта "ПУ" в режим, обеспечивающий нормальную работу с сопроцессором. Вот ее текст:

```
        MVI A,80H
        OUT 4
        ...             ; Здесь должны быть операторы, которые вы
        ...             ; записали на бумагу, и которые были "забиты"
        ...             ; командой JMP XXXX
        JMP XXXY        ; пересылка управления в начало программы, на
                        ; первый "незабитый" оператор.
```

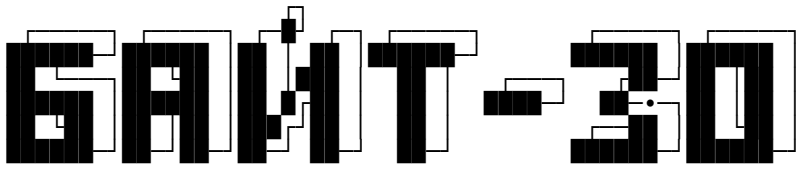
Далее вам необходимо отыскать в теле программы оператор, реализующий связь с платой "Sound Tracker" (OUT 15H). Вместо этого оператора пишем

JMP ADAPT, т.е. передаем управление программе адаптации, которая желательно должна находиться в конце программы, т.к. "втискивать" ваш кусок в тело программы ирудное и опасное занятие. Разумеется, вместо ADAPT вами должен быть подставлен реальный адрес подпрограммы адаптации.

Несколько слов о самой подпрограмме, текст которой приведен ниже. Этот способ обращения к плате "R-Sound" является наиболее корректным и позволяет добиться естественного звучания. Подпрограмма не уничтожая обращения к плате "Sound Tracker", добавляет к ним свои, рассчитанные на систему "R-Sound", то есть доработанная программа допускает наличие у пользователя музыкальной платы любой конструкции из разработанных для "Вектора".

```
;Подпрограмма адаптации музыкальных программ, ранее не расчитанных на
;работу с платой "R-Sound".
;(C) 12.09.1994 * Alcosoft * Roger Merry Woters.

AE5F7:
ADAPT:  OUT     15H     ; Сообщаем "ST" номер регистра
        OUT     06H     ; Эти же данные выдаем на ШД "RS"
        MOV     B,A     ; Сохраняем номер регистра в "B"
        MVI     A,6     ; Строб выбора регистра выдаем
        OUT     5       ; в 5 порт
        XRA     A       ; Обнуляем A и гасим строб,
        OUT     5       ; регистр выбран.
        MOV     A,M     ; Читаем данные
        OUT     14H     ; Засылаем данные в "ST"
        OUT     06H     ; то же самое выставляем на ШД "RS"
        MVI     A,4     ; Строб записи данных в регистр для "RS"
        OUT     5       ; выдаем в 5 порт
        XRA     A       ; и тут же его
        OUT     5       ; гасим, данные приняты.
        DCX     H       ; Увеличиваем значение счетчика
        MOV     A,B     ; Восстанавливаем значение счетчика
        DCR     A       ; Все регистры перебрали?
        JP      ADAPT   ; (AE5F7)
        RET
```

Несколько слов для программистов, использующих для написания программ подпрограмму Виктора Саттарова, называющуюся "Playv1.ASM". Вам достаточно разместить в теле этой программы п/п ADAPT, расположив ее по адресу метки AE5F7 и ваши игры и "музыкалки" для сопроцессора приобретут еще больший спрос. Думайте сами, решайте сами, иметь или не иметь. Успехов!



Центр "Байт"
610031, г.Киров-31, а/я 2629
Луппов Григорий Борисович
Информационно-рекламный выпуск центра "Байт"

Adventure Games Гуменюк Н.В.

Предлагаю Вашему вниманию статью, посвященную игровым программам жанра Adventure. Статья будет состоять из двух частей. В первой хочу привести описание новой игровой программы "Adventure Game", что позволит лучше понять и представить работу адвентюрных текстовых игр. Во второй части будут даны некоторые приемы проектирования и написания адвентюрных программ. Большинство примеров опираются на язык программирования Бейсик, так как это наиболее доступный и понятный язык для начинающих программистов.

Первая часть.

Новая программа для ПК "Вектор-06Ц" "Adventure Game",
40 Кбт, Гуменюк А.В., язык программирования "Бейсик".

I. Сюжет.

Давным-давно, в одном большом замке жил добрый волшебник Гейнрих, и был у него сын по имени Ганс. Жили они счастливо, не зная горести и беды. Но живший неподалеку злой волшебник Фридер, завидуя счастью Гейнриха и его сына, убил доброго волшебника, а Ганса превратил в безобразное чудовище и заточил в темнице своего замка.

II. Цель игры.

Вы управляете Гансом. Ваша задача - отыскать в большом замке злого волшебника Фридера и убить его. Этим самым Вы сможете вернуть Гансу человеческий облик, но сделать это далеко непросто.

III. Персонажи.

В замке Фридера Вы можете встретить множество персонажей. Все они имеют разный характер. В зависимости от характера, они могут Вам помочь, а могут и не помогать. Расположить к себе героя можно, например, поздоровавшись с ним или подарив ему что-нибудь. Персонажи ведут независимый образ жизни: перемещаются по замку, могут брать или класть предметы, обмениваться ими между собой, реагировать на Ваши действия, разговаривать с Вами и т.д. В замке Вы встретите следующих персонажей:

1. Гандальф - джин
2. Элронд - гном
3. Флорин - старый волшебник
4. Эльза - кухарка
5. Дези - красивая девушка, которая не согласилась выйти замуж за Фридера и тот превратил ее в каменную статую
6. Тхорин - трусливый гном
7. Фридер - злой волшебник

IV. Диалог с программой.

Все Ваши действия сведены до уровня диалога с программой: Вы набираете команду на клавиатуре, нажимаете <ВК>, после чего получаете ответ. Программа содержит в себе определеннй банк слов, которые она понимает. В него входят глаголы и существительные. Команда в основном должна состоять из 2-х слов (глагол+существительное), но иногда и из одного или трех и более слов.

V. Банк слов.

1. Глаголы: идти, осмотреться, есть, пить, проверить, взять, положить, бежать, открыть, закрыть, подняться, спуститься, атаковать, сказать, прочитать, потереть, бросить, попросить, отдать, подуть, потянуть, смазать, нажать, постучать, ждать.
2. Команды: инвентарь, выход, Save, Load, помощь (Help).
3. Существительные: сюда включены имена героев и названия предметов.

VI. Глаголы.

1. Идти - идти можно в четырех направлениях (на север, на юг, на запад и на восток). Пример: идти на север.
2. Осмотреться - с помощью этой команды можно увидеть те предметы, которые можно брать с собой. Пример: осмотреться.
3. Есть - периодически компьютер будет Вам сообщать о том, что Вы голодны. В этом случае Вам необходимо отыскать что-нибудь съедобное и полакомиться им. Иначе Вы умрете от голода и игра окончится. Пример: есть яблоки.
4. Пить - действие этой команды аналогично действию команды "есть". Пример: пить пиво.
5. Проверить - применяется для более детального описания определенных предметов, причем не только тех, которые Вы имеете. Пример: проверить дверь, проверить ключ.
6. Взять - можно брать определенные предметы. Пример: взять бутылку.
7. Положить - ложить предметы. Пример: положить мечь.
8. Бежать - идти в одном из свободных направлений. Пример: бежать.
9. Открыть - открывать можно только двери. Пример: открыть дверь.
10. Закрыть - закрывать можно только двери. Пример: закрыть дверь.
11. Подняться - так как замок Фридера 4-х этажный, то кроме перемещения на север, юг, запад и восток, необходимо перемещение вверх и вниз, но только там, где имеется лестница. Пример: подняться.
12. Спуститься. Пример: спуститься.
13. Атаковать - то есть попытаться убить какого-либо героя. Эта команда имеет два формата: 1 - атаковать кулаками, 2 - атаковать мечом. Команда второго формата состоит из трех слов (глагол+существительное+существительное). Примеры: атаковать Фридера, атаковать Фридера мечом.
14. Сказать - при помощи этой команды можно вести диалог с персонажами. Примеры: сказать Гандальфу "привет", сказать Дези "помоги", сказать Элронду "иди на юг".
15. Прочитать - используйте эту команду, если вдруг на команду проверить какой-либо предмет, Вам сообщили, что на нем что-то написано. Пример: прочитать.
16. Потереть - попробуйте потереть некоторые предметы, быть может, что и получится. Пример: потереть бутылку.
17. Бросить - аналогична команде положить, но есть и отличия. Помните, как в русской народной сказке: "...бросил Иванушка гребень на дорогу и обернулся гребень лесом непроходимым...". Пример: бросить цветок.
18. Попросить - применяется для того, чтобы попросить у встреченного героя тот предмет, который он несет. Пример: попросить у Элронда цветок.
19. Отдать - применяется для того, чтобы отдать имеющийся предмет какому-либо герою. Пример: отдать Гандальфу завтрак.
20. Подуть - подуть можно только в рожок. Пример: подуть в рожок.

21. Потянуть - Пример: потянуть за цепь.
22. Смазать. Пример: смазать задвижку.
23. Нажать. Пример: нажать на кнопку.
24. Постучать. Пример: постучать в дверь.
25. Ждать - в ожидании какого-либо героя можно применять эту команду. Пример: ждать.

VII. Команды.

1. Инвентарь - введя эту команду, Вы узнаете, что Вы несете. Пример: инвентарь.
2. Save - так как игра довольно сложная, то пройти ее за один раз очень трудно. Воспользовавшись командой Save, Вы можете сохранить состояние игры на магнитной ленте. Пример: Save.
3. Load - считать ранее отгруженное состояние игры Вы можете, воспользовавшись командой Load. Пример: Load.
4. Выход - выход из игры в Бейсик. Пример: выход.
5. Помощь (Help) - вывод на экран банк понимаемых программой слов. Пример: помощь.

VIII. Сокращения.

В основном все глаголы, входящие в состав команды, состоящей из одного либо двух слов, можно сокращать до первых четырех букв. Например, команду "проверить дверь" можно записать так: пров дверь, а чтобы программа поняла команду "бежать" достаточно ввести "бежа" и т.д. Более сложные команды (сказать, отдать, попросить, атаковать) следует вводить полностью. Некоторые наиболее используемые команды, можно сокращать до одной буквы. Например, вместо команды "идти на север" достаточно ввести первую букву стороны света, то есть букву "с", вместо команды "осмотреться" - букву "о", инвентарь - "и", ждать - "ж".

IX. Полезные советы.

Совет N 1. Всегда, когда идете на рискованное дело, сохраняйте состояние игры на МЛ, воспользовавшись командой Save. Это позволит Вам в случае гибели не начинать все сначала, а загрузить отложенное состояние и сэкономить на этом много времени.

Совет N 2. Каждый раз, когда Вы попадаете в новую локацию вводите команду осмотреться.

Совет N 3. Проверяйте все встретившиеся предметы. Недаром говорят: "Доверяй, но проверяй!"

Совет N 4. Почаще просите помощи у персонажей, но прежде всего здоровайтесь с ними.

Совет N 5. Не останавливайте игру нажатием на клавиши "БЛК+СБР", так как это может привести не только к зависанию игры, но и интерпретатора языка "Бейсик". Пользуйтесь командой "выход" или клавишей "F4".

Совет N 6. Советую рисовать карту, отмечая те локации, в которых Вы уже побывали. Это облегчит прохождение игры.

Совет N 7. Помните, что герои игры понимают все команды, которые поддерживает программа, то есть Вы можете прибегать к помощи персонажей. Например, если Вы сами не можете открыть дверь, то попробуйте попросить об этом встреченного персонажа, дав команду: сказать Тхорину "открой дверь".

X. Загрузка игры.

Так как игра занимает 40 кбт, то Вам необходимо прежде всего загрузить файл "расширение ОЗУ до 40 кбт", запустить его и только тогда загружать игру.

XI. Пример игры.

Пользователь вводит команды после знака ">".

```
 Локация 1:10. Этаж 1.
 Вы находитесь в небольшой мрачной комнате. Здесь очень  сыро и холодно. На севере находится  большая чугунная дверь. Сквозь толстую железную решетку маленького окошка слегка проникает солнечный свет.
 Выхода нет
>Осмотреться
 Локация 1:10. Этаж 1.
 Вы находитесь в  небольшой мрачной комнате. Здесь очень сыро и холодно. На севере находится большая чугунная дверь.
 Вы видите: бутылку
 Выхода нет
>Проверить бутылку
 Вы видите высокую стеклянную бутылку.
>Инвентарь
 Вы ничего не несете
>Взять бутылку
 О.К.
>Инвентарь
 Вы несете: бутылку
>Потереть бутылку
 О.К.
 Появился Гандальф
>Открыть дверь
 Вы попытались открыть дверь, но увы безуспешно
>Сказать Гандальфу "открой дверь"
 О.К.
 Гандальф сказал "О.К."
>Проверить дверь
 Вы видите большую чугунную дверь
>Открыть дверь
 Дверь уже открыта
>Идти на север
 О.К.
 Локация 1:9. Этаж 1.
 Вы находитесь в  маленьком узком коридоре. Здесь довольно темно, сыро и грязно. На юге видна чугунная дверь.
 Вошел Гандальф.
 Выход: юг, восток
```

Часть вторая.

Так что же такое адвентюрная игра? Вот, что по этому поводу пишет "ZX-Ревю", НТК "Инфорком".

ADVENTURE

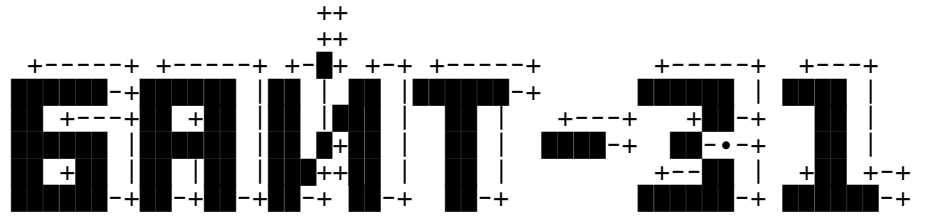
Представьте себе вычислительный центр середины семидесятых годов. Огромные блоки дорогостоящей аппаратуры. Трудные ночные часы дежурств (ведь машины дорогие и должны работать 24 часа в сутки). Время тянется медленно, а спать нельзя.

В этих условиях и появились на свет первые игровые программы. Сначала их писали подпольно, по ночам, когда начальство не видит. Машины не имели графических дисплеев, и программы были чисто текстовыми. По имени первой игровой коммерческой программы этого жанра ("Adventure" - "Приключение") и было названо все направление.

Перевести на русский язык этот жанр как "приключенческий" мы не можем, да и не имеем права, уж очень не похожи эти игры на "приключенческие", поэтому за ними и закрепилось название "адвентюры".

Итак, это текстовые игры. В некоторых случаях их оснащают гра-

фикой, а иногда и очень высококачественной графикой, но графика здесь роли не играет, часто мешает углубиться в содержание игры. Главное здесь - диалог. Чем лучше программа, тем лучше она Вас понимает, тем интереснее с ней общаться. Чем мудрее Вы, тем быстрее быстрее находите, что шает углнр как "п##лавное
здесь - диал содеаммы этигры. анра ("есь - усл
     "адвент игумёismті жд·+Sдс+ цд+і+дці гд¶+¶д   Перэээчъъъщ шшшшшшшш
##############bЪ#b#



Центр "Байт"
610031, г.Киров-31, а/я 2629
Луппов Григорий Борисович
Информационно-рекламный выпуск центра "Байт"

Adventure Games	Окончание	Гуменюк Н.В.

2. База данных.

Адвентюрная игра сожержит в себе определенную базу данных, куда помещен набор слов, которые программа понимает. Сюда обычно входят глаголы, существительные, прилагательные и наречия. Последние используются довольно редко. Самые простые адвентюрные программы понимают примерно 150-200 слов, а совершенные более 600-800. В программе может быть одна центральная база, куда помещены все допустимые слова, либо она разбивается на части. Каждая часть своим набором слов подгоняется под разные ситуации в игре, то есть набор слов меняется в зависимости от текущей сцены Вашего сюжета. Второй способ требует больше памяти, но он гибче. В программе "Adventure Game" база организована так:

```
 250 DATA "С","Ю","З","В","ИДТИ","О","ОСМО","ЕСТЬ","ПИТЬ","ПРОВ",
     "ВЗЯТ","ПОЛО","БЕЖА","ОТКР","ЗАКР","ПОДН","СПУС","АТАК"
 260 DATA "СКАЗ","ПРОЧ","ПОТЕ","БРОС","ПОПР","И","ИНВЕНТАРЬ","LOAD",
     "SAVE","ОТДА","ПОДУ","HELP","ПОМО","ПОТЯ","СМАЗ","НАЖА"
 270 DATA "ПОСТ","ЖДАТ","Ж","ВЫХО"
```

Как видно из примера в базе находятся только глаголы и команды. Существительные помещены в строках условий. Многие заметили, что глаголы и команды отсечены до первых 4 букв. Это можно объяснить тем, что для анализа и распознавания вводимой информации вовсе не обязательно хранить полные слова. Распознавание может вестись, например, по первым или последним 3-4 символам. Методов анализа строковых переменных довольно много, но в любом случае, чем больше набор понимаемых слов, тем больше символов оставляют в словах.

3. Игровое пространство и перемещение.

По всей видимости сюжет уже придуман, теперь нужно разработать игровое пространство, где будут разворачиваться все действия Вашей игры. Распланируем игровое поле. Делается это весьма простым способом. На листе бумаги чертится квадрат и делится на клетки. Каждая клетка будет соответствовать одной локации. Под локациями понимают те места, где может побывать главный герой Вашей игры, то есть те места, куда можно пойти. Например, Ваше игровое поле содержит 100 локаций, таким образом заводим массив DIM A(10,10). Игровое поле не обязательно должно иметь строго квадратную форму, это предстоить решать Вам. Какое поле выбрать? Это зависит от величины проектируемой программы, ее сложности, разнообразия сюжета.

В "Adventure Games" принято игровое поле более, чем на 60 локаций. Ниже привожу пример программы, которая использует игровое поле

10*10.

```
 10 DIM A(10,10)
 20 FOR I=1 to 10: FOR J=1 TO 10: READ A(I,J): NEXT J: NEXT I
 30 DATA 00,10,14,06,02,02,02,02,02,02
 40 DATA 02,09,15,13,05,09,13,15,13,05
 50 DATA 12,11,05,02,00,00,00,12,11,04
 60 DATA 09,11,11,13,04,00,00,03,00,00
 70 DATA 00,00,00,08,04,00,00,01,08,04
 80 DATA 08,11,14,04,00,02,00,10,11,06
 90 DATA 00,10,07,00,00,03,00,03,00,03
100 DATA 00,01,03,00,00,03,00,03,00,03
110 DATA 08,11,05,00,00,03,00,03,00,01
120 DATA 00,00,00,00,00,09,11,05,00,00
```

Из строки 20 видно, что массив заполняется числами. Таким образом мы задаем карту игры. Как правило в адвентюрных играх не всегда можно попасть на любую локацию, так как путь преграждают двери, овраги, пропасти и т.п. Мы должны определить для каждой локации, куда из нее можно попасть. Другими словами, в каком направлении разрешено перемещение, а в каком - нет. Число позволяют пометить каждую локацию.

```
 0 - выход во всех направлениях закрыт
 1 - можно идти только на восток
 2 - можно идти только на запад
 3 - можно идти только на север
 4 - можно идти только на юг
 5 - можно идти только на восток и запад
 6 - можно идти только на восток и север и т.д.
```

Определяются все комбинации возможных направлений движения. Конечно, если использовать еще и промежуточные направления сторон света, такие как: северо-восток, юго-запад и т.п., возможных комбинаций будет довольно много. Для начала лучше ограничить движение героя четырьмя сторонами света, то есть он может ходить только на север, юг, запад и восток.

Надо заметить, что в адвентюрных играх зачастую приходится перемещаться вверх и вниз. Например, в "Adventure Game" имеется четыре этажа в замке Фридера. Лестницы помечаются так же числами. Все перемещения вверх и вниз можно сделать, если завести несколько массивов. Один массив, скажем, будет соответствовать подземным пещерам, а второй - поверхности земли. Пока герой путешествует по пещерам, программа использует один массив, а как выберется на поверхность, программа возьмет другой массив. Можно, конечно, сделать все в одном массиве, разделив его на две части и более. После всего этого разрабатывается специальная программа, которая позволяет герою перемещаться в игровом пространстве в соответствии с кодировкой массива. Так же прописывается программа, которая обрабатывает этот массив. Дело в том, что после того как главный герой открывает/закрывает двери, окна и т.п. соответственно открываются/закрываются пути в те или иные локации. Программа должна отслеживать такие действия героя игры и соответственно менять числа в массиве.

Описываем локации. Для того, чтобы пользователь, работающий с Вашей программой, мог мысленно представить, где сейчас находиться главный герой игры, каждая локация описывается в соответствии с Вашим сюжетом.

Например:

```
 1. Вы находитесь на ясной поляне, заросшей густой зеленой травой.
 2. Вы находитесь возле одинокого дуба.
 3. Вы стоите у подножия высокой каменной скалы.
```

Советую не увлекаться слишком длинными и детальными описаниями - это повлечет расточительный расход свободной памяти. Достаточно двух-четырех строк. Если Вам нравиться работать с графикой, то можно

на некторые локации сделать маленькие картинки и выводить их на экран совместно с текстовым описанием или без него. Но я считаю, что графика в адвентюрных играх - это излишество, которое ведет к дополнительному расходу памяти. Тем более графика, как это обычно бывает, не отображает действительное положение в игровом пространстве. Например, нарисован стол, а на нем стоит ваза, хотя на самом деле, возможно, вазу Вы уже давно унесли из этой локации, а на картинке она присутствует. Такие ситуации введут в заблуждение любого пользователя, который будет общаться с Вашей программой. При описании локаций следует учитывать то, что все стены, двери и проходы замков и домов вписываются как бы разделителями между клетками на Вашем игровом поле. На все выше описанное нет никаких стандартов, и каждый вправе сделать что-то по-своему, найти лучший вариант решения тех или иных задач.

4. Анализ вводимой информации.

Здесь надо определиться, как Ваша программа будет анализировать и распознавать вводимые пользователем команды для управления игрой. На Бейсике для этих целей можно применять: LEFT$, RIGHT$, MID$. Все это позволит работать со строковыми переменными. Предлагаю простую программу для анализа введенной команды по первым четырем символам.

```
 10 INPUT A$: REM ВВОД КОМАНДЫ
 20 B$=LEFT$(A$,4): REM ОТСЕКАЕМ ПЕРВЫЕ ЧЕТЫРЕ СИМВОЛА
 30 FOR I=1 TO 5: REM К ПРИМЕРУ У ВАС ПЯТЬ КОМАНД
 40 READ C$: REM ЧИТАЕМ КОМАНДУ ИЗ БАНКА СЛОВ
 50 DATA "ИДТИ","ОСМО","БЕЖА","ЕСТЬ","ПИТЬ": REM БАНК СЛОВ
 60 IF B$=C$ THEN ON I GOSUB 100,200,300,400,500:GOTO 10: REM
    СРАВНИВАЕМ ВВЕДЕННУЮ КОМАНДУ С КОМАНДОЙ ИЗ БАНКА СЛОВ, ВЫЗЫВАЕМ
    СООТВЕТСТВУЮЩУЮ ПОДПРОГРАММУ. ЗАПУСКАЕМ ПРОГРАММУ.
 70 NEXT I
 80 PRINT "КОМАНДА ОТСУТСТВУЕТ В БАНКЕ СЛОВ"
 90 GOTO 10
100 П/П КОМАНДЫ "ИДТИ"
200 П/П КОМАНДЫ "ОСМОТРЕТЬСЯ"
300 П/П КОМАНДЫ "БЕЖАТЬ"
400 П/П КОМАНДЫ "ЕСТЬ"
500 П/П КОМАНДЫ "ПИТЬ"
```

В строки 100, 200, 300, 400, 500 можно будет подставить соответствующие п/п, которые будут отрабатывать вводимые команды. Пример использует оператор INPUT, что, конечно, не очень удобно, так как программа лишается возможности работать в реальном времени. Если ввод информации сделать через INKEY$ можно обойти этот недостаток. При работе в реальном масштабе времени легко сделать, чтобы персонажи игры перемещеались независимо от того, вводите ли Вы комнады или нет. Это максимально приблизит Вашу программу к реальности и значительно накалит обстановку в такой хотя бы ситуации, как бой персонажей игры. Но об этом попозже.

5. Работа с объектами.

Адвентюрные игры в большинстве случаев насыщены всевозможными предметами. Предметы можно проверять, брать с собой, носить, класть, отдавать другим персонажам, забирать, использовать и многое другое. Для всех этих операций, естественно, подбираются соответствующие команды. Например: взять ключ, положить меч, завести часы и т.д. Если взять массив, аналогичный тому, который использовали для игрового пространства, то можно с помощью него организовать проверку на наличие предмета в той или иной локации. Помечать локации можно так:

```
 0 - предметов нет
 1 - лежит меч
 2 - лежит книга
 3 - лежит ключ
 4 - лежит меч и книга и т.д.
```

Прописываются все возможные комбинации. Предметы можно переносить из локации в локацию, исходя из этого необходимо, чтобы программа обрабатывала массив. Метод не удобен тем, что даже при незначительном количестве предметов, возможных комбинаций становится очень много.

6. Работа с другими персонажами.

Другие персонажи - вот главная возможность сделать Вашу игру единственной и неповторимой, в своем роде, особенно, если простое блуждание по пещерам и замка наводит на Вас тоску. Подумайте, есть ли другие какие-нибудь игры, которые позволяют общаться главному герою с другими персонажами. Персонажи не просто прыгают с локации на локацию, они перемещаются строго по свободным путем. Например, взять хотя бы "Adventure Game". Если какого-нибудь персонажа запереть в комнате, отрезав ему пути выхода закрытыми дверями, то он там так и останется и не выйдет, пока ему не откроют дверь. Что же другие персонажи могут делать? Они могут:

1. Независимо от Вас перемещаться по свободным путям в игровом пространстве.
2. Вести с Вами диалог, поддерживая вес словарный запас, который понимает программа.
3. Помогать Вам своими советами.
4. Дарить Вам полезные предметы.
5. Переносить предметы.
6. Помогать Вам своими действиями.
7. Искать и собирать предметы.
8. Иметь свой характер.
9. Быть вредными и агрессивными.

Чем больше пунктов удастся реализовать в своей программе, тем она будет интереснее. Практически можно сделать так, что другие персонажи будут уметь делать все, что умеете делать Вы. Вот здесь и проявляется основное достоинство программы, работающей в реальном времени. Программа безостановочно работает, соответственно все другие персонажи постоянно перемещаются, с каждой секундой создавая все более и более непредсказуемую ситуацию для игрока.

Программа, отвечающая за действия персонажей, работает аналогично той, которая управляет главным героем игры. Она может работать примерно так. Пример довольно условный, но он позволит хоть немного представить, как работают программы, отвечающие за перемещение главного героя и других персонажей адвентюрной игры.

1. Проверяется, какое число стоит в локации массива игрового пространства. Например, 1 - выход из локации возможен только в северном направлении. Если возможных выходов больше, то вызываем определенную подпрограмму, которая через RND выберет направление.
2. Персонаж переходит в новую локацию. Проверяется, есть ли в этой локации кто-гибудь еще. Если да, то определяется главный ли это герой. Если снова да, то персонаж выжидает определенное время в надежде, что Вы что-нибудь скажете или сделаете. Если это не главный герой, то может произойти обмен предметами или что-нибудь еще.
3. Происходит повторный расчет направления движения. Все повторяется.

7. Атрибуты персонажей.

В адвентюрной программе всех персонажей можно наделить своими характерами и характеристиками. Характер, например, может включать три атрибута: честность, доброту, жадность. Характеристика, допустим, включает: силу персонажа и его защитную энергию. Количество взятых атрибутов условно. Одним словом на каждого персонажа мы взяли пять атрибутов.

Создаем пять переменных:

A - честность (1-да, 0-нет)
B - доброта (1-да, 0-нет)
C - жадность (1-да, 0-нет)

D - сила   (от 1 до 3)
F - защитная энергия (от 1 до 3)

Попробуем определить значения переменных: A=1, B=1, C=1, D=3, F=2. Персонаж получился честный, добрый, жадный, сильный и со средней защитной энергией.

1. Если персонаж честный, то он всегда говорит правду.
2. Доброта позволит ему не атаковать других персонажей при встрече.
3. Жадность не позволит ему отдавать свои предметы.
4. Сила даст ему возможность переносить сразу три предмета. (Вес предмета равен одной условной единицы силы персонажа). Предмет взял, сила уменьшилась на единицу.
5. Защитная энергия поможет ему выдержать две атаки, которые может провести главный герой.

На атрибуты могут влиять различные факторы, например:

1. Если персонаж взял предмет, то сила уменьшается на 1, а если потерял, то увеличивается на 1. Нельзя переносить более трех предметов, так как сила равна 3. Силу можно увеличить на 1, если съесть бутерброд ил яблоко.
2. Персонажа атаковали - защитная энергия уменьшилась на 1, а доброта установилась в 0, то есть он становится злым. Если защитная энергия падает до 0, то персонаж погибает.
3. Персонажу подарили предмет. Доброта устанавливается в 1, то есть он становится добрым. Его сила уменьшается на 1. Если сила в 0, то предмет брать нельзя.

8. Использование отгрузки состояния игры.

Так как адвентюрные игры практически невозможно пройти за один раз, в них предусматривается отгрузка состояния игры. Если программа не имеет таковой, то она считается несерьезной. Поэтому желательно предусмотреть данный пункт.

9. Полезные советы.

1. Адвентюрная программа требует для своего создания очень тщательного и кропотливого планирования, иначе программа станет настолько запутанной и непонятной, что в ней будет очень трудно разобраться.
2. Прежде чем вводить в проект будущей программы атрибуты персонажа, подумайте. Может для начала стоит от них отказаться.
3. Если Вы сомневаетесь, хватит ли Вам свободной памяти, то лучше сразу упростить сюжет и ограничиться малым.
4. Неплохих результатов можно добиться, если в течение некоторого времени поиграть в адвентюрные программы на ПК "ZX-SPECTRUM". Например, The Hobbit, Sherlock, Aftershock, Spellbound, Worn in Paradise и т.п. Это будет хорошим и наглядным примером для подражания.
5. Если возникнут вопросы, связанные с созданием адвентюрных программ, Вы можете написать по адресу: 662793, г.Саяногорск, 5-72-36, Гуменюк А.В. (автору "Adventure Game").

```
+-----------------------------------------------------------+
|  Вечная жизнь                                Самойлов С.А. |
+-----------------------------------------------------------+
```

СРАЖЕНИЕ. Для установки в программе "Сражение" вечных жизней и нескончаемого запаса патронов (черепков), нужно:

а) загрузить и запустить программу "Сражение"
б) нажать комбинацию клавиш УС+ВВОД+БЛК
в) загрузить и запустить Монитор-отладчик (сразу по рабочим адресам)
г) внести маленькие изменения:

```
#A3188          #A3B2D
3188 ORA A      3B2D RET
3189 <F4>       3B2E <F4>
```

д) после внесенных изменений можно запускать игру: #G100

ДИЗЗИ. Для установки вечных жизней в программе "Диззи" (фирма "Элита" город Астрахань), нужно:

а) загрузить и запустить саму игру "Диззи"
б) нажать комбинацию клавиш УС+ВВОД+БЛК
в) загрузить и запустить Монитор-Отладчик (загружающийся сразу по своим рабочим адресам, см. Байт-2)
г) внести в программу маленькие изменения:
   #A0D93
   0D93 ORA A
   0D94 <F4>
д) запустить игру: #G100

TERMINUS. Для установки вечных жизней в программе "Terminus" (если не ошибаюсь, то это программа Сергеева А. из фирмы "Элита" г.Астрахань), нужно:

а) загрузить и запустить программу "Terminus"
б) нажать комбинацию клавиш УС+ВВОД+БЛК
в) загрузить и запустить Монитор-Отладчик (загружающийся сразу по рабочим адресам, см. Байт-2)
г) внести в программу маленькие изменения:
   #A0348
   0348 ORA A
   0349 <F4>
д) запустить программу: #G100.

Внимание! Изменился почтовый адрес центра "Байт"(см.выше)! Каталоги, услуги и все остальное осталось без изменений!

В связи с реорганизацией центра несколько месяцев у нас были трудности, но кому сейчас легко? С октября месяца мы вновь набираем рабочие обороты. Будет регулярно выходит инфовыпуск "Байт", будем давать рекламу новинок, оперативно и качественно выполнять Ваши заказы, отвечать на вопросы. В общем вновь начинаем работать для Вас на высоком уровне. Подготовлен очередной 45-й выпуск игровых программ в кодах, новый 32-й "Байт" Вы уже держите в руках, будет изменен вид каталога, имеются планы в отношении новых программ. Работайте с нами - мы оправдаем Ваше доверие! Кстати, цены у нас ниже, чем у других. Приглашаем к сотрудничеству авторов статей по "Вектору". Стоимость "Байта" существенно повышается. Авторские номера авторам статей - бесплатно! Ждем Ваших откликов!

Всегда рады Вам помочь! Центр "Байт"

Вечная жизнь	Самойлов С.А.

-Для установки нескончаемой энергетической хащиты в программе "Галактический патруль" (фирма "Элита" город Астрахань), нужно:

а) загрузить и запустить программу "Галактический патруль"
б) нажать комбинацию клавиш УС+ВВОД+БЛК
в) загрузить и запустить Монитор-Отладчик (загружающийся сразу по своим рабочим адресам, см.Байт-2)
г) внести в программу маленькие изменения:
#A2883 2883 ORA A 2884 <F4>
д) и после этого можно запускать игру #G100

И теперь игра будет с нескончаемой энергетической защитой!!!

-Для установки в программе "Stalker" (фирма "Cinema Soft" город Иркутск) вечных жизней, а также чтобы всегда были и не кончались:
КИРКА, КЛЮЧ, ВОДА и ВСЕГДА СВЕТИЛА ЛАМПА С МАКСИМАЛЬНОЙ МОЩНОСТЬЮ!
Нужно: а) загрузить и запустить игру "Stalker"
б) нажать комбинацию клавиш УС+ВВОД+БЛК
в) загрузить и запустить Монитор-Отладчик (загружающийся сразу по своим рабочим адресам, см.Байт-2)
г) внести теперь уже не маленькие изменения:

```
#A824            060C <F4>     #A3B7            025C <F4>
  0824 ORA A   #A2DD             03B7 ORA A   #M1100,1117,1101
  0825 <F4>      02DD ORA A      03B8 <F4>    #A1100
#A39D            02DE <F4>     #A3C4            1100 MVI A,1
  039D ORA A   #A3AA             03C4 ORA A     1101 <F4>
```

```
   039E <F4>      03AA ORA A      03C5 <F4>    #A1126
 #A60B            03AB <F4>     #A25B            1126 MVI A,80
   060B ORA A                     025B ORA A     1127 <F4>
```

д) и после этих изменений можно запускать игру #G100

Игра теперь будет со всеми нужными Вам штуками (и причем эти штуки не будут уменьшаться!).

Кстати, если при первом запуске программы удерживать клавишу УС, то жизней устанавливается сразу аж 80 (авторы предусмотрели!)!

-Чтобы в игре "Eraser" (Новиков С.А., фирма LANSSoft) не уменьшалось количество жизней, время и всегда была энергия волшебного яблока - нужно:

а) загрузить Монитор-Отладчик (любой режим) и написать небольшую программу: #A100

```
 0100 DI         0102 STA 4DF3   0108 JMP 4A59
 0101 XRA A      0105 STA 4B84   010B <F4>
```

б) сохраняем набранную программу на магнитной ленте:#O>1,1,FF

в) активизируем ROM-загрузчик, комбинацией клавиш ВВОД+БЛК

г) загружаем и запускаем программу "Eraser"

д) нажимаем комбинацию клавиш УС+ВВОД+БЛК

е) загружаем и запускаем нашу сохраненную программу (см. пункты а и б), она сама вносит нужные изменения и запускает игру!

И теперь у Вас в ходе игры не будет уменьшаться количество жизней, время и всегда будет энергия волшебного яблока!

-В игре "Гротоход" вместо 4 байт можно менять только 1, по адресу 1060H записать B7H (ORA A)!

Хакерам об EXOLONe — Городецкий И.И.

В "Байтах" уже писалось, как можно получить бессмертие, но можно не ограничиваться этим и еще более упростить (или усложнить) себе игру.

Далее приведен список ячеек и результат, который получится, если в данную ячейку занести код 0C9H (RET). Все это относится к игре "Exolon" без всяких насадок, неупакованной и имеющей объем 30,25 кб (то есть все адреса рабочие).

1. 1665H - Exolon теряет возможность прыгать
2. 4535H - Exolon не погибнеть на минах, даже без скафандра (он их просто не замечает)
3. 557AH - Exolon не замечает бачков с патронами и гранатами
4. 0698H - исчезают звуковые и визуальные эффекты, связанные со взрывами
5. 4BCBH - исчезает способность Exolona к телепортации
6. 4432H - перестают стрелять большие пушки и верхние двустволки с надстройкой
7. 464FH - перестают стрелять двустволки без надстройки и нижняя пушка из двустволки с надстройкой
8. 3E7EH - не начисляются очки за прохождение двустволок, как обычных, так и со взрываемой надстройкой.

Может быть, кто-то уже разобрался, как кодируются экраны? Было бы очень интересно узнать об этом!

Письмо читателя — Филиппов М.Н.

Уважаемые господа!

Мне довелось ознакомиться с издаваемым вами информационно-рекламным выпуском "БАЙТ". Выпуски понравились и мне захотелось представить для вас и пользователей "Вектора" кое-что из своего опыта работы, а именно

небольшую статью, посвященную устранению некоторых недостатков Basic "Вектор-06Ц" v2.5.

Я более года пользуюсь компьютером "Вектор-06Ц" в комплекте с принтером МС-6113 (EPSON). Наиболее часто используются программы: RETEX и BASIC v2.5. Однако, это выявило некоторые недостатки BASIC-а, а именно:

1. При запуска по БЛК+СБР слишком долго приходится ждать появления заставки;
2. После запуска Бейсика сигнал "STROBE" имеет низкий уровень, что приводит к зацикливанию при попытке выводить информацию на принтер сразу после сброса;
3. Retex передает информацию принтеру в модифицированной альтернативной кодировке ГОСТ, а Бейсик либо в КОИ-7, либо в экзотической кодировке принтера EPSON-FX85, что требует постоянного переключения DIP-переключателей принтера;
4. Мне не нравятся некоторые символы "родного" знакогенератора Бейсика.

Мною разработана процедура, позволяющая исправить все вышеперечисленные недостатки Бейсика. Для выполнения этой процедуры необходимы Basic v2.5 и Монитор-Монстр.

Заставка программы формируется обычной программой на Бейсике, и поэтому, изменив эту программу можно ускорить запуск Бейсика и выполнить заодно некоторые операции по настройке интерпретатора. К новой стартовой программе предъявляются следующие требования:

а) объем ее не должен превышать 1011 байт, то есть число, выдаваемое по команде PRINT FREE(0) должно быть не меньше, чем 14636;
б) она должна содержать операторы очистки экрана, установки: палитры, типа принтера;
в) стартовая программа для устранения недостатка (2) должна содержать оператор OUT 5,16.

Ниже приводится мой вариант стартовой программы:

```
0 CLS:FORI=0TO15:SCREEN0,I,0:NEXT:I=255:SCREEN3,I,I,I,I,I,I,I,I:COLOR
  15,0,0:PLOT2,246,2:LINE2,1,BS:PRINT"BASIC:LINE1,1,BS
1 PRINT" ";CHR$(34);"ВЕКТОР-06Ц";CHR$(34);" V2.5":CUR0,23:PRINT"П/О
  СЧЕТМАШ КИШИНЕВ 1988":PRINT"=>":OUT5,16:SCREEN6,1
2 SCREEN0,1,128,16,208,6,134,22,173,173,197,34,192,2,152,82,173:B=256:
  DIMX(50),Y(50):COLOR8:SCREEN2,8;CLS
3 FORI=0TO50:PLOTX(I),Y(I),0:C=INT(RND(1)*B):D=INT(RND(1)*B):X(I)=C:
  Y(I)=D:PLOTC,D,1:IFNOTINKEY$=" "THEN5
4 NEXT:GOTO3
5 CLS:SCREEN0,7,54,0:SCREEN2,15:COLOR15:CUR0,23:NEW
```

Эту программу (или аналогичную программу) следует вывести на МЛ в формате МОНИТОРА для последующего вмонтирования в Бейсик командой:

```
BSAVE"FONT",17152,32690-FRE(0)
```

Для изменения "родного" знакогенератора необходимо отредактировать его при помощи любого редактора знакогенератора Бейсика и затем вывести на МЛ командой:

```
BSAVE"FONT",14624,15263
```

Теперь можно приступить к созданию "новой" версии Бейсика. Для этого надо загрузить Монитор-Монстр и выбрать вариант размещения <3>. Затем по команде ВМ (загрузка со смещением 100H) загружаем файл BASIC.ROM.

Далее вставляем в него созданные ранее стартовую программу и новый фонт:

```
#ISTART
#RFA68      (загрузка стартовой программы)
#IFONT
#R100       (загрузка нового фонта)
```

Для устранения третьего недостатка следует изменить таблицу, по которой, при выбранном принтере EPSON, драйвер принтера перекодирует сим-

волы кириллицы. Для этого надо по директиве S изменить содержимое следующих ячеек памяти:

```
#S1EBF
  1EBF C0 9E  1EC7 B7 95  1ED0 C1 9F  1ED8 BD 9B
  1EC0 A1 80  1EC8 AA 88  1ED1 B2 90  1ED9 A9 87
  1EC1 A2 81  1EC9 AB 89  1ED2 B3 91  1EDA BA 98
  1EC2 B8 96  1ECA AC 8A  1ED3 B4 92  1EDB BF 9D
  1EC3 A5 84  1ECB AD 8B  1ED4 B5 93  1EDC BB 99
  1EC4 A6 85  1ECC AE 8C  1ED5 A8 86  1EDD B9 97
  1EC5 B6 94  1ECD AF 8D  1ED6 A3 82  1EDE 5F B1
  1EC6 A4 83  1ECE B0 8E  1ED7 BE 9C  1EDF .
              1ECF B1 8F              #
```

Эти изменения позволяют отказаться от постоянных переключений кодовых таблиц в принтере и работать только с альтернативной кодировкой ГОСТа.

Теперь недостатки Бейсика перечисленные в начале статьи устранены и осталось вывести на МЛ исправленную версию:

```
#OBASIC и #O>1,40,FF
```

Приведенная выше процедура модификации Бейсика позволяет, на мой взгляд, любому пользователю приспосабливать интерпретатор под свои потребности.

125183, г.Москва,п-д Черепановых,56 корпус 1, 9

Проценты и рубли

Горелов А.В.

Многие, у кого есть свободные средства, и их пока маловато или некуда потратить, предпочитают не хранить деньги в чулке, где они потихоньку обесцениваются, а вложить их в банк под проценты на определенный срок. Естественно возникает мысль, что неплохо бы использовать родной "Вектор" для расчета ожидаемых доходов. Наш постоянный читатель Горелов А.В. из поселка Тисуль предлагает свой вариант программы на Бейсике для расчета процентов по вкладам. С ее помощью Вы сможете определить, где наиболее выгоднее хранить свои кровные.

Математика тут простая. Сумма процентов за весь срок хранения рассчитывается по формуле:

$$\text{Проценты} = \frac{\text{Ставка} * \text{Дни} * \text{Сумму вклада}}{\text{Год} * 100\%}$$

Где "год" - число дней в году. Разные банки считают по разному. Одни считают, что в году 365 дней и месяцы все разные по числу дней. Другие считают, что в году 360 дней, а все месяцы одинаковые по 30 дней.

Часто бывает ситуация, когда положив средства, например, на месяц, вкладчик снова перекладывает их на новый срок вместе с набежавшими процентами. Чтобы определить доход в этом случае при постоянной годовой ставке, нужно определить коэффициент роста=(сумма вклада + проценты)/сумму вклада, возвести его в степень, равную числу переоформлений, вычесть 1 и умножить на первоначальную сумму вклада. Например, переоформляя три раза, при коэффициенте роста 1.1, мы увеличим первоначальный капитал в 1.1^3=1.331 раза, то есть прибыль составит 33% от первоначальной суммы вклада.

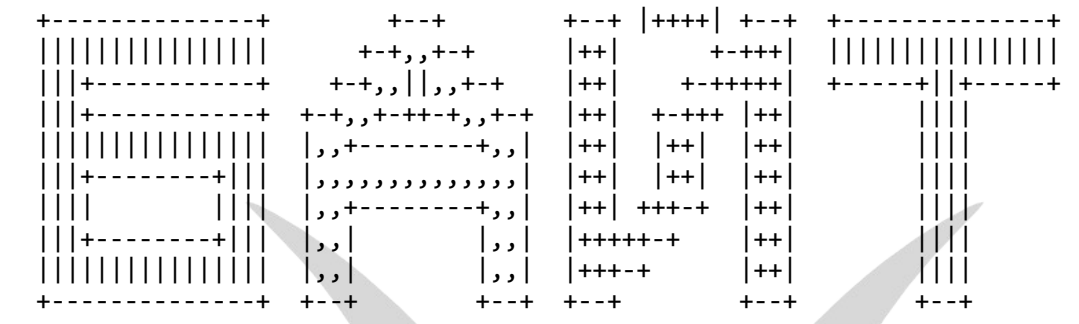
```
Сумма вклада---------100                Сумма вклада---------100
Срок хранения--------2                  Срок хранения--------2
Годовой процент------90                 Проценты за месяц----7
Проценты за 2 месяца=15                 Проценты за 1 месяц= 7
        Общая сумма= 115 руб.                Общая сумма=114.49 руб.
Доход за 2 месяца хранения= 15     Доход за 2 месяца хранения= 14.49
```

```
1 REM ГОРЕЛОВ А.В.
2 REM ТИСУЛЬ 17.03.95
10 SV$="СУММА ВКЛАДА----------"
```

```
   20 SH$="СРОК ХРАНЕНИЯ---------"
   30 PM$="ПРОЦЕНТЫ ЗА МЕСЯЦ-----"
   40 PG$="ГОДОВОЙ ПРОЦЕНТ-------"
   50 PZ$="ПРОЦЕНТЫ ЗА "
   60 ME$="МЕСЯЦ = "
   70 DZ$="ДОХОД ЗА"
   80 MH$="МЕСЯЦЕВ ХРАНЕНИЯ = "
   90 OS$="        ОБЩАЯ СУММА = "
   100 PRINT"ВЫБЕРИТЕ ВИД ВКЛАДА (1ИЛИ 2)"
   110 PRINT"     1 ДЕПОЗИТ ГОДОВОЙ"
   120 PRINT"     2 ДЕПОЗИТ ЕЖЕМЕСЯЧНЫЙ"
   128 LPRINT CHR$(27);CHR$(82):REM НАСТРОЙКА ПРИНТЕРА
   130 A$=INKEY$:IF A$="" THEN 130
   140 IF A$="1" THEN 278
   150 IF A$="2" THEN 160
   155 GOTO 130
   160 D=1
   170 GOSUB 390
   178 PRINT PM$;
   188 INPUT C: LPRINTPM$;C
   198 C2=A*C/100
   208 GOSUB 450
   228 D=D+1
   248 IF D>B THEN 268
   258 GOTO 198
   268 GOSUB 480
   269 RUN
   278 GOSUB 390
   320 D=B
   328 PRINT PG$;
   338 INPUT C: LPRINT PG$;C
   348 C2=A*C/100/12
   358 C2=C2*B
   360 F=A
   368 GOSUB 450
   380 GOSUB 480
   388 RUN
   390 PRINT SV$;
   400 INPUT A:LPRINTSV$;A
   410 F=A
   420 PRINT SH$
   430 INPUT B: LPRINT SH$
   440 RETURN
   450 PRINT PZ$;D;ME$;C2
   452 LPRINT"ПРОЦЕНТЫ ЗА";D;"МЕСЯЦ=";C2
   460 A=C2+A
   465 PRINTOS$;A;"РУБ."
   467 LPRINTOS$;A;"РУБ."
   470 RETURN
   480 PRINT DZ$;B;MH$;A-F
   485 LPRINTDZ$;B;MH$;A-F
   490 RETURN

  +-----------------------------------------------------+
  |  Вечная жизнь                          Вахрушев А.Н. |
  +-----------------------------------------------------+
-Fire Rescue
 Число  жизней задается ячейкой 0958H, хранится  в  ячейке 7AB1H,  для
установки "вечной жизни" нужно  заменить содержимое ячейки 1521H с DCR
```

A на NOP.
-Color Ball
 Число жизней задается в ячейке 0A59H, хранится в ячейке 7314H, для установки вечной жизни нужно заменить в ячейке 0B2FH с DCR A на NOP.
-Eric and the Floaters
 Число жизней задается в ячейке 010BH, хранится в ячейке 730FH, для вечной жизни нужно заменить в ячейке 0CF2H с DCR A на NOP.
-Ninjya
 Число жизней задано в ячейке 2E31H, для установки вечной жизни заменить в ячейке 12ABH с DCR A на NOP.
 Надеюсь, что эти сведения кому-нибудь помогут и будут полезны.

     +---+
     | Секреты игровых программ   Соловьев А.Г. |
     +---+
-Down to Earth (автор Заставной г.Волгоград)
 Можно менять уровни, нажав во время игры одновременно клавиши: стрелка влево вверх, стрелка вверх, СТР, стрелка влево, стрелка вниз и стрелка вправо.
-Down to Earth (автор Лебедев А.З. г.Волгоград)
 Можно менять уровни, нажав во время игры одновременно клавиши "УС"+"СС".
-Side Arm
 Код бессмертия 222517. Для того, чтобы выбрать любое оружие, нужно во время игры сначала дойти до одного из видов оружия и задеть космонавтом, только после этого можно выбрать взятое оружие, нажав клавишу "РУС/ЛАТ". Всего имеется пять видов оружия: 1-й вид имеется у космонавта в начале игры - шарик, 2-й вид - бомбочка, 3-й вид - два шарика, 4-й вид - две ракеты, 5-й вид - три ракеты.
-Пекманенок
 Менять уровни можно, нажав во время игры клавишу "ПС" и нажимая клавишу "стрелка вправо".
-Diamond Country
 В этой игре всего 10 уровней. Для того, чтобы прибавить жизни (максимальное число - 79 жизней), нужно во время игры удерживать нажатой клавишу "пробел" и нажимать клавишу "Z". Для выбора нужного уровня игры достаточно во время появления заставки (названия игры) нажать одновременно клавиши "X" и "Y", появится запрос на нужный уровень сложности. Для перебора номеров уровней следует нажать клавиши "стрелка влево" и "стрелка вправо".
-Буря в пустыне
 Для того, чтобы добраться до города надо набрать 1200 очков, до дворца - 1280 очков, где по его краям стоят два танка (их нужно быстро уничтожить). В пустыне: за расстрел танка прибавляется 10 очков, БТР-а - 7 очков, пушки - 4 очка, человека - 1 очко, его можно и раздавить.

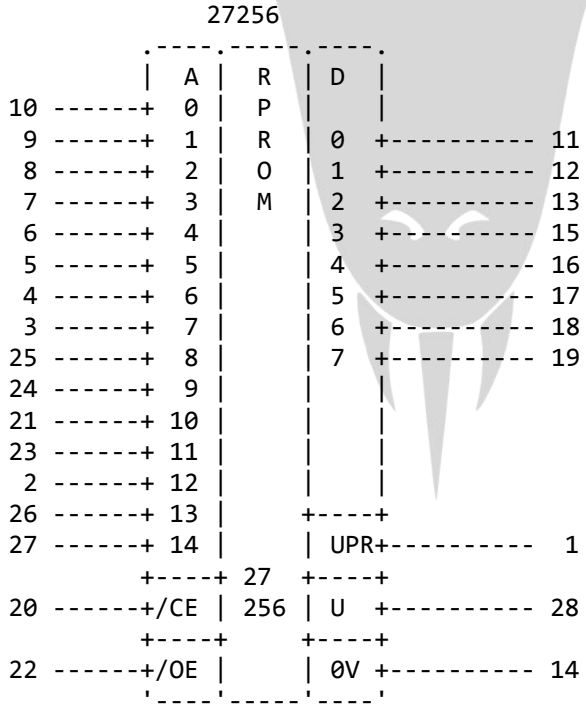


33 выпуск

610011, г. Киров, а/я 326, Шустов Алексей Михайлович

НЕКОТОРЫЕ МОМЕНТЫ ЗАМЕНЫ ПЗУ КР573РФ2 (КР573РФ5, 2716) НА 27256

Вы никогда не думали, что можно заменить векторовскую двух килобайтную ПЗУ-ху с универсальным загрузчиком на более вместимую, например на 27256 на 32 кбт? Мне в голову никогда такие мысли не приходили с тех самых времен, когда я в свой Вектор поставил универсальный загрузчик. Я был очень рад, что все прекрасно работает. Но, тут недавно понадобилось сделать так, чтобы можно было быстро грузить некоторую программу, но не с магнитофона и не с дисковода. "А как же еще?",-подумал я. Ставить целую плату "картриджа для Вектора" Шашкова - не разумно. Тогда Виктор Саттаров предложил просто-напросто поменять ПЗУ Вектора на 32 кбт 27256. (Он, кстати говоря, знал об этом, но мне не говорил, потому что никогда разговор на эту тему не заходил. Хорошие у него мысли в голове.) Короче, я понял, как это делается - очень просто, так же просто, как просто включить телевизор. Вот так, примерно, выглядит 27256 (сейчас я ее нарисую):



Не совсем по ГОСТу получилось, вернее - совсем не по ГОСТу, ну да ладно. Значит так.

1. Берете в руки паяльник и выпаиваете панельку из под ПЗУ (если она у вас в панельке, конечно).
2. Берете другую панельку на 28 ножек и впаиваете обратно так, чтобы 1, 2, 27 и 28 ножки остались висеть в воздухе (для них дырок должно не оказаться), т.е. запаивать начинаете с 3 ножки. Кстати, панельку можно и не менять, а втыкать ПЗУ в старую 24-ножечную панельку. Тогда, опять же, 1, 2, 27 и 28 ноги 27256 останутся снаружи и на них придется напаивать провода.

Поехали дальше.

3. Отсоединяете от 26 ноги +5В.
4. Отсоединяете от 23 ноги +5В.
5. На 1 ногу кидаете +5В.
6. На 28 ногу +5В.
7. К 23 ноге подключаете 16 ногу D1 (КР580ВА87)
8. Ко 2 ноге подключаете 15 ногу D1
9. К 26 ноге подключаете 14 ногу D1
10. К 27 ноге подключаете 13 ногу D1

И последнее.

11. Сигнал, который идет на 8 ногу D10 (К155ЛА1) отрезать от 8 ноги D10 и соединить с 8 ногой D1.
12. Вот и все, ребята.

Все очень просто, только сейчас надо прошить ПЗУ. Прошивать можно по разному, вернее разным. Например, я себе оставил универсальный загрузчик (который сделали Темиразов и Соколов), а далее поместил нужные мне программы. Если при включении компьютера нажать клавишу F5 (или при его сбросе), то происходит перекидка моей программы из ПЗУ в память с 100Н адреса. Затем нажимаю "БЛК"+"СБР", и запускается моя маленькая программка, которая рисует менюшку с именами программ, которые я прошил в ПЗУ. Выбираю программу и запускаю ее. Все происходит очень быстро. Себе я зашил 4 программы: "DISK LOADER", "TESTKD", "TEST-ДОЖДЬ" и "ТЕСТ УСТРОЙСТВ".

Программу универсального загрузчика пришлось чуть-чуть изменить, например, таким образом:

```
        ORG     006CH
        JMP     006DA
        ;
        ORG     006DAH
        LDA     0DEF7H
        CPI     07FH        ; клавиНа F5
        JNZ     006FH
        LXI     H,0800H     ; перекидка
        LXI     D,0100H
        LXI     B,7800H
P1:     MOV     A,M
        STAX    D
        INX     H
        INX     D
        DCX     B
        MOV     A,B
        ORA     C
        JNZ     06EBH
        MVI     A,020H      ; рисование строчки
        JMP     05D7H       ; столбиков ( просто для прикола )
;
```

Чтобы вставить эту подпрограмму в универсальный загрузчик, нужно ее отассемблировать, получить HEX-файл, затем в SID загрузить универсальный загрузчик с 100Н адреса, а затем вгрузить HEX-файл этой подпрограммки в

универсальный загрузчик таким образом:
  #IPODPR.HEX
  #R100
  Далее нужно подгрузить с 900H адреса программы, которые вы будете запускать из ПЗУ-хи с подпрограммкой, которая будет запускать их оттуда. Эта подпрограммка осуществляет простую перекидку выбранной программы с адреса ее нахождения в ПЗУ на 100H адрес, а потом запускает ее. Весь файл должен занимать не более 32 кбт - универсальный загрузчик 2 кбт и ваши программы с запускальщиком 30 кбт. Как только вы все это сделаете, остается прошить ПЗУ и вставить в панельку. Если вы все правильно сделали - должно все правильно заработать. Вот только дисковод перестанет грузить файлы с 0 дорожки. Это происходит потому, что универсальный загрузчик использует память с 1100H адреса для работы с дисководом. Когда универсальный загрузчик зашит в ПЗУ на 2 кбт, то при чтении из памяти в интервале с 0000H по 07FFH адресов происходит выборка ПЗУ, то есть байт памяти читается из ПЗУ, а при следующих адресах - из ОЗУ, а когда стоит ПЗУ 27256, то чтение из ОЗУ возможно только с 8000H адреса. Поэтому, когда программа универсального загрузчика, зашитая в 27256, читает байт памяти, например, с 1104H адреса, то происходит чтение этого байта из ПЗУ, а не из ОЗУ, куда перед этим была прочитана информация с 0 дорожки дискеты. Поэтому мне пришлось чуть-чуть изменить подпрограмму работы с дисководом. Все, конечно, заработало, но мне захотелось большего - я выкинул всю эту подпрограмму и вставил свою, которая занимает примерно в 2.5 раза меньше памяти и в 4-5 раз быстрее грузит. Моя подпрограмма не расчитана на работу с 40 дорожечными дискетами одинарной плотности, которые, я думаю, уже никто не использует. Файл на системных дорожках должен запускаться с 100H адреса.

```
;
        ORG     02E7H
        MVI     A,0C3H
        STA     0000H
        LXI     H,0100H     ; адрес запуска загруженного
        SHLD    0001H       ; файла
        LXI     H,07CAH     ; рисование дискетки
        MVI     A,8AH
        CALL    06B2H
        MVI     A,34H       ; задание упр. слова
        STA     0DED0H      ; дисковод A
        CALL    UPSL0       ; ожидание готовности
        XRA     A           ; восстановление
        OUT     1BH
        CALL    INDEX
        MVI     C,1         ; читаем 1 сектор 0 дорожки 0 стороны
        LXI     H,0DFE0H    ; на адрес 0DFE0H
        CALL    CHTSECS
        JNZ     0000        ; если произошла ошибка
;
        LXI     H,0DFE0H    ; проверка
        LXI     D,0DEF1H    ; правильности чтения
        MVI     C,1FH
        MVI     A,66H
P1:     ADD     M
        MOV     B,A
        MOV     A,M
        STAX    D
        INX     H
        DCX     D
        DCR     C
        MOV     A,B
```

```
        JNZ     P1
        SUB     M
        JNZ     0000        ; если не правильно
        STAX    D
;
        LDA     0DFE4H      ; тут длина читаемого файла
        MOV     B,A         ; в секторах
        LXI     H,0080H     ; адрес загрузки файла
B2:     MVI     C,1         ; начинаем чтение с 1 сектора
B3:     CALL    CHTSECS
        JNZ     0000        ; ошибка чтения
        DCR     B
        JZ      P2          ; проверка на конец чтения
        INR     C           ; след. сектор
        MVI     A,6
        CMP     C
        JNZ     B3
        LDA     0DED0H      ; след. сторона
        XRI     04H
        STA     0DED0H
        CALL    UPSL0
        MOV     A,D
        ANI     04H
        JZ      B2
        MVI     A,58H       ; след. дорожка
        OUT     1BH
        CALL    INDEX
        JMP     B2          ; чтение далее
;
P2:     LXI     H,0100H     ; все правильно прочитали
        LDA     0DFE4H      ; подсчет размера файла в байтах
        RLC
        RLC
        MOV     B,A
P3:     CALL    03B5H       ; рисование загрузочных столбиков
        INR     H
        DCR     B
        JNZ     P3
        RET                 ; выход на запуск
;
CHTSECS:CALL    INDEX       ; подпрограмма чтения сектора
        MOV     A,C
        OUT     19H
        LXI     D,103H
        MVI     A,80H
        OUT     1BH
CHTSS:  IN      1BH
        RRC
        JNC     CHTSS
CHTSS1: IN      1BH
        ANA     E
        SUB     D
        JZ      CHTSS1
        IN      18H
        MOV     M,A
        INX     H
        JP      CHTSS1
        DCX     H
        IN      1BH
```

```
        ANI     9CH
        RET
;
INDEX:  LDA     0DED0H      ; подпрограмма проверки
        OUT     1CH         ; готовности контроллера
        IN      1BH
        RRC
        JC      INDEX
        RET
;
UPSL0:  MOV     D,A         ; подпрограмма проверки
UPSL:   OUT     1CH         ; готовности дисковода
        IN      1BH
        RLC
        MOV     A,D
        JC      UPSL
        RET
;
```

Отассемблируете эту подпрограмму и вставите в универсальный загрузчик так, как написано выше.

Вот и все.

Если кому-то не понятно все это, то пишите и спрашивайте. А еще лучше, присылайте мне 30 кбт ваших файлов (если сжатых, то тогда с распаковщиком или с упаковщиком), ПЗУ 27256 (или 15.000 руб.почтовым переводом на нее), 10.000 руб. за работу, и я вам вышлю прошитую, полностью рабочую ПЗУ, которую вам остается только всунуть в панельку и больше не мучиться. Можно прислать и 60 кбт ваших файлов, ПЗУ 27512 (на 64 кбт) или 15.000 руб. и 10.000 руб. за мою работу. Тогда вместе с прошитой ПЗУ я вышлю информацию, куда дополнительно нужно бросить проводок для правильной работы 27512 ПЗУ-хи. Тогда и никаких "картриджей для Вектора" Шашкова не надо. Пропилили себе дыру над ПЗУ в корпусе Вектора и только знай меняй их.

Я не писатель.
Меринов Сергей (FMSSOFT)
610031, Киров, а/я 2729

От редактора

Это все номера издания «Байт», которые удалось найти. Возможно, что были другие, но мне о них не известно. Надеюсь, что кто-то сочтет возможность прикоснуться к компьютерной истории ранних 1990-х интересной, полезной, или хотя бы забавной. Издание было посвящено «Вектору-06Ц» – возможно самому яркому, цветному и многозвучному из советских компьютеров.

Не все номера сохранились одинаково хорошо. Я постарался сохранить максимум информации при минимуме вмешательства в содержимое. Кое-где я позволил себе изменить форматирование, использовать атрибут подчеркивания, заменить псевдографику на ASCII. Некоторые таблицы и заголовки я восстановил в соответствии с их внешним видом в эмуляторе. Некоторые заголовки и схемы приведены в виде скриншотов из программы Emulator 3000. Не всю информацию удалось сохранить, но в основном это касается таблиц справочного характера, которые можно найти где-нибудь еще. К сожалению даты номеров встречаются лишь эпизодически, особенно в поздних номерах. Первый номер вышел в июне 1991-го года, а вот о дате выхода последнего 33-го номера остается только догадываться.

Огромное спасибо Александру Тимошенко, которых сохранил у себя весь архив номеров и любезно предоставил его мне. Без него вся информация из этого сборника считалась бы без вести пропавшей, может быть навсегда. Отдельное спасибо Евгению Троицкому, автору Emulator 3000, за хороший и удобный эмулятор.

Дата последней редакции: 15.11.08