

«ИНФОРКОМ»

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР
ZX-SPECTRUM
Сборник статей



ZX-ФОРУМ №1



МОСКВА-1994

PDF version by Deny (Денисенко Д.А.)
e-mail: DenyDA@mail.ru
2007

ZX-FORUM

ABUOL



Данная книга является сборником трудов отечественных пользователей персонального компьютера ZX-Spectrum. В ней представлены рекомендации и полезные советы, касающиеся аппаратных доработок компьютера, улучшение программной совместимости. Большой объем книги составляют материалы, посвященные обмену опытом между программистами. Любители компьютерных игр найдут на страницах книги полезные советы и подсказки.

СОДЕРЖАНИЕ

ОТ РЕДАКЦИИ

СПЕКТРУМ В РОССИИ

С. Симонович. **Положение на современном рынке программ**

С. Симонович. **Об основных этапах развития "Спектрума" в России**

С. Симонович. **Обучающее программное обеспечение**

СИСТЕМА iS-DOS

В гостях у фирмы "SLOT" (интервью)

М. Головин. **Опыт программирования в операционной системе iS-DOS. Пример написания драйвера принтера**

СЕКРЕТЫ TR-DOS

А. Бастраков. **Применение процедур TR-DOS без использования области системных переменных**

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

А. Локтев. **Mouse + Спрессу или подключение манипулятора MM8 031 к персональному компьютеру "ZX-Spectrum"**

СИСТЕМНЫЕ ПРОГРАММЫ

С. Куковьякин. **Расширение функций оператора PRINT**

В. Риккер. **Удобный интерфейс**

К. Парчевский. **Применение прерываний 2-го рода. Резидентная программа "Часы"**

ОБМЕН ОПЫТОМ

Д. Пьянков, С. Машарин. **Опыт русификации программ**

С. Сурков. **Вывод спрайтов на экран с маской**

В. Витвинов. **Генерация шаблонов 2x2 с использованием статического RND**

Е. Горцуев, А. Шарапов. **Улучшение считываемости программ с магнитофона**

Л. Черкашин., А. Бастраков, Д. Барановский **Повышение совместимости схемы "Ленинград-1"**

И. Хромин. **Доработки компьютера "Пентагон-128"**

И. Игольников. **Повышение совместимости и некоторые усовершенствования для компьютера "Дельта-С"**

А. Гармаисанов. **Подключение музыкального сопроцессора**

СОВЕТЫ ЭКСПЕРТА

П. Троеглазов, Г. Троеглазов. **"SimCity"**

В. Пахомов. **"Hero Quest"**

МОЗГОВОЙ ШТУРМ

"Magnetron"

КОМПЬЮТЕРНАЯ НОВЕЛЛА

О. Дмитриев. **"The Little Puff"**

ИГРЫ: РОКЕС

ИГРЫ: ПАРОЛИ

ПРИЛОЖЕНИЕ: Программа для ввода шестнадцатеричных дампов

ОТ РЕДАКЦИИ

Дорогой читатель!

Четыре года назад, в сентябре 1990 года мы объявили о том, что начинаем выпуск периодического издания "ZX-РЕВЮ" для всех, кто любит бытовой персональный компьютер "ZX-Spectrum". Тогда ещё не было "Закона о печати" и выпускать журнал или какое-либо другое средство массовой информации мы никак не могли. Так "ZX-РЕВЮ" стал "персональным письмом" (никогда не было законов, запрещающих нам писать в Ваш адрес персональные письма, а Вам их читать и писать нам ответы). Это сегодня у нас есть лицензия на издательскую деятельность, а тогда не было ничего, кроме желания пропагандировать знания об этом замечательном компьютере и рассказывать людям о его необъятных возможностях, о которых при его скромном внешнем виде и небольших габаритах несведущему человеку порой очень трудно догадаться.

Прошло почти четыре года. Сейчас в стране действуют миллионы активных пользователей компьютером "ZX-Spectrum". Ушли в прошлое те времена, когда нам приходилось рассказывать о том, как программируют на БЕЙСИКе, чем отличаются операторы РЕЕК и РОКЕ. Десятки тысяч наших читателей за эти годы активно освоили программирование в машинных кодах и сейчас многие пишут свои прикладные и развлекательные программы.

На глазах выросли наши читатели, но вместе с ними вырос и сам "Спектрум". Если бы сэр Клайв Синклер взглянул на свое детище у нас в России, то, наверное, не узнал бы его. "Спектрум" обзавелся мегабайтом оперативной памяти, несколькими дисковыми операционными системами (CP/M, TR-DOS, iS-DOS), новыми развитыми ПЗУ. Современные модели типа "Профи", "АТМ-Турбо 2", "Скорпион 256" и другие намного опередили мечты своего гениального создателя. Сейчас "Спектрум" уверенно занимает положение где-то между развлекательными игровыми машинами (видеопроставками) и профессиональными компьютерами. Здесь напрашивается мысль о том, что он взял и у тех и у других что-то очень хорошее, но это не так. Он ничего у них не брал. Он от рождения имеет многое такое, чем не обладают ни современные игровые приставки, ни дорогие профессиональные машины.

Гибкость в выборе программного обеспечения, широкий спектр языков программирования, графических и музыкальных редакторов, несравнимая ни с чем доступность широчайшего спектра игровых программ, простота программирования, легкость объединения графики, анимации и текста в своих проектах, самый лёгкий доступ к управлению памятью, высокое отношение скорости процессора к объёму экранной памяти, простота архитектуры, плюс далеко ещё неисчерпанный потенциал развития - вот только некоторые из его особенностей, делающие его компьютером для миллионов. Таким он и стал в России.

На очереди новые задачи - подключение "жесткого диска", создание коммуникационных пакетов, переход на работу в сетях, создание интерфейсных устройств для подключения к другим компьютерным платформам. И всё эти работы уже ведутся. Над развитием "Спектрума" работает вся страна.

Редакция "ZX-РЕВЮ" получает много писем, в которых читатели сообщают о своих находках, достижениях, делятся проблемами. Как можем, мы отражаем их в разделе "ФОРУМ", который стал самым любимым нашими читателями. Но, к сожалению, далеко не все из того, что к нам приходит, может быть опубликовано из-за недостатка места. Так родилась идея создания книги "ZX-ФОРУМ". Над этой книгой вместе с нами работала вся страна.

Многим, наверное, хотелось бы, чтобы в этой книге были опубликованы только новейшие достижения и самые передовые идеи. Возможно, что это имело бы большой смысл. Но то, что для одного является великим открытием, для другого - давно пройденный этап. Смысл же нашего раздела "ФОРУМ" всегда сводился к тому, что каждый наш читатель имеет право на рассказ о своих достижениях и должен быть услышан. Даже если молодому человеку всего тринадцать лет, а он освоил машинный код и с тяжелым трудом сумел разобраться в сложной программе и найти свой способ "обессмертить" её, то он уже принес свой маленький кирпичик и положил его в фундамент будущего здания, которое мы с Вами строим. За этим "кирпичиком" последуют другие,

и через несколько лет у нас будет на одного талантливое программиста больше.

Огромное спасибо всем, кто написал в нашу редакцию, всем, кто внёс свой вклад в дело развития "Спектрума". Мы проделали большую работу, собрав, отредактировав, обобщив и опубликовав этот интересный материал, но если бы не Вы, уважаемые коллеги, эта книга, наверное, не появилась бы на свет. Мы надеемся, что она поможет новым пользователям этой замечательной машины ещё лучше узнать её возможности и познакомиться с Вашими трудами и с Вашими достижениями.

С глубокой признательностью, "ИНФОРКОМ".

Авторов материалов, вошедших в данный сборник, просим связаться с нами по тел. (095) 945-28-67 (вторник, среда) до 01.10.94 с целью получения авторского вознаграждения.

СПЕКТРУМ В РОССИИ

Положение на современном рынке программ.

© Сергей Симонович, 1994. (Директор по перспективному развитию МКП "ИНФОРКОМ")

Прошли те времена, когда программист, желающий создать что-то свое, был обречён на неизвестность и непризнание. Высокая стоимость магнитных носителей, низкая стоимость интеллектуального труда, доступность зарубежного программного обеспечения, отсутствие системы коммерческого распространения авторских разработок никак не способствовали тому, чтобы у квалифицированных программистов был смысл и желание что-то делать, кроме как по прямому заказу какой-либо зарубежной фирмы (с оплатой в валюте по минимальным европейским стандартам).

Сейчас положение изменилось. Понемногу, постепенно, в ходе экономической реформы выяснилось, что одна хорошая авторская программа вполне может стоить в несколько раз больше, чем дискета, на которой она записана, и при этом может укладываться в платежеспособный спрос населения.

Первыми работами отечественных программистов стали адаптации известных зарубежных программ, как прикладных, так и развлекательных (русификация, дискофикация, расширение возможностей и пр.). Затем последовали авторские работы "по мотивам" известных фирменных программ с использованием "фирменного" стиля, музыки, спрайтов, но построенные по собственному авторскому сценарию. И, наконец, стали появляться полностью авторские программы, в которых всё реже и реже встречаются элементы заимствования и всё чаще и чаще проскальзывают собственные оригинальные находки.

Как самая демократичная машина, "Спектрум" характерен тем, что практически любую программу в законченном виде для него может написать ограниченный коллектив авторов (один - два человека). Для нас этот факт очень важен, поскольку на Западном рынке программного обеспечения сложилось положение, когда современные программы для основных аппаратных платформ создают коллективы в 30-50 человек. Стоимость разработки типовой программы для IBM-PC или для "АМИГИ" уже превышает миллион долларов. Работа фирмы, выпускающей современное программное обеспечение, уже сейчас похожа по масштабам на работу развитой киностудии. Типичный состав группы, выпускающей программы на Западе, сегодня включает:

- ✓ несколько художников и аниматоров;
- ✓ композитора-оранжировщика;
- ✓ звукорежиссерскую группу;
- ✓ видеоинженеров;
- ✓ бригаду сценаристов;
- ✓ штат менеджеров;
- ✓ десятки профессиональных актеров;
- ✓ и... всего лишь 1-3 программистов, на долю которых остаются только три задачи:
 - ✓ увязка всего комплекса в единое целое;
 - ✓ обеспечение совместимости готового продукта с всевозможнейшим аппаратным обеспечением и программным окружением, какое только может оказаться у пользователя;
 - ✓ работы по созданию собственных программно-инструментальных средств, с помощью которых автоматизируется работа над новыми проектами.

Важно, что из всех этих задач фактически только последняя является творческой. Для первых же двух используют стажеров, которые по мере набора опыта переходят на более интересное и более высокооплачиваемое третье (поисковое) направление.

Что же из всего этого следует?

Вероятность того, что в ближайшие годы в России сложатся подобные коллективы и притом смогут успешно конкурировать с западными производителями, конечно есть, и они уже создаются ("НИКИТА", "GAMOS" Ltd, Лаборатория 79 "ЛУНА", "РУСС", "AF-Computers", "Медокс" и некоторые другие), но их пока мало и им ещё предстоит пройти длительный и тяжёлый путь развития. В основном же, как правило, отечественным производителям приходится

ограничивать свой "репертуар" такими жанрами, которые не требуют привлечения всего вышеописанного дорогостоящего штата сотрудников и в которых удельный вес программистского труда больше, чем вес труда графико-, звуко- и видеооформителей.

Таким образом, на долю наших производителей программного обеспечения, как правило, остаются следующие направления.

- ✓ логические игры;
- ✓ активно-логические игры;
- ✓ традиционные игры (и их модификации);
- ✓ обучающие и развивающие программы.

И, надо сказать, в этом направлении мы наиболее далеко продвинулись. Достаточно вспомнить о бессмертном "Тетрисе" (активно-логическая) игра, который вошел во все основные аналитические исследования за рубежом в качестве примера того, как программист в одиночку может потрясти весь мир.

В остальных жанрах нашим производителям программ продвинуться крайне сложно. Нам известны многочисленные попытки различных коллективов "оседлать" жанр программ "ADVENTURE" и "ARCADE/ADVENTURE", но из-за недостаточного финансирования все эти работы были свернуты. Попытки выйти на рынок с программами типа RPG нам вообще неизвестны. Немного лучше обстоит дело с программами-имитаторами. Здесь есть успешные реализации, хоть их и не много и они малоизвестны. По-видимому, здесь наметился прогресс благодаря тому, что отраслевые НИИ работают над подобными системами в качестве тренажеров (по заказам своих министерств), а попутно выпускают демонстрационные и коммерческие версии, зарабатывая на этом если и не деньги, то хотя бы известность и моральное право на продолжение финансирования.

Таким образом, развитие программного обеспечения на ближайшие годы идет по пути формирования крупных творческих коллективов, которым, не приходится рассчитывать ни на какую иную финансовую поддержку в своем развитии, кроме как на собственную прибыль, а как её получить в условиях конкуренции с западным программным обеспечением остается неясным. Зато нам совершенно ясна технология образования подобных коллективов. В основе будущей команды всегда лежит "ядро" из одного - трех высококвалифицированных программистов, "раскрутившееся" на одном из доступных и сильных направлений. И только когда такое "ядро" уже создано и показало, что оно умеет и желает работать, оно начинает постепенно "обрастать" новыми кадрами и переходить к новым проектам.

И здесь нам на помощь приходит наш незаменимый "ZX-Spectrum". Это ЕДИНСТВЕННАЯ в стране аппаратная платформа, для которой сошлись в одной точке две ключевые проблемы:

- ✓ для этого компьютера есть РЕАЛЬНАЯ возможность создания в сжатые сроки малыми коллективами программных продуктов ЛЮБЫХ жанров, классов и категорий;
- ✓ на платформе этого компьютера стоят несколько миллионов потенциальных потребителей (покупателей) программного обеспечения, что дает РЕАЛЬНУЮ возможность "раскрутить" будущую команду, проверить свои идеи и концепции, приобрести первый практический опыт и отладить свои технологические приёмы.

Об основных этапах развития "Спектрума" в России.

© С. Симонович, 1994.

В науке всегда очень важно точно определиться с тем, как идет развитие того или иного направления, вовремя принять решение о целесообразности координации усилий и сосредоточении ресурсов в нужной области. И занимается этим, как Вы понимаете, никто иной, а Академия Наук. "Корпус Бессмертных" (так называют академиков во Франции) как раз этим делом и занят.

У нас, в России, количество "синклеристов" уже раз в пятьдесят превосходит количество научных работников, занятых в системе Академии Наук (несколько миллионов против пятидесяти тысяч), а собственной академии мы до сих пор не имеем, хотя, надо сказать, в отличие от настоящих академиков никто из нас не жалуется на "загнивание" и "застой", отсутствие перспективных задач и государственной поддержки. Просто мы привыкли, что все надо делать собственными силами, а все финансирование выкраивается из скудного семейного бюджета и успешно осваивается на многочисленных радиорынках.

Тем не менее, отсутствие Академии Синклеристов (далее АС) не должно помешать, нам правильно оценить, на каком свете мы находимся, в каком направлении мы идем, и что нас ждет в ближайшем будущем. И тем, кто пришел в нашу систему АС совсем недавно и тем, кто придёт к нам в ближайшее время, немаловажно знать, что же привело его сюда, и на какую карьеру он здесь вправе рассчитывать.

Мы уже писали на страницах "ZX-РЕВЮ" о том, что в читательских кругах сильно возрос интерес к личности сэра Клайва Синклера и по многочисленным просьбам мы опубликовали объёмную статью о его жизненном пути в № 3 за 1994 год. Но поток писем на этом не закончился. Теперь наши читатели просят рассказать о творческом пути "ИНФОРКОМа".

Посоветовавшись, мы решили, что для нас было бы нескромно развивать эту идею, но раз люди просят, значит какой-то (не праздный) интерес все-таки есть. И мы решили, что лучше будет рассказать не о себе, а вкратце осветить те этапы, которые прошел наш общий герой - "ZX-Spectrum". По-видимому, это будет важнее и полезнее.

Малое предприятие "ИНФОРКОМ" работает над информационным обеспечением компьютера "ZX-Spectrum" с 1989 года, то есть уже почти пять лет (хотя неформально - с 1987 года). За эти годы мы получили, наверное, несколько сотен тысяч писем из разных уголков страны, что, как мы надеемся, дает нам некоторое право для того, чтобы сделать ряд обобщений и выводов, касающихся закономерностей развития "Синклеровского" движения, а также, по возможности наметить планы на будущее (хоть это и субъективно).

1 этап 1984 - 1987 г.г.

Аппаратное обеспечение.

Это был этап первичного проникновения "ZX-Spectrum" в Россию. Компьютер ввозился сотрудниками наших учреждений за границей и командированными. Туристам и экскурсантам не хватало для этого ни денег (100 - 120 долларов), ни времени на то, чтобы понять, с чем они имеют дело.

В этот период практически каждый из сотрудников госучреждений за границей, отработав свои положенные три года, возвращался в страну, имея джентльменский набор - видеомагнитофон, стиральную машину-автомат, никелированную мойку и бытовой компьютер. Впрочем, некоторые, вместо всего этого буржуазного изобилия, ввозили Главный Приз - автомобиль.

Ввозимые компьютеры очень различались по цене и способностям. Так, из Европы самые богатые заграничные ввозили, как правило, "Коммодор-64" - и графика и звук у него были побогаче. Чуть менее требовательные ввозили "Атари-800" (ранний аналог "Атари-65"). Эта машина тоже имела неплохие характеристики и могла удовлетворять тех, кто имел достаточно терпения просидеть у магнитофона 20 минут в ожидании загрузки одной программы, хотя без сбоев такая загрузка проходила редко. Самые бедные (но и самые счастливые) ввозили наш любимый "ZX-Spectrum 48", а заодно и программы к нему.

Оказавшись на родной земле, они быстро понимали, сколь удачный выбор был сделан.

Доступность программного обеспечения сделала его здесь самым популярным. Достаточно было открыть еженедельную рекламную газету с частными объявлениями, чтобы узнать номер телефона своего коллеги, продающего программы или джойстики. А, позвонив, можно было разжиться и ещё чем-то (описаниями, книгами, журналами, аппаратными премудростями).

Никто никогда не отказывал в посильных консультациях по телефону совершенно незнакомым людям! Мы сами начинали свое изучение машины с десятков телефонных звонков и свидетельствуем, что это было исключительно благородное сообщество.

В то время, как владельцы "Коммодоров" и "Атари" уныло стояли в очередях в комиссионку, жажда сдать свой аппарат и, проклиная судьбу за то, что не привезли вместо него "видик", счастливые обладатели "Спектрумов" активно налаживали связи между собой и вместе развивали общее дело. Сейчас смешно вспоминать, но в те годы появление "Спектрума" на полках любого комиссионного магазина вызывало мгновенный шквал взаимных звонков и уже через несколько часов за ним с разных концов Москвы мчались покупатели. Практически у каждого обладателя "Спектрума" всегда было три-пять заказов от друзей и знакомых на приобретение по случаю по любой цене этой чудо-машины.

К 1987-му году в стране было порядка 50 тысяч "синклеристов". Их центром на 80 процентов стала Москва, хотя кроме Москвы дело неплохо обстояло в Новосибирске, Душанбе и Ташкенте. Как ни странно, но практически "на нуле" очень долгое время (примерно до 1988-го года) оставался Ленинград. Интересно отметить, что география распространения "Спектрумов" в те годы и сейчас разительно отличаются.

Программное обеспечение.

Что же касается программного обеспечения, то в этом направлении никаких серьезных разработок не велось. Информацию о программировании в кодах мало кто имел, а программированием на БЕЙСИКе занимались только те, кому это требовалось по основной работе (расчёт электрических цепей, некоторые астрономические расчёты, как правило, связанные с необходимостью ориентации остронаправленной антенны для приёма спутникового телевидения, расчёт биоритмов, расчёт систем линейных уравнений, расчёт дифференциальных уравнений 2-го порядка, простейшие программы для своих детей).

К важнейшим достижениям этого периода надо отнести первую русификацию текстового редактора - программы TASWORD (Tasword-RL). Эта работа была выполнена в Москве доцентом МИФИ. "Спектрум" уже начал выходить за пределы чисто развлекательной тематики.

Информационное обеспечение.

Информационное обеспечение в этот период имело важное значение, но, к сожалению, было развито весьма слабо. Делу помогали несколько ведомственных переводов, сделанных в научных организациях и депонированных в центральной научной библиотеке ГПНТБ. Среди них была инструкция по работе с компьютером Виккерса и книга по ремонту и наладке компьютера.

В любительских кругах дело ограничивалось копированием и распространением тех фирменных листочков, которые обычно вкладываются в кассеты, а также ксерокопированием статей из редких номеров журнала Sinclair User.

II этап 1987 - 1990 г.г.

1987 год стал переломным в развитии "Спектрума". Одновременно сошлись несколько разных процессов, которые инициализировали друг друга и дали бурный всплеск всему делу.

Аппаратное обеспечение.

1. Тот факт, что самый "дешевый" зарубежный компьютер стал самым популярным, с гарантированным спросом, привел к тому, что начался интенсивный ввоз этих машин из-за рубежа. Их стали везти не только "для себя", но и "на продажу". Десятки тысяч наших заграничников, решающие ежегодно проблему "что во время отпуска ввезти для продажи, дабы хоть чуть-чуть компенсировать лишения, связанные с пребыванием в тяжком климате чуждых стран" всё чаще и чаще стали выбирать "Спектрум" (тем более, что в Европе в этот период шла интенсивная их распродажа по сниженным ценам в связи с тем, что К. Синклер прекратил их выпуск в 1986 году). Десятки тысяч "Спектрумов" поехали в СССР из Индии, Италии, Испании, Греции, ФРГ, Австрии.

2. Наладились устойчивые коммерческие поставки "Спектрумов" и "Таймексов" из стран

Восточной Европы - Польши, Югославии, Венгрии. Особенно много их завозилось из Польши. Кризисные явления и сопутствующая им инфляция в Польше и Югославии в 88-89 годах привели к очень выгодному для "челноков" завозу "Спектрумов" в СССР.

3. В 1987 году началось экспериментальное создание первых отечественных моделей. В начале года была "расколота" "секретная" микросхема ULA фирменной машины и в Москве и Новосибирске появились первые машины, собранные на отечественных деталях (за исключением самого процессора Z-80).

Большую роль здесь сыграл журнал "Радио". Опубликовав в 1986 году описание компьютера "РК-86", этот журнал вызвал массовое самостоятельное техническое творчество, так что к 1987-1988 годам в стране уже было немало тысяч умельцев, способных приступить к сборке "Спектрумов".

4. Значительную роль сыграл "Закон о кооперации". В 1988-1989 годах несколько десятков первых кооперативов страны приступили к поставке комплектов типа "Сделай сам" для сборки "Спектрумов". При производительности до нескольких тысяч комплектов в год, эти кооперативы позволили поднять первый интерес к новому компьютеру. Определились основные центры их производства (Москва, Череповец, Краснодар, Ростов-на-Дону, Новосибирск - 1988 г., Львов, Харьков, Ленинград, Свердловск - 1989 г.).

5. В этот же период началось производство "Спектрумов" на государственных промышленных предприятиях. Первые разговоры о возможности их промышленного производства относятся ещё к 1987 году. В качестве возможных вариантов рассматривались радиозаводы в Новосибирске и Ташкенте. Но на практике первыми начали такую работу в 1989-1990 годах заводы в Ижевске, Кишиневе, Смоленске, Чебоксарах и др.

6. Конец 1988-го, начало 1989 года ознаменовались ещё одним прорывом - наконец-то была разработана отечественная версия Бета-диск интерфейса. Схема была разработана той же группой, которая двумя годами раньше "расколола" микросхему ULA и разработала первую "Москву", а впоследствии образовала ядро широко известного в свое время НТК "Плюс".

Первое время была попытка "засекретить" новую разработку и наладить её промышленное производство. Но отсутствие средств не дало это сделать. Постепенно разработка стала известной и уже к июлю 1989 года началось штучное производство контроллеров дисководов в подмосковном Зеленограде, а чуть позже и в других городах страны.

Программное обеспечение.

Основным содержанием данного этапа в области программного обеспечения стало практически полное удовлетворение потребностей программистов во всех необходимых программных средствах. До сих пор мы пользуемся большей частью переделками тех системных и прикладных программ, которые в то время поступили из-за рубежа.

Наиболее чувствительный перелом произошел в 1987 году. Так, весной этого года в Москву была ввезена программа "Art Studio". Одной её было достаточно, чтобы "Спектрум" победил в России. Ни на каких других компьютерах в то время ничего подобного ещё не было!

Чуть позже, в июле 87-го в Россию попал редактор "The Last Word 2". Его привезла на электротехническую выставку фирма "Saga", которая выпускала расширенные клавиатуры для "Спектрума". Заинтересованная беседа у стенда фирмы привела к тому, что зарубежные представители подарили нашим коллегам авторучку. Мы ответили, подарив матрешку. Но Запад в долгу не остался и, покопавшись, извлёк откуда-то из-под прилавка кассету TLW-2 с полным описанием. Эта матрешка внесла огромный вклад в наше с Вами дело. Через месяц описание TLW-2 уже пошло по стране (примерно в том виде, в каком оно ходит и сейчас), а ещё через полгода (1988) мы сделали первую (не вполне совершенную, но работающую) русскоязычную адаптацию TLW-2.

Большую роль сыграл образовавшийся в начале 1987-го года московский кооператив "ТАКТ". Он стал первой в стране организацией, взявшейся за поддержку "синклеристов". Услуги по ремонту фирменных машин, продажа джойстиков и интерфейсов для подключения принтера, а также консультации начинающих быстро сделали его популярным. К сожалению, эта фирма недолго проработала на благо "Спектрума", и года через полтора перешла на гораздо более выгодную торговлю IBM PC. Зато она успела распространить несколько языков

программирования (в частности Паскаль HP4T), ту же самую "Art Studio", базу данных "Masterfile 09" и множество игровых программ.

Информационное обеспечение.

В отличие от первого этапа (1984-1987 г.г.) в этот период распространение программного обеспечения, как правило, сопровождалось соответствующим распространением и информационного обеспечения. Во всяком случае, первоочередные потребности программистов в системных и прикладных программах были удовлетворены в достаточной степени. В стране созрели все предпосылки для того, чтобы начинать работы над собственным программным обеспечением.

Значительный вклад в развитие процесса внёс всё тот же кооператив "ТАКТ", подготовивший перевод первой серьезной книги для программирующих в машинных кодах - "Бейсик и программирование в машинных кодах" (Ян Логан). При том, что книгу переводили наёмные переводчики, не знавшие компьютера, при том, что она распространялась в перепечатках на пишущей машинке, при многих сотнях ошибок, это все-таки был значительный шаг, сделавший доступным для многих машинный код и системное ПЗУ. До сих пор экземпляры той несовершенной редакции циркулируют по стране и помогают многим. Мы начинали свой путь в "Спектрум" тоже с изучения этой книги. К этому же периоду относятся появление в России книги Яна Логана и Фрэнка О'Хары The Complete Spectrum ROM-Disassembly и монографий Лэнса Левенталя по процессору Z-80.

III этап 1990 - 1993 г.г.

Этот этап характерен бурным распространением "Спектрума". К 1992-ому году компьютер выпускался уже в десятках городов страны под множеством самых разнообразных названий. Совокупное производство "Спектрумов" в год приблизилось к миллиону машин, причем случаев залеживания на прилавках торговых организаций не отмечалось. После 1991-го года характерным стало широкое распространение дисковой версии компьютера, в 1992-ом году наметился повсеместный переход на 128-ые модели, а в 1993-ем - широкое внедрение музыкального сопроцессора.

В области программного обеспечения появились первые признаки того, что в России есть кадры, способные выпускать собственные программы, не уступающие западным, а в области информационного обеспечения вместе с нами уже работают десятки других организаций в различных городах страны.

Мы не будем останавливаться на этом этапе более подробно, поскольку наши читатели и сами прекрасно знают состояние дел в своих регионах и могут оценить основное содержание данного этапа не хуже нас.

IV этап 1994 - ???? г.г.

Проблемы и противоречия.

По нашим данным, мы вступаем в новый этап развития "Спектрума". И у этого этапа есть свои характерные особенности, вытекающие из объективно возникающих противоречий.

1. Первое противоречие, с которым нам придется иметь дело, состоит в том, что сейчас, когда "Спектрум" находится в высшей точке популярности в России, он попадает в полосу конкуренции. Все годы своего развития у нас этот компьютер был вне конкуренции и не знал, что это такое. Слабая конкуренция со стороны "Коммодора", "Атари" и "MSX" была подавлена ещё в 1987 году и о ней можно не вспоминать. О конкуренции со стороны отечественных "Микрош" и "БК-0010" никто никогда и не говорил серьезно. Фактически с конкуренцией нам приходится иметь дело впервые. Положение осложняется тем, что конкуренты наступают сразу с трёх направлений.

Развернувшаяся на страницах наших изданий (а также на всех отечественных рынках) дискуссия на тему о конкуренции со стороны "АМИГИ" и IBM PC отражает реальное состояние дел, свидетельствует об определенном росте благосостояния программистов и пользователей, но не является в перспективе сколько-нибудь серьёзной проблемой. У "Спектрума" достаточно преимуществ в его дружественности, доступности, распространенности и самообеспеченности, чтобы удержать свою нишу и стать полупрофессиональной машиной, способной длительное время удовлетворять потребности тех, кто хочет ТВОРИТЬ что-то важное и нужное своими руками.

Гораздо более серьезную тревогу могут вызывать так называемые "видеоприставки". Если Вы знакомы с тем бумом, который вызвали в последние два года "Денди" и SEGA, то чтобы представить, что будет в ближайшие годы, надо примерно умножить его на 10...20.

В смертельной схватке в мире схлестнулись два удава - SEGA и NINTENDO. К сегодняшнему дню ими выпущены примерно до полутора десятков разных систем от 8- до 64-разрядных. Они заполнили весь земной шар, кроме России и СНГ. В Америке соотношение сил этих гигантов примерно 50:50. В Европе сложилось устойчивое преимущество в пользу SEGA, в странах Азии Nintendo вне конкуренции. Но последняя схватка ещё впереди. Впереди битва за Россию.

По примитивным характеристикам "Денди" и шестнадцатиразрядных SEGA, циркулирующих на наших рынках, Вам сейчас, может быть, трудно представить, что готовится к броску в Россию, но это действительно будет бум, который превзойдет по своим масштабам все потребительские бумы, известные нам до сих пор. Для последних моделей существуют игры всех возможных жанров от аркадных и логических до графических адвентюр. Активно прогрессируют даже ёмкие RPG, "съедающие" по 4-8 Мб оперативной памяти (зарубежные производители приставок в последний год усиленно "нажимают" на этот жанр). Оснащенные лазерными дисководами модели уже не являются просто "видеоиграми" – это уже видеоинформационные системы, способные хранить на одном диске информацию в объёме тома приличной энциклопедии вместе с графикой, музыкой и видеофрагментами, что они, кстати, уже и делают.

Невзирая на сложность новой техники, все это будет брошено по минимальным ценам. Гиганты смогут себе это позволить (впоследствии они многократно все вернут, торгуя картриджами и лазерными дисками).

А ведь кроме SEGA и Nintendo к прыжку изготовилась и фирма "Атари" со своей новейшей приставкой "Ягуар". Не остается в стороне и "Commodore", недавно выпустивший на базе "АМИГИ" приставку "Amiga-CD 32" (очень вероятно, что это будет самая популярная машина в разряде приставок. За ней стоит интересная философия и её поддерживает серьезные зарубежные издания).

По самым скромным оценкам в Россию будет ввезено в течение нескольких ближайших лет до двадцати миллионов всевозможных видеоигровых систем. Это оборот в десятки миллиардов долларов вместе с картриджами и компакт-дисками. Сравните это с расходами на оборону (!!!).

У "Спектрума" есть только одно преимущество по сравнению с этим морем видеоприставок - он полноценный компьютер и до тех пор, пока в Вас живет желание творить, работать, создавать что-то свое и расти вместе с Вашей машиной, Вы не променяете его на видеоприставку, не требующую мозгов. Но... ведь могут не подойти новые поколения, оттянутые от творческой работы со "Спектрумом" многочасовыми игрушками с приставками.

Выжить в этих условиях "Спектрум" сможет, только если предложит молодежи радость творчества, которую не даёт ни видеоприставка, ни "АМИГА", ни IBM PC.

Срочно нужны простые и доступные программные системы, с помощью которых даже начинающий смог бы реализовать свой потенциал. Потребность в самореализации - огромный стимул. Только с его помощью можно выстоять против напора видеоприставок. На раскачку осталось года два, не больше.

2. Второе противоречие состоит в том, что сейчас, когда "Спектрум" в России достиг вершины славы, почти полностью прекратился приток новых программ для него из-за рубежа. В залоге успеха "Спектрума" всегда было изобилие программного обеспечения. Можно предположить, какую угрозу несёт прекращение новых поступлений.

Единственный способ выстоять в этих условиях - срочно развернуть массовое производство в России программного обеспечения. Хотелось бы сказать "качественного" программного обеспечения, но для первого этапа это даже не столь важно. На каждую "некачественную" программу найдется десяток программистов, которые скажут: "Я могу сделать лучше и сделаю это!" А это как раз и есть самое важное.

3. Третье противоречие связано со вторым. Если стране требуется быстрый разворот производства программного обеспечения, то должны появиться аппаратные и программные средства, позволяющие ускорить этот процесс. Существенный шаг в 1993 году был сделан

фирмами "IskraSoft" и "Slot", выпустившими операционную систему iS-DOS и математическое обеспечение под неё.

Но дело идет крайне медленно. Сдерживание "форматтера", информации по точкам входа в iS-DOS, "засекречивание" новых проектов, конечно, немного защищает производителей от несанкционированного копирования и распространения системы. Но если и сейчас, после более чем года её распространения, эти фирмы будут продолжать "игры в секреты", то через пару лет им уже не от кого будет скрываться. Останется перейти на торговлю линолеумом.

* * *

В заключение отметим, что не все так плохо, как могло бы показаться пессимистам, прочитавшим последний раздел. В человеческой природе есть несколько особенностей, которые могут быть успешно использованы для поддержания и развития "синклеровского" движения даже тогда, когда бум видеоприставок захлестнет страну. Это:

- ✓ потребность в новых знаниях (говоря по бытовому - любопытство);
- ✓ потребность в контактах;
- ✓ потребность в самореализации и самоутверждении.

Вот на удовлетворении этих трех потребностей и должна строиться стратегия развития "Спектрума" в новых условиях. А как к этому подойти, читайте в следующих статьях.

Обучающее программное обеспечение.

© Сергей Симонович, 1994.

В последние полтора года наши читатели активно поднимают вопрос о необходимости развития обучающего программного обеспечения на базе "Спектрума". Проблема не нова. Мы сами с этого начинали лет шесть назад, но тогда эти идеи мало кому были нужны. Впоследствии не раз ещё делались попытки вернуться к этой теме ("Спектрум в школе" всегда был первым разделом в ZX-РЕВЮ), но вопрос всё равно вяз. Сегодня уже не мы его поднимаем, об этом говорят сотни читателей в своих письмах и, поверьте, это не случайно.

Если "Спектрум" входит в полосу конкурентного развития и будет жить "от творчества" масс, то надо подумать и о том поле, на котором это творчество будет происходить. Клайв Синклер "сделал" карьеру "Спектруму", а сам получил титул лорда на том, что оснастил компьютером тысячи английских школ, то есть, удачно сыграл на поле народного образования. И здесь есть над, чем призадуматься.

Нам известны всего лишь девять разных жанров игровых программ. В каждом из них кто-то уже сказал свое весомое слово и очень трудно придумать что-то принципиально новое. Можно, конечно, предположить, что когда-то кто-то выпустит ELITE-35 или DIZZY-49, но все равно логический предел есть, и он не за горами.

Совсем иначе дело обстоит в обучающем программировании. Это поле неисчерпаемо, да и не может быть исчерпано. Если нации уже нечему будет учиться, то... Одним словом, один только список школьных дисциплин во много раз превосходит список жанров игровых программ. А сколько в каждой дисциплине разделов и тем?! А сколько учебников написано по каждой теме?! А почему останавливаться только на школьных проблемах? Существуют ведь вузы и техникумы. А ведь учиться можно не только научным дисциплинам. Сколько вопросов и тем не связаны с наукой, а относятся к культуре и искусству!

Одним словом, если есть неисчерпаемое поле для деятельности, к тому же такое, для подъема которого не надо собирать команды по тридцать человек, а работать в одиночку или в небольшой группе – то это только обучающие программы.

Открытым, правда, остается вопрос, насколько они кому-либо нужны, и будет ли на них спрос? Но это уже зависит от качества программы, от того, насколько талантливо она сделана и что в нее заложено.

Данная статья предназначена как раз для тех, кто хочет попробовать свои силы в этой области. Сразу скажем, что, несмотря на то, что мы интересуемся этим вопросом уже не первый год, все-таки фактического материала у нас маловато. Очень это дело неразвито не только у нас, но и за рубежом. Неслучайно, кстати, многие зарубежные фирмы заказывают программы этого класса нашим российским программистам. Тем не менее, мы постараемся разобраться с некоторыми проблемами, которые возникают при создании обучающих программ.

Тему считаем незаконченной, а только открытой. С огромной радостью примем соображения наших читателей, развивающих эту тему дальше (реализуйте свою потребность в контактах).

1. Опыт классификации обучающих программ.

1.1. Информационная программа.

Концептуально это самый простой вид обучающей программы. Гладкий текст, переписанный из учебника, который воспроизводится на экране по нажатию клавиши (возможно, что и с иллюстрациями) - вот самый негодный пример того, что может быть сделано в этом жанре.

В "простоте" как раз и заключается вся сложность. Как сделать этот текст не просто "читаемым", а ещё и читаемым с увлечением, как сделать, чтобы компьютер был более удобен, чем книга - в этом-то вся проблема и состоит.

Генеральное преимущество компьютера перед книгой - в его интерактивности. То есть, в ответ на разные реакции пользователя книга всегда даст одно и то же, а компьютер должен реагировать по-разному. Как использовать интерактивность в простой информационной программе - решать Вам. Самое простое, что можно здесь предложить - это организовать экранное меню (это уже интерактивность) и, конечно же, время от времени подкидывать пользователю

конкретные вопросы, а по результатам ответов соответственно реагировать.

До последнего времени мы не знали ни одного удачного образца подобной информационной программы для "Спектрума", но сейчас такой образец появился - это последние выпуски журнала "Спектрофон" (авторы Матвеев Ю.А. и Шишлянников С.Я.). Правда, "Спектрофон" - это не обучающая программа, но если информационную часть в нём сделать обучающей, то это будет достойный образец.

Многочисленные шрифты, каждый из которых несет свою органическую нагрузку, умелое управление цветом для выделения нужных мест, звуковое сопровождение, настраивающее на новую тему, удачное инкорпорирование спрайтов в тексте, не перегруженная структура меню, возможность выйти из информационного блока в заранее подготовленную независимую программу и многое-многое другое из оболочки этого журнала, посвященного игровым программам, может быть использовано в качестве пожеланий разработчикам будущих информационно-обучающих систем.

А то, что такие разработчики найдутся, можно быть уверенными. Рынки страны завалены многочисленными демо-роликами, в которых под музыкальное сопровождение при отличной анимации передаются килобайты поздравлений, пожеланий и всякой прочей очень "важной" информации. Впрочем, эти демо-ролики свою задачу исполняют блестяще. Они должны быть демонстрационными, вот и демонстрируют, что в стране есть множество способных программистов, способных на решение подобных задач. Заодно, кстати, они выполняют очень важную "синклер-функцию" (удовлетворяют потребность в самореализации и самоутверждении, что для молодых людей, конечно, необходимо).

1.2. Моделирующая программа.

Это несложный, но очень важный класс обучающих программ. Здесь абсолютно необходима графика и анимация. Суть программы состоит в том, что на экране моделируется некоторый физический процесс или явление.

Классический пример - бросание камня под углом к горизонту. Графически на экране можно показать, как зависит длина полета камня от угла бросания.

Другой пример - моделирование ситуации при обгоне на дороге (см. рис. 1). Автомобиль А должен обогнать автомобиль В, едущий со скоростью V_1 . Им навстречу едет автомобиль С со скоростью V_2 . Должны также учитываться такие параметры, как расстояние L и длина обгоняемого автомобиля L_b . Пользователь должен ввести скорость обгона (V_a) и программа покажет ему, что с ним будет в данной ситуации.

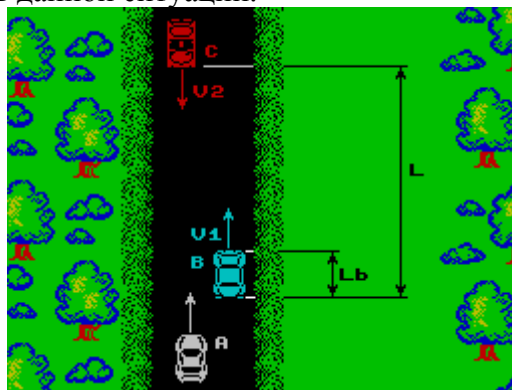


Рис. 1

Звук, музыка, скрип тормозов, звон разбитого стекла - все должно быть использовано, чтобы отучить будущего водителя обгонять с выездом на полосу встречного движения без достаточных оснований, а заодно побудить попробовать свои силы в программе ещё разок.

Моделировать можно что угодно. Очень интересно моделировать колебательные процессы, например колебания маятника, пружины, уровня жидкости в сообщающихся сосудах, заряда конденсатора в колебательном контуре и т.п. Возможность пользователя изменять параметры системы по ходу процесса, (например, менять сопротивление в колебательном контуре) делает программы этого класса интерактивными. Это отличная замена лабораторным работам, особенно в учебных заведениях, не располагающих для этого достаточной технической базой.

1.3. Демонстрационная программа.

По своей сути она не очень отличается от моделирующей, здесь тоже надо средствами графики и анимации показать процесс или явление, но поскольку речь идет только о демонстрации, она может быть не интерактивной, то есть не ждёт от пользователя никаких вводов данных, а просто показывает заранее заложенный сценарий (например, работу четырехтактного двигателя внутреннего сгорания, ядерного реактора, токарного станка...).

Поскольку демонстрация здесь самый главный момент, программа обязательно должна работать красиво. Если в моделирующей программе можно допустить, чтобы автомобили были кубиками, то здесь это абсолютно не проходит.

1.4. Программа-экзаменатор.

Программы этого типа знакомы всем, кто когда-нибудь сдавал экзамены в ГАИ. Программа содержит блок вопросов (по возможности сопровождающихся иллюстрациями) и заранее определенный перечень готовых ответов, только один из которых правильный.

Программы этого типа хорошо подходят практически для любых наук и дисциплин. Разновидностью подобных программ могут быть программы по арифметике "решить пример", по русскому языку "вставить пропущенное слово", по физике "определить размерность" и многие другие.

2. Поощрять или наказывать?

Большая проблема в обучающем программировании связана с системой поощрений и наказаний. Как поощрить пользователя за правильный ответ? Наказывать или не наказывать за неправильный, а если наказывать, то как? Последнее слово в этой проблеме ещё до конца не сказано, и у каждого из наших читателей есть необъятное поле для экспериментов.

Но, прежде чем говорить о поощрениях за правильный ответ, нам надо ещё определиться со стимулированием учебной активности. Если у пользователя нет стимула для того, чтобы добраться хотя бы до середины программы, то все Ваши ухищрения и изобретения так, и останутся неоцененными и неузнанными.

Наибольшую трудность в смысле стимулирования представляют программы 1-го типа (информационные программы). Само собой разумеется, что если текст посвящен чему-то очень интересному (как, например, в журнале "Спектрофон"), то само содержание уже является достаточным стимулом для того, чтобы читатель двигался вперед. Ну, а если программа посвящена такой скучной теме, как "Распределение трудовых ресурсов по климатическим поясам?".

В этом случае Вы должны использовать сильный стимул, который можно выразить словами: "А что же дальше?". Приведем пример. Если при рассказе о зонах пустынь и степей в Вашей программе будет звучать музыка, характерная для этих регионов, то у пользователя невольно появится желание дойти до описания трудовых ресурсов в тундре, дабы послушать, какая музыка будет там.

То же самое касается и графики, особенно анимации. Она должна быть использована для стимулирования желания идти дальше. При этом неважно, что Ваша анимация, может быть, простой и бесхитростной (дым от костра, движущиеся облака и пр.). Опыт показывает, что компьютерная анимация при любой простоте является очень выразительной.

Но и статическая графика тоже является важным элементом. Достаточно сделать разные фоновые экраны для разных разделов программы, и у пользователя появится желание, как можно быстрее просмотреть все темы. В качестве барьера можно поставить контрольные вопросы в конце каждой темы. Здесь есть хорошо знакомый пример - программа Bomb Jack. Главный герой игры собирает "бомбы" и получает за это очки. Различным уровням игры соответствуют различные фоновые картинки (от древнеегипетских пирамид до фантастических городов будущего). Горячее желание узнать, что будет дальше, стимулирует пользователя к тому, чтобы приложить максимум возможных усилий для перехода на новый уровень.

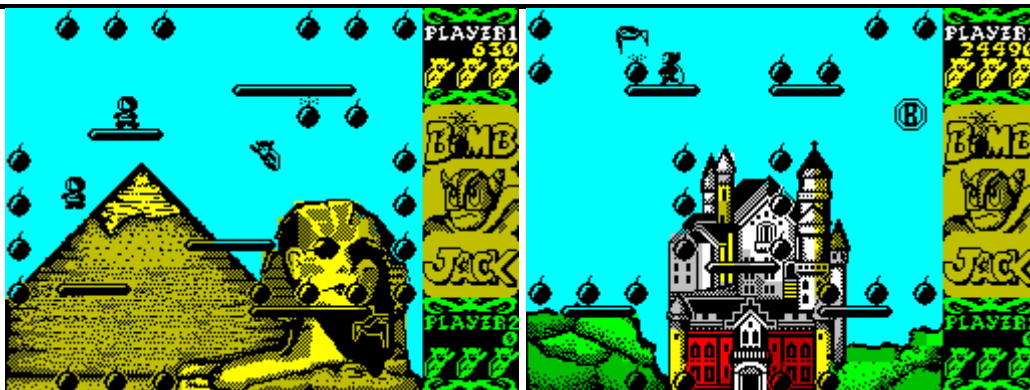


Рис. 2,3.

На этом же строится и система поощрений. Фактически, у Вас есть только три системы поощрений: текстовым сообщением, графикой, музыкой. И все эти средства должны вызывать желание узнать, "что же дальше?". Так, например, получив, пять раз подряд сообщение "Отлично! Молодец!" пользователь может потерять интерес к работе, понимая, что ничего нового ему эта программа не сулит.

Достаточно широко в обучающих программах применяется система поощрений в виде постепенно проявляющейся или постепенно открывающейся картинке. Можно усложнить игру, "выдавая" за правильные решения какие-то фрагменты рисунка, из которых потом можно будет сложить красивую картинку (вознаграждать фрагментом за правильные ответы, а игрой по их выкладыванию - за пройденную тему).

Здесь есть характерная ошибка, нередко встречающаяся в обучающих программах, в которых пользователь имеет крайне ограниченное время на ответы. Естественный стресс, который здесь возникает, совсем не располагает к тому, чтобы получать удовольствие от созерцания подобных "наград". Либо пользователь не обращает на них внимания, либо они его раздражают, сбивая с ритма. В остродинамичных ситуациях лучше "поощрять" звуком, а не графикой, оставляя её на итоговое поощрение по окончании раздела (темы, блока).

Теперь о системе наказаний. Как наказывать, да и надо ли это делать - большая проблема. В игровых программах в большинстве случаев система "наказаний" присутствует. Это потеря очередной "жизни" с последующим возвратом к началу игры (или к началу уровня). Так получается, что если в качестве "поощрения" пользователю дают что-то новенькое, интересное и по возможности неожиданное, то в качестве "наказания" его вынуждают заниматься чем-то известным, пройденным и уже неинтересным. Покопавшись в своей библиотеке игровых программ, Вы легко выделите те игры, в которых система наказаний является "гуманной" и дружественной и в которых система наказаний неоправданно "жестока" и нецелесообразна. Самое жестокое наказание, с которым Вам приходится сталкиваться в игровых программах, относится к кассетным играм с многочисленными догружаемыми уровнями, в которых после "гибели" на N-ном уровне Вам приходится перематывать ленту на начало, перезагружать первый уровень и проходить всё заново с новыми подзагрузками. Такие игры не пользуются устойчивым спросом - их создатели явно "перемудрили".

Таким образом, "наказанием" в обучающей компьютерной программе должна быть необходимость повторения каких-то процессов, уже не являющихся для пользователя новыми. И можно даже вывести эмпирическую зависимость: "Чем серьезнее провинность, тем более длительным (по времени) должен быть повторяющийся блок". Поскольку "повторение - мать учения", в обучающих программах этим можно отлично пользоваться для закрепления уже пройденного материала.

Понятие "длительным", однако, здесь не всегда совпадает с понятием "скучным". Например, если пользователь не решил примера ($7*8=?$), то можно предложить ему "расстрелять" несколько воздушных шариков, на которых записан правильный ответ и чем чаще встречается подобная ошибка, тем "шариков" может быть больше. При этом всегда следует иметь в виду два важных момента:

- ✓ выдавая правильный ответ, всегда надо напоминать сам вопрос (увлекаясь "охотой за шариками", играющий забывает, к чему они относятся);

✓ следует помнить, что во время активных действий (работа с клавиатурой или джойстиком) поле зрения играющего сужается непосредственно до места положения курсора и он "не видит" никакую ин формацию, если она помещена в стороне (см. рис. 4).

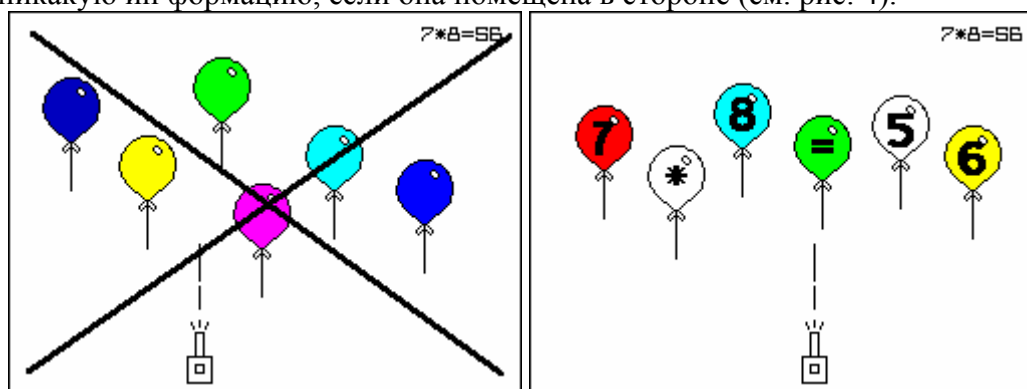


Рис. 4.

В заключение этого раздела мы призываем всех наших читателей к исследованию возможных способов "наказаний" и "поощрений" в обучающих программах. Тема практически неисчерпаема и каждый, независимо от стажа работы с компьютером и от личного опыта, может внести свой вклад в общее исследование.

3. Некоторые методические приемы.

Методические приемы, заложенные в основу будущей обучающей программы (игры) могут быть крайне многообразны. Их могут быть тысячи, хотя никто до сих пор не пробовал их обобщить и проанализировать. Это опять же огромное поле для Вашей самостоятельной деятельности и для Вашей фантазии. В качестве же первого примера мы остановимся здесь только на некоторых.

Самым важным, что объединяет любые приёмы, является принцип интерактивности. Поскольку интерактивность - основное свойство, отличающее компьютерную обучающую программу от книги или учебника, то при разработке любых своих новых приемов следует всегда спрашивать себя: "А где здесь интерактивность?". То есть, может пользователь своими действиями повлиять на ход развития сюжета или нет? Если он этого сделать не в состоянии, лучше такую программу не выпускать, а вместо нее написать методическое пособие, распечатать его на принтере и раздать ученикам.

3.1. Иерархическая структура.

Программное меню - это уже элемент интерактивности и этим надо уметь пользоваться. Иерархическая структура программы не должна быть ни слишком простой, ни слишком сложной. Желательно не допускать глубокой вложенности меню (более трех уровней). В играх это не очень важно, но в обучающих программах пользователь должен быть в состоянии удержать эту структуру в уме без особого напряжения. Это способствует созданию и закреплению цельного видения всей темы. В образовании крайне важно не только хорошо знать суть того или иного вопроса, но и хорошо представлять хотя бы в общих чертах, как он увязан с другими смежными разделами. Так, например, изучая в физике тему "Работа", ученик должен понимать, как к ней относятся темы "Сила", "Энергия", "Единицы измерения" и т.п.

Из тех же соображений целостности подхода должны быть ограничены и размеры каждого из вложенных меню. Они не должны превышать 7 пунктов. Единственный случай, когда их может быть больше (до 10) - при включении в меню служебных (квазистандартных) функций типа LOAD, SAVE, SOUND ON/OFF и т.п.

3.2. Интерактивная графика.

Это очень сильный, но редко применяющийся в силу естественной сложности прием. Среди обучающих программ для "Спектрума" мы не знаем известных примеров, но в игровых программах он применяется очень широко в аркадных, традиционных, логических и других играх (практически всегда в аркадных адвентюрах есть интерактивная графика).

Приведем пример из программы "Locomotion". На экране в беспорядке "разбросаны" фишки, представляющие фрагменты рельсового пути локомотива. Передвигая, их Вы должны собрать непрерывный путь, по которому локомотив пройдет из пункта А в пункт В без аварий.

Можно предложить аналогичную схему для лабораторных работ по физике. Выбирая из меню стандартные элементы (резисторы, элементы питания, проводники, выключатели, лампочки, амперметры, вольтметры и т.п.), пользователь создает на экране свою электрическую цепь. Сканируя экран, программа заполняет соответствующий массив в оперативной памяти, а затем по заранее разработанным процедурам вычисляет суммарное сопротивление цепи, значения тока в ветвях и т.п. Любые изменения в графике на экране (перестановки элементов) приводят к изменениям в массиве данных и отражаются на результатах расчета (рис.5).

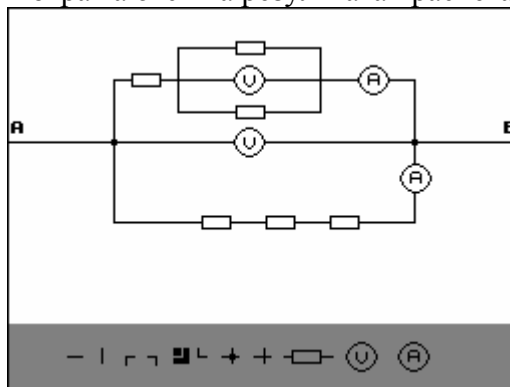


Рис. 5.

Способов использования интерактивной графики может быть сколько угодно и это один из наиболее прогрессивных путей создания обучающих программ.

3.3. Контекстные приемы.

Ключевые слова в обучающей программе полезно выделять цветом (и/или шрифтом) и создавать для них свой программный словарь (строковый массив) терминов. Он может пригодиться. Как в адвентюрных играх программа "понимает" слова из заранее заготовленного словаря, так обучающая программа должна "понимать" ключевые слова, относящиеся к данному уроку. Так, например, в описательной части программы, посвященной двигателю внутреннего сгорания, можно сказать, что его основными элементами являются ЦИЛИНДР, ПОРШЕНЬ, ШАТУН, КЛАПАН ВПУСКНОЙ, КЛАПАН ВЫПУСКНОЙ и пр. Выделенные слова являются ключевыми. Далее, по ходу описания, можно предлагать пользователю самому вставлять пропущенные в тексте слова, выбирая их из ключевых. При этом даже такой рутинный процесс, как чтение, делается интерактивным.

Периодически можно тестировать усвоение прочитанного материала с помощью иллюстраций, в которых пользователь должен сам проставить названия узлов и деталей из того же словаря.

Наличие словаря "ключевых слов" и выражений позволяет организовывать контекстный поиск, контекстные подсказки и даже гипертекстные структуры. Если у Вас есть такой словарь, то Вы в состоянии организовать сканирование экрана по "горячей клавише" в поисках ключевых слов. "Горячую клавишу" на БЕЙСИКе несложно организовать с помощью INKEY\$, а из машинного кода можно использовать как процедуры, опрашивающие клавиатуру, так и с помощью прерываний 2-го рода (правильно работают не на всех отечественных моделях).

Наиболее широко контекстный поиск применим в программах, обучающих языкам (по одной клавише можно выдавать перевод слова, на котором установлен курсор на экране, а по другой, например транскрипцию.) Контекстный поиск может быть также использован в любых других программах для выдачи определений, справочной информации и даже графики. Так, например, если Ваша программа посвящена русской литературе XIX века и А.С.Пушкин включен в контекстный словарь ключевых слов, то при нажатии на "горячую клавишу" и установке курсора на это ключевое слово, программа может выдавать на экран краткую справку о творчестве великого поэта и даже показывать его портрет. Предыдущий экран необходимо сохранять в памяти, чтобы, выйдя из "справки", вернуться в него без проблем.

4. Проблемы ближайшей перспективы.

4.1. Организация структур данных.

Основной проблемой на ближайшую перспективу является необходимость наличия автоматических средств для создания обучающих программ, иначе широкое производство

отечественного программного обеспечения в сжатые сроки не развернуть. К счастью, широкое распространение по России дисковых систем, работающих под системами TR-DOS и iS-DOS, позволяет решить эту проблему.

Главным направлением Ваших исследований на ближайший период должно стать развитие технологий, используемых для работы с базами данных применительно к обучающим программам. Принципиально структура данных в обучающей программе ничем не отличается от базы данных. Можно считать, что обучающая программа - это БАЗА ЗНАНИЙ.

Структура файла базы данных включает такие понятия, как ПОЛЯ и ЗАПИСИ и имеет вид прямоугольной матрицы.

	ПОЛЕ 1	ПОЛЕ 2	ПОЛЕ 3	ПОЛЕ N
ЗАПИСЬ 1					
ЗАПИСЬ 2					
.....					
ЗАПИСЬ M					

Мы не будем здесь останавливаться на объяснении технологии организации данных в такую структуру. Этому посвящено большое количество книг и публикаций в журналах. Рассмотрим только, какие поля могут Вам потребоваться в обучающих программах.

Символьное поле.

В этом поле может быть записана любая текстовая информация. Длина поля (в байтах) равна необходимой Вам максимальной длине записи.

Целочисленное поле.

Поле для записи целых чисел от 0 до 65535. Длина - 2 байта.

Числовое поле.

Поле для записи чисел с плавающей точкой. Длина такого поля должна быть равной пяти байтам, поскольку в "Спектруме" действительные числа представляются в пятибайтной форме.

Поле даты.

Для записи календарной даты общепринятым считается использование 8-ми байтов. Рекомендованным форматом может быть, например ДД.ММ.ГГ (например 25.03.94). В качестве разделителя мы использовали точку (.), хотя иногда применяют "слэш" (/) или двоеточие (:).

Поле даты практически повсеместно применяется в базах данных, предназначенных для ведения какого-либо учёта (складского, регистрационного и пр.). На первый взгляд, в нем нет необходимости для обучающих программ, но это не так. Существует целый спектр проектов, основанных на том, что программа сама оценивает способности учащегося по результатам предыдущих сеансов (они записаны в протокольном файле на диске) и может оценить динамику его успехов, если даты всех "успехов" и "неудач" ей известны. Так программа автоматически настраивается на уровень пользователя и демонстрирует искусственный интеллект. В свое время (1991г.) мы успешно распространяли первую выполненную в таком стиле программу обучения английскому языку DINEX (Динамический Экзаменатор).

Поле переменной длины.

Если ожидается, что символьное поле будет иметь очень большую длину, причём в разных записях эта длина будет сильно варьироваться, целесообразно сделать его полем переменной длины (мемо-поле). Несмотря на свое название, это поле может иметь вполне фиксированную длину, например для системы TRDOS - 8 байтов, а для iS-DOS - 12 байтов. В нём может содержаться только имя файла, в котором на самом деле и хранится текстовая информация. Этот файл может иметь любую длину и при нем есть небольшой указательный (индексный файл), с помощью которого для любого номера записи будет быстро найдена и выбрана информация из основного файла.

Графическое поле.

Хранить графику в базе данных, разумеется, нецелесообразно. И здесь надо поступать так же, как и для поля переменной длины. Достаточно хранить номер экрана (один или два байта). Поскольку графические экраны имеют стандартную длину, то отпадает необходимость в индексном файле. По номеру экрана он будет найден в основном файле и выдан на дисплей. Если экраны хранятся в сжатом виде, то придется либо создавать сопутствующий

индексный файл, либо разделять экраны друг от друга с помощью маркеров. Хотя использование индексного файла - безусловно, более грамотный и быстроработающий приём.

Поле спрайтов.

Если Вы хотите, чтобы в текст Вашей программы автоматически имплантировались графические шаблоны, то можно предусмотреть и поле спрайтов. Длина этого поля может быть равна одному байту, если Вы используете не более 255 разных спрайтов. Все спрайты могут храниться в одном файле и извлекаться оттуда по номеру (если их длины одинаковы) или с помощью индексного файла, если их длины различаются.

Для себя (или централизованно в виде квазистандарта) следует решить вопрос о том, в каком формате хранятся спрайты. Как хранится информация о координатах левого верхнего угла, о ширине и высоте, о графических атрибутах.

Логическое поле.

Логические поля имеют длину в один байт, который может иметь только два значения: 0 или 1, T(true) или F(false), Y(es) или N(o). В программе используются как флаги. Так, например, если пользователь уже решил данную задачу (обработал N-ную ЗАПИСЬ), то в логическом поле нужно поставить 1 и программа больше не предложит ему эту задачу для решения (или не предложит её сегодня, но может предложить завтра, сверившись с календарной датой по полю ДАТА).

Можно при первом запуске программы предварительно занулять все логические поля, но можно этого и не делать, тогда программа сможет "знать" предысторию того, как Вы с ней работали в прошлом.

Логические поля удобны для отгрузки состояния программы на диск. Во многих случаях достаточно отгрузить только логическое поле под именем, заданным пользователем. Продолжая программу на следующий день, пользователь может подгрузить логическое поле (вернуться к ранее отложенному состоянию) или начать все сначала. Правда, на практике отгрузки ТОЛЬКО логического поля бывает недостаточно. Для полного сохранения состояния программы нужно отгрузить ещё и все программные переменные, и текущий образ экрана.

Вычисляемые поля.

В своей программе Вы можете связывать свои поля с помощью каких-то математических, логических и других отношений. При этом у Вас могут быть поля, которые при создании программы заполнять и не надо. Они заполнятся автоматически по программе после начала работы.

Такими соотношениям могут быть, например:

ПОЛЕ 3 = ПОЛЕ 2 * ПОЛЕ 1

ПОЛЕ 3 = ПОЛЕ 2 / ПОЛЕ 1 - ПОЛЕ 4

ПОЛЕ 3 = (1 ТО 3) ПОЛЕ 2

Здесь под (1 ТО 3) мы понимаем "вырезание" первых трёх символов из содержимого символьного ПОЛЯ 2 и помещение результата в ПОЛЕ 3. Таким образом, вычисляемые поля могут быть не только числовыми, но и символьными и любыми другими.

Прочие поля.

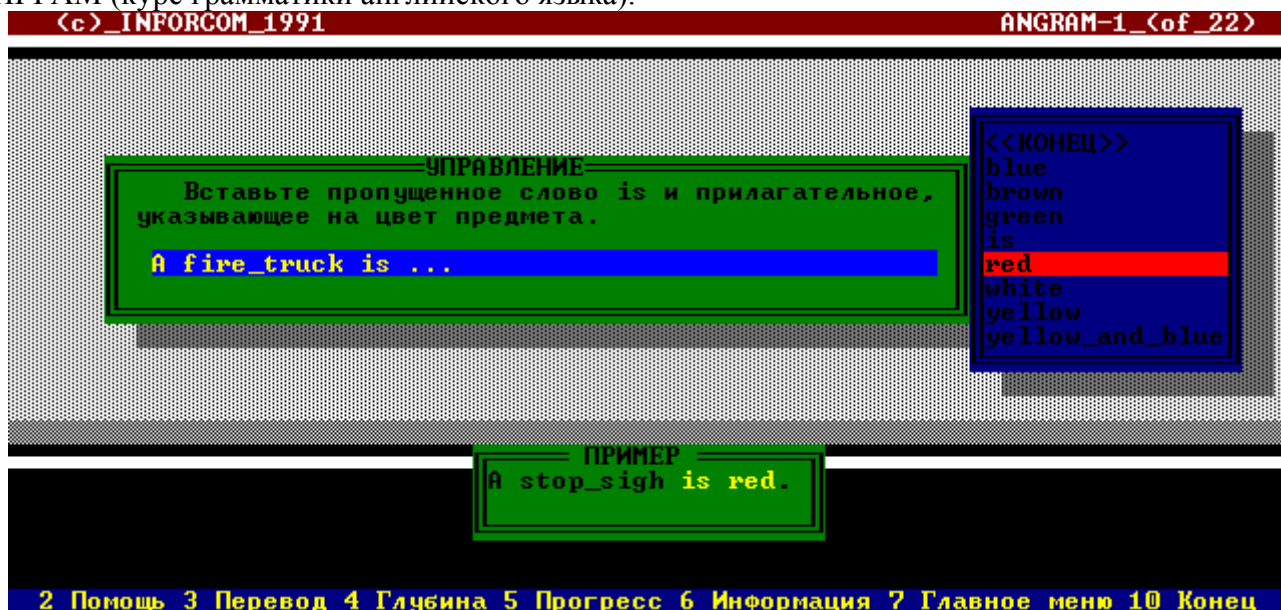
Возможность создания в базе полей любых других типов - дело Вашей фантазии.

Пример организации данных в обучающей программе.

ПОЛЕ	Тип	Длина	Назначение
1	Симв	100	Описание задачи
2	Граф	1	Номер иллюстрации
3	Симв	30	Формулировка вопроса
4	Симв	50	Демонстрационный пример
5	Числ	5	Правильный ответ
6	Лог	1	Контрольный флаг (задача уже решена)
7	Симв	30	Подсказка 1-го уровня при неверном решении
8	Лог	1	Контрольный флаг (один неверный ответ)
9	Симв	30	Подсказка 2-го уровня при повторном неверном ответе
10	Лог	1	Контрольный флаг (два неверных ответа)

11	Целое	2	Адрес процедуры, запускаемой при решении задачи (поощрение)
12	Целое	2	Адрес процедуры, запускаемой при не решении задачи с трех попыток (наказание)
13	Дата	8	Дата проведения сеанса

В качестве примера ниже приведен фрагмент экрана из нашего программного комплекса АНГРАМ (курс грамматики английского языка).



4.2. Над чем стоит поработать нашим читателям.

Несмотря на то, что "Спектруму" в России уже не менее десяти лет, существует ещё немало вопросов, которыми никто глубоко не занимался. Каждый может внести свой посильный вклад в общее дело. Вот прикидочный (далеко не полный) список вопросов, которыми стоит заняться нашим читателям:

- ✓ предложить новые виды и типы обучающих программ;
- ✓ предложить приемы поощрения и "наказания" пользователя при работе с обучающей программой;
- ✓ предложить приемы повышения интерактивности обучающей программы;
- ✓ предложить формат "хэдера" файла прямого доступа базы знаний;
- ✓ разработать и утвердить форматы для хранения в базе следующих объектов:
 - экранов и сегментов экрана в компрессированном виде;
 - графических шаблонов и спрайтов;
 - мемо-полей (полей переменной длины);
- ✓ - разработать релоцируемые процедуры для работы с файлами прямого доступа:
 - поиск начала нужной записи;
 - поиск поля;
 - считывание/запись информации в поле;
 - упаковка файла после удаления промежуточной записи;
 - раздвижка файла при вставке записи;
 - сортировка записей по заданному полю;
 - и мн. др.
- ✓ разработать формат файла для создания библиотек процедур в объектном коде;
- ✓ подготовить программу - библиотекарь для включения новых процедур в библиотеки;
- ✓ модернизировать АССЕМБЛЕР для работы с библиотеками процедур.
- ✓ подготовить программу ЛИНКЕР (редактор связей), объединяющую пользовательский объектный код с процедурами, взятыми из библиотек по их имени.

СИСТЕМА iS-DOS

В гостях у фирмы "Slot" (интервью).

© Сергей Симонович, 1994 г.

В последние месяцы среди любителей компьютера "Спектрум" широкое распространение нашла система iS-DOS. Мы уже писали об этой системе в "ZX-РЕВИЮ". Были и публикации статей её разработчиков, но многие вопросы, особенно касающиеся перспектив, открывающихся перед её пользователями, пока не дошли до наших читателей. Получив массу писем по этим вопросам, "ИНФОРКОМ" провёл совещание с руководством фирмы "Slot", в результате которого и получилось то интервью, которое сегодня мы предлагаем нашим читателям. Надеемся, что оно даст ответ на ряд их вопросов и пожеланий.

ДЕЙСТВУЮЩИЕ ЛИЦА:

Фридман Александр Наумович,	директор фирмы "ИНФОРКОМ".
Бурилин Борис Михайлович,	директор фирмы "Slot".
Симонович Сергей Витальевич,	зам. директора "ИНФОРКОМ" по перспективному развитию.
Самыловский Сергей Владимирович,	зам. директора фирмы "Slot" по научной работе.
Шульгина Ирина Викторовна,	коммерческий директор фирмы "Slot".

ИФК: Господа! Позвольте поблагодарить Вас за то, что Вы, смогли уделить время для общения с нашими читателями. Мы получаем большую почту, связанную с системой, распространяемой Вашей фирмой, и, должны прямо сказать, что эта почта противоречива. Многие ещё не поняли, в чем суть данной системы и на каждый десяток писем, поддерживающих Ваше дело, есть несколько писем недоуменных, а иногда даже и негативных.

Slot: Да, действительно, у нас тоже большой почтовый оборот и ситуация нам знакома и понятна. Мы полагаем, что работа с новой системой так далеко уводит обычного пользователя от традиционной работы со "Спектрумом", что у многих и возникает недоумение: "А зачем все это нужно, тем более за деньги?" Те же, кто имеет опыт работы с другими машинами (IBM, "Ямаха" и пр.), хорошо понимают, куда идет процесс и безоговорочно его поддерживают.

ИФК: Давайте начнем по порядку. Вопрос первый, самый болезненный: "Зачем нужна Ваша система, если в неё не входит форматизирующая программа и способность пользователя творить что-то свое близка к нулю?"

Slot: Сегодня это уже не совсем так. Закончив первый этап распространения системы, мы начали распространять вторую версию и теперь распространяем и программу форматирования дискет, правда, пока эта программа защищена и лимит пользователя - 8 копий.

ИФК: А как возникают подобные ограничения?

Slot: Здесь надо сказать, что авторских прав на систему мы не имеем - её разработала С.-Петербургская фирма IskraSoft и в маркетинговых вопросах мы обязаны считаться с пожеланиями и ограничениями, которые накладывают авторы. Другое дело, что они внимательно прислушиваются к нашим рекомендациям и постепенно система открывается. Это ещё не последний шаг, дальше она будет открываться больше.

В то же время, мы не просто дистрибьюторы системы. Мы активно и сами работаем над её развитием. Мы пишем математическое обеспечение, прикладные пакеты, работающие в этой системе, и участвуем в подготовке системных программ.

ИФК: Способ распространения системы без форматизирующей программы выглядит несколько "недружественным" по отношению к пользователям, честно оплатившим её стоимость. Мы понимаем, что, учитывая глубоко укоренившуюся в народе страсть к пиратскому копированию, приходится идти на подобные меры, но получается, что страдают невинные. Что Вы думаете по этому поводу?

Кстати, с моральной точки зрения "синклеристы" - самый благоприятный контингент в стране. Многие из них покупают программы законным путем, даже имея возможность купить и незаконно. Просто потому, что сами любят "Спектрум", уважают и любят тех, кто с ним работает, и готовы поддержать несколькими тысячами рублей энтузиастов,двигающих его вперед. Среди

пользователей IBM-PC-совместимых машин такой благожелательной аудитории никогда не было и неизвестно, когда будет. Многие привыкли все получать у себя на работе бесплатно и не представляют, как можно что-то купить, если можно скопировать. А если и скопировать нельзя, то надо подождать, пока организация выделит на это казенные деньги.

Как-то надо поддерживать легальных пользователей!?

Slot: На самом деле отсутствие у пользователя программы форматера не является таким уж сильным ограничением. В базовой дискете iS-DOS подробно описана технология получения неограниченного числа копий из TR-DOS-овских дискет. Эти копии только не могут быть самозагружаемыми - т.е. загрузиться необходимо с дискеты-оригинала. Согласитесь - это не столь уж страшно.

Не менее интересно и другое - при легальном приобретении системы покупатель получает регистрационную карточку, которую может заполнить и отправить в наш адрес. Так он становится зарегистрированным пользователем. В дальнейшем он будет получать новые версии системы с 50%-ной скидкой, плюс консультирование, плюс самую свежую информацию о новых разработках и мы рассчитываем, что через какое-то время он окупит свои затраты. Когда мы покрепче будем стоять на ногах, мы начнем новые проекты для поддержки именно зарегистрированных пользователей (типа электронного журнала или, скажем, распространения программ Shareware, не знаем, может быть, дело дойдет и до организации конференций).

Нелегальный пользователь, конечно, тоже сможет как-то жить, но не столь удобно. В общем, конечно, наша политика направлена на то, чтобы зарегистрированные пользователи почувствовали, что они не зря приобрели систему.

Хотя, надо сказать, что из нескольких тысяч лицензированных покупателей карточку прислали только 500 человек, так что может быть они просто не поняли её значение или, как это обычно бывает, в круговерти дел забывают её выслать.

ИФК: Второй вопрос, который волнует наших читателей, зачем вообще всё это нужно? Ну, зачем им файловая структура, если в TR-DOS и без неё игрушки нормально запускаются?

Slot: Конечно же, тем, кто их "запускает", эта система не нужна, но будет нужна чуть позже, когда появятся "винчестеры". Попробуйте-ка поработать с "винчестером" в TR-DOS, не имея структуры каталогов. Зато iS-DOS очень нужна тем, кто эти игрушки создает.

ИФК: Должны Вас порадовать. Недавно мы проводили совещание с редакцией журнала "Спектрофон". Этот электронный журнал произвел вполне благоприятное впечатление на страну, и мы не удержались от вопросов о технологии его изготовления. Оказалось, что делают его в Вашей системе... (конечно, приобрели её легально).

Slot: ...Спасибо! Это действительно радостно слышать. На нас тоже "Спектрофон" произвел более чем благоприятное впечатление, хотя немного и расстроил, опубликовав письмо читателя в духе "Кому это все нужно?" по поводу адаптации игровых программ под систему iS-DOS. Может быть, такая адаптация кому-то и не нужна, но ведь в нашей статье (ZX-РЕВИЮ № 1, 1994) было чётко сказано, что это ПРИМЕР того, как надо действовать, если ВДРУГ перед Вами ВСТАНЕТ ТАКАЯ ПРОБЛЕМА. Ну, разве можно нас критиковать за то, что мы подсказываем, как и что можно сделать, если будет нужно?

ИФК: Наверное, не надо расстраиваться, поскольку мнение редакции не обязано совпадать с мнением читателя, а любая дискуссия на эту тему служит интересам и читателей и Вас, как разработчиков и дистрибьюторов, поскольку несёт знание в массы. Другое дело, что редакция могла бы дать и пояснительный комментарий от себя. Тут уж, как говорится, на ошибках учатся.

Но позвольте вернуться к "винчестеру". Потребность в накопителе большой ёмкости давно назрела и, как мы полагаем, Ваша система создавалась не только для того, чтобы дать пользователям удобный RAM-диск, но и доступ к "винчестеру". Что может ждать наших читателей в будущем?

Slot: Проблема "винчестера" уже давно решена. Ещё в конце прошлого года у наших С.-Петербургских коллег стоял и работал "Синклер" с "винчестером". Необходимые драйверы давно разработаны. Дело уперлось в то, что контроллер "винчестера" не желал "жить" со всеми аппаратными решениями "Спектрума". Возникли проблемы. Запускать не полностью совместимый контроллер в производство никто не решился, пришлось проводить исследования,

выявляя пределы совместимости и расширяя их шаг за шагом.

Сейчас контроллер работает практически со всеми основными аппаратными версиями. Дело находится в стадии принятия решения о запуске его в производство. Правда, при установке винчестера возникает необходимость ещё и в дорогом блоке питания, в общем начинают давить ценовые проблемы. Как бы не получилось, что эта конфигурация превысит платежеспособный спрос, а производство - дело дорогое, негибкое и не может рисковать.

ИФК: Может быть, не запускать его в производство, а заняться реализацией не товара, а информационного обеспечения и схемных решений? Рынок достаточно ёмок, чтобы вознаградить труды разработчиков, а инвестиций, в отличие от производства, требуется немного, да и гибкость не теряется. В комплект поставки можно включить печатную плату, альбом схем, если надо - прошитую ПЗУ, инструкцию по сборке и наладке, необходимые программные драйверы на диске. В общем, все это можно сделать за несколько месяцев, не ожидая пока заводы раскатаются.

Slot: Возможно, что Вы правы. Так дело пойдет гораздо быстрее, но надо согласовать этот подход с нашими партнерами. Правда, здесь есть угроза того, что информация, вышедшая из рук разработчиков, распространится мгновенно, не вернув никакого вознаграждения авторам.

ИФК: Да, так оно происходит, если у изобретателя ограничен круг контактов, но, используя "ZX-РЕВЮ" он сразу получает доступ к нескольким тысячам потенциальных заказчиков и может даже сделать такую вещь, как приём предварительных заказов, заранее гарантировав себе, что распространение информации не начнётся, пока не будет собран некий минимальный пакет.

На Западе ведь так и делают. Правда, там банковская система другая. Если клиент послал производителю чек на сотню долларов, то он их не потратил. Они не ушли с его счёта, пока тот не инкассирует этот чек. А производитель не может его инкассировать (банк не позволит), пока не выполнит заказ и не отправит его заказчику. Получается возможность собирать деньги за будущую продукцию (игры, системы, изделия) в любом необходимом объёме, а в то же время заказчики их не теряют. Производитель как бы точно гарантирует себе успех от продаж нового продукта, зная, что оплата за него уже почти лежит у него в кармане и станет его собственностью на следующий день после того, как он закончит и сдаст работу.

Одним словом, если Вы и IskraSoft примете решение о дистрибуции не готового контроллера, а информации о том, как его сделать, то мы готовы Вас поддержать, можете на нас рассчитывать.

Slot: Спасибо, мы обязательно учтём это. Но если говорить о накопителе большой ёмкости, у нас есть ещё один интересный проект. Это ЭЛЕКТРОННЫЙ ВИНЧЕСТЕР на энергонезависимой памяти. Расчёты показывают, что 4-х мегабайтов вполне достаточно пользователю, чтобы держать на нём основные программы и иметь к ним легкий доступ. При этом он получается намного дешевле, чем обычный, нет проблем с энергопотреблением, у него скромные габариты даже вместе с контроллером и очень просто решается программная поддержка. Плюс ещё то, что его объём можно постепенно наращивать.

ИФК: А логически что он будет представлять для пользователя? И как долго энергонезависимая память хранит информацию без регенерации?

Slot: Это будет как бы дополнительный RAM-диск большой ёмкости, способный держать информацию несколько недель, почти месяц.

ИФК: Но ведь это уже совсем другой стиль программирования? Теперь можно одновременно на диске держать и редактор, и ассемблер и редактор связей, и функциональные библиотеки процедур?

Slot: Плюс еще программу - библиотекарь. Да, это будет новый стиль программирования на "Спектруме", который позволит намного ускорить работу над новыми проектами, поднять их качество, приблизить стиль работы к тому, к которому привыкли программисты на IBM PC. Отныне кроме прикладных, системных и развлекательных программ на рынок пойдут коммерческие библиотеки процедур.

ИФК: Нам представляется, что необходимость в этом давно назрела. Нужны графические библиотеки для работы с экраном и спрайтами. Совершенно необходимы библиотеки процедур, осуществляющие обмен информацией с файлами прямого доступа, математические библиотеки и

многие-многие другие.

Slot: Если посмотреть на запад, то обычный язык программирования (оболочка + компилятор + редактор связей + стандартная библиотека) стоит десятки долларов. Специализированные библиотеки процедур стоят уже сотни и даже тысячи, а свои собственные уникальные библиотеки, с помощью которых выпускаются выдающиеся программы, фирмы ценят на вес золота. Это самая высокая ценность, какая у них есть, и её никакими деньгами не измерить. В общем, теперь у наших программистов тоже появится возможность разрабатывать и выпускать на рынок авторские библиотеки, присваивая им громкие и славные имена.

ИФК: Надо только согласовать стандартные требования к процедурам и форматам библиотек и подготовить новые АССЕМБЛЕРЫ, которые не будут морщиться, если встретят в исходном модуле вызов процедуры не по адресу, а по имени и не станут выдавать сообщение об ошибке, а передадут работу редактору связей (линкеру), который вынет из библиотеки нужную процедуру, поставит в нужное место и правильно оформит вызов.

Slot: Да, и такие работы должны начаться в ближайшее время. Сейчас же целесообразно выработать стандарты.

ИФК: Кстати, читатели очень просят АССЕМБЛЕР, работающий под iS-DOS? Что Вы можете им предложить?

Slot: Подготовлен МАКРОАССЕМБЛЕР и, когда читатели будут знакомиться с этим интервью, он уже будет запущен в дистрибуцию.

ИФК: А что касается поддержки модема?

Slot: Все готово. Программная поддержка модема уже существует на нескольких уровнях для тех, у кого есть порт RS232.

ИФК: Что это за уровни?

Slot: Низший уровень - простой ZX-модем, позволяющий вести связь по телефону между двумя "Спектрумами". Следующий уровень - модем для ZX с поблочной передачей информации, с возможностью автонабора номера, уплотнения информации и т.п. Третий уровень - подключение стандартного IBM-овского модема через порт RS232, выход на внешние сети, связь с BBS, с IBM-совместимыми машинами, работа с электронной почтой. В общем, коммуникационный пакет имеется. Осталось наладить производство необходимого оборудования. Некоторые заводы заинтересовались этой задачей и ведут с нами переговоры.

ИФК: А какие ещё есть интересные аппаратные разработки?

Slot: Наверное, рано говорить, о том, что ещё только разрабатывается...

ИФК: На Западе разработчики открывают свои идеи за несколько месяцев до готовности продукта. Надо ведь успеть распространить информацию и сделать продукт известным ко времени его выхода.

Slot: Да, из этих соображений скажем, что готовится новый контроллер экрана, который позволит иметь разрешение на экране 80x32 знакоместа.

ИФК: В итоге, конечно, получится интересный компьютер, но ведь это уже будет никак не "Спектрум"?

Slot: Ни в коем случае, это просто дополнительный режим работы. Иногда это может быть полезным. Работая в информационных сетях с модемом, пользователь будет получать информацию, подготовленную на IBM и отформатированную по 80 символов в строке. Конечно, текстовый редактор позволит её и просмотреть и отредактировать и распечатать, но согласитесь, что во время сеанса связи трудно работать на экране, воспроизводящем только 32 символа в строке. Строки будут резаться в самых неудобных местах.

ИФК: Но ведь за это скажут спасибо и те, кто сейчас работает с системой CP/M!

Slot: Мы на это надеемся.

ИФК: Скажите, а Вам не кажется, что при разработке новых проектов, Вы забываете о необходимости информационно поддерживать потребителей. После того, как в конце 1993-го года вышла книга-описание системы iS-DOS и пара статей в ZX-РЕВЮ, Вы полгода не даёте новой информации и держите страну в сомнениях, стоит или нет осваивать iS-DOS? Когда будут новые книги, когда будет опубликована внутренняя информация по системе?

Slot: Вы совершенно правы. Мы пересмотрим свой подход и постараемся уже в течение

двух-трех месяцев дать дополнительную информацию. Например, вместе с IskraSoft мы готовим к изданию руководство программиста iS-DOS. Надо сказать, что многое из него уже вошло в HELP-структуру, прилагающуюся к АССЕМБЛЕРУ и поставляется на дискетах. Тем не менее, в новой книге будет, о чём рассказать - о точках входа в систему, об оконной технологии iS-DOS и многое-многое другое, без чего пользователь может применять систему для своих задач, но не может её развивать по своему желанию.

ИФК: Если Вы согласны с нами, и всё это станет доступным для наших читателей, то это будет самым лучшим итогом нашей встречи. Мы ещё раз хотим поблагодарить Вас за этот интересный рассказ о будущем новой системы. Как с Вами связаться, если у читателей будут вопросы, пожелания, предложения?

Slot: Нам можно написать по адресу:

Москва, 117330, а/я 707

Опыт программирования в операционной системе iS-DOS.

© Михаил Головин, 1994.

Пример написания драйвера принтера.

В данной статье разбирается построение драйвера печати принтера для компьютера PROFI типа ZX-SPECTRUM. Прочитав эту статью и исправив соответствующим образом листинг программы, Вы сможете "оживить" свой принтер в операционной системе iS-DOS и, тем самым, сможете в полной мере воспользоваться любым программным продуктом работающим с принтером.

По своей структуре драйверы делятся на два типа: так называемые драйверы блочных и драйверы символьных устройств. К первым, например, относится драйвер электронного диска для машин с памятью больше 48Кб (ed_80ips.blk, pr384ips.blk ...). Драйверы символьных устройств - это, например, драйверы принтеров (*.lpr), а также драйверы клавиатуры (ttyin.key, ttyin_mh.key, ttyin48.key...).

По своим функциям каждый тип драйверов подразделяется на драйверы ввода и драйверы вывода информации. Рассматриваемый в данной статье драйвер - это символьный драйвер вывода. В его функции входит принятие от программы байта данных и посылка его на символьное устройство вывода, в данном случае на принтер.

Обычно в драйвере символьного устройства вывода присутствуют:

1. Вектор символьного устройства вывода, в котором определяется основная "стратегия" данного устройства.
2. Программное тело драйвера, в котором определяются основные процедуры, ответственные за вывод.
3. Область настройки абсолютных адресов, что делает драйвер универсальным по нахождению его в памяти. Операция настройки абсолютных адресов выполняется всегда при первой загрузке драйвера в память, а также при его перемещении в ней.

Благодаря наличию этих трёх составляющих, написание драйвера не составит для Вас особого труда. Обратимся к конкретной программе, листинг которой приведен ниже.

В данной программе были использованы следующие обозначения:

L???? - адресная метка или метка данных;

MEM?? - метка используемая для настройки абсолютных адресов программы.

При написании программы драйвера принтера использовались: iS-Edit (редактор), а также пакет программ MONS & GENS (дисконные версии).

Вы набираете текст данной программы в iS-Editor(e), используя все его возможности для редактирования, соблюдая только одно условие: два пробела в начале каждой строки. Затем копируете текстовый файл с программой на TR-DOS диск, загружаете GENS и текстовый файл с программой и сразу же за ним специально созданный файл. Этот файл создается следующим образом:

В GENS(e) создается текст, содержащий одну строку, например:

```
"10 ;";
```

Далее этот текст пишется на TR-DOS-диск, например:

```
"p10,10,END".
```

После загрузки в GENS двух файлов у образовавшегося текстового файла удаляется последняя строка. Затем идёт обычное ассемблирование с записью результатов на TR-DOS-диск. Теперь осталось перезагрузить компьютер и считать на iS-DOS-диск файл, который мы получили в процессе ассемблирования на TR-DOS-диске. С помощью такой технологии можно писать любые программы для iS-DOS, а не только драйверы для принтеров.

P.S. В связи с поступлением в продажу мощной системы программирования - макроассемблера iS-DOS (диск № 17 "Slot" Co., Ltd), в дальнейших материалах обязательно будут рассмотрены его возможности и особенности использования, а также техника программирования непосредственно в самой системе iS-DOS.

```
; eprof.lpr (Драйвер печати)
```

```
; 02.02.94 23:56
```

```

; Адрес загрузки: 50000/#C350
      ORG      50000
; Вектор символьного устройства вывода
MEM00      EQU      $-1
           DEFW    LC3CE      ; Адрес ПП инициализации
MEM01      EQU      $-1
           DEFW    LC37D      ; Адрес ПП вывода символа
MEM02      EQU      $-1
           DEFW    LC3BF      ; Адрес ПП инверсия режима вывода
MEM03      EQU      $-1
           DEFW    LC3BF      ; Адрес ПП установки координат
           DEFB    #03        ; Длина курсора текущего драйвера
           DEFW    #0000      ; Текущие логические координаты печати
           DEFW    #0000      ; Текущие физические координаты печати
           DEFB    #01        ; Тип устройства (0-экран/1-принтер)
           DEFW    #0000      ; Адрес программы обработки ошибок
LC360      PUSH    BC
           PUSH    DE
           PUSH    HL
           LD      BC,#0006
           ADD    HL,BC
MEM04      CALL   LC372
           POP     HL
           POP     DE
           POP     BC
           DI
           RET     NC
           POP     HL
           JR      LC3B8
LC372      EI
           PUSH   AF
           LD      C,#08
           RST    #10        ; Ожидание состояния клавиатуры
                               ; с не нажатыми клавишами.
           POP     AF
           LD      E,(HL)
           INC    HL
           LD      D,(HL)
           EX     DE,HL
           JP     (HL)
; ПП вывода символа (A="код символа")
LC37D      PUSH   AF
           LD      C,#12
           RST    #10        ; Возврат вектора символьного
                               ; устройства вывода.
           EXX
           POP     AF
           CP     #0D
           JR     NZ,LC396    ; Переход, если не конец строки
           BIT    0,(HL)      ; (HL)= 0-экран/1-принтер
           JR     Z,LC396    ; Переход, если вывод на принтер
MEM05      CALL   LC396      ; Иначе, вывести символ на принтер
           RET     C          ; вернуться, если нажат <BREAK>
MEM06      CALL   LC3BF

```

```

LD      A, #0A
LC396   JR      LC396      ;Послать символ "LF"
LD      E, A
LD      A, #0F
OUT     (#3F), A
DI
LC39C   CALL    #1F54      ; Проверка нажатия клавиши <BREAK>
JR      NC, MEM07      ; Переход, если нажат <BREAK>
IN      A, (#5F)
RLA
JR      C, LC39C      ; Цикл: пока нет готовности принтера
LD      A, E
OUT     (#3F), A
LD      A, #0E
OUT     (#5F), A
LC3AD   CALL    #1F54      ; Проверка нажатия клавиши <BREAK>
JR      NC, MEM08      ; Переход, если нажат <BREAK>
IN      A, (#5F)
RLA
JR      NC, LC3AD     ; Цикл: пока нет готовности принтера
LC3B7   XOR     A
LC3B8   LD      E, #0F
LD      C, #5F
OUT     (C), E
EI
LC3BF   RET
LC3C0   LD      A, #96
JR      LC360
MEM07   CALL    LC3C0
JR      LC39C
MEM08   CALL    LC3C0
JR      LC3AD
; ПП инициализации принтера
LC3CE   INC     A
RET     Z
LD      A, #88
OUT     (#7F), A      ; Настройка порта Ввода/Вывода
JR      LC3B7
;Область адресов настройки
DEFW   #FFFF      ; Маркер начала области
DEFW   MEM00, MEM01, MEM02, MEM03
DEFW   MEM04, MEM05, MEM06, MEM07
DEFW   MEM08
; *** Конец программы ***

```

СЕКРЕТЫ TR-DOS

Применение процедур TR-DOS без использования области системных переменных.

© Андрей Бастраков, г. Челябинск.

Основная трудность полной адаптации программ типа "ELITE" к системе TR-DOS состоит в том, что они занимают всю память, включая и область системных переменных, что не позволяет использовать для обращения к диску Бейсик и даже подпрограммы TR-DOS, вызываемые через регистр "С" (несчастные империалисты, не могли байтов 500 оставить, или хотя бы пару адресов добавить к Z-80). В таких случаях выход может быть только в непосредственном обращении к ПЗУ TR-DOS.

Ниже приводятся листинги подпрограмм, из которых видно, как это можно сделать. Адреса подпрограмм ПЗУ указаны для версий TR-DOS 5.03 и 5.04.

Несколько слов об этих версиях. Версия 5.04s отличается от 5.03 только тремя байтами: 2 - номер версии и 1 - флаг переноса в подпрограмме определения скорости перемещения головки. Версия 5.04t - это те же три байта + другая программа форматирования. В остальном они абсолютно идентичны.

Теперь о доработке программы. Подпрограммы обслуживания диска размещаются вместо подпрограмм работы с лентой. Отгрузочный блок размещается в отдельном файле, рассчитанном на одного игрока. При загрузке программы последней операцией должен быть поиск файла на диске, вычисление дорожки и сектора и установка головки на нужную дорожку. Все адреса в листингах даны для версии "JOYSTICK CLUB", будьте внимательны.

Здесь, конечно, я не претендую на истину в последней инстанции, так как все это было написано при остром дефиците информации по TR-DOS.

Подпрограмма "LOAD".

```

55622 21D0DE      LD      HL,57040
55625 CD53DA      CALL   55891      ;Ввод имени пилота.
55628 CD6DD9      CALL   55661      ;Запуск дисковода.
55631 211B2F      LD      HL,12059  ;Команда чтения сектора.
55634 CD2DDA      CALL   55853      ;Переход в ПЗУ TR-DOS.
55637 210048      LD      HL,18432  ;Начало буфера чтения.
55640 1154D0      LD      DE,53332  ;Блок данных ELITE.
55643 016600      LD      BC,00102  ;Длина блока.
55646 EDB0        LDIR
55648 3E01        LD      A,#01
55650 3207D9      LD      (55559),A ;Прошла загрузка.
55653 3ED6        LD      A,#D6     ;Сообщение "FOUND COMMANDER"
55655 CD5ADA      CALL   55898      ;Печать сообщения.
55658 C308DA      JP      55816     ;Переход на продолжение.
                    ORG      55816
55816 2154D0      LD      HL,53332
55819 1132D2      LD      DE,53810 ;Прим. ред.: Здесь есть разночтение,
                    ;которое при редактировании книги
                    ;устранить не удалось в связи с
                    ;отсутствием у нас данной версии
                    ;игры. Адрес #D232 = 53810, а не
                    ;53814, как указано у автора.

55822 015D00      LD      BC,00093
55825 EDB0        LDIR
55827 C327D9      JP      55591     ;Перекомпоновка.
                    ;Возврат в меню.

```

Подпрограмма "SAVE".

```

                    ORG      55680
55680 2154D0      LD      HL,53332

```



```

55683 CD53DA      CALL  55891      ;Ввод имени пилота.
55686 210771     LD     HL,28935
55689 11B1D0     LD     DE,53425  ;Новые данные в
55692 010900     LD     BC,00009  ;отгрузочный блок.
55695 EDB0       LDIR
55697 2154D0     LD     HL,53332
55700 110048     LD     DE,18432  ;Подготовка буфера
55703 016600     LD     BC,00102  ;к записи.
55706 EDB0       LDIR
55708 CD6DD9     CALL  55661      ;Запуск дисководов.
55711 21732D     LD     HL,11635  ;Команда записи сектора.
55714 CD2DDA     CALL  55853      ;Переход в ПЗУ TR-DOS.
55717 00         NOP
55718 00         NOP              ;Удалить ненужные
                          ;команды

```

Следует заметить, что использование подпрограмм по адресам 12059 и 11635 при ошибке чтения или записи приводят к зависанию программы, так как в них в этих случаях используется команда HALT при запрете прерываний. Надо сказать, что в TR-DOS ничего не сделано для контроля записи (о том, что есть, можно даже не упоминать, настолько это не существенно) и зачастую об ошибке записи узнаешь только при считывании.

Подпрограмма "Запуск дисководов".

```

                ORG     55661
55661 3A50DA     LD     A,(55888)  ;Взять номер дорожки.
55664 21502F     LD     HL,12112   ;Команда поиска дорожки для ВГ-93.
55667 F3        DI     ;
55668 CD2DDA     CALL  55853      ;Переход в ПЗУ TR-DOS.
55671 21FD3D     LD     HL,15869   ;Задержка на
55674 CD2DDA     CALL  55853      ;раскрутку диска.
55677 FB        EI     ;
55678 C9        RET     ;

```

Подпрограмма "Переход в ПЗУ TR-DOS".

```

                ORG     55853
55853 E5        PUSH  HL      ;Запомнить адрес команды.
55854 ED5B4FDA   LD     DE,(55887) ;E - номер сектора.
                          ;D - номер дорожки.
55858 212D3D     LD     HL,15661   ;Команда переключения ПЗУ.
55861 1812      JR     55881   ;Переход на продолжение.
                ORG     55881
55881 E5        PUSH  HL      ;Запомнить адрес команды.
55882 210048     LD     HL,18432   ;Установить адрес буфера.
55885 C9        RET     ;Выполнение.

```

Чтобы все это работало, в ячейки 55887 и 55888 нужно занести номера сектора и дорожки. Это может быть сделано так. После загрузки всех блоков программы (можно даже из Бейсика), кроме отгрузочного, запускается следующий загрузчик:

```

                NAME    DEFM  "elitedatC" ;Имя файла.
21 XX XX       LD     HL,NAME  ;Начальный адрес имени.
0E 13         LD     C,#13     ;Переписать имя в
CD 13 3D       CALL  15635   ;системную область.
01 0A 09      LD     BC,02314  ;Искать номер
78            LD     A,B     ;файла в каталоге.
32 06 5D      LD     (23814),A    ;Число символов для поиска.
CD 13 3D       CALL  15635
79            LD     A,C     ;Теперь в А - номер файла.
0E 08         LD     C,#08    ;Прочитать заголовок
CD 13 3D       CALL  15635   ;файла из каталога.
2A EB 5C      LD     HL,(23787)

```

22 4F DA	LD	(55887),HL	;Запомнить дорожку и сектор.
3A EC 5C	LD	A,(23788)	;Определение дорожки для
1F	RRA		;2-стороннего диска (используется
32 50 DA	LD	(55888),A	;в подпрограмма ПЗУ).
01 05 01	LD	BC,00261	;Команда чтения файла.
EB	EX	DE,HL	;В DE - дорожка и сектор.
21 00 48	LD	HL,18432	;Адрес буфера.
CD 13 3D	CALL	15635	;Прочитать файл (установить головку
			;на нужной дорожке, в системный
			;регистр записать номер стороны).
31 XX XX	LD	SP,SPACE	;Указатель стека на свободное
		;пространство.

Здесь при считывании файла сразу убиваются два зайца: головка устанавливается на необходимую дорожку и программируется системный регистр (TM9), что избавляет от необходимости делать это специально.

Затем производится окончательная компоновка программы, если нужно - ввод POKE's (обязательно в машинных кодах), а затем запуск:

F3	DI	
3E FD	LD	A,#FD
ED 47	LD	I,A
ED 5E	IM	2
31 FD FF	LD	SP,65533
AF	XOR	A
32 E3 70	LD	(28899),A
C3 66 73	JP	29542

Где взять отгрузочный файл? Два самых простых способа.

1. Взять магнитофонный вариант и любым копировщиком, например, "PCOPIER" Н.Родионова, переписать его на диск, а затем переименовать, к примеру, в "elitedat".

2. Прodelать такую операцию:

```
RANDOMIZE USR 15619: REM : SAVE "elitedat"CODE 255,102
```

Затем загружаете "ELITE" и в "ACCESS MENU" клавишей "2" заполняете файл нужными Вам данными (до этой операции "1" лучше не трогать!).

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

Mouse + Spрессу или подключение манипулятора MM8031 к персональному компьютеру ZX-Spectrum

© Александр Локтев, г. Херсон, 1994.

От автора.

Этот материал адресован, прежде всего, горячим поклонникам компьютера ZX-Spectrum, натурам творческим и неугомонным, желающим как-то дополнить свой Spрессу, расширить его возможности и область применения, проявить себя в новом качестве, не только как банального потребителя игрушек - LOAD "" <ENTER>, но и как создатель чего-то нового. Испытывающим радость, скорее от самого процесса творения, чем от конечного результата.

Хочу поблагодарить моего коллегу и товарища Степнова А.Н. за доверительные беседы по данной теме, за идеи, которые я так нескромно использовал, за помощь при написании и моральную поддержку.

Чарльзу Муру - автору языка FORTH и сэру Клайву Синклеру, давшему мне возможность познакомиться с этим языком, посвящается.

Предисловие.

Каждый, кто работает с этим замечательным компьютером довольно длительное время и знает о существовании, кроме клавиатуры и джойстика, ещё и о другом способе управления программой - с помощью мыши - часто задает вопрос: "Ну почему этого нет у меня? Почему я должен ждать, нажав клавишу или отклонив ручку джойстика в сторону, пока курсор или стрелка медленно переместится из одного конца экрана в другой. Неужели нельзя подключить "колобок" и почувствовать себя полноценным пользователем, а Spectrum - полноценной машиной?"

Конечно, всё это уже давно существует, подключено, работает, имеет несколько разновидностей, но не у нас. Впрочем, может быть, у нас тоже что-то имеется, но мне об этом ничего не известно. Я долго ждал чего-нибудь похожего, совместимого, и сразу работающего, но не дождался, плюнул и сделал по-своему.

Необходимо добавить, что все нижеследующие рассуждения относятся только к мыши MM8031 известной в народе, как "корветовская", которая была выбрана в качестве базовой из-за своей низкой цены и широкого распространения на Херсонской "туче".

1. Theoretics или теория или кое-что о mous'ax.

1.1. Что из всего этого получится.

Сразу же хочу немного умерить радость, возможно возникшую у Вас после прочтения предыдущих строк. Поскольку, проделав все необходимые действия, связанные с подключением, Вы, в итоге, получите устройство, которое не будет работать ни с одной из известных программ, допускающих управление от мыши и, тем более, такого управления не допускающих. То есть у Вас будет, если Вы того пожелаете, ни Kempston-mouse, ни AMX-mouse, ни какая-либо другая, а просто MM8031, подключенная к Спектруму. Так зачем же огород городить, возможно, спросите Вы? Да, действительно, изобретать то, что уже давно работает и при этом гораздо лучше, чем то, что сделаешь сам, занятие неблагодарное и не интересное. Да я бы и не стал этим заниматься, но дело в том, что подключить, по крайней мере, этот манипулятор (что-то язык не поворачивается назвать его мышью) настолько просто, что не сделать это после того, как станет, понятен принцип его работы, просто неприлично.

Кроме того, программу поддержки манипулятора, обеспечивающую опрос и передачу параметров последнего в головную программу или драйвер, как принято говорить, можно написать на чем угодно - начиная от Бейсика и кончая машинными кодами. Последнее является немаловажным фактом, поскольку возникает возможность (при использовании машинных кодов) встроить этот драйвер в любую программу даже и не имеющую опцию "мышь" и реализовать более удобное управление. Да и при написании Вами Ваших собственных программ наличие

вышеуказанной опции придаст им более профессиональный вид, а новый способ управления будет во многих случаях предпочтительнее, чем традиционный.

1.2. Что же у мыши внутри и как все это работает.

Сейчас я хочу затронуть вопрос принципа работы мыши вообще и данного экземпляра в частности. Тем, кому не терпится привести в исполнение заветную мечту немедленно взять в руки паяльник и приступить к делу, скажу следующее. Ради бога, не делайте "сопли" под прикрытием фраз типа: "Неужели будет работать?", "Надо же проверить" или же еще чего-нибудь, в любом случае относясь с недоверием к автору схемы и машине. Работать будет! Поэтому смело лепите набело без черновиков и промежуточных проверок. Впрочем, это дело хозяйское. Удачи!

Для остальных же, которые стойко решили дочитать до конца, продолжаю. На сегодняшний день существует очень большой ассортимент мышей. Поскольку стандартов на них нет, то каждый, кто берется за их разработку, руководствуется лишь своими соображениями. Различия начинаются уже с самого принципа действия - преобразования физического перемещения мыши в последовательность электрических сигналов. По этому критерию можно выделить три класса мышей: механические, оптомеханические и оптические. Для последних нужен ещё и специальный планшет, по которому происходит перемещение. По способу общения с компьютером мыши можно разделить на последовательные, имеющий интерфейс RS-232 и параллельные - с интерфейсом типа - CENTRONICS. Начинка у них тоже отличается. Обычно, всю обработку информации, получаемую от датчиков и поддержку протокола обмена данными с компьютером выполняет специальная БИС, находящаяся в самой мыши. Но, зачастую, применяется более упрощенный вариант - на "рассыпухе", как в MM8031, или ещё проще - с одними лишь датчиками и четвёркой триггеров Шмитта, как в "графическом манипуляторе" для компьютера ЕС-1841. В этих случаях обработка данных выполняется или на плате интерфейса, или программно.

Устройство, о котором будет идти речь (MM8031), занимает промежуточное положение, являясь оптомеханической мышью с параллельным интерфейсом что-то вроде CENTRONICS'a, и простейшей обработкой сигналов датчиков.

Любое перемещение мыши происходит на плоскости и, поэтому, наиболее просто для фиксации её местоположения использовать декартову систему координат с известными со школьного курса осями X и Y. Поскольку экран телевизора или монитора - тоже плоскость и тоже требует двух координат для задания положения точки или курсора, то, установив соответствие между аналогичными осями экрана и мыши, можно однозначно определить положение последней на плоскости стола, стула, коленке или на чем угодно, и отобразить на экране.

Механически это реализовано так. Шарик, катающийся по поверхности, обычно стола, у нас - с положенным сверху, в лучшем случае куском ватмана, у них - специального резинового коврика, передает свое вращение двум взаимно перпендикулярным осям. На каждой из осей имеется по одному диску с прорезями, расположенными аккурат между оптопарой светодиод-фотодиод. Диск при вращении то заслоняет, то пропускает луч от свето- к фотодиоду, что приводит к появлению импульсов на последнем. То есть величина перемещения Вашей руки с мышью по одной из декартовых координат, к примеру по X, за определенный промежуток времени однозначно определяется количеством оборотов диска, насаженного на ось, совпадающую с декартовой координатой Y (именно Y, так как направление перемещения не совпадает с той осью, которая вращается) за то же время. Или количеством прорезей, успевших пересечь луч светодиода. Или, в конечном итоге, количеством импульсов, поступивших с фотодиода в схему, опять-таки за данный промежуток времени.

Если до этих пор Вам все понятно, то вполне резонен вопрос: - А как же быть с направлением перемещения? Ведь по этой оси руку можно двигать и в ту, и в другую сторону, а количество импульсов, при равном относительном перемещении, будет все равно одно и то же. Так вот, на каждом диске стоит не одна, а две оптопары. И расположены они таким образом, чтобы работать в противофазе. То есть, если напротив одного фотодиода расположена прорезь и через неё падает свет от напарника светодиода, то другой фотодиод в это время затемнен.

В некоторых манипуляторах, я имею ввиду MM8031, регулировка данного соответствия выполняется обыкновенными спичками. Так что, если у Вас в дальнейшем возникнут какие-либо

проблемы с таким экземпляром, то вот одна из возможных неисправностей - спички высохли, выпали и "юстировка" нарушилась. Исправить положение, думаю, Вы сможете и без моих подсказок.

Продолжаю. Импульсы с обоих фотодиодов поступают на реверсивный счетчик с предустановкой типа K555ИЕ13. Импульсы с одного из них подаются на вход "разрешение счёта", а со второго на вход "направление счёта" и, кроме того, через задерживающую цепочку, на вход "счёт", непосредственно считающий их количество. На выходе счетчика имеем 4-х битный код, который может, как увеличиваться, так и уменьшаться, в зависимости от направления вращения диска и принимает значения от 0 до 15. Из него необходимо, каким-то образом, получить две величины - направление перемещения и его абсолютное значение. Первая определяется, как ни странно, довольно просто, поскольку это предусмотрено самой схемой. Для этого на входах предустановки заранее выставлено число в двоичном представлении имеющее вид: 1000, а в привычном для нас: 8. Что достигается просто замыканием на землю трех младших разрядов и подключением к +5В старшего. В момент сброса счетчика, производимого внешним сигналом, в него записывается код, имеющийся на входах предустановки, то есть 8. Теперь, если Вы будете сравнивать последующие показания счётчика с этим числом, то легко определите знак, а следовательно, и направление перемещения по данной координате.

Что касается определения второй величины, то, как Вы уже поняли, таким малым диапазоном чисел - всего 16, а при учёте вышесказанного и того меньше - 7, невозможно передать абсолютную координату мыши. И решение этой задачи разработчики устройства заботливо переложили на плечи тех, у кого возникнет желание опробовать его в работе. Ну а это не очень сложно. Как при измерении, скажем, длины комнаты можно пользоваться и десятиметровой рулеткой и линейкой в пол метра, так и в данном случае величину перемещения мыши можно представить суммой величин небольших перемещений. А в переложении на "железо", количеством импульсов: от 8 до 15 в одну сторону и от 8 до 0 в другую. То есть вместо одного измерения производится несколько и величина каждого последующего прибавляется к предыдущему, разумеется, со своим знаком. Таким образом, мышь как бы непрерывно отслеживается по любой из координат.

Дальше. Четырехбитный код с выхода каждого из счётчиков (X-координаты и Y-координаты) поступает на мультиплексор типа K555КП15, а с него как четыре шины данных (D3, D2, D1, D0) на разъём. Управлять мультиплексором для считывания четверки данных соответствующих координате X или Y можно с помощью внешнего сигнала.

Итак, для того, чтобы узнать величину и направление перемещения манипулятора отдельно по каждой оси необходимо:

- ✓ подать сигнал сброса на счетчики;
- ✓ через какое-то время считать значение с шины данных для одной координатной оси;
- ✓ переключить мультиплексор и сразу же считать значение с шины данных для другой координатной оси.

Здесь имеют место некоторые тонкие моменты. Дело в том, что, если время между сбросом и считыванием будет слишком мало, то Вы рискуете не считать вообще ничего. Так как диск у манипулятора, возможно, не успеет повернуться ни на одно деление даже при самом быстром перемещении Вашей руки. И, наоборот, если это время будет слишком большим, то счётчик, работая, к примеру, на увеличение и начиная с 8, успеет перейти через 15 и начнет с 0. А Вы, считав его значение (пусть будет 3), определите, что манипулятор переместился вверх на $8-3=5$ единиц, хотя, на самом деле, вниз и на $16-8+3=11$.

Для тех, кто не знаком с работой счётчика, поясню, что значения на его выходе изменяются циклически. То есть, если счётчик работает на увеличение, то с приходом каждого последующего импульса, значение кода на его выходе увеличивается на единицу и при достижении значения 1111 (в десятичном представлении - 15) следующий импульс обнуляет счётчик и процесс повторяется. Аналогично и при работе на уменьшение - код на выходе уменьшается на единицу и при достижении 0000, следующий импульс устанавливает на выходе 1111, затем 1110, 1101 и так далее.

Исходя из вышесказанного, для связи (электрической) устройства с компьютером

необходимы следующие сигналы:

- ✓ данные с шины мультиплексора (от манипулятора к компьютеру);
- ✓ сигнал инициализации или сброса (от компьютера к манипулятору);
- ✓ сигнал переключения мультиплексора (от компьютера к манипулятору).

С шиной данных вопросов не возникает - это четыре провода D3, D2, D1 и D0. А вот сигналы инициализации счётчиков и переключения мультиплексора разработчики для уменьшения количества отходящих проводов решили объединить в один и обозвали "Синхро". Данное решение кое к чему нас обязывает, но об этом позже. Кроме того, я не упомянул о кнопках, имеющихся на каждой мыши, количество которых колеблется в широких пределах, но обычно равно трём (имею ввиду ихние tous'ы). Наш MM80310 (как звучит!) тоже снабжен парой "микриков", работающих на замыкание. Два конца у них общие и соединены с землей, а два других выведены на внешний разъём и с ними можно делать все, что душе угодно. Можно завязать бантиком, отрезать, или вообще, не трогать - пусть болтаются. Но, обычно, по ним передают сигналы в компьютер. И, при этом, левую кнопку используют, как аналог клавиши "ENTER", то есть ввод или подтверждение, а правую, как "ESC", то есть выход или отмена. А простому: "ДА" и "НЕТ".

Итак, полный набор сигналов или проводов:

- D3
- D2 (четырёхбитный код D0-D3)
- D1
- D0
- синхро-сброс и переключение мультиплексора;
- КН1 кнопка "ДА";
- КН2 кнопка "НЕТ";
- +5В питание схемы манипулятора;
- 0В земля.

2. Hardware или аппаратное обеспечение, или "железо".

Ну что, если теория Вам ясна, то практика, как и было, обещано, будет ещё проще. При разработке схемы подключения я не стал изобретать велосипед, а воспользовался имеющейся на плате компьютера микросхемой параллельного интерфейса KP5 8 0BB55A на порт А которого как-то давным-давно повесил джойстик. Поэтому схема получилась такая, как на рис. 1. В некоторых компьютерах эта микросхема уже имеется, кроме того, многие устанавливают ее сами для подключения периферии. Она хороша своей универсальностью и сговорчивостью. И джойстик, и программатор, и световое перо, и даже принтер находят с ней общий язык. Поэтому у кого этого простого интерфейса нет - рекомендую.

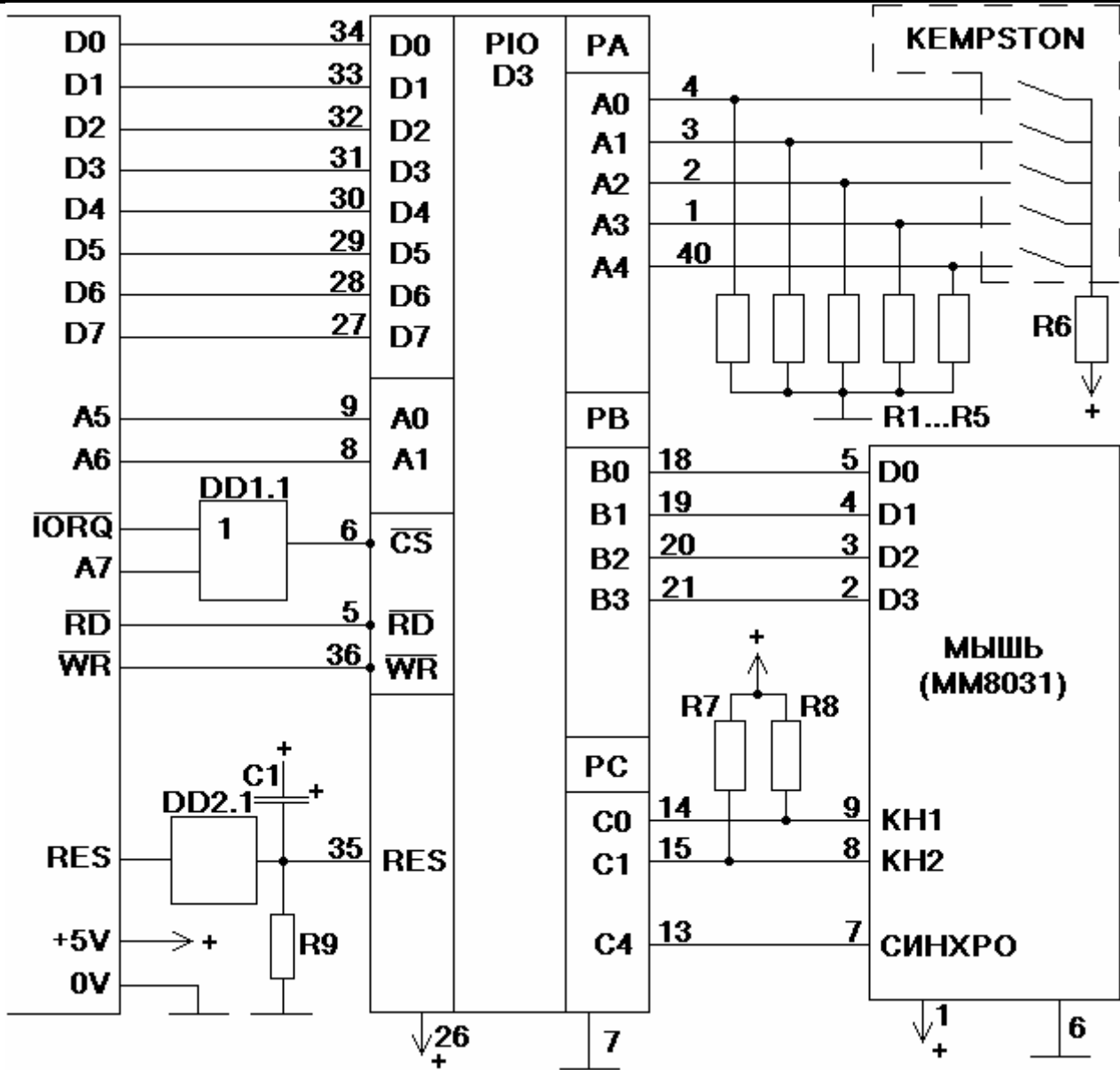


Рис. 1.

Немного о схеме. PIO (KP580BB55A) включается в работу или, как говорят, выбирается сигналом низкого уровня, поступающего на вход CS, что возможно, согласно схеме, при наличии 0 на шинах IORQ и A7 микропроцессора. Сигналы с шин A5 и A6, поступают на входы A0 и A1 PIO, управляют выбором активного в данный момент порта - A, B или C. Сигналы чтения PIO - RD, записи в PIO - WR, а также шина данных D0...D7 подключены напрямую к Z80. К порту A, как говорилось выше, можете подключить Kempston-джойстик. Этот порт работает только на ввод чтение информации с внешнего устройства). В порт B заведена четырехбитная шина данных от MM8031 и он, порт B, также работает только на чтение. Порт C разделен как бы на два - по четыре шины данных в каждом. Младшая четверка - D0, D1, D2, D3 настраивается на ввод информации, а старшая - D4, D5, D6, D7 на вывод. И в общем случае порт C читает информацию при опросе состояния кнопок КН1, КН2 и работает на запись (выдает информацию на внешнее устройство) при активизации сигнала "Синхро". Поскольку провода, идущие к джойстику и кнопкам имеют значительную длину и, на самом деле, болтаются в воздухе (имеют неопределенный потенциал), то в них может появиться или навестись напряжение произвольной величины. Это приведет к ошибочному результату при опросе PIO по соответствующему порту. Для исключения этого явления и служат сопротивления R1...R5, R7, R8. Величина их, как и остальных, не критична и может находиться в диапазоне от 1 до 20 кОм. Обычно принимают номинал 10 кОм. Ну вот, пожалуй, и всё. Да, забыл сказать, что порт A, в данной реализации имеет адрес 31, порт B - 63, и порт C - 95. А в качестве элементов D1 и D2 можно использовать микросхемы К155ЛЛ1 и

K155ЛН1 соответственно.

3. Проверка и ещё раз проверка.

Если Вы уже успели подключиться и горите желанием немедленных действий, то не спешите - неплохо было бы проверить качество Вашей работы и работоспособность устройства в целом. Следующие программы дадут Вам такую возможность и, кстати, могут пригодиться в дальнейшем при ремонте, доработках и в других экстренных случаях.

```
10 OUT 127,147: OUT 95,16
20 PRINT AT 10,10;IN 63;" ": GO TO 20
```

Эта программка производит проверку тракта координат. Первый оператор OUT производит настройку PIO, второй - переключает мультиплексор (в данном случае на Y-координату), а строка 20 опрашивает счетчик манипулятора и выводит его значение на экран. Для проверки X-координаты достаточно заменить оператор OUT 95,16 на OUT 95,0. При правильном функционировании интерфейса, Вы в обоих случаях должны наблюдать что-то вроде счётчика от 0 до 15, показания которого зависят от перемещения MM8031.

Уверенность в работоспособности кнопок дадут Вам следующие строчки:

```
10 OUT 127,147
20 IF IN 95<3 THEN BEEP 0.1,0
30 GO TO 2 0
```

Если все три теста прошли успешно, то комплексную проверку системы поможет провести вот такая программка:

```
10 OUT 127,147: PRINT AT 10,8;"X=";AT 10,6;"Y="
20 PRINT AT 12,8;IN 63;" ": OUT 95,16: PRINT AT 12,6;IN 63; " ": OUT 95,0
30 LET S=IN 95: IF S<3 THEN BEEP 0.1,S*10
40 GO TO 20
```

Правильное подключение и верно набранный листинг даст Вам возможность наблюдать зависимость величин относительных перемещений X и Y от скорости и направления манипулятора, а остановка последнего обнулит их. Кроме того, при нажатии кнопок клавиш, строка 30 обязательно передаст Вам своё восхищение слаженностью работы устройства звуковыми сигналами разной высоты. В противном случае повторяйте тесты и проверяйте разводку, питание, дешифрацию PIO и, может быть, даже саму схему MM8031. Кстати, не знаю какой будет у него разъем - я давал распайку в расчете на часто встречающийся ГРП 15.

Поскольку интерфейс получился очень простым, то вся тяжесть взаимоотношений между Спрессу и MM8031, их диалога и разрешения конфликтных ситуаций ложится на бедную программу поддержки или драйвер. Итак...

4. Software или программное обеспечение, или драйверы.

Как уже говорилось выше, драйвер - это небольшая программа, являющаяся посредником между основной программой и устройством, которое она обслуживает. Все многообразие драйверов устройств ввода или управления курсором (не только мыши, но и джойстика, и, клавиатуры) можно разделить на две группы:

- ✓ драйверы реализующие попиксельное перемещение курсора;
- ✓ драйверы реализующие перемещение курсора по знакоместам.

Первые применяются в программах, работающих с графикой, - графических редакторах, САД и тому подобных системах, а вторые - в текстовых редакторах, языках программирования, базах данных и любых других, работающих с текстом.

Кроме того, драйвер, как уже говорилось, может быть написан на языке любого уровня - начиная с Ассемблера и кончая Бейсиком и даже при их совместном использовании в зависимости от каждого конкретного приложения.

Приведу несколько конкретных примеров реализации вышесказанного с подробными комментариями.

4.1. Познакоместный драйвер на Бейсике.

Допустим, у Вас возникло желание получить перемещение курсора по знакоместам. В таком случае Вам будут нужны значения X и Y координат плюс состояние кнопок манипулятора (кстати, для перемещения по пикселям нужно то же самое). На Бейсике реализация Вашего

желания может выглядеть так.

```

10 OUT 127,147: REM Настройка портов А и В на ввод, а порта С на ввод-вывод;
20 LET C=PEEK 2252 8: LET X=6: LET Y=10: LET XX=X: LET YY=Y: REM настройка
  переменных; XX и YY переменные текущих, а X и Y переменные новых
  координат; C -значение атрибута первого знакоместа;
30 LET X1=IN 63-8: OUT 95,16: LET Y1=IN 63-8: REM определение относительного
  перемещения манипулятора по координате X, переключение мультиплексора, то
  же самое по координате Y;
40 LET X=X+X1: IF S<0 THEN LET X=0
50 IF X>31 THEN LET X=31: REM Проверка выхода абсолютной координаты X за
  границы экрана и установка крайних значений;
60 LET Y=Y-Y1: IF Y<0 THEN LET Y=0
70 IF Y>21 THEN LET Y=21: REM Аналогично для координаты Y;
80 IF Y=YY AND X=XX THEN OUT 95,0: GO TO 30: REM Если MM8031 не перемещался,
  то сброс счетчиков и все сначала;
90 LET ADR=22528+YY*32+XX:POKE ADR,C: REM В противном случае помещение в
  ячейку области атрибутов, соответствующую текущему знакоместу исходного
  значения;
100 LET ADR=22528+Y*32+X: POKE ADR,47: REM и помещение в ячейку области
  атрибутов, соответствующую тому знакоместу, к которому переместился
  манипулятор, нового значения;
110 LET XX=X: OUT 95,0: LET YY=Y: REM Сброс счетчиков и присваивание
  переменным текущих координат новых значений;
120 GO TO 30: REM Все сначала.
  
```

Такой вариант драйвера является скорее демонстрационной, чем рабочей программой, на которой, благодаря своей простоте и открытости можно попрактиковаться и уяснить некоторые ключевые вопросы взаимодействия hardware и software. Я назвал этот драйвер познакоместным, но это не совсем так, поскольку он реализует перемещение не по знакоместам, а по области их атрибутов. И у пользователя создается лишь иллюзия того, что курсор - тёмный квадрат - переходит от одного знакоместа к другому. Но, в общем, очень похоже. Весь фокус заключён в строках с номерами 90 и 100. В них как раз и рассчитываются адреса области атрибутов, соответствующие текущему знакоместу и новому, на которое якобы переполз курсор. Строка 80 введена для того, чтобы курсор не "блывал" в случае остановки манипулятора. И ещё, в целях упрощения не производится опрос клавиш.

А теперь о некоторых временных характеристиках. Если помните, я говорил о промежутке времени между сбросом и считыванием, а также на что его величина влияет. Назову этот промежуток времени – временем ожидания. И сейчас Вы сможете лично лицезреть и на примере уяснить зависимость качества выполнения программы от величины этого промежутка (при наличии, конечно, подключенного устройства и набранного драйвера). Напомню, что считывание координат осуществляется командой IN 63, а сброс счётчиков командой OUT 95,0. Время ожидания зависит от того, в каком месте программы будет находиться оператор OUT. То есть оно равно времени, которое понадобится интерпретатору Бейсика на выполнение всех команд, находящихся между операторами OUT 95,0 и IN 63. В данном случае это команды LET YY=Y и GO TO 30. Конечно, при необходимости время ожидания можно регулировать просто оператором PAUSE N, подбирая N, но уж слишком расточительно заставлять простаивать и без того медленный интерпретатор. Почему это плохо, будет ясно из дальнейшего.

Если Вы поместите, оператор OUT в строку 30 перед первым оператором IN, уменьшив, тем самым время ожидания до минимума, то курсор у Вас будет ползать как черепаха. А если куда-нибудь в строку, эдак 90, то... сами увидите, что будет. В первом случае манипулятор еле поспевал за драйвером - программа фиксировала только самые быстрые перемещения. Во втором, наоборот - для нормального отображения курсора необходимы медленные передвижения Вашей руки с MM8031. Так что, устанавливая, оператор OUT в разные точки программы Вы можете подобрать для него наиболее подходящее место. Но это ещё не все. Кроме времени ожидания - между сбросом и считыванием существует и другое - между считыванием и сбросом. Это плохое

время. Назову его мёртвым. Оно служит лишь для работы самой программы, и в нём нет места для диалога с устройством, движение которого в этот период не прекращается и должно быть отработано. Посмотрите на рис. 2. Это график зависимости перемещения манипулятора с постоянной скоростью и в одном направлении от времени.

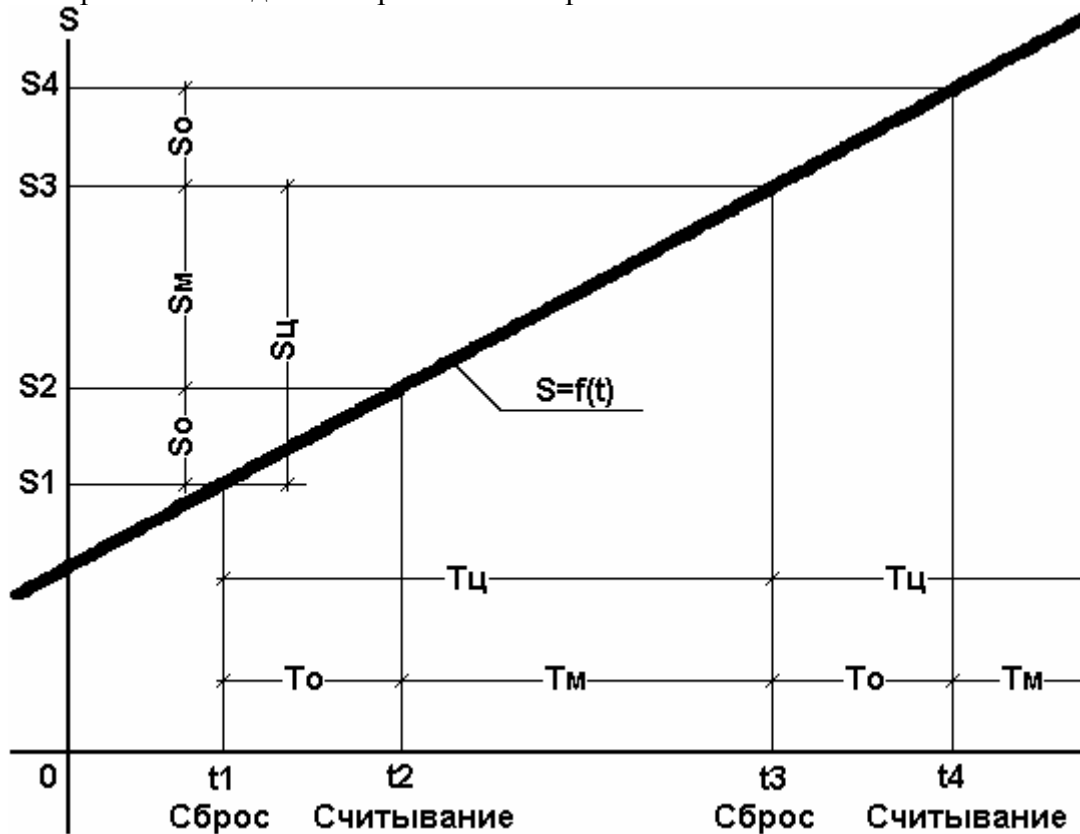


Рис. 2.

За время ожидания $T_0=t_2-t_1$ Ваша рука с устройством перемещается на расстояние $S_0=S_2-S_1$. Драйвер считывает это значение, обрабатывает, добавляет к предыдущему и так далее, совершенно забыв о продолжающем своё движение манипуляторе. Все это происходит за время $T_M=t_3-t_2$. В момент t_2 драйвер, наконец, подготовился к следующему опросу, но манипулятором уже пройдено расстояние $S_m=S_3-S_2$, которое так никем и не было зафиксировано. И чем больше мёртвое время T_M , тем больше это неучтённое расстояние S_m . Что же в этом плохого? - возможно, спросите Вы. А то и плохо, что для перемещения курсора на десяток другой знакомств Вам надо отмахнуть рукой с полметра - падает чувствительность всей системы, и результат работы этого симбиоза выглядит ужасно примитивным и неинтересным.

Попробуем определить для этой системы понятие К.П.Д., зачем он нужен и как с ним бороться. Если мне не изменяет память, то К.П.Д. есть отношение полезной работы ко всей совершенной. В нашем случае полезной работой является работа, совершаемая Вашей рукой при перемещении манипулятора на расстояние S_0 , принимаемое в расчет драйвером, а всей совершенной - работа при перемещении на расстояние S_0+S_m . То есть:

$$К.П.Д. = \frac{A_0}{A_0 + A_M} * 100\% \quad (4.1)$$

где A_0 - работа на участке S_0 ; A_M - работа на участке S_m .

Поскольку и скорость и масса неизменны, то, я думаю, ни у кого не вызовет сомнений следующее выражение:

$$К.П.Д. = \frac{T_0}{T_0 + T_M} * 100\% \quad (4.2)$$

Ни для кого не секрет, что чем выше К.П.Д. системы, тем она выгоднее, экономичнее, эргономичнее и, вообще, заслуживает всяческих похвал. Аналогично и здесь. Ну, насчёт каких-то особых выгод и похвал я не знаю, а вот эргономические свойства, на мой взгляд, играют большую

роль. И К.П.Д. увеличивать все-таки придется.

Для этого, как видно из (4.2), нужно либо увеличивать T_o , либо уменьшать T_m . Однако, как Вы знаете, время T_o нельзя чрезмерно увеличивать, как, впрочем, и уменьшать. Оно лежит в очень узких пределах. Остается время T_m .

Что характерно, время одного цикла $T_{ц} = T_o + T_m$ для данной конкретной реализации драйвера постоянно и, поэтому, можно написать, что $T_o + T_m = \text{const}$. Из этого следует, что, если мы изменяем время T_m только лишь перемещением оператора OUT 95,0 внутри самой программы, то одновременно изменяем и T_o , только в противоположную сторону. Стало быть, такой способ увеличения К.П.Д. довольно проблематичен и малоэффективен. Могу предложить ещё два - это уменьшение объема программы и увеличение скорости её выполнения.

Отсюда выводы:

- ✓ устанавливать время ожидания очень небольшим;
- ✓ всеми силами стремиться к уменьшению мертвого времени.

Второй вывод особенно важен для медленных драйверов - на языках высокого уровня.

Теперь что-нибудь реальное.

```

10 LET C=PEEK 22528: LET X=16: LET Y=16: LET ADR=22528+Y*32+X: LET MOUSE=10
20 OUT 127,147
30 LET X1 = IN 63-8: OUT 95,16: LET Y1=IN 63-8
40 LET X=X+X1: IF X<0 THEN LET X=0
50 IF X>31 THEN LET X=31
60 LET Y=Y+Y1: IF Y<0 THEN LET Y=0
70 IF Y>21 THEN LET Y=21
80 LET S = IN 95-16: IF S<3 THEN RETURN
90 IF X1=0 AND Y1=0 THEN OUT 95,0: GO TO 30
100 POKE ADR,C: LET ADR=22528+Y*32+X: POKE ADR,47
110 OUT 95,0: GO TO 30
    
```

Как использовать этот драйвер? Очень просто. В Вашей головной программе - адаптируемой, либо лично выстраданной - в том её месте, где по Вашему замыслу от потенциального пользователя (п.п.) требуются какие-то действия - что-либо выбрать или на что-то указать - необходимо установить оператор GO SUB MOUSE. Это даёт п.п. возможность неограниченно долго гонять квадратик по экрану. И, когда эта забава ему наскучит, он, наконец, вспомнит, что от него требуется, установит курсор по своему желанию и разумению и нажмет одну из альтернативных кнопок - вот тогда, следуя строке 80, и будет осуществлен возврат в головную программу. При этом Вам в руки передаются три переменные величины: X-координата (от 0 до 31), Y-координата (от 0 до 21) и код нажатой клавиши S (1 или 2). Дальнейшее использование этого бесценного дара зависит от Ваших желаний и фантазии.

Конечно, драйвер на Бейсике это хорошо. Это простота, легкость настройки, возможность включения в любой бейсиковский диалект будь то Mega-, Beta- или другой. Но если Вам понадобится как-то изменить его - задействовать дополнительные расчеты, проверки, условия или что-то, что приведёт к увеличению размера самого драйвера, то Вы неизбежно заплатите за это тем, ради чего, собственно, всё и затевалось - удобством управления. Вы забыли о T_m , а оно обязательно возрастет со всеми вытекающими последствиями.

Единственный выход из этого положения - придумать, как увеличить скорость выполнения самой программы. Первое, что приходит в голову - это компилятор. Он даст возможность в несколько раз, а то и десятков раз увеличить скорость, что нам и нужно. Подойдет любой, например Mcoder2. (Всех страждущих узнать, что это такое, зачем и как работает, отсылаю к [1].) Правда, в результате этой операции Вы потеряете солидный кусок памяти Sпрессу - он понадобится самому компилятору (5375 байт). Зато, какой эффект. Программу для компиляции нужно немного изменить. В частности, 30 и 80 строки будут выглядеть так:

```

30 LET X1=IN 63-8: OUT 95,16: LET Y1=IN 63-8: OUT 95,0
80 LET S=IN 95: IF S<3 THEN POKE ADR1,X: POKE ADR2,Y: POKE ADR3,S: RETURN
    
```

где ADR1, ADR2 и ADR3 - выбранные Вами адреса для передачи параметров от откомпилированного драйвера в основную программу.

Из строки 120 оператор OUT 95,0 следует удалить. Если Вы смотрели на работу всей

системы до компиляции с большой приятностью - "Это ж надо, работает!" - то после этого процесса будете поражены ещё больше. Впрочем, не знаю как Вы, а я так просто обалдел. Вот Вам, пожалуйста, и эргономика, если кто желает. Хочу обратить Ваше внимание на величину времени T_m . Оно сократилось до предела и определяется лишь временем выполнения команд OUT 95,16 и LET Y1=IN 63-8. А T_0 почти полностью совпало с временем выполнения самой программы. Это произошло за счёт резкого сокращения T_c (времени выполнения одного цикла программы). И, чтобы время T_0 не уменьшилось во столько же раз, пришлось оператор OUT 95,0 передвинуть на максимально возможное расстояние. Вот и выходит, что главной задачей является уменьшение времени цикла - T_c , а там уже разберемся и с T_0 и с T_m .

4.2. Познакоместный драйвер на языке FORTH с включением машинных ходов.

Возможно, в названии этой части я сделал упрощение и следовало бы к словам "на языке FORTH" добавить ещё и слова: "для языка FORTH". Поскольку, как драйвер на Бейсике работает только с Бейсиком, так и драйвер на Форте работает только с Фортком. Для чего я решил привести такой экзотический пример? С его помощью происходит убиение сразу двух зайцев. Первый заяц - это машинные коды или Ассемблер, он для начинающих - большой и страшный - хуже волка. Второй - язык программирования Форт - им (начинающим), вероятно, вообще не знаком и в данном случае происходит, скорее всего, его презентация, а уж потом расправа. Профессионалы программирования на Ассемблере, Паскале и Си этот раздел могут смело опустить. Если у первых машиннокодовые вставки вызовут, разве что улыбку, то остальные настолько увлечены своими средствами программирования, что любой альтернативный язык ими воспринимается, чуть ли не как оскорбление лучших чувств и вызывает, если не раздражение, то уж недоверие - точно.

Как видно из вышесказанного, автор этих строк, как раз и является приверженцем одного из альтернативных языков - Форты. Но в данном случае я не собираюсь переманивать Вас, почтенный читатель, на свою сторону, охаивать и подвергать сомнению существующие системы, методы и способы программирования и агитировать "за Форт". Желаю всего лишь высказать своё мнение о нем и, быть может, пробудить интерес. Любое творение, плохое или хорошее, заслуживает хотя бы простейшего ознакомления с ним (иначе как узнать какое оно), тем более, претендующее на такое высокое звание, как язык.

Немного о том, почему именно Форт. Он заинтересовал меня, прежде всего, своей необычностью и каким-то изяществом. Это и обратная польская нотация, когда вместо $3+5$ нужно писать $5\ 3\ +$ да ещё с пробелами, многими воспринимаемая как нагрузка, и наличие, одновременно, свойства компилятора и интерпретатора, и малый объём занимаемой памяти - всего 7 килобайт (!), и главное, высокая скорость вычислений. При этом в нём благоприятно сочетаются положительные свойства языков высокого и низкого уровней. От Ассемблера он приобрёл скорость и способность работы на уровне железа, что дает ему возможность служить операционной системой, а от языков высокого уровня простоту написания и отладки программ.

Сам процесс программирования на Форте сильно отличается от общепринятого. Вы не пишете программу, подгоняя её под известные операторы того или иного языка, а видоизменяете и дополняете сам язык так, чтобы он наиболее соответствовал Вашей задаче и её решению. Если представить программу зданием, которое нужно построить, Вас - строителем, а базу, оборудование и материалы - как средства и возможности самого языка, то Вы, приступая к строительству, сначала изготавливаете кирпичи, причем именно такие, какие нужны здесь и сейчас, а уже с их помощью строите все остальное. В то время, как у других строителей в своих грузовиках имеется большой ассортимент готовых строительных материалов и изделий - кирпичей, блоков, плит и другого - у Вас этого поменьше, а многого и вообще, нет, зато сколько угодно песка, глины, воды... с помощью которых и с божьей помощью Вы можете наштамповать прямо на стройплощадке таких красавцев, какие не снились ни одному строителю. Да что кирпичи, можно изготовить целые стены, порталы, перекрытия и даже (!) новые механизмы для производства материалов. Если на других языках можно сделать отдельно стул, отдельно стол, то на Форте Вы можете создать себе целую рабочую комнату по своему вкусу и со своими предметами. Это может быть и Ассемблер, и текстовый редактор, и графический пакет, и все что пожелаете, причём средствами только лишь самого языка.

Форт язык добрый и доверчивый, как младенец. Он разрешает Вам делать с ним всё, что

угодно, практически не проверяя Ваших действий. Вы должны сами заботиться о нём, тщательно проверяя границы массивов, стек и, вообще, быть более внимательным, чем, скажем, программируя на Бейсике. Зато Форт ничего от Вас и не скрывает - ни своей структуры, ни принципов работы, а, наоборот, поощряет их изучение и предполагает их использование в Ваших программах. Готовая программа на Форте не есть нечто отдельное, как в других языках: вот компилятор, вот интерпретатор, вот сама программа - это сам язык, только расширенный и измененный Вашими стараниями. Это единый кодовый блок, пронизанный множеством связей. Форт как живой организм, который Вы растите и чем талантливее родители, тем лучше их отпрыск.

Конечно, есть и у него недостатки, впрочем, о них, как и о других его достоинствах, Вы можете узнать из специальных источников, которые приведены в разделе литература. Я пользуюсь кассетной версией 1982 года ZX-FORTH 1.1 фирмы Artie Computing, которую адаптировал к диску. В таком виде имеется возможность работать с любым из 255 экранов (блоков) по одному килобайту каждый, размещённых на диске.

<pre> SCR # 10 0 (MOUSE) 1 : code create here here 2 - !; 2 : c; current @ context ! smu dge ; hex 3 code IMouse e5 c, c5 c, 01 c , 7f c, 00 c, 26 c, 93 c , ed c, 61 4 c, c1 c, e1 c, c3 c, 84 c, 6 2 c, c; 5 code XMouse e5 c, c5 c, 01 c , 3f c, 00 c, ed c, 68 c, 01 c, 5d 6 c, 00 c, 26 c, 10 c, ed c, 6 1 c, 26 c, 00 c, c1 c, e3 c, c3 c, 7 84 c, 63 c, c; 8 code YMouse e5 c, c5 c, 01 c , 3f c, 00 c, ed c, 68 c, 01 c, 5d 9 c, 00 c, 26 c, 00 c, ed c, 6 1 c, c1 c, e3 c, c3 c, 84 c, 62 c, 10 c; 11 code SMouse e5 c, c5 c, 01 c , 5d c, 00 c, ed c, 68 c, 26 c , 00 12 c, c1 c, e3 c, c3 c, 84 c, 6 2 c, c; 13 decimal --> 14 15 </pre>	<pre> SCR # 11 0 (MOUSE) 1 16 variable X 10 variable Y 22528 c@ constant C 0 variable S 2 0 variable XX 0 variable YY 3 : ADR 32 * + 22528 + c! 4 : MOUSE begin 7 X @ Y @ ADR begin XMouse 8 - XX ! YMouse 8 - Y Y 5 ! SMouse S ! S @ XX @ 0 = + Y Y @ 0 = + 5 = while repeat C X @ 6 Y @ ADR XX @ X +! YY @ -1 * Y +! X @ 31 > 7 if 31 X ! then X @ 0 < if 0 X ! then Y @ 23 > if 23 Y ! then 8 Y @ 0 < if 0 Y ! then S @ 3 < until ; 9 IMouse 10 11 12 13 14 15 </pre>
--	---

Хвалебную оду я пропел, теперь непосредственно о программе. В отличие от Бейсиковского собрата этот драйвер работает значительно быстрее и, даже, чуть-чуть быстрее, чем Бейсик-драйвер после компиляции. Впечатление оставляет о себе самое приятное и занимает в памяти после трансляции 446 байт (для сравнения, Бейсик-программа после компиляции занимает 613 байт). Саму программу можно условно разделить на две части. Первая - это создание новых слов для работы с портами (изготовление кирпичей). Вторая - основная часть, непосредственно использующая их. Они для наглядности так и располагаются - первая на 10-м экране, а вторая на

11-м.

В стандартной поставке языка нет слов, работающих с портами ввода-вывода, поэтому пришлось написать на Форте (но в кодах!) несколько новых, необходимых для конкретных целей.

IMouse - инициализация PIO (аналог OUT 127,147).

XMouse - считывание X-координаты (аналог IN 63).

YMouse - тоже самое для Y.

SMouse - опрос состояния клавиш КН1 и КН2 (аналог IN 95).

Слова code и c; являются служебными. Hex и decimal переключают транслятор языка в шестнадцатеричную и десятичную системы считывания соответственно.

Вторая часть, если присмотреться внимательно, является почти полным аналогом нашего первого драйвера на Бейсике. Здесь, кроме аналогичных переменных X, Y и других, имеется переменная S, которая содержит результат опроса клавиш. Выход из цикла происходит при нажатии хотя бы одной из них. Собственно говоря, весь драйвер это одно слово Mouse. И Вы можете вставлять его в любое место своей программы. Так же как и драйвер на Бейсике, драйвер на Форте открыт для любых изменений, переделок и доработок, но в отличие от первого не топчется на месте, а обрабатывает положения курсора быстро и четко.

Если работу слова Mouse ещё могут понять хотя бы те, кто знаком с языком, то слова IMouse, XMouse и так далее, останутся за пределом понимания большинства, за исключением, быть может, какого-нибудь фаната, выучившего наизусть в результате длительного общения с Ассемблером коды процессора Z-80. Поэтому доведу до Вас смысл и содержание этих четырёх слов. Итак, слово IMouse. На Ассемблере оно выглядит так:

```
PUSH HL ;сохранение регистров на стеке
PUSH BC
LD BC,#007F ;в регистр BC адрес порта 127
LD H,#93 ;в регистр H число 147
OUT (C),H ;запись 147 по порту 127
POP BC ;восстановление регистров со стека
POP HL
JP #6284 ;возврат к Форт-интерпретатору
```

Слово YMouse.

```
PUSH HL ;сохранение регистров на стеке
PUSH BC
LD BC,#003F ;в регистр BC адрес порта 63
IN L,(C) ;чтение порта 63 с занесением Y-координаты в L
LD BC,#005D ;в регистре BC адрес порта 95
LD H,#00 ;
OUT (C),H ;запись в порт B нуля (сброс) с одновременным
;переключением мультиплексора на Y-координату
POP BC ;восстановление регистра BC
EX (SP),HL ;обмен регистра HL со стеком, в результате
;чего Y-координата оказывается на вершине
;стека, а регистр HL восстанавливается
JP #6284 ;возврат к Форт-интерпретатору
```

Слово XMouse.

```
PUSH HL ;сохранение регистров на стеке
PUSH BC
LD BC,#003F ;в регистр BC адрес порта 63
IN L,(C) ;чтение порта 63 с занес. X-координаты в L
LD BC,#005D ;в регистре BC адрес порта 95
LD H,#10 ;
OUT (C),H ;переключение мультиплексора на Y-координату
LD H,#00
POP BC ;восстановление регистра BC
EX (SP),HL ;обмен регистра HL со стеком, в результате
```

```

;чего X-координата оказывается на вершине
;стека, а регистр HL восстанавливается
JP #6284 ;возврат к Форт-интерпретатору
Слово SMouse.
PUSH HL ;сохранение регистров на стеке
PUSH BC
LD BC,#005D ;в регистре BC адрес порта 95
IN L,(C) ;чтение порта 95 с занесением результата в L
LD H,#00
POP BC ;восстановление регистра BC
EX (SP),HL ;результат опроса клавиатуры на вершину стека
JP #6284 ;возврат к Форт-интерпретатору

```

Результат опроса порта 95 может быть таким:

- 3 - не нажата ни одна клавиша,
- 2 - нажата клавиша КН2,
- 1 - нажата клавиша КН1.

Некоторые комментарии. Поскольку Форт является стековым языком, и в данной реализации стек его параметров и машинный стек совпадают, то проще и удобнее любые значения из машинных кодов в Форт передавать именно через него - стек.

Оттранслированные коды любого из этих фрагментов и есть то, что записывается в определении соответствующего слова. Например, в каждом из них первыми записываются коды "e5" и "c5", что в переводе с машинного кода на Ассемблер означает PUSH HL и PUSH BC, а "с," это слово самого Форты, означающее запись предыдущего байта в следующую свободную ячейку памяти словаря. Можно привести аналогию со списками DADA для Бейсика, в которые Вы записываете машинные команды, а оператор READ в совокупности с POKE, NEXT и FOR переносит его в ячейки памяти, указанные Вами. При обращении к этой программе Вы набираете заветное RANDOMIZE USR, не забыв указать адрес. На Форте слово "code" - это и DADA, и READ, и POKE, и так далее. При этом Вам не нужно следить за адресами, транслятор сам выделит в памяти место под этот фрагмент. Обращение же или вызов нужной Вам подпрограммы, а теперь слова - по его имени, опять-таки без всяких адресов. Вот так просто, красиво и лаконично.

Временная характеристика драйвера подобна характеристике драйвера на Бейсике, подготовленного к компиляции. Время Т_о практически равно времени выполнения всей программы, а Т_м - всего лишь времени выполнения следующей последовательности XMouse 8 - XX !, что аналогично Бейсиковскому LET XX=IN 63-8.

4.3. Драйвер на Бейсике для попиксельных перемещений.

Быстро бегающий по экрану квадратик, чутко реагирующий на малейшее движение кисти Вашей руки, впечатляет. Но, согласитесь, как-то мало от этого проку и не видно пока конкретных результатов всей канители с подключением. Чтобы не разочаровывать Вас предлагаю разобраться и запустить следующий вариант драйвера, являющегося, одновременно, и программой рисования кривых произвольной формы. Программа очень коротенькая, но эффектная и, надеюсь, окажет на Вас благотворное влияние.

В этом варианте клавиши манипулятора имеют несколько расширенный потенциал своих действий. Левая работает как триггер и нажатие на нее приводит к переключению режима работы пера, являющегося, по совместительству, и курсором. Он или оставляет за собой цвет чернил или нет. Вернее, во втором случае он тоже оставляет след, но имеющий цвет бумаги и, поэтому, в этом режиме его можно использовать как стирательную резинку. После срабатывания правой раздается звуковой сигнал, которым программа сообщает об ожидании дальнейшего нажатия клавиш. Левая в этом случае очистит экран и вернет Вас в программу, а правая выгрузит на ленту картинку и снова запустит программу.

```

5 LET V=0: LET X=100: LET Y=100: OUT 127,147
10 LET X1=IN 63-8: OUT 95,16: LET Y1=IN 63-8
20 LET X=X+X1: IF X<0 THEN LET X=0
30 OUT 95,0: IF X<255 THEN LET X=255

```

```

40 LET Y=Y+Y1: IF Y<0 THEN LET Y=0
50 IF Y>175 THEN LET Y=175
60 PLOT X,Y
70 LET S=IN 95: IF S<3 THEN GO SUB 100
80 OVER V: PLOT X,Y: OVER 0
90 GO TO 10
100 IF S=2 THEN LET V=NOT V: RETURN
110 BEEP 0.5,10
120 IF IN 95=1 THEN INPUT "Name "; LINE A$: SAVE A$ SCREEN$: RETURN
130 IF IN 95=2 THEN CLS: RETURN
140 GO TO 120

```

За счёт большой величины времени цикла, обусловленной в основном операцией PLOT, вместо гладких и непрерывных кривых Вы получите, особенно при быстрых перемещениях, пунктирные. Хотя с художественной точки зрения этот эффект или, скорее всего, дефект выглядит неплохо - его при желании можно легко устранить компиляцией.

Так как целочисленный компилятор понимает не все операторы Бейсика, в том числе SAVE, и NOT, то в программу необходимо внести некоторые изменения и дополнения. Ниже приведены строки, которые корректируются и добавляются. Кроме того, из строки 40 оператор OUT 95,0 нужно удалить.

```

10 LET X1=IN 63-8: OUT 95,16: LET Y1=IN 63-8: OUT 95,0
70 LET S=IN 95: IF S<3 THEN GO TO 100
100 IF S=2 THEN IF V=0 THEN LET V=1: GO TO 80
105 IF S=2 THEN IF V=1 THEN LET V=0: GO TO 80
120 IF IN 95=1 THEN RETURN
130 IF IN 95=2 THEN CLS : GO TO 80
140 GO TO 120
150 STOP
160 RANDOMIZE USR adr: INPUT "Name "; LINE A$: SAVE A$ SCREEN$: GO TO 160

```

Строка 150 введена для указания границы компиляции ("MCoder2" воспринимает STOP как конец программы и всё, что идёт после него, не компилирует). Параметр adr это адрес запуска откомпилированной программы, обычно равный 40000. Строка 160 служит для запуска программы, сохранения картинки и возвращения обратно. То есть после успешной компиляции Вам достаточно набрать RUN 160 или GO TO 160 и Вы в программе.

4.4. О драйверах на ассемблере.

Если все то, о чём я говорил так "долго и нудно", у Вас работает, Вы этим довольны, и оно соответствует Вашим задачам, то это просто прекрасно. Но не обольщайтесь, со временем запросы Ваши возрастут, а задачи, позволяющие их реализовать, выйдут из круга простых. Первой и весьма тривиальной (по запросам, а не по осуществлению) может оказаться обычная адаптация некоторых любимых Ваших программ к "мышиному" управлению. Конечно, если это Бейсик - вопросов нет. А если какая-нибудь Art Studi'я, или TLW, или того похлеще, игровая? Обычно в такой ситуации реальную помощь может оказать только Ассемблер. Да и, работая на обычном Бейсике, или одном из его диалектов как Вы будете действовать, если экран разбит на 64 или, что ещё интересней, 42 знакоместа? Атрибуты уже здесь не помогут. Очевидно, придется что-то куда-то печатать, а этот процесс займет гораздо больше времени, чем обычный РОКЕ. Вот и приходится брать в руки разные MONS'ы, GENS'ы и вперед.

По принципу работы драйверы на Ассемблере можно разделить на два класса - работающие как обычные подпрограммы и работающие по прерываниям второго рода. Первый класс аналогичен Бейсиковским подпрограммам, только работающим быстрее. Второй класс более интересен, поскольку драйвер, к нему относящийся, независим от основной программы, будь то Бейсик или Ассемблер (если, конечно в них самих не используется этот режим прерываний). То есть, если, а первом случае Ваш курсор бежит только тогда, когда этого хочет головная программа, а в остальное время или стоит на месте или исчезает, то во втором - хочет того программа или нет, а работа драйвера не прекращается. И Вы контролируете курсор в любой момент её выполнения. И тот и другой способ имеет свои достоинства и недостатки, но, как мне

кажется, при разработке новых программ предпочтение следует отдавать второму.

Приведу алгоритм работы одного варианта драйвера, который может служить заготовкой для обоих классов (рис. 3).

В блоке PUSH производится сохранение на стеке всех регистров, используемых драйвером в своих целях. Следующий блок - это опрос клавиш. Условие в ромбике проверяет сам факт их нажатия. И, если он имел место, то следующий шаг - обработка их кода S. Это может быть просто сохранение S в памяти и выход из подпрограммы, или что-то более сложное - тут уж на Ваше усмотрение. Если ни одна из клавиш не нажата, то считывание координаты X, затем Y с последующим сбросом (именно в такой последовательности!) и сравнение их с числом 8. В случае равенства их 8 - манипулятор стоит на месте - переход к блоку POP и возврат из подпрограммы, иначе обработка перемещения курсора и здесь же все то, что Вы считаете нужным. Блок POP это восстановление со стека всех использованных регистров, а RET ... он и есть RET -возврат.

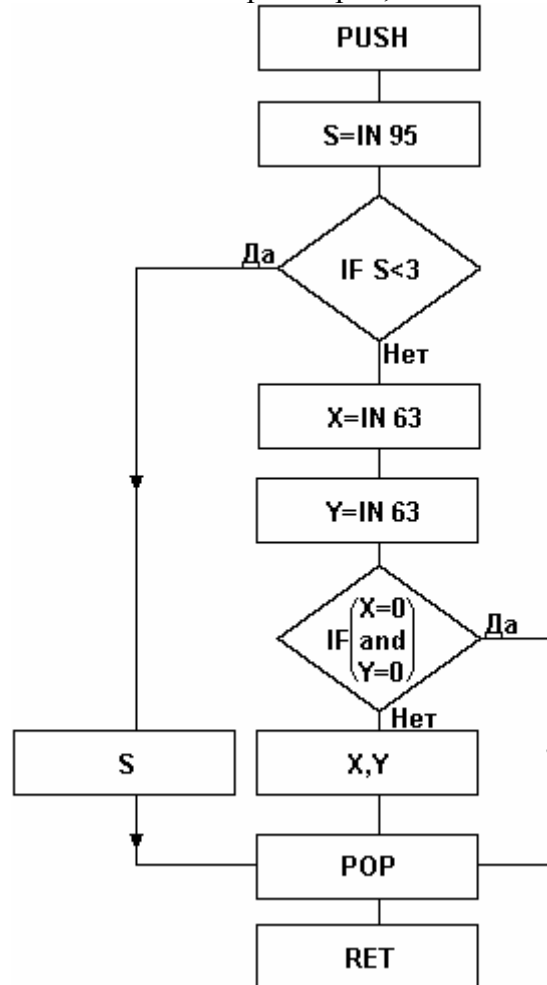


Рис. 3

Если драйвер получился довольно насыщенным и автономным, в том смысле, что выполняет все свои обязанности и, даже, более того, то есть смысл несколько изменить его алгоритм. Сделать, например, возврат в головную программу только при нажатии клавиши. Такой вариант будет служить заготовкой драйвера относящегося к первому классу (Рис. 4).

В данном разделе я не привел ни одного примера драйвера на Ассемблере. Поскольку, во-первых, это уже нечто более серьезное, чем все предыдущие забавы, во-вторых, слишком уж выходит за пределы выбранной темы о подключении, и, в-третьих, я надеюсь продолжить этот разговор и в дальнейшем: подготовить материал о работе мыши-манипулятора с конкретными программами при помощи конкретных драйверов. Если у кого-то возникнет желание двигаться в этом направлении, не дожидаясь меня или других благоприятных событий, то такое решение я буду только приветствовать.

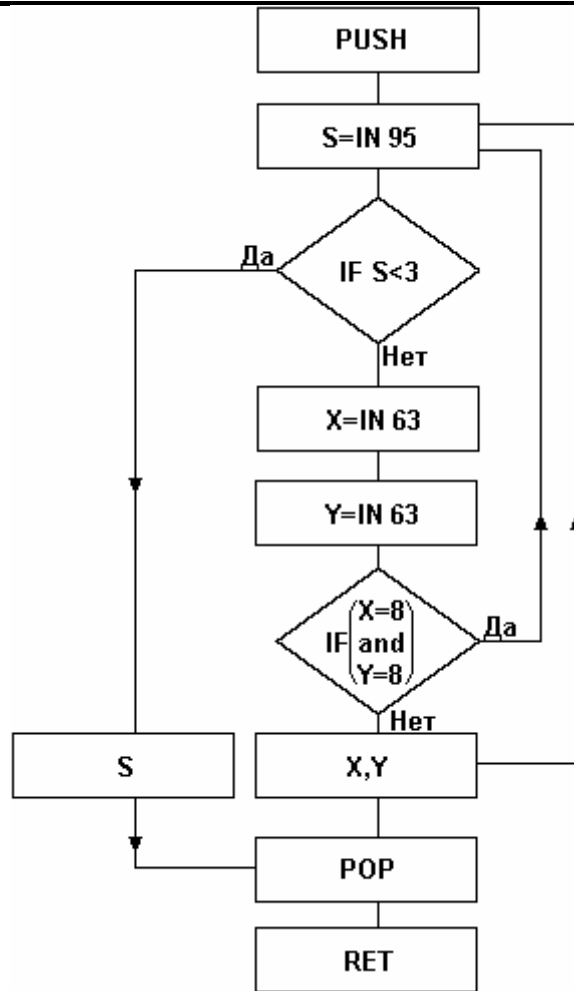


Рис. 4.

Послесловие.

Признаюсь, в мои планы не входило создавать такую монографию о "грызунах" и чуть ли не исследование в области управления. Поначалу хотел дать всего лишь схему подключения и парочку программ. Но этот процесс так захватил, что, сам того не ведая, я увлекся и ... и, наконец, разобрался с принципом работы устройства и всем остальным. Поскольку до этого подключение и программную поддержку делал на полуинтуитивном уровне, почти "методом тыка". Теперь же все это выглядит вполне научно и заслуживает, если уж не похвалы, то хотя бы нескольких критических замечаний.

Зачем я все так подробно описывал? Ну, во-первых, раз это интересно мне, то, быть может, найдется кто-то, кому эти заметки тоже придется по душе. Во-вторых, хотел дать максимум информации для самостоятельной работы, дабы начать заполнять пробел в вопросе "мышефикации" Spessу. И, в-третьих, пробудить интерес к Форту как языку программирования на все случаи жизни и операционной системе. Не знаю, удалось что-то мне или нет, но если Вы хотя бы дочитали до конца, то и на том спасибо.

Абсолютно не рассчитываю, что данный "мышинный интерфейс" станет стандартом. Моей задачей являлось, скорее всего, дать толчок или, как сейчас принято говорить, пригласить к диалогу по этому вопросу, или проблеме, или - как хотите. Думаю, что будут созданы интерфейсы для каждой из имеющихся на рынке мыши и написаны свои драйверы - это не сложно. Главный вопрос состоит в их взаимозаменяемости и совместимости при работе с каждой конкретной программой, или, по крайней мере, незначительной переделке программной поддержки, что предъявляет определенные требования к разработчикам системных, прикладных и игровых программ, если они хотят видеть свои детища законченными и пользующимися широким спросом программными продуктами.

В заключение хочу пожелать Вам удачи во всем. Буду рад любой обратной связи.

Литература.

1. "Диалекты Бейсика для ZX-Spectrum". Под ред. Родионова Н., Ларченко А. МП "Питер" (Piter Ltd), 1992 г. - 320 с.
2. Ларченко А., Родионов Н. "ZX-Spectrum для пользователей и программистов". МП "Питер" (Piter Ltd), 1991 г. - 160 с.
3. Семенов А. "Программирование на языке Форт". - М.: Радио и связь, 1991 г. - 204 с.
4. Баранов С.Н., Ноздрунов Н.Р. "Язык Форт и его реализации". - Л.: Машиностроение. 1988 г. - 137 с.
5. Броуди Л. "Начальный курс программирования на языке Форт". Пер. с англ. - М.: Финансы и статистика, 1990 г. - 352 с.
6. Келли М., Спайс Н. "Язык программирования Форт". Перевод с англ. - М.: Радио и связь, 1993г. - 320 с.

СИСТЕМНЫЕ ПРОГРАММЫ

Расширение функций оператора PRINT.

© 1994 Куковякин С.В., п. Залесье, Ивановской область

Предлагаю программу в машинных кодах, расширяющую возможности оператора PRINT в ПЗУ. Новые функции включаются "печатанием" соответствующих управляющих кодов, которые Вы можете изменить по собственному вкусу:

- ✓ переключение на русский и английский FONT'ы;
- ✓ печать шрифтом высотой 8 и шириной от 3 до 7 пикселей;
- ✓ печать шрифтом высотой 16 и шириной 8 пикселей;
- ✓ печать 2-х типов меню с тенью или без неё;
- ✓ печать собственных или системных токенов ключевых слов.

Сюда также входит подпрограмма, выдающая номер выбранной опции из меню и подпрограмма, похожая на ПЗУ-шную 203CH, только требующая для своей работы в регистре А - номер печатаемого стринга и в ОЗУ - таблицу начальных адресов этих стрингов.

А теперь перехожу к описанию самой программы. Список новых управляющих кодов:

128 - включает английский FONT.

129 - включает русский FONT.

130 - включает печать шрифтом шириной от 3 до 7 пикселей (ширина символов задается в операторе TAB, например, PRINT TAB 5 установит ширину символов в 5 пикселей).

131 - включает печать шрифтом шириной 8 и высотой 16 пикселей.

132 - включает стандартную печать.

133 - печатает меню 1 типа.

134 - печатает меню 2 типа.

Коды 133 и 134 требуют после себя наличия 6-ти байтов в таком порядке: 1 байт - цвет PAPER, 1 байт - цвет INK, 2 байта - координаты верхнего левого угла меню (как в операторе AT), 1 байт - высота меню в знаках, 1 байт - ширина в знаках.

135 - последнее напечатанное меню окаймляется тенью.

Листинг процедуры, при помощи ассемблера ZEUS: набранный

```

00010      ORG 45000      00250      EX  DE,HL
00020      ENT      00260      JP   (HL)
00030 GENERAL PUSH AF  00270 TABLE DW  FONT1
00040      LD  A, (CONTROL) 00280      DW  CODE130
00050      LD  B,A      00290      DW  CODE131
00060      POP AF      00300      DW  CODE132
00070      DEC B      00310      DW  MENU1
00080      JR  Z, PRINT  00320      DW  MENU2
00090      CP  128      00330      DW  DARK
00100      JR  C, PRINT  00340 ; .....
00110      JR  Z, FONT0  00350 PRINT  PUSH BC
00120      CP  165      00360      PUSH DE
00130      JP  NC, TOKEN 00370      PUSH HL
00140      CP  136      00380      LD  HL, (ADDRESS)
00150      JR  NC, PRINT 00390      CALL 111
00160      LD  HL, TABLE 00400      LD  HL, (23633)
00170      SUB 129      00410      LD  BC, GENERAL
00180      SLA A      00420      LD  E, (HL)
00190      LD  D, 0      00430      LD  (HL), C
00200      LD  E, A      00440      INC  HL
00210      ADD HL, DE    00450      LD  D, (HL)
00220      LD  E, (HL)  00460      LD  (HL), B
00230      INC  HL      00470      LD  A, B
00240      LD  D, (HL)  00480      CP  D

```

СИСТЕМНЫЕ ПРОГРАММЫ

00490	JR	NZ, CHNG	01040	Change1	LD	C, 146
00500	LD	A, C	01050	LOOP1	LD	A, (WINDOW)
00510	CP	E	01060		DEC	A
00520	JR	Z, EXIT	01070		LD	(WINDOW), A
00530	CHNG	LD	(ADDRESS), DE	01080	JR	NZ, NLINE
00540	EXIT	POP	HL	01090	LD	C, 153
00550		POP	DE	01100	CALL	PRAT
00560		POP	BC	01110	LD	C, 149
00570		RET		01120	JR	NOINCR
00580	;		01130	NLINE	DEC	C
00590	FONT0	LD	HL, 15360	01140	DEC	C
00600		LD	BC, CUTER1-32	01150	CALL	PRAT
00610		JR	FONT1+6	01160	LD	A, D
00620	FONT1	LD	HL, FNT1-256	01170	CP	E
00630		LD	BC, CUTER2-32	01180	JR	Z, INCR
00640		LD	(23606), HL	01190	CP	2
00650		LD	(Cuter+1), BC	01200	JR	Z, INCR
00660		RET		01210	JR	NOINCR
00670	CODE130	LD	HL, PRTPIX	01220	INCR	INC C
00680		JR	CODE132+3	01230		INC C
00690	CODE131	LD	HL, PRT16	01240		INC C
00700		JR	CODE132+3	01250	NOINCR	DEC D
00710	CODE132	LD	HL, 2548	01260	JR	NZ, LOOP1
00720		LD	(ADDRESS), HL	01270	LD	(ROW), HL
00730		RET		01280		RET
00740	;		01290	PRAT	CALL	PRTHLC
00750	MENU1	CALL	MENUSET	01300		LD A, (WIDTH)
00760		LD	B, 6	01310		DEC A
00770		SUB	B	01320		DEC A
00780		LD	(TIME), A	01330		LD B, A
00790		LD	A, 146	01340		INC C
00800		LD	(Change1+1), A	01350	LOOP2	CALL PRTC
00810		LD	A, 4	01360		DJNZ LOOP2
00820		LD	(WINDOW), A	01370		INC C
00830		JR	MENU	01380		INC L
00840	MENU2	CALL	MENUSET	01390		JR PRTC
00850		LD	B, 3	01400	PRTHLC	LD A, 22
00860		SUB	B	01410		CALL PRINT
00870		LD	(TIME), A	01420		LD A, L
00880		LD	A, 158	01430		CALL PRINT
00890		LD	(Change1+1), A	01440		LD A, H
00900		LD	(WINDOW), A	01450		CALL PRINT
00910	MENU	CALL	CODE132	01460	PRTC	LD A, C
00920		LD	A, 17	01470		CALL PRINT
00930		RST	16	01480		RET
00940		LD	A, (PAPER)	01490	MENUSET	LD DE, PAPER
00950		RST	16	01500		LD HL, (TEMPDE)
00960		LD	A, 16	01510		INC HL
00970		RST	16	01520		LD BC, 6
00980		LD	A, (INK)	01530		LDIR
00990		RST	16	01540		DEC HL
01000		LD	HL, (ROW)	01550		LD (TEMPDE), HL
01010		LD	A, (HIGH)	01560		LD HL, (TEMPBC)
01020		LD	D, A	01570		LD BC, 6
01030		LD	E, A	01580		SBC HL, BC

СИСТЕМНЫЕ ПРОГРАММЫ

01590	LD	(TEMPBC), HL	02140	RET	C
01600	LD	HL, (POS)	02150	;	
01610	LD	(ROW), HL	02160	LD	B, 0
01620	LD	A, (TIME)	02170	LD	C, A
01630	LD	(HIGH), A	02180	Cuter	LD HL, CUTER1-32
01640	PUSH	AF	02190	ADD	HL, BC
01650	ADD	A, L	02200	LD	E, (HL)
01660	DEC	A	02210	LD	BC, (23606)
01670	LD	(POS), A	02220	LD	H, 0
01680	POP	AF	02230	LD	L, A
01690	RET		02240	ADD	HL, HL
01700	;	02250	ADD	HL, HL
01710	DARK	LD HL, (ROW)	02260	ADD	HL, HL
01720	INC	H	02270	ADD	HL, BC
01730	LD	C, 143	02280	LD	C, 8;high
01740	CALL	PRTHLC	02290	LD	A, (SIZE)
01750	LD	A, (WIDTH)	02300	ADD	A, 247
01760	DEC	A	02310	LD	D, A;skolko cut
01770	LD	B, A	02320	NBYTE	PUSH DE
01780	LOOP3	CALL PRTC	02330	LD	A, D
01790	INC	H	02340	SUB	247
01800	DJNZ	LOOP3	02350	LD	B, A
01810	LD	A, (HIGH)	02360	LD	A, (HL)
01820	LD	B, A	02370	NBIT	INC D
01830	LOOP4	CALL PRTHLC	02380	CALL	NZ, CUT
01840	DEC	L	02390	DEC	D
01850	DJNZ	LOOP4	02400	RLA	
01860	RET		02410	CALL	C, PLOT
01870	;	02420	PUSH	AF
01880	TOKEN	SUB 165	02430	LD	A, (COORDX)
01890	LD	DE, TOKENS	02440	INC	A
01900	JP	#0C0A	02450	LD	(COORDX), A
01910	;		02460	POP	AF
01920	FNT1	EQU 40000	02470	DJNZ	NBIT
01930	ADRESS	DW 2548	02480	;	
01940	ROW	DB 0	02490	INC	HL
01950	COL	DB 0	02500	PUSH	HL
01960	HIGH	DB 10	02510	LD	HL, (COORDX)
01970	WINDOW	DB 3	02520	DEC	H
01980	;		02530	LD	A, (SIZE)
01990	TOKENS	DB 255	02540	SUB	L
02000	DM	/RN/	02550	NEG	
02010	DB	196	02560	LD	L, A
02020	DM	/INKEY/	02570	LD	(COORDX), HL
02030	DB	164	02580	POP	HL
02040	;	02590	POP	DE
02050	;	PRINT WIDTH 3-8 PIXELS	02600	DEC	C
02060	;		02610	JR	NZ, NBYTE
02070	PRTPIX	CP 13	02620	;	
02080	JP	Z, ENTER1	02630	CHECK1	LD HL, (COORDX)
02090	CP	22	02640	LD	A, (SIZE)
02100	JP	Z, ATT1	02650	LD	B, A
02110	CP	23	02660	ADD	A, A
02120	JP	Z, TAB	02670	DEC	A
02130	CP	32	02680	ADD	A, L

СИСТЕМНЫЕ ПРОГРАММЫ

02690	JR	C, NEWROW	03240	AT1	LD	DE, AT2
02700	LD	A, B	03250		JP	#0A7D
02710	ADD	A, L	03260	AT2	LD	DE, (ATCLOSE)
02720	LD	L, A	03270		CALL	#0A70
02730	LD	A, 8	03280		LD	HL, (23566)
02740	ADD	A, H	03290		LD	(COORDX), HL
02750	LD	H, A	03300		XOR	A
02760	JR	SETXY	03310		LD	(CONTROL), A
02770	NEWROW	INC	03320		RET	
02780	JR	Z, SET00	03330		;	
02790	DEC	H	03340	TAB	LD	DE, TAB1
02800	LD	A, 6	03350		JP	#0A80
02810	CP	H	03360	TAB1	LD	DE, PRTPIX
02820	JR	NC, SET00	03370		CALL	#0A7D
02830	LD	L, 0	03380		LD	A, (23566)
02840	JR	SETXY	03390		LD	(SIZE), A
02850	SET00	LD	03400		RET	
02860		CALL	03410		;	
02870	SETXY	LD	03420	WAITCLS	PUSH	BC
02880		RET	03430		PUSH	HL
02890	;		03440		LD	BC, 0
02900	CUT	SLA	03450		CALL	7997
02910		RET	03460		CALL	8020
02920		RLA	03470		JR	NC, BREAK
02930		INC	03480		LD	DE, 16385
02940		RET	03490		LD	HL, 16384
02950		JR	03500		LD	BC, 6143
02960	;		03510		LD	(HL), 0
02970	PLOT	PUSH	03520		LDIR	
02980		PUSH	03530		POP	HL
02990		PUSH	03540		POP	BC
03000		PUSH	03550		RET	
03010		LD	03560		;	
03020		CALL	03570	BREAK	POP	HL
03030		POP	03580		POP	HL
03040		POP	03590		POP	HL
03050		POP	03600		JP	7035
03060		POP	03610		;	
03070		RET	03620	CONTROL	DB	0
03080	;		03630	ATCLOSE	DW	2548
03090	ENTER1	LD	03640	COORDX	EQU	23728
03100		LD	03650	COORDY	EQU	23729
03110		SUB	03660	SIZE	EQU	23681
03120		NEG	03670	CUTER1	EQU	40768
03130		LD	03680	CUTER2	EQU	40864
03140		LD	03690	;	
03150		JR	03700	;	PRINT BY SIZE	8, 16
03160	;		03710	;		
03170	ATT1	LD	03720	PRT16	CP	13
03180	;		03730		JR	Z, ENTER2
03190	AT	LD	03740		CP	22
03200		LD	03750		JR	Z, ATT2
03210		LD	03760		CP	32
03220		LD	03770		RET	C
03230		JP	03780		LD	DE, (23606)

СИСТЕМНЫЕ ПРОГРАММЫ

03790	LD	L, A	04340	DJNZ	LOOP8
03800	LD	H, 0	04350	INC	DE
03810	ADD	HL, HL	04360	DEC	C
03820	ADD	HL, HL	04370	JR	NZ, LOOP7
03830	ADD	HL, HL	04380	POP	HL
03840	ADD	HL, DE	04390	RET	
03850	EX	DE, HL	04400	;	
03860	LD	BC, (COORDX)	04410	ENTER2	LD HL, (COORDX)
03870	LD	A, C	04420		LD H, 32
03880	AND	#18	04430		JR CHECK2
03890	OR	#40	04440	;	
03900	LD	H, A	04450	ATT2	LD HL, PRT16
03910	LD	A, C	04460		JP AT
03920	AND	7	04470	;
03930	OR	A	04480	NUMBER	PUSH AF
03940	RRA		04490		LD A, 2
03950	RRA		04500		CALL #1601
03960	RRA		04510		POP AF
03970	RRA		04520		LD HL, CHART-2
03980	ADD	A, B	04530		LD BC, 2
03990	LD	L, A	04540	LOOP9	ADD HL, BC
04000	PUSH	BC	04550		DEC A
04010	CALL	PRTHALF	04560		JR NZ, LOOP9
04020	LD	A, 32	04570		LD E, (HL)
04030	ADD	A, L	04580		INC HL
04040	LD	L, A	04590		LD D, (HL)
04050	JR	NC, NOSEGM	04600		INC HL
04060	LD	A, 8	04610		LD C, (HL)
04070	ADD	A, H	04620		INC HL
04080	LD	H, A	04630		LD B, (HL)
04090	NOSEGM	CALL PRTHALF	04640		LD L, C
04100		POP HL	04650		LD H, B
04110	;		04660		AND A
04120	CHECK2	INC H	04670		SBC HL, DE
04130		LD A, 31	04680		LD C, L
04140		CP H	04690		LD B, H
04150		JP NC, SETXY	04700	LOOP10	LD A, B
04160		INC L	04710		OR C
04170		INC L	04720		DEC BC
04180		LD H, 0	04730		RET Z
04190		LD A, 22	04740		LD A, (DE)
04200		CP L	04750		LD (TEMPBC), BC
04210		JP NC, SETXY	04760		LD (TEMPDE), DE
04220		LD L, 0	04770		CALL GENERAL
04230		JP SET00+3	04780		LD BC, (TEMPBC)
04240	;		04790		LD DE, (TEMPDE)
04250	PRTHALF	PUSH HL	04800		INC DE
04260		LD C, 4	04810		JR LOOP10
04270	LOOP7	LD B, 2	04820	TEMPDE	DW 0
04280	LOOP8	LD A, (DE)	04830	TEMPBC	DW 0
04290		LD (HL), A	04840	CHART	EQU 40960
04300		SRL A	04850	;
04310		OR (HL)	04860	;	MENU LINE
04320		LD (HL), A	04870	;	
04330		INC H	04880	LINE1	LD HL, 35000-6

СИСТЕМНЫЕ ПРОГРАММЫ

04890		LD	BC, 6	05440		NEG	
04900	LOOP11	ADD	HL, BC	05450		LD	L, A
04910		DEC	A	05460		LD	(POS), HL
04920		JR	NZ, LOOP11	05470		RET	
04930		LD	DE, PAPER	05480	END	LD	A, (TIME)
04940		LDIR		05490		LD	(TIME1), A
04950	;			05500		LD	HL, (POS)
04960	LINE2	CALL	BEGIN	05510		ADD	A, L
04970		CALL	NEWLINE	05520		LD	L, A
04980	INKEYS	LD	BC, 33	05530		LD	(POS), HL
04990		CALL	7997	05540		RET	
05000		LD	A, (23556)	05550	NEWLINE	LD	A, (INK)
05010		CP	65	05560		SLA	A
05020		JR	Z, DOWN	05570		SLA	A
05030		CP	81	05580		SLA	A
05040		JR	Z, UP	05590		LD	B, A
05050		CP	13	05600		LD	A, (PAPER)
05060		JR	NZ, INKEYS	05610		ADD	A, B
05070	QUIT	LD	BC, (TIME1)	05620	FILL	PUSH	AF
05080		LD	B, 0	05630		LD	DE, (POS)
05090		RET		05640		CALL	ATribut
05100	DOWN	LD	BC, #2C3C	05650		LD	A, (WIDTH)
05110		JR	UP+3	05660		LD	B, A
05120	UP	LD	BC, #2D3D	05670		POP	AF
05130		LD	HL, Change2	05680	LOOP12	LD	(HL), A
05140		LD	(HL), C	05690		INC	HL
05150		LD	HL, Change3	05700		DJNZ	LOOP12
05160		LD	(HL), B	05710		RET	
05170		LD	A, (PAPER)	05720	ATribut	LD	A, E
05180		SLA	A	05730		AND	#18
05190		SLA	A	05740		SRL	A
05200		SLA	A	05750		SRL	A
05210		LD	B, A	05760		SRL	A
05220		LD	A, (INK)	05770		OR	#58
05230		ADD	A, B	05780		LD	H, A
05240		CALL	FILL	05790		LD	A, E
05250		LD	A, (TIME1)	05800		AND	7
05260	Change2	INC	A	05810		OR	A
05270		LD	(TIME1), A	05820		RRA	
05280		LD	B, A	05830		RRA	
05290		LD	A, (TIME)	05840		RRA	
05300		CP	B	05850		RRA	
05310		CALL	C, BEGIN	05860		ADD	A, D
05320		DEC	B	05870		LD	L, A
05330		CALL	M, END	05880		RET	
05340		LD	HL, (POS)	05890	PAPER	DB	5
05350	Change3	INC	L	05900	INK	DB	1
05360		LD	(POS), HL	05910	POS	DW	#0003
05370		CALL	NEWLINE	05920	TIME	DB	21
05380		JR	INKEYS	05930	WIDTH	DB	5
05390	BEGIN	LD	A, 1	05940	TIME1	DB	21
05400		LD	(TIME1), A	05950	;	
05410		LD	HL, (POS)	05960	N1	DB	133, 5, 1, 1, 1
05420		LD	A, (TIME)	05970		DB	15, 10
05430		SUB	L	05980		DB	131, 22, 2, 2

СИСТЕМНЫЕ ПРОГРАММЫ

05990	DM	/FILENAME/	06270	DM	/MOVE/
06000	DB	130,23,5	06280	DB	22,4,21
06010	DB	22,16,127	06290	DM	/DELETE/
06020	DM	/Mega BASIC/	06300	DB	22,5,21
06030	DB	22,16,119	06310	DM	/RENAME/
06040	DM	/Beta BASIC/	06320 N3	DB	134,0,7,9,20
06050	DB	22,16,111	06330	DB	7,9,129
06060	DM	/LOGO/	06340	DB	21,1,130,23
06070	DB	22,16,103	06350	DB	5,22,184,103
06080	DM	/PASCAL/	06360	DM	/FAJL/
06090	DB	22,16,95	06370	DB	22,165,87
06100	DM	/PROLOG/	06380	DM	/KOPIROWATX/
06110	DB	22,16,87	06390	DB	22,165,79
06120	DM	/ZEUS/	06400	DM	/PO^ISTITX/
06130	DB	22,16,79	06410	DB	22,165,71
06140	DM	/GENS 4D/	06420	DM	/UDALITX/
06150	DB	22,16,71	06430	DB	22,165,63
06160	DM	/MONS 4D/	06440	DM	/PEREIMENOWATX/
06170	DB	22,16,63	06450	DB	128,132
06180	DM	/boot/	06460 N4	DB	0
06190	DB	132,16,0,135	06470 ;		
06200 N2	DB	134,2,7,0,20	06480	ORG	CHART
06210	DB	7,8	06490	ENT	
06220	DB	20,1,22,0,22	06500	DW	N1,N2,N3,N4
06230	DM	/FILE/	06510 ;		
06240	DB	20,0,22,2,21	06520	ENT	
06250	DM	/COPY/	06530 START	LD	A,1
06260	DB	22,3,21	06540	JP	NUMBER

Итак, в строках 30-260 происходит обработка поступающего на печать символа. Строки 30-80 нужны для нормальной работы оператора АТ при печати узкими символами. Если переменная CONTROL равна 1, а в это состояние она устанавливается после поступившего кода с номером 22, то сразу следует переход на процедуру PRINT. Туда же идет переход, если код символа менее 128 или лежит в диапазоне от 136 до 164. Если код больше 164, то это токен, и его печатью займется процедура TOKEN. Теперь остались только новые управляющие коды, за исключением 128-го (переход на его обработку был сделан, когда код поступившего символа проверялся, что он меньше 128). В регистр HL загружается начало таблицы, содержащей адреса процедур, обрабатывающих соответствующий код. Регистр А уменьшается на 129 и умножается на 2 так как адреса 2-хбайтные. Потом А загружается в DE и DE складывается с HL. Теперь HL указывает на место в памяти, в котором содержится адрес нужной процедуры. Этот адрес загружается в DE, происходит обмен между DE и HL и следует переход на этот адрес.

PRINT. Эта подпрограмма аналогична подпрограмме, напечатанной в ZX-РЕВЮ № 5,6 1992 на стр. 112.

FONT0. В HL загружается адрес набора символов в ПЗУ минус 256, в BC - адрес таблицы, данные которой показывают, какие столбцы в символе вырезать (это нужно при печати узкими символами; об этом позже в PRTPIX) и HL загружается в CHARS (23606), а содержимое BC подставляется после кода операции LD HL,NN в подпрограмме PRTPIX. Можно было бы завести еще одну переменную, например, CUTER и сделать так:

```
LD (CUTER), BC
LD HL, (CUTER) .
```

FONT1. Все, как и в FONT0, только другой адрес набора символов и таблицы "вырезки".

CODE130. Установка печати узкими символами.

CODE131. Установка печати символами 8 на 16.

CODE132. Установка печати стандартными символами.

Дальше начинается подпрограмма печати меню. Данные для меню передаются из

машинного кода (хотя желающие могут легко доработать программу, чтобы данные передавались из BASIC'a, например, через функцию DEF FN) от подпрограммы NUMBER. Сейчас нам нужно знать только то, что она хранит в переменной TEMPDE, где находится текст, и в TEMPBC - длину этого текста. Подпрограмма MENU передает параметры для DARK (тень от меню) и MENU LINE (выбор опции из меню). Меню строится с помощью символов UDG. Их расположение строго определенное (см. рис.1).

КОД	ВИД	КОД	ВИД	КОД	ВИД	КОД	ВИД
144	┌	150	└	156	▒	162	┐
145	═	151	─	157	▒	163	┘
146	┐	152	┌	158	▒	164	┘
147	▯	153	▯	159			
148	ПРОБЕЛ	154	—	160	ПРОБЕЛ		
149	▯	155	▯	161			

Рис.1. Символы UDG.

Печатью обоих меню занимается подпрограмма MENU. Разница состоит только в подготовительных операциях.

MENU1. Сразу вызывается подпрограмма MENUSET - данные для меню переносятся на место переменных подпрограммы MENU LINE. Затем, чтобы нормально работала подпрограмма NUMBER, TEMPDE увеличивается на 5 (откуда берется текст), а TEMPBC уменьшается на 6 (длина этого текста). Потом координаты левого верхнего угла меню копируются в переменную ROW и COL (эти переменные однобайтные и должны располагаться - сначала ROW, затем COL); высота меню копируется в переменную HIGH. Затем к строке, с которой начинается меню, прибавляется высота-1 и запоминается в POS, которая хранит координаты линии меню. Затем продолжает работу MENU1. На выходе из MENUSET в регистре A содержится высота меню. И для подпрограммы MENU LINE корректируется, из скольких пунктов (или опций) состоит меню. В данном случае 1 верхняя и 1 нижняя строка, плюс название, высотой 2 строки и 2 строки, чтобы отделить название от пунктов меню, - итого 6 строк. И в переменную TIME заносится скорректированное количество пунктов. Затем в A загружается символ UDG, определяющий правый верхний угол и подставляется на место Change1+1 (в последующем этот код будет загружаться в регистр C). Дальше в переменную WINDOW заносится число 4, которое определяет высоту "шапки" (или окна) меню, в котором печатается его название (эта переменная введена исключительно для печати 1-го меню; чтобы обойти ее действие при печати 2-го меню, ей достаточно присвоить значение большее чем 21) и следует переход на общую подпрограмму MENU. Здесь вызывается подпрограмма CODE132, включающая обычную печать (если Вы, например, до этого печатали нестандартным шрифтом), устанавливается цвет PAPER и INK, в HL загружаются начальная строка и столбец, в регистры D и E заносится высота меню и D становится счётчиком по которому определяется все ли меню напечатано, а по E определяется, что надо переходить на новый ряд из 3-х символов UDG (Тут даже можно было бы сделать проще: сравнивать D с HIGH, а E использовать как WINDOW). В общем, принцип печати такой: берётся первый ряд из трёх символов UDG, печатается 1-й символ, затем 2-й символ печатается в цикле WIDTH-2 раз, потом печатается 3-й символ. Если это первая или последняя строка, то берётся следующий ряд из трёх символов, иначе печатается этот же ряд с новой строки. По завершении печати координаты строки и столбца, находящиеся в HL, сохраняются для того, чтобы программа рисования тени знала, где рисовать.

DARK. С помощью этой подпрограммы последнее напечатанное меню "отбросит" тень. Тень печатается символом блочной графики с кодом 143. Работа подпрограммы достаточно понятна и без описания.

TOKEN. Здесь печатаются токены ключевых слов. Вы можете задать собственные токены, например, слоги слов - если Вы пишете текстовую программу, то можете сэкономить не один килобайт. Если же Вы хотите оставить стандартные токены, то строку 130 надо переписать так: JP NC,2548. А подпрограмму TOKEN можно стереть.

Прежде чем приступить к описанию подпрограммы печати узкими символами, хочу сначала рассказать о принципе печати символов, шириной меньше чем 8 пикселей, на примере буквы А. Стандартное изображение символа представляет собой шаблон размером 8x8 пикселей и задается с помощью 8-ми байт (см. рис.2).

Каждая строка шаблона - это один байт, состоящий из 8-ми битов. Чтобы напечатать букву

А шириной 5 "вырезать" по 3 бита, т.е. "вырезать" в шаблоне 3 столбца. Пусть это будут столбцы 7,5 и 4. Тогда буква А примет вид как на рис.3,4.

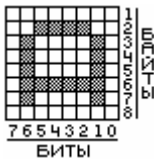


Рис.2. Шаблон.

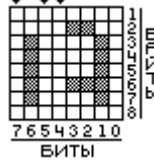


Рис.3. "Урезание" буквы.

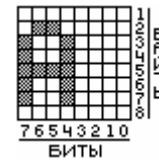


Рис.4. "Урезанная" буква.

"Вырезание" столбцов в программе выполняется просто пропуском ненужных битов. Чтобы программа знала, какие столбцы в каждой букве можно вырезать, создаются 2 таблицы (для английского и русского шрифтов), каждая длиной 96 байт (символьный набор состоит из 96 символов). Включенные в байте биты показывают, какие столбцы можно "вырезать". Вообще-то, для каждого шрифта желательно иметь 2 такие таблицы: для режимов - 6-7 пикселей ширина и 3-5 пикселей так как с одной таблицей изображение некоторых букв в режиме 6 пикселей получается не очень хорошее (например, Т, V, X, Y). Но иметь 2 таблицы из-за нескольких букв - большая роскошь. Можно сделать так, чтобы программа смотрела - если ширина печати равна 6, и печатаемая буква одна из вышеуказанных, то для неё устанавливается новый байт, показывающий, какие столбцы "вырезать". В программе на эти 2 таблицы указывают переменные CUTER1 (для английского шрифта) и CUTER2 (для русского). Создавать таблицы удобнее из BASIC'a таким образом:

```
10 DATA BIN 11110000, " "
```

.....

```
330 DATA BIN 10111000, "A"
```

```
340 DATA BIN 10111000, "B"
```

И с помощью следующей программки занести в память:

```
2 LET CUTER=40768: REM адрес расположения таблицы в памяти
```

```
4 FOR X=0 TO 95: READ A,A$: POKE CUTER+X,A: NEXT X
```

Да, ещё насчёт экономии памяти: символы с кодами от 32 до 63 одинаковы в обоих шрифтах, поэтому для русского шрифта можно хранить в памяти только изображение букв, а для печати символов, с кодом меньшим 64, использовать символьный набор ПЗУ (и сделать это довольно просто, а экономия приблизительно равна 200 байтам - $(64-32)*8=256$ и минус примерно байт 40 на доработку программы).

PRTPIX. В строках 2070-2140 идёт обработка управляющих кодов ENTER, AT (формат данных для AT, как в операторе PLOT) и TAB (установка ширины символов). Другие управляющие коды просто игнорируются. Далее начинаются подготовительные операции для печати символа, суть которых сводится к установке регистров следующим образом: HL содержит адрес шаблона символа, E - какие биты вырезать, D - сколько этих битов вырезать, C - высота символа. Символ печатается слева направо и сверху вниз по точкам (включенный бит показывает, что надо напечатать точку на экране). Программа похожа на программу, печатающуюся на страницах ZX-РЕВЮ, поэтому остановлюсь только на нескольких моментах. В строках 2320-2610 печатается весь символ; в 2370-2470 печатается один байт (строка) бит за битом слева направо. Далее горизонтальная координата печати увеличивается на ширину символа и, если выходит за пределы экрана, устанавливается в начало строки, а строка увеличивается на 1 (точнее сказать номер строки, так как вертикальная координата уменьшается на 8 пикселей - высоту символа). Если же новая строка не помещается на экране, то происходит инициализация координат в подпрограмме SET00 (забыл упомянуть, что начало координат - это точка с координатами 0,175), причём происходит обращение к подпрограмме WAITCLS, в которой ожидается нажатие любой клавиши и затем очищается экран. WAITCLS в общем-то не нужна тем кто программирует только в кодах и заранее расписывает где, как и что будет печататься на экране, но понадобится тому, кто даст из BASIC'a команду LIST при включенном новом режиме печати.

Подпрограмма CUT занимается "вырезкой" ненужных битов из байта (при входе в неё в A - печатаемый байт, в D - сколько битов надо "вырезать", в E - какие биты "вырезать"). Делает она это следующим образом. Регистр E сдвигается влево, при этом 7-ой бит поступает в CARRY-флаг, и если бит не включен (CARRY-флаг не включен), то и "вырезать" ничего не надо, поэтому сразу

выход (RET NC) . Дальше из изображения символа "вырезается" 1 бит (RLA) и, если хватит "вырезать", то выход из подпрограммы, иначе пытается "вырезать" следующий бит (у, какая кровожадная!).

PLOT вводит координаты точки и строит её на экране при помощи подпрограммы ПЗУ. ENTER1 обрабатывает поступивший код 13. Все, что нужно для этого сделать - установить в HL текущие координаты, горизонтальную координату сделать максимальной и перейти на подпрограмму, проверяющую строку на выход за пределы экрана.

AT занимается обработкой кода 22. Так как эта подпрограмма общая для PRTPIX и PRT16, то на входе в неё в HL содержится адрес программы печати, устанавливаемой после принятия 2-ой координаты. Программа аналогична подпрограмме, находящейся в ПЗУ, но вначале заносит в переменную CONTROL единицу (так как координаты могут быть в диапазоне 0-255 и 0-175), чтобы, например, если код равен 170, программа не "подумала", что это код токена и не печатала его. Когда обе координаты приняты, CONTROL сбрасывается, а координаты, находящиеся сейчас в системной переменной TV_DATA, переносятся в переменную COORDX и COORDY.

TAB обрабатывает код 23 (как Вы помните, этим кодом управляется ширина печатаемых букв). Принимает следующий за 23-м код и загружает его в SIZE.

PRT16 - это программа печати символами 8 на 16 пикселей. Координаты задаются в стандартных знаках, как в операторе PRINT. В строках 3720-3770 происходят переходы на обработку управляющих кодов, или сразу выход из программы, если это не управляющий код, печатный символ или токен. Затем по коду символа в символьном наборе ищется изображение символа (строки 3780-3840), и теперь оно хранится в DE. Далее (строки 3860-3990) по координатам знака рассчитывается адрес в дисплейном файле (см. "Элементарная графика", ИНФОРКОМ).

В подпрограмме PRTHALF символ как бы "вытягивается" в высоту, то есть берутся первые 4 байта изображения символа и каждый байт заносится в дисплейный файл по два раза. Потом для печати нижней половины символа к адресу дисплейного файла надо прибавить 32, если при этом включился флаг переноса, то значит нужно перейти к следующему сегменту. После того, как была напечатана и нижняя половина, горизонтальная координата увеличивается и, если не выходит за пределы, то переход к следующей строке (так как здесь строка высотой 2 знака, то и увеличивается она на 2). Если строка не помещается на экране, то в HL устанавливаются начальные координаты и следует переход на SET00+3 (+3 - для того, чтобы пропустить установку начальных координат для программы печати узкими символами; о SET00 см. выше). Кстати, в подпрограмме PRTHALF в строках 4300-4320 печатаемый символ делается толще (эти строки лучше удалить, если Вы много будете печатать по-русски, так как такие буквы как "ш", "ю" и т.п. сливаются в сплошное пятно на экране).

NUMBER - эта подпрограмма служит для печати текстовых сообщений. На входе в неё в регистре A должен быть номер текстового сообщения, и в памяти должна находиться таблица из 2-байтных адресов, указывающих на начало каждого текстового сообщения. Сообщения, как и адреса, указывающие на эти сообщения, должны располагаться в памяти строго друг за другом. На начало этих адресов указывает метка CHART. Программа NUMBER сама рассчитывает длину печатаемого текста (как разницу между началами следующего и печатаемого в данный момент сообщения), и распечатает его с помощью LOOP10 (в DE - адрес, где находится текст, BC - длина текста). Пример, показывающий работу этой программы, приведен в строках 5960-6540.

MENU_LINE - эта программа предназначена для выбора пункта из меню. Войти в неё можно в двух точках: LINE2 - обычная точка входа после того, как Вы напечатали какое-либо меню (перед печатью меню подготавливаются данные для этой программы - см. MENUSET), и LINE1 - если Вас по какой-либо причине не устраивает первая (например, несколько меню уже напечатаны). Если вход в программу через LINE1, то Вам нужно подготовить где-то в памяти (адрес минус 6 загружается в HL см. строку 4 880) данные в таком виде: 1-й байт - цвет PAPER курсора, 2-й байт - цвет INK, 3-й и 4-й - координата самого нижнего пункта меню, 5-й - сколько пунктов в меню и 6-й - ширина меню. На входе в LINE1 в регистре A содержится номер меню.

В строках 4960-4970 выделяется 1-ый пункт цветом, инвертированным к цвету меню (т.е. PAPER становится INK и наоборот). Далее в 4980-4990 выдерживается пауза и в строках 5000-

5060 опрашиваются клавиши: Q-вверх, A-вниз, ENTER - подтверждение выбора.

QUIT - сюда мы попадаем, если нажата клавиша ENTER. В регистр BC вводится номер текущего пункта и осуществляется выход из программы (т.е. можно из BASIC'а дать команду LET A=USR LINE2, где LINE2 - адрес программы LINE2, полученный в результате ассемблирования, и в переменной A Вы получите номер пункта меню).

Если нажаты клавиши вниз или вверх, то попадаем на подпрограмму DOWN или UP соответственно. Подпрограммы передвижения курсора вниз и вверх отличаются только тем, что при движении курсора вниз, вертикальная координата курсора и текущий номер пункта увеличиваются, а при движении вверх все происходит наоборот. Поэтому в этих подпрограммах происходит только настройка общей программы (подменяются операции в строках, помеченных как Change2 и Change3: для подпрограммы DOWN - это INC A (код #3C) и INC L (код #2C); для UP - DEC A (код #3D) и DEC L (код #2D)). Затем курсор передвигается на следующий пункт таким образом: текущий курсор закрашивается цветом меню (в строках 5170-5230 вычисляется байт для закраски, для этого цвет PAPER умножается на 8 и к нему прибавляется цвет INK, и подпрограмма FILL, рассчитав с помощью ATribut по координатам адрес в файле атрибутов, сотрёт текущий курсор. Потом номер текущего пункта, хранящегося в TIME1, увеличивается или уменьшается на единицу (в зависимости от того, куда перемещается курсор - вниз или вверх) и проверяется, что курсор не вышел за пределы меню. Если курсор находился на последнем пункте и при этом была нажата клавиша вниз, то курсор перепрыгнет на первый пункт меню (установка курсора с последнего пункта на первый выполняется подпрограмма BEGIN). Если курсор находился на первом пункте меню и при этом была нажата клавиша вверх, то курсор перепрыгнет на последний пункт (подпрограмма END). В завершение выделяется новый пункт и переход на опрос клавиатуры.

Включение новых функций для оператора PRINT выполняется открытием нового пользовательского канала (как в ZX-РЕВЮ № 5,6 1992):

```
5 LET START=PEEK 23635+256*PEEK 23636-1
10 LET L=START-256*INT (START/256) : LET H=INT (START/256)
15 FOR X=23296 TO 23304: READ A: POKE X,A: NEXT X
16 RANDOMIZE USR 23296
20 FOR X=START TO START+5: READ A: POKE X,A: NEXT X
21 POKE 23578,21
25 DATA 1,5,0,33,L,H,195,85,22
30 DATA L,H,196,21,83
```

В строках 25-30 L и H - младший и старший адрес программы в кодах.

И ещё несколько слов в заключение. Программа не проверяет вводимые координаты на допустимые пределы, но это не трудно сделать тем, кто захочет.

Подпрограмма печати узкими символами не сможет напечатать в двух нижних строках, так как использует процедуру ПЗУ для печати точек. Чтобы устранить этот недостаток, Вы можете воспользоваться программой печати точек, напечатанной в "Элементарной графике" ИНФОРКОМа. Если Вы хотите уменьшить и высоту символов до 7 пикселей, то надо только заменить число 8 (сейчас высота 8 пикселей) на 7. Если же до 6, то тут придется немного труднее (создать ещё одну таблицу, указывающую какую горизонтальную строку можно выкинуть, и дополнить саму программу).

Программу печати меню можно сократить почти вдвое, если не использовать символы UDG, а печатать меню пробелом и окантовывать его при помощи PLOT-DRAW. Но преимущество UDG в том, что вид меню можно сделать разнообразным.

Для демонстрации работы новой процедуры печати предназначена демонстрационная Бейсик-программа, фрагменты работы которой Вы видите на рис. 5,6:

В приводимых ниже листингах Бейсик-программ подчеркнутые символы в скобках вводятся в графическом режиме. Так, например, выражение (g8) означает, что сначала надо включить графический регистр (CAPS SHIFT+9), затем нажать клавишу "8" (будет напечатан символ псевдографики с кодом 128). Для набора (q1) надо в графическом регистре нажать "1" (код 129) и т.д.



Рис. 5.

```

1 BORDER BIN : PAPER BIN : INK 7: CLEAR 29999
2 RANDOMIZE USR 15619: REM : LOAD "UDG MENU"CODE 65368
3 RANDOMIZE USR 15619: REM : LOAD "DEMO"CODE
5 POKE 23562,NOT BIN
10 DATA 1,5,0,33,58,93,195,85,22
11 DATA 40,160,196,21,83
12 RESTORE 10: FOR n=23296 TO 23304: READ a: POKE n,a: NEXT n: RANDOMIZE USR
  23296: FOR n=23866 TO 23870: READ a: POKE n,a: NEXT n: POKE 23578,21:
  POKE 23633,58: POKE 23634,93
20 PRINT "(g8) (g3)";AT 10,13;"D E M O": PAUSE 65: RANDOMIZE USR 42250
25 PRINT "(g2) (g1)";TAB 7;AT BIN ,175;" Напечатав оператором (g8) PRINT
  (g1) код 130 Вы включите печать мелким шрифтом. "" Оператором (g8) TAB
  (g1) можно управлять ши-риной шрифта (от 3 до 8 пикселей). "
30 PRINT "" Код 128 включает (g8) english(g1), а код 129-русский шрифт."
35 PRINT "" Шрифт высотой 16 и шириной 8 пик-селей включается кодом 131."
40 PRINT "" Переключиться на стандартную пе-чатать можно, включив в оператор
  (g8) PRINT(g1) код 132."
45 PRINT "" Эта программа позволяет также пе-чатать 2 типа меню (коды 133 и
  134) с тенью(код 135) или без нее и выби-рать из этих меню опции(номер
  опции содержится в регистровой паре (g8)BC) (g1). "
50 GO SUB 9000
55 PRINT AT 0,100;" Передвижение курсора клавишами: "" (g8)Q(g1)-вверх,
  (g8)A(g1)-вниз, (g8)ENTER(g1)-ввод."
60 GO SUB 9000
65 PRINT "(g8)": POKE 40969,1: RANDOMIZE USR 40968: LET a=USR 41822
70 POKE 40969,2: RANDOMIZE USR 40968: LET a=USR 41822
75 POKE 40969,3: RANDOMIZE USR 40968: PRINT INK 1;CHR$ 135: LET a=USR 41822
80 PRINT #BIN ;AT NOT BIN ,BIN;"© 1994 Amateur Software System": PAUSE BIN :
  CLS
100 PRINT CHR$ 130;AT 0,175: FOR x=3 TO 7: PRINT TAB x: LIST 60 : PRINT '':
  BEEP .05,30: NEXT x : PAUSE BIN : BEEP .1,BIN : CLS
110 PRINT CHR$ 131;AT BIN ,BIN ;: LIST 60
200 GO TO 20
9000 PRINT CHR$ 131;AT 22,5;"Нажмите любую клавишу";CHR$ 130;: PAUSE BIN :
  RANDOMIZE USR 42250: RETURN

```

```

60>GO SUB 9000
65 PRINT "": POKE 40969,1: RANDOMIZE USR 40968: LET a=USR 41822
70 POKE 40969,2: RANDOMIZE USR 40968: LET a=USR 41822

60>GO SUB 9000
65 PRINT "": POKE 40969,1: RANDOMIZE USR 40968: L
ET a=USR 41822
70 POKE 40969,2: RANDOMIZE USR 40968: LET a=USR 4
1822

60>GO SUB 9000
65 PRINT "": POKE 40969,1: RANDOMIZE USR
40968: LET a=USR 41822
70 POKE 40969,2: RANDOMIZE USR 40968: LE
T a=USR 41822

60>GO SUB 9000
65 PRINT "": POKE 40969,1: RANDOMI
ZE USR 40968: LET a=USR 41822
70 POKE 40969,2: RANDOMIZE USR 409
68: LET a=USR 41822
75 POKE 40969,3: RANDOMIZE USR 409
68: PRINT INK 1;CHR$ 135: LET a=USR

60>GO SUB 9000
65 PRINT "": POKE 40969,1: RAN
DOMIZE USR 40968: LET a=USR 4182
2
70 POKE 40969,2: RANDOMIZE USR
40968: LET a=USR 41822

```

Рис. 6

Для переключения режимов управляющие символы можно непосредственно вставлять в оператор PRINT (строки 20, 25, 30, 40, 45, 55, 65), но можно и "печатать" при помощи CHR\$ (строка 9000).

Ниже приводится шестнадцатеричный дамп файлов, необходимых программе.

Файл "DEMO" CODE 40000,2291.

9C40	00	00	00	00	00	00	00	00	DC
9C48	00	10	10	10	10	00	10	00	34
9C50	00	28	28	00	00	00	00	00	3C
9C58	00	28	7C	28	28	7C	28	00	8C
9C60	00	10	7C	50	7C	14	7C	10	F4
9C68	00	44	08	10	20	44	00	00	C4
9C70	00	20	50	20	54	48	34	00	6C
9C78	00	08	10	00	00	00	00	00	2C
9C80	00	04	08	08	08	08	04	00	44
9C88	00	40	20	20	20	20	40	00	24
9C90	00	00	28	10	7C	10	28	00	18
9C98	00	00	10	10	7C	10	10	00	F0
9CA0	00	00	00	00	00	00	10	20	6C
9CA8	00	00	00	00	7C	00	00	00	C0
9CB0	00	00	00	00	00	00	10	00	5C
9CB8	00	00	04	08	10	20	40	00	D0
9CC0	00	7C	44	44	44	44	7C	00	64
9CC8	00	10	30	10	10	10	7C	00	50
9CD0	00	7C	04	04	7C	40	7C	00	28
9CD8	00	7C	04	1C	04	04	7C	00	94
9CE0	00	44	44	44	7C	04	04	00	CC
9CE8	00	7C	40	7C	04	04	7C	00	40
9CF0	00	7C	40	7C	44	44	7C	00	C8
9CF8	00	7C	04	04	08	10	10	00	40
9D00	00	7C	44	7C	44	44	7C	00	DD
9D08	00	7C	44	44	7C	04	7C	00	A5
9D10	00	00	00	10	00	00	10	00	CD
9D18	00	00	00	10	00	00	10	20	F5
9D20	00	00	08	10	20	10	08	00	0D
9D28	00	00	00	7C	00	7C	00	00	BD
9D30	00	00	20	10	08	10	20	00	35
9D38	00	7C	44	04	1C	00	10	00	C5

СИСТЕМНЫЕ ПРОГРАММЫ

9D40	00	5C	54	74	54	54	5C	00	05
9D48	00	7C	44	44	7C	44	44	00	ED
9D50	00	7C	40	7C	44	44	7C	00	29
9D58	00	44	44	44	44	44	7C	04	C9
9D60	00	3C	24	24	24	24	7C	44	89
9D68	00	7C	40	70	40	40	7C	00	2D
9D70	00	7C	54	54	54	7C	10	00	11
9D78	00	7C	40	40	40	40	40	00	D1
9D80	00	44	28	10	28	44	44	00	49
9D88	00	44	44	4C	54	64	44	00	F5
9D90	10	44	44	4C	54	64	44	00	0D
9D98	00	44	48	70	48	44	44	00	01
9DA0	00	3C	44	44	44	44	44	00	CD
9DA8	00	44	6C	54	44	44	44	00	15
9DB0	00	44	44	7C	44	44	44	00	1D
9DB8	00	7C	44	44	44	44	7C	00	5D
9DC0	00	7C	44	44	44	44	44	00	2D
9DC8	00	7C	44	44	7C	24	44	00	4D
9DD0	00	7C	44	44	7C	40	40	00	6D
9DD8	00	7C	40	40	40	40	7C	00	6D
9DE0	00	7C	10	10	10	10	10	00	49
9DE8	00	44	44	44	7C	04	7C	00	4D
9DF0	00	54	54	38	54	54	54	00	69
9DF8	00	7C	44	78	44	44	7C	00	D1
9E00	00	40	40	7C	44	44	7C	00	9E
9E08	00	44	44	74	54	54	74	00	BE
9E10	00	7C	04	18	04	04	7C	00	CA
9E18	00	44	54	54	54	54	7C	00	C6
9E20	00	7C	04	1C	04	04	7C	00	DE
9E28	00	44	54	54	54	54	7C	04	DA
9E30	00	44	44	44	7C	04	04	00	1E
9E38	00	1C	20	38	20	20	7C	00	06
9E40	00	00	5C	54	74	54	5C	00	B2
9E48	00	00	3C	04	7C	44	7C	00	62
9E50	00	00	78	40	7C	44	7C	00	E2
9E58	00	00	44	44	44	44	7C	04	86
9E60	00	00	3C	24	24	24	7C	44	66
9E68	00	00	7C	44	7C	40	7C	00	FE
9E70	00	00	7C	54	54	7C	10	00	BE
9E78	00	00	7C	40	40	40	40	00	92
9E80	00	00	44	28	10	28	44	00	06
9E88	00	00	44	4C	54	64	44	00	B2
9E90	00	10	44	4C	54	64	44	00	CA
9E98	00	00	44	48	70	48	44	00	BE
9EA0	00	00	3C	44	44	44	44	00	8A
9EA8	00	00	44	6C	54	44	44	00	D2
9EB0	00	00	44	44	7C	44	44	00	DA
9EB8	00	00	7C	44	44	44	7C	00	1A
9EC0	00	00	7C	44	44	44	44	00	EA
9EC8	00	00	7C	44	7C	14	64	00	1A
9ED0	00	00	7C	44	44	7C	40	00	2E
9ED8	00	00	7C	40	40	40	7C	00	2E
9EE0	00	00	7C	10	10	10	10	00	3A
9EE8	00	00	44	44	7C	04	7C	00	0A
9EF0	00	00	54	54	38	54	54	00	16

СИСТЕМНЫЕ ПРОГРАММЫ

9EF8	00	00	7C	44	78	44	7C	00	8E
9F00	00	00	40	40	7C	44	7C	00	5B
9F08	00	00	44	44	74	54	74	00	6B
9F10	00	00	7C	04	18	04	7C	00	C7
9F18	00	00	44	54	54	54	7C	00	73
9F20	00	00	7C	04	1C	04	7C	00	DB
9F28	00	00	44	54	54	54	7C	04	87
9F30	00	00	44	44	7C	04	04	00	DB
9F38	00	00	60	20	3C	24	3C	00	F3
9F40	F0	E1	CA	8B	C3	C3	A5	C3	F3
9F48	E1	87	E2	E2	C3	C3	C3	E2	3E
9F50	B8	E2	B8	F0	A6	B8	B8	E4	2B
9F58	B8	B8	C6	C6	E2	F0	E1	F0	96
9F60	9C	B8	B8	B8	B8	B8	B8	B8	A3
9F68	B8	E2	B8	B2	B8	9C	B4	B8	CB
9F70	B8	B4	B8	B8	C6	B8	9C	AC	B1
9F78	9C	AA	D2	E2	C6	CC	C6	F0	59
9F80	C6	B2	D8	C6	B2	B2	E2	B2	2D
9F88	B2	C6	E2	D2	E2	AA	B2	9A	2B
9F90	B2	B1	C6	B2	C6	B2	C6	AA	F2
9F98	C6	B2	93	E2	E2	C6	E2	0F	BD
9FA0	F0	E1	C6	47	8B	93	8E	C3	8C
9FA8	E1	87	C6	C6	C3	C3	C3	E2	66
9FB0	B2	C6	B2	E2	B2	B2	9A	E2	3B
9FB8	B2	B2	C6	C6	C6	F0	C6	E2	A5
9FC0	CA	B2	B2	B2	9A	96	AA	B2	CB
9FC8	B2	8B	AA	B2	B2	9A	B2	B2	B0
9FD0	B2	9A	B2	B2	C6	B2	AA	AA	EB
9FD8	9A	AA	D2	AA	E2	AA	B2	8E	03
9FE0	CA	B2	9A	B2	9A	B2	AA	B2	EF
9FE8	B2	9A	9A	B2	B2	9A	B2	9A	B7
9FF0	B2	9A	B2	B2	C6	B2	AA	AA	0B
9FF8	B2	AA	D2	AA	E2	AA	B2	9A	47
A000	20	A4	8F	A4	C1	A4	0A	A5	AB
A008	3E	01	C3	11	A3	00	00	00	5E
A010	00	00	00	00	00	00	00	00	B0
A018	00	00	00	00	00	00	00	00	B8
A020	00	00	00	00	00	00	00	00	C0
A028	F5	3A	9A	A2	47	F1	05	28	98
A030	2D	FE	80	38	29	28	4B	FE	4D
A038	A5	D2	F4	09	FE	88	30	1E	20
A040	21	50	A0	D6	81	CB	27	16	50
A048	00	5F	19	5E	23	56	EB	E9	0B
A050	8A	A0	98	A0	9D	A0	A2	A0	D1
A058	A9	A0	BE	A0	69	A1	C5	D5	43
A060	E5	2A	88	A1	CD	6F	00	2A	9E
A068	51	5C	01	28	A0	5E	71	23	70
A070	56	70	78	BA	20	04	79	BB	60
A078	28	04	ED	53	88	A1	E1	D1	5F
A080	C1	C9	21	00	3C	01	20	9F	C7
A088	18	06	21	40	9B	01	80	9F	62
A090	22	36	5C	ED	43	A4	A1	C9	22
A098	21	8E	A1	18	08	21	9D	A2	08
A0A0	18	03	21	F4	09	22	88	A1	C4
A0A8	C9	CD	3A	A1	06	06	90	32	87

СИСТЕМНЫЕ ПРОГРАММЫ

A0B0	1D	A4	3E	92	32	E9	A0	3E	DA
A0B8	04	32	8D	A1	18	11	CD	3A	EC
A0C0	A1	06	03	90	32	1D	A4	3E	CB
A0C8	9E	32	E9	A0	32	8D	A1	CD	EE
A0D0	A2	A0	3E	11	D7	3A	19	A4	CF
A0D8	D7	3E	10	D7	3A	1A	A4	D7	43
A0E0	2A	8A	A1	3A	8C	A1	57	5F	F2
A0E8	0E	92	3A	8D	A1	3D	32	8D	8C
A0F0	A1	20	09	0E	99	CD	15	A1	84
A0F8	0E	95	18	12	0D	0D	CD	15	61
A100	A1	7A	BB	28	06	FE	02	28	CD
A108	02	18	03	0C	0C	0C	15	20	1F
A110	D9	22	8A	A1	C9	CD	28	A1	36
A118	3A	1E	A4	3D	3D	47	0C	CD	4F
A120	35	A1	10	FB	0C	2C	18	0D	FF
A128	3E	16	CD	5E	A0	7D	CD	5E	90
A130	A0	7C	CD	5E	A0	79	CD	5E	5C
A138	A0	C9	11	19	A4	2A	4B	A3	28
A140	23	01	06	00	ED	B0	2B	22	F5
A148	4B	A3	2A	4D	A3	01	06	00	F8
A150	ED	42	22	4D	A3	2A	1B	A4	1B
A158	22	8A	A1	3A	1D	A4	32	8C	FF
A160	A1	F5	85	3D	32	1B	A4	F1	3B
A168	C9	2A	8A	A1	24	0E	8F	CD	B5
A170	28	A1	3A	1E	A4	3D	47	CD	27
A178	35	A1	24	10	FA	3A	8C	A1	84
A180	47	CD	28	A1	2D	10	FA	C9	FE
A188	F4	09	00	00	0A	03	FE	0D	3E
A190	CA	2E	A2	FE	16	CA	3B	A2	86
A198	FE	17	CA	64	A2	FE	20	D8	14
A1A0	06	00	4F	21	20	9F	09	5E	DD
A1A8	ED	4B	36	5C	26	00	6F	29	D1
A1B0	29	29	09	0E	08	3A	81	5C	D9
A1B8	C6	F7	57	D5	7A	D6	F7	47	D0
A1C0	7E	14	C4	16	A2	15	17	DC	77
A1C8	1E	A2	F5	3A	B0	5C	3C	32	D2
A1D0	B0	5C	F1	10	EC	23	E5	2A	9C
A1D8	B0	5C	25	3A	81	5C	95	ED	43
A1E0	44	6F	22	B0	5C	E1	D1	0D	21
A1E8	20	D1	2A	B0	5C	3A	81	5C	C7
A1F0	47	87	3D	85	38	09	78	85	5F
A1F8	6F	3E	08	84	67	18	13	24	88
A200	28	0A	25	3E	06	BC	30	04	2D
A208	2E	00	18	06	21	00	AF	CD	93
A210	77	A2	22	B0	5C	C9	CB	23	B0
A218	D0	17	14	C8	18	F8	F5	C5	47
A220	D5	E5	ED	4B	B0	5C	CD	E5	72
A228	22	E1	D1	C1	F1	C9	2A	B0	F3
A230	5C	3E	08	94	ED	44	67	2E	CE
A238	FF	18	B2	21	8E	A1	22	9B	B0
A240	A2	11	4C	A2	3E	01	32	9A	8E
A248	A2	C3	80	0A	11	52	A2	C3	A1
A250	7D	0A	ED	5B	9B	A2	CD	70	3B
A258	0A	2A	0E	5C	22	B0	5C	AF	75
A260	32	9A	A2	C9	11	6A	A2	C3	19

СИСТЕМНЫЕ ПРОГРАММЫ

A268	80	0A	11	8E	A1	CD	7D	0A	28
A270	3A	0E	5C	32	81	5C	C9	C5	53
A278	E5	01	00	00	CD	3D	1F	CD	F6
A280	54	1F	30	10	11	01	40	21	48
A288	00	40	01	FF	17	36	00	ED	A4
A290	B0	E1	C1	C9	E1	E1	E1	C3	B3
A298	7B	1B	00	9D	A2	FE	0D	28	42
A2A0	63	FE	16	28	66	FE	20	D8	3D
A2A8	ED	5B	36	5C	6F	26	00	29	E2
A2B0	29	29	19	EB	ED	4B	B0	5C	EC
A2B8	79	E6	18	F6	40	67	79	E6	CD
A2C0	07	B7	1F	1F	1F	1F	80	6F	8B
A2C8	C5	CD	F0	A2	3E	20	85	6F	E0
A2D0	30	04	3E	08	84	67	CD	F0	94
A2D8	A2	E1	24	3E	1F	BC	D2	12	1E
A2E0	A2	2C	2C	26	00	3E	16	BD	B3
A2E8	D2	12	A2	2E	00	C3	0F	A2	B2
A2F0	E5	0E	04	06	02	1A	77	CB	ED
A2F8	3F	B6	77	24	10	F7	13	0D	51
A300	20	F1	E1	C9	2A	B0	5C	26	BA
A308	20	18	CF	21	9D	A2	C3	3E	13
A310	A2	F5	3E	02	CD	01	16	F1	5F
A318	21	FE	9F	01	02	00	09	3D	C2
A320	20	FC	5E	23	56	23	4E	23	4A
A328	46	69	60	A7	ED	52	4D	44	51
A330	78	B1	0B	C8	1A	ED	43	4D	66
A338	A3	ED	53	4B	A3	CD	28	A0	41
A340	ED	4B	4D	A3	ED	5B	4B	A3	41
A348	13	18	E5	00	00	00	00	21	1C
A350	B2	88	01	06	00	09	3D	20	9A
A358	FC	11	19	A4	ED	B0	CD	C0	EF
A360	A3	CD	E2	A3	01	21	00	CD	E7
A368	3D	1F	3A	04	5C	FE	41	28	68
A370	0F	FE	51	28	10	FE	0D	20	D4
A378	EB	ED	4B	1F	A4	06	00	C9	D0
A380	01	3C	2C	18	03	01	3D	2D	12
A388	21	A4	A3	71	21	B7	A3	70	EF
A390	3A	19	A4	CB	27	CB	27	CB	D9
A398	27	47	3A	1A	A4	80	CD	F0	DE
A3A0	A3	3A	1F	A4	3C	32	1F	A4	14
A3A8	47	3A	1D	A4	B8	DC	C0	A3	84
A3B0	05	FC	D3	A3	2A	1B	A4	2C	DF
A3B8	22	1B	A4	CD	E2	A3	18	A4	4A
A3C0	3E	01	32	1F	A4	2A	1B	A4	80
A3C8	3A	1D	A4	95	ED	44	6F	22	BD
A3D0	1B	A4	C9	3A	1D	A4	32	1F	47
A3D8	A4	2A	1B	A4	85	6F	22	1B	39
A3E0	A4	C9	3A	1A	A4	CB	27	CB	A5
A3E8	27	CB	27	47	3A	19	A4	80	62
A3F0	F5	ED	5B	1B	A4	CD	02	A4	02
A3F8	3A	1E	A4	47	F1	77	23	10	79
A400	FC	C9	7B	E6	18	CB	3F	CB	B7
A408	3F	CB	3F	F6	58	67	7B	E6	0B
A410	07	B7	1F	1F	1F	1F	82	6F	DF
A418	C9	05	01	03	00	15	05	15	BD

СИСТЕМНЫЕ ПРОГРАММЫ

A420	85	05	01	01	01	0F	0A	83	ED
A428	16	02	02	46	49	4C	45	4E	54
A430	41	4D	45	82	17	05	16	10	6B
A438	7F	4D	65	67	61	20	42	41	78
A440	53	49	43	16	10	77	42	65	07
A448	74	61	20	42	41	53	49	43	43
A450	16	10	6F	4C	4F	47	4F	16	D0
A458	10	67	50	41	53	43	41	4C	27
A460	16	10	5F	50	52	4F	4C	4F	15
A468	47	16	10	57	5A	45	55	53	17
A470	16	10	4F	47	45	4E	53	20	D6
A478	34	44	16	10	47	4D	4F	4E	EB
A480	53	20	34	44	16	10	3F	62	D6
A488	6F	6F	74	84	10	02	87	86	21
A490	02	07	00	14	07	08	14	01	75
A498	16	00	16	46	49	4C	45	14	9C
A4A0	00	16	02	15	43	4F	50	59	AC
A4A8	16	03	15	4D	4F	56	45	16	C7
A4B0	04	15	44	45	4C	45	54	45	20
A4B8	16	05	15	52	45	4E	41	4D	FF
A4C0	45	86	00	07	09	14	07	09	63
A4C8	81	15	01	82	17	05	16	B8	6F
A4D0	67	46	41	4A	4C	16	A5	57	0A
A4D8	4B	4F	50	49	52	4F	57	41	E8
A4E0	54	58	16	A5	4F	50	4F	5E	37
A4E8	49	53	54	49	54	58	16	A5	2C
A4F0	47	55	44	41	4C	49	54	58	F6
A4F8	16	A5	3F	50	45	52	45	49	0B
A500	4D	45	4E	4F	57	41	54	58	18
A508	80	84	11	E8	03	AF	08	21	85
A510	00	40	01	00	18	AF	BE	28	A3
A518	0F	08	3E	01	08	1B	7A	B3	63
A520	20	03	11	E8	03	1A	A6	77	1B
A528	23	0B	78	B1	20	E7	08	3D	70
A530	28	DB	C9	00	00	00	00	00	A1

Дамп файла "UDG MENU" CODE 65368,168.

FF58	00	7F	7F	60	6F	68	68	68	5C
FF60	00	FF	FF	00	FF	00	00	00	5C
FF68	00	FE	FE	06	F6	16	16	16	A1
FF70	68	68	68	68	68	68	68	68	AF
FF78	00	00	00	00	00	00	00	00	77
FF80	16	16	16	16	16	16	16	16	2F
FF88	68	68	68	6F	60	7F	7F	00	8C
FF90	00	00	00	FF	00	FF	FF	00	8C
FF98	16	16	16	F6	06	FE	FE	00	D1
FFA0	68	68	68	68	6F	68	68	68	E6
FFA8	00	00	00	00	FF	00	00	00	A6
FFB0	16	16	16	16	F6	16	16	16	3F
FFB8	FF	FF	FF	FF	FF	FF	FF	FF	AF
FFC0	FF	FF	FF	FF	FF	FF	FF	FF	B7
FFC8	FF	FF	FF	FF	FF	FF	FF	FF	BF
FFD0	80	80	80	80	80	80	80	80	CF
FFD8	00	00	00	00	00	00	00	00	D7
FFE0	01	01	01	01	01	01	01	01	E7
FFE8	80	80	80	80	80	80	80	FF	66

FFF0	00	00	00	00	00	00	00	00	FF	EE
FFF8	01	01	01	01	01	01	01	01	FF	FD

96-байтная таблица для "вырезки" английских символов.

```
5 LET P=40768: RESTORE 10: FOR I=1 TO 96: READ A$,A: POKE P+I-1,A: NEXT I:
  STOP
```

```
10 DATA " ",BIN 11110000
20 DATA "!",BIN 11100001
30 DATA "kawy^ki",BIN 11001010
40 DATA "#",BIN 10001011
50 DATA "$",BIN 11000011
60 DATA "%",BIN 11000011
70 DATA "&",BIN 10100101
80 DATA "'",BIN 11000011
90 DATA "(",BIN 11100001
100 DATA ")",BIN 10000111
110 DATA "*",BIN 11100010
120 DATA "+",BIN 11100010
130 DATA ",",BIN 11000011
140 DATA "-",BIN 11000011
150 DATA ".",BIN 11000011
155 DATA "/",BIN 11100010
160 DATA "0",BIN 10111000
170 DATA "1",BIN 11100010
180 DATA "2",BIN 10111000
190 DATA "3",BIN 11110000
200 DATA "4",BIN 10100110
210 DATA "5",BIN 10111000
220 DATA "6",BIN 10111000
230 DATA "7",BIN 11100100
240 DATA "8",BIN 10111000
250 DATA "9",BIN 10111000
260 DATA ":",BIN 11000110
270 DATA ";",BIN 11000110
280 DATA "<",BIN 11100010
290 DATA "=",BIN 11110000
300 DATA ">",BIN 11100001
310 DATA "?",BIN 11110000
320 DATA "@",BIN 10011100
330 DATA "A",BIN 10111000
340 DATA "B",BIN 10111000
350 DATA "C",BIN 10111000
360 DATA "D",BIN 10111000
370 DATA "E",BIN 10111000
380 DATA "F",BIN 10111000
390 DATA "G",BIN 10111000
400 DATA "H",BIN 10111000
410 DATA "I",BIN 11100010
420 DATA "J",BIN 10111000
430 DATA "K",BIN 10110010
440 DATA "L",BIN 10111000
450 DATA "M",BIN 10011100
460 DATA "N",BIN 10110100
470 DATA "O",BIN 10111000
480 DATA "P",BIN 10111000
490 DATA "Q",BIN 10110100
```

```
500 DATA "R",BIN 10111000
510 DATA "S",BIN 10111000
520 DATA "T",BIN 11000110
530 DATA "U",BIN 10111000
540 DATA "V",BIN 10011100
550 DATA "W",BIN 10101100
560 DATA "X",BIN 10011100: REM BIN 10101100
570 DATA "Y",BIN 10101010
580 DATA "Z",BIN 11010010
590 DATA "[",BIN 11100010
600 DATA "\",BIN 11000110
610 DATA "]",BIN 11001100
620 DATA "^",BIN 11000110
630 DATA " ",BIN 11110000
640 DATA "`",BIN 11000110
650 DATA "a",BIN 10110010
660 DATA "b",BIN 11011000
670 DATA "c",BIN 11000110
680 DATA "d",BIN 10110010
690 DATA "e",BIN 10110010
700 DATA "f",BIN 11100010
710 DATA "g",BIN 10110010
720 DATA "h",BIN 10110010
730 DATA "i",BIN 11000110
740 DATA "j",BIN 11100010
750 DATA "k",BIN 11010010
760 DATA "l",BIN 11100010
770 DATA "m",BIN 10101010
780 DATA "n",BIN 10110010
790 DATA "o",BIN 10011010
800 DATA "p",BIN 10110010
810 DATA "q",BIN 10110001
820 DATA "r",BIN 11000110
830 DATA "s",BIN 10110010
840 DATA "t",BIN 11000110
850 DATA "u",BIN 10110010
860 DATA "v",BIN 11000110
870 DATA "w",BIN 10101010
880 DATA "x",BIN 11000110
890 DATA "y",BIN 10110010
900 DATA "z",BIN 10010011
910 DATA "{",BIN 11100010
920 DATA "|",BIN 11100010
930 DATA "}",BIN 11000110
940 DATA "~",BIN 11100010
950 DATA "©",BIN 00001111
```

96-байтная таблица для "вырезки" русских символов.

```
5 LET P=40864: RESTORE 10: FOR I=1 TO 96: READ A$,A: POKE P+I-1,A: NEXT I:
  STOP
6 POKE 23606,80: POKE 23607,194: STOP
10 DATA " ",BIN 11110000
20 DATA "!",BIN 11100001
30 DATA "kawy^ki",BIN 11000110
40 DATA "#",BIN 1000111
50 DATA "$",BIN 10001011
```

```
60 DATA "%",BIN 10010011
70 DATA "&",BIN 10001110
80 DATA "'",BIN 11000011
90 DATA "(",BIN 11100001
100 DATA ")",BIN 10000111
110 DATA "*",BIN 11000110
120 DATA "+",BIN 11000110
130 DATA ",",BIN 11000011
140 DATA "-",BIN 11000011
150 DATA ".",BIN 11000011
155 DATA "/",BIN 11100010
160 DATA "0",BIN 10110010
170 DATA "1",BIN 11000110
180 DATA "2",BIN 10110010
190 DATA "3",BIN 11100010
200 DATA "4",BIN 10110010
210 DATA "5",BIN 10110010
220 DATA "6",BIN 10011010
230 DATA "7",BIN 11100010
240 DATA "8",BIN 10110010
250 DATA "9",BIN 10110010
260 DATA ":",BIN 11000110
270 DATA ";",BIN 11000110
280 DATA "<",BIN 11000110
290 DATA "=",BIN 11110000
300 DATA ">",BIN 11000110
310 DATA "?",BIN 11100010
320 DATA "@",BIN 11001010
330 DATA "A",BIN 10110010
340 DATA "B",BIN 10110010
350 DATA "C",BIN 10110010
360 DATA "D",BIN 10011010
370 DATA "E",BIN 10010110
380 DATA "F",BIN 10101010
390 DATA "G",BIN 10110010
400 DATA "H",BIN 10110010
410 DATA "I",BIN 10001011
420 DATA "J",BIN 10101010
430 DATA "K",BIN 10110010
440 DATA "L",BIN 10110010
450 DATA "M",BIN 10011010
460 DATA "N",BIN 10110010
470 DATA "O",BIN 10110010
480 DATA "P",BIN 10110010
490 DATA "Q",BIN 10011010
500 DATA "R",BIN 10110010
510 DATA "S",BIN 10110010
520 DATA "T",BIN 11000110
530 DATA "U",BIN 10110010
540 DATA "V",BIN 10101010
550 DATA "W",BIN 10101010
560 DATA "X",BIN 10011010: REM BIN 10101100
570 DATA "Y",BIN 10101010
580 DATA "Z",BIN 11010010
590 DATA "[",BIN 10101010
```


600 DATA "\",BIN 11100010
610 DATA "j",BIN 10101010
620 DATA "^",BIN 10110010
630 DATA " ",BIN 10001110
640 DATA "`",BIN 11001010
650 DATA "a",BIN 10110010
660 DATA "b",BIN 10011010
670 DATA "c",BIN 10110010
680 DATA "d",BIN 10011010
690 DATA "e",BIN 10110010
700 DATA "f",BIN 10101010
710 DATA "g",BIN 10110010
720 DATA "h",BIN 10110010
730 DATA "i",BIN 10011010
740 DATA "j",BIN 10011010
750 DATA "k",BIN 10110010
760 DATA "l",BIN 10110010
770 DATA "m",BIN 10011010
780 DATA "n",BIN 10110010
790 DATA "o",BIN 10011010
800 DATA "p",BIN 10110010
810 DATA "q",BIN 10011010
820 DATA "r",BIN 10110010
830 DATA "s",BIN 10110010
840 DATA "t",BIN 11000110
850 DATA "u",BIN 10110010
860 DATA "v",BIN 10101010
870 DATA "w",BIN 10101010
880 DATA "x",BIN 10110010
890 DATA "y",BIN 10101010
900 DATA "z",BIN 11010010
910 DATA "{",BIN 10101010
920 DATA "|",BIN 11100010
930 DATA "}",BIN 10101010
940 DATA "~",BIN 10110010
950 DATA "©",BIN 10011010

Удобный интерфейс.

© Риккер В.В. 1993 г., п. Провидения Магаданской область

Как известно, любой программе нужно удобное управление. Особенно удобно, если выбор режимов и команд производится джойстиком. Ниже приведена процедура, которая перемещает по экрану стрелочку и при нажатии кнопки "огонь" имеет на выходе в регистровой паре HL адрес в дисплейном файле, соответствующий кончику стрелки и в регистровой паре BC соответствующий адрес в области атрибутов. Эти адреса можно лихо использовать для определения дальнейших действий. Например, имея адрес атрибутов, можно проверить, скажем, бит 6 - яркость, попали ли Вы на команду или стрелка болтается, где ни попадя. Буду очень рад, если приведенная программка кому-нибудь пригодится.

```

00010 ;-----
00020 ; Написана в августе 1993г
00030 ; Провидения
00040 ; автор Риккер В.В.
00050 ;
00060      ORG      40000      ;нужный адрес ассемблирования
00070      ENT
00080      EI
00090      JP      START
00100 ;-----
00110 ;
00120 ; *****
00130 ; * Переменные программы *
00140 ; *****
00150 ;
00160 BUFFS  DEFS    30      ;Буфер для сохранения того,
                                ;что под стрелкой на экране.
00170 STS   DEFW    0      ;Адрес в видеообласти, соответствующий
                                ;острию стрелки.
00180 CRDS  DEFW    0      ;Координаты острия стрелки.
00190 ;
00200 ;      Шаблоны стрелки
00210 ;
00220 ORSH   DEFB   #C0,#E0,#F0 ;Шаблон для операции OR
00230      DEFB   #F8,#FC,#FE
00240      DEFB   #FF,#FE,#FC
00250      DEFB   #1C,#0E,#0E
00260      DEFB   #07,#07,#03
00270 XORSH  DEFB   #C0,#A0,#90 ;Шаблон для операции XOR
00280      DEFB   #88,#84,#82
00290      DEFB   #81,#86,#B4
00300      DEFB   #14,#0A,#0A
00310      DEFB   #05,#05,#03
00320 ;
00330 STEP  EQU     2      ;Скорость перемещения стрелки
                                ;(с увеличением скорости
                                ;уменьшается точность)
00340 ;-----
00350 ; *****
00360 ; *      Подпрограммы      *
00370 ; *****
00380 ;
00390 SCRBF  LD      HL, (CRDS) ;Координаты острия стрелки.

```

СИСТЕМНЫЕ ПРОГРАММЫ

```

00400      LD      C,H      ;Подготавливаем регистры для вы-
00410      LD      B,L      ;числения адреса по координатам.
00420      LD      A,B
00430      CALL   #22B1     ;Вычисляем адрес (процедура ПЗУ).
00440      LD      (STS),HL  ;Сохраняем полученный адрес.
00450      LD      DE,BUFFS  ;В DE адрес буфера.
00460 RB0    LD      B,15    ;Высота стрелки в пикселах.
00470 RB1    PUSH   HL
00480      LD      C,2      ;Сколько смежных байт сохранять.
00490 RB2    LD      A,(HL)
00500      LD      (DE),A
00510      INC     L
00520      INC     DE
00530      DEC     C
00540      JR     NZ,RB2
00550      POP    HL
00560      CALL   DAD1     ;Получаем адрес в дисплейном
                        ;файле на 1 пиксел ниже.

00570      DJNZ  RB1
00580      RET
00590 ;-----
00600 DAD1    INC     H
00610      LD      A,H
00620      AND    7
00630      RET    NZ
00640      LD      A,H
00650      SUB    8
00660      LD      H,A
00670      LD      A,L
00680      ADD    A,32
00690      LD      L,A
00700      RET    NC
00710      LD      A,H
00720      ADD    A,8
00730      LD      H,A
00740      RET
00750 ;-----
00760 BFSCR   LD      HL,#771A ;Подпрогр. восстановления экрана.
00770      LD      (RB2),HL  ;Сначала перепрограммируется
                        ;процедура SCRBF.

00780      CALL   SCRBF
00790      LD      HL,#127E  ;Восстанавл. код процедуры SCRBF.
00800      LD      (RB2),HL
00810      RET
00820 ;-----
00830 ; Подпрограмма создания образа стрелки
00840 ; по имеющимся координатам
00850 FORM   LD      A,(CRDS+1) ;По горизонт. координате получаем
00860      AND    7          ;номер бита в байте экрана.
00870      EX    AF,AF'     ;Сохраняем его.
00880      LD      IX,ORSH   ;Накладываем маску по OR.
00890      CALL   FORMS
00900      LD      IX,MS3
00910      LD      (IX+0),#AA ;Изменяем процедуру FORMS.
00920      LD      (IX+9),#AB

```

```

00930          PUSH    IX
00940          LD      IX,XORSH    ;Накладываем маску по XOR.
00950          CALL   FORMS
00960          POP     IX
00970          LD      (IX+0),#B2 ;Восстановл. код процедуры FORMS.
00980          LD      (IX+9),#B3
00990          RET
01000 FORMS    LD      C,15        ;Число линий в стрелке.
01010          LD      HL,(STS)    ;Берем адрес в экране.
01020 MS1     LD      D,B         ;Обнуляем DE.
01030          LD      E,B
01040          EX     AF,AF'
01050          LD      B,A         ;В рег.В номер бита в адр.экрана.
01060          EX     AF,AF'
01070          DEC    B            ;Проверяем его на ноль.
01080          INC    B
01090          LD      D,(IX+0)    ;В DE байт из шаблона стрелки.
01100          JR     Z,MS21      ;Если 0, то обходим ротацию DE.
01110 MS2     SRL    D            ;Сдвигаем шаблон.
01120          RR     E
01130          DJNZ  MS2
01140 MS21    LD      A,(HL)      ;Накладываем полученную маску.
01150 MS3     OR     D            ;Команды OR D и OR E заменяются
                                ;на XOR D и XOR E и обратно.
01160          LD      (HL),A
01170          INC    L            ;Переходим к следующему байту.
01180          LD      A,L
01190          AND   #1F          ;Если вышли за правую границу
01200          JR     Z,MS4        ;экрана, то обход.
01210          LD      A,(HL)      ;Иначе накладываем маску и на
                                ;соседний байт.
01220          OR     E
01230          LD      (HL),A
01240 MS4     DEC    L            ;Переходим на нижнюю линию
01250          INC    IX          ;и на следующий байт в шаблоне.
01260          CALL  DAD1
01270          LD      A,H         ;Проверяем, вышли ли
01280          CP     #58          ;за низ экрана.
01290          RET    NC          ;ДА - выход.
01300          DEC    C            ;НЕТ - уменьшаем счетчик строк
01310          JR     NZ,MS1      ;в стрелке и повторяем фокус.
01320          RET
01330 ;-----
01340 ; *****
01350 ; * Головная программа *
01360 ; *****
01370 ;
01380 START    LD      HL,#6464    ;Начальные координаты стрелки.
01390          LD      (CRDS),HL
01400          PUSH  IX
01410 Q0      CALL   SCRBF        ;Сохраняем..
01420          CALL  FORM          ;Формируем..
01430          HALT   ;Отдыхаем..
01440          IN    A,(31)        ;Опрашиваем..
01450          PUSH  AF

```

```

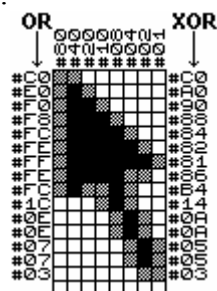
01460      CALL    BFSCR      ;Восстанавливаем..
01470      POP     AF
01480      LD      C,A
01490      LD      E,STEP
01500      LD      D,254      ;Задали максимум по X.
01510      LD      HL,CRDS+1  ;Изменяем координаты согласно
01520      BIT     0,C        ;тому, что нажали
01530      CALL    NZ,PL      ;на джойстике.
01540      BIT     1,C
01550      CALL    NZ,MIN
01560      LD      D,190      ;Максимум по Y.
01570      DEC     HL        ;Перешли на координату Y.
01580      BIT     2,C
01590      CALL    NZ,PL
01600      BIT     3,C
01610      CALL    NZ,MIN
01620      BIT     4,C
01630      JR      NZ,EXIT
01640      JR      Q0
01650 ;-----
01660 PL      LD      A,(HL)
01670      LD      B,E
01680 L0      CP      D
01690      JR      Z,END
01700      INC     A
01710      DJNZ   L0
01720 END    LD      (HL),A
01730      RET
01740 ;-----
01750 MIN     LD      A,(HL)
01760      LD      B,E
01770 INP     AND     A
01780      JR      Z,END
01790      DEC     A
01800      DJNZ   INP
01810      LD      (HL),A
01820      RET
01830 ;-----
01840 EXIT    LD      HL,(STS)
01850      PUSH   HL
01860      LD      A,H
01870      AND    #18
01880      RRCA
01890      RRCA
01900      RRCA
01910      OR     #58
01920      LD      B,A
01930      LD      C,L
01940      POP    HL
01950      POP    IX
01960      RET
01970 ;
01980 ; Конец программы

```

Комментарий "ИНФОРКОМА"

В строках 220-310 программы задается рисунок стрелки. Две маски (по OR и по XOR)

нужны для того, чтобы при попадании стрелки на область, закрашенную INK, стрелка не сливалась бы с фоном, а приобретала оконтуривание цветом PAPER. Образ стрелки и значения байтов двух масок приведены на рисунке.



Набор программы был выполнен нами при помощи ассемблера ZEUS. Для тех, кто хочет без лишних хлопот с ассемблером получить готовую программу, приводим шестнадцатеричный дамп (с контрольными суммами) кодового блока с адреса 40000. Его длина 328 байтов.

Теперь несколько слов тем, кто захочет усовершенствовать программу. Приведенный вариант программы работает только от KEMPSTON-джойстика (см. строку 1440). Это может оказаться не для всех удобно. Поэтому со своей стороны мы можем порекомендовать ввести дополнительные строки для опроса клавишей, например, SINCLAIR-джойстика, сохранив и KEMPSTON.

9C40	FB	C3	27	9D	00	00	00	00	5E
9C48	00	00	00	00	00	00	00	00	E4
9C50	00	00	00	00	00	00	00	00	EC
9C58	00	00	00	00	00	00	00	00	F4
9C60	00	00	00	00	00	00	C0	E0	9C
9C68	F0	F8	FC	FE	FF	FE	FC	1C	FB
9C70	0E	0E	07	07	03	C0	A0	90	29
9C78	88	84	82	81	86	B4	14	0A	7B
9C80	0A	05	05	03	2A	64	9C	4C	A9
9C88	45	78	CD	B1	22	22	62	9C	A1
9C90	11	44	9C	06	0F	E5	0E	02	27
9C98	7E	12	2C	13	0D	20	F9	E1	0A
9CA0	CD	A6	9C	10	F0	C9	24	7C	B4
9CA8	E6	07	C0	7C	D6	08	67	7D	2F
9CB0	C6	20	6F	D0	7C	C6	08	67	22
9CB8	C9	21	1A	77	22	98	9C	CD	F2
9CC0	84	9C	21	7E	12	22	98	9C	83
9CC8	C9	3A	65	9C	E6	07	08	DD	3A
9CD0	21	66	9C	CD	F6	9C	DD	21	EC
9CD8	0E	9D	DD	36	00	AA	DD	36	EF
9CE0	09	AB	DD	E5	DD	21	75	9C	01
9CE8	CD	F6	9C	DD	E1	DD	36	00	B4
9CF0	B2	DD	36	09	B3	C9	0E	0F	F3
9CF8	2A	62	9C	50	58	08	47	08	BB
9D00	05	04	DD	56	00	28	06	CB	D2
9D08	3A	CB	1B	10	FA	7E	B2	77	76
9D10	2C	7D	E6	1F	28	03	7E	B3	B7
9D18	77	2D	DD	23	CD	A6	9C	7C	E4
9D20	FE	58	D0	0D	20	D5	C9	21	CF
9D28	64	64	22	64	9C	DD	E5	CD	3E
9D30	84	9C	CD	C9	9C	76	DB	1F	8F
9D38	F5	CD	B9	9C	F1	4F	1E	02	4C
9D40	16	FE	21	65	9C	CB	41	C4	E3
9D48	62	9D	CB	49	C4	6C	9D	16	DB
9D50	BE	2B	CB	51	C4	62	9D	CB	80
9D58	59	C4	6C	9D	CB	61	20	16	7D

9D60	18	CD	7E	43	BA	28	03	3C	C4
9D68	10	FA	77	C9	7E	43	A7	28	DF
9D70	F9	3D	10	FA	77	C9	2A	62	19
9D78	9C	E5	7C	E6	18	0F	0F	0F	3D
9D80	F6	58	47	4D	E1	DD	E1	C9	67

Другим, более изощренным моментом является изменяемая скорость перемещения стрелки по экрану. Эти строки мы адресуем тем, кто захочет использовать предложенный интерфейс в серьезной программе, претендующей на повышенное удобство для пользователя. Как Вы видите из приведенного листинга, скорость перемещения стрелки постоянна и зависит от значения, заданного в строке 330. Слишком быстрая скорость перемещения стрелки приведёт к необходимости более точно "прицеливаться" для попадания в нужное место и к более быстрой реакции Вашей руки, но, в то же время, не потребуется много времени для того, чтобы "перегнать" стрелку от одного края экрана к другому. При медленном перемещении всё наоборот. Великолепная точность попадания и удобство "прицеливания" сочетаются с томительным ожиданием, когда же, наконец, заветная фигурка доползет до нужного поля меню, находящегося почему-то как раз в противоположном конце экрана.

Проблему можно решить следующим образом. Организуется цикл подсчёта количества прерываний, произошедших с момента нажатия любой клавиши перемещения джойстика до её отпускания. При этом начальная скорость перемещения стрелки - минимальна. Если, скажем, через 64 или 32 прерывания кнопка не отпущена, изменяется параметр, задающий скорость перемещения стрелки - её движение убыстряется. Как только кнопка отпущена - счётчик тут же обнуляется и восстанавливается начальное значение параметра, определяющего скорость. Можно ограничиться одной ступенью увеличения скорости, а можно продолжать отсчёт дальше. Если ещё через 32 цикла кнопка не отпущена - скорость увеличивается ещё больше (в одной из версий ART STUDIO, например, увеличение скорости четырехступенчатое - управление при этом очень удобно). Это, так называемое интеллектуальное управление. Конечно, все числовые параметры должны быть тщательно подобраны и сбалансированы для обеспечения максимального удобства управления.

Применение прерываний 2-го рода для создания резидентных процедур всегда представляет особенную трудность для тех, кто делает только первые шаги в машинном коде. Мастерское же владение этим инструментом открывает перед программистом огромные возможности для создания собственных программ. Мы довольно редко получаем письма от наших читателей, в которых есть примеры подобных процедур, да и те далеко не всегда выдерживают наше тестирование. Поэтому мы особенно рады предложить нашим читателям работу, выполненную сотрудником Крымской Астрофизической Обсерватории. Программа прошла тестирование без проблем.

Резидентная программа "Часы".

© Константин Парчевский, п. Научный, Крым, 1994.

Внутренние часы SPECTRUMа "тикают" примерно 50 раз в секунду. Эти "тики" подсчитываются резидентной программой и хранятся в переменной TIC. Каждые полсекунды время выводится в правый верхний угол экрана в формате "час: мин", причем, когда TIC=25 - с двоеточием, а когда TIC=50 - без двоеточия). Этим достигается мигание двоеточия с частотой 1 Гц. Однако все оказалось не так просто. Так как внутри процедуры происходит вывод на экран, то мы должны следить за тем, чтобы случайно не сменить текущий канал, ведь процедура может быть вызвана в любой момент времени. Последовательность действий при выводе информации на экран следующая.

1. Очистить место на экране, куда будет выводиться информация, ведь вывод ведётся в режиме OVER 1 (наложить) и новая информация наложится на старую, сделав её нечитаемой.

2. Запомнить, какой канал в данный момент открыт.

3. Открыть канал 2 (вывод на основной экран), вывести текущее время.

4. Восстановить прежний канал.

Ещё одна тонкость, связанная с выводом. Мы хотим видеть на экране десятичное представление времени, в то время как в компьютере вся информация представлена в HEX формате (в конечном итоге в двоичном коде), значит, в процессе вывода мы должны перекодировать числа из шестнадцатеричной системы в десятичную. Это трудоёмко и занимает много времени. (Не забывайте, что программа работает постоянно и весьма важно, чтобы время её выполнения было как можно меньше). Для уменьшения времени перекодировки из HEX- в DEC-систему часы, минуты и секунды представлены в упакованном BCD формате (если, скажем, значение часов равно 21, то в упакованном BCD формате оно будет равно #21). Таким образом, для перекодировки достаточно занести старший и младший полубайты #21 в отдельные байты, прибавить к ним #30 и напечатать получившиеся ASCII коды, как символы. Теперь проблема состоит в том, что мы не можем производить арифметические действия с такими байтами как с обычными числами. К счастью, у профессора Z-80 есть команда DAA десятичной коррекции при работе с числами в BCD формате, так что достаточно давать эту команду после каждой команды, выполняющей арифметические действия - и всё будет в порядке.

Хотелось бы остановиться на методе определения номера текущего потока. По-видимому, в ПЗУ нет стандартной процедуры, выполняющей такие действия (по крайней мере, я такой процедуры не нашёл). Где-то в оперативной памяти (не важно, по какому адресу) есть область, содержащая информацию о каналах (смахивает на начало русской народной сказки "В тридевятом царстве, в некотором государстве..."). На каждый канал нам отводится по 5 байт. Адрес этой области можно взять из системной переменной CHANS (#5C4F). В системной переменной CURCHL (#5C51) находится адрес первого байта описания текущего канала (в области информации о каналах). Чтобы получить смещение адреса текущего канала относительно начала, необходимо от содержимого CURCHL отнять содержимое CHANS. Запомним этот адрес и обозначим его, скажем, как OFFSET_1. Далее, в области системных переменных с адреса STRMS (#5C14) расположена таблица двухбайтовых чисел, указывающих на то, какой поток с каким каналом связан. Первое число соответствует потоку - 3, следующее - 2, и т.д. Эти двухбайтовые числа содержат значение на единицу большее смещения канала, который связывается с данным потоком, относительно начала области информации о каналах (Уф, ну и предложеньце! Хорошо бы в грамматику ввести скобки, указывающие приоритет). Короче, если мы начнем сравнивать OFFSET_1+1 с этими двухбайтовыми числами, то номер того числа, с которым совпадет это значение и укажет нам номер текущего потока. Напомним, что потоки нумеруются с -3.

После запуска программы выяснилось, что часы отстают примерно на 2,5 "тика" в секунду. Это, видимо, связано с тем, что дважды в секунду происходит вывод на экран, причем он длится чуть больше 1/50 секунды, так что два "тика" пропускаются полностью и ещё чуть-чуть. В исходный текст уже внесена коррекция (вместо 25 стоит 24, а вместо 50 стоит 48), но отставание на "полтика" в секунду осталось. Можно сократить число "тиков" на единицу, поставив 47, но тогда часы будут спешить на "полтика" в секунду. Количественно отставание на "полтика" в

секунду проявляется в том, что часы за час отстают на 30 секунд, что не так уж и плохо, особенно если учесть, что существенно большей точности добиться невозможно, так как часы останавливаются, когда запрещены прерывания (скажем при работе с магнитофоном (ввод/вывод).

Листинг резидентной программы "Часы".

```

00010      ORG      #FDFD
00020      DISP    0 - #9000
00030      DEFW    CLOCK      ;Вектор прерывания.
00040 TIME  DEFB    21,1,22,0,26 ;PRINT OVER 1; AT 0,26;
00050      DEFM    " 00 00"   ;Маска для печати времени.
00060 HOR   DEFB    0         ;Часы в BCD формате.
00070 MIN   DEFB    0         ;Минуты в BCD формате.
00080 SEC   DEFB    0         ;Секунды в BCD формате.
00090 TIC   DEFB    0         ;Счетчик "тиков".
00100 MSG   DEFB    #80       ;Таблица сообщений для вывода на экран.
00110      DEFM    "Input hour:"
00120      DEFB    #A0,13
00130      DEFM    "Input min.:"
00140      DEFB    #A0
00150 ; Резидентный обработчик маскируемого прерывания.
00160 CLOCK PUSH    AF
00170      PUSH    BC
00180      PUSH    DE
00190      PUSH    HL
00200      LD      HL,TIC      ;Увеличить значение
00210      INC     (HL)        ;счетчика "тиков".
00220      LD      A,(HL)
00230      CP      24         ;Если (TIC)=24
00240      JR      NZ,LOC1    ;то переход.
00250      LD      HL,TIME+8   ;Прошло полсекунды, (TIC)=24.
00260      LD      (HL),":    ;Установить ":" в буфер печати.
00270      PUSH    HL
00280      CALL   PTIME      ;Напечатать время.
00290      POP     HL
00300      LD      (HL),#20
00310      JR      END
00320 LOC1  CP      48
00330      JR      NZ,END     ;Переход, если (TIC)=48.
00340      LD      HL,TIC     ;Прошла секунда, (TIC)=48.
00350      LD      (HL),0     ;Обнулить счётчик "тиков".
00360      DEC     HL         ;HL=SEC.
00370      LD      B,1
00380 LOOP1 LD      A,(HL)   ;Число секунд (минут)
00390      INC     A         ;увеличить на 1.
00400      DAA
00410      CP      #60
00420      JR      NZ,LOC2    ;Переход, если нет переполнения.
00430      LD      (HL),0     ;Иначе обнулить секунды.
00440      DEC     HL         ;HL=MIN (HL=HOR).
00450      DEC     B
00460      JR      NZ,LOOP1   ;Повторить для минут.
00470      LD      A,(HL)   ;HL=HOR.
00480      INC     A
00490      DAA
00500      CP      #24
00510      JR      NZ,LOC2    ;Переход, если нет переполнения.

```

```

00520      LD      (HL),0
00530      JR      LOC7
00540 LOC2   LD      (HL),A      ;Сохранить часы, мин. или сек.
00550 LOC7   CALL    PTIME      ;Печать с пробелом вместо ":".
00560 END    POP     HL
00570      POP     DE
00580      POP     BC
00590      POP     AF
00600      JP      #38          ;На стандартную обработку прерывания
00610 ; Процедура печати времени на экране.
00620 PTIME  LD      DE,TIME+6
00630      LD      HL,HOR
00640      LD      A,(HL)
00650      CALL    SBYT        ;Преобразовать HOR из BCD в ASCII.
00660      INC     HL
00670      INC     DE
00680      INC     DE
00690      LD      A,(HL)
00700      CALL    SBYT        ;Преобразовать MIN из BCD в ASCII.
00710      CALL    CLST        ;Очистка места для печати.
00720      CALL    CHNL        ;Возвращает в В текущий поток.
00730      LD      A,B
00740      CP      2
00750      JR      NZ,LOC5
00760      LD      DE,TIME      ;Текущим является вывод на экран
00770      LD      BC,11
00780      CALL    #203C        ;Напечатать время.
00790      RET
00800 LOC5   PUSH    AF        ;Текущий поток=2, запомнить его.
00810      LD      A,2
00820      CALL    #1601        ;Открыть поток "S".
00830      LD      DE,TIME
00840      LD      BC,11
00850      CALL    #203C        ;Напечатать время.
00860      POP     AF
00870      CALL    #1601        ;Восстановить текущий поток.
00880      RET
00890 ; Процедура определения текущего потока.
00900 ; Возвращает в В номер текущего потока (-1,0,1,2,...)
00910 CHNL   LD      B,#FF
00920      LD      HL,(#5C51)    ;HL=адрес байта текущего канала.
00930      LD      DE,(#5C4F)    ;DE=адрес области с информацией о
                                ;каналах.
00940      XOR     A            ;CY=0.
00950      SBC    HL,DE
00960      INC     HL          ;HL содержит ту же информацию,
                                ;что содержится в таблице STRMS.
00970      LD      DE,#5C14      ;DE=STRMS.
00980 LOOP2 LD      A,(DE)      ;Поиск в табл.STRMS значения HL.
00990      CP      L
01000      JR      Z,LOC3        ;Переход, если L совпало.
01010      INC     DE
01020 LOC4   INC     DE
01030      INC     B            ;Перейти к следующему слову в таблице.
01040      JR      LOOP2        ;При этом В содержит номер потока.

```

```

01050 LOC3      INC      DE
01060          LD       A, (DE)
01070          CP       H
01080          RET      Z           ;Закончить, если и H совпало.
01090          JR       LOC4
01100 ; Процедура очистки области на экране.
01110 ;Т.к. вывод времени на экран ведется в режиме OVER 1,
01120 ;то прежде, чем что-нибудь выводить, необходимо очистить
01130 ;место. Ниже приводятся адреса экрана, куда будет
01140 ;осуществляться вывод и которые нужно очистить.
01150 ; #401A #401B... #401F, #411A #411B... #411F,...
01160 ; ... #471A #471B...#471F.
01170 ;
01180 CLST      LD       HL, #401A
01190 LOOP4    PUSH     HL
01200 LOOP3    LD       (HL), 0
01210          INC      HL
01220          LD       A, L
01230          CP       #20
01240          JR       NZ, LOOP3
01250          POP      HL
01260          INC      H
01270          LD       A, H
01280          CP       #48
01290          JR       NZ, LOOP4
01300          RET
01310 ; Процедура перевода упакованного BCD байта в 2 байта
01320 ;кода ASCII и запись их в буфер печати.
01330 ;Входные параметры: A - упакованный BCD байт,
01340 ;DE - указатель на буфер ASCII символов.
01350 SBYT      PUSH     AF
01360          SRL      A
01370          SRL      A
01380          SRL      A
01390          SRL      A
01400          ADD     A, #30
01410          LD       (DE), A
01420          INC     DE
01430          POP     AF
01440          AND     #F
01450          ADD     A, #30
01460          LD       (DE), A
01470          RET
01480 ; Процедура инициализации часов.
01490 INIT      IM      1           ;Часы - останавливаются.
01500          LD       A, #FD
01510          LD       I, A         ;I=старший байт адреса вектора
                                ;прерывания.
01520          CALL    #D6B        ;Очистить экран.
01530          LD       A, 2        ;Открыть
01540          CALL    #1601       ;поток 2.
01550          LD       DE, MSG     ;DE=адрес таблицы сообщений.
01560          XOR      A           ;A=0.
01570          LD       (SEC), A    ;Удобный случай обнулить секунды
01580          CALL    #C0A        ;Печать сообщения "Input hour:".

```

```

01590      CALL    INBYT          ;Ввод значения для установки часов.
01600      LD      (HOR),A        ;Установить часы.
01610      LD      DE,MSG
01620      LD      A,1
01630      CALL    #COA          ;Печать сообщения "Input min.:".
01640      CALL    INBYT          ;Ввод значения для установки минут.
01650      LD      (MIN),A        ;Установить минуты.
01660      IM      2              ;Запустить часы.
01670      RET
01680 ; Процедура ввода байта в BCD формате.
01690 ;Возвращает: A - байт в BCD формате.
01700 INBYT  CALL    INDIG
01710      LD      B,A
01720      SLA     B
01730      SLA     B
01740      SLA     B
01750      SLA     B
01760      CALL    INDIG
01770      OR      B
01780      RET
01790 ; Процедура ввода BCD цифры.
01800 ;Возвращает: A - цифра в BCD формате.
01810 INDIG  LD      HL,#5C3B
01820      RES     5,(HL)
01830 IN1    BIT     5,(HL)
01840      JR      Z,IN1          ;Ждать нажатия клавиши.
01850      LD      A,(#5C08)      ;Читать код клавиши из LAST K.
01860      PUSH    AF
01870      RST     #10           ;Эхо на экран.
01880      POP     AF
01890      SUB     #30           ;Преобразовать код клавиши в BCD цифру.
01900      RET
01910 ; Процедура деинициализации часов.
01920 SHUT   IM      1
01930      RET

```

Перед тем, как начать набор текста в ZEUS ASSEMBLER, необходимо выйти в бейсик командой "Q", после чего дать команду CLEAR 27999 и вернуться в ZEUS (RANDOMIZE USR 57344).

После набора текста и его ассемблирования сохраните готовый машинный код командой: SAVE "clock c" CODE 28159,331.

Бейсик-загрузчик.

```

10 REM SET Clock = 65269
20 REM OFF Clock = 65351
30 CLEAR 65022
40 LOAD "clock c" CODE 65023
50 RANDOMIZE USR 65269

```

ИФК: Кодовый блок программы "часы" можно набрать, пользуясь шестнадцатеричным дампом с контрольными суммами, который мы даём ниже.

FDF8	00	00	00	00	00	00	00	2A	1F
FE00	FE	15	01	16	00	1A	20	30	92
FE08	30	20	30	30	00	00	00	00	B6
FE10	80	49	6E	70	75	74	20	68	26
FE18	6F	75	72	3A	A0	0D	49	6E	0A
FE20	70	75	74	20	6D	69	6E	2E	09
FE28	3A	A0	F5	C5	D5	E5	21	0F	A4

СИСТЕМНЫЕ ПРОГРАММЫ

FE30	FE	34	7E	FE	18	20	0E	21	43
FE38	09	FE	36	3A	E5	CD	74	FE	D1
FE40	E1	36	20	18	28	FE	30	20	03
FE48	24	21	0F	FE	36	00	2B	06	FF
FE50	01	7E	3C	27	FE	60	20	11	BF
FE58	36	00	2B	05	20	F3	7E	3C	89
FE60	27	FE	24	20	04	36	00	18	19
FE68	01	77	CD	74	FE	E1	D1	C1	90
FE70	F1	C3	38	00	11	07	FE	21	91
FE78	0C	FE	7E	CD	E1	FE	23	13	E0
FE80	13	7E	CD	E1	FE	CD	CD	FE	53
FE88	CD	AE	FE	78	FE	02	20	0A	A1
FE90	11	01	FE	01	0B	00	CD	3C	B3
FE98	20	C9	F5	3E	02	CD	01	16	98
FEA0	11	01	FE	01	0B	00	CD	3C	C3
FEA8	20	F1	CD	01	16	C9	06	FF	69
FEB0	2A	51	5C	ED	5B	4F	5C	AF	27
FEB8	ED	52	23	11	14	5C	1A	BD	70
FEC0	28	05	13	13	04	18	F7	13	37
FEC8	1A	BC	C8	18	F6	21	1A	40	ED
FED0	E5	36	00	23	7D	FE	20	20	C7
FED8	F8	E1	24	7C	FE	48	20	F0	A5
FEE0	C9	F5	CB	3F	CB	3F	CB	3F	BA
FEE8	CB	3F	C6	30	12	13	F1	E6	E2
FEF0	0F	C6	30	12	C9	ED	56	3E	4F
FEF8	FD	ED	47	CD	6B	0D	3E	02	AC
FF00	CD	01	16	11	10	FE	AF	32	E3
FF08	0E	FE	CD	0A	0C	CD	24	FF	E6
FF10	32	0C	FE	11	10	FE	3E	01	A9
FF18	CD	0A	0C	CD	24	FF	32	0D	29
FF20	FE	ED	5E	C9	CD	35	FF	47	79
FF28	CB	20	CB	20	CB	20	CB	20	D3
FF30	CD	35	FF	B0	C9	21	3B	5C	61
FF38	CB	AE	CB	6E	28	FC	3A	08	4F
FF40	5C	F5	D7	F1	D6	30	C9	ED	14
FF48	56	C9	00	00	00	00	00	00	66

ОБМЕН ОПЫТОМ

В последнее время все больше стало появляться известных программ, переведённых на русский язык. Наверное, это объясняется не только желанием играть в русифицированную игру. Скорее всего, это, с одной стороны, непреодолимая тяга к самостоятельному творчеству, а с другой - недостаточные навыки в программировании в машинных кодах. Создать свою собственную программу для начинающего довольно сложно. Кроме чистого машинного кода существует ещё масса проблем компоновки, символьного набора и других. Если все это попытаться преодолеть сразу, то можно зарыться так, что и не раскопаешься потом. Поэтому первое, с чего обычно начинают изучение машинного кода это перевод фирменных программ на русский язык. А когда психологический барьер преодолен, и поток новой информации улёгся в голову, можно двигаться и дальше. Однако сказанное не означает, что переводом программ занимаются только начинающие. Многие продолжают и дальше заниматься этой работой профессионально, оттачивая свое мастерство. Для тех, кто делает только первые шаги в переводе программ на русский язык, полезной может оказаться следующая информация.

Опыт русификации программ.

© Дмитрий Пьянков, Сергей Машарин, Горно-Алтайск, 1994.

Хотели бы поделиться небольшим опытом, накопленным в результате перевода на русский язык программ "Dizzy 1-2", Gremlins. Во всех (практически) фирменных программах используется свой шрифт. Поэтому символьный набор надо найти. Следующие две программы это делают.

Нахождение символьного набора.

```
10 FOR i=93 TO 255
20 POKE 23607,i
30 FOR k=32 TO 127: PRINT CHR$ k;: NEXT k
40 IF INKEY$=CHR$ 13 THEN POKE 23607,60: PRINT (i+1)*256: GO TO 70
50 IF INKEY$="" THEN GO TO 40
60 PRINT AT 0,0;: NEXT i
```

Эта программа находит примерное расположение шрифта. Если он пока не встретился, нажимаете любую клавишу и поиск продолжается. Если символьный набор уже встретился, то нажмите ENTER - будет выведен текущий адрес - примерный адрес расположения символьного набора. Его можно использовать в следующей программе.

```
100 POKE 22528,24: FOR i=24500 TO 65535
105 PRINT AT 1,1;i
110 FOR k=0 TO 7: POKE (16384+256*k),PEEK (i+k)
120 NEXT k: PAUSE 0
130 NEXT i
```

Здесь вместо i=24500 в строке 100 надо поставить примерный адрес, найденный предыдущей программой.

ИФК: можем порекомендовать вставить дополнительно строку:

```
125 IF INKEY$=" " THEN LET i=i-1: GO TO 105
    так Вы обеспечите себе возможность "реверса".
```

Нахождение и распечатка текстовых сообщений.

```
10 FOR i=24500 TO 65535
20 IF PEEK i<128 THEN IF PEEK i>31 THEN PRINT PAPER 6;CHR$ PEEK i;: NEXT i
30 PRINT : NEXT i
```

Эта небольшая программка распечатывает только текстовые сообщения. Если же в строке 30 убрать PRINT, то её можно использовать для нахождения сообщений.

И последняя программа - перевода сообщений:

```
1 LET b$=""
10 FOR i=24500 TO 65535
20 IF PEEK i<128 THEN IF PEEK i>31 THEN LET b$=b$+CHR$ PEEK i: NEXT i
25 IF LEN b$<2 THEN GO TO 60
```

```

30 PRINT AT 21,1; PAPER 6;b$
35 LET d=LEN b$
40 POKE 23606,40: POKE 23607,95: REM Russian
45 INPUT b$: IF b$="" THEN GO TO 60
46 IF LEN b$<>d THEN GO TO 45
50 FOR k=0 TO d-1: POKE (i+k-d),CODE b$(k+1): NEXT k
60 LET b$="": POKE 23606,0: POKE 23607,60: CLS : NEXT i

```

Эта программа выводит текстовое сообщение, которое надо перевести на русский язык, затем запрашивает перевод (длина сообщений должна быть одинакова) и размещает новое сообщение вместо старого. В строках 1...30 программа ищет сообщение и пишет его на экране, строки 35...60 выполняют ввод нового сообщения и замену старого.

Эта программа является базовой, то есть для каждого конкретного случая к ней можно добавлять или убирать строки. Например, убрав РОКЕ в строке 60 программа может применяться для редактирования уже переведенных сообщений. Не очень сложно сделать, чтобы она работала в режиме, когда последний символ текстового сообщения инвертирован, то есть с включенным 7 битом. В общем, программу можно развивать и переделывать для конкретного случая.

ИФК: Рекомендуем дополнить программу строками для выгрузки на ленту или диск уже готового материала (см. статью Алексева А.Г. "МАСТЕРФАЙЛ 09 полная русификация" в ZX-РЕВЮ № 1-2 за 1992 г.) - мы пробовали, это очень удобно.

В заключение ещё пара советов. Найдя символьный набор, выгрузите его на ленту или диск командой:

```
SAVE "имя" CODE адрес,768
```

это позволит без изменений загружать символьный набор для редактирования в ART STUDIO.

Для тех, кто работает с магнитофоном, пригодится следующий приём. Перед переводом программу надо "разметить", то есть выделить в ней участки, где есть текстовые сообщения. Сохранять можно только блок кодов с текстовыми сообщениями, это значительно сэкономит время при загрузке-выгрузке. Например, кодовый блок программы CODE 24500,40000 содержит текстовые сообщения, расположенные с адреса 32000 и сосредоточены они в массиве длиной 5000 байт. Сохраните этот фрагмент программы и в дальнейшем работайте с ним. А после завершения работы объедините этот блок со всей программой.

Вывод спрайтов на экран с маской.

© Сергей Сурков, г. Саянск, Иркутская область, 1994.

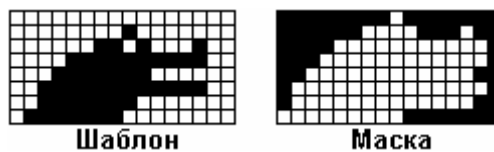
В книге "Прикладная графика" ИНФОРКОМ затронул одну из самых важных проблем, возникающих при написании игровых программ - это вывод или печать шаблона изображения на экран. В этом издании предложены два способа печати. Это печать командой LD, непосредственно выводя шаблон поверх фонового изображения и печать с помощью команды XOR, смешивая шаблон и фоновую картинку. Оба способа имеют каждый свои недостатки, заметные при качественной графике фонового изображения. В первом случае шаблон "выкусывает" прямоугольник фона в зависимости от своего формата, а во втором - при наложении на фон шаблона происходит частичное искажение последнего.

Тем не менее, существует способ вывода шаблона на экран, который не искажает фоновую картинку и выводимый шаблон. При использовании этого способа, выводимый на экран шаблон, попадая на изображение фона, отделяется от последнего "оболочкой" шириной в один пиксел. Чтобы нагляднее это представить, достаточно взглянуть на главного героя известной игры DIZZY.

Прежде, чем переходить к описанию метода, необходимо сказать несколько слов о формате хранения шаблона и так называемой маски, имеющей следующий вид:

1. Размеры маски те же, что и шаблон.
2. Изображение маски должно из себя представлять инвертированный силуэт шаблона с отступом во все стороны в одну точку.

К примеру:



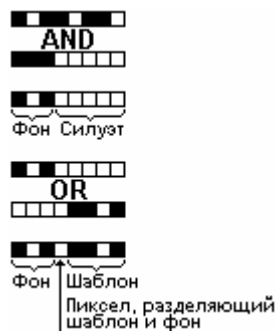
Шаблон и маску необходимо подготовить в графическом редакторе или генераторе спрайтов и упаковать их в такой формат:



Обратите внимание на то, что шаблон и маска хранятся в несколько отличном от предложенного "Прикладной графикой" формате - шаблон и маска хранятся в виде последовательности линий изображения, а не знакомест.

Теперь о сути метода. Прежде всего, используя команду AND процессора необходимо наложить маску на то окно экрана, где будет отпечатан шаблон. В результате получится окно с пустым силуэтом шаблона и неискаженной фоновой картинкой по краям силуэта (см. рисунок).

Затем осталось только впечатать шаблон в силуэт. Это позволяет сделать команда OR (см. рисунок).



Принцип действия программы в деталях отличается от описанного. Сделано это для ускорения процедуры. Программа выбирает байт окна экрана, накладывает на него

соответствующий байт маски, затем - байт шаблона, заносит полученное в буфер и переходит к следующему байту. После обработки всего окна программа выводит содержимое буфера на экран.

```

ORG 25000
COORD      DEFW    0           ; Координаты окна.
SPRITE     DEFW    0           ; Адрес начала маски и шаблона.
SRE         DEFW    0           ; Размер маски (шаблона), который
                                           ; программа выбирает из
                                           ; информации о печатаемом шаблоне.
BUFFER      DEFW    0           ; Адрес буфера, куда копируется окно.
    
```

Первая часть программы - расчёт адреса по координатам. Подробное описание расчёта можно найти в первой книге сериала "Персональный компьютер ZX-SPECTRUM. Элементарная графика".

```

LD      DE, (COORD)
LD      A, E
AND     #18
OR      #40
LD      H, A
LD      A, E
AND     #7
OR      A
RRA
RRA
RRA
RRA
ADD     A, D
LD      L, A
PUSH    IX           ; На стеке запоминаются индексные
PUSH    IY           ; регистры, это необходимо для
                       ; успешного возврата в BASIC.
PUSH    HL           ; 2 раза на стеке запоминается
PUSH    HL           ; адрес в экране, рассчитанный в
                       ; предыдущей части программы.

LD      HL, (SPRITE)
INC     HL           ; HL устанавливается на байт
INC     HL           ; высоты.
LD      A, (HL)     ; Высоту - в SIZE.
LD      (SIZE), A
INC     HL           ; HL устанавливается на байт длины.
LD      A, (HL)     ; Длину - в (SIZE+1).
LD      (SIZE+1), A
LD      HL, (SPRITE)
LD      E, (HL)
INC     HL           ; HL настраивается на начало
LD      D, (HL)     ; шаблона.
DEC     HL
ADD     HL, DE
PUSH    HL           ; Это аналог команды LD IX, HL,
POP     IX           ; которой нет в системе команд

LD      HL, (SPRITE)
LD      DE, L       ; Здесь DE настраивается на
ADD     HL, DE      ; начало маски.
EX      DE, HL
LD      IY, (BUFFER)
POP     HL
    
```

процессора

К этому моменту регистры настраиваются таким образом: DE - адрес начала маски, HL - адрес в экране, IX - адрес начала шаблона, IY - адрес начала буфера.

Следующая часть программы обрабатывает окно и помещает его в буфер.

	LD	A, (SIZE)	; В - счётчик
	LD	B, A	; рядов.
STR1	PUSH	BC	
	PUSH	HL	
	LD	B, 8	; В - счётчик линий в ряду.
	PUSH	HL	
	LD	A, (SIZE+1)	; В регистре В устанавливается
	LD	B, A	; счётчик столбцов.
COL1	LD	A, (HL)	; Выбран байт из окна.
	EX	DE, HL	; Накладывается байт маски на
	AND	(HL)	; байт окна.
	EX	DE, HL	
	OR	(IX+O)	; Накладывается байт шаблона.
	LD	(IY+O), A	; Полученный байт засылается в буфер.
	INC	HL	; Переход к следующим байтам
	INC	DE	; окна экрана, маски, шаблона
	INC	IX	; и буфера.
	INC	IY	
	DJNZ	COL1	; Конец цикла по столбцам.
	POP	HL	
	POP	BC	
	INC	H	; Переход на следующую линию.
	DJNZ	LINE1	; Конец цикла по линиям.
	POP	HL	
	POP	BC	
	LD	A, #20	; Переход на следующий
	ADD	A, L	; ряд.
	LD	L, A	
	JR	NC, PASS1	; Корректировка номера
	LD	A, 8	; сегмента.
	ADD	A, H	
	LD	H, A	
PASS1	DJNZ	STR1	; Конец цикла по рядам.

Итак, в буфере сейчас окно с впечатанным в него шаблоном, осталось только вывести его на экран.

	LD	DE, (BUFFER)	
	POP	HL	; в HL - адрес в экране
	LD	A, (SIZE)	; В - счётчик рядов
	LD	B, A	
STR2	PUSH	BC	
	PUSH	HL	
	LD	B, 8	
LINE2	PUSH	BC	; В - счётчик линий
	PUSH	HL	
	LD	A, (SIZE+1)	; В - счётчик
	LD	B, A	; столбцов
COL2	LD	A, (DE)	; пересылается байт буфера в байт
	LD	(HL), A	; экрана
	INC	DE	; переход к следующим байтам
	INC	HL	
	DJNZ	COL-2	; конец цикла по столбцам
	POP	HL	

	POP	BC	
	INC	H	
	DJNZ	LINE2	; переход на следующую строку
	POP	HL	; конец цикла по линиям
	POP	BC	
	LD	A, 32	; переход на следующий ряд
	ADD	A, L	
	LD	L, A	
	JR	NC, PASS2	; корректировка
	LD	A, 8	; номера
	ADD	A, M	; сегмента
	LD	H, A	
PASS2	DJNZ	STR2	; конец цикла по рядам
	POP	IY	; индексные регистры
	POP	IX	; восстанавливаются со стека
	RET		; возврат

В заключение, хотелось бы отметить, что такой способ вывода шаблона на экран все же медленнее других, и, поэтому, он вряд ли подойдет для "стрелялок" с большим количеством стремительно движущихся объектов, но это им и не нужно - графика таких игр редко бывает настолько качественна, что нельзя обойтись другими способами печати.

Генерация шаблонов 2x2 с использованием статического RND.

© Владимир Витвинов, р.п. Пышма, Свердловская область

Очень много памяти компьютера при использовании графики занимают шаблоны. К примеру, шаблон 2x2 знакоместа занимает 32 байта. Зачастую приходится для экономии памяти ограничивать количество различных шаблонов, что приводит к однообразию игровых локаций. Программисты используют различные "хитрости" для экономии памяти.

Здесь реализован один из подходов для генерации шаблонов 2x2. Принцип работы программы таков:

1. Инициализируются три переменные для RND – R1, R2, R3. Значения можете выбрать сами. После вызова предлагаемой процедуры шаблоны пойдут в строгой последовательности. Переменная, указывающая на начало UDG-графики, устанавливается на начало буфера BUFF - это для демонстрации.

2. Последовательно вызывается процедура RND, забивая 8 байтов случайными числами.

Если сейчас Вы напечатаете полученный символ, то получите ерунду в виде бессистемно разбросанных точек. "Хитрость" заключается в том, что это знакоместо слева направо зеркально копируется во второе, а затем эти два зеркально сверху вниз в следующие два. В результате получается шаблон 2x2 знакоместа вполне приемлемый для последующего использования в программах (например, для построения из них каких-нибудь объектов, лабиринтов и пр.).

В приведенной ниже программе реализована генерация 4-х видов шаблонов. Процедура построена с 5-ю точками входа: 1 - инициализация, 2 - разворачивает шаблон, как описано выше, 3 - оконтуривает его вдобавок, 4 - включает биты в виде "+", 5 - в виде "x". Последние операции придают законченность шаблону. Один раз, вызвав процедуру, можно сколько угодно печатать полученный шаблон из буфера. Вызвав её ещё раз, получите следующий по порядку и так далее.

Листинг генератора шаблонов, набранный в ZEUS ASSEMBLER.

```

00010          ORG      40000
00020          ENT
00030 ; 1 - инициализация.
00040          LD      HL,RNEW          ;Три начальных значение
00050          LD      DE,R1            ;копируются в переменные
00060          LD      BC,3              ;R1, R2, R3.
00070          LDIR
00080          LD      HL,BUFF          ;Переключаем переменную UDG на
00090          LD      (23675),HL       ;начало буфера (граф. А,В,С,Д
                                         ;будут использ.для демонстрации.

00100          RET
00110 ; 2 - разворачивание шаблона.
00120          LD      HL,BUFF          ;8 байтов буфера заполняются
00130          LD      B,8              ;случайными числами
00140 L0       CALL    RND              ;по статическому RND,
00150          LD      (HL),A           ;что будет соответствовать
00160          INC     HL                ;"А" в графическом режиме.
00170          DJNZ   L0
00180          JR      PROD              ;Переход на развертку символов.
00190 ; 3 - оконтуривание шаблона.
00200          LD      HL,BUFF
00210          LD      B,8              ;То же, но с включением
00220          LD      A,#FF            ;верхних и левых пикселов.
00230          JR      L2
00240 L1       CALL    RND
00250          SET    7,A
00260 L2       LD      (HL),A
00270          INC     HL
00280          DJNZ   L1

```

```

00290          JR      PROD
00300 ; 4 - включение битов в виде "+".
00310          LD      HL,BUFF
00320          LD      B,7          ;То же, но с включением
00330 L3       CALL    RND          ;нижних и правых пикселов.
00340          SET     0,A
00350          LD      (HL),A
00360          INC     HL
00370          DJNZ   L3
00380          LD      A,#FF
00390          LD      (HL),A
00400          JR      PROD
00410 ; 5 - включение битов в виде "x".
00420          LD      HL,BUFF
00430          LD      B,8          ;То же, но с включением
00440          LD      C,#80        ;пикселов по диагонали:
00450 L4       CALL    RND          ;с верхнего левого
00460          OR      C          ;к нижнему правому.
00470          RR      C          ;#80=10000000 - побитовая рас-
00480          LD      (HL),A      ;кладка, эту единицу мы будем
00490          INC     HL          ;вращать и накладывать по OR.
00500          DJNZ   L4
00510 ; Развертка полученного знакоместа.
00520 PROD     LD      HL,BUFF      ;Откуда.
00530          LD      DE,BUFF+8    ;Куда.
00540          LD      BC,#0808    ;B=счётчик битов, C=счётчик байтов.
00550 L5       PUSH   BC
00560          LD      A,(HL)       ;Принимаем байт.
00570 L6       RRA              ;Переносим 0 бит в флаг CARRY.
00580          RL      C          ;Подхватываем его в регистр C.
00590          DJNZ   L6          ;Разворачиваем 8 битов.
00600          LD      A,C          ;Зеркальный байт перебрасываем
00610          LD      (DE),A      ;в новое место.
00620          INC     HL          ;Увеличиваем
00630          INC     DE          ;указатели.
00640          POP    BC
00650          DEC     C          ;Проверяем, все ли 8 байтов
00660          JR      NZ,L5       ;переброшены. Символ "B" готов.
00670          LD      HL,BUFF      ;Откуда.
00680          LD      DE,BUFF+23   ;Куда, здесь перебрасываем
00690          LD      C,2          ;с конца, развернуть надо
00700 JUMP     LD      B,8          ;два символа.
00710 L7       LD      A,(HL)     ;Переброска
00720          LD      (DE),A      ;задом-наперед.
00730          INC     HL          ;Изменяем
00740          DEC     DE          ;указатели.
00750          DJNZ   L7          ;Все ли байты переброшены,
00760          DEC     C          ;оба ли символа "C" и "D"
00770          RET     Z          ;развернуты, если ДА, то выход.
00780          LD      HL,BUFF+8    ;Указатели для
00790          LD      DE,BUFF+31   ;символа "D".
00800          JR      JUMP
00810 RND      LD      A,(R1)      ;Процедура взята из
00820          LD      D,A          ;№ 7-8, 1993 г., стр.154.
00830          LD      A,(R2)      ;Можете составить свою.

```

```

00840      LD      (R1),A
00850      ADD     A,D
00860      LD      D,A
00870      LD      A,(R3)
00880      LD      (R2),A
00890      ADD     A,D
00900      RLCA   ;
00910      LD      (R3),A
00920      RET
00930 R1     DEFB   0
00940 R2     DEFB   0
00950 R3     DEFB   0
00960 RNEW   DEFB   5           ;Начальные значения для RND.
00970       DEFB   90         ;Можете
00980       DEFB   37         ;подставить свои.
00990 BUFF   DEFS   32         ;Резервируем 32 байт для буфера.

```

Приведенная ниже BASIC-программа демонстрирует возможности процедур.

```

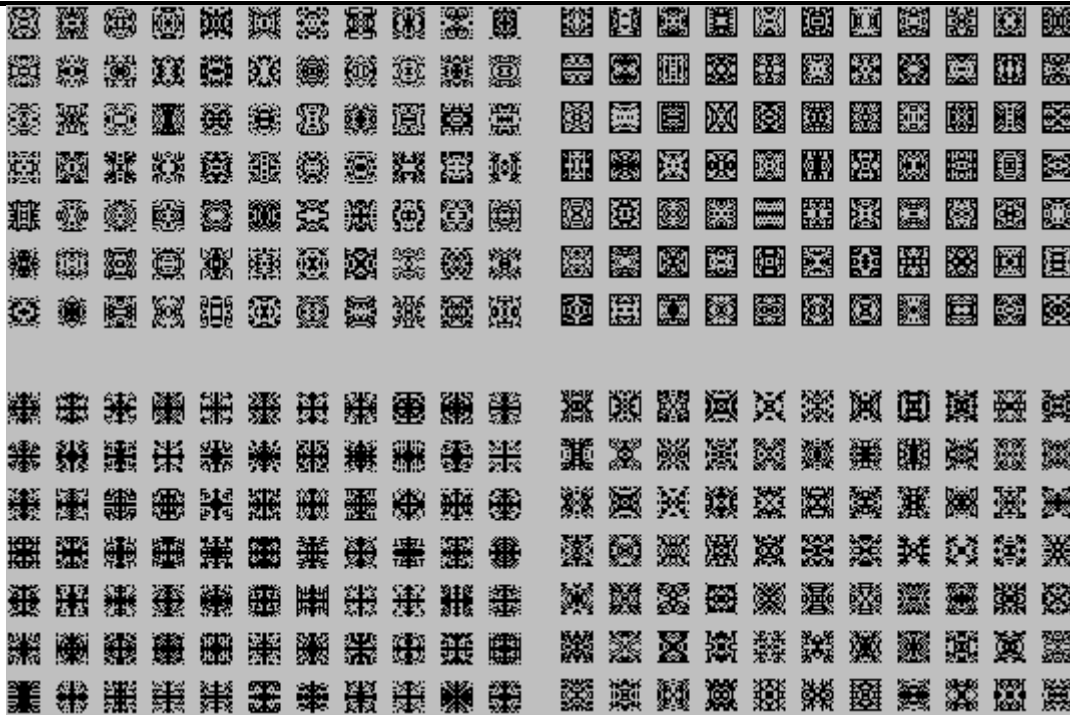
10 CLEAR 39999: LOAD "rnd" CODE 40000
20 RANDOMIZE USR 40000: LET z=40018
30 GO SUB 500: CLS
40 FOR y=0 TO 18 STEP 3
50 FOR x=0 TO 30 STEP 3
60 RANDOMIZE USR z
70 PRINT AT y,x;"AB": PRINT AT y+1,x;"CD"
80 NEXT x
90 NEXT y
100 GO TO 30
500 PAUSE 0: IF INKEY$="" THEN STOP
510 IF INKEY$="1" THEN LET z=40018: RETURN
520 IF INKEY$="2" THEN LET z=40032: RETURN
530 IF INKEY$="3" THEN LET z=40052: RETURN
540 IF INKEY$="4" THEN LET z=40071: RETURN
550 RETURN

```

Подчёркнутые символы в строке 70 набираются в графическом регистре.

Типы шаблонов переключаются клавишами "1"..."4". "0" - стоп, остальные клавиши - продолжить генерацию шаблонов старого типа.

Результат работы генератора шаблонов Вы можете увидеть на рисунках.



Для тех, кто не владеет (пока) ассемблером ZEUS, приводим шестнадцатеричный дамп (с контрольными суммами) кодового блока программы. Нули в последних адресах дампа - это зарезервированный 32-байтный буфер для работы программы.

Дамп файла "rnd"CODE 40000,200.

9C40	21	E5	9C	11	E2	9C	01	03	11
9C48	00	ED	B0	21	E8	9C	22	7B	C3
9C50	5C	C9	21	E8	9C	06	08	CD	91
9C58	CA	9C	77	23	10	F9	18	38	4D
9C60	21	E8	9C	06	08	3E	FF	18	04
9C68	05	CD	CA	9C	CB	FF	77	23	A0
9C70	10	F7	18	24	21	E8	9C	06	FA
9C78	07	CD	CA	9C	CB	C7	77	23	7A
9C80	10	F7	3E	FF	77	18	11	21	21
9C88	E8	9C	06	08	0E	80	CD	CA	DB
9C90	9C	B1	CB	19	77	23	10	F6	FD
9C98	21	E8	9C	11	F0	9C	01	08	7F
9CA0	08	C5	7E	1F	CB	11	10	FB	8D
9CA8	79	12	23	13	C1	0D	20	F1	E4
9CB0	21	E8	9C	11	FF	9C	0E	02	AD
9CB8	06	08	7E	12	23	1B	10	FA	3A
9CC0	0D	C8	21	F0	9C	11	07	9D	93
9CC8	18	EE	3A	E2	9C	57	3A	E3	96
9CD0	9C	32	E2	9C	82	57	3A	E4	AF
9CD8	9C	32	E3	9C	82	07	32	E4	60
9CE0	9C	C9	00	00	00	05	5A	25	65
9CE8	00	00	00	00	00	00	00	00	84
9CF0	00	00	00	00	00	00	00	00	8C
9CF8	00	00	00	00	00	00	00	00	94
9D00	00	00	00	00	00	00	00	00	9D

Улучшение считываемости программ с магнитофона.

© Горцуев Е.И., г. Куйбышев, Новосибирская обл.

У меня были проблемы чтения с магнитофона, особенно чужих кассет. Тогда я сравнил три схемы формирователей сигнала: "ZX-Ленинград-1", "Пентагон 48", "Вариант НЭТИ" (см. рис.1).

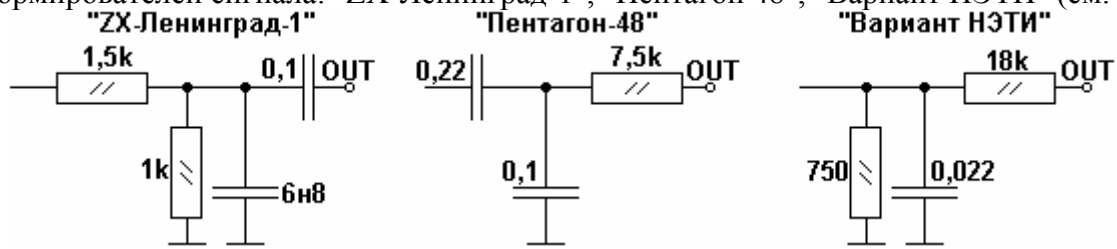


Рис. 1.

Свой "Пентагон" я доработал следующим образом (см. рис.2).

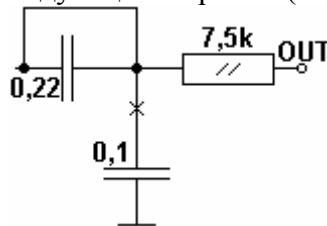


Рис. 2.

Результатом остался очень доволен. Даже сумел продать несколько кассет с записью. Рекламай не было. Потом доработал также ещё три компьютера, только в них ещё применил упрощенный вариант компаратора (см. рис.3) - хозяйева довольны.

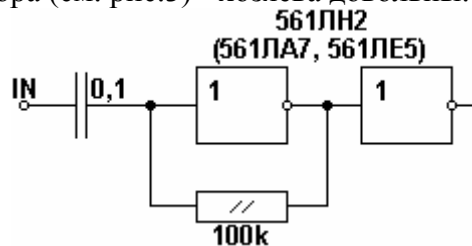


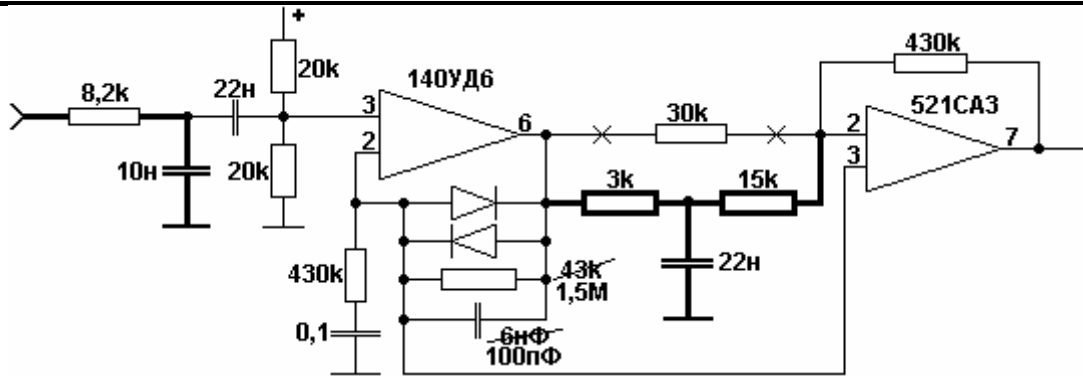
Рис. 3.

В качестве справки: "Вариант НЭТИ" отличается от "Новосибирского", больше известного у нас как "Спектр ПЛМ" отсутствием ПЛМ. В этой схеме формирователь RGB выполнен на 556РТ5, синхросмесь "защита" в 556РТ4, дешифратор ПЗУ, ОЗУ выполнен также на 556РТ4. Плата, наряду с "Новосибирским" вариантом, пользовалась большим успехом у любителей хорошей машины.

Свое знакомство с "ZX-48" я начал именно с "Варианта НЭТИ". Позднее "купился" на контроллер дисковода и порт "FF" в "Пентагоне-48", о чём иногда жалею. С начала 1993 года в Новосибирске получил распространение "Ленинград-1" с системной шиной. Насколько мне известно, это тоже работа ребят из НЭТИ. В этой версии "Ленинграда" учтены все доработки, которые я встречал в ваших сборниках. Так что у нас "Ленинград-1" переживает второе рождение.

© Александр Шарапов, Уфа, 1994.

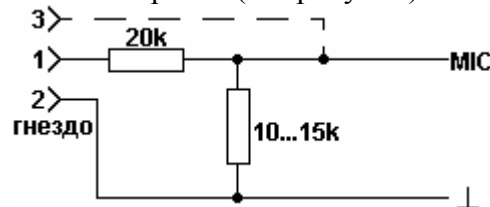
Привожу схему доработки входного каскада считывания компьютера "Ленинград" (см. рисунок). Утолщенными линиями выделены доработки. Мой компьютер "жует" теперь все подряд с любого магнитофона.



ИФК: Суть изменений, как видите, сводится к введению фильтров верхних звуковых частот. Основной достигаемый эффект при этом не завал верхних частот, а сопровождающее его выравнивание фазовой характеристики сигнала - она становится более благоприятной для обработки компаратором.

Между тем, наш корреспондент продолжает "магнитофонную" тему.

У многих есть импортные магнитофоны, не имеющие входного (для записи) гнезда. Устранить этот недостаток можно очень просто (см. рисунок).



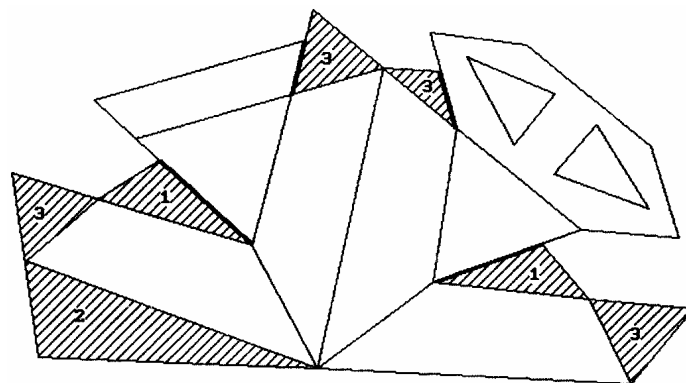
В любом удобном месте в корпус врезается гнездо (любое). Берутся два провода, идущие к микрофону. Если микрофон электретный (имеет 3 провода), то нужно взять провод идущий к входному конденсатору и на массу. Эти провода припаиваются к гнезду через делитель.

Если микрофон все же необходим, то можно:

- скоммутировать сигнал через дополнительный переключатель;
- вывести на гнездо третий провод напрямую и подключить микрофон через разъем.

ИФК: У Александра интересное хобби. Он прислал нам несколько разверток для склейки космических кораблей из программы ELITE. Особенно впечатляет его Viper, украшенный бортовыми номерами и другими элементами. Этот корабль теперь висит у нас над видеомонитором "Спектрума", слегка "маневрируя" в теплых потоках воздуха.

Вот его развертка:



Утолщенные линии - места разреза. По заштрихованным областям производится склейка. Думаем, принцип построения выкроек понятен нашим читателям и все желающие смогут "материализовать" космический флот.

Повышение совместимости схемы "Ленинград-1".

© Черкашин Л.В., г. Красноярск.

На тему повышения совместимости схемы "Ленинград-1" с исходным "Спектрумом" было сказано уже много. И все-таки может кому-то это и пригодится. Итак:

Проблема № 1. "Моргание" бордюра в такт со звуком на некоторых игрушках. Устранять можно двумя способами.

1. Установка диодов на IOWR и A0 (этот способ был уже описан, но иногда приходится подбирать номинал резистора на общий провод).

2. Установка вместо D39 (ТМ9) - 531ИР18 - ставится один к одному, нужно только на вывод ИР18 подать А0 с ближайшей точки (1 вывод D37,38). Единственная здесь проблема - ИР18 потребляет примерно 140 мА, не всем это может понравиться.

Проблема 2. Кемпстон-джойстик. Про его адресацию я писать не буду, она очень мало, где мешает. А вот его коммутация составляет эту самую проблему. В схеме "Ленинград-1" три старших разряда брошены в воздухе, хотя там должны быть 0. Если же их заземлить, то перестают загружаться очень многие программы. Здесь вторая часть проблемы. Так как в "Ленинграде-1" совпадают порт чтения с магнитофона и порт чтения с джойстика (это 11 и 10 выводы D38), то на 10 выводе в режиме чтения не должен быть 0, а в режиме игры на джойстике должен быть 0. Схемное решение таково: на D38 выводы 6 и 13 заземлить, а на вывод 10 подать сигнал с устройства чтения с магнитофона (нужно только посмотреть, что там имеется без сигнала на входе, если 0, то 10 и 11 выводы можно соединить перемычкой, если же "1", то через инвертор - см. рисунок).

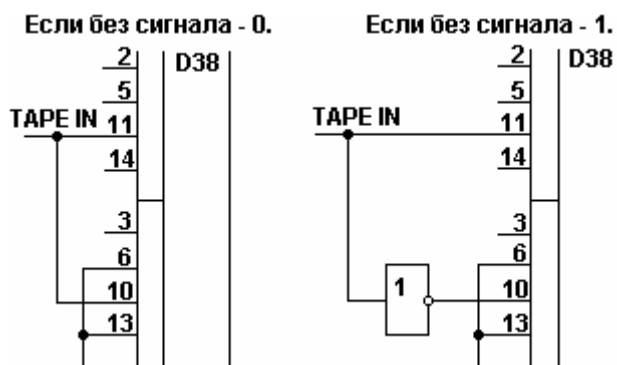


Схема опробована как минимум на 10-15 машинах. Работает. Есть ещё один вариант: ставить в цепь 10 вывода D38 транзистор и им коммутировать режим чтения или игры на джойстике.

И ещё об одной проблеме - второе ПЗУ (или дополнительное). При цене на микросхемы 27128 и 27256 примерно одинаковой, имеет смысл ставить 27256. В одну её половину "зашивается" стандартная "ОС" - 1982 г., а в другую Вы можете зашить что угодно: Турбо-90 с монитором, 1988 год - с режимом двойной плотности чтения либо ещё что-то. Коммутируя старший адрес микросхемы 27256, вы можете пользоваться любой системой с минимальными доработками компьютера.

© Андрей Бастраков, г. Челябинск.

Еще несколько слов о совместимости "Ленинграда-1". Чтобы устранить дрожание спрайтов в "ARKANOID-2", можно перерезать проводник, идущий на конденсатор RC-цепи формирователя INT, с вывода 8 триггера D8, и на этот вывод конденсатора подать сигнал КС с вывода 6 микросхемы D40 (ЗИ-HE).

© Дмитрий Барановский, г. Ростов-на-Дону.

Почему-то все считают недешифрованный внешний порт FE в "Ленинграде" преступлением первой степени. Хотя почти полная дешифрация достигается применением одной микросхемы 555ЛА4 по схеме:

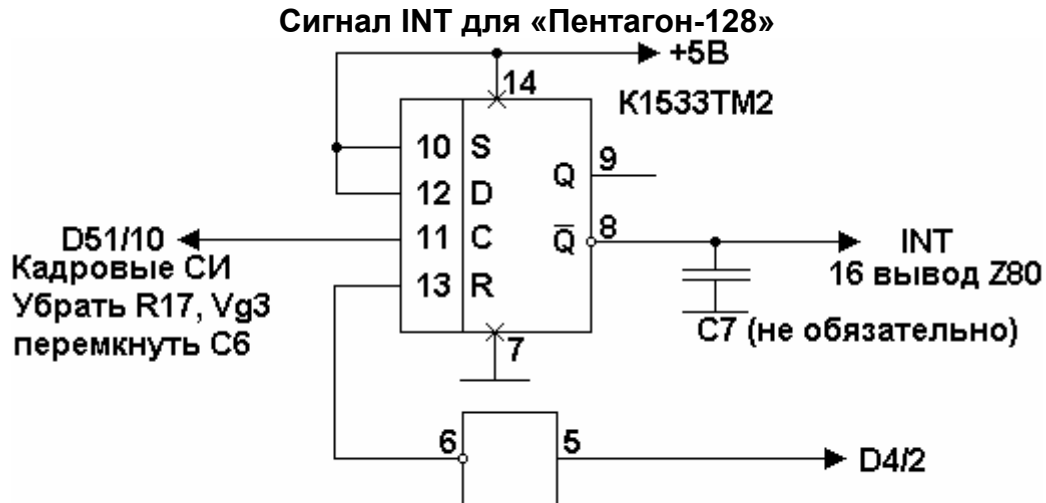


Вот и всё! Работы - минут на двадцать. Больше было разговоров.

Доработки компьютера "Пентагон-128".

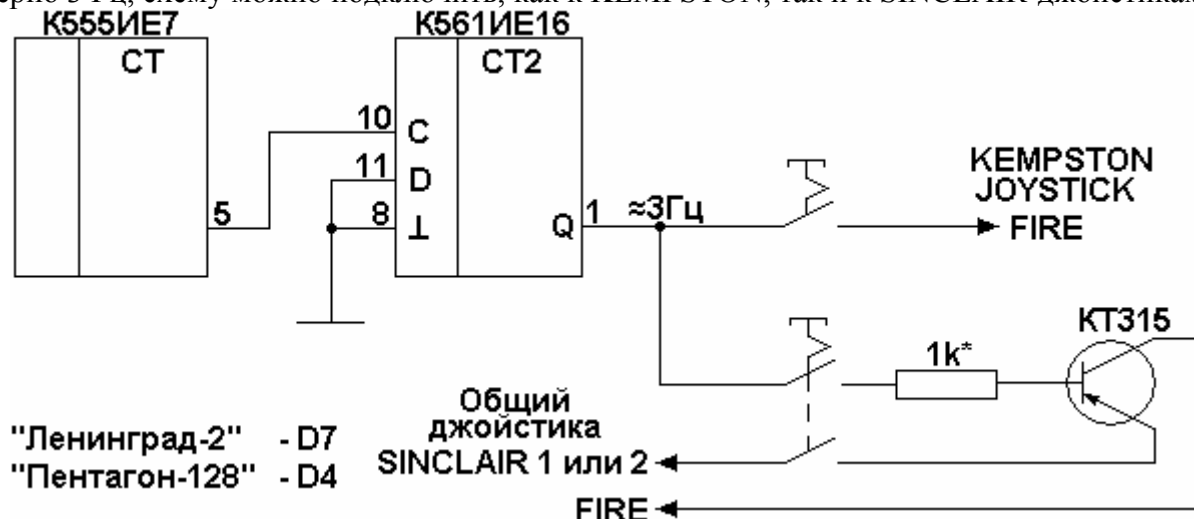
© Хромин И.Ю., г. Хабаровск.

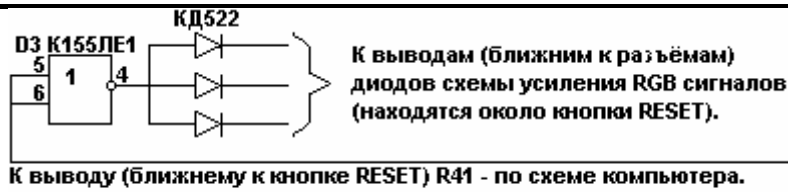
Я занимаюсь "Спектрумом" с лета 1992 г. Уже немного освоил машинный код не без вашей помощи, но мне больше нравится заниматься доработкой компьютера, сейчас у меня "Пентагон-128", до этого у меня был "Ленинград-2". Пентагон, в общем, не плохая машина, на сигнал INT сделан на нём не лучше, чем на "Ленинграде-1". В связи с этим на нём не идут, например, ARKANOID-2, дрожат спрайты, рисунки на бордюре вылезают не там где надо и т.д. Для доработки понадобится микросхема 1533TM2 и 1 инвертор ЛН1. Его можно взять в микросхеме D51, выводы 5 и 6 - неиспользуемые, или любой другой инвертор. После этой доработки не требуется никакой настройки, и нормально работают "ARKANOID-2", в "ACADEMY" рисунок на бордюр выводится с точностью до пиксела.



D51 K1533ЛН1

Для использования в играх, в которых количество боеприпасов не ограничено (типа "ZYNAPS" и т.п.), к компьютеру можно подключить несложную приставку для автострельбы. Даю схемы для компьютеров "Ленинград-2" и "Пентагон-128", но можно подключить к любым другим. K561ИЕ16 подключается к счетчикам синхрогенератора так, чтобы на её выходе сигнал был примерно 3 Гц, схему можно подключить, как к KEMPSTON, так и к SINCLAIR-джойстикам.





В Чебоксарской "Дельте" нужно удалить две перемычки около DD51

Кстати, если в Курской "Дельте" заменить ПЗУ на стандартное, то не будет никаких проблем с выводом на принтер (и не приходится ставить никакие резисторы), и повышается совместимость (начинают работать BOMB JACK и MIKIE).

Для ещё большей совместимости предлагаю схему формирователя сигнала INT.

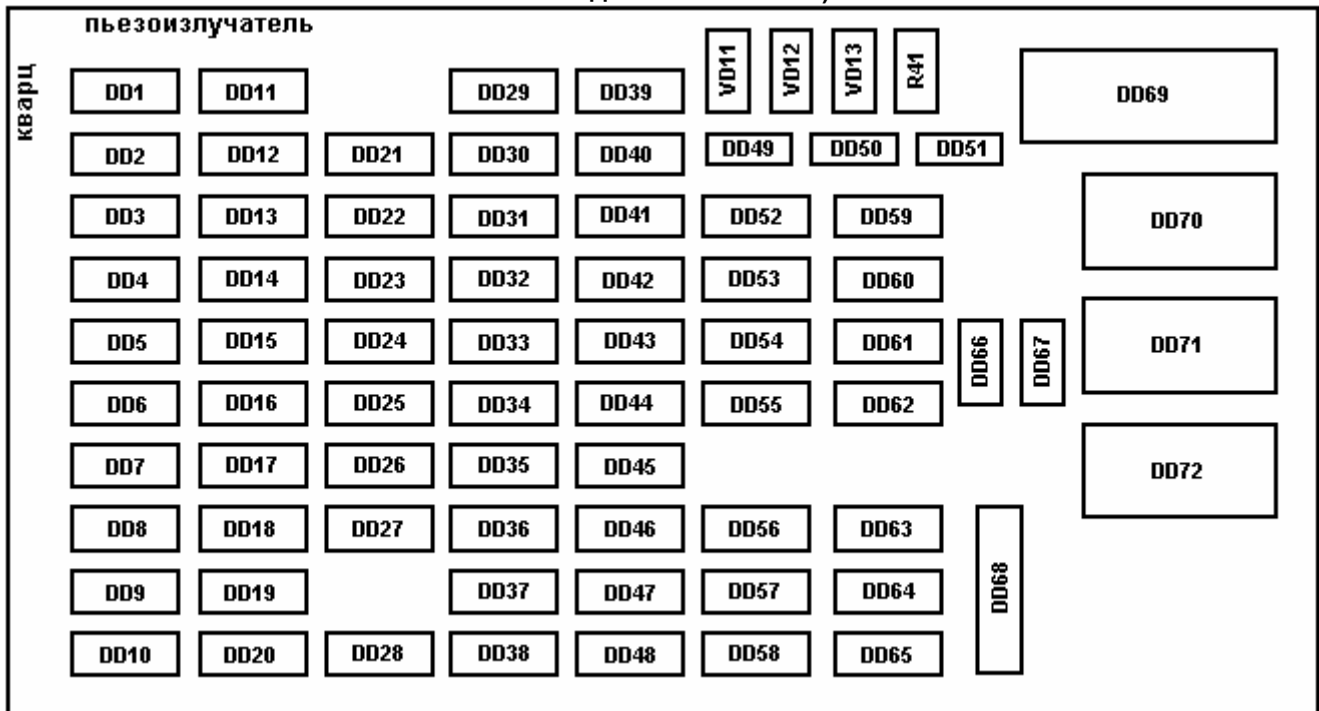
Формирование стандартного сигнала INT



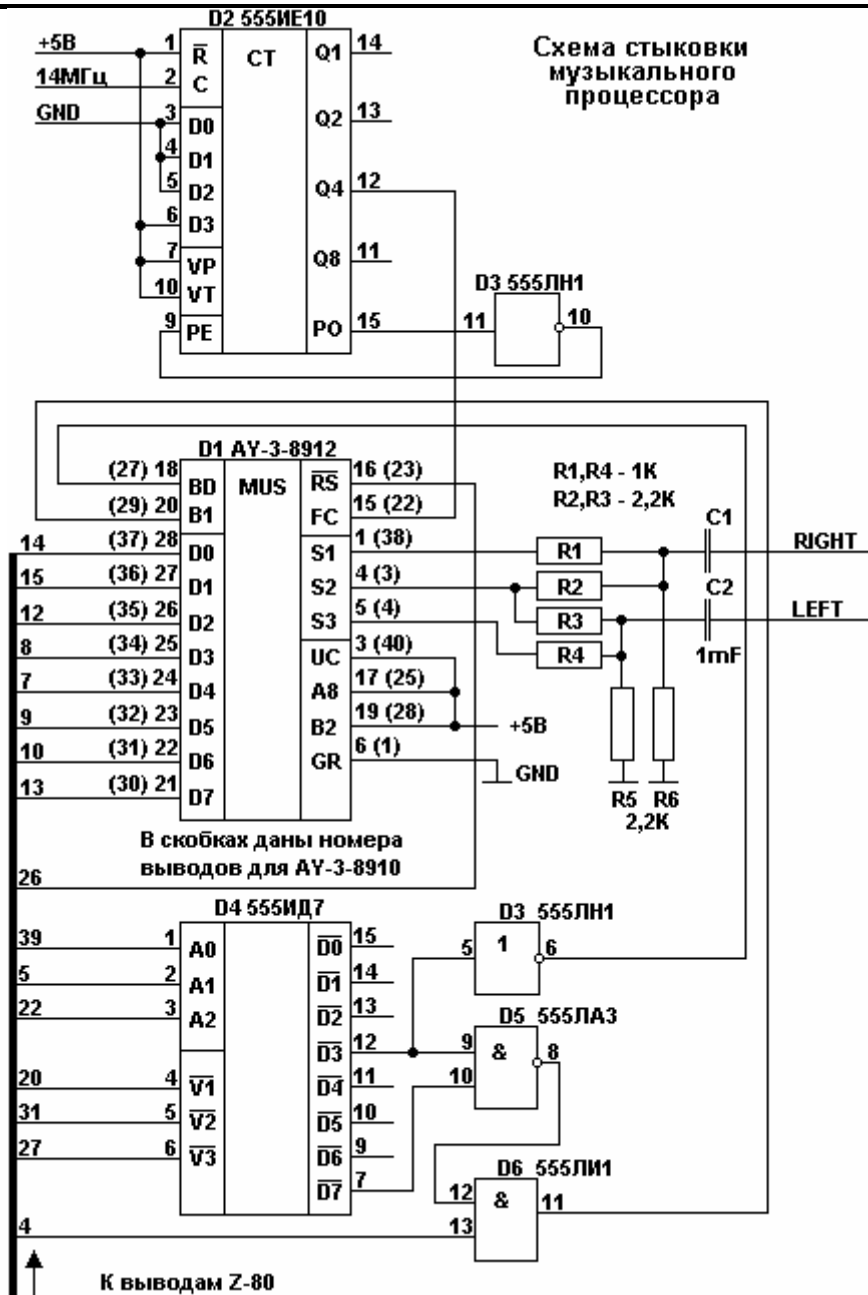
Обрезать сигнал INT, идущий к процессору.

Все схемы я "придумывал" на скорую руку, так что можно сделать "экономия в микросхемах". Но я думаю, что это поможет синклеристам.

Условный рисунок печатной платы (со стороны элементов) компьютера «Дельта-С» (даны только необходимые элементы)



Ещё я предлагаю схему подключения музыкального процессора (кстати, с ним несколько снижается совместимость с 48 версией Спектрума, т.к. некоторые программы, обнаружив музыкальный процессор, "думают", что это 128-килобайтная машина и... программа не идёт. (Это фрагмент из схемы компьютера "АТМ турбо-2".)



Я проверил свою "Дельту" (Чебоксарскую) со всеми перечисленными доработками, работает как зверь!!! Идут все программы из моей коллекции.

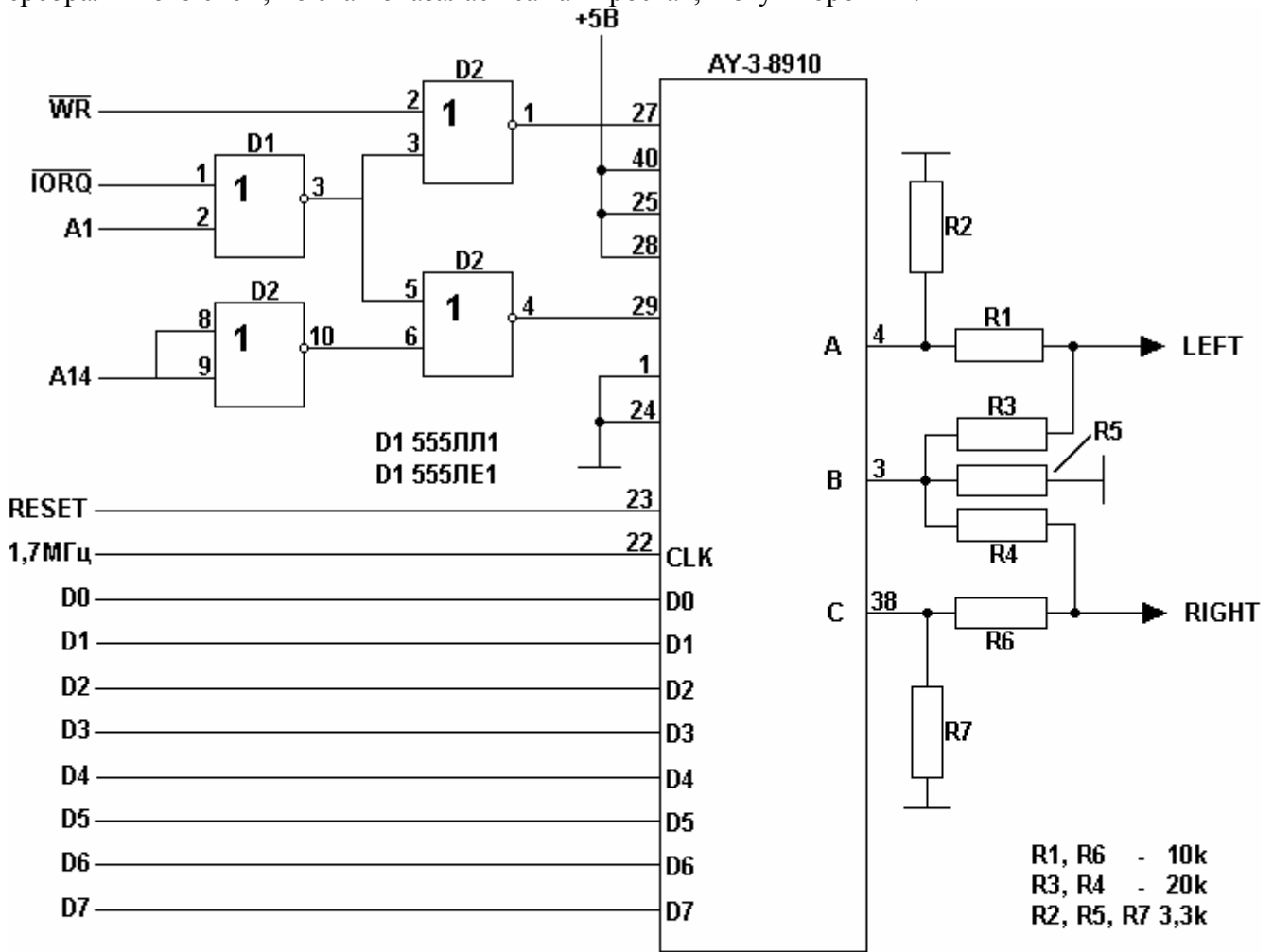
Я открыл на своей квартире клуб (студию) "NORTH", пишите по любым вопросам.

663255, г. Талнах-3, ул. Енисейская, 22-140 Студия
"NORTH" Игольникову Ивану Александровичу.

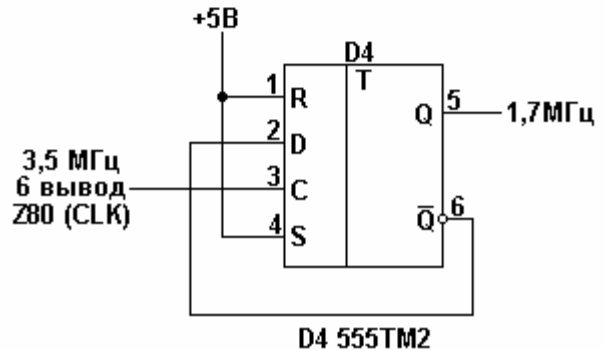
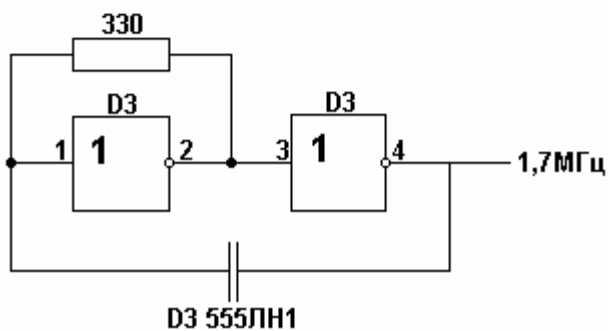
Подключение музыкального сопроцессора.

© Анатолий Гармаисанов, г. Белгород.

Поделюсь с читателями своим опытом подключения музыкального сопроцессора. Я перебрал много схем, но эта показалась самая простая, и звук хороший:



1,7 МГц можно взять с платы компьютера или:



СОВЕТЫ ЭКСПЕРТА

"SIM CITY"

© П. Троеглазов, Г. Троеглазов, Хабаровский Край, 1994.



Программа представляет имитатор городского хозяйства и позволяет Вам строить город и оперативно управлять его социальным и экономическим развитием. Вы выступаете в роли мэра города.

После загрузки на Вашем экране (сверху-вниз):

1. Игровое меню: SYSTEM, OPTIONS, DISASTERS, WINDOWS.
2. Пиктограммы.
3. (Слева) Выбранная пиктограмма и название города
4. (Справа) Месяц, год, наличие финансов.
5. Полоса оперативных сообщений.
6. (Центр экрана) SELECT DIFFICULTY LEVEL (выбор уровня сложности):
 1. EASY (легкий)
 2. MEDIUM (средний)
 3. DIFFICULT (трудный)

От уровня игры зависит как начальный капитал (20000\$, 10000\$, 5000\$), так и динамическая устойчивость развития города.

7. INPUT NEW CITY NAME (Ввести новое имя города).

Управление игрой - SINCLAIR или Q, A, O, P, Sp.

Изменять имя города нужно до выбора уровня сложности, так как после него игра стартует с предварительно заданным именем - HERESVILLE.

После того, как игра стартует, будет сгенерирован случайный ландшафт и Вы сможете начать управление. Режимы SYSTEM, OPTIONS, DISASTERS и WINDOWS служат для изменения параметров игры, введения в нее случайных внешних факторов (стихийных бедствий) и получения справочной информации. Пиктограммы же служат для выбора тех или иных объектов городского хозяйства.

Карта города не укладывается в один экран и необходим скроллинг, который можно производить в любое время курсорными клавишами, но если Вы хотите управлять игрой синклердждойстиком или клавиатурой (Q, A, O, P, Sp), то используйте для этого первую слева пикторграмму.

1. Пиктограммы.

1. Скроллинг карты.....-
2. BULLDOZE (бульдозер - очистка территории, подготовка к застройке, укрепление берегов), 1\$
3. ROAD (строительство автодороги).....10\$
4. POWER LINES (строительство линии электропередачи).....5\$
5. RAILROAD (железнодорожный путь).....20\$
6. PARK (разбивка парка, озеленение территории).....10\$
7. RESIDENTIAL (выделение территории под жилищное строительство).....100\$
8. COMMERCIAL (выделение территории под коммерческие структуры).....100\$

9. INDUSTRIAL (выделение территории под промышленные предприятия).....	100\$
10. POLICE STATION (строительство полицейского участка).....	500\$
11. FIRE STATION (строительство пожарного депо).....	500\$
12. STADIUM (строительство стадиона).....	3000\$
13. POWER STATION (электростанция).....	3000\$-5000\$
14. SEAPORT (морской порт).....	5000\$
15. AIRPORT (аэропорт).....	10000\$

Начинать игру следует со строительства электростанции. Без неё город существовать не может. Во избежание будущих неприятностей следует сразу предусмотреть, чтобы станция не оказалась в тех местах, где будут наиболее дорогие "спальные" и коммерческие районы будущего города. Лучше сделать ее в стороне и не пожалеть денег на строительство линий электропередачи.

При выборе пиктограммы "электростанция", наведите курсор на место, где хотите установить эту станцию и нажмите "FIRE". Вы получите возможность выбрать тип станции:

NUCLEAR POWER STATION (атомная эл.станция).....	5000\$
COAL POWER STATION (угольная эл.станция).....	3000\$
NONE (отказ от выбора)	

Станции различаются и по цене и по степени загрязнения окружающей среды и по производительности. Атомная станция "тянет" большее количество промышленных предприятий, чем угольная и обычно является более экологически чистой. Вокруг неё охотнее селятся люди. Но не дай Бог Вам пережить аварию на такой станции. Последствия радиации несколько сотен игровых лет сказываются на всём городском хозяйстве и ничем их удалить невозможно, пока радиоактивные вещества не распадутся сами.

Все остальные пиктограммы однозначны.

2. Управление игрой.

Разберем подробнее 4 опции, определяющие режим игры.

SYSTEM (Системное меню).

- ABOUT SIMCITY - сведения о программе и её авторах.
- START NEW CITY - рестарт игры с новым ландшафтом.
- USE OLD LANDSCAPE - рестарт игры со старым или выбранным ландшафтом.
- LOAD CITY - загрузить с ленты отложенную игру.
- SAVE CITY - сохранить игру.
- EXIT SIMCITY - выход из игры.
- EXIT MENU - выход из текущего меню.

OPTIONS (Изменение параметров игры).

AUTO-BULLDOZE (Авто-расчистка) - ON (OFF)

Обычно, прежде чем что-то построить в этой игре, надо подготовить поверхность с помощью бульдозера. Но этого можно избежать, включив режим AUTO-BULLDOZE. Это будет стоить немного дороже, т.к. в этом режиме готовится вся поверхность, даже и та, для которой этого делать не требуется. На низком уровне сложности это незаметно (денег достаточно), но на высшем уровне режим лучше отключать и работать ювелирно, экономя каждый доллар.

Используя пиктограмму BULLDOZE, Вы сможете сносить любые объекты и лесопарковые массивы. Чтобы снести участок леса, парка, дороги или линии электропередачи, достаточно установить квадратик курсора на требуемый участок и один раз нажать "FIRE". Если транспортные или энергетические пути были разрушены в результате бедствия или плохого финансирования их содержания, нужно сначала очистить бульдозером разрушенный участок, а затем, проложить трассу заново.

Чтобы снести часть какого-либо объекта, нужно нажать "FIRE" дважды. Чтобы снести объект целиком (аэропорт, жилой район, электростанции и т.п.) нужно установить курсор в центр объекта и, нажав клавишу "FIRE", задержать её на 1-2 секунды.

Кстати, мосты и подводные туннели, бульдозер сносит так же лихо, как и любые объекты на суше. Чудеса техники!

AUTO-BUDGET (Авто-бюджет) - OFF (ON)

Все бюджетные расходы должны Вами утверждаться. Когда денег много, этого можно и не

делать, упрощая игру. Включите этот режим. В условиях острой нехватки средств приходится экономить и часть запросов от полиции, пожарных и дорожных служб не удовлетворять. В этом случае режим следует отключить.

SOUND (Звук) - ON (OFF)

ZONE DEMOLISH WAIT (фиксация на зоне стихийного бедствия).

Если Вы хотите, чтобы в случае стихийного бедствия в каком-либо районе города, удаленного от того места, где установлен курсор, карта автоматически скроллировалась в зону разрушений, включите режим (ON). Это даст Вам возможность устранить последствия (например, обваловать бульдозером зоны пожаров), а потом вернуться к обычной работе. Если Вас не волнуют такие события, и Вы полагаете, что пожарные службы автоматически всё сделают сами, выключите этот режим, чтобы он Вас не отвлекал от основной работы.

SIMULATION SPEED (вызов меню "Установка скорости игры")

FAST - быстро;

NORMAL - нормально;

SLOW - медленно;

PAUSED - пауза.

Опцию "FAST" включать лучше тогда, когда жизнь города отлажена и идет постоянная прибыль в бюджет. Можно включить ещё и AUTO-BUDGET, тогда накопление денег будет идти автоматически, в ускоренном режиме.

Опция "PAUSED" останавливает время игры. Вы можете спокойно расставить объекты, соединить их электролиниями и дорогами, и затем, задав необходимую скорость, продолжить игру.

EXIT MENU (выход из текущего меню).

DISASTERS (Катастрофы, стихийные бедствия).

FIRE - пожар (Может происходить где угодно).

FLOOD - наводнение (Происходит на полях, граничащих с водой. Подготовка береговой линии с помощью бульдозера уменьшает вредное воздействие от этого бедствия).

EARTHQUAKE - землетрясение (Само по себе хоть и разрушительно, но не очень страшно, т.к. кратковременно. Страшнее те пожары, которые при этом возникают. Лучший способ борьбы - достаточное количество пожарных депо).

AIR CRASH - авиакатастрофа (Возникает только при наличии аэропорта. Основная опасность - пожары).

TORNADO - смерч. (Долгодействующее стихийное бедствие, также сопровождающееся пожарами).

NUCLEAR MELTDOWN - расплавление активной зоны ядерного реактора, утечка ядерного топлива и, соответственно, радиоактивное заражение местности. (Возникает достаточно редко, но имеет самые неприятные последствия. Нередко является следствием авиакатастроф или пожаров. Вокруг атомной станции целесообразно бульдозером делать зоны отсечки, чтобы исключить возгорание станции в случае пожара в соседних районах).

DISABLE ALL - отключение стихийных бедствий. Обеспечивает нормальное развитие города без катастроф.

EXIT MENU - выход из текущего меню.

WINDOWS (Справочный раздел).

MAPS - карты города;

BUDGET - отчёт по бюджету;

EVALUATION - результаты опроса общественного мнения.

Опция "MAPS" позволяет посмотреть карты города. Карта разделена на 9 участков, перекрывающих друг друга. Осмотр участков производится одновременным нажатием двух клавиш - SS+1 (2,3,...9). Слева экрана, под квадратом с номерами участков карты, находится меню выбора карты:

CITY - архитектурный план города;

POLLUTN - карта экологической обстановки (загрязнение); POLICE - полиция (карта защищенности города силами охраны порядка);

FIREDEP - карта защищенности города пожарными подразделениями;
 TRAFFIC - карта плотности грузопотоков;
 POWER - карта обеспечения города электроэнергией;
 TRANSPORT - карта обеспеченности районов транспортным сообщением;
 CRIME - карта криминогенной обстановки.

Опция BUDGET выводит в меню бюджета:

FISCAL BUDGET - отчет по доходам и расходам за год. Здесь можно изменить размер налога, взимаемого с горожан (от 0% до 20 %) - TAX RATE. Графа TAX COLLECTED показывает сумму собранного налога.

Вы можете изменить размер выплат на содержание дорог и железнодорожных путей - TRANS. Но этого делать не советуем, т.к. дороги начнут быстро разрушаться, придется в негодность и придется их восстанавливать, на что уйдет гораздо больше денег, чем Вы сэкономите на их содержании. Кроме того, плохое состояние дорог в течение длительного периода ведет к свертыванию производства, коммерции, понижению цены на земельные участки, а, следовательно, сокращает новые поступления в бюджет.

Оплату полиции (POLICE) не стоит снижать ниже 90%, - в городе ухудшается криминогенная ситуация. Оплата пожарной службы также должны быть не менее 90%. Только в игре с отключенными стихийными бедствиями (DISASTERS) можно позволить себе пренебречь защитой от пожаров.

Далее:

CASH FLOW - годовой доход;
 PREVIOUS FINO - состояние городской казны на начало отчетного периода;
 CURRENT FINOS - состояние городской казны на конец отчетного периода;
 GO WITH THESE FIGURES - выход в игру без изменения текущих параметров.

Опция EVALUATION даёт оперативную информацию, полезную для анализа хода развития города. Здесь экран разделён на четыре окна.

В первом окне - результаты опроса общественного мнения IS THE MAYOR DOING A GOOD JUST - сведения о Вашей популярности: YES (да) - одобряют Ваши действия, NO (нет) - не одобряют (%%).

Во втором окне - сведения об основных проблемах развития города - WHAT ARE THE WORST PROBLEMS (показано в процентах):

Traffic - нехватка дорог (проблема решается строительством новых дорог или, в более острых ситуациях, заменой автомобильных дорог на железнодорожные пути).

Crime - криминогенность (проблема решается созданием новых полицейских участков или увеличением финансирования действующих. Наиболее неблагоприятны с точки зрения преступности рабочие кварталы, где плотность промышленных предприятий выше. Обращайте на них особое внимание. Если хотите экономить на полиции, стройте промзоны в стороне от жилых районов. Защищать жилые районы важнее.)

Taxes - налоги. Если это Ваша основная проблема, то всё в порядке. Когда не на что жаловаться, население жалуется на непомерные налоги.

Pollution - загрязнение окружающей среды. (Проблема решается вынесением промышленных зон подальше от жилых кварталов и созданием лесопарковых защитных полос).

Fire - плохое противопожарное обеспечение. Решение очевидно - создание дополнительных пожарных депо.

House prices - высокая стоимость жилья. Эта проблема, строго говоря, проблемой не является. В принципе, Ваша цель в том и состоит, чтобы сделать город процветающим и стоимость жилья должна быть высокой. Ваши основные шаги - строительство дорог, парков, спортивных сооружений ведут к увеличению стоимости жилья и, соответственно, к повышению поступлений от налогов в городскую казну. Тем не менее, доводить дело до крайности не рекомендуется, иначе может сократиться прирост населения. Поэтому, если у Вас все в порядке и стоимость жилья является основной проблемой, отведите ещё несколько участков под жилую застройку. В ответ получите прирост населения.

Unemployment - безработица. Создайте дополнительные рабочие места, выделив участки

под промышленное строительство.

В третьем окне - информация о социальных процессах в городе.

POPULATION - количество жителей;

MIGRATION - миграция жителей (въезд-выезд);

ASSESSED VALUE - валовой доход города.

В четвертом окне справочная информация:

CATEGORY - категория;

LEVEL - уровень;

SCORE - очки.

По мере роста строительства и заселения будет меняться категория:

VILLAGE - деревня;

TOWN - поселок, маленький городок;

CITY - город средних размеров;

CAPITAL - большой город, центр страны или региона;

METROPOLIS - мегаполис - город очень больших размеров.

Полезные советы.

Внимательно следите за строкой оперативных сообщений и вовремя откликайтесь на требования, сообщаемые в ней (зоне требуются коммерческие центры, индустриальные центры, стадион, морской порт и т.п.). Если Вы долго не выполняете требования населения, снижается въезд новых жителей в город.

Во избежание загрязнения окружающей среды располагайте электростанции, морские порты и индустрию за чертой города.

Чтобы застроить всю карту и не испытывать затруднений с электроэнергией, необходимо иметь не менее 4-х станций. Нагрузка на станции должна быть изолирована. При нехватке энергии в каком-либо районе, объединение линий 2-х или 3-х станций ничего не даст. Нужно просто уничтожить лишние объекты или отключить район, где не хватает энергии, и поставить для него новую станцию.

Оптимальный налог 7%. Можно поднимать, не боясь забастовок, налог до 11%. Если Вам срочно нужна большая сумма денег, добейтесь (при нормальной сумме налога - 7%) устойчивых годовых доходов и постоянного притока населения. Затем, дождавшись конца года, в декабре, резко поднимите налог (20%!), а в январе, сразу снимите его до 11%. В феврале забастовщики успокоятся. Теперь каждый год снижайте налоги на одну единицу, пока не вернетесь к 7%. Таким способом, за несколько лет, можно удваивать и, даже утраивать бюджет.

В ответ на нашу публикацию о ролевых играх (см. ZX-РЕВЮ № 1, 1994г.) наши читатели прислали много писем с просьбами рассказать о них поподробнее или порекомендовать что-то из игр этого жанра, доступное на ZX-Spectrum.

Надо сказать, что игры типа RPG не характерны для машин с малой оперативной памятью, т.к. самой важной особенностью, выделяющей их из игр прочих жанров, является гигантский объём игрового пространства. Тем не менее, кое-что все-таки существует. В качестве примера мы даем здесь работу, присланную нашим читателем по типичной ролевой игре.

От себя добавим, что эти игры существуют очень давно, но сейчас переживают второе рождение. Дело в том, что видеоигровые приставки долгое время не могли подступиться к этому жанру и достичь уровня персональных компьютеров. Именно в этих играх проходил основной водораздел между компьютерами и приставками. В последний год ситуация стала меняться. Появление лазерных дисков и картриджей ПЗУ большой емкости (4-8 Мб) сделали эту проблему решаемой. Так, в прошлом году ведущие зарубежные компании по производству видеоприставок Nintendo и SEGA объявили, что жанр RPG является для них решающим (публике надоели бесконечные "каратеки" и имитаторы футбола и т.п.).

"HERO QUEST"

GREMLIN GRAPHICS LIMITED

© Василий Пахомов, п. Кубинка Московской обл.



Хотите очутиться в глубине древних времён, когда по земле ещё ходили рыцари и в мире было много волшебства? В те времена было много нечистой силы, колдунов и магов. И вот один злой колдун Моркар начал творить на земле разные тёмные дела. Он приносил несчастье многим людям, богатым и бедным. И тогда люди решили послать четырёх добровольцев на борьбу с этим колдуном. Каждый воин обладал не только физической силой, но и волшебной. Добровольцами стали BARBARIAN, DWARF, ELF и WIZARD.

И вот Вы попадаете в замок колдуна Моркара. В этом замке 14 этажей. Ваша задача пройти все и выполнить 14 заданий. Только тогда Моркар будет повержен. На каждом этаже Вас ждут различные противники и ловушки. Вам придется решать разнообразные задачи. По ходу игры Вы сможете заработать золотые монеты, которые можно использовать для покупки вооружения и доспехов.

Описание 14 этапов будет дано ниже. А сейчас перейдём к самой игре. Она сделана по типу настольных игр, но имеет некоторые преимущества перед ними. В этой игре есть неожиданность и непредсказуемость. Ходы делаются в зависимости от того, сколько очков выпало на костях. Это делается так: когда вы переходите к очередному персонажу, вверху начинают мигать различные числа в случайном порядке, для их остановки нужно нажать "огонь" и останется число, обозначающее количество очков - на сколько шагов сможет пройти герой.

В эту игру могут играть до четырёх человек, тогда у каждого будет свой персонаж, но можно играть в меньшем составе, только тогда надо поделить оставшихся героев между собой. Конечно, можно сделать меньшее количество персонажей, но тогда будет немного сложнее играть, потому, как каждый герой имеет какие-то свои характерные особенности. Теперь о самой программе. После загрузки появляется меню:

PLAY GAME (начать игру)
 BUY EQUIPMENT (закупить вооружение)
 CREATE CHARACTER (создание персонажа)
 SET CONTROLS (выбор управления)
 LOAD OTHER

Режим PLAY GAME служит для запуска самой игры. После выбора этого пункта появляется таблица заклинаний для волшебника (WIZARD) и для эльфа (ELF) в виде четырех классических стихий: воздух, огонь, вода, земля.

AIR SPELLS:	FIRE SPELLS:
CENIE	COURAGE
TEMPEST	FIRE OF WRATH
SWIFTWIND	BALL OF FLAME
EARTH SPELLS:	WATER SPELLS:
ROCK SKIN	SLEEP
HEAL BODY	VEIL OF MIST
PASS THRY ROCK	WATER OF HEALING

Сначала выбирается группа заклинаний для мага, потом группа для эльфа, остальные две группы тоже отдаются магу. В итоге получается, что у мага 9 заклинаний, а у эльфа - 3 заклинания.

После этого выбирают лабиринт. Все 14 миссий можно пройти не последовательно, а сделав выбор в меню. После загрузки миссии появляется описание миссии, а потом сама игра, но к ней мы перейдем ниже.

Следующая функция в начальном меню - BUY EQUIPMENT. Она служит для закупки вооружения, но чтобы что-нибудь купить, нужно золото, которое можно заработать по ходу игры. Чем лучше оружие, тем дороже оно стоит. Ниже приведено меню и цена оружия или доспехов.

WEAPONS (оружие)

STAFF (боевой посох)	100
SHORT SWORD (короткий меч)	150
BROAD SWORD (широкий меч)	250
BATTLE AXE (боевой топор)	400
CROSSBOW (лук)	350
HAND AXE (ручной топор)	150
SPEAR (копье)	150

ARMOUR (доспехи)

SHIELD (щит)	100
HELMET (шлем)	120
CHAIN MAIL (кольчуга)	450
ARMOUR (броня)	850
TOOLKIT (снаряжение)	250

TOOLKIT служит для обезвреживания ловушек. Для использования нужно подойти к ловушке и выбрать пиктограмму с мешочком. Если у этого воина есть TOOLKIT, то внизу будет надпись USE TOOLKIT, нужно навести стрелку на ловушку и нажать "огонь" - она исчезнет. Первоначально TOOLKIT есть только у гнома (DWARF), но его может купить любой другой воин.

CREATE CHARACTER - эта функция нужна для того, чтобы давать другие имена воинам. Она пригодится, когда в игру будут играть два и более человек. Ещё с помощью этой функции можно оживить некоторых павших героев, или, наоборот, вывести их из игры. Например, если в каком-нибудь лабиринте убили варвара (BARBARIAN), то после завершения лабиринта можно войти в это меню, подвести курсор к своему герою и нажать "огонь". Табличка с надписью "DEAD" ("мертв") сменится на его имя. Если же Вы, наоборот, хотите вывести героя из игры, подведите к нему курсор и нажмите "огонь". Табличка сменится на "DEAD".

SET CONTROLS служит для выбора управления. Но, к сожалению, с её помощью можно выбрать только KEMPSTON-джойстик. Стандартные же клавиши:

О – вверх Z – влево BREAK – огонь

К – вниз X - вправо

Но если у Вас SINCLAIR-джойстик, или Вы хотите играть на "синклер"-клавишах, то есть (6 - влево, 7 - вправо, 8 - вниз, 9 - вверх, 0 - огонь), то во время игры, когда будут появляться случайные числа (выбрасываться кости), нажмите "огонь" на SINCLAIR-джойстике или Q - на клавиатуре и управление перейдет на SINCLAIR.

Функция LOAD OTHER ещё полностью не исследована, но есть предположение, что с её помощью можно загружать собственные лабиринты, составленные в предназначенном для этого редакторе (если он существует).

Теперь перейдем к самой игре. После загрузки уровня появляется первый герой - BARBARIAN. Вы кидаете кости и делаете ход этим воином. Вы можете управлять им с помощью пиктограммы с нарисованными стрелками. Но можно управлять героем прямо на игровом поле, для этого нужно подвести стрелку на тот квадрат, куда Вы хотите направить воина.

Чтобы перейти к другому воину, нужно подвести стрелку к пиктограмме с его изображением.

Также на экране есть пиктограммы с изображением других предметов. "Карта" показывает изображение лабиринта. Исследованные участки на карте открыты. Карта открывается постепенно и поэтому чем дальше развивается игра, тем она становится всё больше интересной. На карте также изображаются Ваши герои, враги и разные объекты. Пиктограмма "глаз" служит для осмотра помещений. При её использовании появляется меню с выбором того, что можно искать. Искать можно TRAP, DOOR (ловушки, двери) и TREASURE (сокровища). Во время выбора TRAP, DOOR выявляются ловушки или потайные двери, если только они в этой комнате есть. При выборе функции TREASURE Вы можете найти золотые монеты или полезные снадобья, но также можете и вызвать врага, с которым придётся сражаться. Так что, эта функция может принести добро, а может и зло. Пиктограмма "мешочек" служит как опция "ИНВЕНТАРЬ" и показывает то, что есть у данного персонажа. Он может иметь:

POTIONS (полезные снадобья, эликсиры). Если у Вас есть подобные вещи, то ими можно воспользоваться. Вот список некоторых эликсиров, которые могут быть найдены в лабиринте:

HEALING (целебный эликсир) - дает воину дополнительные "жизни".

SPEED - увеличивает скорость героя (за один ход он сможет сделать больше шагов).

HOLY WATER (святая вода) - дает воину дополнительную силу, то есть у него становится больше шансов победить врага.

HEROIC BREW (эликсир героизма) - дает возможность воину дважды за один ход нанести противнику удар. После первой атаки появляется меню с выбором. Хотите ли Вы атаковать ещё раз? Если вы врага не убили, то выбирайте "YES", если же враг убит - "NO".

STRENGTH (эликсир силы) - этот напиток даёт почти то же самое, что и святая вода.

RESILIANCE (упругость) - повышает стойкость к ударам противника.

Следующая функция QUEST TREASURE нужна для использования найденных сокровищ и ценных предметов. Но эти сокровища очень сложно найти в лабиринте.

WEAPONS - нужна для использования оружия, которое вы купили после прохождения какого-нибудь лабиринта.

ARMOUR - аналогична предыдущей, только для доспехов.

SPELLS - эта функция есть только у эльфа и мага - она служит для использования заклинаний. Существуют заклинания, лечебные которые используют для свои героев, но есть и боевые - для врагов. Заклинания отмеченные знаком (-) применяют для врагов (боевые), а знаком (+) - для своих (лечебные).

CENIE (-) магический удар по противнику.

TEMPEST (-) усыпляет врага (некоторое время он не дерётся).

SWIFTWIND (+) удваивает скорость героя.

COURAGE (+) дает герою дополнительное мужество.

FIRE OF WRATH (-) магический удар.

BALL OF FLAME (-) магический удар.

ROCK SKIN (+) повышает стойкость героя.

HEAL BODY (+) лечебное заклинание.

PASS THRY ROCK (+) повышает стойкость к ударам.

SLEEP (-) усыпляет врага.

VEIL OF MIST (+) ставит магическую защиту от врагов.

WATER OF HEALING (+) эквивалентно применению лечебного эликсира.

Как видите, многие заклинания имеют действия, аналогичные эликсирам. Разница в том, что заклинаниями могут пользоваться только герои, обладающие магическими способностями от рождения (маг и в меньшей степени эльф). Эликсиром же могут пользоваться любые герои, независимо от своего игрового амплуа.

За один ход можно пользоваться только одним заклинанием. После выбора заклинания укажите на карте, на кого хотите обратить это заклинание. После этого заклинание исчезает.

Пиктограмма с изображением меча означает атаку. Чтобы атаковать врага, надо встать на соседнюю с ним клетку по горизонтали или вертикали. Потом выбрать "меч" и на карте указать на врага, находящегося рядом с Вашим героем. После этого появляются изображения воинов, а внизу изображается результат сражения. В зависимости от того, сколько щитов и мечей у воина и врага, можно определить результативность атаки. Пользоваться этой функцией можно только один раз за ход.

Есть ещё маленькая тонкость в использовании пиктограмм "глаз" и "меч". Ими лучше всего пользоваться либо в начале хода, когда герой ещё не сделал ни одного шага, либо когда у него шаги кончились, потому что выбор этих опций между шагами прекращает ход.

А теперь перейдём к самим миссиям:

1. THE MAZE.

Четырем героям нужно найти выход из этого лабиринта. Тому, кто первым найдет выход и выйдет из него, выдается премия в размере 100 золотых монет.

2. THE RESCUE OF SIR RAGNAR.

Сэра Рэгнара похитили злые орки. И четыре воина отправляются на его спасение. Тому, кто найдет и спасет сэра Рэгнара, он заплатит 200 золотых монет.

3. LAIR OF THE ORC WARLORD.

Принц Магнус посылает рыцарей в логово Орков. Там находится предводитель орков - Улаг, который и похитил сэра Рэгнара. Победителя Улага, ждет премия в размере 100 золотых монет.

4. PRINCE MAGNUS GOLD.

Злые силы украли у принца Магнуса сундуки с золотом. Героев посылают в подземелья Черной Горы, чтобы они нашли похищенные сундуки. За каждый найденный и принесённый сундук выдается награда в 200 золотых монет.

5. MELAR'S MAZE.

Много лет тому назад сильный колдун по имени Мелар создал талисман, который обладает огромной волшебной силой. Злой колдун Моркар завладел талисманом. Рыцарей посылают на поиск волшебного талисмана. Талисман окружает множество ловушек, его сторожат охранники, обладающие магией.

6. LEGACY OF ORC WARLORO.

Грак решил отомстить людям за то, что они убили Улага. Он заманил их в лабиринт и заточил в тюрьму. У них отобрали всё оружие. Ваша задача - выбраться из тюрьмы, найти своё оружие и выбраться из лабиринта.

7. THE STONE HUNTER.

Исчез личный колдун принца Карнель. Принц боится, что на него могут напасть злые силы. Он посылает Вас на поиски Карнеля и, если он вернётся живым и невредимым, Вам заплатят 100 золотых монет.

8. THE FIRE MAGE.

Балур - огненный маг помогает Оркам в набегах на мирные поселения. Колдун принца не в силах противостоять огненной магии. И тогда принц посылает Вас на борьбу с Балуром. Тот, кто убьёт его, получит премию в 150 золотых монет.

9. RACE AGAINST TIME.

Проводник заманил Ваших героев в ловушку и сбежал. Воины оказались в западне. Вы

должны сбежать, иначе погибнете в этой тёмной забытой дыре.

10. CASTLE OF MYSTERY.

Страшный колдун Оллар обнаружил вход в золотой рудник. Там он построил магический замок. Потом он поставил возле входа охранников и расставил ловушки. Вы должны найти вход в этот замок.

11. BASTION OF CHAOS.

Бесконечные налёты Орков и гоблинов надоели принцу. И он решил послать добровольцев в главную крепость врага, чтобы они уничтожили врагов. За уничтожение гоблина дается 10 золотых монет. За орка - 20 золотых монет. За фимира или воина - 30 золотых монет.

12. BARAK TOR, HOME OF THE WITCH LORD.

Произошел конфликт между орками и императором. И император решил послать воинов найти легендарный талисман THE ANCIENT STAR в логове орков. Эта звезда упоминалась в старой королевской легенде и служила оружием в борьбе с очень сильными злыми силами. В лабиринте обитает WITHLORD (повелитель ведьм), которого можно попробовать уничтожить найденным оружием.

13. QUEST FOR THE SPRINT BLADE.

На императора опять стали нападать орки и он опять посылает вас в лабиринт. Вы должны победить как можно больше врагов. Другая задача - найти THE SPRINT BLADE.

14. RETURN TO BARAK TOR.

В этой миссии Вы должны найти WITHLORD'a (повелителя ведьм) и любой ценой его уничтожить, но это не всегда удается. Этой миссией Вы заканчиваете приключения в лабиринте.

В этой игре есть еще много особенностей, которые не имеет смысла описывать, чтобы не потерялся интерес. Игра очень увлекательна и Вы не пожалеете о проведенном времени.

МОЗГОВОЙ ШТУРМ

В "ZX-РЕВИЮ" № 1 за 1994 год в разделе "ФОРУМ" мы опубликовали вопрос Балакина Е.А., он спрашивал об игре "MAGNETRON", его интересовало, что делать с реакторами и как уничтожить дроидов неправильной формы. Мы получили три письма об этой игре. Наиболее полные данные содержатся в письме Виталия Бородулина из Пскова. Благодарим также Михаила Сотвина (возможна ошибка, фамилия неразборчива) из Тулы и Болгова А.А. из г. Новомосковска Тульской обл. Мы обобщили и систематизировали присланный материал и предлагаем эту коллективную работу читателям.

MAGNETRON

Ваша задача - отключить 4 реактора в секторе (а не уничтожить всех роботов, как в QUAZATRONe). Всего в секторе находится 4 реактора: Альфа, Бета, Гамма и Дельта.

Реакторы.

Необходимо их уничтожить, а для этого надо дать им недопустимую нагрузку или вообще устроить "короткое замыкание". Реактор отключается после изменения одного или нескольких цифр в окошках (обычно, когда сумма цифр либо отрицательна, либо больше "+6"). "Коротнуть" реактор можно, уравнив его потенциал.

Например: -3 0 +2 +1

или так: +4 -3 +1 -2

Четыре вывода в сумме дают 0, реактор отключается.

Перегрузить реактор можно, сделав общий потенциал положительным, вернее, так, чтобы выводы были со знаком "+". Можно поступать так: чтобы перегрузить реактор, надо изменить цифру со знаком "-", а отключить - цифру со знаком "+". Одним словом, Вы можете экспериментировать, как хотите и вскоре научитесь отключать реакторы.

Всего в секторе 4 реактора и все надо вывести из строя. Когда это будет выполнено, сектор поменяет свой цвет (как в QUAZATRONe).

После отключения всех реакторов в секторе можно воспользоваться лифтом - той площадкой, на которой появляется KLEPTO в начале игры (или другими аналогичными площадками).

А сейчас некоторые данные о тех объектах, которые используются роботами в игре.

WEAPON (Оружие).

PROTON DISK - HIGH ENERGY WEAPON DESIGNED FOR VOID COMBAT

Высокоэнергетическое оружие, предназначено для боевых воидов. Стреляет протондисками.

PROTON DISK2 - HIGH ENERGY PROTON DISK WITH UPRATED SPEED

Оружие с увеличенной начальной скоростью диска и более мощное, чем первая серия.

CRUISER - PROTOTYPE INTELLIGENT WEAPON FOLLOWS TERRAIN BUT STILL UNDER TRIAL

Прототип умного оружия преследования, но до сих пор под испытанием. Стреляет сгустками энергии. Они (сгустки) хорошо отскакивают от стен и долго летают.

PLASMARON - HIGH ENERGY WEAPON WITH HIGH SPEED

Высокоэнергетическое оружие с высокой скоростью. Аналог "DISINTEGRATORa" (в QAZATRONe). Мощное оружие.

MORTAL - HIGH ENERGY SPHERE

Высокоэнергетические шары. От "MORTAL" - смертельный. Стреляет черными "шариками", обладающими высокой скоростью и мощностью.

ANTIDRIVE - DESIGNED TO STOP ALL DRIVE UNITS IN RANGE

Предназначено для остановки всех двигателей. Все роботы выдерживают 4-6 попаданий ANTIDRIVom, даже защищенные от него.

BOUNCING BOMB - LONG RANGE ENERGY SPHERE

Дальнобойное, стреляет энергетическими шарами.

SEEKER - PROTOTYPE WEAPON WITH INTRUDER SEEKER DEVICE

Прототип оружия с элементами самонаводящегося устройства. Это оружие сходно с DISINTEGRATOR (в QUAZATRONe), но сильнее его, т.к. поражает с первого попадания и имеет возвратное движение как у бумеранга, кстати, по форме снаряд тоже похож на бумеранг. SEEKER -страшное оружие, никогда не используйте его, если на экране нет вражеских дроидов. Если Вы выстрелите им в этот момент, оно уничтожит Вас самих, если Вы не умеете от него скрыться на другом экране.

DEVICE (Дополнительное оборудование).

DUROCELL - HIGH ENERGY BACKUP POWER SUPPLY

Высокоэнергетическое устройство для повышения запасов энергии.

OVERDRIVE - DESIGNED FOR UPRATED DRIVE PERFORMANCE.

Предназначено для повышения летных качеств двигателя.

SUPERSHIELDS - DESIGNED TO PROTECT AGAINST ENERGY WEAPON

Предназначено для защиты против энергетического оружия.

ULTRARAM - DESTROYS ALL CHASSIS ON CONTACT

Разрушает все шасси при контакте.

ANTIGRAV - MAKES CHASSIS BOUNCE

Делает шасси прыгающим.

GRAPPLE - COPY OF INTRUDER GRAPPLE DEVICE STILL UNDER TRIAL

Устройство для вторжения во вражеского дроида и копирования (использования) его "внутренностей", до сих пор под испытанием. Ниже подробно приведена методика его применения.

ANTIFREEZE - DESIGNED TO COUNTER ANTIDRIVE WEAPON

Предназначено против оружия антидрайв.

Роботы.

Их можно уничтожить оружием или захватить и "взломать" с целью использования "внутренностей", применяя GRAPPLE-блок. В этом случае, когда Вы вступаете в контакт с роботом, появляется таблица, в которой нужно будет за определенное время расставить элементы по порядку (по горизонтальным строкам - одинаковые), причём времени выдается тем меньше, чем выше класс робота, взламываемого KLEPTO.

Если все элементы в таблице расставить правильно, то загорятся все три надписи и появится новый KLEPTO, которым Вы будете управлять. Этот робот будет иметь все характеристики взломанного Вами робота. При захвате дроида Вы не можете взять его внутренности на выбор (как в QUAZATRONe), он отдается Вам весь с запасом энергии, так что (в отличие от QUAZATRONa) никогда не захватывайте дроида классом ниже. Если после взломки робота высокого класса, вы взламываете робота низкого класса, то будете иметь характеристики последнего, что не очень выгодно. Но есть выход: в таблице нужно сложить строку DETONATOR, не складывая остальных, в этом случае у KLEPTO остаются характеристики последнего взломанного робота, а того, с которым он вступил в схватку, он уничтожает, подзарядившись от него энергией. Для подзарядки робота высокого класса нужно будет либо взломать робота близкого к нему класса, либо нескольких - низшего класса.

Характеристики роботов.

Найти описание каждого дроида можно, входя в монитор на уровнях. Информация выдается лишь о тех роботах, класс безопасности которых ниже Вашего в момент подключения к монитору (классы безопасности будут рассмотрены ниже вместе с описанием роботов).

SECURITY CLASS EPSILON (класс безопасности ЭПСИЛОН).

VOID DEVICE (оборудованный воид).

V5 - GENERAL VOID WORKER SUITABLE FOR ALL MENTAL WORK

Обычный рабочий воид, годный для всех низких работ.

DRIVER: HOVERDRIVE

POWER: CYBATIC 1

WEAPON: PROTON DISK

DEVICE: DUROCELL

V4 - SPECIAL VOID ENGINEER EQUIPPED WITH UPGATED WEAPON FOR DEMOLITION DUTY

Специальный воид, инженер, оснащенный улучшенным оружием для разрушения.

DRIVER: MAGNADRIVE
 POWER: CYBATIC 2
 WEAPON: PROTON DISK 2
 DEVICE: DUROCELL

V3 - FAST LIGHT PROID SIUTABLE FOR MENIAL WORK

Быстрый, лёгкий дроид, подходящий для низких работ.

DRIVER: ELECTODRIVE
 POWER: CYBATIC 3
 WEAPON: PROTON DISK
 DEVICE: DUROCELL

V2 - MENIAL DROID WITH HIGH CLASS DRIVER UNIT FOR HEAVY WORK

Низкий дроид с высоким классом двигателя для тяжёлых работ.

DRIVER: INTERPHASE
 POWER: CYBATIC 3
 WEAPON: PROTON DISK
 DEVICE: DUROCELL

SECURITY CLASS DELTA (класс безопасности ДЕЛЬТА).

PURSUIT CYBORG (киборг преследования).

P4 - NEW CLASS OF HIGH SPEED SECURITY DROID EQUIPPED WITH OVERDRIVE DEVICE

Новый класс с высокой скоростью, дроид безопасности, оснащенный дополнительным устройством овердрайв.

DRIVER: ASTRON
 POWER: CYBATIC 4
 WEAPON: PROTON DISK 2
 DEVICE: OVERDRIVE

P3 - UPATED SECURITY DROID WITH PROTOTYPE INTELLIGENT WEAPON

Дроид безопасности с прототипом умного оружия.

DRIVER: ASTRON
 POWER: CYBATIC 4
 WEAPON: CRUISER
 DEVICE: OVERDRIVE

P2 - SPECIAL SECURITY DROID EQUIPPED TO COUNTER ANTIDRIVER WEAPON

Специальный дроид безопасности, оснащенный устройством противодействия противодвигательному оружию.

DRIVER: SYNCRON
 POWER: CYBATIC 4
 WEAPON: PLASMARON
 DEVICE: ANTIFREEZE

P1 - SPECIAL SECURITY DROID EQUIPPED WITH ULTRARAM

Специальный дроид безопасности, оснащенный ультраарамом.

DRIVER: SYNCRON
 POWER: CYBATIC 4
 WEAPON: PLASMARON
 DEVICE: ULTRARAM

GUARD CYBORG (охранный киборг).

G3 - DROID CLASS SUITABLE FOR COMBAT BUT HEAVY CHASSIS MAKES THIS DROID MORE SUITABLE FOR GUARD AVTV

Класс дроидов, предназначенный для боев, но тяжёлое шасси делает этих дроидов более пригодными для охраны. Этот дроид - просто ужас. Имеет приспособление - GRAPPLE UNIT и при попытке ближнего боя он уничтожает KLEPTO. Но его все-таки можно расстрелять, только сначала надо обзавестись оружием MORTAL - теми самыми "черными шариками".

DRIVER: CYCLIC
 POWER: NEURON 1
 WEAPON: MORTAL

DEVICE: GRAPPLE

G2 - A HIGH DROID WITH UPRATED PERFORMANCE IN LOW GRAVITY

Высокий дроид с лёгкими качествами для низкой гравитации. Этот дроид может быть уничтожен в GRAPPLE-схватке или просто расстрелян.

DRIVER: CYCLIC

POWER: NEURON 1

WEAPON: MORTAL

DEVICE: DUROCELL

G1 - COMBAT DROID WITH HEAVY DEVICE AND WEAPON UNITS

Боевой дроид с тяжёлым оборудованием и оружием. Их надо по возможности остерегаться.

DRIVER: CYCLIC

POWER: NEURON 1

WEAPON: ANTIDRIVE

DEVICE: SUPERSHIELD

SECURITY CLASS GAMMA (класс безопасности гамма).

FURY ROBOT (яростный робот).

F3 - NEW PROTOTYPE COMBAT DROID RUSHED INTO SERVICE COUNTER INTRUDER DROIDS

Новый прототип боевого дрoида, предназначенный для борьбы против вторгшихся дроидов.

DRIVER: INTERPHASE

POWER: NEURON 2

WEAPON: BOUNCING BOMB

DEVICE: SUPERSHIELD

F2 - ULTRARAM VERSION OF THE PROTOTYPE FURY DROID STILL UNDER TRIAL

Версия ULTRARAMa, прототип яростного дрoида, до сих пор под испытанием.

DRIVER: INTERPHASE

POWER: NEURON 2

WEAPON: BOUNCING BOMB

DEVICE: SCRAMBLE

SECURITY CLASS BETA (класс безопасности БЕТА).

ELITE ROBOT (элитный робот).

E3 - HIGH CLASS DROID EQUIPED WITH NEW ANTIGRAV DEVICE AND PROTOTYPE SEEKER WEAPON

Высший класс дроидов оснащенный новым устройством антиграв и прототипом самонаводящегося оружия.

DRIVER: ULTRAPHASE

POWER: NEURON 3

WEAPON: SEEKER

DEVICE: ANTIGRAV

E2 - HIGH CLASS DROID DESIGNED TO DESTROY INTRUDER DROIDS

Высший класс дроидов, предназначенный для разрушения вторгшихся дроидов.

DRIVER: ULTRAPHASE

POWER: NEURON 3

WEAPON: SEEKER

DEVICE: ANTIFREEZE

E1 - ?

Этого дрoида только видели, но захватить не удалось, поэтому о его "внутренностях" пока данных нет.

КОМПЬЮТЕРНАЯ НОВЕЛЛА

"LITTLE PUFF"

CODE MASTERS

© Дмитриев Олег, г. Москва, 1994.



Привет! Я маленький волшебный дракончик Пафф. По своей натуре я очень общителен и большой любитель путешествовать.

Однажды после длительного путешествия я вернулся домой и застал в нашем королевстве большие перемены...

У входа в нашу страну стоял какой-то тип. Это был охранник. Но что ему нужно? Оказалось, что охранник без пропуска никого не пускал в королевство. Что же мне делать? Свой-то я ещё весной посеял в нашем лесу. Ничего не поделаться, придётся мне идти в лес за пропуском. Хорошо, что я знал эти места, и у меня была с собой карта местности. С ней-то мне намного легче будет найти пропуск.

В траве я заметил бутылочку с магическим напитком. Я решил выпить его. И что же произошло? Я стал огнедышащим! По дороге я съел немного ягод и сам того не заметил, как налетел на трухлявый пенёк. Придя в себя после удара, я решил перепрыгнуть его, но он оказался очень высоким, и мне пришлось его спалить. Перепрыгнув через угли, я пошел дальше. По дороге я увидел яму, но решил не прыгать в нее. За ямой оказался тупик. Осмотревшись, я увидел конверт

и бутылку с газировкой. Конверт мне пока не достать, так как его охраняет кокос. А вот газировка... Я знал, что она лишает меня огненного дыхания. Но я был добрым драконом и никому не хотел причинять зла, поэтому, не задумываясь, осушил бутылочку, а заодно и удалил жажду.

Теперь надо прыгать в яму, но на карте показано, что внизу находится ловушка и надо быть осторожным, чтобы не попасть в неё. Прыжок оказался удачным. Пройдя налево... Здесь наверняка полно всякой нечисти. О! Я так и знал! Вот одно из них. Какой-то ёжик! В испуге я запрыгнул на стоявшую рядом коробку. Но ёжик не уходил, норовя меня укусить.

Я выбрал момент, спрыгнул с коробки и, прихватив её с собой, побежал налево. Здесь можно передохнуть. Около стены я увидел паука, болтавшегося на веревке. Подойдя поближе, я заметил под ним надувной круг. А вдруг на пути попадет река? Тут-то круг мне и поможет. Я проскочил под паука, схватил круг и быстрее обратно. Чуть не попался пауку на обед.

С кругом и коробкой я пошел направо. На дороге я увидел валяющийся пистолет. Логически поразмыслив, я подумал, что пистолет мне вроде бы ни к чему и двинулся дальше. За пистолетом лежал штопор. Его-то я захвачу. А вдруг мне придется открыть какую-нибудь бутылку. Чуть дальше лежал такой же конверт, как и под кокосом. Что же там? Это же одна из частей моего пропуска! Но медлить нельзя, так как неподалеку лазил ещё один ёжик.

А потом началось такое, что мурашки по всему телу бегают.

Начну с того, что карта показывала тупик. Когда на самом деле здесь была яма. И непонятно, что в яме: ловушка или простой проход. Для начала я просто прошел яму. За ней летала водяная помпа. Поменяв её на коробку, я возвратился к яме, рискнул и прыгнул. Вот повезло. Здесь нет никакой ловушки, и даже гамбургер кто-то оставил. Рядом валялась каска.

Пройдя налево, я увидел подземное озеро... с попугаями. Я знал, что плавать можно, только имея при себе спасательный круг и помпу. Б-р-р-р! Какая холодная вода. Осторожно! Попугай-то, оказывается, кидается яйцами. Еще мгновение и одно из них попало бы в меня. Хорошо, что я успел нырнуть в воду. Маневрируя между яйцами, я доплыл до другого берега.

Но, как только я вышел из пещеры, меня сразу атаковали червяки и пчелы. Еле увернулся от них! На дороге я увидел ягоду. Страсть как обожаю! Перепрыгнув через яму и зайдя в дом, я увидел молоток. Но брать его не стал и, выйдя через другую дверь, пошел дальше.

Перепрыгивая через ямы и съедая по дороге продукты, я добрался до другого дома, около которого лежала монета, и бегал ежик. Зайдя в дом, я увидел абажур и бутылку. Проголодавшись, я подбежал к бутылке, но она была закрыта. К счастью, у меня был штопор. Я открыл бутылку, но вместо воды в ней оказался какой-то листок бумаги... Это была вторая часть от моего пропуска. Выйдя из дома через вторую дверь, я отправился дальше. Пройдя мимо бриллианта, я увидел яблоко. Уж с ним-то я знал, как обойтись. Дальше лежала ещё одна часть моего пропуска, но она охранялась... электричеством. Ох уж мне эта цивилизация! Но раз есть электричество, то должна быть и кнопка, чтобы его отключить.

Возвратившись в первую хижину, я взял молоток, вышел через левую дверь и прыгнул в первую яму. Удача улыбнулась мне. Я попал в тупик, где оставил коробку. Молотком я быстро разбил её и достал ключ. Теперь нужно найти дверь, чтобы открыть её этим ключом. Я видел дверь в первой хижине. Надо её проверить. Снова проплыв через озеро, я оказался в первом доме. И верно: ключ подошел. Я вышел из дома. Но где я? Да это же место старта! Теперь появилась возможность достать ту часть пропуска, которая лежит под кокосом. Но что мне надеть на голову, чтобы кокос не зашиб. Ах да! Каску, которая лежит около озера! Вернулся за каской и прошел с ней к третьему конверту. Ничего со мной не случилось. Кокос упал, но я был в каске. Сняв каску и захватив кокос, я пошел обратно. Но снова, замечтавшись, я прошел мимо дома и, устав, на что-то присел. Это "что-то" опустилось вниз. Ба! Да это же та самая кнопка!

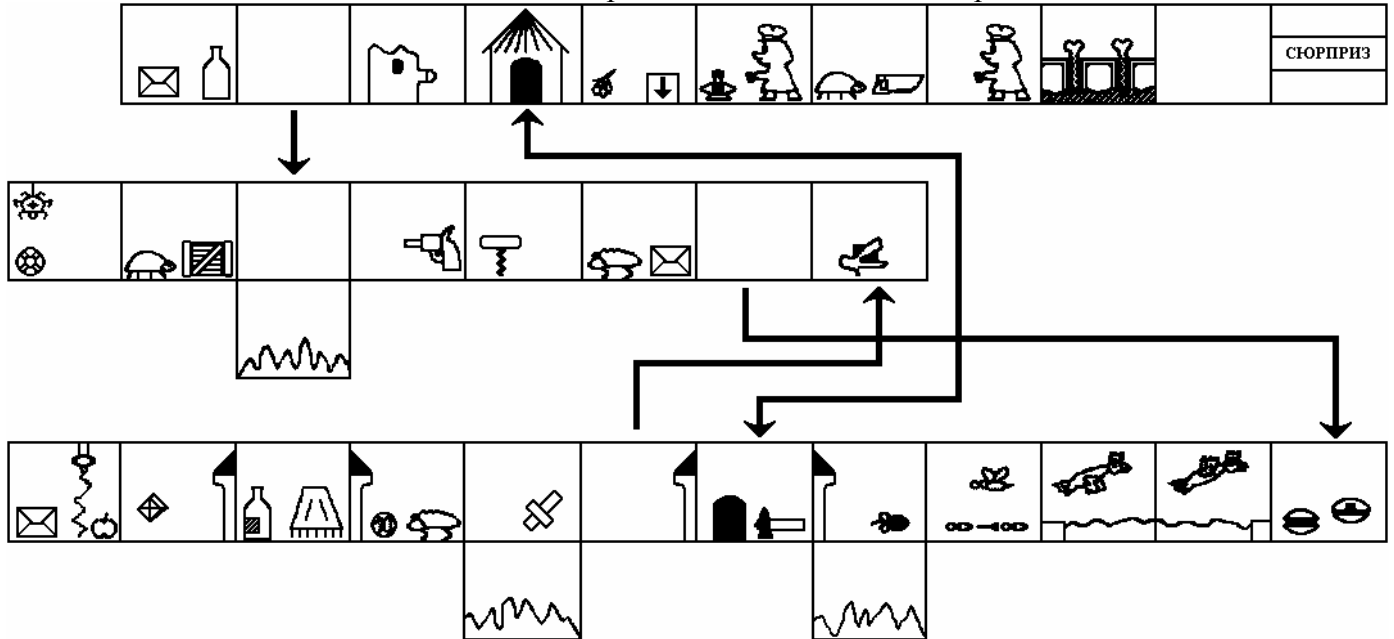
Но, как только я с неё слез, она возвратилась в первоначальное положение. Как же её зафиксировать? И тут меня осенило. Я решил положить на кнопку кокос. Кнопка осталась нажатой. Я возвратился за последней частью пропуска, снова вернулся к первому охраннику, показал ему пропуск и он меня пропустил.

Но за первым охранником стоял ещё один. Второй охранник требовал плату за вход. Но где мне взять деньги? На дороге я подобрал ножовку и, когда увидел кокос, то вспомнил, что давно не

КОМПЬЮТЕРНАЯ НОВЕЛЛА

ел. Распилив кокос, я удалил свой голод и вдруг заметил, что внутри лежит золотая монета. Я бегом помчался к охраннику и отдал ему эту монету, после чего он пропустил меня. За охранником текла река, в которой водились змеи. К счастью, над рекой был мост. Я быстро побежал по мосту к своему дому.

Дорогие читатели! Специально для Вас дракончик Пафф дорисовал карту и отметил на ней местоположение всех вещей и животных. Приятного Вам отдыха за игрой LITTLE PUFF.



POKES

Предлагаем огромный коллективный труд наших читателей со всех уголков страны. Наиболее полную коллекцию POKES прислал наш читатель Царев А. В. из Нижнего Новгорода. Большая часть присланных им POKES содержит комментарии, на какой параметр игры оказывает влияние конкретный РОКЕ или их комбинация - энергия, время и т.д. Такими же подробными комментариями к своим солидным трудам снабжены письма Алексея Татаренко из Красноярска. Он регулярно присылает нам письма с такими POKES, которые мы нигде раньше не встречали. Видимо, Алексей активно занимается этим делом. То же можно сказать и про письма Андрея Степанова из г. Новочебоксарска, он имеет за спиной полтора года взламывания программ и охотно делится со всеми читателями POKES, найденными самостоятельно. Виталий Казаков из г. Армавир прислал письмо с POKES, найденными им вместе со своим другом по "Спектрumu" **Дмитрием Денисенко**. Объемные коллекции POKES прислали Артур Артамонов из г. Червоноград Львовской области, Патрахин В.В. из Оленегорска, Быков А.В. из Владивостока. Но особенно приятны нам письма, в которых читатели делятся своими несколькими первыми (но, конечно же, не последними) POKES, найденными самостоятельно после того, как они начали активно изучать машинный код. Большинство из них пишут, что им очень помогли в работе наши книги по программированию в машинных кодах. Это Селифонтов Михаил из г. Углич Ярославской обл., Степанищев Е.В. из Лениногорска, Андрей Казаченко из Владивостока, Савосин А.В. из Подмосковного г. Мытищи, Вячеслав Солдатов из г. Краснореченск Приморского края, Сукачев С. из г. Златоуст, Николай Губин из Новгорода, Денисов Алексей из Пензы, Павел Перин из г. Первоуральск Свердловской обл., Игорь Комаров из Иркутска, Кирилл Мухин из г. Северодвинск Архангельской обл., Жуковец М.В. из Томска, Дмитрий Пьянков из Горно-Алтайска, Алексей Тальников из Якутска, N.A.A. из Томска, неизвестный автор (письмо не подписано) из г. Чебоксары, Андрей Бастраков из Челябинска, Андрей Рубин из г. Энгельс Саратовской обл., Сергей Беляков из г. Дзержинска Нижегородской обл. Спасибо Вам, друзья.

Наши читатели А. Щегольков, Н. Шмаков, И. Бычков и А. Кицен (возможно, ошибка - фамилия неразборчива) из Усть-Илимска кроме присланных POKES делятся своим методом поиска.

КОРР: Мы, в основном, ищем их не по количеству жизней, а по надписи, свидетельствующей об окончании игры: "GAME OVER", "TIME OVER", "MISSION FAILED" и т.д. После нахождения в памяти соответствующего текста и адреса процедуры, вызывающей печать этого текста, выявляем условие печати. Для поиска пары байт пользуемся программой, которую мы назвали POISK CODE - она сделана в кодах и выполняется гораздо быстрее, чем её аналог на Бейсике. Наш вариант программы располагается с адреса 23296 и занимает 42 байта. Для его ввода служит программа:

```
1 FOR G=23296 TO 23337 : READ A : POKE G,A : NEXT A
2 DATA 62, 2, 205, 1, 22, 1, 180, 95, 205, 17, 91, 3, 120, 177, 32, 248,
   201, 10, 254, 0, 192, 3, 10, 11, 254, 0, 192, 197, 2 05, 43, 45, 2 05,
   227, 45, 193, 62, 6, 215, 62, 6, 215, 201
```

Меняя содержимое ячеек 23315 и 23321 и запуская программу командой RANDOMIZE USR 23296, Вы сможете отыскать необходимые Вам два следующих друг за другом байта. Программа распечатает адреса, в которых они встречаются.

ИФК: Переходим теперь непосредственно к обобщенной информации, присланной нашими читателями. В длинный список POKES, приведенный ниже, не вошли те данные, которые уже были опубликованы нами в ZX-РЕВЮ в 1991-1994 г., однако, если читателями были присланы другие данные, то они здесь приведены.

Знак "/" является разделителем между POKES, полученными из разных источников. Это могут быть значения, предложенные разными авторами или же для разных версий игры.

В скобках даны возможные иные значения (в разных источниках содержатся разные цифры).

Если, к примеру, Вы видите запись: 47007,255, то, скорее всего, это означает, что в указанной ячейке хранится начальное значение жизней, и можно задавать их не обязательно 255, а

столько, сколько Вы считаете для себя достаточным (47007,N). Значение 255, как Вы понимаете, это "псевдо"-вечная жизнь, но, как правило, такого большого числа жизней оказывается достаточно для одного сеанса игры.

Наименование игры	POKES
1942	47007,255 (47001,255) 65368,198 65369,N-этаж 65370,50 65372,182 65373,201 52377,205 52378,88 52379,255 RANDOMIZE USR 24833
1985	49303,0
3D SPACEWARS	26244,0 43364,182 46507,182 49130,182
3D TUNNEL	29711,0
3DC	34298,201-защита 34036,0-прыжки 26955,0-кислород
720	40774,0-жизни 40360,0-деньги 37357,0-билеты
A WHOLE NEW BALL GAME	50056,182-жизни 51485,0-время
ABU SIMBEL	49290,N-жизни 47684,0 42169,0 42170,0 42171,0
ACRO JET	25148,10
ACTION FIGHTER	53196,0-жизни 38178,0-время
ACTION FORCE 2	47874,182 51904,55 51450,36 50262,N-попытки 50145,36-энергия 51450,201-энергия 51730,182-время 51905,182-жизни 59021,201- тюрьма не взрывается 51904,0-жизни 51454,182-энергия 51734,201-время на TAWK
ACTION REFLEX	23988,54 23349,201 RANDOMIZE USR 23935 50770,0 50771,0 50772,0 50964,0 50965,0 50966,0 RANDOMIZE USR 50000
AD ASTRA	28591,0 28592,0 28593,0 35853,0
ADVANCED PINBALL	35237,0
AFTER THE WAR 1	48949,0-жизни 52367,0-энергия 46962,N-жизни 48950,195 48951,217 48952,190
AFTER THE WAR 2	49000,0-жизни 56059,0-энергия 48831,N-жизни
AGO RAI0	34808,24-неуязвимость 34813,0-жизни 35284,22 35285,136 Примечание: Если попадете в безвыходное положение, нажмите BREAK
AH DIDDUMS	24920,255 24786,0
AIR FORCE 2	51904,0
AIR WOLF 2	50262,N 53471,0 53038,201 - неуязвимость 53474,0 53433,201
ALCHEMIST	31055,0 47340,0 47414,0 49745,195
ALIEN HIGHWAY	39443,0 39142,0 (39412,0) 35125,0
ALIEN SYNDROME	47503,182-жизни 42557,0-время
ALIEN-8	43735,201 43255,201 44460,201 44526,0 51736,0 45121,N (45121,1) 51736,0-жизни 49864,0-прыжки 44461,97 44462,185 51736,0 RANDOMIZE USR 25344 44256,0 51736,0
ALIENS	31014,0-оружие 31834,0-энергия 30738,201 (30738,0)-живучесть 25195-вход в двери 34484,195
AMAUROTE	37311,0 37316,0 37321,0 40615,0 46312,0 46381,201
ANARHY	42405,255
ANDROID	52250,32 55249,24 53897,0 33702,0 25280,0 25312,0 46610,24
ANDROID 2	52262,0 (52252,0) 53894,0 52249,24 52250,62 53897,0

Наименование игры	POKES
ANTIRIAD	57501,0 57502,0 57503,0 54639,0 54640,0 54641,0
AQUAPLANE	25448,0 25449,195
AQUARIUS	31054,0 31055,0
ARCTIC FOX	50242,0 49396,0
ARKANOID	33702,0 23822,255 33842,0 37120,0 39410,1 35665,0 (35665,9)
ARKANOID 2	40413,1-два мяча 40413,4-бита 40414,1-следующий экр. 40413,32-лазер 40415,64-тяжелый шар 33423,0 33429,0-нет перемещения кирпичей
ARKANOID 2 (128K)	37586,0
ARKOS 1	52117,0 52118,0 52119,0
ARKOS 2	52126,0 52127,0 52128,0
ARMY MOVES	54597,0
ASTEMEX	43516,0 43517,52 43517,191 43518,194 33746,N-жизни 43518,34 43519,0 43520,0
ASTERIX & MAGIC CAULDRON	36276,0 37836,0 37837,5 37838,0
ASTRO M.C.	26799,0 39421,0-жизни 45087,201 - время 28820,201 29646,201-ловушки
ASTRO M.C. 2	40866,182-жизни 32452,201 - энергия 45070,0-время 32161,0-бомбы 29220,24-огонь 29312,24-монстры
ATIC ATTACK	36519,0 36518,129 36353,0 37260,175
ATV SIMULATOR	60250,0-горючее 57318,201-время
AUF WIEDERSEHEN MONTY	42160,201 37002,0 (37222,0) 32768,0 41128,0 41129,0
AUTCARS	61320,0
AUTOMANIA	64589,183
AVALON	23782,2 23786,201 23878,204 23879,227 58316,201 RANDOMIZE USR 62283
AVENGER	27453,0 55519,0 51934,201 51527,0 51528,0 51529,0 RANDOMIZE USR 18434
B. DASH C.K.	25371,255
B.C.BILL	47589,201 RANDOMIZE USR 25856
BACK TO SKOOL	29131,80 29135,45 32748,1
BARBARIAN (Palace)	51005,255 27580,0
BARBARIAN 2	27680,0
BARBARIAN 3	37132,0 27533,255 (перед первым USR)
BATMAN	54067,0 54832,201 54708,0 54719,0 37512,166 36798,0 39908,201 37430,0 37521,62 37522,12 RANDOMIZE USR 25984
BATMAN 2	46520,119 46526,24
BATMAN THE MOVIE	26043,0-враги не ходят по лестницам 26056,0-быстрое раскачивание на веревке 34368,0-энергия 24887,105 24973,182
BATTY	48437,183
BEDLAM	58691,0 (58601,0) 59430,0 59783,0 59972,0 60069,0
BEYOND THE ICE PALACE	38274,182
BIG BEN STRIKES AGAIN	43120,0-жизни 43105,201-неуязвимость
BIONIC COMMANDO	34690,0 34247,0 34274,0
BLACK LAMP	33609,127 34487,127
BLACK MAGIC	24730,0

Наименование игры	POKES
BLACK RIDER	28924,0 28928,0 28929,0
BLADE WARRIOR	37161,0 39490,0 39490,60 39490,182 34864,201-неуязвимость
BLAZING THUNDER	35054,0-жизни 36539,0-бомбы 34484,0 36514,0 37334,0-энергия 34701,N-жизни
BLIND ALLEY	25284,0
BLUE MAX	43983,195 43984,163 43985,167
BLUE THUNDER	48552,0
BMX KIDZ	52108,0
BOBBY BEARING	28094,36 29750,182 - время 32173,N-жизни 29688,175
BOBSLEIGH	43589,0 43549,0 43614,0-время
BOING	36610,0
BOMB FUSION	35460,0-жизни 35132,0-время 35567,0-число мячей не увеличивается 32869,N-число мячей 35141,183 35253,183
BOMBSCARE	56256,0 23606,46 23607,181
BOOTY	58294,0 RANDOMIZE USR 52500
BOR FIES	43393,0
BOULDER 3	26012,0 26013,0 26030,0 31480,0 31481,0 31482,0-бессмертие 26011,0 - время 31007,0 31008,0 31009,0
BOUNDER	36610,0
BRAIN ACHE	28064,0
BRAT ATTACK	52845,0
BRAVE STAR	51803,24 51804,18-время
BREAK NECK	46090,0 -жизни 46013,N -номер комнаты (1-200) 51473,0
BRIDE OF FRANKENSTEIN	40476,201-энергия 37605,201-нет монстров 35486,201-эликсир жизни
BRUCE LEE	51795,0 48996,57 48997,57
BUBBLE BUSTER	32807,0 32967,0
BUG EYES	43393,0 36003,201 RANDOMIZE USR 36000 43993,0 RANDOMIZE USR 42200
BUMPY	47771,0-жизни 47386,0-молотки 51717,0 52296,0 52807,0-капли 51576,0 52155,0-ключи
C.O.R.E.	31079,175-бессмертие 40363,201-нет врагов
CALL ME PSYCHO	48050,0
CAMELOT WARRIOR	53929,0 55918,201
CAPITAN SEVILLA	40088,201-энергия
CAPITAN SEVILLA (1-2)	40204,0-бессмертие 40183,0 40139,0 40140,0
CAPITAN TRUENO 1	25100,N
CAPITAN TRUENO 2	24785,0-жизнь 24984,N-энергия 28707,0 28708,24-энергия
CASANOVA	53153,182-жизни 52918,182 - оружие 53152,24 53153,4
CAULDRON	40060,0 40061,0 60060,0
CHASE H.Q.	45221,183 34755,0
CHEEKAN'S EXPLOITS	28418,0
CHEVY CHASE	41764,0-время 41787,0-время
CHIER	34025,0
CHIMERA	23794,0 23799,0 23804,0 23805,0 23810,0 23815,0
COBRA	36515,183 43647,255 36518,0 36491,24
COLISEUM	48875,0 48876,0 48877,0
COMBAT ZONE	51219,0-топливо 52803,0-энергия 62107,183-жизни

Наименование игры	POKES
COMMANDO TRACER	50613,0-энергия 56255,195-проверка времени 56240,0 56241,0 56242,0-жизни и энергия
CON-QUEST	38003,201 23225,201
CONQUESTADOR	59490,0-неуязвимость 62370,0-нет монстров
COOKIE	35730,52 28697,0 28698,0
COP OUT	32817,255 44929,0
CORSARIOS	29225,175 27500,183
CORSARIOS 1	29227,183
CORSARIOS 2	27500,183
COSMIC CRUISER	28523,0 28560,0 28567,0 25373,0
CRAZY CARS	29403,0-время
CROSSWIZE	32955,255 33370,N-жизни 33873,0 33874,0-энергия 51617,0-защита 33939,12-бессмертие
CRYSTAL CASTLES	63733,182 63733,134
CURSE OF SHERWOOD	63033,0 64013,0 (64613,0)
CYBERNOID	39402,0 39403,0 37402,0 24917,255 36687,0 (36678,0) 31818,0 28081,255 36678,0 31818,0 39422,0 39423,0
CYBERNOID 1	28042,195-включение музыки для AY 28042,0 28043,0 28044,0- включение обычной музыки
CYBERNOID 2	32202,0 36197,0 34769,0 25396,255 36198,0 36216,0 36217,0 36218,0
CYCLONE	37536,0 16547,24 38291,50 63892,160 63893,146
DAN DARE	36168,175 45954,104 46888,201 43256,0 46887,0 44413,201 47710,201
DAN DARE 2	53822,255 45891,0 60605,52 54390,0 56461,195 56462,101 56463,220 RANDOMIZE USR 52390
DANDY	49661,36 62004,201 RANDOMIZE USR 24100
DANGEROUS GARDENS	30361,201
DARK FUSION	50172,N 50407,0 50408,0
DEATH STAR	36682,0 38351,0 36714,0 38348,0 - жизни 36679,0 38170,0-энергия
DEATH WISH 3	38678,183 39353,183 43301,183 (43301,201)
DEFLECTOR	34473,8 41784,0 42707,201 (42207,201)
DEJA VU	45703,0-энергия 50296,0-жизни 47234,0-оружие 43293,0 48840,0- время
DEMON SLAYER	31679,0-время субигры 53425,0-оружие 55476,0-время 55571,195- энергия
DEMON'S REVENGE	30699,17-энергия
DENIZEN	58594,0 56506,205-факел 61226,0 - пули 56355,0 60614,0 64053,0 - энергия (60882,0) 60822,0 60823,0 60824,0 61107,0 61108,0 61109,0-двери
DEVIANTS	47639,0-энергия 35272,0 35286,0 - оружие 26172,201-мгновенная установка бомбы
DIE ALIEN SLIME	32855,24 33227,195
DIGGER DAN	25559,0 26363,0

Наименование игры	POKES
DIXY 2	25022,N-попытки 26894,0-энергия, также увеличивается скорость передвижения шарика
DIZZY 3	42481,N-жизни 63001,0 30273,182 63001,61 41218,N-жизни
DIZZY 3,5	32880,182 32880,0 32881,195
DIZZY 4	63001,183 (63001,0) 29623,0 29624,195 28594,N-жизни
DIZZY 5	51291,0 41815,0 52130,24-энергия 48955,0-звезды
DIZZY DOWN THE RAPIDS	31072,N 31116,255
DOMINATOR	27844,0 27845,0
DONKEY KONG	35370,0 33875,255
DOUBLE DRAGON	37693,0 37815,0 37813,0 37794,0
DOUBLE TAKE	39475,201 40243,201
DOWN TO EARTH DRACONUS	40141,182-жизни 39646,0 39815,0 65486,0-время 64215,0 64215,128 62866,208 (62866,0) 55810,255
DRAGON BREED	36364,175 36365,0-неуязвимость 38047,24-любой уровень 37662,0 37666,24 36474,0-кредиты 36395,0-жизни 36364,24 36380,201-неуязвимость
DRAGON NINJA	41770,0 43455,8 38918,0 38684,1
DRAGON TORC	58309,201 RANDOMIZE USR 24700
DRUID	24890,201 30039,0 31326,0 32808,0 - энергия 34082,183-боеприпасы
DUKES OF HAZARD	44246,0
DUN DARACH	34999,255 34378,24 34032,24 43333,201 RANDOMIZE USR 24100
DUSTIN	52216,195 52932,0 52937,0-время 52939,0-нет наказаний 52904,0-защита
DYNAMITE DAN	29002,182 52678,0 57035,0 58770,201 59093,201
DYNAMITE DUX	44277,0 44401,0
DYNATRON MISSION	45116,0-жизни 45112,0-время
EARTHLIGHT	50062,0-жизнь 51284,0-горючее 53153,0-вооружение
EDD DACK	24854,0
EDD DUCK	32956,0
EDD THE DUCK	33681,0-жизни 34944,N-количество звезд
EGG HEAD	56648,0 55774,N RANDOMIZE USR 57787 позволяет войти в редактор и выбрать начальную комнату и бесконечные жизни
ELITE	53775,0 56993,0-повреждение оборудования и потеря топлива
ELITE (JOYSTICK CLUB)	39517,0-энергия 28820,0-энергетическая бомба 39959,0-ракеты 39987,0-спасательная капсула 56997,0-топливо 46759,0-нет нагрева лазера 46848,0-уничтожение врага с первого выстрела
ELITE (M128)	56280,0 56290,0-гиперпереход в любое место на карте 54887,0-топливо 42381,201-энергия 42870,0-ракеты 42921,0-спасательная капсула 37485,0-энергетическая бомба 54829,0-межгалактический двигатель
ELVEN WARRIOR	38094,0-ключи 38710,0-стрелы
EMPIRE STRIKES BACK	43624,0-защита
ENIGMA FORCE	46807,0-пули неэффективны, а лазер робота - вечный
EQUINOX	31958,0-время

Наименование игры	POKES
ETHNIPOD	61196,0-1 игрок 61456,0-2 игрок
EVERYONE'S A WALLY	61193,0 61453,0-бессмертие 56988,0 57286,0-бомбы 28215,0
EVERYONES A WALLY	58217,0 58218,0 58219,0 RANDOMIZE USR 33156 46191,148
EXPLO 31	58153,201 58180,201-жизнь
EXPRESS RAIDER	32054,0 32058,0 32179,0 32183,0
F1-TORNADO	60503,0-сила 60514,0-жизнь 61100,0-время
FALL GUY	60873,0
FALLEN ANGEL	27235,0
FANTASTIC VOYAGE	44204,0
FARENHEIT-3000	25234,0-жизни 25718,0-время 28139,0-энергия
FAST FOOD	54277,0 (54227,0) 54492,0
FIFTH QUADRANT	30818,0 RANDOMIZE USR 29924
FIGHTER WARRIOR	47844,0 28494,0
FIRE FLY	40712,0-оружие 49664,201-нет врагов 60991,0
FIRELORD	37258,0-энергия 43351,195-телепортация 42877,24-быстрый вход на станцию 34509,0
FIRETRAP	39974,0 39975,195 (29975,0)
FLASH GORDON	46509,195-жизни 47629,0-время 45654,195 45655,122-защита
FLUNKY	49123,0-пули 50732,0-время
FLYING SHARK	35320,183
FOOTBALLER OF THE YEAR	54462,201 54379,3 42464,255
FOXX FIGHT BACK	43007,182-бомбы 48631,182 - жизни
FRANK'N'STEIN	43006,62 43007,2 43008,167 (43006,0 43007,0 43008,0) 48605,0
FRED	43606,0 43607,0 48630,0 43631,0 43632,0 48980,0 48982,0 48983,0
FREEZ BEES	50253,0
FROST BYTE	46535,0-жизни 48071,0 (48061,0)-энергия 28287,N
FULL THROTTLE	28277,0 28278,0 34124,0 33723,0
FULL THROTTLE 2	31171,0-энергия 37729,0-патроны RANDOMIZE USR 30285
FUTURE KNIGHT	34619,0 (34610,0)
FUTURE PROJECT	36331,24 37117,0-неуязвимость
GALAXIANS	30992,191 30993,218 -жизни 28237,191 28238,218 -кислород
GALVAN	46608,0 46609,0 46610,0
GAMES DESIGNER	54476,24-нет дисквалификации
GAUNTLET	31683,0 (31623,0)
GHOST BUSTERS 2 (LEVEL 1)	29332,0-исчезают все движущиеся препятствия
GHOST'N GOBLINS	59354,7
GHOULS'N GHOST	23290,0 23414,0
GIANT'S REVENGE	36919,0
GILLIGAN'S GOLD	44887,24-нет генерации новых монстров 48489,0 48497,0-энергия 36919,1 - время 44566,1-ключи 38050,0 38052,0 - супероружие 43414,0-амулет 43617,24 - лекарства 37492,24-стрельба сквозь стены 38769,62 38770,192-старт с места гибели 49815,0-энергия 50006,0-верёвка 49494,0 49624,0 49712,0- боеприпасы 25080,201-без этого на некоторых компьютерах после смерти игра зависает
	39857,135 39858,50 39859,180 36057,36 33433,0 36083,0 39860,191 39857,135 39858,50 39859,180 39860,191 33352,201 36083,0
	32827,255
	24504,8
	52881,0 52882,0 52883,0 52896,0 38935,18 38941,18 38939,0
	52353,0 52354,0 52436,0 52437,0 55219,0
	52786,0 38939,0 38935,18 38941,18 RANDOMIZE USR 53379

Наименование игры	POKES
GLIDER RIDER	34973,0-неуязвимость 34829,0-замедление времени 34818,0-остановка времени 34931,0-бомбы 37439,0 37440,0-выключить лазер 37461,0 37462,0 37463,0-ровный полёт
GOLDEN AXE	33332,N
GONZZALEZZ 1	46652,0-кредиты 43010,0-жизни 37085,0 31498,0 31501,0
GONZZALEZZ 2	37085,183 31890,1 35749,0-жизни 48058,0-оружие 35749,183
GOODY	45601,0 45602,0 45603,0
GOONIES	33400,183 33409,0 33247,N
GOTCHA	55926,0
GOTHIK	43934,58 42110,58
GRAND PRIX 2	43074,201-время
GREAT ESCAPE	41953,183 51243,201 47044,0 47045,0
GREEN BERET	42076,0 46828,0 46829,0 49830,0 40074,0 40075,0 40076,0 40077,0
GREGORY	62675,0-время
GREMLINS 2	35622,0
GROUND ATTACK	24872,0 (27872,0) 29063,0
GUN FRIGHT	49233,54 49234,1 49235,0 49236,0 45832,183 47952,183 48032,183 49233,183 23446,201 23446,33 42355,0 42882,3 46344,0 48464,0 48544,0 49745,0
GUTZ	47389,0 38915,62
GYROSCOPE	53887,201 59149,0 54754,200
H.A.T.E.	53247,183
HADES NEBULA	49883,0
HALLS OF THINGS	32717,0 35923,255 RANDOMIZE USR 24576
HARD DRIVING	41927,0-время 42458,N-инерция полета; начальное значение =2, но задав 0, можно упростить игру: машина не разбивается, падая с большой высоты 42240,N-езда по ухабам при съезде с дороги начальное значение=0, но задав 1...35, можно усложнить игру
HAWK STORM	32148,24 35300,201 36604,0 37804,0-энергия 34869,0-боеприпасы
HEAD OVER HEELS	41841,0 41842,0 41843,34 41844,25 41848,33 41851,33 42195,0 (42185,0) 43132,0
HEARTLAND	23563,201 41280,0
HEAVY ON THE MAGIC	33102,0 33103,0 33201,48 (33201,0) 33130,201 33189,201 33240,201 40207,24
HELL FIRE	31440,0 35088,0 40084,0 44636,0-жизни
HELTER SKELTER	36955,0-жизни 39565,201-время
HERBERT'S DUMMY	51925,48
RUN	
HIGHWAY ENCOUNTER	40905,0 40772,195 40773,123 40774,0
HONG KONG PHOOEY	33128,0
HORACE GOES SKIING	29270,0 30027,0 30644,0
HUMAN KILLING	35068,0 35069,0 35070,0
MACHINE	38748,183 40519,183
HUNCHBACK	24760,55 26888,0
HUNGRY HORACE	26426,0 (26429,0)
HYDROFOOL	64068,185 64069,248 64070,255
HYSTERIA	44588,201 44527,201

Наименование игры	POKES
I'BALL	49165,0 49483,0 43612,0 43619,0 49165,0-жизни 49000,201-неуязвимость
I'BALL 2	43384,0 49483,0 49165,0 49483,0 45392,0-жизни 43611,175-неуязвимость
ICE TEMPLE	63132,0 63133,0
IMPOSSA MOLE	52919,0 52920,0
IMPOSSABALL	37539,0 37706,0 41185,0 (41158,0)
INDI JONES (LAST CRUSADE)	33310,N
INSIDE OUTING	54216,201-энергия 52733,201-враги не двигаются
INTERNATIONAL KARATE	31200,0
INTO THE EAGLE'S NEST	41136,0-бессмертие 36640,0-патроны 36641,0 40512,0-ключи 40513,0
IRON SOLDIER	55250,202
JACK & THE BEANSTALK	56115,0 56116,0 56388,62 56389,27 56390,0 42404,255
JACK THE NIPPER	44278,58 44285,58 43519,201
JACK THE NIPPER 2	34232,56 RANDOMIZE USR 54476 34631,0 43251,0 23739,244 RANDOMIZE USR 34240 44278,58 44285,58 RANDOMIZE USR 24833
JACKSON CITY	34420,0 -жизни 34501,0
JAMPING JACK	30094,182
JET PAC	36966,24 25018,0
JET SET WILLY	35899,0 (39899,0) 56389,27 56390,0 36353,44-возможность лезть по левой стене 34795,N (0<N<60)- старт из комнаты N 36477,1-упав, не разбиваешься 34275,10-ключ к другой комнате: нажмите 9 и другую цифру-телепортация 36358,0-высокие прыжки 35123,0-нет движущихся врагов 59900,255- преодоление карнизов RANDOMIZE USR 33792 41983,255-возьмите краник от ванной и подойдите к кровати - увидите, что будет...
JET SET WILLY 2	35899,0 34497,202 34498,135
JETMAN	30019,255 30027,32 34686,150 36965,0 36966,224 36945,3
JOE BLADE	65029,50 65032,50 34802,0 39084,0 39133,0-ключи 38486,201-время
JOE BLADE 1	34802,0 39084,0 39133,0-ключи 38486,201-время
JOE BLADE 2	52083,201-время 58304,0 58890,0 59378,0 60386,0 -время субигры 58108,20 58109,225-выход из подигры 59116,20 59117,225-время
JOE BLADE 3	42723,201-жизни 40547,0-оружие 45336,0 -время 45364,0-время субигры 41486,0-никого нет 40549,24-патроны 43059,195-энергия 42830,24 42831,3 -против мин
JOE BLADE 4	40642,0 -жизни 38019,0 -оружие 41624,0 -время 41640,201-время субигры 39559,0 39560,0 39561,0 39562,0-никого нет 41587,201-время 41647,0 41648,0-время субигры
JUDGE DREDD	24936,24 (24936,255)
JUMPING JACK	30094,182
JUNGLE WARRIORS	30125,0 30093,0 30094,0 30095,0 30096,0 44971,0 44637,201-энергия
KARNOV	32972,0 25620,0 32855,255
KENDO WARRIOR	38342,182 38342,0

Наименование игры	POKES
KGB SUPER SPY	41352,0-жизни 42016,0 42022,24 36841,201-энергия
KINETIK	61998,0
KING KONG 2	42523,0 43100,0 44111,0
KIREL	35932,0 59322,154
KNIGHT LORE	34961,0-жизни 40563,183-время
KNIGHT TYME	51661,0-прыжки
KOKOTONI WILF	24584,255 (24548,255) 24585,255 45322,255 45323,255 41456,0 41457,0
KOSMIC KANGAROO	42214,255
KWICK SNAX	43742,0
LAP OF THE GODS	28929,9 28934,8 28939,8 RANDOMIZE USR 41712
LAST NINJA	36212,0 (36212,3)
LAST NINJA 2	40905,N
LAZY JONES	46563,0 46564,0
LEGEND OF KAGE	47039,201 RANDOMIZE USR 47000
LEGEND OF THE AMAZON WOMAN	53790,201 RANDOMIZE USR 57680
LEVEL 5	35993,198 36576,198 36751,198
LICENCE TO KILL	36575,0
LITTLE PUFF	36578,175 36579,60 41855,0
LIVE & LET DIE	36578,0 36579,175 (36579,0)
LIVINGSTONE	29966,N
LIVINGSTONE 2	40777,0
LOCOMOTION	56693,0 (56693,N)
LORD CROM	37065,0 63427,240 63428,210 54000,62 54001,0 54002,50 54003,57
LUNAR JETMAN	54004,144 54005,195 54006,33 54007,249 RANDOMIZE USR 18434
MAD MIX	57960,0
MAGIC CARPET	57590,183
MAGLAXIANS	45123,201-энергия
MANIC MINER	48180,183 48432,183 48640,183 59210,183 59321,183 55816,0-жизни
MARAUDER	53023,0 53024,0 53025,0-неуязвимость
MARBLE MADNESS	27599,0
MEGANOVA	24391,0
MERLIN	24391,0
MERMAID MADNESS	46331,183
METAL ARMY	51948,0
MIDWAY 1943	50416,0 55543,0
MIG-29	23439,201 36963,0
MILK RACE	36966,224 36945,3
MISSION JUPITER	36965,0 37999,201
MONTY MOLE	40296,0
MONTY ON THE RUN	39947,N
	29530,0
	59354,7
	33885,7-уровень 34269,N-жизни 35136,0-жизни
	36160,0
	34795,0 34796,0 34797,0 34798,0 34799,0 34800,0
	34231,182 26178,255
	35160,175 35161,50 35162,95 35163,250-жизнь 34231,0-бомбы
	26178,N-жизни (N<251) 26183,N-гранаты (N<251)
	39579,0 (38579,0)
	32382,0
	36599,1 36561,1
	31333,201-жизни 31217,201-время
	35722,255
	37061,0 37062,195-бессмертие 42813,0-патроны
	44664,0
	42965,183
	30531,201 35482,201-время 32186,201 31105,0-энергия
	60468,0
	38003,0 (38004,0)
	34714,0
	39039,N-жизни

Наименование игры	POKES
MOON ALERT	42404,255 39754,0 42654,195 37035,201
MOONLIGHT MADNESS	59945,N 57747,0 57145,167 57833,0 57834,0 57835,0
MOTOCROSS	27663,0 29661,0 44142,0 44287,0
MOTOS	42241,0
MR.WEENIS & SHE VAMPS	30742,2 30973,0-лекарства 40019,0 39974,0 41228,0-энергия
MR. WIMPY	43012,0 42908,0 27504,62 27505,10 27506,50 27507,226 27508,105 27510,0 27511,224 27512,105 33693,0 33501,0
MR. WONG	25003,0 25985,0
MUTANT MONTY	54933,0 55761,60 56483,183 54959,0
MUTANTS	29667,0
MYTH	1 уровень 32468,0 2 уровень 32409,0 3 уровень 32441,0 4 уровень 32526,0 5 уровень 32349,0
N.O.M.A.D.	40167,0 40703,0 40249,N
NAVY MOVES	49962,0
NAVY MOVES 2	54047,0 55805,0 55837,0
NETHER EARTH	44399,18 44615,33 51841,0 51842,0 51843,0 51844,0 RANDOMIZE USR 42496
NETHERWORLD	32577,0
NIGHT GUNNER	24763,182
NIGHT SHADE	57499,0 53442,0 53443,12
NINJA WARRIORS	42929,201-время 29666,0 29721,0 30398,0 30419,0-энергия
NODES OF YESOD	32661,0
NONAMED	36870,0 36871,0 36872,0-нет монстров 33715,0
NONTERRAQUEOUS	30596,24 36047,195 36222,0 36464,24 25699,24 25646,195
NOSFERATU THE VAMPIRE	32499,0 39791,0 (39791,201) 36792,201
OBLITERATOR	33595,0 33596,0 33595,182-энергия 33546,201-неуязвимость 34172,0-оружие
OBLITERATOR	32948,182 26802,255
OCTAN	58309,0 58289,183-жизни 63003,0-неуязвимость
ODYSSEY 1	25300,0
OLLI & LISSA	35874,0 36062,0-энергия 36710,24 36095,3 36060,201 35748,201
OLLI & LISSA 2	38298,201
OPERATION WOLF	40734,195 40691,0 40710,0
ORBIX	32127,0 32188,0 (32128,0) 65529,191
OUT RUN	39204,0 40623,0-время
OVER KILL	42968,0
OVERLANDER	29521,0 27296,0 27320,0 61422,0
PACMANIA	35141,182 34914,0
PANAMA JOE	38629,0 38630,0 38633,183
PAPER BOY	48023,201 49263,0-газеты 50577,190-жизни 50495,201-защита

Наименование игры	POKES
PARABOLA	38303,0
PARADISE	50066,0
PARK PATROL	53949,0
PENETRATOR	49917,0 50751,0
PENTAGRAM	49917,0 (44917,9) 50751,0
PETER PACK RAT	27243,100
	27640,0
PHANTOM CLUB	56486,0
PHANTOMAS	44819,0 46149,201
	44518,0
PHANTOMAS 2	26606,0 28452,0
PHEENIX	29375,0
	32232,N
PHOENIX	32232,N (0<N<10)
PI-R2	38473,201 (38173,201) 38752,0
PIBALLED	46441,0 44416,5 46457,0
PIJAMARAMA	48669,0
	48670,0
PINBALL	35237,0-мячи 31566,0
PINBALL WIZARD	48182,0 49054,0 45566,N (0<N<14)
PLATOON	31138,0-гранаты 31268,0 31269,0-удары 31270,0-энергия
POGO	44259,0 44260,33
POPEYE	26095,N 26242,0-время 30057,195-жизни 30490,181 30514,0-ключи
POTTY PIGEON	24556,0 25931,0 27397,0-не отнимают червяка у голубя
POWER BOAT	39375,N
	38421,0 39774,50
PREDATOR	39439,0 39441,0-энергия 36166,201-бомбы
PREDATOR 2	36892,141 36893,0-энергия
	36929,0-пули 36599,0 36600,0-энергия
	36926,0 36895,0 (однако не стоит попадать в мирных жителей, инвалидов и т.п.)
PRINCE GLUMSY	48548,0
PRO SKATE BOARD	34281,0
SIMULATOR	
PROHIBITION	25421,182
PSYTRAXX	48930,N
PULSOID	48113,182 48126,182
PUZZNIC	37913,0
	46080,0-время
	36919,255 (36919,0)
	46045,246 37913,0-попытки
PYJAMARAMA	48670,0 43883,2
	48683,0 33764,201
PYRACURSE	56278,0 23325,201 RANDOMIZE USR 23299 33466,201 (33446,201)
	RANDOMIZE USR 29600
PYRAMANIA	30254,0 30255,0 30256,0 30266,0 30267,0 30268,0 30278,0 30279,0
	30280,0
PYRAMID	44685,0
QUAZATRON	58243,0 58244,0 58245,0 58267,0 58268,0 58269,0
	58288,0
R-TYPE	30873,0
	34374,0 (37374,0)
RAID OVER MOSCOW	37362,201-неуязвимость 65133,24-загрузка любого уровня
	29252,183 40299,182 43364,182 46507,182 49130,182-жизни
	38094,0-время
	40229,0 40300,195 43364,0 43365,0 49130,0 49136,0
RAINBOW ISLANDS	65111,0
RAMBO	27401,52 60263,0
	38841,24 27401,0 27402,201
RAMBO 2	27145,0 27146,0
RAMBO 3	57830,0 57831,0 57832,0
RAMPARTS	35148,201-энергия 1-го игрока
RASTAN	48909,N-жизни

Наименование игры	POKES
RASTAN (128K)	39710,0-энергия 40084,62-жизни 39393,0-оружие
RASTAN SAGA	55629,182
RAT RACE	56449,0 50940,201 55599,201
REBEL	49958,0-защита 51139,0 49239,0-энергия
RED HEAT	36297,183 33644,183-жизни 33377,N-1-номер уровня 33372,N-жизни
RENEGADE 2	39576,201-устранение зависания после 1-й смерти 41048,195- бессмертие
RETURN OF THE JEDI	52140,0 47817,201 47164,0-уничтожает всех врагов
RIPTOFF	35427,0
RIVER RESCUE	33199,255 33426,0 33452,0 36193,N 36255,255
ROAD RACER	27150,0
ROADBLASTERS	47025,0-турбодвигатель 47541,0-нет углов 48635,0-кредит
ROBIN HOOD	Заменить 195 в списке DATA на 201, после остановки программы: 50532,255 RANDOMIZE USR 35481
ROBIN OF THE WOOD	48690,0 49911,0 48684,0
ROBOCOP	25917,0-бессмертие 25424,0-время 24039,0-турбо 25795,0-без пауз 39577,24-энергия 46229,0 46230,195-энергия 46149,0 46150,201-жизни 39582,0 39588,0 39639,0 40108,0-энергия
ROBOCOP 2	39584,0-энергия
ROCKFORD	61889,255
ROCKY	29538,0
ROGUE TROOPER	30374,0 30924,0 35091,0 30874,0
ROLLER COASTER	38988,255 36594,0 36595,0 36596,0
ROLLING THUNDER	39792,135
ROMMEL'S REVENGE	42976,0
S.D.I.	41229,183 41302,183 46626,58
S.U. CREW	56364,0-время 51765,0-ошибки
SABOTAGE	32979,62 32980,6 32981,50 32982,132 32983,169 33138,0 43396,255
SABOTEUR	29894,0-энергия 47009,0 47010,0 47011,0
SABOTEUR 2	37122,0-время 37121,0 61340,201 61338,182 61382,182 (61382,0)-энергия
SABRE WULF	44786,0 43575,N 39702,30
SABRINA	28738,0-жизни 29071,195-бомба не взрывается
SACRED SWORD	64748,205 64749,86 64750,5 64751,62 64752,0 64753,50 64754,216 64755,221 64756,201
SAI COMBAT	37421,1 (32421,1) 63364,201
SAIGON COMBAT UNIT 1	43528,0-бессмертие 43650,0-бомбы
SAIGON COMBAT UNIT 2	43854,0-бессмертие 43732,0-бомбы
SALAMANDER	48291,24 48261,182-жизни 38902,201-неуязвимость 38498,0 35615,0 38902,201-бессмертие Адаптация к некоторым версиям Spectrum: 25599,255 34635,99
SAM SPADE	25215,0 26381,0

Наименование игры	POKES
SAMANTHA FOX	20728,62 20729,6 20730,50 20731,112 20732,91 20733,195 20734,224 20735,81 24130,6 23408,6
SAMURAI WARRIOR	33006,N-жизни 33013,0 37866,0 37875,0
SATAN	53965,0-время 46589,0-жизни 46389,53 46263,0 46246,0-энергия
SAVAGE 1	35454,N-попытки 37457,0 51573,0 52546,0 37299,255
SAVAGE 2	29177,60-бессмертие 32625,62 32626,0 32627,0 32664,183
SAVAGE 3	44251,0-энергия 34398,255 34459,255 30785,62 30786,N 30787,0-жизни 50482,195
SCHOOL DAZE	30464,201 63935,255 63937,0
SCRAMBLE	30192,201 28051,201
SCUMBALL	49098,N-жизни 54445,0 54446,0 65036,0 65037,0
SECTOR 90	36585,182-жизни 38500,201-неуязвимость
SENDA 1	56955,201
SENDA SALVAJE	57139,0-энергия
SENDA SALVAJE 2	56449,0-жизни
SENTINEL	31684,0
SHADOW SKIMMER	53872,182
SHANGHA WARRIOR	45539,201 (45639,201)-жизни 45539,0-очень интересная вещь!
SHAOLIN'S ROAD	44843,0 44844,0 50274,183
SHARK	54462,201
SHIZOIDS	25102,0
SHORT CIRCUIT	34777,0 33511,255 35485,0 34921,0
SHOWMAN	63197,0
SIDE ARMS	25240,127 29411,120 39511,24-защита
SIDE WIND 2	20653,255
SIGMA-7	34202,19 58524,19 58852,19
SILK WORM	38052,0 38561,0 38505,0-неуязвимость
SIR FRED	46650,183 59788,0-энергия 46650,167-энергия
SIR LANCELOT	33892,0 33893,0 25726,0 25727,0
SIR WOOD 1	38486,0
SIR WOOD 2	38486,0 40742,0
SIR WOOD 3	36268,0
SKATE CRAZY	46473,201 46646,126 (42646,126) 46409,201
SLIGHTLY MAGIC	40975,0
SMASHOUT	59500,0 59501,0 47004,0
SNARE	46840,0-жизни 50263,0-время
SOLDIER OF FORTUNE	44796,0 46691,0
SOLDIER OF LIGHT	50552,0 51119,0 50035,255
SOLOMON'S KEY	49344,0 (49344,183)-жизни 50831,0-fireball 49318,201-время и неуязвимость 37663,N-жизни 51048,0
SOS	33951,0 33091,0 35238,0 34764,201 RANDOMIZE USR 32768
SPACE RAIDERS	25962,0
SPACE WARS	26244,0 43364,182 46507,182 49130,182
SPACE ZOMBIES	29553,0

Наименование игры	POKES
SPECTRAL INVADERS	25062,254
SPECTRAL PANIC	28522,0
SPECTRES	25680,183
SPHEROID	59366,183 (59366,0)
SPITFIRE-40 (128K)	29030,201-переход к другой миссии
SPLIT	54397,99 54398,2
PERSONALITIES	
SPOOKED'89	54665,201
	61222,201
	60504,255
SPY HUNTER	54824,201-жизни 57450,0-время
STACK UP	39220,0
STAINLESS STEEL	46957,60
STAR FOX	35876,0-горючее
STAR GLIDER	54647,201-жизни 54690,201-защита
STAR PILOT	44130,0 44393,0
STAR RAIDERS 2	44107,60 44108,0 (44108,3)-энергия
	46214,195-жизни
STAR WARS	45268,0
STARCLASH	25381,183
STARION	46271,0 46272,0 46273,0
	46599,98 46600,2
	46600,0
STONKERS	24576,0
STORMLORD	58105,0 56877,0 (56877,201) 56890,255 34650,0
	35923,0 35710,0 35711,0 35712,0
STORMLORD 1	RANDOMIZE USR 60720: RANDOMIZE USR 24500:POKE 56878,58: RANDOMIZE USR 18432
STORMLORD DEMO	57412,0
STRANGLE LOOPE	63159,0 63161,0 63160,182
STREAKER	50221,0-жизни
STREET FIGHTER	41740,24-время
STREET GANG	37128,0 37129,0 37130,0 37364,0 37365,0 37366,0
	37127,39 39254,0
STREET HASSLE	49883,255
STREET HAWK	39990,91-энергия
STRIKE FORCE COBRA	48389,0 46499,0 46500,0 46501,0
STRIP DICE	27418,45 27656,45
STYX	31763,0 31764,0 31765,0 31766,0
	29856,255
SUBTERRANIAN	43435,0
NIGHTMARE	
SUBWAY VIGILANTE	53767,0 53781,0-жизни 53359,0 53383,0 27115,0 27127,0 27134,0
SUPER CHOPPER	48827,0
SUPER TANK	44022,N (44027,N)
	37295,201 56813,201
SUPER TRUX	24498,134
SUPER WONDER BOY	45043,0-кредиты
	32670,195-кредиты
SUPERCYCLE	43560,150-время
SUPERHERO	49625,0 40444,0 46334,255 46335,255 46336,255 46337,255
	RANDOMIZE USR 40341
SWEWO'S WORLD	33219,0 34912,0 35732,37
	37008,255
TAI-CHI TORTOISE	56064,N-оружие 65121,183-жизни
	65121,0
TALKING HEDZ	52611,0
	50455,0
TANIUM	39552,0-неуязвимость
	34253,0 38648,0
TANK	24629,0
TANTALUS	42228,0-жизни 42247,0-защита
	42871,0 RANDOMIZE USR 22000

Наименование игры	POKES
TARGET RENEGADE	62728,0 23431,255 62936,0-время
TARZAN	51002,183 51185,183 52268,183-энергия 51013,0-время
TECHNICAN TED	44258,0
TECHNICIAN TED (128K)	43766,0 43788,24-жизни 43147,33 44485,24-защита 41067,0-время 43201,255-падение с любой высоты 44092,0-сквозь стены
TENEBRAUM	55231,0-жизни 33541,201-время
TERMINATION	55274,0-энергия (жизни) 53120,0-время 55330,60-субигра
TERRA CRESTA	35050,0 35051,0 35052,0 37636,255 45282,0
TERRAMEX	36844,0
TERRORDAKTIL	37629,0
THE FAST'N'FURIOUS	59553,24 59554,75 59353,58 59526,58-энергия 60770,201-защита
THE MUNSTERS	34001,24-энергия
THE RACE AGAINST TIME	58589,201-время
THE ROCKY HORROR SHOW	неуязвимость от: 56304,175-дворецкого 48788,0-лазера 48788,0- время 47250,0
THE WIZARD	47250,0
THING BOUNCHES BACK	63852,107-нефть 63853,175-жизнь 63855,145 63856,176-выходы 63858,32 63859,178-нет дверей 63865,205 63862,177-нет лазера
THRUST 2	34200,0-защита
THRUSTA	30455,0 RANDOMIZE USR 30700
THUNDER	59487,201
THUNDERCEPTOR	54487,201
TIGER ROAD	51489,195
TIME FLIGHT	27357,0
TIME SCANNER	27448,0
TIR-NA-NOG	31365,201 34751,201 33727,24 30801,195 35421,24 34202,255
TITANIC	54642,0-оружие 59190,0-жизни 59043,0-энергия
TITANIC 2	54694,0-оружие 59531,0-жизни 59389,0-энергия
TLL	35006,0 (55006,0) 33807,0
TOKYO GANG	28920,0
TOP CAT	62988,201-энергия
TOP GUN	26460,0-жизни 1 игрока
TOTAL ECLIPSE 2	46365,0 48496,0 48152,0 51270,0
TRANSMUTER	28878,0
TRANTOR	54236,0 56596,0 49505,0-жизни (энергия) 49620,0-время 47228,0
TRANZ AM	25446,0
TRASHMAN	52037,0
TRAVEL W.T.	38656,183
TREASURE ISLAND	59672,0 60126,0 61142,0 61281,0 59669,0 60127,0 61143,0
TRIBBLE TRUBBLE	32110,0 32849,0 34214,0 35026,0 31981,0 31984,24
TROUGH TRAP DOOR 2	47492,0
TUJAD	27212,183 27212,0 27213,0 31443,0-жизни 34549,0 34626,0 34474,0-оружие
TUNNEL	27604,0
TURBO ESPRIT	29893,0
TURMOIL	57557,0
TWISTER	42412,0 42490,0-энергия 42422,201
TYPHOON	39143,0 39303,0
U.C.M.	23296,255 36337,0 43734,0
U.N. SQUADRON	29427,0
UNDER WURLDE	59376,0 38041,0 38042,0 59591,0 45019,201 59380,0
UNITRAX	29228,0-энергия 29264,0-жизнь

Наименование игры	POKES
URIDIUM	55419,0 55420,255 35403,34 38998,201 35427,195
VENDETTA	40228,201-время
VENOM STRIKES BACK	45600,0-оружие 49941,201-нет монстров 45803,201- водонепроницаемость
VICTORY ROAD	39385,255 39402,255 34784,182
VIDEO POOL	38753,0
VIEW TO A KILL	28032,255 28087,8 29243,8 30543,8
VIGILANTE	48735,0 39921,255 48735,60
VINDICATORS	37913,0 38094,0
VIRUS	44945,0
W.D.W. 2	49743,N -гранаты 48748,N -жизни 51847,0 50833,0
WALLY	58215,182
WAR	38394,0 37033,0 33033,0
WAY OF THE EXPLODING FIST	27055,0 27056,0 27063,1
WAY OF THE EXPLODING FIST 2	27064,243 27065,243 27233,255 RANDOMIZE USR 61710
WAY OF TIGER	45532,201 45178,201 45806,201 45554,183
WAY OF TIGER 3	45828,183
WEC LE MANS	25879,0-время 26110,34
WEST BANK	52518,0 52544,183 (52544,0) 52573,183-жизни
WHO DARES WINS 2	50814,0-жизни 51422,0-гранаты 49743,255 49748,255 51847,0 50833,0
WIKING	63033,1-жизни
WIZBALL	37052,0-жизни 48190,201-защита
WOJTSOFT	32179,0
WONDER BOY	34362,0-жизнь 34106,195 40816,195 41250,195 41870,195-защита 34338,0-топор 35305,24-прохождение любого уровня
WONDER BOY IN MONSTERLAND	40131,0 45852,0-энергия 45845,0-время
WORSE THINGS	33221,0 35303,0 40133,0 42215,0
WRIGGLER	50173,0
WULFAN	25862,0-жизни 31273,0-энергия
XARAX	41352,0
XENO	22225,1
XEVIOUS	53592,N-жизни / 54050,0 54501,0 55333,62 55334,5 55335,50 55336,4 55338,201 PRINT USR 24833
YENTUROMA	53633,0
YETI	47894,0-жизни 47665,0 47666,0 47667,0-неуязвимость; неуязвимость от: 56313,0-пушки 53751,0-колья 58103,0-пулемет
YOGI-BEAR	33912,24
ZAXXON	48825,N 50244,183
ZOLYX	50476,0-время 50800,0-жизни
ZUB	37473,201
ZYBEX	45277,0 (45222,0) 45368,0
ZYNAPS	45592,24 45424,0 45425,0 45426,0

ПАРОЛИ И СОВЕТЫ.

Наименование игры	Пароли и советы
AGENT 1000	Пароль: there's no escaping it
ALIENS	Пароли: 7324G 2727H 1506E 3761H 7100D 7123G
ARKANOID	Для бессмертия в таблице результатов напечатать PBRAIN
ARKANOID 2	Вместо имени ввести МААААН или во время игры нажать BONK
ARKOS 2	Пароль 19378
ARKOS 3	Пароль 15720
BIGGLES 1	Вместо имени ввести DADD
BLACK CRYSTAL	Пароли: 512661220, 1126690200, 1126671220, 3126641220, 2126671220, 1126290200
BMX SIMULATOR	Нажать ТАЕС
CHASE H.Q.	Перезадать клавиши: SHOCKED, нажать "1" для загрузки 2 уровня, "2" для 3 уровня и т.д.
COMBAT ZONE	В режиме ПАУЗА нажать: 0+P+ENTER+Z-значение жизней и энергии = 255 0+P+ENTER+X-значение топлива = 60000 0+P+ENTER+C-супербомба
CYBERNOID	Нажать YXES
EMPIRE STRIKES BACK	Для бессмертности нажать в меню ZXCV<CapsShift>
GHOST BUSTERS	Номера счетов: HERBIE-05250624, PETER-50338, LH-70204700, ICH-12345678 TANG BILLY-15570011
HAMMERFIST	Пауза, нажать YCY, затем L-переход в следующий экран. Для бессмертия нажать NYC
HELICHOOPER	Пароли: PASSWORDS SHOW VOREVER RESTART CLEAR
HELTER SCELTER	Пароли KICK TOUR DALI DRUM JOKE LISE
IRON SOLDIER	Для бессмертия нажать GAD
JACK THE NIPPER	Для перехода в режим "CHEAT" набрать ZAPIT
JASONS GAME	В меню нажать SWA
MOONLIGHT MADNESS	Падая, нажмите пару раз "L"
NARCOPLICE	Пароль CONGRA переводит на следующий уровень
NIGHTMARE RALLY	Нажать <S.S.>+Q или <S.S.>+W
PIPEMANIA	05 уровень-DISK, 09 уровень-NAIL, 13 уровень-ONCE, 17 уровень-ROPE, 21 уровень-PENS, 25 уровень-SLIP, 29 уровень-EACH, 33 уровень-RISE
RED HEAT	Нажать SymbShift + все цифровые клавиши
RENEGADE 3	В меню нажать "Q" и "T"
REPTON	Пароли: SCREEN ONE, ASP, CROCODILE, EARTHWORM, SEASNAKE, ANEMONE, BASILISK, CEPHALOPOD, ANNELID, LEVIATHAN, OPHIDIAN, KING COBRA
SALAMANDER	Адаптация к некоторым версиям Spectrum: POKE 25599,255: POKE 34635,99
SAMURAI WARRIOR	Вместо имени ввести CHEAT
SANXION	Перезадать клавиши CHEAT и вписать пароль: CYNN
SAVAGE 2	Пароль SABBATTA
SAVAGE 3	Пароль FERGUS
SENDA SALVAJE 2	Пароль PLATANON
SHINOBI	Клавишу MAGIC определить как <C.S.> и <ENTER>
STAINLESS STEEL	Для бессмертия нажать ALIK (или SLIK-на КВАНТе)
STAR FORCE	Перезадать клавиши TRONIC
STAR STRIKE	Нажав паузу, набрать I WANNA CHEAT и для выхода набрать BORED
STAR STRIKE 2	Нажав паузу, набрать HEAR AND OBEY
STAR WARS	В коридоре к реактору нажмите 8DFG<Space>, выйдя нажмите YOUI<Space>
STORMLORD 2.2	Пароль AK2PFESK (RRRPFDHW) / Отключение пароля: POKE 26897,201
STORMLORD 2.3	Пароль RA5EFECD (ALVEFCXO) / Отключение пароля: POKE 26897,201
STREET HASSLE	Нажать L для загрузки следующего уровня
TASK FORCE	Перезадать клавиши CRASH
THEATRE EUROPE	Пароль: NUCLEAR CODE? Отзв: MIDNIGHT SUN
TOI-2	Код 517
TOI-3	Код 124

Наименование игры	Пароли и советы
TOI-4	Код 500
UNTOUCHABLES	Ввести вместо имени HUMPHREY BOGART
VAMPIRE	Нажать 12345
WHEELIE	Пароли WITTY SHARK BEBOP XENON ZX833 2MQL3 HRME2
WIZARD'S LAIR	Пароли HAWLO CAIVE VAULT LIAYR DUNGN CRYPT LYONS
XARQ	Ввести XARQ
XENON	Во время игры нажать BREAK затем TINY
ZAXXON	Для невидимости ввести RED
ZODIAK STRIP	Нажмите J

И в заключение, более подробный материал по игре "BRAIN SPORT".

© Андрей Бастраков, г. Челябинск.

Ещё великий Остап Ибрагимович говорил: "Главное - помочь детям..." (видимо, из этого исходят и фирмы, издающие описания игр, тем не менее, с массой недостатков). Больно глядеть на бесчисленные страдания детей моих знакомых, желающих узнать, а что же там, на 99 уровне, когда их отрывают от игры в самый ответственный момент, чтобы посмотреть очередную "Марию" или "Барбару". Я подумал, что, наверное, такое безобразие творится везде, и решил затронуть проблему игры "BRAIN SPORT".

Это сравнительно простая логическая игра, цель которой - установка ящиков в лабиринте в заданные места. Однако 99 экранов за один раз пройти физически очень сложно. В игре предусмотрены режимы, существенно облегчающие жизнь, но они не отмечены в многочисленных описаниях.

Режим "SELECT FIELD" позволяет начать игру с любого уровня, если, конечно, известны пароли. Но его особенность в том, что он может быть установлен, только начиная со второго уровня. Сделать это можно так:

1. Находясь во втором или более высоких уровнях, нажмите клавишу "Т", затем "У" и выйдете в основное меню.
2. Выберите пункт "SELECT FIELD" и нажмите ENTER. На экране появится номер уровня, в котором Вы только что были и предложение ввести пароль.
3. Нажмите BREAK (CAPS SHIFT+SPACE), останется только номер уровня.
4. Клавишей DELETE (CAPS SHIFT+0) удалите номер, введите нужный и нажмите ENTER.
5. Введите пароль этого уровня и - приятного развлечения!

Для самых нетерпеливых несколько паролей:

10 - GOTOTOGO	40 - MAD MATH	70 - POKETOWN
20 - HATE	50 - ICEBERGH	80 - HEDGENOG
30 - OVERKILL	60 - WAS EASY	90 - ACIDBEAT

Кроме этого, если в ходе игры нажать "S", то текущее состояние экрана будет записано в памяти. Теперь, если Вы ошиблись, нажмите "B" и оно будет восстановлено. Запоминать свое состояние можно неограниченное число раз, а надпись "BACK=" в нижней части экрана показывает количество возвратов.

ПРИЛОЖЕНИЕ

Программа для ввода шестнадцатеричных дампов с контрольными суммами.

```

100 BORDER 7: PAPER 7: INK 0: CLEAR 29999
110 DIM a(10)
120 DEF FN A(a$)=(CODE a$(1)-48-(7 AND a$(1)>"9"))*16+(CODE a$(2)-48-(7 AND
  a$(2)>"9"))
1000 POKE 23658,8: INPUT "DATA: "; LINE h$
1020 LET a$h=h$
1030 LET sum=0
1040 FOR i=1 TO 2
1050 LET b$a$(2*i-1 TO 2*i)
1060 LET a(i)=FN A(b$)
1070 NEXT i
1080 LET add=a(1)*256+a(2)
1090 LET sum=a(1)+a(2)
1100 FOR i=3 TO 10
1110 LET b$h$(2*i-1 TO 2*i)
1120 LET a(1)=FN A(b$)
1130 LET sum=sum+a(1)
1140 POKE add,a(1)
1150 LET add=add+1
1160 NEXT i
1170 LET b$h$(21 TO )
1180 LET cs=FN A(b$)
1190 LET csl=sum-256*INT (sum/256)
1200 IF cs<>csl THEN PRINT a$; INVERSE 1;" ERROR ! ": BEEP 1,0: LET
  add=add-8: GO TO 1000
1210 PRINT a$( TO 4): GO TO 1000

```

Запустите программу: RUN и начинайте ввод кодового блока, согласно шестнадцатеричному дампу с адресом вначале и контрольной суммой в конце. Данные надо вводить без разделителей и пробелов. Например, для строки

```
9C40: 21 E5 9C 11 E2 9C 01 03:11
```

ввод должен выглядеть так:

```
9C4021E59C11E29C010311<ENTER>.
```

В процессе набора периодически сохраняйте частично набранный кодовый блок на магнитной ленте или дискете. О работе программы см. также "ZX-РЕВЮ" № 1-2 за 1993 г. стр. 7.

Для набора шестнадцатеричных дампов Вы можете воспользоваться также программой "Memory Editor" Александра Гмарь (см. "ZX-РЕВЮ" № 1 за 1994 г.), а также её усовершенствованными (релоцируемыми) версиями ("ZX-РЕВЮ" № 3 за 1994 г.).

**ВНИМАНИЮ ЛЮБИТЕЛЕЙ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА "ZX-SPECTRUM" и
совместимых**

г. МОСКВА, ул. Новый Арбат, д. 2. 19-е отделение связи, 1-ый этаж операционного зала, с 10 до 17, обед с 14 до 15, выходной воскресенье.

г. МОСКВА, радиорынок в Митино, проезд до ст. метро Тушинская или поездом до пл. Трикотажная (Рижское направление). Суббота, воскресенье, 9-14, место А-37.

г. МОСКВА, радиорынок в Царицыно, проезд до ст. метро Царицыно. Суббота, воскресенье, 9-14.

г. 163051, г. Архангельск-51, а/я 51.

г. БЕЛГОРОД, магазин "РАДИОТОВАРЫ", ул. Ленина, 32.

г. БЕЛГОРОД, Октябрьская 84, кв. 103, "СТУДИЯ КОМПЬЮТЕР".

г. ВЛАДИВОСТОК, Океанский проспект, 140, магазин "ПАНОРАМА". Проезд трамваем до ост. Некрасовская.

г. ВОРОНЕЖ, Студия компьютерных игр SAN-SAN. Магазин-салон "ЭЛЕКТРОНИКА", тел. 14-00-73.

г. ДНЕПРОПЕТРОВСК, ул. Шевченко, 34. Фирма "ЭКОС".

г. ЕКАТЕРИНБУРГ, магазин "СПЕКТРУМ", Главный проспект, 99.

г. ЕКАТЕРИНБУРГ, ул. Коминтерна, 5, общежитие РТИ, магазин "РАДИСТ".

г. ИЖЕВСК, радиорынок "Березовая Роща".

г. КЕМЕРОВО, магазин "ТЕХНИЧЕСКАЯ КНИГА", ул. Весенняя, 24.

г. КЕМЕРОВО, магазин "ОРБИТА", пр. Ленина, 133.

г. КИРОВ, "Дом науки и техники", Магазин-салон "МАРС", ул. Производственная, д. 27.

г. КРАСНОЯРСК, радиорынок, проезд до ост. "Затон", суббота, воскресенье.

г. МИНСК, Сторожевский радиорынок, сектор 8, место 7.

г. НАБЕРЕЖНЫЕ ЧЕЛНЫ, Татарстан. Новый Город, салон-магазин "ПРИНТЕР".

г. НИЖНИЙ НОВГОРОД, ТОО "КРИОНН", Магазин "ФОТОЛЮБИТЕЛЬ", ул. Горького,

146.

г. ОРЕНБУРГ, магазин "ВОЕННАЯ КНИГА", ул. Советская.

г. ПЕРМЬ, Комсомольский проспект, 86, тел. 64-32-23, магазин "ЮНЫЙ ТЕХНИК".

г. ПРОКОПЬЕВСК, Кемеровской об. Тырган, магазин "ОДЕЖДА", фирма "РОНЭТ".

г. ПРОКОПЬЕВСК, Кемеровской обл., ул. Артема, 8, магазин фирма "АРБАТ".

г. РЫБИНСК, ул. Гоголя, 1. ТТЦ "ГНОМ".

г. С.-ПЕТЕРБУРГ, радиорынок, "АВТОВО", киоск № 49 ("SCORPION-256").

г. ТУЛА, Мясовский рынок, суббота, воскресенье.

г. ТУЛА, 300012, Тула-12, а/я 409.

г. ХАБАРОВСК, ул. Запарина 65, магазин "ФИЛАТЕЛИЯ".

г. ЧЕБОКСАРЫ, магазин "ЭКСПРЕСС", НПК фирма "НОВА", ул. Привокзальная, д. 6.

г. ЧЕЛЯБИНСК, ул. Елькина, 45 (напротив Публичной библиотеки).

г. ЯРОСЛАВЛЬ, магазин "РАДИОСПОРТ-ТУРИЗМ", Ленинградский проспект.

