

ЛЭЗ

ZX
Спектрум

ГАЗЕТА ДЛЯ СПЕКТРУМИСТОВ /// № 25, 2005

СЕГОДНЯ В НОМЕРЕ:

Новости 2-3 стр.

Новинки ПО 4-6 стр.

СС'2005 6-8 стр.

Исповедь производителя

8-11 стр.

CD-ROM на Спектруме

11-27 стр.

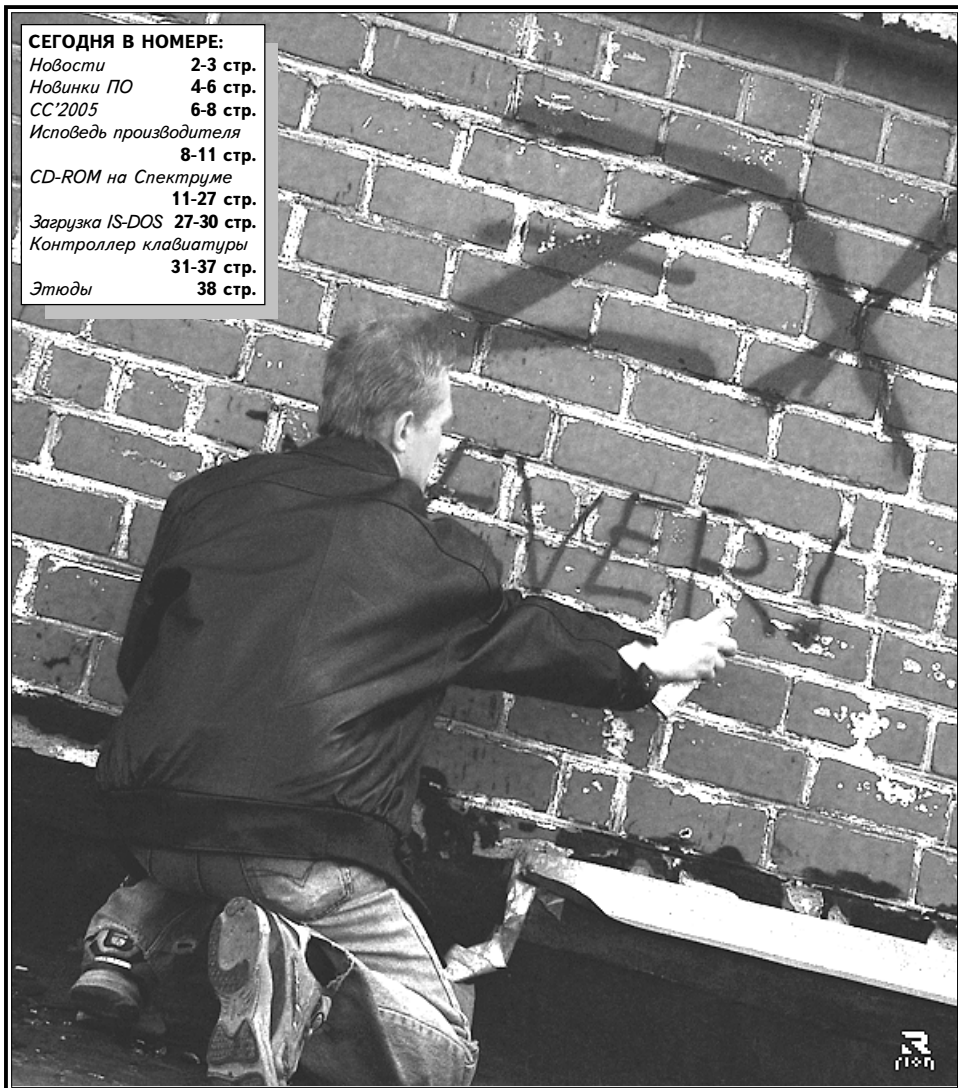
Загрузка IS-DOS 27-30 стр.

Контроллер клавиатуры

31-37 стр.

Этюды

38 стр.



Новости ///

Новости редакции

«Твоя игра-3». Сроки конкурса все еще не определены, а вот призовой фонд уже начал расти. На данный момент он составляет 4355 рублей (\$155). Мы все еще ждем ваши ответы на вопросы связанные с конкурсом (см. 24 номер газеты).

Dune 2. На выкуп данной игры собрано 1630 рублей. Осталось собрать 1370 рублей. По поводу того, как распространять игру предложений пока не поступало. Если они не поступят на момент сбора полной суммы выкупа игры, то будем поступать согласно предложению высказанному в прошлом номере «Абзаца».

Активный спектрумист. Акция временно приостановлена ввиду отсутствия активности голосующих.

Книги. Благодаря Роману Миндлину у нас появилась большая «библиотека» спектрумовской литературы. Книги будут распространяться по почте и каждый желающий сможет приобрести интересующую его книгу. Скорее всего, будет определена единая цена на любую книгу «библиотеки». Все средства полученные от продажи книг пойдут в фонд конкурса «Твоя игра», а так же на изготовление будущей брошюры по программмированию. Остальные подробности, а также полный перечень книг будет опубликован в следующем номере.

Заказ плаката. Стоимость полноцветного плаката формата А3 (подробнее смотрите в 24 номере нашей газеты) с учетом пересылки по России составляет **60 рублей**. Заказы принимаются на адрес редакции.

Новости от Дмитрия Быстрова

Как и было обещано, вышли ALASM 5.0 и STS 7.0, отличающиеся от старых версий полной терпимостью к Turbo и более плавным движением курсора. Поскольку версий с этими номерами я выпустил несколько, то укажу самые важные отличия последних из них от первых.

ALASM:

+ SAVEOBJ4 может записывать не с адреса Start. Так можно сделать 3-символьное расширение;

+ Enter на 8-й позиции не сдвигает строку влево (нужно для отрезания меток);

- INCLUDE мог забыть имя главного исходника;

- MOVE не работало на DOS6.05E;

+ новая директива «RUN адрес» - вызывает при компиляции любую подпрограмму пользователя;

+ при выборе текущей страницы (Alter) выводятся номера страниц;

+ настройка адреса начала страницы макросов (для 128k машин).

STS:

- страницы при трассировке переключает только порт #fd со сброшенным A15 (для отладки программ под винчестер).

- при дисковых операциях принудительно включается IM 1 и IY=23610. Отличия STS7.0 от других старых версий гораздо шире. Например, резидент сокращен на 3 байта и не портит RAMTOP при размещении резидента в системных переменных; полный #3d13 (невозможно «запороть» диск); возврат в вызвавшую программу через RET (для Скорпионов очень важно) и др. Полный список - в описании. Теперь STS гораздо удобнее версий 6.x, но поддержку памяти не по порту #7ffd обеспечивает, к сожалению, только v6.x.

Параллельно был наконец-то дописан **Gluk6.0R** (а потом и Gluk6.1R - в приложении к Info Guide #7), прошивка ПЗУ, не побоюсь этого слова, НОВОГО поколения. Отличий от старых версий Gluk в ней больше сотни. Были внедрены лучшие идеи из популярной прошивки MadROM. Добавлены многие важные сервисы, такие как сохранение ALASM в памяти, показ регистров и стека на момент сброса, цветовая таблица, дисковый доктор, настройщик CMOS, загрузчик с HDD, разрезание файлов и др. И все это проверено на двух настоящих «Пентагонах». При этом есть ощущение (но чувства, конечно, часто обманывают), что Gluk6.x уже сэкономил мне потраченное на него время.

В состав Gluk6.1R входят DOS6.10E и Perfect Commander 2.1. В DOS (напомню, она отличается от 6.05E направлением роста RAM-диска - теперь можно задавать RAM-диск любого размера) убрал якобы «фикс» команды PEEK, из-за которого менялось расширение (#5ce5) при команде чтения файла, а в результате глючил WolfEd (редактор уровней для Wolf2004). В Perfect (он используется также как отдельная программа) появилась поддержка памяти ATM Turbo. По предложению Zeg/Fenomen я отправил статью про Gluk в журнал «Радиомир» - обещали опубликовать в #8.

Наш регулярный (раз в полугодие) журнал **Info Guide #7**, немного задержавшись, вышел в объеме двух дисков. Текста там гораздо больше, чем в прошлый раз, и даже больше, чем в Inferno #4, державшем прошлый рекорд. В Info Guide также опубликована ПЕРВАЯ операционная система на ZX для работы с файлами на FAT16 (автор - Дмитрий Аврята, Харьков). Уточню: FAT16 - это стандартная файловая система для винчестеров на IBM PC, так что теперь стало возможно перебрасывать с одной машины на другую ГИГАНТСКИЕ объемы данных. Между прочим, Дмитрий Аврята на днях доработал известную TR-DOS-прошивку Влада Мат-

Новости

лаша для HDD («прозрачно» работающую с образами дисков на винчестере) под стандартные порты Немто, и она может функционировать у большего числа людей.

Другая программа, также попавшая в приложение к журналу - **Pro Tracker v3.693**. Там, в частности, исправлены ошибки с копированием, редактор сэмплов увеличен на 3 строчки, показывается режим Edit Melody. Подверсий было две, первая (неработоспособная) была в копии журнала, выложенной на Scenergy, вторая - в копии журнала, помещенной в эхоконференцию ZX.SPECTRUM.

В ACEdit0.95 добавлено по сравнению с 0.91:

- + ssl, если курсор в окошке поиска стоит внизу, ищет точное совпадение кол-ва пробелов (а если вверх - неточное);

- + еще три кодировки текста (в том числе кодирование файлов для передачи через плохие сетевые узлы) и инструкция по декодированию других двух;

- + имя файла после разрезания изменяется (чтобы случайно не сохранить поверх);

- + новый плейер от Сергея Бульбы: играет PT2 и PT3;

- + плагин-игра acePITON.

И исправлено:

- не работали ss/Break, ss/Space, ss/Enter, в 4-й раскладке не работали заглавные «Э» и «Ж», запятая и др.;

- ssl переставлял курсор замены на 0-ю позицию;

- MOVE не работало на DOS6.05E;
- после ответа Y/N символ ответа попадал в текст;

- Бейсик-загрузчики нельзя было запускать с другого дисководы;

- при постраничном листании вниз был неверный номер строки.

В приложении к ACE0.93 (наряду с ACNews#40) и 0.95 (наряду с ACNews#42) были размещены (в листалке RarView) различные тексты для общего развития - начинаем понемногу воплощать в жизнь заброшенный было проект «электронных книг».

В ZXUnRar0.58 исправлен поиск, также исправлена упомянутая листалка RarView и ее аналог fom&bush (он отличается палитрой) - последний рекомендуется все-таки заменить на диске с журналом Fantadrom #4 (а недавно вышел #5, там есть и мой рассказ).

В ZXRAR0.27 добавлена функция упаковки каждого файла в отдельный архив, при этом одноименные архивы не стираются, и исправлен глюк: после прерывания Pack test не работала обычная упаковка.

В диск-докторе для винчестеров под названием **HDDoct04** наконец-то появилась работа с CD-ROM (но инициализировать его нужно внешней программой, например, CD-Walk, т.к. я не смог подобрать алгоритм инициализации), а также запись/чтение секторов на дискету, автоматическое переключение номера сектора при листании и др.

Новости от группы NedoPC

22 июля 2005 года был завершен один из самых долгих и ожидаемых проектов на компьютере ATM. А именно была закончена прошивка eXtra BIOS под ATM-turbo 2+, содержащая в себе уже давно разрекламированный **эмулятор 1818BG93 (так называемый vTR-DOS)**, программный перехват обращения к портам памяти (впервые на Спектруме есть подобие защищенного режима!), поддержку работы с внешними модулями и многое другое.

Прошивка сделана в виде двух модификаций - **xBIOS v.1.32LE (Light Edition)** - облегченная, для ПЗУ 64Кб, для тех, у кого нет возможности прошить более объемную микросхему. Но за все надо платить - в результате получаем меньше возможностей, функций, а также невозможность дальнейшего расширения и совершенствования. Развитие этой версии остановлено.

xBIOS v1.32XT (eXTended) - базовая, для ПЗУ 128Кб и выше (вплоть до 1024Кб). Полный набор всех функций, открытая архитектура (возможность пользователю самостоятельно добавлять в свободные страницы ПЗУ модули собственной разработки с последующей работой с ними через стартовое меню). Данная система будет развиваться и дальше, но уже по принципу совместимости снизу-вверх.

Кроме того, сейчас ведутся работы по переносу эмулятора BG93 на Pentagon-1024SL. Разумеется, переносу подвергнется только то, что можно перенести. Например, при отсутствии там диспетчера памяти и скрытых портов реализовать защищенный режим невозможно.

Саму прошивку (LE и XT) вместе с документацией, а также пакет утилит для работы с эмулятором BG93 из среды CP/M, можно скачать на сайте <http://atmturbo.narod.ru>

Проект не остановлен - прошивка будет развиваться и дальше, будут дополняться документация и совершенствоваться прикладные утилиты. Но главная и самая трудная работа уже позади!

В завершение хочу выразить огромную благодарность Юре Радаеву (ukms[z]) за предоставленную титаническую работу по созданию ядра эмулятора BG93.

Максим ТИМОНИН

Новости с мягкого фронта

Релизы западных игр

Star Wars Collection. Starwars - new hope, Starwars - empire strikes back, Starwars - return of jedi. (c) 1987-1988 Domark. В данной коллек-



RECOMMENDED!

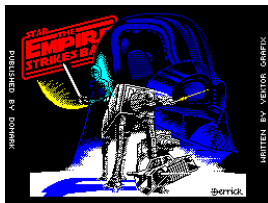
ции представле- ны три игры из серии «Звезд- ные войны», одна другой ин- тереснее.

Наиболее распространен- ной была пер-

вая часть, суть в которой сводится к следующе- му: Вы в качестве пилота корабля X-Plane долж- ны будете проложить путь до «Звезды смерти», по пути уничтожая истребители и прочие кораб- ли противника, с тем, чтобы заложить заряд и разнести «Звезду» на клочки. Вы сможете вы- брать несколько уровней сложности, characterи- зующиеся не только количеством противника, но и другими особенностями, которые раскроются в процессе игры.

Empire strikes

back имеет много общих черт с пер- вой частью, однако различны их цели, а также кое-какие детали в оформле- нии. Также часть полета будет про- ходить над поверхностью планет, где ваш корабль попадет под шквальный огонь большого ко- личество разнообразных роботов. В игре они доставят вам массу неприятностей.



RECOMMENDED!



RECOMMENDED!

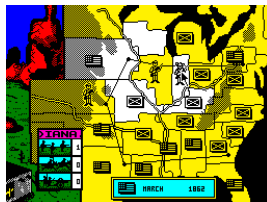
Return of Jedi

отличается от сво- их собратьев очень значительно, за ис- ключением основ- ной цели игры. Здесь вы не уви- дите стай проволо- чных кораблей и

тучи снежинок, символизовавших выстрелы врагов. Теперь вы оседлае реактивный мото- цикл, робота, а также старые добрые косми- ческие корабли и будете смотреть на мир в изомет- рической проекции

North and South 128k (C)1989. Dupuis/ Infogrames. В этой игре затронута такая истори- ческая веха, как гражданская война в США. Уч- итывая, что на территории этой страны было всего два масштабных конфликта - война за незави- симость и война Севера и Юга, все же стоит заост-

RECOMMENDED!



рить свое вни- мание на этом событии, благо оно обращено в такую симпатичную форму. Наряду с теми особенно- стями, извест- ными по ранее выпущенным версиям, эта обла- дает своими специфическими особенностями. В частности, вставлена дополнительная графика и, внимание, потрясающий final cut.

Mr. Heli +3 128kb (c) Irem Software.

Новый релиз от группы Triumph пове- ствует нам о приключениях мистера Heli - прямоходящего летающей устан- овки залпового огня. Являясь конверсией (скорее всего, с какого-то игрового ав- томата), эта игра, тем не менее, выдержана в луч- ших традициях спектрумовских космических стре- лялок, что позволяет ей занять достойное место среди Zynaps, R-Type, Far Star и многих других.

Вы будете управлять некой машиной, способ- ной как летать, так и достаточно шустро бегать по полу лабиринта. Кроме того, в процессе игры, возможно увеличить тактико-технические харак- теристики mr. Heli, если разрушать бетонные бло- ки. Внутри можно найти либо деньги на модер- низацию (количество наличности обозначено в центре над игровым экраном), либо какой-то бо- нус, увеличивающий силу уже имеющегося ору- жия. Кроме того, в каждом уровне вас будут под- стерегать два босса: один в середине, после сра- жения с которым, перед вами предстанет и вто- рой. Благополучно разобравшись с последним босом, вы получите небольшую денежную пре- мию и перейдете на следующий уровень.

Slow Glass (c) 1990-2005, M. Dominguez A. Perez.

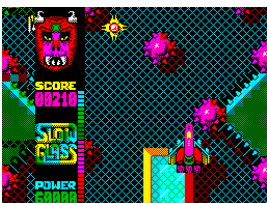
У любителей космических стрелялок празд- ник. Очередной релиз от Triumph - и опять тот же жанр. Но это не обычная игра, в ней есть... Но обо всем по порядку.

После загруз- ки вы выбе- рите себе управление, и игра начнется. Управляя небольшим кораблем, вы будете лететь вперед по



RECOMMENDED!

RECOMMENDED!



Новости с мягкого фронта

туннелю, уничтожая различные препятствия на своем пути. А их не так уж и мало. Караваны вражеских истребителей, мины, лазерные заграждения - и в довершении всего огромный монстр в конце уровня. Но добраться до него будет непросто. Не последнюю роль в этом играет странная особенность в передвижении вашего корабля. Во-первых, вы не сможете двигать его по диагонали. Во-вторых, пейзаж начинает скроллироваться, когда корабль находится в верхней части игрового экрана, что затрудняет обзор и иногда является причиной лобового столкновения с врагами. Тем не менее, продавшись сквозь тучу врагов, вы наткнетесь на главного монстра, где придется изрядно попотеть, прежде чем избавиться от его навязчивого присутствия. Однако на этом ваше сражение с боссом не закончится, оно лишь перейдет в иную стадию.

Если вы успели заметить, позади монстра имеется набор маленьких кирпичиков. Так вот, после убийства босса игра переключится в стадию, очень похожую на игры арканоидного типа - двигая небольшую платформу, вы должны шариком выбить все кирпичики и тем самым расчистить себе путь. Управление платформой осуществляется клавишами О, Р вне зависимости от того, какое управление было выбрано в стартовом меню.

FRIGHTFUL (c) 1989 New Frontier. В этой игре



вы будете исполнять роль викинга, который будет бродить по подземельям и катакомбам, уничтожая бесчисленное множество са-

мых разнообразных врагов и избегая подлых ловушек. В ваших руках окажется магический меч, способный выпускать сгустки энергии, благодаря которой можно убивать врагов, находясь на достаточном расстоянии. Однако будьте экономны, ибо зарядов мало, а пополнить их запас сложно.

Стороной обходите черные провалы в стенах - неизвестно, кто там прячется, но если не прыгать, вас туда затянет - и прощай жизнь. К слову, на игру дается запас из 9 жизней. Вас ожидают чумовые прыжки, сумасшедшие взмахи меча и головокружительные забеги по лабиринту!

Отечественные игры

NOCTURNAL ILLUSION (c) 1997 Excellent, (c) 2005 SamSTYLE. Портитование на ZX хентай-похождения продолжается. Благодаря усилиям Sam STYLE Вы сможете окунуться в атмосферу таинственности, переплетенную с большим количеством сюжетных линий, завершающихся

несколькими непохожими концовками, от души одобренную открывенными сценами.

Главный герой после ужасающей бури случайно оказывается в странном доме, да к тому же не один, а в компании не менее странных персонажей. Возвращенный к жизни печальной хозяйкой особняка, он сталкивается с множеством тайн, окутавших это странное место.

Почему по ночам из сарая раздаются звериный рев и женский смех? Что скрывает старый темный подвал? Почему ворота не выпускают никого? Эти и многие вопросы встанут перед вами через несколько секунд игры, затягивая все больше и больше. У каждого персонажа игры свой «скелет в шкафу», своя тайна, свое горе; помочь им можете только вы. Хозяйка дома знает очень многое, однако информацией делится не очень охотно, дает туманные намеки и сразу же уходит.

Игра характерна, прежде всего, своей качественной двухэкранной графикой, лихо закрученным сюжетом и новой системой перемещения между локациями, благодаря которой можно попасть в противоположный конец дома всего за пару секунд.



RECOMMENDED

Пресса

Fantastic 101. Слухи, дошедшие до меня о том, что с 100 номера Александр Смирнов заканчивает выпуск своей научно-фантастическо-мистической газеты, не оправдались. Новый номер совсем недавно появился. Оформление несколько изменилось, процесс чтения тоже. Теперь перед вами будет сплошной текст, меню выбора статей нет, а сама газета разделена на две части. Несмотря на нехватку времени, автор планирует еженедельный выход газеты. Если вы хотите узнать больше, присоветуйтесь к миру загадок, то эта газета для вас.

FANTADROM 5. Луноход-1 предлагает Вашему вниманию новый выпуск литературно-художественного журнала «Fantadrom». В издании достаточное количество соответствующих статей, так что любителям подобных газет/журналов номер обязательно понравится.

Произошло воскрешение очередного спектрумовского издания - готовится к выпуску 12-й по счету номер газеты TargeT от Sinus'a и FlyFOXx'a. Еще один электронный «долгострой» никак не доводится до кондиции - газета ZX Time от DWT. Обещанные и несколько раз сдвигавшиеся сроки не были соблюдены, но мы надеемся, что все препятствия при подготовке газеты будут благополучно преодолены,

Новости с мягкого фронта

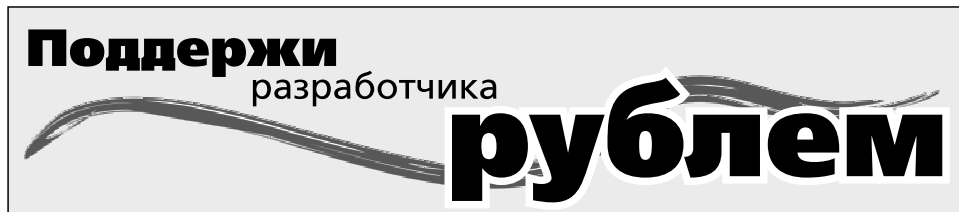
и все спектрумисты смогут насладиться чтением.

Системные программы

Turbo Word v0.1 (c) budder 2004. E-mail: budder@mail.ru Проект уже давно заморожен - последние изменения в него вносились больше года назад, поэтому версия эта только для ознакомления. Тем не менее, редактор обладает достаточным количеством возможностей для редактирования текстов, использует режимы 32, 42, 51, 64 символа в строке. Максимальный размер редактируемого файла 16 килобайт - по этому показателю он про-

игрывает тому же acedit'y. Имеется возможность указывать не только цвет ink, но и raret, однако непосредственной раскраски в редакторе пока нет. Для этого придется воспользоваться другим инструментом. Имеется достаточное количество глюков и недоработок, наиболее частые из них описаны в прилагающемся файле. Хотя при должном старании автора этот редактор мог бы составить конкуренцию подобным программам, но по разным причинам редактор так и не доведен до рабочей версии.

Обзор подготовил Антон ЯКОВЛЕВ



Читатель читателю

Chaos Construction 2005. Перспективы

Прочитал в 24 номере «Абзац» прогноз Димы Быстрова: «В этом году на СС'05 будет привезено больше работ, чем прислано, но организаторы примут из привезенных только половину. Произойдет большой конфликт». Захотелось все это прокомментировать.

Действительно, такое предположение могло возникнуть у Димы после посещения СС'04 в прошлом году, когда организация Спектрума была продумана, мягко говоря, весьма слабо. У организаторов (за сценой) стояло всего два компьютера, причем один из них - простой Спектрум 128 без наворотов. О том, как именно происходил прием работ, и почему многие из них были отвергнуты, я подробно описал в своей статье «Chaos Construction 2004: личные впечатления» - «Абзац» № 21.

В этом году главным организатором, отвечающим за Спектрум, являюсь я. Поэтому я хочу

вкратце написать, что будет на СС'05, и чем фестиваль в этом году будет отличаться от предыдущего.

Изначально, если вы помните, фестиваль Chaos Constructions возник как Spectrum party. Основной платформой был Спектрум, и основные участники и посетители также были спектрумистами. Позже фестиваль стал межплатформенным, на нем появился ПЦ. Но стали ли конкурсы ПЦ преобладать над конкурсами Спектрум платформы - большой вопрос. Ведь основное, что составляет содержание Chaos Construction - это творчество. А как обстоят дела с творчеством на ПЦ - думаю, знают все. У меня весьма негативное мнение по этому поводу.

Став Spectrum-организатором на СС'05, я поставил перед собой определенные задачи. Вот они в порядке приоритетов:

1. Показывать работы для Спектрума с реальных Спектрум-машин. Исключить использование

эмуляторов. (Здесь играют роль не только материальные факторы, такие, как наличие или отсутствие необходимых компьютеров - эмуляторы сейчас весьма хороши, и их в принципе можно было бы использовать для запуска работ, но также и факторы психологические. Все-таки, Спектрум - это платформа, и программы, написанные на реальном компьютере, должны запускаться и работать только на нем).

2. Обеспечить возможность приема работ в последний момент. (Думаю, ни для кого не секрет, что спектрумисты приносят работы всегда в последний день, за час-два перед началом фестиваля. Конечно, причины у всех разные, и не последнюю роль играет традиция. То, что организаторы СС'04 недостаточно хорошо продумали процедуру приема работ в день проведения фестиваля - их очень большой просчет).

Так что упрек Димы Быстрова был бы справедлив, если бы орга-

Читатель читателю ///

низация СС'05 оставалась на том же уровне, что и в прошлом году. Однако это совершенно не так! Подготовкой спектрум-части активно занимаются люди, которые на Спектруме далеко не новички: это BlackFox (музыкант, занявший ряд призовых мест на различных пати в номинации AY-music), Кирилл Фролов (железячник, кодер, великолепно знающий архитектуру, особенности и разновидности Спектрумов), а также ряд других людей. Соответственно, уже сейчас можно с уверенностью сказать, что различные аспекты Спектрума будут разработаны достаточно хорошо: это музыкальная часть (BlackFox), программная часть (Vega, BlackFox, FK0), аппаратная часть (FK0, KGB+, Chaos, Jam). Т.е. рабочие компьютеры (Спектрумы) будут полностью собраны, будут исправно работать, будет предусмотрен и подготовлен весь набор всевозможных системных программ, в которых может возникнуть необходимость: копировщики, редакторы, программы для показа, поисковики, командеры и т.д.).

Теперь пара слов о том, как все это будет выглядеть «изнутри», т.е. как все это будет организовано. Для приема работ будет выделен компьютер (Scorpion), оборудованный Спектрум-линком (связь Спектрума с PC через LPT-порт, (с) MAS), винчестером, CD-ROM'ом, 5,25" и 3,5" дисковыми. Слинкован Спектрум будет с PC-ноутбуком. Это позволит осуществлять прием работ с любых носителей: TR-DOS дискет, MS-DOS дискет, компакт-диск, flash-носителей, и т.д. Рядом будет находится другой Спектрум (Scorpion, HDD, GMX; возможно, также и дополнительно реальный Pentagon), с которых будет осуществляться оцифровка работ: все работы будут записаны с помощью DVD-камеры в AVI файлы (видео файлы), которые будут показываться с главной PC-компо машины. Т.е. фактически для про-

игрывания будут использованы Спектрумы, но это не будет реал-тайм показ: фестиваль Chaos Constructions уже выходит на достаточно высокий уровень, и позволить себе реал-тайм показ никто не может, во избежание накладок. Однако работать ваше творение будет на живом Z80, а не на эмуляторе-оболочке другой платформы. Процесс между приемом работы и его оцифровкой займет не больше 1 часа. Однако из этого вовсе не следует, что всем необходимо приносить работы в последний момент! Заранее присланная работа гарантирует вам, что накладок с ее показом не произойдет вообще никаких.

Естественно, в случае низкого уровня работы (если она, допустим, не пройдет преселект) эта работа показана не будет. Но, думаю, это не причина для упреков.

Преселект будет осуществлять организаторами-спектрумистами и специально привлеченными для этого спектрумистами не организаторами. В Music Compo преселект будут осуществлять, в частности, BlackFox, Vega, NewArt. В Grafix Compo - Demonic, NewArt. Demo Compo и Intro Compo - все организаторы-спектрумисты. Поэтому оценка на преселекте будет объективной. Основной принцип - не выносить на конкурс абсолютно безграмотные работы: в Music - набор нот, в Grafix - набор линий и точек, в Demo - бегущую строку на Бейсике. Отбор не будет строгим и жестким: если вы начинающий кодер/музыкант/художник, смело присылайте свою работу на конкурс!

Обратите внимание! В отличие от предыдущих фестивалей, в этом году нет привязки к конфигурации Pentagon 128: теперь ваша работа может быть написана и заявлена для показа с любой разновидности Спектрума: Pentagon 128, Scorpion 256, KAY и т.д. Это объясняется не только увеличившимися возможностями организа-

торов, но также может рассматриваться и как попытка привлечь к участию в конкурсах любых спектрумистов: сейчас, когда новые Спектрумы практически не выпускаются, не каждый человек сможет найти в своем городе реальный Pentagon, чтобы написать свою программу специально для него. Поэтому работа будет запускаться в той конфигурации компьютера, какая использовалась для ее создания. Это же относится и к Music Compo: вы указываете расположение AY каналов, в котором музыка должна проигрываться (ACB, CAB и т.д.), и мы проигрываем музыку именно таким образом.

Кроме организации приема и показа работ оргкомитет планирует ряд дополнительных инициатив. В частности - организация realtime-конкурсов для Спектрума. Планируется Realtime Music-compo, Realtime Coding-compo. Для их организации будет использоваться порядка пяти-восьми Спектрумов. Для музыкантов будет представлен ряд инструментов и редактор Pro Tracker. Участникам Coding Compo будет выдан набор справочных материалов, таких, как описание ассемблера из книги Ларченко и Родионова, таблица соответствия мнемоник ассемблера и их кодов, описание процедур TR-DOS и переменных Бейсика, таблица тактов команд ассемблера. Единственное условие для всех участников - работа на реальном Спектруме с 58-клавишной клавиатурой.

Также пара Спектрумов, объединенных в сеть с помощью Visomn-протокола, будет представлена на выставке компьютеров, где все желающие смогут увидеть их работу и пообщаться с ними сами.

Поэтому долг и обязанность любого спектрумиста приехать на этот фестиваль, а если вы художник, музыкант или программист - выставить на конкурс свою работу. Мы ждем вас в

Читатель читателю ///

Петербурге, в августе 2005 года!

Если вы решили стать участником фестиваля, то можете присылать свои работы по следующим адресам:

E-mail: spectrum@cc5.org.ru

FidoNet: 2:5030/1512

ZXNet: 500:812/19

Не забудьте указать свое имя

(ник), название группы (если вы входите в группу), название работы, номинация конкурса, на который вы выставляете свою работу.

Фестиваль Chaos Constructions будет проводиться в Санкт-Петербурге 20 и 21 августа, стоимость входного билета будет около 150-350 рублей. Место проведения фе-

стиваля - Ленинградский Дворец Молодежи (недалеко от станции метро «Петроградская»). Более подробную информацию о ходе подготовки к фестивалю вы можете получить, зарегистрировавшись на сайте <http://cc5.org.ru>. Мы ждем вас!

Влад СОТНИКОВ

Мысли вслух ///

Почему нет нового железа или Исповедь производителя

Давно хотел написать эту статью, т.к. хочу показать вам ситуацию с точки зрения разработчика спектрумовского железа. Оговорюсь, я не © Нето и не умею писать красиво с математическими выкладками и прочим, поэтому буду писать просто и открыто. Разумеется, это не открытое письмо, а статья, в которой я постараюсь отразить все, с чем пришлось столкнуться как производителю железа. У внимательного читателя наверняка возник вопрос, почему я веду диалог от одного лица? Этому вы найдете объяснение в статье.

«Давай производить железо»

На очередной вечеринке, пьянке или просто в беседе вдруг возникла такая идея. Круто! Все покивали головой, согласились и решили «что будем». Дело осталось за малым - организация производства. Пусть вы решительный человек и все-таки отважились на эту затею. Что вам для этого нужно:

1. Конструкторская документация. Естественно надо знать, что производить. И уж тем более, если вы решили разработать все с нуля (с чем придется столкнуться, я расскажу чуть ниже).

2. Подготовка файлов для производителя печатных плат. Те, кто думает что это просто, сильно заблуждаются. Дай бог, производитель разберется в вашем РСВ файле (файле разводки), а ведь ему может понадобиться еще файл сверловки, маски и много чего еще.

3. Производство печатных плат. Первое, что вы откроете для себя это ЦЕНА подготовки производства. ЦЕНА - написано большими буквами, потому что эта цена достаточно высока, как правило порядка 1000 рублей за два шаблона (для двухслойных плат). Если вы решите делать «зеленые платы», то прибавьте еще столько же для шаблонов маски. Ну и производство плат с маской добавляет к их конечной стоимости от 15% до 30%.

4. Проверка плат. Спектрумисты чрезвычайно небогатые люди и не могут себе заказать печатные платы с электрической проверкой (это увеличивает их стоимость). Поэтому запасаемся лупой, хорошим фонариком и, лазаем по плате за платой, просматривая замыкания и разрывы. А они обязательно будут, ибо вместо дорогого МФ4 для производства плат мы использовали дешевый молдавский текстолит.

5. Радиодетали. Вот платы получены, и вы бежите в ближайший радиомагазин за деталями. И что мы видим, так необходимая кр555ир23 стоит 10 рублей. Уже с меньшим энтузиазмом едем на «Митинский» рынок и покупаем кулек «неликвидов».

6. Сборка. Наступает самый интересный момент. Первая плата собирается за три часа. Гордо присматриваешься и обнаруживаешь, что пара микросхем припаяна другой стороной, у одной загнулась нога, а если еще и плата без маски, то олово капнуло на самое скопление дорожек. Следующие пару-тройку дней вы боретесь с браком.

7. Прошивка ПЗУ. Ну конечно, ловко собрали «навороченный программатор» из журнала «Радио». Включаем - не работает. Читаем внимательно статью, да этот программатор только для PIC/AVR, жутко ругаемся. Едем опять на Митинский рынок - покупаем BiDiPro за 1000 рублей или TurboV6.

8. Наладка. Первое включение и тишина. На экране монитора в лучшем случае разноцветный мусор, в худшем развертки нет совсем. Что делать? Пытаемся что-то найти с помощью вольтметра и, естественно, ничего не находим. Очередные сапоги жене откладываются «на потом», едем покупать в ближайший НИИ осциллограф С2-xxx. Лезем в плату и обнаруживаем, что КР555КП11 Бакинского завода 1993 года выпуска, проданные улыбочивым азербайджанцем, не выдают сигналов, а вот этот ЛАЗ на выходе дает какую-то чушь. Тут-то и помянешь всю «неликвидность» деталей и настырно начнешь их ликвидировать с платы. На это ухо-

Мысли вслух

дит еще тройка-пятерка дней.

9. Документация. Все работает. Ну что еще этим пользователям надо? Разве не понятно, что в этот разъем нужно воткнуть, а что в этот? Придется писать документацию. Понятную и красивую - страниц на 50. А чтобы ее распечатать, откладываем покупку кофточка для жены и покупаем лазерный принтер.

10. Шлейфы. Ну как же можно было забыть такую мелочь? Покупаем в «Буром медведе» необходимые бухты проводов и разъемов. Жена осталась без похода в ресторан.

11. Реклама. Все отлично. Теперь можно писать Саше в газету, пусть дает рекламу.

Конечно это очень примитивное обозрение цикла производства.

Производство именно нового железа

Подозреваю, что мало кто из читателей представляет себе этот процесс. Это очень трудоемкий, ресурсоемкий и длительный процесс, который подразумевает следующий алгоритм:

1. Постановка задачи и техническое задание. Собственно определяется, какими характеристиками должно обладать устройство и, в каких рабочих условиях функционировать.

2. Схемное проектирование. Проектирование функциональных блоков и собственно реализации схемы устройства.

3. Моделирование. Проверка функциональности схемной реализации. В основном с помощью специальных программных пакетов.

4. Опытное изделие. Изготовление опытного изделия. Проверка функционирования, доработка схемной реализации и переход к пункту 2. Таких итераций может быть достаточно много, зависит только от профессиональности разработчиков.

5. Подготовка окончательного пакета документации для производства. Собственно подготовка производства.

6. Производство.

7. Проверка соответствия изделия пункту 1. При несоответствии изделия, обычно разработчики или сами доводят изделие до работоспособности (в виде «соплей» из проводов, например), либо им вклеивают строгача и переходят на пункт 2.

8. Техническое описание изделия и подготовка пакета документации для потребителя.

9. Менеджмент. Собственно реклама и прочие процедуры раскрутки изделия.

Как видите - разработать новое изделие по трудозатратам в разы больше, чем суммарная стоимость компонент изделия. Поэтому всегда проще взять уже готовое изделие и доработать его, чем изготавливать что-то с нуля.

К примеру, разработка Спринтера 2000 и всего программного обеспечения к нему, фирме «Петерс-Плюс» обошлось в несколько тысяч долларов, но проект оказался провальным, т.е. принес только убытки. Поэтому я решил производить уже известное изделие, на которое у меня есть вся необходимая документация.

Друзья

Вернемся к тому моменту, когда группа сотоварищей рьяно решила все же делать железо. Пусть у нас уже есть конструкторская документация на ИЗДЕЛИЕ. Вы как настоящий организатор, предлагаете «давайте скинемся» на производство плат. Как вы думаете, сколько «друзей» согласятся? Правильно - в лучшем случае половина. Ну разве это проблемы вот 10000 рублей на производство собраны, оплачиваем - получаем платы.

Накрывается стол, раскладываются платы, все радуются. Вы, как организатор, предлагаете: «давайте скинемся на детали...» Исход известен, детали уже оплачиваются в основном из вашего кармана, и кармана, ну может, самого лучшего друга. Жена уже смотрит недобро.

А что дальше? Правильно: осциллограф, программатор, принтер, паяльник (паяльная станция). Как вы думаете, на чьи деньги все это куплено?

Ну ладно, деньги достались из заначки, ведь потом все отобьем.

Займемся сборкой. Приглашаются друзья: «Ну что мужики, берем каждый по плате...». Один: «я не умею». Второй: «у меня руки трясутся». Третий возьмется. Вы, счастливый, что кто-то взялся, даже не подозреваете, чем это обернется.

Через неделю (а то и две) «помощник» принесет вам то, что осталось от платы и деталей. Теперь вы понимаете, почему я начал писать от единственного лица? Среднестатистический человек работает только за выгоду, или за другие блага. Те, кто работают за идею, уже давно кончились (во всех смыслах этого слова).

Ладно, с помощью разобрались. Давайте попросим, кого-нибудь сделать документацию. Результат опять же предсказуем - делать ее будете вы. Не буду называть имени, но один зарубежный спектрумист пообещал мне сделать перевод на английский. Хватило его ровно на один восьмистраничный документ. Оставшиеся пятидесяти двух страничный и сорока восьмью страничный документы достались вашему покорному слуге. А заказ-то из-за границы уже оплачен...

Свободное время

Производя железо для спектрумистов, думать, что на нем можно заработать, крайне утопичная мысль. И если вы не сын Рокфеллера или хотя

Мысли вслух

бы президента «Банка Возрождения», то всякая работа по производству происходит в свободное от основной работы время. А это значит - в ваше личное время. Пусть вы гений электроники и асс паяльной станции, и вам удастся собрать изделие за три часа, и настроить за столько же. Как правило, максимум свободного времени в рабочий день это полтора-два часа. Т.е. сборка изделия займет примерно неделю (к этому времени добавляется распечатка документации, изготовление шлейфов и прочее). Вы скажете: «А как же выходные?». А кто из семейных решится в выходные не пойти с детьми в зоопарк, или с женой в кино, или на дачу? Дай бог, в выходные будут те же 3-4 часа свободного времени. За которые, кстати, надо сходить на почту и отправить жаждущим пользователям изделие, или получить от них деньги (отстояв длинную очередь).

Так что если за первое время вам удалось пару ночей полноценно поспать, от недостатка внимания от вас не ушла жена к соседу, то вам повезло.

Расходы и прибыль

Начав работы, вы мечтаете вернуть свои расходы. Давайте сразу их прикинем. Я приведу пример производства АТМ2 Турбо. Благо у меня уже были осциллограф, паяльник, программатор и принтер, поэтому я буду учитывать только расходы именно на производство.

1. Печатная плата. Получилось примерно 600 рублей за штуку.

2. Радиодетали. Суммарно 1500 рублей на одно изделие.

3. Расходники. CD диски, бумага, скрепки, олово, флюс, упаковка - 50 рублей.

Т.е. суммарно 2150 рублей.

Пусть вы решили сделать партию из 15 плат. Тогда вам нужно $2150 \times 15 = 32250$ рублей. Сразу задайтесь вопросом - готовы вы на такие расходы?

Теперь посчитаем прибыль. Пусть вы опять же гений и умудряетесь за месяц сделать четыре платы. За сборку платы я беру 600 рублей, за месяц получается 2400 рублей. Общая прибыль со всех реализованных плат компьютеров (15 штук) 9000 рублей, а это даже меньше 30% от затраченных средств.

Динамика продаж

Вот вы все подготовили, сделали плату, самую нужную спектрумистам. Сделали рекламу в Интернете и в «Абзаце», показали платы на СС и КидСофте. Начали ее продавать. Я опять же не © Немо, и не умею красиво описывать теорию. Но очевидные факты приведу, так как секрета я из них создавать не вижу смысла, а остудить горячие головы нужно.

Правило 1. Чем дороже изделие, тем оно хуже будет продаваться. Причем это совершенно не зависит от его назначения. Спектрумисту заподло тратить деньги на новый АТМ, когда на помойке есть много «крутых Пентиумов» в два раза дешевле.

Правило 2. Из тех, кто дает предварительную заявку на изделие, покупают максимум 15% человек. Это особенность чисто Российской и объяснения ей я пока не нашел.

Правило 3. Реклама может дать положительный результат, а может и отрицательный.

Привожу данные о продажах:

Изделие	Продано	Стоимость	Период
АТМ в сборе	6	2950	октябрь, 2004 г.
АТМ голая плата	9	700	октябрь, 2004 г.
PAL-кодер	12	350	ноябрь, 2004 г.
IDE-контроллер	27	300	август, 2004 г.

Таким образом, ожидать, что ваше изделие будут «отрывать с руками», совершенно не стоит.

«Благодарные» спектрумисты

А теперь самое интересное. Вы раскрываете изделие, а вам приходит в ответ примерно следующее, постараюсь изложить основные фразы и прокомментировать:

1. «Да у вас не поддерживается Скорпион, Профи, Гигаскрин, Пентагон, Кворум... (нужное подчеркнуть)». Естественно не поддерживается, а если и поддерживается, как, например, в «Спринтере 2000», то вы все равно не купите. Так что сидите в своем эмуляторе и не трогайте производителя, который старается, но, слава Богу, не для вас.

2. «А что так дорого, вот Пентиум всего за ...». А Пентиуму вашему сколько лет? А это новое изделие - только что собранное. И стоит оно практически по себестоимости (читаем выше). А если вам нужно дешево, идем на помойку и ищем Пентиумы.

3. «А почему бы не приделать к ... поддержку PCI, AGP, Serial-ATA, USB, FireWire вертикального взлета и посадку...». Обязательно приделаем, высылайте чек на пару-тройку десятков тысяч долларов и все будет. Даже автоматический слив из-под вас, когда вы все это включите.

4. «Я очень заинтересовался вашим изделием, но у меня сейчас кризис, ушла жена, злой дядя отнял все деньги, отдал все на поддержку Ирака, проиграл в Казино, пропил на СС04, поэтому, пожалуйста, почините мне мой Пентагон, Кай, Скорпион». Извините, времени нет, меня шофер ждет, чтоб в Казино вести, а там будем помогать Ираку.

Мысли вслух ///

5. «Ой, да ваш ... хуже калькулятора, вот мой Пентиум/Ксеон/Атлон гитарцы гоняет». А еще он гоняет «Кваку» и ваши денежки к буржуям. А вы стреляйте дальше и не мешайте людям творить. Ведь творчество для вас непостижимо, вы только потребитель.

Вот неполное собрание выдержек из писем и ответы на них. Если понравилось, буду пополнять в следующих выпусках «Абзаца».

Выводы

Производитель железа для спектрумистов - то же самое, что и сумасшедший трудоголик без свободного времени. Побыв какое-то время в такой шкуре, возможно, сразу бросишь все это, и будешь вспоминать как страшный сон. Откровенно говоря, я так бы и сделал, но, хочется довести работу до конца. И есть еще маленький нюанс, это счастливые лица тех, кто получил, наконец, изделие и может теперь реализовать все забытые Пентиумом желания...

Я их вижу, и это мне дает силы и желание сделать что-то еще. А так же появилось самое главное - желание ТВОРИТЬ дальше.

Спасибо

- Тимонину Максиму - за все работы по составлению и составлению документации, прошивок, программатора, оптимизм и прочее...

- Каримову Камилу - за хорошие идеи и помощь с прошивками;

- Миндлину Роману - за поддержку в трудную минуту, и как основному потребителю всех наших устройств;

- Илясову Евгению - за поддержку, за помощь в распространении рекламы;

- Шушкову Александру - за рекламу и взаимную поддержку;

- Быстрову Дмитрию - за рекламу и опять же за поддержку.

Спасибо вам, что дочитали эту статью до конца.

Я конечно рано или поздно заброшу сборку компьютеров, потому что это меня сильно утомляет, и буду предлагать только голые платы с документацией или набор-конструктор. Наверно это будет сделано со второго квартала 2005 года (когда у меня закончатся закупленные детали). К тому же это освободит время для разработки новых устройств.

Роман ЧУНИН

Читатель читателю ///

Особенности программирования CD-ROM на Спектруме

1. Введение

В газете «ZX News» № 54 мной была опубликована статья об особенностях программирования винчестера на Спектруме (обновленная версия была напечатана в газете «Абзац» № 21, 2004 г.). Также мной было в 2002 году написано продолжение, статья о программировании CD-ROM. Однако из-за того, что эта статья была опубликована лишь в Интернете (электронный вариант газеты «ZX News» № 55 так и не вышел на Спектруме), она не получила широкого распространения среди тех, кому она была бы действительно интересна, т.е. пользователям ZX Spectrum.

Публикуемый здесь вариант - исправленная и дополненная версия того, что было опубликовано на сайте газеты «ZX News» в Интернете.

Появление CD-ROM на Спектруме не было случайным - предпосылкой было создание IDE-контроллеров. Это и SMUC на Scorpion'e, и HDD-контроллер © Nemo. И хотя основной целью создания этих устройств было подключение винчестера, тем не менее «побочным» результатом стала возможность работы с CD-ROM. Мной совместно с

Павлом Васильевым (POL) была написана программа, позволяющая читать данные с компакт дисков. Это CD_Walk, которую вы можете найти на специально созданном для программы Кириллом Фроловым сайте: <http://cdwalk.narod.ru>. Программа поддерживает оба контроллера (SMUC и NEMO). Так что все написанное здесь, служит единственной цели - облегчить людям написание подобных программ, но никак не может претендовать на документальное описание устройства. Так что могут встретиться погрешности и неточности. Тем не менее, информация, приведенная здесь, должна помочь людям, решившим сделать поддержку CD-ROM'a в своих программах.

В отличие от моей предыдущей статьи, рассчитанной на начинающих пользователей, здесь я решил не объяснять лишний раз элементарных вещей и понятий, чтобы ограничить текст в объеме. Я предполагаю, что читатель уже знаком с моей предыдущей статьей (газета «Абзац» № 21 или «ZX News» № 54) и владеет базовыми навыками программирования на ассемблере и программирования винчестера. Назначение регистров и многие подпрограм-

Читатель читателю ///

мы, используемые мной здесь, вы можете найти в моей статье о программировании винчестера.

2. Немного о CD-ROM'e

По своим особенностям CD-ROM сильно отличается от винчестера. Во-первых, CD-ROM можно отнести к разряду Read-only устройств (пишущие CD-ROM мной здесь не рассматриваются в связи с ограниченным объемом информации об этих приводах. Возможно следующая версия CD_Walk, поддерживающая запись компакт-дисков, послужит основой для написания следующей статьи, посвященной данной тематике). CD-ROM является сейчас одним из популярных внешних носителей информации, в основном благодаря дешевизне компакт-дисков и удобству использования его в качестве архива.

CD-ROM является внешним носителем информации. Он состоит из CD-проигрывателя, куда вставляются компакт-диски (объем дисков колеблется от 640 до 700 мегабайт и выше), а также еще одной важной части - трей. Это платформа, на которую кладутся компакт-диски. Она является подвижной частью устройства, которой можно управлять как вручную, так и программно.

Как и винчестер, CD-ROM может быть как Master, так и Slave устройством. Однако принцип подачи команд несколько отличается от того, который использовался с винчестером, и был описан мной в предыдущей статье. Для программирования винчестера использовался набор регистров (или портов), каждый из которых выполнял какую-либо определенную роль (регистр команд, регистр состояния, регистр данных), и команды подавались через регистр команд, причем набор этих команд был ограничен. Совсем иначе дело обстоит с CD-ROM. Команда подается через пакет данных, состоящий обычно из 12 байт. Этот стандарт получил название ATAPI (ATA Packet Interface - Пакетный протокол ATA). В Регистр Команд подается команда «запись данных», и затем через Регистры Данных CD-ROM'у передается пакет из 12 байт. Причина этих сложностей состоит в том, что набор команд и регистров ATA не подходит для некоторых командных структур CD-ROM. Поэтому устройства ATAPI поддерживают только минимальный набор традиционных команд ATA.

3. Форматы компакт-дисков

3.1. «Красная книга»

Изначально компакт-диски были созданы фирмами Philips и Sony для воспроизведения звуковой информации. Сейчас это обычные аудио-диски, которые можно воспроизвести на любом CD-

проигрывателе. Формат данных определен как «Красная книга» и выглядит следующим образом: на диске находится некоторое количество звуковых дорожек (треков). Трек - это, как правило, одна песня. Дорожка, в свою очередь, делится на сектора, которые являются 1/75 секунды по длине и содержат 2352 байта данных в звуковой форме.

3.2. «Желтая книга»

Позже эти же компании-производители представили другой стандарт дисков, известный как «Желтая книга». В нем была предоставлена возможность сохранения на диске информации, не являющейся audio. Был пересмотрен формат сектора: теперь 2352 байта звукового сектора воспринимались следующим образом:

- 12 байт синхронизации;
- 4 байта информации о головке;
- 2048 байт данных пользователя;
- 288 байт исправления ошибки.

Это основной стандарт. Существуют еще стандарты «Зеленая книга», «Оранжевая книга», но все они являются вариацией этих двух основных стандартов. Кроме того, существует стандарт ISO-9660, который изначально предназначен для хранения данных и имеет собственную файловую структуру. В нем не предусмотрено хранение аудио-информации. При этом нужно иметь в виду, что современные CD-ROM'ы самостоятельно определяют стандарт, в котором записан диск, и работают либо с аудио, либо с данными. Во втором случае нам нужно считывать 2048 байт данных. Это и есть длина сектора компакт-диска и минимальная длина считываемых данных.

3.3. ISO-9660

Стандарт ISO-9660 определяет структуру хранения файлов на диске. Она будет подробно описана далее в главе 12 - «Структура файловой системы диска». На самом деле, это довольно простая для понимания файловая структура. Диск делится на сектора по 2048 байт. В 16 секторе находится информация о диске и местоположении корневого каталога. Далее следуют все каталоги компакт-диска, и затем идут файлы. Причем информация в файлах располагается последовательно, сектор за сектором. Эта структура очень напоминает структуру дисков TR-DOS.

4. Подпрограммы для контроллера

Еще раз приведу порты SMUC и контроллера (с) NEMO, которые соответствуют регистрам IDE-

Читатель читателю ///

устройства. Так же следует напомнить, что порты SMUC вызываются из-под TR-DOS.

Название регистра	Контроллер SMUC	Контроллер (с) NEMO
Регистр команд/состояния	#FFBE	#F0
Регистр накопителя/головки	#FEBE	#D0
Регистр цилиндра (старшая часть)	#FDBE	#B0
Регистр цилиндра (младшая часть)	#FCBE	#90
Регистр номера сектора	#FBBE	#70
Регистр счетчика секторов	#FABE	#50
Регистр ошибки/доп. возможностей	#F9BE	#30
Регистр данных (младшая часть)	#F8BE	#10
Регистр данных (старшая часть)	#D8BE	#11

4.1 Подпрограммы для контроллера SMUC

Здесь приведены подпрограммы для работы с контроллером SMUC. В принципе, эти подпрограммы соответствуют тем, что были приведены в статье про винчестер, здесь они лишь более систематизированы.

```
;ЗАПИСАТЬ ЗНАЧЕНИЕ В ПОРТ.
;IN: [BC] - НОМЕР ПОРТА
; [A] - ЗНАЧЕНИЕ
OUT_A LD IX,#3FF0
      PUSH IX
      JP #3D2F
```

```
;СЧИТАТЬ ЗНАЧЕНИЕ ИЗ ПОРТА.
;IN: [BC] - НОМЕР ПОРТА
;OUT: [A] - ЗНАЧЕНИЕ
IN_A LD IX,#3FF3
     PUSH IX
     JP #3D2F
```

```
;ОЖИДАНИЕ ОСВОБОЖДЕНИЯ УСТРОЙСТВА.
NO_BSY LD BC,#FFBE
      CALL IN_A
      RLCA : RET NC
      JR NO_BSY
```

```
;ОЖИДАНИЕ ГОТОВНОСТИ ПЕРЕДАЧИ ДАННЫХ.
WAIT_DRQ LD BC,#FFBE
        CALL IN_A
        BIT 3,A
        RET NZ
        JR WAIT_DRQ
```

```
;СМОТРИМ, НЕ ПРОИЗОШЛА ЛИ ОШИБКА.
NO_ERROR LD BC,#FFBE
        CALL IN_A
        RRCA
        RET
```

```
;ВЫБОР УСТРОЙСТВА.
;IN: A=#00 - MASTER.
; A=#B0 - SLAVE
SEL_DRIVE LD BC,#FEBE
        CALL OUT_A
        RET
```

```
;ЗАПИСЬ ЧИСЛА В РЕГИСТР ЦИЛИНДРА.
HL_TO_LEN LD BC,#FCBE
          LD A,L
          CALL OUT_A
          LD BC,#FDBE
          LD A,H
          CALL OUT_A
          RET
```

```
;ЧТЕНИЕ ЧИСЛА ИЗ РЕГИСТРА ЦИЛИНДРА.
LEN_TO_HL LD BC,#FCBE
          CALL IN_A
          LD L,A
          LD BC,#FDBE
          CALL IN_A
          LD H,A
          RET
```

4.2 Подпрограммы для контроллера (с) NEMO

Для тех, кто пользуется контроллером винчестера (с) NEMO, также приведены отдельные подпрограммы. Этот контроллер удобен тем, что для его работы не нужно отдельное ПрофПЗУ, как для контроллера SMUC на Scorpion. Контроллер (с) NEMO достаточно вставить в системный разъем любого Spectrum-совместимого компьютера. Кроме того, для обращения к нему не нужно каждый раз вызывать ПЗУ TR-DOS - порты здесь открыты, что, в свою очередь, увеличивает скорость работы с CD-ROM'ом. И последнее - схема контроллера настолько проста, что его вполне можно собрать самостоятельно.

```
;ЗАПИСАТЬ ЗНАЧЕНИЕ В ПОРТ.
;IN: [BC] - НОМЕР ПОРТА
; [A] - ЗНАЧЕНИЕ
OUT_A OUT (C),A
      RET
```

```
;СЧИТАТЬ ЗНАЧЕНИЕ ИЗ ПОРТА.
;IN: [BC] - НОМЕР ПОРТА
;OUT: [A] - ЗНАЧЕНИЕ
```


Читатель читателю ///

```

IN_A      IN (C),A
          RET

;ОЖИДАНИЕ ОСВОБОЖДЕНИЯ УСТРОЙСТВА.
NO_BSY    LD BC,#F0
          CALL IN_A
          RLCA
          RET NC
          JR NO_BSY

;ОЖИДАНИЕ ГОТОВНОСТИ ПЕРЕДАЧИ ДАННЫХ.
WAIT_DRQ  LD BC,#F0
          CALL IN_A
          BIT 3,A
          RET NZ
          JR WAIT_DRQ

;СМОТРИМ, НЕ ПРОИЗОШЛА ЛИ ОШИБКА.
NO_ERROR  LD BC,#F0
          CALL IN_A
          RRCA
          RET

;ВЫБОР УСТРОЙСТВА.
;IN: A=#00 - MASTER.
;      A=#B0 - SLAVE
SEL_DRIVE LD BC,#D0
          CALL OUT_A
          RET

;ЗАПИСЬ ЧИСЛА В РЕГИСТР ЦИЛИНДРА.
HL_TO_LEN LD BC,#90
          LD A,L
          CALL OUT_A
          LD BC,#B0
          LD A,H
          CALL OUT_A
          RET

;ЧТЕНИЕ ЧИСЛА ИЗ РЕГИСТРА ЦИЛИНДРА.
LEN_TO_HL LD BC,#90
          CALL IN_A
          LD L,A
          LD BC,#B0
          CALL IN_A
          LD H,A
          RET

```

Далее по тексту (для подпрограмм определения наличия CD-ROM'а и чтения данных) пользователям HDD-контроллера (с) NEMO нужно замкнуть регистры (порты) контроллера SMUC соответствующими портами контроллера (с) NEMO.

5. АТА-команды

Это вспомогательные команды, которые передаются в CD-ROM через Регистр Команд, а не

через пакет. Таких команд всего 4:

1. **#A0** - команда передачи АТАPI пакета. Подробнее ее использование будет рассмотрено ниже.
2. **#08** - Программный сброс. Выполняется полная инициализация CD-ROM'а.
3. **#A1** - Идентификация CD-ROM'а. Аналогична команде **#EC** для винчестера.
4. **#EC** - Воспринимается как ошибка, но в Регистре Цилиндра устанавливается значение **#14EB** - признак АТАPI устройства.

6. Определение наличия CD-ROM'а

Чтобы определить наличие CD-ROM'а, необходимо выполнить следующие действия:

1. Выбрать устройство Slave/Master.
2. Посмотреть отсутствие сигнала BSY. Если занято, значит, устройства нет. (Обратите внимание, чтобы перед этим CD-ROM не выполнял никаких ваших команд, иначе флаг BSY также может быть установлен).
3. Подать команду общего сброса. Хотя можно обойтись и без этого.
4. В Регистр Цилиндра записать число, отличающееся от **#14EB**. Для простоты можно записать 0 (ноль).
5. В Регистр Команд подать АТА команду **#EC** и подождать пару прерываний.
6. Прочитать Регистр Цилиндра. В нем должно находиться число **#14EB**. Если это так, значит, подключен CD-ROM. В противном случае это может быть винчестер. При этом нужно учитывать, что в Регистре Ошибок CD-ROM сообщит об ошибке. Это нормально.

Здесь я приведу подпрограмму, которая определяет наличие CD-ROM'а. Хочу заметить, что CD-ROM на Спектруме - это как правило Slave устройство, находящееся рядом с винчестером, и проверять его наличие я рекомендую в таком порядке: вначале Slave, затем Master.

```

;IN: A=#00 - MASTER.
;      A=#B0 - SLAVE
CD_INI    CALL SEL_DRIVE
          LD BC,#FFBE ;#F0 ДЛЯ NEMO
          CALL IN_A
          INC A
          JP Z,NO_CDROM
          LD HL,0
          CALL HL_TO_LEN
          LD BC,#FFBE ;#F0 ДЛЯ NEMO
          LD A,#EC
          CALL OUT_A
          CALL NO_BSY
          CALL LEN_TO_HL
          LD BC,#EB14
          OR A

```

Читатель читателю ///

```
SBC HL,BC
JP NZ,NO_CDROM
RET
```

```
NO_CDROM ...
```

Теперь мы убедились, что у нас подключено ATAPI устройство. Однако для полной уверенности не лишним будет проверить, точно ли это CD-ROM. Для этого нужно посмотреть 1-й байт в идентификационной таблице CD-ROM'a (передаваемой по команде #A1).

Чтобы определить, какой диск вставлен в CD-ROM (диск с данными или аудио-диск), необходимо выполнить подпрограмму проверки, приведенную в конце главы 9.11

7. Идентификационная таблица CD-ROM'a

Идентификационная таблица - это один сектор данных, в котором находится информация о конкретном CD-ROM'e. Сектор вызывается ATA командой #A1. После передачи команды нам необходимо принять 2048 байт от CD-ROM'a. Обратите внимание! Данные в этой таблице, как и в аналогичной таблице винчестера, перевернуты! Сначала идет старший байт, затем младший. Вам необходимо поменять каждые два рядом стоящих байта местами. Ниже приведен пример такой подпрограммы:

```
;IN: [HL] - АДРЕС ДЛЯ ПРИЕМА ДАННЫХ.
; [BC] - КОЛИЧЕСТВО БАЙТ.
TRANS_IN PUSH BC
CALL NO_BSY
CALL WAIT_DRQ
POP BC
OR A
RR B
RR C
JR NC,$+3
INC BC
TRANS_IN1 PUSH BC
LD BC,#F8BE ;#10 ДЛЯ NEMO
CALL IN_A
LD (HL),A
INC HL
LD BC,#D8BE ;#11 ДЛЯ NEMO
CALL IN_A
LD (HL),A
INC HL
POP BC
DEC BC
LD A,B
OR C
JR NZ,TRANS_IN1
RET
```

Итак, чтобы считать идентификационную таблицу, нам нужно выполнить следующую подпрограмму:

```
CD_INITAB LD A,#A1
LD BC, #FFBE ;#F0 ДЛЯ NEMO
CALL OUT_A
LD HL,MY_BUFF
LD BC,#0200
CALL TRANS_IN
RET
```

И у нас в памяти находится идентификационная таблица. В ней находится следующая информация:

00, bit 0 - Длина пакета:

0 - 12 байт,

1 - 16 байт.

01, bit 0-4 - Тип устройства: 5 (%00000101) - CD-ROM.

20-39 - Серийный номер.

40-45 - Что-то от производителя

46-53 - Версия прошивки

54-93 - Название модели

94-95 - Что-то от производителя

Остальные параметры для нас не так существенны. Таким образом, чтобы убедиться, что у нас действительно подключен CD-ROM, а не другое ATAPI устройство, нам нужно считать в аккумулятор байт из таблицы со смещением 1, выполнить команду AND #1F и посмотреть, не равен ли он 5. Хотя вероятность подключения к Спектруму ATAPI-устройства, не являющегося CD-ROM'ом, весьма мала, но, тем не менее, советую эту проверку проводить.

8. Передача ATAPI-пакета

Это ни что иное, как непосредственное программирование CD-ROM. Команды, которые он воспринимает, передаются ему в пакете, состоящем из 12 байт. Первый байт пакета является кодом команды, а остальные 11 байт - параметрами команды. Для начала я приведу подпрограмму, которая передает 12-байтный пакет в CD-ROM. Она выполняет следующие действия: записывает в Регистр Команд ATA команду #A0 (передача ATAPI пакета) и передает через регистры данных 12 байт являющихся ATAPI пакетом.

Внимание! При передаче пакета нужно всегда указывать устройство, с которым мы работаем (Master/Slave).

```
;ПЕРЕДАЧА ATAPI ПАКЕТА.
```

```
;IN: [HL] - АДРЕС ПАКЕТА.
```

```
SEND_ATAPI PUSH HL
LD A,MASTER/SLAVE #00/#B0
CALL SEL_DRIVE
CALL NO_BSY
```

Читатель читателю ///

```
LD HL,#0800
CALL HL_TO_LEN
LD BC,#FFBE ;#F0 ДЛЯ NEMO
LD A,#A0
CALL OUT_A
POP HL
LD BC,12
```

```
;ПЕРЕДАЧА ДАННЫХ
;IN: [HL] - АДРЕС ДАННЫХ.
; [BC] - КОЛИЧЕСТВО БАЙТ.
TRANSFER_OUT
```

```
    PUSH BC
    CALL NO_BSY
    CALL WAIT_DRQ
    POP BC
    OR A
    RR B
    RR C
    JR NC,$+3
    INC BC
TRANS_OUT1 PUSH BC
    INC HL
    LD A,(HL)
    LD BC,#D8BE ;#11 ДЛЯ NEMO
    CALL OUT_A
    DEC HL
    LD A,(HL)
    LD BC,#F8BE ;#10 ДЛЯ NEMO
    CALL OUT_A
    INC HL
    INC HL
    POP BC
    DEC BC
    LD A,B
    OR C
    JR NZ,TRANS_OUT1
    RET
```

И пара слов о том, как должен выглядеть этот командный пакет. В своем описании я буду представлять его в таком виде:

#00 - «Пустышка».

```
DB 0
DS 11
```

Первый байт - это код команды. Остальные - параметры. Неиспользуемые байты я описываю командой ассемблера DS X (последовательность X нулей). Хотя неиспользуемые данные не анализируются, для совместимости их лучше устанавливать в 0.

Итак, сам пакет можно оформить таким образом:

```
AP_00 DB 0
      DS 11
```

И вызываться он будет так:

```
LD HL,AP_00
CALL SEND_ATAPI
...
```

9. Описание команд CD-ROM'a**Общие команды.****9.1. «Пустышка»**

Эта команда была приведена мной в качестве примера в предыдущей главе. Пакет состоит из 12 нулей. Эта команда не выполняет никаких действий. Однако она крайне необходима - дело в том, что в силу непонятных причин CD-ROM отказывается выполнять многие команды и сообщает об ошибке. В частности, это происходит непосредственно за сменой носителя (Master -> Slave или наоборот). Поэтому настоятельно рекомендую вам перед каждой командой вызывать «пустышку». Именно так я добился работоспособности своей программы CD_Walk на Спектруме. Иными словами, выполнение команды, описанной пакетом AP_?? должно выглядеть так:

```
LD HL,AP_00
CALL SEND_ATAPI
LD HL,AP_??
CALL SEND_ATAPI
...
```

9.2. Позиционирование**##01 - Установить головку на начало диска.**

```
DB #01
DS 11
```

Эта команда управляющая, и в своей программе я не использовал ее ни разу.

##2B - Позиционирование в формате MSF.

```
DB #2B
DS 2
DB МИНУТЫ
DB СЕКУНДЫ
DB ДОЛИ СЕКУНДЫ
DS 6
```

MSF (Minute/Second/Frame) - это один из способов позиционирования на диске. Оно определяется в минутах/секундах/долях секунды, которые прошли бы при чтении с начала диска со скоростью 1. Формат удобен при работе с аудио-дисками.

9.3. Управление треем**##1B - Управление треем.**

```
DB #1B
DS 3
DB FUNCTION
```

Читатель читателю ///

DS 7

FUNCTION:

- 0 - Войти в режим Sleep.
- 1 - Остановить проигрывание/чтение.
- 2 - Выдвинуть трей.
- 3 - Задвинуть трей.

Как вы видите, функции этой команды не ограничиваются одним управлением треем. Режим Sleep - это режим пониженного энергопотребления, диск останавливается или замедляет свое вращение до прихода команды чтения данных. Функцию с кодом 1 удобно использовать для остановки проигрывания мелодии. Управление треем осуществляется только в том случае, если он не заблокирован.

;<#1E - Блокировка трея.

DB #1E
DS 3
DB FUNCTION
DS 7

FUNCTION:

- 0 - разблокировать трей.
- 1 - заблокировать трей.

Команда блокирует действие команды #1B.

Итак, если трей не заблокирован (можно временно его разблокировать), с помощью этой подпрограммы мы можем его выдвинуть:

```
TRAY_OUT    LD A,2
             LD (AP_1B+4),A
             LD HL,AP_1B
             CALL SEND_ATAPI
             RET
```

А эта подпрограмма задвинет его обратно:

```
TRAY_IN     LD A,3
             LD (AP_1B+4),A
             LD HL,AP_1B
             CALL SEND_ATAPI
             RET
AP_1B       DB #1B,0,0,0,0,0,0
             DB 0,0,0,0,0,0,0
```

Информационные команды.

9.4. Чтение параметров изготовителя (Inquiry)

Здесь содержатся практически те же данные, что и в идентификационной таблице, вызываемой АТА-командой #A0.

;<#12 - Чтение строки параметров изготовителя.

DB #12

DS 3

DB ДЛИНА СТРОКИ.

DS 7

Длина строки обычно 36. Возвращаются следующие данные:

- +0 - тип устройства (CD = 5)
- +1 - bit 7: поддерживаются сменные носители
- +2 - версии ISO, ECMA и ANSI.
- +3 - 0, для совместимости со SCSI-2
- +4 - длина оставшегося блока
- +5 - резервные
- +7 - 0, для совместимости со SCSI-2
- +8 - строка изготовителя (там бывает "ATAPI")
- +16 - название продукции
- +32 - версия изделия

9.5. Получение общих параметров

Эта команда мной не изучена, поэтому привожу ее из описания без изменений и комментариев. Вам предстоит либо изучить ее действие самим, либо пропустить, поскольку она сообщает специфические сведения о CD-ROM'е и для Спектрума решающего значения не имеет.

;<#5A - Получить общие параметры.

DB #5A
DB 0
DB PAGE - ОПРЕДЕЛЯЕТ
ТРЕБУЕМЫЕ ПАРАМЕТРЫ,
СОСТОИТ ИЗ ДВУХ БИТОВЫХ
ПОЛЕЙ:

BITS 1..5 - НОМЕР ТРЕБУЕМОЙ
СТРАНИЦЫ ПАРАМЕТРОВ:
%00000001 - ПАРАМЕТРЫ ИСПРАВЛЕНИЯ
ОШИБОК

%00001011 - ОБЩИЕ ПАРАМЕТРЫ

%00001110 - АУДИО-УПРАВЛЕНИЕ

%00101010 - ПАРАМЕТРЫ УСТРОЙСТВА
(ТОЛЬКО ЧИТАЕТСЯ)

%00111111 - ВСЕ СТРАНИЧКИ

BITS 6..7 - БИТЫ ТИПА ТРЕБУЕМОЙ
СТРАНИЦЫ:

%00000000 - ТЕКУЩИЕ ЗНАЧЕНИЯ
%01000000 - ИЗМЕНЕННЫЕ ЗНАЧЕНИЯ
%10000000 - ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ
%11000000 - СОХРАНЕННЫЕ ЗНАЧЕНИЯ

DS 4

DW LENGTH; ДЛИНА ТАБЛИЦЫ

DS 3

Читатель читателю

Команда информационная, выдает соответствующую страничку параметров.

Общий заголовок:

00-01 - Длина всего блока (без первого слова)

02 - Состояние привода

03-07 - ?

Заголовок каждой страницы:

08 - Номер страницы из запроса

09 - Длина страницы

Страница #01 - исправление ошибок

10 - Параметр исправления ошибок

11 - Счетчик повторов чтения

Страница #0D - общие параметры

10 - ?

11 - Множитель таймера неактивности

12-13 - Число S единиц в M единице для формата MSF (60)

14-15 - Число F единиц в S единице для формата MSF (75)

Страница #0E - аудио параметры

10 - Параметр не исп., но изменяется

11-12 - ?

13 - =0 LBA всегда равен номеру сектора, старший бит указывает правильность следующего поля

14-15 - Число логических блоков в секунду для проигрывания. (как правило, не используется)

16 - мл. тетрада - биты выходного порта канала 0

17 - громкость канала 0

18 - мл. тетрада - биты выходного порта канала 1

19 - громкость канала 1

20 - мл. тетрада - биты выходного порта канала 2

21 - громкость канала 2

22 - мл. тетрада - биты выходного порта канала 3

23 - громкость канала 3

Страница #2A - параметры устройства (только чтение)

; биты присутствия/отсутствия функций

12 - bit 0 - проигрывание аудио

bit 1 - композитный аудио/видео поток

bit 2 - Digital out to port 1

bit 3 - Digital out to port 2

bit 4 - чтение секторов Mode 2 Form 1

bit 5 - - - // - - - - Mode 2 Form 2

bit 6 - Чтение многосессионных дисков

bit 7 - ?

13 - bit 0 - Чтение «Красной Книги» через команду Read-CD

bit 1 - Чтение «Красной Книги» with

accurate stream»

bit 2 - Чтение субканала

bit 3 - Поддержка деинтерливинга данных субканала

bit 4 - Поддержка «C2 error pointers»

bit 5 - Поддержка чтения ISRC

bit 6 - Поддержка чтения UPC

bit 7 - ?

14 - bit 0 - блокировка носителя

bit 1 - чтение статуса блокировки

bit 2 - Disk prevent jumper present

bit 3 - команда выброса носителя

bit 4 - ?

bit 5...7 - Тип загрузчика:

0 - Caddy

1 - Tray

2 - Pop-Up

3 - Reserved

4 - Ченджер с индивидуально меняемыми

дисками

5 - Картридж

6 - Reserved

7 - Reserved

15 - bit 0 - Раздельная регулировка каналов

bit 1 - Раздельный коммутатор каналов

bit 2 - Информация о наличии диска

16-17 - Максимальная скорость обмена в килобайтах

18 - ?

19 - Число градаций регулировки громкости

20-21 - Размер буфера в килобайтах

22-23 - Текущая скорость обмена

; - - - - иногда может отсутствовать - - - - -

24 - ?

25 - bit 0 - Digital out по фронту/спаду сигнала BCKF

bit 1 - LRCK индицирует левый/правый канал

bit 2 - данные в формате LSB/MSB

bit 3 - ?

bit 4 \ BCKs: 0 - 32

bit 5 | 1 - 16

bit 6 / 2 - 24

3 - 24 (I²S)

bit 7 - ?

Длина каждой из страниц может быть разной. Здесь описаны только те поля, которые более-менее стандартны. Если запрашиваются все странички (биты 0...5), то в выходном блоке будет присутствовать один общий заголовок и последовательно расположенные странички со своими заголовками.

Читатель читателю ///**9.6. Установка общих параметров**

Действие команды аналогично предыдущей, с той лишь разницей, что она не возвращает значения, а записывает их. Коды и формат страниц вы можете посмотреть из описания предыдущей команды.

;<#55 - Установка общих параметров.

```
DB #55
DB ?; БИТ 1 - СОХРАНЯТЬ
;B NVRAM (?)
DB PAGE; ТРЕБУЕМАЯ
;СТРАНИЦА ПАРАМЕТРОВ
DS 4
DW LENGTH; ДЛИНА ТАБЛИЦЫ
DS 3*
```

9.7. Код ошибки операции**;<#03 - Чтение состояния привода.**

```
DB #03
DS 3
DB #12; ДЛИНА ТАБЛИЦЫ
;ОШИБОК
DS 7
```

Затем нужно считать 18 байт данных, где второй байт будет кодом ошибки операции. CD-ROM возвращает следующие ошибки:

- 0 - бессмысленные данные
- 1 - повторная ошибка
- 2 - нет готовности
- 3 - ошибка среды
- 4 - ошибка оборудования
- 5 - неверный запрос
- 6 - обслуживание устройства
- 7 - защита данных
- 11 - команда прервана
- 14 - неверное сравнение

К сожалению, я не могу привести здесь полное описание значения каждой ошибки, так как подробно ими не занимался. Хотя обработка ошибок при программировании CD-ROM наиболее важна, чем при программировании винчестера. Дело в том, что при работе с компакт-диском CD-ROM довольно часто возвращает ошибку. Но это не всегда значит, что произошла ошибка в прямом смысле этого слова. Например, это может свидетельствовать о неготовности CD-ROM'a передавать данные из-за недостаточной скорости вращения компакт-диска (то есть, попросту говоря, диск еще не раскрутился).

Я приведу пример из моего непосредственного опыта написания копировщика для CD-ROM. Если какое-то долгое время не происходит обращения к диску, то CD-ROM переходит в режим пониженного энергопотребления. Начинает медленнее вращаться. И при обращении к диску, например, при попытке считать данные, он снимает флаг BSY,

однако скорости вращения диска еще недостаточно для выполнения операции. CD-ROM будет возвращать ошибку «2 - нет готовности» до тех пор, пока скорость вращения диска не позволит считать данные и передать их компьютеру.

Действия при чтении данных (сектора) должны иметь следующую последовательность:

1. Передаем пакет с командой чтения данных (описан ниже).
 2. Ожидаем снятия флага занятости BSY.
 3. Смотрим 0-й бит Регистра состояния (нет ли ошибки)
 4. Если нет ошибки, продолжаем чтение данных.
 5. Передаем пакет «03». Читаем 18 байт. Смотрим код ошибки.
 6. Если код равен 0 - «бессмысленные данные», то необходимо продолжить чтение. В противном случае переходим в состояние ошибки.
- Для более подробной информации смотрите главу 10 - «Пример чтения данных на Спектруме».

9.8. Определение размера диска**;<#25 - Размер диска в секторах.**

```
DB #25
DS 11
```

Затем необходимо считать 8 байт. Первые четыре байта - количество секторов на текущем диске, и следующие - размер сектора (как правило, не зависит от диска и равен 2352). Напомню, что размер диска также можно узнать из TOC-сектора диска (глава 12.1).

9.9. Информация об аудио-дисках

Следующая команда позволяет получить информацию, относящуюся к audio-дискам. Работа этой команды мной не исследована, поэтому привожу ее описание практически без комментариев. Может быть, вам удастся извлечь отсюда полезную информацию.

;<#42 - смешанная информация (чтение субканала)

```
DB #42
DB SCMSF - БИТ 1 - ТИП ВЫДАЧИ
АДРЕСОВ 0 - НОМЕР
СЕКТОРА, 1 - MSF
DB FULLINFO - ВАРИАНТ ЗАПРОСА
(ПОЛНЫЙ/КРАТКИЙ -
6-Й БИТ)
DB FUNC - ПОДФУНКЦИЯ (ТОЛЬКО
ДЛЯ ПОЛНОГО ЗАПРОСА)
DS 3
DW LENGTH - ДЛИНА ТАБЛИЦЫ
DS 3
```

Команда выдает блок следующей информации:
+01 DB - состояние проигрывания аудио:

- 0 - неизвестно или не поддерживается

Читатель читателю ///

- 1 - Играет аудио
- 2 - Аудио стоит
- 3 - Аудио остановилось на конце
- 4 - Открыта дверь или ошибка запуска
- 5 - Прочее

+02 DW длина следующих за заголовком данных (0 - нет)

Внимание! +04 и далее присутствует при наличии бита 40h в FullInfo и зависят от Func.

Func не равен 2 или 3:

- +04 DB 01 (формат данных субканала = 1)
- +05 DB Ctrl/Addr
- +06 DB Номер трека.
- +07 DB Point or Index
- +08 DB 0
- +09 DS 3 - MSF/SECTOR на диске
- +12 DB 0
- +13 DS 3 - MSF/SECTOR на дорожке

- Подфункция 2 - (Получить UPC код)

- +04 DB 2 (формат данных субканала = 2)
- +05 DS 3
- +08 DB #80 - флажок наличия UPC (если нет, то UPC отсутствует)
- +09 DS 12 - здесь хранится UPC код (6 цифр в BCD коде)
- +23 DS 3 - Положение чего-то на диске в формате MSF

- Подфункция 3 - (получить ISRC код)

- +04 DB 03h (формат данных субканала = 2)
- +05 DB Ctrl/Addr
- +06 DB Номер трека- не всегда используется.
- +07 DB ?
- +08 DB #80 - флажок присутствия (аналогично функции 2)
- +09 DB далее запись ISRC

9.10. Информация о секторе

Эта команда создает некий компромисс между двумя типами адресации информации на компакт-диске: аудио-адресацией (MFS) и секторной адресацией. На входе команды задаются минуты/секунды/фреймы, а на выходе формируется информация о секторе, находящемся в этом месте - его номере и формате.

№44 - информация о реальных метках положения (Read HEADER).

DB #44

DB SL - БИТ 2 - ЧТО ЗАПИСЫВАТЬ В ВЫХОДНОЙ БУФЕР (ИСХОДНЫЙ НОМЕР СЕКТОРА ИЛИ СЧИТАННЫЙ)

DB 0

DS M, S, F - НОМЕР СЕКТОРА

DB 0

DW LEN - ДЛИНА ВЫДАВАЕМОЙ ИНФОРМАЦИИ

DS 3

Команда возвращает таблицу из 8 байт:

+00 Тип формата сектора (data mode)

+01 0,0,0,0

+05 Адрес сектора (M,F,S).

Информация возвращается только в том случае, если CD смог считать заданный сектор и определить его тип.

9.11. Информация о дорожках

Эта команда также мною не исследована, поэтому привожу ее без комментариев и пояснений. Я эту команду использовал единственно для выяснения, какой компакт-диск находится в CD-ROM'e аудио или информационный. Подпрограмма, позволяющая это делать, будет приведена в конце описания команды.

№43 - информация о дорожках (READ TOC).

DB #43

DB SCMSF - BIT 1:

ТИП ВЫДАЧИ АДРЕСОВ
(0-НОМЕР СЕКТОРА,
1-MSF)

DB FORMAT - ИНФОРМАЦИЯ,
КОТОРУЮ МЫ ХОТИМ
ПОЛУЧИТЬ.

DS 3

DB BEGTRK - НАЧАЛЬНАЯ ДОРОЖКА/
СЕССИЯ (ОТ 1,0
ЗАМЕНЯЕТСЯ НА 1)

DW LENGTH - ДЛИНА ТАБЛИЦЫ
(В СЕКТОРАХ)

DB FUNC - ВАРИАНТЫ ВЫДАЧИ
ИНФОРМАЦИИ (0/#40/#80)

DS 2

Описание байта Format:

0 - выдаются данные для всех сессий.

1 - выдается номер первой завершенной сессии, последней завершенной сессии и начальный адрес последней сессии (в этом случае BegTrk должно быть 0).

2 - начиная с номера сессии, заданной в BegTrk, выдается информация обо всех последующих сессиях.

3 - выдаются данные РМА (учитывающее пространство диска для записи сессий).

4 - выдаются данные АТИР (абсолютное время канавки для незаписанных CD-болванок).

Читатель читателю ///

BegTrk должно быть 0, а MSF - 1).
5 - выдаются данные CD-TEXT

Команда информационная, выдает таблицу дорожек. Максимальная длина таблицы = $8 * \#64 + 4$ байт или $\#64$ (100) дорожек.

Func = $\#00$ - получить обычную таблицу дорожек
= $\#40$ - получить таблицу сессий
= $\#80$ - получить обычную таблицу в расширенном формате

Общий формат таблицы:

DW Len - длина последующих полей в байтах
DB BegTrk - первая дорожка
DB EndTrk - последняя дорожка

Описание дорожек может быть трех форматов:

1) 5 байт на дорожку (внутренний формат, наружу из CD не выдается):

DB Ctrl/Addr - тип дорожки и флаги
DB Index - индекс дорожки (номер)
DB Start (3 байта) - адрес начала дорожки

2) 8 байт на дорожку (Func= $0/\#40$):

DB ?
DB Ctrl/Addr - тип дорожки и флаги
DB TrackNumber - номер дорожки
DB ?
DB Start (4 байта) - адрес начала дорожки

3) 11 байт на дорожку (Func = $\#80$):

DB Res1
DB Ctrl/Addr - тип дорожки
DB Res2
DB Index - индекс дорожки
DB Res3
DB Res4
DB Res5
DB Start (4 байта) - адрес начала дорожки

Возвращаемые переменные имеют следующие значения:

Ctrl/Addr - тип дорожки
Ctrl (младшая тетрада, отдельные биты):
01 - есть pre-emphasis
02 - разрешено копирование
04 - дорожка данных
08 - 4 канала (а не 2)

Addr (старшая тетрада, коды):

0 - нет субканала
1 - в субканале закодирована позиция

2 - в субканале закодирован UPC
3 - в субканале закодирован ISRC
прочее - зарезервировано

Самые распространенные коды:

14h - ROM
10h - audio

Index - кодируется в BCD и для обычной дорожки находится в интервале 01-99. Коды $\#A0$ и выше имеют специальное значение, они не соответствуют физическим дорожкам на диске, а носят служебный характер - информируют о числе дорожек, начале диска, конце диска и т.п.

Start - в зависимости от запроса, может быть либо номером сектора, либо адресом сектора в формате MSF.

С помощью следующей подпрограммы можно узнать, с каким диском мы имеем дело: с данными либо с аудио-диском. Если установлен флаг нуля (Z), то у нас аудио-диск, и пытаться искать на нем файловую систему бесполезно.

```
TEST_AUDIO LD HL,AP_00
            CALL SEND_ATAPI
            LD HL,AP_43_AUDIO
            CALL SEND_ATAPI
            LD HL,AUDIO_BUF
            LD BC,#0006
            CALL TRANSFER_IN
            LD A,(AUDIO_BUF+5)
            CP #10
            RET
```

```
AP_43_AUDIO DB #43,0,0,0,0,0
            DB 1,0,6,#80,0,0
```

```
AUDIO_BUF DS 6
```

Команды управления аудио-дисками.

Поскольку изначально CD-ROM'ы использовались для воспроизведения звука, и лишь позже их приспособили для хранения информации, то поэтому среди команд управления CD-ROM'ом присутствует большое количество команд, работающих непосредственно с audio-форматом дисков. Это так называемые звуковые дорожки, или треки. Как правило, каждый CD-ROM может работать с audio-треками. Так, на передней панели CD-ROM'a существует аудио-выход, куда можно подключить наушники или усилитель, и кнопка для последовательного переключения треков. Звук можно снимать также в цифровом виде и считывать его компьютером (на IBM PC для этого существуют так называемые аудиограбберы - про-

Читатель читателю ///

граммы, считывающие звуковые дорожки и преобразующие их в *.wav или *.mp3 файлы), но на Спектруме, к сожалению, этого сделать пока нельзя. Поэтому программно можно ограничиться только переключением проигрывания выбранного трека, но звук при этом должен сниматься на усилитель либо наушники только с передней панели CD-ROM'a или с аналогичного audio-выхода задней панели.

9.12. Проигрывание в блоках

Эта команда задает начало проигрывания и длину проигрывания в блоках. Причем количество проигрываемых блоков 16-разрядное (максимум - 65535).

;**#45** - проигрывать audio в блоках.

```
DB #45
DB 0
DD STARTBLOCK - БЛОК НАЧАЛА
ПРОИГРЫВАНИЯ
(FFFFFFFF - С ТЕК.
ПОЛОЖЕНИЯ).
«DD» ОЗНАЧАЕТ, ЧТО
ЗАДАЕТСЯ 4 БАЙТА
(«DB»-1, «DW»-2).

DB 0
DW LENGTH - ЧИСЛО БЛОКОВ
DS 3
```

;**#A5** - проигрывать audio в блоках.

От предыдущей эта команда отличается только тем, что количество проигрываемых блоков задается 32-разрядным числом (соответственно, с помощью нее можно задать проигрывание довольно большого отрезка диска).

```
DB #A5
DB 0
DD STARTBLOCK - БЛОК НАЧАЛА
ПРОИГРЫВАНИЯ
(-1 - С ТЕК.
ПОЛОЖЕНИЯ).
DD LENGTH - КОЛИЧЕСТВО БЛОКОВ.
DS 2
```

9.13. Проигрывание audio в MSF

Эта команда задает проигрывание в формате MSF. Такой способ позволяет задать очень точное (до долей секунды) место начала проигрывания и время проигрывания. Таблицу расположения дорожек в этом формате можно получить с помощью команды «**#43** - информация о дорожках» (глава 9.11).

;**#47** - проигрывать audio в MSF.

```
DB #47
DS 2
DB M, S, F - НАЧАЛО ОТРЕЗКА
```

(FF:FF:FF - ТЕКУЩАЯ
ПОЗИЦИЯ)

DB M, S, F - КОНЕЦ ОТРЕЗКА
DS 3

9.14. Остановить/продолжить проигрывание

Эта команда служит для остановки/продолжения проигрывания музыки.

;**#4B** - Start/stop audio.

```
DB #4B
DS 7
DB FUNC
DS 3
```

Func=0 - остановить проигрывание.

Func=1 - продолжить проигрывание.

9.15. Остановка проигрывания

Эта команда очень простая, и служит единственной цели - остановке проигрывания звуковой дорожки. К тому же она не требует дополнительных параметров.

;**#4E** - остановить проигрывание.

```
DB #4E
DS 11
```

Команды чтения данных

Существует две команды чтения данных. Одна из них позволяет читать данные в конкретных секторах, а другая в параметрах MSF (описанных выше). Если чтение в MSF актуально для audio-дисков, когда мы хотим ровно считать звуковую дорожку для преобразования ее, например, в файл (*.wav или *.mp3, минимальный размер такого файла - несколько мегабайт, для Спектрума неактуально). Хотя можно написать программу, проигрывающую с audio-диска короткие области для последующего преобразования их в сэмплы для General Sound. Тогда читать данные удобно как раз в MSF. Чтение же в секторах как раз удобно для чтения файлов (поскольку все параметры файлов указываются именно в секторах). Ниже я приведу описание двух вариантов чтения и пример для чтения секторов на Спектруме.

9.16. Чтение данных в MSF

;**#B9** - Чтение данных в MSF.

```
DS 3
DB #B9
DB FMT
МОЖЕТ БЫТЬ =
#00 ГОДИТСЯ ЛЮБОЙ
ФОРМАТ
#08 ОБЫЧНЫЙ CD-ДИСК
(ЖЕЛТАЯ КНИГА)
#10\ РАЗНОВИДНОСТИ
#14/ ЗЕЛЕНОЙ КНИГИ.
```

Читатель читателю ///

```

DB 0
DB M,S,F      НАЧАЛО ЧТЕНИЯ
DB M,S,F      КОНЕЦ ЧТЕНИЯ
DB FLG        ФЛАГИ ЧИТАЕМОЙ
              ЧАСТИ СЕКТОРА:

BIT 0 - НЕ ИСПОЛЬЗУЕТСЯ
BIT 1 \ ПОЛЕ ОШИБКИ
BIT 2 / (НЕАКТУАЛЬНО)
BIT 3 - EDC/ECC (ДЛЯ НАС ТАКЖЕ
        НЕ АКТУАЛЬНО)
BIT 4 - ПЕРЕДАВАТЬ ДАННЫЕ
        ПОЛЬЗОВАТЕЛЯ.
BIT 5 \ КОД ЗАГЛОВОК
        (%00 - НЕ ПЕРЕДАВАТЬ,
        %01 - ЗАГОЛОВЕК
BIT 6 / %10 - ПОДЗАГОЛОВЕК,
        %11 - И ТО, И ДРУГОЕ)
BIT 7 - ДОБАВИТЬ ДАННЫЕ
        СИНХРОНИЗАЦИИ БЛОКА
DB 0          ВСЕГДА Д.Б.=0,
              ИНАЧЕ ОШИБКА
DB 0

```

Если в этой команде адрес начала чтения совпадает с адресом конца чтения, то происходит позиционирование на указанную позицию и тестируется чтение. Данные не передаются.

9.17. Чтение данных в секторах

;**#BE** - Чтение данных в секторах.

```

DB #BE
DB FMT -      ФОРМАТ ЧТЕНИЯ
              (КАК В #B9).
DB SEC -      НАЧАЛО ЧТЕНИЯ -
              4 БАЙТА.
              (ИСП. ТОЛЬКО 24 БИТА)
DB 0
DW SCNUM -    ЧИСЛО СЕКТОРОВ
DB FLG -      ФЛАГИ ЧИТАЕМОГО
              КУСКА СЕКТОРА
              (КАК В #B9)
DB 0 -        ВСЕГДА Д.Б.=0,
              ИНАЧЕ ОШИБКА.
DB 0

```

Команда полностью аналогична команде **#B9** - «Чтение данных в MFS», за исключением задания адресов области чтения в секторах.

10. Пример чтения данных на Спектруме

Ниже я приведу пример подпрограммы чтения секторов на Спектруме. Перед ее вызовом необходимо выставить значения в пакете **AP_BE** (чтение данных в секторах). Пример такого пакета вы видите перед собой.

SECTOR - сектор для чтения.

SECTORS - количество секторов для чтения (к сожалению, по непонятным причинам, при попытке считать более одного сектора в принимаемых данных возникают ошибки. Поэтому советую вам читать по одному сектору и эту переменную не менять).

```

AP_BE      DB #BE,#00
SECTOR     DB #00,#00,#00,#1F
           DB 0
SECTORS    DB #00,#01
           DB #10
           DB 0,0

```

И собственно подпрограмма чтения:

;IN: [HL] - АДРЕС ДЛЯ ЧТЕНИЯ.

```

LOAD_SECTOR PUSH HL
              CALL NO_BSY
              LD HL,AP_00
              CALL SEND_ATAPI
              LD HL,AP_BE
              CALL SEND_ATAPI
              CALL NO_BSY
              CALL NO_ERROR
              JP C,ERROR
              CALL WAIT_DRQ
              POP HL
              LD B,8
LOAD_SECTOR1 PUSH BC
              CALL READ_S
              POP BC
              DJNZ LOAD_SECTOR1
              RET

READ_S      LD B,0
READ_1     PUSH BC
              LD BC,#F8BE ;#10 ДЛЯ NEMO
              CALL IN_A
              LD (HL),A
              INC HL
              LD BC,#D8BE ;#11 ДЛЯ NEMO
              CALL IN_A
              LD (HL),A
              INC HL
              POP BC
              DJNZ READ_1
              RET

```

В момент чтения может произойти ошибка. Если она будет иметь код 0 - «бессмысленные данные», то дело, вероятно, в том, что диск еще недостаточно успел раскрутиться. В таком случае нужно продолжать чтение. В случае же остальных ошибок нужно производить исправляющие действия (менять диск, сообщать об ошибке) - в общем, на усмотрение программиста. Соответству-

Читатель читателю ///

ющая процедура приведена ниже:

```

ERROR      LD HL,AP_00
           CALL SEND_ATAPI
           LD HL,AP_03
           CALL SEND_ATAPI
           LD HL,BUF_ERROR
           LD BC,#0012
           CALL TRANSFER_IN
           LD A,(BUF_ERROR+2)
           OR A
           JP Z,LOAD_SECTOR
           ... - сообщение
           об ошибке и т.д.
BUF_ERROR  DS 18

```

Для работы процедуры обработки ошибок ERROR необходим вызов пакета определения ошибки (глава 9.7). Вот он:

```

AP_03      DB #03
           DS #03
           DB #12
           DS #07

```

Иногда возникает такая ситуация: после очередного чтения проходит некоторое время, и CD-ROM автоматически переходит в режим пониженного энергопотребления. Диск начинает вращаться медленнее. И при попытке очередного чтения начинается инициализация CD-ROM'a, длящаяся 10-15 секунд. Это очень неудобно при работе. Я решил для себя эту проблему следующим образом: после чтения данных, когда некоторое время обращения к CD-ROM'у не будет и он может остановиться, я вставляю головку CD-ROM'a на начало диска (произвожу «холостое» чтение из сектора 0). При этом команда «01 - Позиционирование» мне по каким-то причинам не подошла. Привожу работоспособную процедуру из моей программы CD_Walk:

```

RECALIBRATE LD HL,0
           LD (SECTOR),HL
           LD (SECTOR+2),HL
           LD (SECTORS),HL
           LD HL,AP_00
           CALL SEND_ATAPI
           LD HL,AP_BE
           CALL SEND_ATAPI
           LD HL,#0100
           LD (SECTORS),HL
           RET

```

11. Пример работы с аудио-дисками

Если вы определили диск, вставленный в ваш CD-ROM как аудио-диск (гл. 9.11), то мы с помо-

щью подпрограмм, описанных в главах 9.12 - 9.15 можем управлять проигрыванием расположенных на нем мелодий (в виде звуковых дорожек или треков). Еще раз напомним, что звук в этом случае необходимо снимать с передней панели CD-ROM или со специального разъема на его задней панели.

11.1 Считывание таблицы треков. Анализ таблицы

Следующая подпрограмма позволит нам считать таблицу треков на аудио-диске.

```

READ_AUDIO LD HL,AP_TAB_AUDIO
           CALL SEND_ATAPI
           LD HL,BUF_AUDTAB
           LD BC,#0100
           CALL TRANSFER_IN
           RET

```

```

AP_TAB_AUDIO DB #43,2,0,0,0,0,1,1,0,0,0
BUF_AUDTAB DS #0100

```

CD-ROM возвращает таблицу, данные в которой удобно интерпретировать следующим образом: вначале идет 6 байт информации (первые 4 байта - общее время проигрывания диска?). Затем идет последовательность 8-байтных описателей, которые имеют следующий формат:

- +0 - номер трека.
- +1 - как правило, 0.
- +2 - часы?
- +3 - минуты
- +4 - секунды
- +5 - фреймы (доли секунды)

Нас интересует номер трека, и его местонахождение на диске, находящееся в 3,4 и 5 байтах описателя. Причем оно указано во временных единицах. Чтобы выяснить время проигрывания трека, нам необходимо из времени следующего трека вычесть время текущего. Самый последний трек в таблице обозначается байтом #AA вместо номера. Итак, давайте посчитаем, сколько у нас треков на диске:

```

;OUT: [A] - КОЛИЧЕСТВО ТРЕКОВ.
COUNT_TRACKS
           CALL READ_AUDIO
           LD HL,BUF_AUDTAB-2
           XOR A
           LD B,A
           LD DE,8
COUNT_TRACK1
           INC B
           ADD HL,DE

```

Читатель читателю ///

```
LD A, (HL)
CP #AA
JR NZ, COUNT_TRACK1
LD A, B
RET
```

11.2 Запуск проигрывания трека

Чтобы запустить проигрывание звукового трека, нам необходимо указать CD-ROM'у, с какого места по какое ему требуется играть. Для этого нужно указать начало нахождения этого трека на диске (3,4,5 байты описателя) и его конец - те же байты следующего описателя. Если конец проигрывания не указывать - то будут проигрываться все последующие треки до конца диска.

```
PLAY_TRACK LD HL, BUF_AUDTAB-2
            LD DE, 8
PLAY_TRACK1 ADD HL, DE
            CP (HL)
            JR NZ, PLAY_T1 ; НАХОДИМ
                                ; НУЖНЫЙ
                                ; ОПИСАТЕЛЬ

            LD BC, 3
            LD DE, 3
            ADD HL, DE
            LD DE, AP_PLAY+3 ; НАЧАЛО
                                ; ПРОИГРЫВАНИЯ

            LDIR
            LD DE, 5
            ADD HL, DE
            LD BC, 3
            LD DE, AP_PLAY+6 ; ОКОНЧАНИЕ
                                ; ПРОИГРЫВАНИЯ

            LDIR
            LD HL, AP_PLAY
            SEND_ATAPI
            RET

AP_PLAY DB #47, 0, 0, 0, 0, 0
        DB 0, 0, 0, 0, 0, 0
```

11.3 Остановка и продолжение проигрывания

Эти команды не представляют никакой трудности. Для остановки (паузы) проигрывания трека необходимо выполнить следующую подпрограмму:

```
STOP_PLAY XOR A
          LD (AP_4B+8), A
          LD HL, AP_4B
          CALL SEND_ATAPI
          RET
```

Для продолжения проигрывания делаем следующее:

```
CONT_PLAY LD A, 1
          LD (AP_4B+8), A
          LD HL, AP_4B
          CALL SEND_ATAPI
          RET

AP_4B DB #4B, 0, 0, 0, 0, 0
      DB 0, 0, 0, 0, 0, 0
```

12. Структура файловой системы диска

В главе 3.3 я уже говорил о файловой структуре диска, имеющей название ISO-9660. Этот стандарт получил широкое распространение, и сейчас практически невозможно встретить диски, не являющиеся ISO-9660. Поэтому можно смело ориентироваться на эту файловую структуру, и не бояться, что с какими-нибудь дисками возникнет несовместимость.

12.1. ТОС (Оглавление диска)

Что из себя представляет эта файловая структура? Как на диске располагаются файлы? Вкратце это выглядит следующим образом: в начале диска, в 16-м секторе, находится так называемая ТОС (Table of Contents, оглавление). Информация в этом секторе располагается следующим образом:

0 - тип диска:

0 - boot disk

1 - основной диск (как правило, установлен этот байт)

2 - дополнительный диск

1-5 - идентификатор ISO-9660. При этом он всегда выглядит как «CD001».

6 - версия файловой структуры (как правило, «1»).

40-71 - имя диска (т.н. «метка тома»).

80-87 - размер диска.

156-189 - информация о корневом каталоге. При этом строка описания имеет именно такой вид, как и описатель любого файла. Здесь есть информация о местоположении каталога, его длине, времени его создания. Формат описателя подробно рассмотрен в следующей главе. Тем не менее, чтобы узнать, где находится корневой каталог, смотрите байты 162-165 - здесь в 4-байтном виде хранится начальный сектор корневого каталога диска.

813-829 - дата создания

Что касается появившихся в последнее время, благодаря пишущим CD-ROM'ам, так называемых мультисессионных дисков (на которые запись была произведена не один раз), то с ними ситуация не так проста, как кажется на первый взгляд. При первой сессии (записи) в 16 дорожку записывается ТОС, и никак там перезаписаться она не может.

Читатель читателю ///

Все последующие сессии располагаются где-то в другом месте, причем неясно, имеют ли они собственную ТОС. Узнать, сколько сессий было на диске, можно через команду #43, задав переменную func = #40. Обратно вернется 12-байтная таблица, в которой 3-й и 6-й байты будут указывать на количество сессий. Так что при работе с CD-ROM на Спектруме настоятельно рекомендую избегать мультисессионных дисков. Тем не менее, в дальнейшем я надеюсь эту проблему решить.

12.2. Каталог. Описание файлов

Корневой каталог - это ни что иное, как один из каталогов на диске. Формат всех каталогов одинаков: последовательно располагается информация о файлах/каталогах, находящихся в описываемом каталоге. Все описатели файлов имеют неодинаковую длину, из-за разной длины имен файлов. Здесь мною приведена схема такого описателя:

0 \ Размер описателя
1 / файла/каталога.

2 \
3 - Положение файла
4 - на диске.
5 /

6 \
7 - Положение файла
8 - на диске (перевернутое).
9 /

10 \
11 - Размер файла.
12 - в байтах
13 /

14 \
15 - Размер файла в байтах
16 - (перевернутый)
17 /

18 - год
19 - месяц
20 - число
21 - час
22 - минуты
23 - секунды
24 - смещение по Гринвичу

25 - атрибуты.

26 - размер файлового элемента

27 - чередование секторов (interleave)

28 \ порядковый номер
29 / тома.

30 \ порядковый номер
31 / тома (перевернутый).

32 - длина имени файла (file_len)
33... - имя файла

32+file_len - байт выравнивания (0)
32+filelen+1 - стандартом не определено.

А теперь подробнее.

1. Размер описателя файла - это количество байт, сколько занимает данный описатель. Чтобы узнать начало следующего описателя, нужно к текущему адресу прибавить размер описателя.

2. Положение файла на диске - это номер сектора, с которого начинается файл. Это число приводится в прямом и перевернутом виде. Читайте, как вам удобно. Далее файл располагается по секторам последовательно, здесь структура ISO 9660 очень похожа на TR-DOS.

3. Размер файла. Приводится в байтах. Для каталога он кратен длине сектора - 2048 (может быть 2048, 4096 и т.д.).

4. Дата создания. На смещение по Гринвичу, думаю, внимания обращать не стоит.

5. Здесь описываются довольно специфические и бесполезные для нас атрибуты. Однако только благодаря этому байту мы можем узнать, с чем имеем дело: с файлом или с подразделом. Если байт равен нулю, то мы находимся в описателе файла. Если он равен 2 (установлен 1-й бит), то, значит, это подраздел.

6. Размер файлового элемента. Связан с режимом чередования секторов. Но т.к. чередование практически не используется, то этот байт теряет значение и равен 0.

7. Чередование секторов. Практически всегда равно 0, то есть отсутствует, и сектора располагаются последовательно, один за другим.

8. Порядковый номер тома. Количество томов, как правило, всегда равно 1, т.е. здесь мы найдем единицу.

9. Длина имени файла. Включает в себя и собственно имя, и точку, и расширение. Так что, например, длина «autoexec.bat» будет равна 12. Кроме того, надо иметь в виду, что в конце имени файла (не подраздела) может находиться последовательность из двух байт: #3B #31. Они также включаются в байт длины имени файла. Если описатель описывает подраздел, то после имени перед следующим описателем находится еще дополнительная информация.

В начале каждого каталога обязательно нахо-

Читатель читателю ///

дится запись с информацией о текущем подразделе, а за ней - запись с информацией о родительском подразделе. Так, в описателе подраздела INFO (... \DIR \DOCS \INFO) в начале будет описание подраздела INFO, а за ним - описание подраздела DOCS. Предполагалось, что это позволит быстро осуществить переход в предыдущий подраздел. Соответственно, в начале корневого каталога будет два одинаковых описателя, указывающих на сам корневой каталог.

Наибольшую трудность при этом вызывает определение количества файлов в каталоге и определении конца каталога. Нам, конечно, известно количество секторов, отведенных под описание каталога. Однако где в последнем секторе находится последний файл, нам неизвестно. Поэтому, чтобы определить количество файлов, нужно двигаться по описателям до тех пор, пока длина очередного описателя не будет равна 0. На машинах с большим объемом памяти предполагается, что все описатели будут считаны в память, однако, когда файлов в одном каталоге больше 700-1000, на Спектруме сделать это становится затруднительным. Поэтому после перехода в очередной каталог нужно проделывать определенные операции: посчитать количество файлов (и директорий), создать в памяти таблицу расположения описате-

лей в каталоге (при необходимости) и т.д.

13. Заключение

Автор выражает большую благодарность Константино Норватову, человеку, одним из первых изучившему протокол ATAPI и предоставившему мне его описание. Также хочу поблагодарить Павла Васильева, помогавшего мне при создании программы и подготовке этого материала, и Мишу Жарова, без которого не были бы написаны программа и данная статья.

В заключение я хочу еще раз уточнить, что статья является лишь практическим руководством по работе с CD-ROM на Спектруме. Поэтому я привел только ту информацию, которая соответствует цели написания данной статьи. Также вполне вероятны небольшие неточности. Если вам нужна более подробная информация по CD-ROM, рекомендую обратиться к замечательной книге Владимира Кулакова «Программирование дисковых подсистем» (СПб., «Питер», 2002).

Я с удовольствием выслушаю ваши замечания по поводу статьи. Мои электронные адреса:

E-mail: vega@zxvega.spb.ru

ZXNet: 500:812/19

Fido: 2:5030/1512

Влад СОТНИКОВ

Загрузка системы iS-DOS с винчестера, подключенного по стандарту (с) Nemo HDD

Для того чтобы понять принцип загрузки системы и ее запуска сначала надо разобраться в строении самой системы. Начнем с системы iS-DOS Classic.

Пример распределения памяти в iS-DOS Classic

0	ПЗУ BASIC-48
16384	экран
23296	программа exebat.com
23552	системные переменные
23900	программа mon.com (menu.com)
24000	область com-файлов
36000	кэш
43000	каналы
44000	резиденты и драйверы
50304	уровень 4 SHELL
54800	уровень 3 WIN
58472	уровень 2 COM
60726	уровень 1 DUD
63296	уровень 0 DOS

Посмотрим, что же из памяти компьютера надо сохранять, чтобы потом можно было загрузить систему.

Область для программы exebat.com в сохранении не нуждается, поскольку эта программа и так загружается каждый раз при запуске bat-файлов.

Нам потребуются системные переменные TR-DOS, а также область каналов Бейсика, т.к. там хранится адрес обработчика RST #10 для 4-го канала, через который вызываются все рестарты iS-DOS. Системные переменные TR-DOS нужны для работы с дискетой.

Область памяти для программ mon.com и menu.com можно не сохранять по той же причине, что и для программы exebat.com.

Область для com-файлов сохранять не нужно.

Кэш сохранять не требуется, т.к. он и без того заново пересоздается при запуске системы.

Остается область каналов и все, что выше. Вот эту область памяти необходимо обязательно сохранять. При этом остаются все ранее установленные драйверы и резиденты, а также там находится само ядро системы (уровни 4-0). Кроме того, нам необходимы каналы устройств. Именно так и работает программа sv.com. Кроме того, в этой области памяти находится обработчик прерываний режима IM 2. Таким образом, нам необходимо сохранить область системных переменных (о ней рассказано чуть позже) и ядро системы со всеми резидентами, драй-

Читатель читателю ///

верами и каналами (в файле `is_dos.sys` или `*_dos.sys`).

Теперь посмотрим, как распределена память в iS-DOS Chic

Пример распределения памяти в IS-DOS Chic

#0010	RST #10	
#003B	подпрограмма драйвера электронного диска	
#0047	признак типа системы/компьютера	
#0069	таблица для драйвера электронного диска	
#0101	буфер для драйвера электронного диска	
#0201	знакогенератор t64 (1K)	
#0601	не используется	
#093A	ядро системы (неизменяемая часть)	
#3800	знакогенератор t42 (2K)	
#4000	экран	
#5B00	программа <code>exebat.com</code>	
#5C00	системные переменные	
#5D5C	программа <code>mon.com</code> (<code>menu.com</code>)	
#5DC0	область <code>com</code> -файлов	
#A5A2	кэш (49 блоков)	
#D6F0	каналы	
#DAD8	резиденты и драйверы	
#F365	уровень 4: SHELL	изменяемая часть
#F78C	уровень 3: WIN	
#FA88	уровень 2: COM	
#FC0C	уровень 1: DUD	
#FE66	уровень 0: DOS	

Как видим, система состоит из двух частей - изменяемой и неизменяемой. Неизменяемая часть расположена в нижней странице памяти (раздел CPU0, адреса #0000-#3FFF). Изменяемая часть расположена в адресах #4000-#FFFF.

Неизменяемая часть один раз создается и записывается в файл `is_dos.com`. В дальнейшей работе ее не потребуется заменять, она только загружается.

А изменяемая часть с учетом предыдущего пункта (iS-DOS

Classic) сохраняется с области каналов до конца памяти. Также сохраняется область системных переменных.

Допустим, что файлы с системой сохранены на диске программой `sv.com`. Однако как его загрузить, если не знаем, где он расположен? У нас есть только один винчестер и больше ничего. Следовательно, нам надо ввести привязку месторасположения файла `is_dos.sys` к чему-нибудь, а именно к 0-му блоку устройства, где располагается файл `is_dos.sys`. Для введения такой привязки служит программа `con.com`. Она сохраняет описатель файла `is_dos.sys` в 0-м блоке текущего устройства по смещению +32 дес. Заодно она считает контрольную сумму этого самого описателя и сохраняет ее в том же самом блоке по смещению +27 дес. Теперь загрузив 0-й блок устройства можно узнать, где располагается файл `is_dos.sys` с сохраненной системой.

Это решает только часть проблем. Откуда мы узнаем, где именно на винчестере располагается нужное нам устройство и вообще нем неизвестна конфигурация винчестера (количество головок, секторов и цилиндров). Можно было бы изначально в программу-загрузчик ввести конфигурацию винчестера, но она в таком случае окажется не универсальной, рассчитанной только под один тип винчестера.

Тут все делается просто. Дело в том, что можно загрузить 0-й блок любого устройства, не зная конфигурации винчестера вообще. Такой блок будет располагаться на 1-м секторе и 0-й головке одного из треков винчестера. А 1-й сектор и 0-я головка есть у любого винчестера. Следовательно, нам надо только перебрать все треки винчестера и читать на них сектор 1 при выбранной головке 0 (нумера-

ция секторов начинается с 1, головок с 0, треков с 0). Но устройств на винчестере может быть несколько, и неизвестно, какое из них было использовано как загрузочное. Для этого 0-й блок загрузочного устройства помечают особым образом, чтобы программа, сканирующая винчестер, сразу узнала, что это то устройство, которое нам надо.

Стандартной программой для подключения системы для ее последующей загрузки с винчестера является программа `uni_con.com`. Она записывает в 0-й блок текущего устройства идентификатор «КАУ» по смещению +127 дес. Именно по наличию такого идентификатора программа загрузчик определяет, что с данного устройства нужно загружать систему. Из этого следует, что загрузочное устройство на винчестере может быть только одно...

До этого мы говорили про сохранение и загрузку ядра системы с набором драйверов и резидентов, расположенного в файле `is_dos.sys`. С этим все более-менее ясно. А вот как же поступить с системными переменными? Их тоже надо куда-то сохранять - без них запустить систему невозможно. В данном случае они сохраняются в отдельном файле, а именно совмещены с непосредственным загрузчиком системы. Сама загрузка и запуск системы с винчестера осуществляются в два этапа: предварительная загрузка с последующим определением параметров винчестера, затем загрузка непосредственно самой системы и ее запуск.

Чтобы сохранить систему на винчестере и подключить ее для последующей загрузки нам требуются:

1) программа `sv.com` для сохранения системы в файле на винчестере;

2) программа `uni_con.com`, которая подключает все нужные

Читатель читателю ///

файлы к 0-му блоку устройства;

3) файл `is_dos.rom`, в котором хранится неизменяемая часть системы iS-DOS Chic (для подключения системы iS-DOS Classic требуется лишь формально);

4) загрузчик системы с винчестера `uni_clas.sys` (или `uni_chic.sys` для iS-DOS Chic), в дальнейшем переименовывается в файл `uni_boot.sys`;

5) загрузчик системы, который вызывается на исполнение из ПЗУ при включении компьютера. В компьютере KAY-1024 это `hdd_boot`, прошитый в ПЗУ. Его можно вызвать на выполнение при выборе соответствующего пункта меню. Для других компьютеров запуск системы возможен только в случае прошивки загрузчика в ПЗУ или если его предварительно загрузить с диска. Для системы iS-DOS Chic файлы `is_dos.rom` и `is_dos.sys` должны быть предварительно настроены под нужный тип компьютера (KAY, Profi, Scorpion и т.д.). Процесс настройки в данной статье не приводится. Он приведен в текстовых файлах на диске с базовым комплектом iS-DOS Chic.

Порядок действий следующий:

Сначала выбираем устройство, на котором будем сохранять систему, и с которого будем в последующем ее загружать. Единственное ограничение - номер трека устройства не должен быть более 255. Дело в том, что при сканировании винчестера программа перебирает треки 0-255. Чтобы узнать номер трека, на котором расположено устройство, нужно запустить программу `ide_tune.com` для текущего драйвера винчестера.

На выбранном устройстве создается для удобства какой-нибудь каталог, куда копируются файлы: `uni_con.com`, `is_dos.rom`, `uni_boot.sys` (файл `uni_clas.sys` или `uni_chic.sys` пе-

реименовываются в зависимости от типа системы, которая подключается).

Если подключается система iS-DOS Classic файл `is_dos.rom` должен присутствовать формально. То есть необязательно брать файл от системы iS-DOS Chic длиной 64 сектора, а просто создать, к примеру, в текстовом редакторе файл с таким же названием и любой длины. Этот файл требуется только при подключении iS-DOS Chic, но так как программа `uni_con.com` является универсальной, она проверяет наличие файла `is_dos.rom` в каталоге. Кстати, после загрузки системы с винчестера, этот файл можно будет удалить.

В системе надо устройство S переключить на то устройство, где мы будем сохранять систему. Это делается потому, что программа `uni_con.com` подключает систему именно на устройство S. Система уже должна быть настроена (установлены необходимые драйвера и резидент, установлен размер кэша). Затем система сохраняется программой `sv.com` в каталог, куда мы копировали файлы. Кстати, обязательно должен быть установлен резидент `calc.res`, иначе при работе программы `uni_con.com` будет ошибка.

Наконец, запускается программа `uni_con.com`, которая проверяет все файлы и подключает к 0-му блоку устройства. После нормального завершения работы программы система станет загружаться с винчестера.

Теперь рассмотрим подробно, что же делает программа `uni_con.com`.

1) проверяется присутствие файлов `is_dos.sys`, `is_dos.rom` и `uni_boot.sys` в том же каталоге;

2) сохраняются в 0-м блоке контрольные суммы файлов `is_dos.sys` и `is_dos.rom` и сами описатели;

3) сохраняется признак загрузочного устройства «KAY» в 0-м блоке;

4) по номеру текущего устройства определяется драйвер устройства, и если это драйвер винчестера, то конфигурация винчестера из драйвера записывается в 0-й блок устройства;

5) системные переменные системы (с #5C00 длиной 512 байт) сохраняются в файл `uni_boot.sys`;

6) подсчитывается контрольная сумма файла `uni_boot.sys` (он ведь был изменен);

7) файл `uni_boot.sys` защищается от удаления, чтения и записи. Это для того, чтобы программа `arzt+.com` не передвинула этот файл куда-нибудь или чтобы его случайно не удалили. Очень важно, чтобы этот файл оставался на одном и том же месте - ведь на его расположение есть ссылка в 0-м блоке устройства. Кстати, с файлом `is_dos.sys` этого не делается, потому что `arzt+.com` сама отслеживает изменение месторасположения этого файла и изменяет 0-й блок устройства.

Теперь посмотрим, что же система сохраняет в 0-м блоке устройства:

Байт	Длина	Описание
27	1	Контрольная сумма описателя <code>is_dos.sys</code>
28	2	Байты 8 и 7 из вектора конфигурации системы. Назначение их неизвестно.
32	32	Описатель <code>is_dos.sys</code>
64	32	Описатель <code>is_dos.rom</code>
96	1	Контрольная сумма описателя <code>is_dos.rom</code>
97	24	Часть драйвера винчестера, где находится информация о структуре винчестера и его характеристиках
125	3	Признак "KAY"
128	32	Описатель <code>uni_boot.sys</code>

В строках, выделенных серым цветом, находятся стандартные параметры, которые указаны в документации на систему.

Осуществляется при вызо-

Читатель читателю ///

ве загрузчика hdd_boot из ПЗУ компьютера. Как уже упоминалось выше, без знания параметров винчестера можно загрузить с любого трека 1-й сектор с 0-й головкой. Последовательно перебираются 1-е сектора на каждом треке с 0 по 255. Если все треки перебраны, то программа «виснет». Ограничение количества треков числом 255 обусловлено использованием одного байта для счетчика треков. Конечно, можно было ввести два байта для счетчика треков, однако представьте себе, что будет перебираться несколько сот треков на винчестере. Времени это займет очень много.

Если на считанном блоке обнаруживается признак загрузочного устройства «КАУ», то из этого блока берется конфигурация винчестера и заносится в загрузчик. Теперь, зная конфигурацию винчестера, можно уже читать любое количество информации с него. Теперь можно взять информацию о непосредственном загрузчике системы uni_boot.sys, программа загружает его в память и запускает на выполнение.

Теперь начинается загрузка непосредственно самой системы. Кстати, 0-й блок уже загружен в память программой hdd_boot. Процессы загрузки iS-DOS Classic и Chic несколько различаются.

Программа uni_boot.sys (для системы Classic) делает следующее:

1) берет из 0-го блока информацию о файле is_dos.sys и за-

ружает его в память (предварительно устанавливается 0-я банка памяти).

2) устанавливается вектор прерывания системы из 0-го блока (находится в описателе файла is_dos.sys по смещению +22 dec);

3) системные переменные (занесенные в файл uni_boot.sys программой uni_con.com) переносятся из тела программы в память по адресу #5C00;

4) система уже в памяти, системные переменные перенесены, становится доступно обращение к рестартам системы через RST #10, поэтому определяем размер электронного диска в блоках и создаем сам электронный диск (кэш);

5) дата из 0-го блока переносится в систему;

6) переключаемся на устройство «S», с которого и сохранялись;

7) выходим в систему с запуском autoexec.bat.

Для загрузки iS-DOS Chic процесс загрузки несколько меняется:

1) Устанавливаем банк 8 памяти в раздел CPU3, в раздел CPU0 включается банк 0 ОЗУ. Кстати, для КАУ и PROFi загрузчики должны быть разные, т.к. в КАУ используется порт #1FFD, а в PROFi - порт #DFFD.

2) Берем информацию о файле is_dos.rom и загружаем файл в под-ПЗУшную область;

3) берет из 0-го блока информацию о файле is_dos.sys и загружает его в память (предварительно устанавливается 0-я бан-

ка памяти).

4) устанавливается вектор прерывания системы из 0-го блока (находится в описателе файла is_dos.sys по смещению +22 dec);

5) системные переменные (занесенные в файл uni_boot.sys программой uni_con.com) переносятся из тела программы в память по адресу #5C00;

6) система уже в памяти, системные переменные перенесены, становится доступно обращение к рестартам системы через RST #10, поэтому определяем размер электронного диска в блоках и создаем сам электронный диск (кэш);

7) дата из 0-го блока переносится в систему;

8) переключаемся на устройство «S», с которого и сохранялись;

9) выходим в систему с запуском autoexec.bat.

При написании статьи использованы материалы по iS-DOS, опубликованные в электронном издании ZX-Format 5, описания рестартов системы из приложения к диску с iS-Assembler, а также исходные тексты программ uni_con.com, uni_clas.sys, uni_chic.sys, hdd_boot.com.

Сергей БАГАН

E-Mail: Prusaker@list.ru

P.S. Тем, кто желает ознакомиться с полной версией статьи следует заказать себе сборник «iS-files» № 3, № 4, где данная статья опубликована со всеми приложениями.

Поддержи
разработчика

рублем

Подключение IBM-периферии к ZX Spectrum. Клавиатура

К настоящему времени существует несколько вариантов подключения клавиатуры от IBM PC к Спектруму, причем практически все они являются универсальными, то есть могут подключаться к любой из его модификаций.

Сейчас я не буду разбирать достоинства и недостатки этих вариантов, а просто предлагаю свой вариант контроллера клавиатуры, основой которого является микроконтроллер AT90S2313 (или ATTiny 2313, приходящий ему на замену) фирмы ATMEL. Имея при 20-выводном корпусе 15 линий ввода/вывода (у Tiny модификации 18 линий), память программ 2 килобайта и ОЗУ объемом 128 байт, этот микроконтроллер позволяет создать простое и компактное устройство, позволяющее подключить к Спектруму стандартную IBM-клавиатуру.

Для начала хочу напомнить, что в Спектруме клавиатура представляет собой матрицу из 8 колонок и 5 строк, в узлах которой находятся контакты, замыкаемые при нажатии на расположенные над ними клавиши (рис 1).

Это базовая схема 40-клавишной клавиатуры. Расширенные клавиатуры отличаются наличием клавиш замыкающих сразу два контакта в матрице. Причем клавишей-модификатором обычно является либо Symbol Shift (S.Sh), либо Caps Shift (C.Sh).

5-разрядная шина строк матрицы подключена к шине данных процессора Z80 через порт клавиатуры, который выбирается младшей половиной адресной шины процессора Z80 и доступен как порт ввода с адресом 0FEh (десятичный адрес 254). Каждая из 5 линий строк подтянута к шине питания (+5V) резистором с номиналом обычно 10 Ком, что

при чтении обеспечивает логическую «1» в соответствующих разрядах порта клавиатуры. Старшие 8 бит адресной шины процессора Z80 позволяют выбрать одну или несколько из 8 колонок клавиатурной матрицы как порт ввода, адрес которого состоит из адреса порта клавиатуры и кода старшей половины адресной шины процессора Z80 с нулевым логическим уровнем в соответствующем разряде. Для исключения замыкания адресных линий при нажатии нескольких клавиш на каждой из них включены развязывающие диоды (как вариант могут устанавливаться буферные схемы с открытым коллектором, например 555ЛП9). Для проверки замыкания контакта в одном или нескольких узлах клавиатурной матрицы последовательно считываются состояния портов клавиатуры с низким уровнем в выбранной колонке. При не нажатых клавишах, с шины данных клавиатуры за счет их подтяжки к +5V, читаются сигналы высокого уровня (логическая «1»). Если нажать одну или несколько клавиш, то при чтении порта в соответствующих нажатым узлам битах будут читаться сигналы низкого уровня (логический «0»). Если выбирается несколько колонок, то нажатие клавиш в соседних колонках приводит к формированию в порту сигнала объединенного схемой проводного «И». Таким образом, имеется 40 контактов, замыкание каждого из которых может анализироваться программно чтением 8 портов ввода, состояния каждого из которых определяется замыканием этих контактов.

Принцип работы клавиатуры IBM PC практически совпадает с описанным выше, но если в Спектруме клавиатура является составной частью

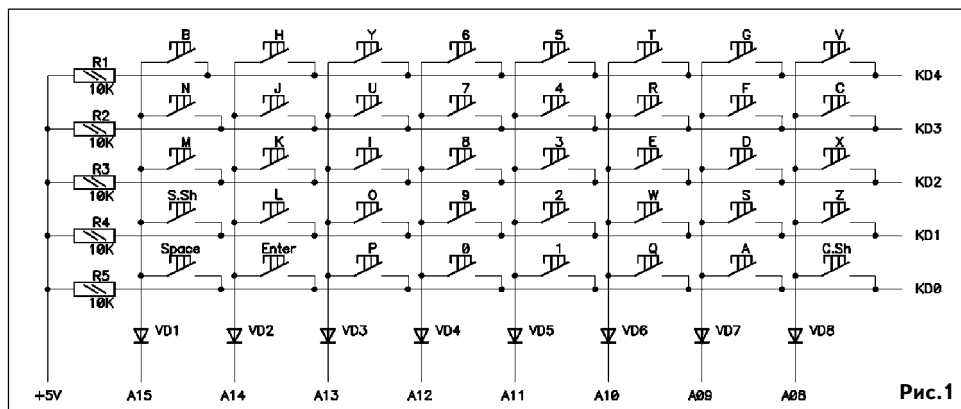


Рис.1

Железо

схемы компьютера, то в IBM это отдельное устройство со своим контроллером. Для передачи информации о нажатых клавишах из клавиатуры в компьютер используется последовательный двухпроводный интерфейс, состоящий из линии данных DAT и линии синхронизации CLK.

Существуют два стандарта обмена информацией с клавиатурами. Первый стандарт использовался для приема информации от XT-клавиатур и был однонаправленным - только передача скан-кода нажатых (или отжатых) клавиш от клавиатуры. Второй стандарт используется для современных AT-клавиатур и PS/2 мышей и предусматривает как прием, так и передачу информации от компьютера к клавиатуре.

По понятным причинам подключение XT-клавиатуры мы рассматривать не будем.

На **рис. 2 и 3** показаны временные диаграммы передачи информации из AT-клавиатуры в компьютер и передачи в AT-клавиатуру команды от компьютера соответственно.

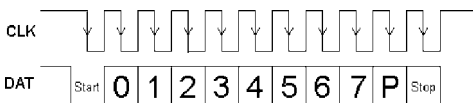


Рис. 2

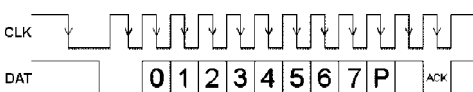


Рис. 3

Здесь:

0..7 - номера передаваемых битов;

P - бит паритета (нечетный);

ACK - бит подтверждения приема от клавиатуры.

Теперь немного о том, как кодируется передаваемая из клавиатуры информация о состоянии клавиш. Контроллеры, встроенные в AT-клавиатуры поддерживают три различных таблицы скан-кодов (Scan Code Set):

Scan Code Set 1 - таблица скан-кодов XT-клавиатуры;

Scan Code Set 2 - таблица скан-кодов AT-клавиатуры;

Scan Code Set 3 - дополнительная таблица скан-кодов AT-клавиатур.

Переключение таблиц выполняется передачей в клавиатуру команды Set mode с номером, соответствующим выбранной таблице (в некоторых клавиатурах переключится в режим 1, не удается).

Характерной особенностью 3-го режима является возможность работы практически всех клавиш в режиме Make/Break, в котором при удержании кла-

виш не выполняется автоповтор передачи скан-кода нажатой клавиши, в отличие от режима 1 и 2 где используется режим Typematic (при удержании, автоповтор с заданной частотой). Еще одно преимущество этого режима - уникальный скан-код для всех клавиш без использования дополнительного префиксного кода (0E0h), который в режиме 1 и 2 предшествует нажатию и отпусанию некоторых клавиш. Это уменьшает время необходимое на прием информации от клавиатуры, реализованный в контроллере программно без использования прерываний. Сделано это для того, что бы уменьшить время реакции системы на опрос порта клавиатуры, который выполнен с использованием механизма аппаратного прерывания микроконтроллера AT90S2313.

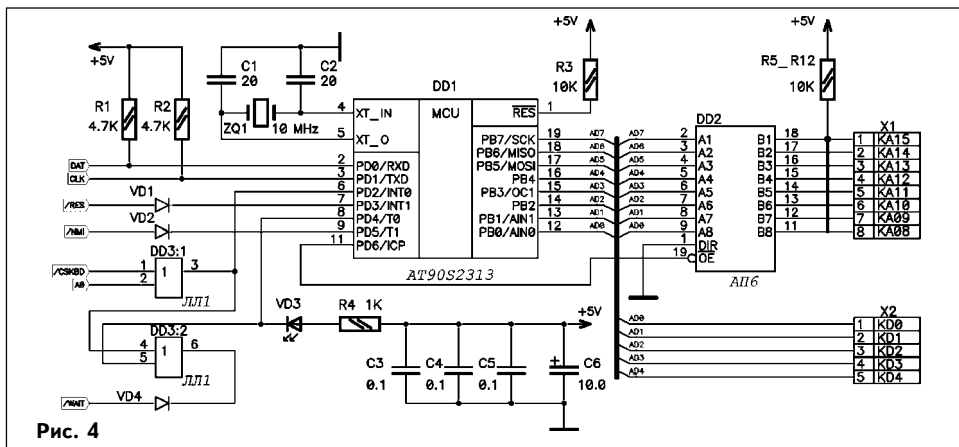
Итак, задачей контроллера будет управление режимом работы клавиатуры, прием, анализ, буферизирование и преобразование скан-кодов клавиатуры IBM и, по запросу от процессора Z80, выдачу в порт клавиатуры Спектрума кода, соответствующего замкнутым контактам его матрицы. Эти задачи выполняет контроллер клавиатуры, схема которого представлена на **рис. 4**.

Как уже отмечено, основной схемой является микроконтроллер AT90S2313 фирмы ATMEL. Тактовая частота его работы задается кварцем ZQ1 на 10 МГц, типовое включение, которого требует установки двух конденсаторов C1 и C2 на 15-27 пФ.

Порт В микроконтроллера используется для чтения старших адресных линий процессора Z80 и для выдачи в порт клавиатуры Спектрума 5 разрядов данных. Для развязки шины адреса и данных используется любой 8-разрядный не инвертирующий буфер, например АП6, управляемый битом PD6 микроконтроллера. Учитывая наличие на адресных линиях клавиатуры развязывающих диодов или буферов с открытым коллектором, каждая из адресных линий на входе АП6 подтянута к +5V резистором с номиналом 10 КОм.

Линии CLK и DAT клавиатуры IBM подключены к 0 и 1 битам порта D микроконтроллера. Рекомендуется их подтяжка к +5V резисторами с номиналом порядка 4.7 КОм.

На вход PD2/INT0 микроконтроллера, который используется как вход внешнего прерывания, со Спектрума подается сигнал запроса на чтение порта клавиатуры /CSKBD. Элемент ИЛИ (DD3:1) установлен для применения контроллера совместно со «Скорпионами» и «Каями», у которых сигнал выборки клавиатуры формируется совместно с сигналом выборки Кэмпстон джойстика, а для их разделения используется младший бит (A0) адреса Z80. Для остальных Спектрум совместимых моделей можно либо не ставить этот элемент вовсе, либо подать на оба его входа сигнал /CSKBD.



Элемент ИЛИ (DD3.2) обеспечивает управление сигналом /WAIT процессора Z80. При установленном в «0» бите PD4 микроконтроллер разрешает перевод процессора Z80 в режим ожидания по сигналу выборки клавиатуры на то время, пока будет считан адресный код сканирования, вычислен и выдан код данных в порт клавиатуры. После этого бит PD4 устанавливается в «1», что снимает сигнал /WAIT с входа Z80, переводя его в рабочий режим.

Длительность сигнала /WAIT при тактовой частоте микроконтроллера составляет всего порядка 3-4 мксек, что практически незаметно для работы Спектрума. Но дополнительно для уменьшения торможения Спектрума сигналом /WAIT от контроллера, используется тот факт, что при не нажатых клавишах с порта клавиатуры в любом случае считываются все «1». При этом можно заблокировать битом PD4 выдачу сигнала /WAIT, а на вход порта клавиатуры с выхода микроконтроллера подать все «1». Таким образом, при не нажатых клавишах контроллер клавиатуры никак не влияет на скорость работы Спектрума.

Светодиод VD3 включен таким образом, что при готовности контроллера выдать код клавиатуры он светится (есть нажатые клавиши), а иначе нет.

Свободные биты PD3 и PD5 микроконтроллера используются для формирования сигналов /RES (СБРОС) и /NMI (Не маскируемое прерывание) для процессора Z80.

Теперь самый важный вопрос - привязка клавиш IBM клавиатуры к клавиатуре Спектрума.

В начале о клавишах выполняющих специфические функции.

Клавиша Pause/Break так и просится на роль формирователя сигнала остановки работы компьютера. При ее нажатии в программе контроллера устанавливается флаг ожидания и при пер-

вом же обращении Спектрума к клавиатуре сигнал /WAIT устанавливается в «0» и битом PD4 удерживается в этом состоянии до тех пор, пока на клавиатуре не будет нажата любая клавиша.

Для сброса компьютера по традиции обычно используется нажатие трех клавиш Ctrl+Alt+Del. Я отказался от этого варианта, использовав для формирования сигнала /RES факт нажатия клавиши Print Screen. Дело в том, что в прошивке «Кворума 128», которую я когда-то делал, после /RES анализируются клавиши Caps Shift и Symbol Shift и при их нажатии по Сбросу выход идет либо в Basic 48, либо в Basic 128.

Формирование импульса на выходе PD5 (/NMI) происходит при нажатии клавиши F12.

Теперь о принципе формирования кодов матрицы клавиатуры.

Скан-коды 102 клавиш клавиатуры IBM в режиме 3 имеют уникальный код в диапазоне от 01h до 84h (некоторые коды не задействованы). При нажатии этот код однократно передается по последовательному интерфейсу в контроллер, а при отжатии в начале передается префиксный код (0F0h), а затем ее скан-код. Эти скан-коды принимаются и при нажатии запоминаются в буфере, а при отжатии удаляются из него. Судя по описанию на Z8602/14/15 - один из контроллеров для клавиатур от фирмы Zilog, существует ограничение на число одновременно нажатых клавиш на клавиатуре IBM. В частности в этом контроллере поддерживается до 6 одновременно нажатых клавиш. Из этих соображений размер буфера для нажатых клавиш в контроллере установлен равным 8 (небольшой запас не помешает).

Теперь необходимо взять скан-коды нажатых клавиш из буфера и сформировать по ним коды соответствующие замкнутым контактам в матрице клавиатуры Спектрума. Для этого используем

Железо

следующий подход - каждому скан-коду клавиши IBM приведем в соответствие байт в таблице со следующей структурой:

D7 - флаг клавиши Caps Shift;

D6 - флаг клавиши Symbol Shift;

D5..D3 - номер адресной линии в матрице клавиатуры;

(000 - A08 ... 111 - A15)

D2..D0 - номер бита данных в матрице клавиатуры;

(001 - KD0 ... 101 - KD4)

Номера битов шины данных в коде специально считаются от 1 для того, что бы использовать нулевое значение байта в таблице для скан-кодов, которые не требуют преобразования. Дополнительно введем еще код выбора альтернативной таблицы кодировки, признаком которого будет одновременная установка флагов Caps Shift и Symbol Shift.

В соответствии с этим введем следующие обозначения:

CapSh equ 10000000b ; флаг нажатия клавиши Caps Shift

SymSh equ 01000000b ; флаг нажатия клавиши Symbol Shift

ALTtb equ CSH+SSH ; признак альтернативной таблицы.

; адресные коды

A08 equ 00000000b

A09 equ 00001000b

A10 equ 00010000b

A11 equ 00011000b

A12 equ 00100000b

A13 equ 00101000b

A14 equ 00110000b

A15 equ 00111000b

; коды данных

D0 equ 00000001b

D1 equ 00000010b

D2 equ 00000011b

D3 equ 00000100b

D4 equ 00000101b

Теперь в соответствии с рис. 1 каждому узлу в матрице клавиатуры Спектрума можно задать его код, обозначив соответственно:

KI_CS equ A08+KD0 ; Caps Shift

KI_Z equ A08+KD1

KI_X equ A08+KD2

KI_C equ A08+KD3

KI_V equ A08+KD4

KI_A equ A09+KD0

KI_S equ A09+KD1

KI_D equ A09+KD2

KI_F equ A09+KD3

KI_G equ A09+KD4

KI_Q equ A10+KD0

KI_W equ A10+KD1

KI_E equ A10+KD2

KI_R equ A10+KD3

KI_T equ A10+KD4

KI_1 equ A11+KD0

KI_2 equ A11+KD1

KI_3 equ A11+KD2

KI_4 equ A11+KD3

KI_5 equ A11+KD4

KI_0 equ A12+KD0

KI_9 equ A12+KD1

KI_8 equ A12+KD2

KI_7 equ A12+KD3

KI_6 equ A12+KD4

KI_P equ A13+KD0

KI_O equ A13+KD1

KI_I equ A13+KD2

KI_U equ A13+KD3

KI_Y equ A13+KD4

KI_CR equ A14+KD0 ; Enter

KI_L equ A14+KD1

KI_K equ A14+KD2

KI_J equ A14+KD3

KI_H equ A14+KD4

KI_SP equ A15+KD0 ; Space

KI_SS equ A15+KD1 ; Symbol Shift

KI_M equ A15+KD2

KI_N equ A15+KD3

KI_B equ A15+KD4

Такой способ кодирования позволяет каждому скан-коду привести в соответствие номер адресной линии и номер бита данных. А так же сформировать из буфера содержащего скан-коды нажатых клавиш, еще один буфер из 8 байт, каждый из которых привязан к запрошенной при сканировании адресной линии и содержит логический «0» в позиции замкнутого узла матрицы.

Флаги Symbol Shift и Caps Shift позволяют заданным клавишам обеспечивать двойное нажатие.

Введя такие обозначения, теперь достаточно просто создать таблицу соответствия скан кодов IBM-клавиатуры, кодам, выдаваемым по запросу чтения порта клавиатуры Спектрума.

Эта таблица специально вынесена в отдельное жестко заданное адресное пространство микроконтроллера, что позволяет ее модифицировать и компилировать отдельно от программной части микроконтроллера, в которой ее адрес прописан.

Теперь о флаге альтернативной таблицы. Он введен для того, чтобы ввести привязку кодов для клавиш, имеющих в зависимости от нажатия клавиши Shift на IBM-клавиатуре двойное значение. Например, клавиша с кодами «:» (двоеточие без Shift) и «;» (точка с запятой с Shift).

А также для клавиш, которые на IBM-клавиатуре имеют отличающееся от Спектрума назначение. Например, клавиша «8» на клавиатуре Спектрума

Железо ///

с Symbol Shift выдает «*» (открывающая круглая скобка), а на клавиатуре IBM она обозначена как «*» (знак умножения при нажатии с Shift).

В основной таблице для альтернативных клавиш к признаку ALT добавляется адрес, по которому в дополнительной таблице располагаются два кода соответствующие двум кодовым значениям (первое без Shift, второе с Shift). Принцип кодировки такой же, как и в первой таблице за исключением того, что теперь клавишам можно задать тройное нажатие - основной код плюс два флага Shift.

Как оказалось создание таблицы кодировки не тривиальная задача и зависит от пристрастий пользователя. Поэтому на сегодняшний день имеется как минимум два варианта: авторский, который исходил чисто из логических соображений, и вариант Александра Шушкова, который я бы назвал пользовательским, поскольку он исходил, как я понимаю, из удобства пользования клавиатурой.

Теперь о том, что касается подключения контроллера клавиатуры к конкретным моделям Спектрум-совместимых компьютеров. Его можно подключить в том виде, как он представлен на схеме. Но на практике оказалось, что во многих случаях можно упростить схему контроллера, используя в качестве буфера адреса уже имеющиеся в схемах этих компьютеров регистры. Это касается в частности «Профи», «Пентагона», «АТМ», «Скорпиона», «Кая» и некоторых других моделей, у которых адресная шина процессора перед клавиатурой буферизована. Чаще всего такой буфер включен в режим постоянной выборки (вход /OE заземлен). Если этот вход буфера отключить от земли и подать на него сигнал с выхода PD6 микроконтроллера, тогда необходимость в буфере DD2 (АП6) отпадает. Развязывающие диоды надо будет из схемы исключить, запаяв вместо них перемычки, а подтягивающие резисторы R5..R12 можно не устанавливать.

Во многих случаях можно отказаться и от установки отдельного кварца на микроконтроллере, подав на его вывод 5 сигнал с генератора Спектрума. При этом как показал опыт, хотя для AT90S2313 нормируется тактовая частота до 10 МГц, большинство экземпляров прекрасно работает на частоте 14 МГц. В крайнем случае, можно подать и 7 или 8 МГц с контроллера TRDOS.

Сигнал /WAIT в схемах многих компьютеров либо не задействован и на него подается через резистор напряжение +5V (логическая «1»), либо на него подается сигнал /WAIT с других узлов компьютера. Для работы с контроллером клавиатуры необходимо все сигналы /WAIT на входе Z80 объединить по схеме «И». Проще всего это сделать, если на входе процессора Z80 (вывод 24) установить резистор с номиналом порядка 1-3 КОм,

а все сигналы /WAIT подавать на него через развязывающие диоды (также как включен диод VD4).

По поводу разъемов подключения клавиатуры. В основном теперь продаются клавиатуры PS/2 с разъемом типа MiniDIN с 6 контактами. Но у многих, наверное, сохранились АТ-клавиатуры с 5-контактными вилками типа СГ-5 (импортные типа DIN-5). На рис. 5 показана цоколевка этих разъемов.

Несколько слов о программировании микроконтроллера AT90S2313.

При выпуске все микроконтроллеры выпускаются с разрешенным режимом низковольтного программирования и поэтому могут легко и просто быть запрограммированы с помощью простейшего устройства, подключаемого к LPT-порту любого IBM PC. Существует свободно распространяемая программа «FBPRG», которая позволяет записать заданную прошивку в микроконтроллер. Вполне вероятно, что скоро такая программа появится и для Спектрума, оснащенного портом принтера. Таким образом, любой желающий сможет достаточно легко собрать этот контроллер клавиатуры и подключить его к своему любимому Спектруму.

Ну а теперь таблицы кодировки клавиш:

Во втором столбце (после знака точки с запятой «;») стоит скан-код IBM, сразу за ним идет клавиша IBM.

; Таблица скан-кодов клавиш AT
;(скан-коды до 06h не используются).

```
tab_kbd:
DB 0FFh ;06h
DB SymSh+Kl_Q ;07h F1
DB CapSh+Kl_SS ;08h ESC
;скан-коды следующих 4 клавиш
;в программе перемещены в это место таблицы
;(а эти скан-коды в IBM не задействованы).
DB SymSh+Kl_J ;09h/84h [-]
DB SymSh+Kl_CR ;0Ah/81h Left Flying
Windows
DB SymSh+Kl_CR ;0Bh/82h Right Flying
Windows
DB CapSh+Kl_CR ;0Ch 83h Menu
;—
DB CapSh+Kl_1 ;0Dh Tab
(CapSh+ «1» Edit)
DB AltTb+0 ;0Eh “/~ -> AltTab
```

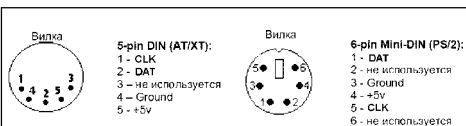


Рис. 5

Железо **////**

DB SymSh+Kl_W ;0Fh	F2	DB AltTb+1 ;41h	,/< ->
; DB 0FFH ;10h		AltTab	
DB Kl_CS ;11h	Left Ctrl	DB Kl_K ;42h	K
CapSh		DB Kl_I ;43h	I
DB Kl_SS ;12h	Left Shift	DB Kl_O ;44h	O
	SymSh	DB AltTab+13 ;Kl_0 ;45h	0/) -> AltTab
DB 0FFH ;13h		DB AltTab+12 ;Kl_9 ;46h	9/(-> AltTab
DB CapSh+Kl_2 ;14h	Caps Lock	DB SymSh+Kl_D ;47h	F9
DB Kl_Q ;15h	Q	DB 0FFH ;48h	
DB Kl_1 ;16h	1/!	DB AltTb+2 ;49h	,/> -> AltTab
DB SymSh+Kl_E ;17h	F3	DB AltTb+3 ;4Ah	//? -> AltTab
DB 0FFH ;18h		DB Kl_L ;4Bh	L
DB CapSh+Kl_3 ;19h	Left Alt	DB AltTb+4 ;4Ch	;/: -> AltTab
DB Kl_Z ;1Ah	Z	DB Kl_P ;4Dh	P
DB Kl_S ;1Bh	S	DB AltTb+5 ;4Eh	-/_ -> AltTab
DB Kl_A ;1Ch	A	DB SymSh+Kl_F ;4Fh	F10
DB Kl_W ;1Dh	W	; DB 0FFH ;50h	
DB Kl_2 ;1Eh	2/@	DB 0FFH ;51h	
DB SymSh+Kl_Y ;1Fh	F4	DB AltTb+6 ;52h	"/» -> AltTab
; DB 0FFH ;20h		DB 0FFH ;53h	
DB Kl_C ;21h	C	DB AltTb+7 ;54h	[/{ -> AltTab
DB Kl_X ;22h	X	DB AltTb+8 ;55h	=/+ -> AltTab
DB Kl_D ;23h	D	DB CapSh+Kl_9 ;56h	F11
DB Kl_E ;24h	E	DB 000H ;57h	Print Screen ->
DB Kl_4 ;25h	4/\$		RESET
DB Kl_3 ;26h	3/#	; скан-коды правых клавиш Ctrl и Shift	
DB SymSh+Kl_U ;27h	F5	; в программе заменяются скан-кодами	
DB 0FFH ;28h		; их левых аналогов.	
DB Kl_Sp ;29h	SPACE	DB 000H ;58h	Right Ctrl -> 11h
DB Kl_V ;2Ah	V	DB 000H ;59h	Right Shift -> 12h
DB Kl_F ;2Bh	F	DB Kl_CR ;5Ah	ENTER
DB Kl_T ;2Ch	T	DB AltTb+9 ;5Bh] / -> AltTab
DB Kl_R ;2Dh	R	DB AltTb+10 ;5Ch	\ / -> AltTab
DB Kl_5 ;2Eh	5/%	DB 0FFH ;5Dh	
DB SymSh+Kl_I ;2Fh	F6	DB 000H ;5Eh	F12 -> NMI
; DB 0FFH ;30h		DB 000H ;5Fh	Scroll Lock
DB Kl_N ;31h	N	;	
DB Kl_B ;32h	B	DB CapSh+Kl_6 ;60h	[Down]
DB Kl_H ;33h	H	DB CapSh+Kl_5 ;61h	[Left]
DB Kl_G ;34h	G	DB 000H ;62h	Pause/Break ->
DB Kl_Y ;35h	Y		WAIT
DB Kl_6 ;36h	6/^	DB CapSh+Kl_7 ;63h	[Up]
DB SymSh+Kl_A ;37h	F7	DB CapSh+Kl_Y ;64h	[Delete]
DB 0FFH ;38h		DB CapSh+Kl_Z ;65h	[End]
DB CapSh+Kl_4 ;39h	Right Alt	DB CapSh+Kl_0 ;66h	BackSpace
DB Kl_M ;3Ah	M	DB CapSh+Kl_V ;67h	[Insert]
DB Kl_J ;3Bh	J	DB 0FFH ;68h	
DB Kl_U ;3Ch	U	DB Kl_1 ;69h	[1]
DB Kl_7 ;3Dh	7/&	DB CapSh+Kl_8 ;6Ah	[Right]
DB AltTab+11 ;Kl_8 ;3Eh	8/* -> AltTab		CapSh+ «8»
DB SymSh+Kl_S ;3Fh	F8	DB Kl_4 ;6Bh	[4]
; DB 0FFH ;40h		DB Kl_7 ;6Ch	[7]
		DB CapSh+Kl_R ;6Dh	[PageDown]
		DB CapSh+Kl_W ;6Eh	[Home]
		DB CapSh+Kl_C ;6Fh	[PageUp]

Железо ///

```
DB K1_0 ;70h [0]
DB SymSh+K1_M ;71h [.]
DB K1_2 ;72h [2]
DB K1_5 ;73h [5]
DB K1_6 ;74h [6]
DB K1_8 ;75h [8]
DB 000H ;76h NumLock
DB SymSh+K1_V ;77h [/]
DB OFFH ;78h
DB K1_CR ;79h [ENTER]
DB K1_3 ;7Ah [3]
DB OFFH ;7Bh
DB SymSh+K1_K ;7Ch [+]
DB K1_9 ;7Dh [9]
DB SymSh+K1_B ;7Eh [*]
DB OFFH ;7Fh
```

; Таблица клавиш с двумя кодами:

; 1 код - без Shift, 2 код - с Shift

ORG 47AH

AltTab:

```
DB SymSh+K1_X,SymSh+K1_A ;0Eh "/~ ->
AltTab+0
```

```
DB SymSh+K1_N,SymSh+K1_R ;41h ,/< ->
AltTab+1
```

```
DB SymSh+K1_M,SymSh+K1_T ;49h ./> ->
AltTab+2
```

```
DB SymSh+K1_V,SymSh+K1_C ;4Ah //? ->
AltTab+3
DB SymSh+K1_O,SymSh+K1_Z ;4Ch ;/: ->
AltTab+4
DB SymSh+K1_J,SymSh+K1_0 ;4Eh -/_ ->
AltTab+5
DB SymSh+K1_7,SymSh+K1_P ;52h "/» ->
AltTab+6
DB SymSh+K1_Y,SymSh+K1_F ;54h [/{ ->
AltTab+7
DB SymSh+K1_L,SymSh+K1_K ;55h =/+ ->
AltTab+8
DB SymSh+K1_U,SymSh+K1_G ;5Bh ]/> ->
AltTab+9
DB SymSh+K1_D,SymSh+K1_S ;5Ch \\/ ->
AltTab+10
DB K1_8,SymSh+K1_B ;3Eh 8/* ->
AltTab+11
DB K1_9,SymSh+K1_8 ;46h 9/( ->
AltTab+12
DB K1_0,SymSh+K1_9 ;45h 0/) ->
AltTab+13
```

Камиль КАРИМОВ,
620147, г. ЕКАТЕРИНБУРГ, а/я 95
E-mail: k2k@list.ru

Проблемы с клавиатурой?**Купи новый КОНТРОЛЛЕР!****Контроллер клавиатуры для Спектр-совместимых компьютеров****Преимущества:**

- поддержка ps/2 и din клавиатур от PC;
- малые габариты;
- очень короткий WAIT (около 3 мсек) и только при нажатии клавиш;
- возможность изменять раскладку клавиатуры.

Недостатки:

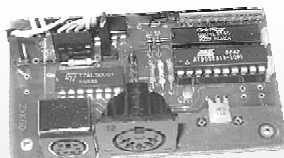
- подключается к плате компьютера жгутом из 14-15 проводников.

Раскладки на выбор:

1. ESC-[CS+SS], F1-[SS+Q], F2-[SS+W], F3-[SS+E], F4-[SS+Y], F5-[SS+U], F6-[SS+I], F7-[SS+A], F8-[SS+S], F9-[SS+D], F10-[SS+F], F11-[CS+9], F12->[NMI], Print Screen->[RESET], Pause/Break->[WAIT], Tab-[CS+1(Edit)], Caps Lock-[CS+2], Left/Right Shift-[SS], Left/Right Ctrl-[CS], Left Flying Windows-[CS+CR], Left Alt-[CS+3], Right Alt-[CS+4], Right Flying Windows-[SS+CR], Menu-[CS+1], BackSpace-[CS+0], Insert-[CS+V], Delete-[CS+Y], Home-[CS+W], End-[CS+Z], PageUp-[CS+C], PageDown-[CS+R]

2. ESC-[CS+1(Edit)], F1-[CS+SS+1], F2-[CS+SS+2], F3-[CS+SS+3], F4-[CS+SS+4], F5-[CS+SS+5], F6-[CS+SS+6], F7-[CS+SS+7], F8-[CS+SS+8], F9-[CS+SS+9], F10-[CS+SS+0], F11(not used), F12->[NMI], Print Screen->[RESET], Pause/Break->[WAIT], Tab-[CS+SP(Break)], Caps Lock-[CS+2], Left/Right Shift-[CS], Left/Right Ctrl-[CS+SS(EXT_MODE)], Left Flying Windows-[SS+CR], Left Alt-[SS], Right Alt-[SS], Right Flying Windows-[SS+CR], Menu-[CS+CR], BackSpace-[CS+0], Insert-[SS+W], Delete-[CS+9], Home-[SS+Q], End-[SS+E], PageUp-[CS+3], PageDown-[CS+4]

Сокращения: CS=Caps Shift, SS=Symbol Shift, SP=Space, CR=Enter.

**Заказ*:**

- Голая плата 50 рублей;
- Микроконтроллер 70 рублей;
- Плата в сборе 200 рублей;
- Пересылка любого пункта 50 рублей.

Адрес для почтового перевода:

Россия, 620147, г. Екатеринбург, а/я 95,
 Каримову Камиллю Харинчановичу.

В почтовом переводе указывайте номер раскладки. По умолчанию прошивается раскладка № 1.

* Цены действительны до 1.01.2006 г.

Этюды

Модификация шрифта

Предлагаем вашему вниманию, несколько интересных процедур, которые «превращают» стандартный шрифт из ПЗУ, в шрифт, пригодный для печати 64-х символов в строке (один символ - 4x8 пикселей).

Вообще-то, все эти процедуры были написаны специально для наших демо 512 байт, и работали только при определенных условиях.

Но во время подготовки данной статьи, мы решили доработать их и сделать более универсальными.

```

FONT EQU #7000;АДРЕС
; РАЗМЕЩЕНИЯ
; ШРИФТА (2048 БАЙТ)
ORG #6000
LD HL, FONT+32
LD DE, #3D00
CY LD BC, #300
LD A, (DE)
RLA
CY2 RLA
JR NC, CY3
RL C
RLA
JR CY4
CY3 RLA
RL C
CY4 DJNZ CY2
LD (HL), C
INC H
INC DE
LD A, H
CP FONT/256+8
JR NZ, CY
LD H, FONT/256
INC L
JR NZ, CY

```

Некоторые новосозданные символы, выглядят немного ко-

риво, следующая программка их подправит.

```

LD HL, FTAB
LD DE, FONT+512+32
DORF LD B, (HL)
INC HL
LD C, (HL)
INC HL
INC DE
DJNZ $-1
LD A, (DE)
XOR C
LD (DE), A
LD A, (HL)
OR A
JR NZ, DORF

```

Посмотреть получившийся шрифт (**см. рисунок внизу страницы**), можно перебросив его на экран, LDIR'ом. HL, font, DE, #4000, BC, 2048, LDIR.

И напоследок, третья процедура, при помощи которой, вы сможете напечатать текст на экране (HL, адрес экрана (#4000), DE, адрес текста, CALL wyw):

```

WYW LD A, (DE)
INC DE
OR A
RET Z
CP 32
JR NC, $+4
LD A, 32
LD C, A
LD A, (DE)
INC DE
PUSH AF, HL, DE
CP 32
JR NC, $+4
LD A, 32
LD E, A
LD B, FONT/256
LD D, B

```

```

LD A, (BC)
LD (HL), A
LD A, (DE)
RLD
INC H
INC D
INC B
LD A, B
CP FONT/256+8
JR NZ, $-11
POP DE, HL, AF
INC L
OR A
JR NZ, WYW
RET

```

Ниже находится таблица для второй процедуры. Вы можете поэкспериментировать с BIN числами, изменяя их, вы измените вид некоторых созданных символов.

```

FTAB DB 5, %000000100
DB 1, %000000011
DB 4, %000000110
DB 20, %000000110
DB 15, %000000010
DB 30, %000000011
DB 8, %000000101
DB 1, %000000001
DB 206, %000000001
DB 23, %000000011
DB 13, %000000101
DB 192, %000000010
DB 22, %000000110
DB 22, %000000001
DB 211, %000000001
DB 7, %000000110
DB 250, %000000100
DB 4, %000000110
DB 20, %000000110
DB 49, %000000101, 0

```

Весь код занимает 145 байт (все три процедуры + таблица).

Алексей ЧЕРВОВ,
Владислав ЮВЖЕНКО,
г. Красноярск

```

! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _
a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~ `

```

Объявления

Покупаем качественное авторское ПО для дистрибуции (гонорар до 100\$ USD).

Обращаться: на адрес редакции.

Приглашаем к сотрудничеству авторов материалов для газеты. В качестве гонорара - бесплатный экземпляр газеты, некоторые скидки на нашу продукцию.

Обращаться: на адрес редакции.

Купим панельки DIP-40 с нулевым усилием.

Обращаться: на адрес редакции.

Куплю сменный блок интерфейса «Centronics» для принтера «Robotron».

Звонить по телефону (34370) 6-07-97 в субботу и воскресенье.

Куплю ПЗУ с русской про-

шивкой для принтера «Epson FX 1050».

Обращаться: 412302, Саратовская обл., г. Балашов, ул. Красина, д. 82, Илясов Евгений Витальевич.

Куплю книгу «Диалекты Бейсика для ZX Spectrum», изд. «Питер», 1992 г.

Обращаться: 412302, Саратовская обл., г. Балашов, ул. Красина, д. 82, Илясову Е.В.

Как купить «Абзац»?

Если вы проживаете на территории **России**, необходимо выслать почтовый перевод в размере: **25 рублей** за один экземпляр газеты. Адрес для почтового перевода: **160035, Россия, г. Вологда, а/я 136, Шушкова Александру Дмитриевичу.**

Если вы проживаете на территории **Украины**, необходимо выслать почтовый перевод в размере: **3 гривен (\$0.6)** за один экземпляр газеты. Адрес для почтового перевода: **79022, Украина, г. Львов-22, а/я 798, Селеву Валерию Анатольевичу.**

Если вы проживаете на территории **Белоруси**, необходимо выслать почтовый перевод в размере: **1400 белорусских рублей (\$0,7)** за один экземпляр газеты. Адрес для почтового перевода: **220094, Беларусь, г. Минск, а/я 218, Баглаю Андрею Николаевичу.**

В разделе «Для письменных сообщений» укажите, например, № 26 (1). Это будет означать, что вам нужен двадцать шестой номер газеты в одном экземпляре.

Убедительная просьба, пишите свой обратный адрес печатными буквами, а так же указывайте полностью свои фамилию, имя, отчество.

Колонка редактора

И вновь приходится начинать с извинений перед читателями, которые хотели увидеть в этом номере свои статьи, но по-прежнему не нашли их. Отчаиваться не стоит, все материалы постепенно будут опубликованы в последующих номерах газеты.

Недолго «продержались» игры конкурса «Твоя игра-2004» на руках законных обладателей. Игры снова опубликованы без нашего согласия на том же сайте, что и в прошлый раз.

За последний год в редакцию по обычной почте пришло порядка 10-12 писем. Корреспондентов, написавших их, в два раза меньше. Люди совершенно отвыкли пользоваться услугами почты. Корреспондентов пишущих статьи становится все меньше и меньше. А тех, кто мог бы написать статью на заказ, по-моему, и вовсе не осталось. Если все же такие люди найдутся, то для них я смогу предоставить список желаемых статей.

Ну а самый печальный факт, который можно констатировать - спектрумисты (и не только) постепенно теряют всякий интерес к простому письменному общению. Возражения о том, что на смену обычному письму пришли sms, e-mail не убедительны, т.к. на этих способах общения люди стараются экономить.

Издается
с октября 2000 года



ГАЗЕТА ДЛЯ СПЕКТРУМИСТОВ



ИЗДАТЕЛЬ

Perspective group

Редактор

Александр ШУШКОВ

Дизайн и верстка

Александр ШУШКОВ

Адрес для писем

160035, Россия,
г. Вологда, а/я 136,
Шушкову
Александру
Дмитриевичу

Телефон: (8172) 75-96-83

e-mail: axor@mail.ru

В оформлении газеты
использован рисунок
Сергея ЗАЛЕСОВА (Rion).

Редакция не всегда разделяет точку зрения авторов отдельных публикаций. За содержание рекламных объявлений редакция ответственности не несет. При перепечатке материалов необходимо сделать ссылку на газету «Абзац».

Тираж 100 экз.

Номер

подписан

в печать

9 августа 2005 г.



ДАВАЙ НА КОНКУРС

