

NedoPC is not PC

Nedo PC 5

ПЯТЫЙ НОМЕР

СПЕЦИАЛЬНЫЙ ВЫПУСК

2008

5 лет вместе
www.nedopc.com



DiHalt
2008

Z-Controller ● «Цифровые» брелки ● Мобильные интерфейсы ● NeoGS

Содержание номера

Колонка редактора CHRV (chunin@mail.ru)	3
Универсальный периферийный контроллер для ZX-Spectrum с ZXBUS шиной: Z-Controller КОЕ (KingOfEvil@yandex.ru) Savelij (savelij@kaluga.ru)	4
Изготовление цифровых брелков CHRV (chunin@mail.ru)	14
Троичные исследования. Новости Mac Buster (mbr@nm.ru)	16
Новости мобильных интерфейсов Mac Buster (mbr@nm.ru)	18
История DiHalt демопати Vinnny (vinnny@aport.ru)	22
Интервью ТmК (один из организаторов DiHalt) интервью провел CHRV (chunin@mail.ru)	25
О больших программах, проблемах их наворачивания, отделении мух от котлет и конвейерах для этой цели Alone Coder (dmitry.alonecoder@gmail.com)	27
NeoGS LVD (lvd.mhm@gmail.com)	30

Номер подготовлен московским отделением NedoPC (www.nedopc.com) в июне 2008.
Вопросы и отзывы по номеру 5 журнала пишите на адрес: chunin@mail.ru .

Верстка журнала осуществлялась в операционной системе Windows с помощью системы подготовки текстов OpenOffice.Org и графического редактора Paint.Net.

Журнал распространяется бесплатно в бумажном виде (A5) и электронном виде (pdf).

Вопросы по журналам и их распространению пишите по следующим адресам:

Е-Mail: journal@nedopc.org Сайт: <http://journal.nedopc.org>



КОЛОНКА РЕДАКТОРА

К сожалению, не все получается делать так как хочется. Вот и этот номер задержался, но все же появился на свет.

Как-то незаметно прошло пять лет и можно подвести итог нашей любительской пятилетке. С первого взгляда вроде бы сделано немного, но все что было сделано, как говорится, от души.

На этот раз номер выпускается силами Московского отделения NedoPC и посвящен больше местным новостям. Пришлось взять этот процесс в свои руки, так как журнал спрашивают и даже требуют. Линейка журналов будет продолжаться и дальше - следующий очередной номер подготавливается Shaos'ом.

Увы, мы становимся все взрослее и все меньше времени остается для хобби, но зато с большей любовью и усердием мы достигаем результата.

В этом номере мы расскажем о Z-Controller'e и программировании карт памяти. Мне кажется, это самый перспективный вид носителей информации на текущий момент.

Также в номере будет много информации из различных областей, связанных не только с электроникой.

Наша группа активно поддерживает фестивали компьютерного творчества. Мы стараемся помогать не только спонсорским участием, но и принимаем непосредственное участие в организации фестивалей. Этот номер будет представлен на фестивале DiHalt2008. Я надеюсь, увижу там своих друзей, причем многие будут не только зрителями, но и участниками. Призываю Вас, читатель, активно участвовать в фестивалях и демопати, это хороший допинг для развития любительского электронного творчества.

Жизнь продолжается и я верю, что мы тоже будем расти и расширяться, наши идеи подхватит кто-то еще и продолжит работу над ними.

Успехов и до будущих встреч.

Редактор номера CHRV

Универсальный периферийный контроллер для ZX-Spectrum с ZXBUS шиной: Z-Controller

Авторы: *KOE¹, Savelij²*

Устройство представляет собой универсальный периферийный контроллер для ZX-Spectrum – совместимых компьютеров, оснащенных системной шиной ZX-BUS. В рамках контроллера реализована поддержка PS/2 мышки и клавиатуры, IDE-интерфейса и карты flash-памяти (SD, подключается по SPI-интерфейсу). При работе с портами ввода-вывода контроллера центральный процессор компьютера не приостанавливается для выполнения пустых тактов ожидания [wait], благодаря чему исключается возможность нарушения привязки к реальному масштабу времени в критичных программах (видеоэффекты, проигрывание цифровой музыки, работа с ЦАП/АЦП).

В предлагаемой вниманию читателя статье рассматривается аппаратная реализация узлов контроллера и описывается методика работы с SD-картой с помощью используемого в устройстве интерфейса SPI.

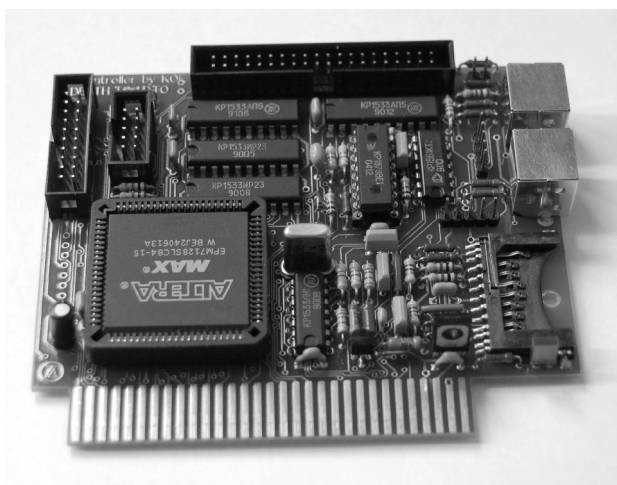


Рисунок 1: Z-Controller

Устройство реализовано в виде периферийной платы, устанавливаемой в один из слотов системной шины ZX-BUS. Конструктив контроллера разрабатывался прежде всего с ориентацией на его использование совместно с компьютером “Pentagon-1024sl” версии 2.2 и, как и конструктив компьютера, ориентирован на АТ-корпус. Для удобства подключения при использовании другого типа компьютера или корпуса предусмотрены отдельные штырьковые 5-контактные соединители для клавиатуры и мышки, разводка контактов в которых соответствует разводке контактов в аналогичных соединителях на материнских платах компьютеров IBM PC.

Рассмотрим вначале аппаратную часть контроллера. Его принципиальная электрическая схема показана на рис.2. В основе схемы лежит связка между микроконтроллером KP1878BE1 (производства ОАО “Ангстрем”, г. Зеленоград) DD2 и ПЛИС EPM7128SLC84 (ф. Altera) DD3. В контроллере реализован PS/2 интерфейс и вся предварительная обработка информации, принимаемой от мышки и клавиатуры и передаваемой далее в ПЛИС. Прием и обработка скан-кодов клавиатуры происходит по прерываниям от линии CLK клавиатуры. Обработка заключается в перекодировке скан-кодов в соответствии с форматом данных Spectrum-клавиатуры. При этом в ОЗУ контроллера формируется вектор из 5 байтов (40 бит), таким образом, что каждый бит соответствует состоянию одной из клавиш ZX-клавиатуры (вспомним, что ZX-клавиатура представляет собой матрицу из 5x8 клавиш). Как нетрудно догадаться, такое представление данных ZX-клавиатуры дает исчерпывающую информацию о ее состоянии.

Мышка опрашивается по прерываниям от встроенного в контроллер таймера. При

1 Сайт автора pentagon.nedopc.com

2 Email автора savelij@kaluga.ru

работе с мышью используется так называемый "Remote mode", в котором мышка передает данные о своем текущем состоянии после получения команды запроса статуса. После получения такой команды мышь возвращает код подтверждения (дабы дать понять хосту, что она находится в рабочем пока еще состоянии и адекватно воспринимает внешние воздействия) и 3 байта, соответствующие координатам X и Y указателя мышки и состоянию ее клавиш. Эти данные в контроллере также проходят перекодировку, в результате которой получаются 3 байта данных, записываемых непосредственно в порты ввода-вывода мышки. Эти 3 байта добавляются к вектору данных клавиатуры и полученный вектор длиной в 8 байтов передается в ПЛИС каждый раз при обработке прерывания от таймера микроконтроллером, либо при нажатии какой-либо клавиши. Для работы с мышкой со стороны Spectrum-а используются стандартные порты Kempston mouse.

Внутри ПЛИС реализованы порты ввода-вывода мышки, матрица клавиатуры, дешифратор портов IDE и SPI интерфейс, работа которого будет описана ниже.

Разъем X6 предназначен для подключения программатора ПЛИС. Незадействованные выводы ПЛИС подключены к разъему X7. Также к разъему подключены сигналы Magic0 и Magic1, предназначенные для соединения с соответствующими цепями в компьютере Pentagon-1024sl 2.2, для эмуляции замыкания кнопки 'MAGIC' клавишей F11.

При включении компьютера с установленным контроллером механическая клавиатура будет заблокирована, вместо нее можно использовать PS/2 клавиатуру.

Т.к. в последнее время развелось достаточно большое количество PS/2 мышек с различной разрешающей способностью, то в алгоритме предварительной обработки информации в микроконтроллере предусмотрена процедура масштабирования координат указателя

мышки. Нажимая на PS/2 клавиатуре клавиши F2 и F3, можно менять масштабный коэффициент для координат (и, как следствие, скорость перемещения по экрану указателя мышки). Для сохранения текущего значения масштабного коэффициента во flash-память микроконтроллера достаточно нажать F1. При включении питания или после сброса контроллер автоматически восстанавливает сохраненное значения масштабного коэффициента.

IDE интерфейс в контроллере полностью (аппаратно и программно) совместим с контроллером IDE Вячеслава Скутина (Nemo). Т.к. ничего нового в этом смысле не сделано, не будем подробно разбирать его функционирование. Скажем лишь, что внутри ПЛИС находится дешифратор его портов ввода-вывода, а буферные микросхемы (DD4 ... DD7), с помощью которых сделан обмен информацией между 16-разрядным IDE интерфейсом и 8-разрядной шиной данных Z80 последовательно по 8 бит, выведены за пределы ПЛИС из соображений экономии ее ресурсов.

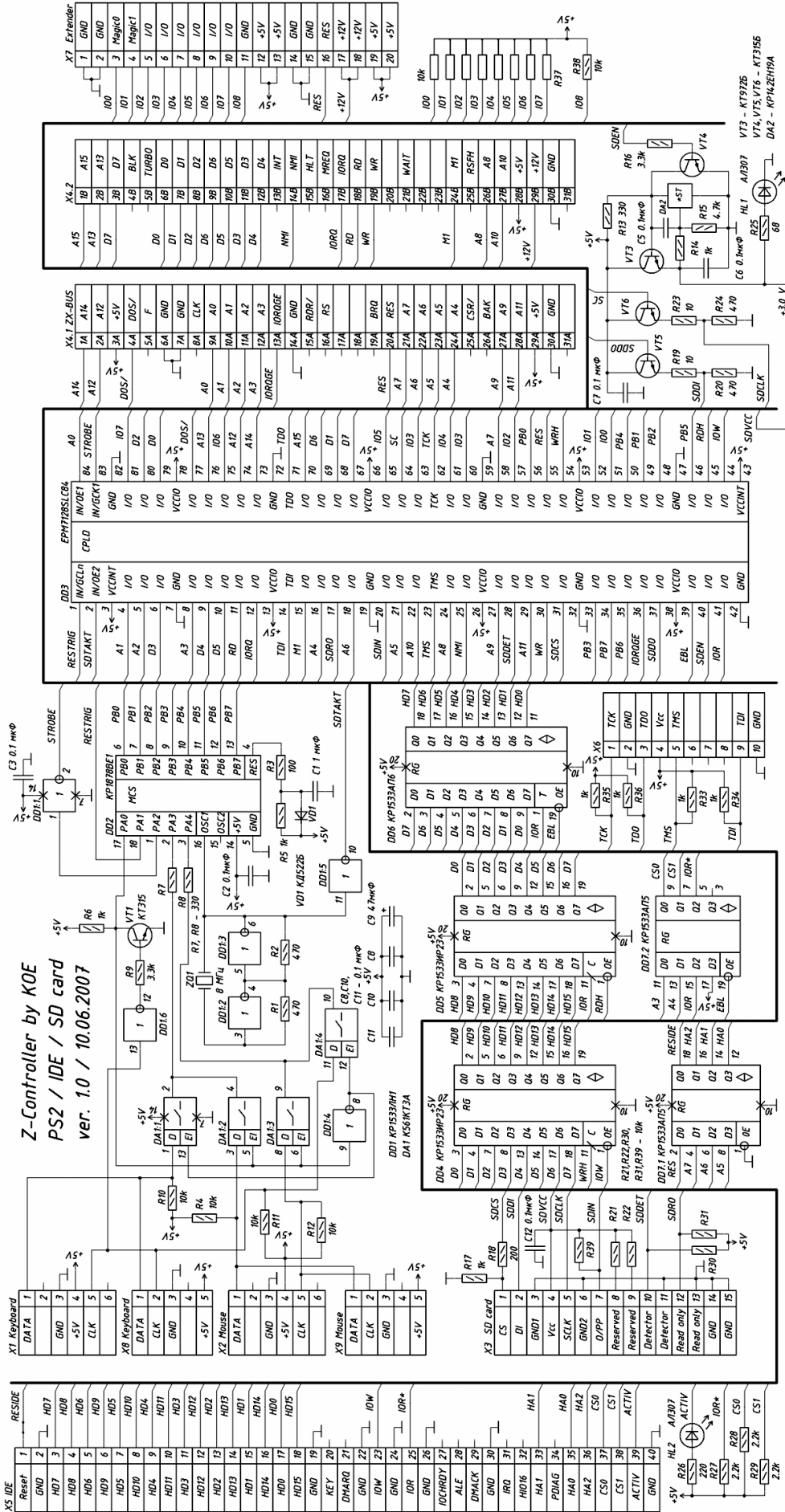
Для подключения SD-карты в ZC реализован последовательный интерфейс передачи данных SPI. Программно со стороны Спектрума интерфейс доступен с помощью двух портов ввода-вывода: порта конфигурации 77h и порта данных 57h. Назначение битов в порте 77h следующее:

На запись:

bit 0 - питание SD-карты (0 - выключено, 1 - включено)
bit 1 - управление сигналом CS
bit 2..7 - не используются

На чтение:

bit 0 - если 0 - SD-карта установлена, 1 - SD-карта отсутствует
bit 1 - если 1 - то на карте включен режим Read only, если 0 - режим Read only не включен
bit 2..7 - не используются.



Порт данных 57h используется как на запись, так и на чтение для обмена данными по SPI-интерфейсу. Временная диаграмма передачи байта данных по SPI показана на рисунке 2.

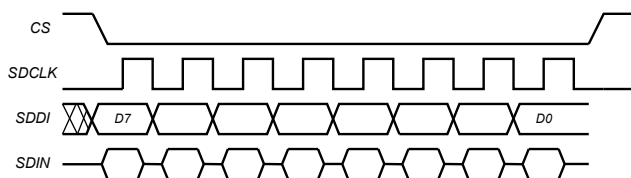


Рисунок 2: Передача данных по SPI

Тактирование (сигнал SDCLK) осуществляется автоматически при записи какого-либо значения в порт 57h. При этом формируются 8 тактовых импульсов на выходе SDCLK, на выход SDDI поступают данные последовательно от старшего бита к младшему с каждым фронтом сигнала SDCLK. Период следования тактовых импульсов составляет 125нс.

При чтении из порта 57h также автоматически производится тактирование. Буферный регистр порта 57h, используемый при чтении, заполняется данными со входа SDIN последовательно от старшего бита к младшему с каждым фронтом сигнала SDCLK.

На транзисторах VT5 и VT6 сделана схема согласования уровней между ПЛИС и SD-картой.

Немного о картах памяти SD и файловой системе.

Карты памяти SD (Secure Digital) является одной из разновидностей флэш-памяти, такие же как и обычные USB-Flash накопители, но основной ее особенностью является ставка на защиту записанной информации.

Разновидностей SD карт на сегодняшний день доступно несколько видов. Различаются они как по физическому размеру, так и по объему. Основным является стандартный размер, для карт размера Mini и Micro существуют переходники до стандартного размера.

Различий по объему всего два: до 2Гб включительно карта называется стандартной и не несет на себе никаких дополнительных указаний кроме объема. Карты начиная с 4Гб обязательно должны нести на себе гордую надпись SDHC (SD High Capacity) и Class 2 (4 или 6), цифра указывает скорость передачи данных в мегабайтах. Если на карте с объемом 4Гб и более нет таких надписей, то карта просто неправильно маркирована.

Набор внутренних команд зависит от спецификации и режима работы. На сегодняшний день последняя версия спецификации имеет номер 2.0, датирована 25 сентября 2006 года и существует на английском языке (возможно и на каких-то других языках), перевода на великий и могучий не обнаружено. Согласно этой спецификации объем карты до 2 Гигабайт включительно относится к стандартным по объему. Размер сектора можно менять программно от 1 до 512 байт, по умолчанию размер сектора равен 512 байтам. Карты объемом от 4 до 32 Гигабайт (ограничение этой версии спецификации на объем) имеют фиксированный размер сектора равный 512 байтам и не может быть изменен.

Карта может работать в одном из двух режимов: непосредственно SD режим и SPI. Для нас наиболее интересен режим SPI, который и реализован в Z-controller-e. Количество команд в SPI режиме значительно меньше, чем в режиме SD. На текущий момент написан и отлажен драйвер для работы с картами любого объема. Не проверена работа с картами калибра Mini и Micro ввиду их отсутствия, но, скорее всего, они будут работать.

Вообще, спецификаций существует несколько. В свободном доступе была найдена только достаточно полная спецификация физического уровня с некоторыми удаленными главами по которой и был написан драйвер. Не была обнаружена спецификация файловой системы, которую весьма интересно было бы изучить. Дело в том, что после изучения описания файловой

системы ФАТ можно по объему карты предположить, какая именно ФАТ используется на карте данного объема по умолчанию. Так вот карты до 2Гб включительно без проблем вписываются в размер ФАТ16 (возможно отформатировать карту и в ФАТ32, защита на это никак не реагирует) и продаются отформатированными именно в ФАТ16. Достаточно много устройств, использующих SD карты, имеют ограничение на объем, чаще всего 0,5-1Гб, возможно до 2Гб. Раньше я этого ограничения не понимал, но не имея спецификации файловой системы могу предположить, что применение ФАТ32 требует наличие большего объема встроенной памяти в устройстве для работы с этой файловой системой. А устройств рассчитанных на применение ФАТ32 на данный момент или нет вообще или их достаточно мало, хотя это уже возможно и изменилось. А карты объемом 4Гб и выше могут нести на себе только ФАТ32. Это подтвердилось приобретением карты SDHC объемом 4Гб. Но удивило не наличие на этой карте ФАТ32, а размер кластера. Фактически, размер кластера на свежеприобретенной карте объемом 4Гб при формате ФАТ32, равен 64 сектора (32Кб), в то время как на карте объемом 1Гб при формате ФАТ16 размер кластера равен 32 сектора (16Кб). При таких размерах кластера на ФАТ32 потери свободного места будут больше, чем на ФАТ16. Что и как прописано в спецификации файловой системы для SD карт остается только гадать и не понятно, почему сделан такой огромный размер кластера. Чтобы уменьшить потери свободного места карты объемом 4Гб и выше придется переформатировать с уменьшением размера кластера. После форматирования через кардридер карта объемом 4Гб из ФАТ32 опять в ФАТ32 размер кластера уменьшился до 8 секторов (4Кб).

Кстати, с разрядностью ФАТа связан один нюанс. Если между ФАТ12 и ФАТ16, кроме разрядности и некоторых изменений в описателе разницы особой нет, то у ФАТ32

структура описателя несколько сложнее. Самое главное, что из-за размера ФАТ таблиц в структуру описателя введен добавочный сектор, в который заносится информация о количестве свободных кластеров и номере кластере, с которого драйвер должен искать свободные кластера для записи. А с учетом того, что при записи на ФАТ этот сектор должен постоянно обновляться, а количество записей на карту ограничено количеством 200-300 тысяч, то срок жизни этого сектора будет значительно меньше срока жизни самой карты при использовании на том же пц. И не имеет особого значения, что у этого сектора есть копия. Копия-то как раз и есть, но всего лишь копия и признаков ее использования той же виндой не заметно. А работа с SD картой через кардридер выявила еще и то, что тот же XP с ФАТ-таблицами работает с ошибкой. А на Спектруме, если карта достаточно сильно забита файлами, приходится свободный кластер искать с самого начала таблицы, что может занять довольно большое время (для карт объемом 1Гб, форматированной как ФАТ32 и заполненной чуть больше половины время поиска в турбо-режиме до 7 секунд). Если карта будет использоваться для записи только Спектруме, то эта ошибка особого значения иметь не будет.

Описание драйвера.

При написании драйвера использовалось описание инициализации от создателя Хард-тапер и исходники с сайта www.zxbada.bbk.org для работы с MMC картой. Драйвер писался из расчета поддержки только SD карт и не поддерживает MMC карты по причине отсутствия как самих карт для проверки, так и спецификаций на них в свободном доступе. При попытке использования MMC карт с большой долей вероятности драйвер будет банально зависать. О причинах - в тексте драйвера.

Текст драйвера снят с рабочего исходника и является полностью рабочим. В этом виде

драйвер занимает около 530 байт. Данный драйвер поддерживает карты любого объема в пределах текущей спецификации, определения типа карты делается во время исполнения команды чтения/записи.

Размер сектора задается командой с кодом 16 и возможен только для стандартных карт. По умолчанию размер сектора равен 512 байт и изменение его в драйвере не предусмотрено в связи с поддержкой карт калибра SDHC (4 Гигабайт и более), у которых изменение размера сектора не возможно. У многосекторных команд чтения/записи нет счетчика количества секторов, остановка чтения/записи осуществляется соответствующими командами остановки.

Команд чтения/записи есть две разновидности: для работы с одним сектором и многосекторная. Зачем сделано это разделение понять трудно, для односекторной работы можно использовать и многосекторную команду, я это проверил, но для себя оставил как есть.

На всех SD картах есть область между MBR и началом раздела, которая, по всей видимости, используется при шифровании содержимого карты. Размер этой области зависит от объема карты и производителя.

Размер любой команды для карты равен 6 байтам (48 бит). Вся команда передается в саму карту от старшего бита до младшего. Формат команды:

Номер бита:	47	46	45-40	39-8	7-1	0
Размер, бит:	1	1	6	32	7	1
Значение:	0	1	X	X	X	1
Описание:	Стартовый бит	Направление передачи	Индекс команды	Аргумент	CRC7	Стоповый бит

После подачи команды первый считанный байт (не равный #FF) является ответом карты и если не равен нулю (команда будет выполнена), то содержит биты ошибки. В драйвере код ошибки чаще всего игнорируется. По спецификации карта выдает несколько разновидностей ответов различающихся длиной, в драйвере используется только первый байт ответа.

Коды ошибок по битам. Возможно, я где-то не так перевел, поэтому привожу

оригинальное описание битов ошибок:

- 7-всегда 0
- 6-parameter error (ошибочный параметр)
- 5-address error (неправильный адрес блока)
- 4-erase sequence error
- 3-com crc error (ошибка CRC команды (при отключенном CRC не должна появляться))
- 2-illegal command (неизвестная команда)
- 1-erase reset
- 0-in idle state (карта находится в режиме инициализации и недоступна)

Неправильный адрес блока появляется только при превышении максимального номера сектора. Неизвестная команда скорее всего будет появляться в ответ на команды которые не поддерживаются. В режиме SPI многие команды SD-режима недоступны.

Пример использования драйвера:

```
CALL COM__SD
DB 1
```

-включение и инициализация карты памяти, на выходе A - смотри коды возвращаемых ошибок,

```
LD HL, АДРЕС ЗАГРУЗКИ
LD BC, СТАРШИЕ 16 БИТ НОМЕРА СЕКТОРА
LD DE, МЛАДШИЕ 16 БИТ НОМЕРА СЕКТОРА
LD A, КОЛИЧЕСТВО БЛОКОВ
CALL COM__SD
DB 3
```

-чтение заданного количества секторов по заданному адресу, начиная с заданного сектора, на выходе не забываем проверять код ошибки в A,

```
CALL COM__SD
DB 1
```

-выключаем питание карты.

```
;Драйвер SD карты
;LAST UPDATE 19.02.2008 savelij
```

```
;Первые две функции входных параметров не имеют.
;Входные параметры общие:
;HL-адрес загрузки в память/запись из памяти
;BCDE-32-х битный номер сектора
;A-количество секторов (сектор=512 байт),
;только для многосекторного чтения/записи
;Выходные параметры:
;HL-адрес следующего байта после окончания
;чтения/записи
;BCDE-не изменяется
;A-смотри коды ошибок, альтернативный A портится
;Ошибки, выдаваемые на выходе драйвера:
```

NedoPC#5

```

;A=0-инициализация прошла успешно или команда
;выполнена успешно
;A=1-карта отсутствует или не ответила
;A=2-карта защищена от записи
;A=3-попытка записи в сектор 0 карты

;адреса портов контроллера
P_DATA EQU #57 ;порт данных
P_CONF EQU #77 ;порт конфигурации

;некоторые коды команд SD карт, используемые в
;драйвере
CMD_12 EQU #4C ;STOP_TRANSMISSION остановка
;чтения
CMD_17 EQU #51 ;READ_SINGLE_BLOCK чтение
;одиночного блока
CMD_18 EQU #52 ;READ_MULTIPLE_BLOCK чтение
;нескольких блоков
CMD_24 EQU #58 ;WRITE_BLOCK запись одиночного
;блока
CMD_25 EQU #59 ;WRITE_MULTIPLE_BLOCK запись
;нескольких блоков
CMD_55 EQU #77 ;APP_CMD последующая команда
;дополнительная
CMD_59 EQU #7B ;CRC_ON_OFF команда принуди-
;тельного отключения CRC16
ACMD_41 EQU #69 ;SD_SEND_OP_COND дополнительная
;команда сброса

;ОБЩАЯ ТОЧКА ВХОДА ДЛЯ РАБОТЫ С SD
;код команды должен быть после команды вызова
;драйвера, без комментариев
COM_SD EX (SP),HL
    EX AF,AF
    LD A,(HL)
    INC HL
    EX (SP),HL
    PUSH HL
    PUSH DE
    LD L,A
    LD H,0
    ADD HL,HL
    LD DE,TABL_SD
    ADD HL,DE
    LD E,(HL)
    INC HL
    LD D,(HL)
    EX DE,HL
    POP DE
    EX (SP),HL
    EX AF,AF
    RET

;коды и описания функций
TABL_SD DW SD_INIT ;0-инициализация карты
        DW SD_OFF ;1-отключение питания карты
        DW RDSINGL ;2-чтение одного сектора
        DW RDMULTI ;3-чтение нескольких
        ;секторов
        DW WRSINGL ;4-запись одного сектора
        DW WRMULTI ;5-запись нескольких
        ;секторов

;подпрограмма инициализации
SD_INIT CALL CS_HIGH;включаем питание карты при
;не выбранной карте
        LD BC,P_DATA;сигнал выбора карты равен
1
        LD DE,#20FF ;записываем согласно
        ;спецификации в порт
        OUT (C),E ;много единиц, главное
        ;отличие: количество
        DEC D ;единиц несколько увеличено
        ;и больше, чем
        JR NZ,$-3 ;требуется по спецификации

LD A,30 ;будем переводить карту в
;режим SPI до 30 раз
EX AF,AF
ZAW001 LD HL,CMD00 ;даем команду сброса, эта
;команда после
CALL OUTCOM ;включения питания пере-
;водит карту в режим SPI
CALL IN_OOUT;читаем ответ
EX AF,AF
DEC A
JR Z,ZAW003 ;если карта 30 раз не
;ответила, то карты нет
EX AF,AF
DEC A
JR NZ,ZAW001
LD HL,CMD08 ;запрос на поддерживаемые
;напряжения
CALL OUTCOM ;команда поддерживается,
;начиная со спецификации
CALL IN_OOUT;версии 2.0 и только SDHC
;картами
IN H,(C) ;в A=код ответа карты
IN H,(C) ;считываем 4 байта длинного
;ответа
IN H,(C) ;но не используем
IN H,(C)
LD HL,0 ;HL=аргумент для команды
;инициализации
BIT 2,A ;если бит 2 установлен, то
;карты стандартная
;здесь содержимое A можно
;передать для подпрограммы SECM200
;смотри саму подпрограмму SECM200
JR NZ,ZAW006
LD H,#40 ;если сброшен, то карта
;SDHC
ZAW006 LD A,CMD_55 ;запускаем процесс
;внутренней инициализации
CALL OUT_COM;для карт MMC здесь должна
;быть другая команда
CALL IN_OOUT;соответственно наличие в
;слоте MMC-карты
LD A,ACMD_41;вызовет зависание
;драйвера, от применения
;общей команды запуска
OUT (C),A ;инициализации я отказался
OUT (C),H ;бит 6 установлен для
;инициализации SDHC карты
;для стандартной сброшен
OUT (C),L
OUT (C),L
OUT (C),L
LD A,$FF
OUT (C),A
CALL IN_OOUT;ждем перевода карты в
;режим готовности
AND A ;время ожидания примерно
;1 секунда
JR NZ,ZAW006
ZAW004 LD A,CMD_59 ;принудительно отключаем
;CRC16
CALL OUT_COM
CALL IN_OOUT
AND A
JR NZ,ZAW004
ZAW005 LD HL,CMD16 ;принудительно задаем
;размер блока 512 байт
CALL OUTCOM
CALL IN_OOUT
AND A
JR NZ,ZAW005
;включение питания карты при снятом сигнале
;выбора карты
CS_HIGH PUSH DE

```

```

PUSH BC
LD E,3
LD BC,P_CONF
OUT (C),E ;включаем питание, снимаем
;выбор карты

LD E,0
LD C,P_DATA ;обнуляем порт данных
OUT (C),E ;делаеть, просто последний
;записанный бит всегда 1,
;а при сбросе через вывод
POP BC ;данных карты напряжения
POP DE ;попадает на вывод питания
RET ;карты и светодиод на
;питании подсвечивается

;возврат по не ответу карты с кодом ошибки 1
ZAW003 CALL SD__OFF ;карта не найдена,
;выключаем питание

LD A,1
RET

;выключение питания карты
SD__OFF XOR A
OUT (P_CONF),A
OUT (P_DATA),A
RET

;выбираем карту сигналом 0
CS__LOW PUSH DE
PUSH BC
LD E,1
LD BC,P_CONF
OUT (C),E
POP BC
POP DE
RET

;запись в карту команды с неизменяемым
;параметром из памяти
OUTCOM CALL CS__LOW
PUSH BC
LD BC,P_DATA
OUTI ;передаем 6 байт команды из
;памяти
OUTI ;адрес команды в HL
OUTI
OUTI
OUTI
OUTI
POP BC
RET

;запись в карту команды с нулевыми аргументами
;A-код команды, аргумент команды равен 0
OUT_COM PUSH BC
CALL CS__LOW
LD BC,P_DATA
OUT (C),A
XOR A
OUT (C),A ;пишем 4 нулевых байта
;аргумента

OUT (C),A
OUT (C),A
OUT (C),A
DEC A
OUT (C),A ;пишем пустой CRC7 и
;стоповый бит

POP BC
RET

;запись команды чтения/записи с номером сектора
;для карт стандартного размера при изменяемом
;размере сектора

;номер сектора нужно умножать на его размер,
;для карт SDHC размер
;сектора не требует умножения
SECM200 PUSH HL
PUSH DE
PUSH BC ;BCDE=номер сектора
PUSH AF
PUSH BC
LD HL,CMD08 ;сюда можно передать код
;ответа из подпрограммы
CALL OUTCOM ;инициализации и еще раз
;проверив бит 2
CALL IN_OOUT;принять решение о типе
;карты и дальнейших
LD BC,P_DATA;действиях
LD H,A ;сохраняем код ответа
IN A,(C) ;пропускаем неиспользуемые
;байты

INC A
JR NZ,$-3
BIT 2,H ;проверяем на ошибку
POP HL ;теперь HLDE=номер сектора
JR Z,SECN200;если ошибки нет, то
;умножение номера сектора
EX DE,HL ;на размер сектора не
;требуется

ADD HL,HL
EX DE,HL
ADC HL,HL
LD H,L
LD L,D
LD D,E
LD E,0 ;сейчас
;HLDE = номер сектора * 512

SECN200 POP AF
CALL CS__LOW
LD BC,P_DATA
OUT (C),A ;записываем код команды
OUT (C),H ;записываем 4 байта
;аргумента
OUT (C),L ;аргумент - HLDE = номер
;сектора
OUT (C),D ;пишем от старшего байта
OUT (C),E ;до младшего
LD A,#FF
OUT (C),A ;пишем замену CRC7 и
;стоповый бит

POP BC
POP DE
POP HL
RET

;чтение ответа карты до 16 раз,
;если ответ не #FF-немедленный выход
IN_OOUT PUSH BC
PUSH DE
LD DE,#10FF
LD BC,P_DATA
IN_WAIT IN A,(C)
CP E
JR NZ,IN_EXIT
IN_NEXT DEC D
JR NZ,IN_WAIT
IN_EXIT POP DE
POP BC
RET

CMD00 DB #40,#00,#00,#00,#00,#95
;GO_IDLE_STATE команда сброса и перевода карты
;в SPI режим после
;включения питания
CMD08 DB #48,#00,#00,#01,#AA,#87
;SEND_IF_COND запрос поддерживаемых напряжений
CMD16 DB #50,#00,#00,#02,#00,#FF

```

NedoPC#5

```
;SET_BLOCKEN команда изменения размера блока
;на размер 512 байт
```

```
;читаем один сектор из карты в память
```

```
RD_SECT PUSH BC
      PUSH DE
      LD BC,P_DATA
      INIR ;читаем первые 256 байт
      INIR ;читаем следующие 256 байт
      IN A,(C) ;здесь считываем CRC16,
              ;выдаваемые
      IN A,(C) ;картой, но не используем
      POP DE ;замена INIR на кучу INI
              ;мало что дает,
      POP BC ;хотя я могу и ошибаться
      RET
```

```
;записываем один сектор из памяти в карту
```

```
WR_SECT PUSH BC
      PUSH DE
      LD BC,P_DATA
      OUT (C),A ;пишем код команды
      OTIR ;пишем первые 256 байт
      OTIR ;пишем следующие 256 байт
      LD A,#FF
      OUT (C),A ;записываем замену CRC16
      OUT (C),A ;равное #FFFF
      POP DE
      POP BC
      RET
```

```
;многосекторное чтение
```

```
RDMULTI EX AF,AF ;прячем счетчик секторов
      LD A,CMD_18 ;даем команду
              ;многосекторного чтения
      CALL SECM200
      EX AF,AF
RDMULT1 EX AF,AF
      CALL IN_OOUT;пропускаем код ошибки
      CP #FE ;и ждем токен готовности
              ;#FE для начала чтения
      JR NZ,$-5
      CALL RD_SECT;читаем сектор
      EX AF,AF
      DEC A
      JR NZ,RDMULT1;читаем сектора пока не
              ;обнулится счетчик
      LD A,CMD_12 ;по окончании чтения даем
              ;команду карте «СТОП»
      CALL OUT_COM;команда мультичтения не
              ;имеет счетчика и
      CALL IN_OOUT;должна останавливаться
              ;здесь командой 12
      INC A
      JR NZ,$-4 ;ждем освобождения карты
      JP CS_HIGH ;снимаем выбор с карты и
              ;выходим с кодом 0
```

```
;чтение одного блока
```

```
RDSINGL LD A,CMD_17 ;даем команду чтения одного
              ;сектора
      CALL SECM200
      CALL IN_OOUT;пропускаем код ошибки
      CP #FE ;и ждем токен готовности
              ;#FE для начала чтения
      JR NZ,$-5
      CALL RD_SECT;читаем сектор
      CALL IN_OOUT
      INC A
      JR NZ,$-4 ;ждем освобождения карты
      JP CS_HIGH ;снимаем выбор карты и
              ;выходим с кодом 0
```

```
;запись одиночного блока
```

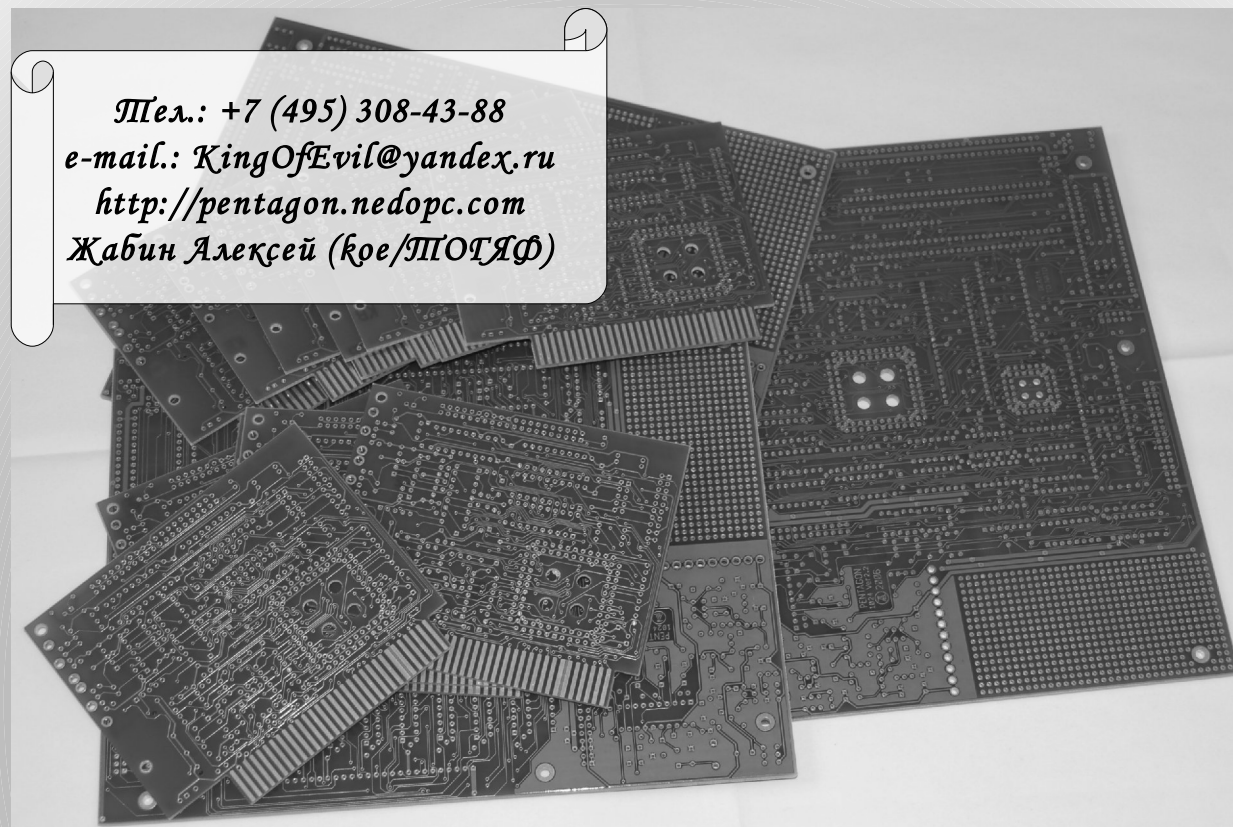
```
WRSINGL XOR A
      IN A,(P_CONF);проверяем защиту от
              ;записи
      AND 2 ;если защита включена
              ;выходим с кодом 2
      RET NZ
      LD A,B ;проверяем на попытку
              ;записи в нулевой
      OR C ;сектор, сделано на всякий
              ;пожарный
      OR D ;я пока экспериментировал
              ;одной карте MBR затер
      OR E
      LD A,3 ;если сектор нулевой,
              ;выходим с кодом 3
      RET Z
      LD A,CMD_24 ;даем команду записи одного
              ;сектора
      CALL SECM200
      CALL IN_OOUT;пропускаем код ошибки
      INC A
      JR NZ,$-4 ;и ждем освобождения карты
      LD A,#FE ;пишем стартовый токен,
              ;сам блок и пустое CRC16
      CALL WR_SECT
      CALL IN_OOUT
      INC A
      JR NZ,$-4 ;ждем освобождение карты
      JP CS_HIGH ;снимаем выбор карты и
              ;выходим с кодом 0
```

```
;многосекторная запись
```

```
WRMULTI EX AF,AF ;сохраняем счетчик блоков
      XOR A
      IN A,(P_CONF);как и в случае с записью
              ;одного сектора
      AND 2 ;проверяем защиту от записи
      RET NZ
      LD A,B ;и попытку записи в MBR
      OR C
      OR D
      OR E
      LD A,3
      RET Z
      LD A,CMD_25 ;даем команду
              ;мультисекторной записи
      CALL SECM200
      CALL IN_OOUT
      INC A
      JR NZ,$-4 ;ждем освобождения карты
      EX AF,AF
WRMULT1 EX AF,AF
      LD A,#FC ;пишем стартовый токен,
              ;сам блок и пустое CRC16
      CALL WR_SECT
      CALL IN_OOUT;пропускаем код ошибки
      INC A
      JR NZ,$-4 ;и ждем освобождения карты
      EX AF,AF
      DEC A
      JR NZ,WRMULT1;продолжаем пока счетчик
              ;не обнулится
      LD C,P_DATA
      LD A,#FD
      OUT (C),A ;даем команду остановки
              ;записи
      CALL IN_OOUT
      INC A
      JR NZ,$-4 ;ждем освобождения карты
      JP CS_HIGH ;снимаем выбор карты и
              ;выходим с кодом 0
```

Пустые платы ZX-Spectrum железа производства КОЕ

Тел.: +7 (495) 308-43-88
 e-mail.: KingOfEvil@yandex.ru
<http://pentagon.nedopc.com>
 Жабин Алексей (кре/ПРОТЯФ)



Разработанные и запущенные в мелкосерийное производство платы:

Пустая печатная плата **Pentagon-1024s1** (ver 2.2, разработка 2006 г.), полный комплект технической документации (DVD-ROM, содержащий принципиальную и монтажную схемы, краткое описание схемы и инструкцию по сборке и настройке, а также архив с разной полезной информацией, литературой, программами в TRD формате и т.д.)..... **800руб**

Пустая печатная плата **Z-Controller** (ver 1.0, разработка 2007 г.), комплект технической документации и файлы прошивок..... **300руб**

Платы делаются по 3-му классу точности в соответствии с ГОСТ 23752-79 на заводе в Зеленограде, имеется защитное масочное покрытие (зел.).

Анонс

В связи с желанием провести реинкарнацию некоторых ZX-проектов, готовится к выпуску пробный комплект плат Pentagon-48 (разработка 1989 г.). Платы полностью оригинальные, рисунок топологии восстановлен по сканированному изображению пустой платы производства прежних лет. Бережно сохранены все ошибки оригинала и, возможно, добавлены новые. Пробный комплект изготавливается без защитного масочного покрытия, платы планируется распространять по себестоимости производства. Ориентировочный срок выхода – июнь-июль 2008 г.

Порядок обработки заказов

Справки о наличии плат можно получить по телефону или по e-mail. Для жителей Москвы возможен самовывоз, в другие города платы отправляются по почте. Цены за пересылку по России составляют **180р.** для отправки не более 2 плат, **220р.** - для 3-5 плат. КОЕ распространяет только пустые печатные платы и, в связи с нехваткой времени, не занимается сборкой/настройкой плат и не осуществляет закупку и поставку радиокомпонентов.

Изготовление цифровых брелков

Автор: CHR³

Сообщество NedoPC, как прогрессивное и современное, не смогло обойти своим вниманием проблему утилизации "цифровых отходов" и предлагает свое решение.

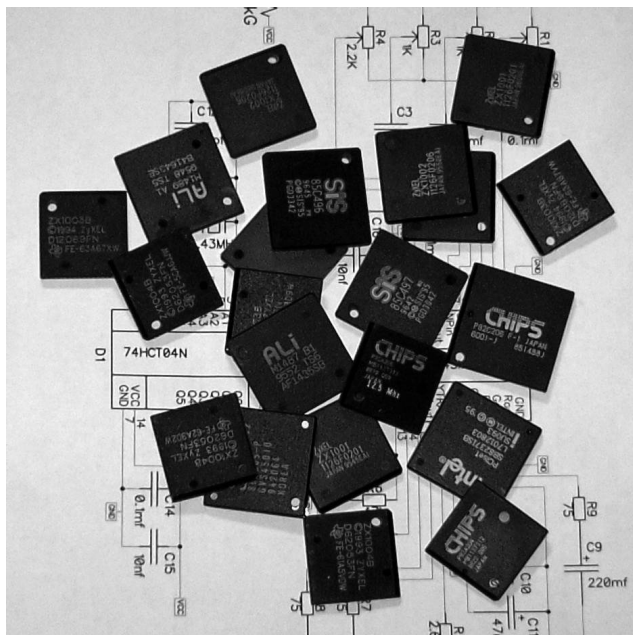


Рисунок 3: "Цифровые" брелки

Вступление

Для изготовления симпатичных брелков не требуется каких-то особых умений или знаний. Достаточно уметь управляться с некоторыми инструментами, которые я здесь перечисляю:

- Дрель (сверло диаметром 2.5мм или 3мм, и диаметром 5мм или 6мм);
- Строительный фен (400 градусов по Цельсию);
- Напильник.

Я надеюсь, что каждый из читателей достаточно владеет этими инструментами.

Конечно не стоит забывать о правилах безопасности:

- При работе с феном обязательно используйте перчатки -- это защитит Ваши руки от ожогов;
- При работе с дрелью не работайте на весу, держа деталь в руках. Зажимайте обрабатываемую деталь или укладывайте на рабочую поверхность - чтобы избежать лишних царапин и травм. Желательно использовать защитные очки во избежание поражения глаз крошкой;
- При работе с напильником, опять же старайтесь не держать деталь на весу.

Да не забываем найти несколько компьютерных плат, содержащих интересующие нас элементы. Для наших задач наиболее подходящими являются микросхемы в корпусе QFP или PLCC. И можно приступать к работе.

Выпаивание микросхем



Рисунок 4: Выпаивание микросхемы феном

Производить выпаивание лучше всего на открытой поверхности с доступом свежего воздуха. Это нужно потому-что платы покрыты лаком и выделяют крайне неприятный запах, кроме того припой частично испаряется и может содержать токсичный свинец. Естественно никаких горючих материалов в радиусе трех метров

3 Email автора chunin@mail.ru

быть не должно. Сразу предупрежу что в процессе выпайки и съема элемента с платы происходит разбрызгивание припоя, поэтому не стоит это делать дома например над ковром или паркетом.

Плату лучше всего закрепить под наклоном, чтобы легче было снимать элементы. Также приготовить какойнибудь инструмент для съема элементов, я обычно использую для этого либо пинцет либо отвертку с длинным жалом.

Включаем фен в режиме 400-600градусов (в общем больше и не нужно). Аккуратно обдуваем деталь в местах пайки и ждем пока припой не нагреется до необходимой температуры плавления. Обычно это видно по тому как припой начинает блестеть. После этого аккуратно либо поддеваем деталь отверткой и резким движением скидываем ее с платы, либо берем пинцетом и аккуратно снимаем микросхему с платы. Спустя две три минуты после отделения элементы от платы, он уже достаточно остынет и его уже можно брать в руки.

Плата освобожденная от элементов больше не представляет для нас интереса, и мы ее отправляем в помойку (Может быть кто-то предложит свой метод утилизации этой части).

Удаление выводов микросхемы



Рисунок 5: Удаление выводов

Для этой операции потребуется плоский слесарный напильник со средним номером насечки. Крупная насечка будет слишком

грубой для применяемой операции и испортит внешний вид изделия, а мелкая насечка не позволит быстро и эффективно освободить микросхему от выводов.

Зажимаем аккуратно микросхему в тисках, так как микросхемы имеют пластмассовый корпус и могут треснуть от большого усилия. Затем мощным движением напильника стачиваем выводы микросхемы. То же самое делаем для каждой из сторон микросхемы.

Если микросхема содержит острые края, то лучше обработать эти края напильником с мелкой насечкой (так называемой бархатной насечкой).

Сверление крепежного отверстия



Рисунок 6: Сверление отверстия

Для этой операции нам понадобится обычная дрель и сверла (желательно по металлу, так как плотность материала микросхемы очень высокая и сверла по дереву быстро затупятся).

Сперва тонким (2.5мм или 3мм) сверлом просверливаем отверстие в любом из краев микросхемы. Надо учитывать, что материал микросхемы достаточно хрупкий, поэтому от края отверстия до края микросхемы должен быть зазор порядка 2мм. Также нужно не применять сильного усилия на дрель, чтобы не отломить материал с обратной от сверла стороны микросхемы. Сверление лучше

производить при средних оборотах, так как при быстрых оборотах будет происходить сильный нагрев, и наша заготовка может разрушиться от внутренних натяжений.

После того как мы сделали отверстие, то неплохо бы обработать его края. Это лучше всего сделать сверлом большего диаметра (5мм или 6мм). Аккуратным кратковременным засверливанием мы удаляем острую кромку края отверстия (рекомендуется сделать с обеих сторон отверстия).

Заключение

Все наше изделие готово, надеваем его на крепежное кольцо и гордимся фирменным брелком перед своими коллегами. Проверено на практике, такой брелок мало кого оставляет равнодушным из тех кто связан с IT-бизнесом.

Можно покрыть изделие прозрачным лаком чтобы подольше сохранялась надпись (обычно она спустя полгода стирается).

Также можно использовать такой брелок как висячую в автомобиле, а девушке сделать элементом бус для вечеринки компании. Несомненно, Ваша фантазия подскажет еще множество применений для нашего изделия. Успехов.



Троичные исследования (новости)

Виртуальная троичная машина «Тунгуска»

В области троичных исследований наметился заметный сдвиг - в начале 2008 года был выпущен в свет эмулятор троичной машины, обладающий большим числом интересных особенностей. Проект, в рамках которого развивается эмулятор, носит название «Тунгуска». Он основан и в настоящее время развивается усилиями гражданина Швеции - Виктора Лофгрена. Еще в конце прошлого года он появился на форуме исследовательского портала Ternary.info с предложением опубликовать исходные тексты эмулятора троичной машины. Члены группы Ternary.info постарались оказать ему посильную помощь, предоставив имеющуюся англоязычную документацию, а так же исходные тексты библиотек реализующих набор элементарных арифметических операций с числами, записанными в уравновешенном троичном

коде. Так же был собран порт эмулятора, который позволил работать в операционной системе Microsoft Windows (изначально эмулятор собирался и работал под управлением ОС Linux). В начале года, после внесения в исходные тексты некоторых дополнений, была опубликована первая версия эмулятора.

«Тунгуска» имитирует работу троичной микро-ЭВМ обладающей достаточно развитыми возможностями, среди которых стоит отметить:

- наличие большого количества оперативной памяти;
- один текстовый и несколько графических режимов достаточно высокого разрешения предназначенных для вывода информации (один из этих режимов векторный);
- возможность сохранять результаты работы программ в образах гибких

дисков (ёмкость каждого из которых соответствует объёму оперативной памяти машины);

- поддержка полноценной клавиатуры;
- работа с мышью.

Архитектура процессора и набор его команд позаимствованы у весьма популярного некогда процессора MOSTEK 6502. В набор команд добавлены несколько специфических инструкций, облегчающих управление устройствами машины. Помимо эмулятора в состав входит ассемблер и несколько примеров программ для троичной машины.

Найти ссылки эмулятор «Тунгуска» можно в форуме сайта <http://www.ternary.info>.

Новый интернет-ресурс

Хочу обратить внимание всех заинтересованных на новый интернет-ресурс, посвященный исследованиям в области троичной информатике и вычислительной техники - <http://www.trinary.ru>. На сайте вы найдете следующие материалы:

- Статьи по троичной информатике.
- Симулятор первой троичной МЦВМ «Сетуль».
- Преобразователь чисел из одной системы исчисления в другую (двоичная, троичная, десятичная, восьмеричная).
- Кодировщик (реализация алгоритма экономного кодирования информации по методу Хаффмана в двоичной и троичной системах).
- Троичные часы. Концепт.
- Форум.
- Ссылки на тематические ресурсы.

Планы развития:

- Создание графического интерфейса симулятора Сетуни.
- Создание полнофункционального приложения троичных часов,

обновление скринсейвера.

- Развитие форума.
 - Создание библиотеки ссылок на статьи и книги доступные в интернете.
- Сотрудничество с ВНТБ.

Принимаются пожелания по расширению материалов. До скорой встречи! :)

Команда trinary.ru

Внимание! В целях обеспечения безопасности, на сайт можно зайти только с использованием безопасных веб-браузеров (FireFox, Opera, Safari), пользователи Internet Explorer не смогут увидеть материалы сайта.

Конференция по троичному коду и логике

Исследовательская группа Ternary.info приглашает всех интересующихся принять участие в конференции по теме практического применения уравновешенного троичного кода и троичной логики, которая состоится в конце июля в Москве. В рамках проводимой конференции планируется обсудить вопросы, относящиеся к истории исследований:

- текущее положение дел;
- печатные и электронные публикации по теме;
- существующие и разрабатываемые эмуляторы троичных машин и аппаратура, необходимая для их работы;
- разработка языков и сред программирования;
- построение оптимальных алгоритмов и поиск эффективных методов программирования;
- разработка логических элементов и оборудования на их основе.

Информация о проведении конференции на форуме сайта <http://www.ternary.info>.

Новости мобильных интерфейсов

Автор: Mac Buster⁴

Что нового в мире мобильных интерфейсов?

Как развивается эта область и каким образом ведущие разработчики заботятся о удобстве работы с мобильными устройствами?

Какие перспективы у новых платформ?



Почти каждому из нас время от времени приходится помимо компьютеров пользоваться мобильными устройствами, будь то сотовый или спутниковый телефон, карманный компьютер или терминал сбора и обработки данных, портативный терминал оплаты товаров и услуг пластиковыми платежными или кредитными картами. Не доводилось ли вам при заполнении 25-й формы с ужасом обнаруживать, что N форм назад вы кажется ввели неправильные данные? Или введя 29-значное кодовое слово и, подумав, что неплохо бы запомнить его в буфере обмена, с тем чтобы не набирать его заново на следующих 9 формах, вы с удивлением обнаруживаете отсутствие

⁴ Email автора mbr@nm.ru

возможности занести что-либо в буфер? А какие эмоции может вызвать тот факт что вернуться назад нельзя, так как в текущей форме не предусмотрена кнопка "Назад", или она есть, однако, нажав ее и вернувшись на предыдущий шаг оказываешься лицом к лицу с незаполненной формой? Или что в этой программе надо ткнуть стилусом в кнопку размер которой всего 5 на 5 точек? И это на устройстве с VGA экраном диагональю всего 2,4 дюйма.

Подобное положение вызывало и продолжает вызывать недовольство пользователей. Это не могло остаться незамеченным. Со временем стали появляться проекты целью которых было внесение разнообразных улучшений в стандартный интерфейс операционной системы устройства, призванных упростить и ускорить работу пользователей. Большинство подобных проектов исправляли один какой-либо просчет в проектировании взаимодействия человека и устройства, вроде упрощения запуска приложений, как это делают, к примеру, Spb Mobile Shell и Zumobi GUI. Особенность второго проекта заключается в том, что он реструктурирует размещение пиктограмм приложений на рабочем экране, позволяя владельцу устройства быстрее находить и запускать необходимое ему приложение. При этом процесс выбора сопровождается ненавязчивой вспомогательной анимацией и фактически одинаков как на смартфонах, где управление осуществляется с помощью цифровой клавиатуры и навигационного джойстика, так и на коммуникаторах и КПК, где для выбора используется сенсорный экран. Причём работа одинаково удобна как на экранах с невысоким разрешением и, соответственно, небольшой диагональю, так и на устройствах с высоким разрешением. Кроме того, каждый разработчик ПО может написать приложение, использующее подобную навигацию уже внутри

собственного интерфейса.

Помимо SPB Mobile Shell и Zumobi существует ещё один интересный подобный продукт – PointUI. Он предлагает пользователю не столько упростить работу с рабочим экраном, как это делают два предыдущих проекта, сколько заменяет его своим интерфейсом, запуская его поверх стандартного. В новом интерфейсе управление осуществляется без использования стилуса или клавиатуры – все операции производятся непосредственным касанием пальцами сенсорного экрана.

Zumobi, PointUI и очень многие подобные проекты распространяются авторами совершенно бесплатно. Установив несколько подобных приложений можно было заметно упростить, ускорить и улучшить работу с устройством. Впрочем, некоторые предприимчивые разработчики собирали и собирают подобные улучшения в своеобразные пакеты, объединяя их в рамках собственных проектов, распространяя их уже на коммерческой основе.

Большинство компаний-разработчиков устройств зачастую никак не реагируют на жалобы пользователей, прикрываясь требованиями по программированию и дизайну приложений для платформы. Остальные откликаются крайне неохотно, внося минимальные изменения в новые версии интерфейсов, устанавливаемых на новых устройствах, оставляя без внимания нужды пользователей уже выпущенных устройств. Отчасти это можно понять, ведь в проектирование и разработку производимых устройств вложены огромные средства и изменения в существующем коде потребуют заметных усилий, а перепроектирование и тестирование потребует немалых расходов. Для справки: один раунд юзабилити-тестирования интерфейса обходится в сумму от 10 до 30 тысяч долларов. К счастью не все компании обременены уже существующими проектами, которые необходимо окупить. Есть и такие, которые выходят на рынок мобильных устройств и сотовых средств

связи впервые. Речь идёт о двух гигантах компьютерного мира - Apple и Google.



Компания Apple сильно взбудоражила весь мобильный мир заявлением о выпуске революционного сотового телефона - iPhone, интерфейс которого предполагает управление исключительно касанием подушечками пальцев, не требуя использования стилуса и почти не используя аппаратные кнопки. Демонстрация прототипа iPhone удивила простотой, удобством и красотой интерфейса, моментально став своеобразным эталоном в области проектирования интерфейсов для мобильных устройств. После выпуска телефона на рынок, его продажи моментально взлетели от нуля до небывалых высот, обогнав, наверное, все остальные телефоны по популярности. На текущий момент в мобильном мире iPhone представляет одну из наиболее перспективных платформ для разработки ПО. Стоит заметить, что компания стояла у

истоков персональных цифровых устройств. Ещё в 1993-м году Apple выпустила один из первых персональных цифровых помощников под названием Newton Message Pad, который так же оказал заметное влияние на умы разработчиков портативной электроники, установив некоторый стандарт набора функций подобных устройств.

Компания HTC очень оперативно ответила на вызов компании Apple, и выпустила целый ряд устройств, оснащённых заметно переработанным рабочим экраном Windows Mobile 6. Так же как в случае iPhone, интерфейс HTC Touch FLO предполагает управление устройством посредством сенсорного экрана без использования стилуса. Компания активно развивает модельный ряд коммуникаторов снабженных этим интерфейсом.



Не осталась в стороне и Google, поначалу заявившая о подготовке в выпуску в свет сотового телефона собственного производства, оснащенного собственной же операционной системой, основанной на ядре линукса. Устройство тут же окрестили GugloPhone или GPhone :). Однако со временем компания отказалась от разработки и выпуска аппаратуры, сообщив, что оставляет разработку аппаратуры платформы третьим компаниям и концентрирует усилия

исключительно на разработке операционной системы под названием Android. По заявлению команды разработчиков, эта система спроектирована таким образом, чтобы сделать использование портативного устройства максимально простым и удобным. Кроме того, система в очень большой степени привязана к интернет-сервисам компании, используя их в качестве основных рабочих инструментов.

Кроме Google серьезную ставку на использование мобильного линукса делает компания Motorola, уже выпустившая более пяти моделей сотовых телефонов, использующих эту операционную систему и, судя по всему, планирует в ближайшем будущем заметно расширить модельный ряд. Разработчикам, в большинстве случаев, предоставляется возможность создавать свои проекты используя известную библиотеку Qt, специальные средства разработки и достаточно подробную документацию, которые можно найти на сайте компании.

С мобильным линуксом, ориентированным на применение в коммуникаторах и сотовых телефонах, связан ещё один проект, который стоит упомянуть. Он называется OpenMoko. Его идеология заключается в полной открытости и проведении совместной работы не только над улучшением исходного кода, но и в возможности вносить разумные изменения в аппаратную часть и внешний вид устройства. Проект существует и достаточно активно развивается уже в течение ряда лет. За это время было выпущено устройств под названием Neo и разработано несколько версий программного обеспечения для него.

Ссылки на использованные материалы:

<http://www.spbsoftwarehouse.com>

<http://www.zumobi.com>

<http://www.pointui.ru>

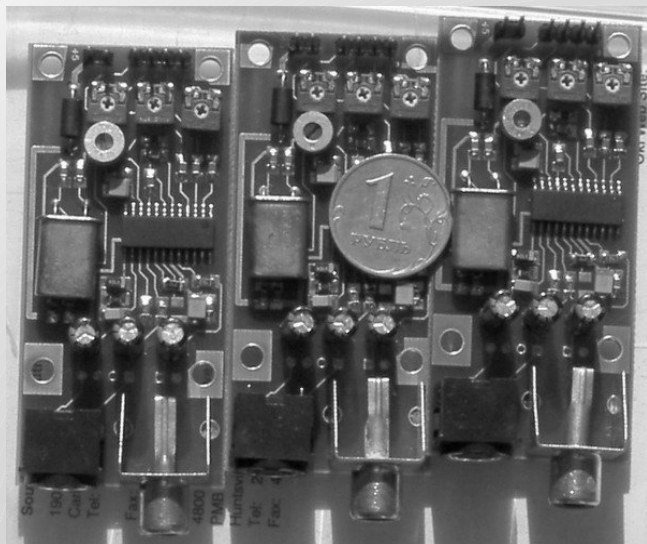
<http://developer.apple.com>

<http://code.google.com/android/>

<http://developer.motorola.com>

<http://www.openmoko.com>

Продукция группы NedoPC (www.nedopc.com)



PAL-кодер телевизионного видеосигнала.

PAL-кодер предназначен для конвертации телевизионного сигнала с разделенной цветностью (RGB) в композитный или S-Video сигнал с кодированием цвета в системе PAL. Исходный видеосигнал должен иметь смешанную синхронизацию.

Устройство обладает небольшими габаритами и потреблением. Требуется источник питания +5в.

Платы изготовлены с защитным (зеленым) покрытием.

Также возможно изготовление плат для кодирования в NTSC стандарте и для работы с другими номиналами питания (например +12в).

Стоимость изделия в сборе 350руб

Звуковая плата TurboSoundFM, для компьютеров ZX Spectrum.

TurboSoundFM позволяет использовать шесть звуковых каналов вместо стандартных трех каналов у AY8910/12 или YM2149. И, дополнительно, шесть программных каналов с FM генерацией звука. Устройство устанавливается через соответствующий переходник на посадочное место AY8910/12..

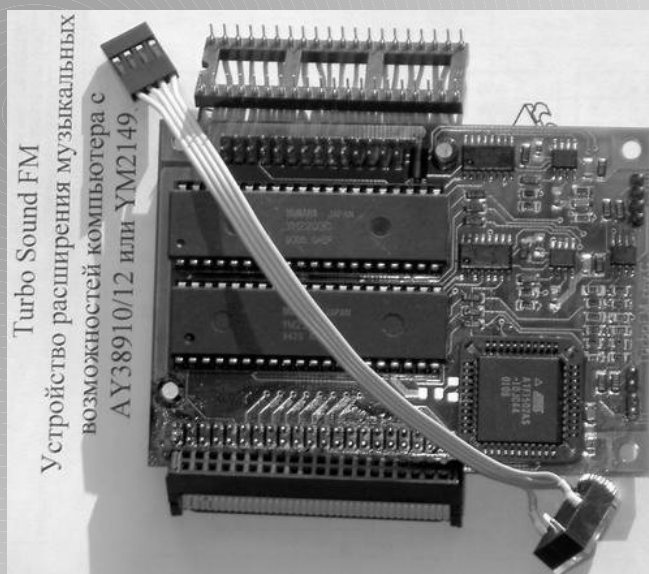
Платы изготовлены с защитным (зеленым) покрытием.

Стоимость изделия в сборе 650руб

Стоимость платы для сборки 150руб

Также предлагаются платы TurboSound для самостоятельной сборки.

Стоимость платы для сборки 150руб



Внимание! В стоимость плат не включена цена услуг почтовой связи.

Порядок обработки заказов:

Присылаем email chunin@mail.ru или звоним по телефону +7(910)4552728, чтобы выяснить - есть ли свободные платы и согласовать цену, комплектацию.

Осуществляем почтовый перевод на адрес 109451, г.Москва, а/я 12, Чунину Роману Валерьевичу. Пожалуйста указывайте свой обратный адрес и ФИО (если через email, то с указанием номера и даты почтового перевода).

После обработки и подготовки (примерно одна/две недели) заказ отсылается по указанному вами адресу.

История DiHalt demoparty

Автор *Vinnny*⁵



Отправной точкой можно считать 1997 год, когда молодая только что сформировавшаяся группа дзержинских

спектрумистов раздобыла диск с спрессованными с Enlight'97. Ранее особое внимание демосцене никто из них не уделял. В лучшем случае это был интерес к cracktro в играх. Увидев содержимое диска, они заразились демосценой на всю жизнь...

Проводя ночи за компами уже к следующему году первая делегация со своими первыми работами была отправлена на FunTop'98. Вернувшись с пати под огромным впечатлением было решено сделать своё собственное. И началось...

Первый DiHalt

Приглашения рассылали лично в руки по обычной почте. Сняли зал в местной школе, завезли несколько Спектрумов. Ни рекламы, ни финансирования, ни организации - ничего не было. Ими двигал только энтузиазм! Несмотря на малое количество посетителей (около 30 человек) DiHalt 1999 собрал практически самое большое количество работ (и самых качественных) из всех демопатей, проводившихся в том году для платформы ZX Spectrum. Как и следовало



5 Официальный сайт www.dihalt.org.ru

ожидать, минусы в подходе к проведению демопати дали о себе знать. Команда организаторов разочаровалась не увидев возлагаемых надежд на DiHalt. От себя скажу, что первый блин, как известно, всегда получается комом, и это надо было учитывать и не сдаваться.

Тем не менее, больше DiHalt было решено не проводить.

Тем временем я случайно из сети Fido узнаю о прошедшем демопати. Мой интерес к демосцене тоже только зарождался. Вместе с другом под ником ВІВ мы поехали на своё первое пати Chaos Constructions 2000 в Питер. Такую атмосферу я увидел первый раз, и мне это очень понравилось. На следующий год я уже ехал со своей работой (правда её отсеяли на преселекте, но первый блин всегда комом :-). Разочарования не было, оно наступило позже, когда на следующий 2002 год я узнал, что СС больше не проводится.

Выбор пал на ближайшее по расстоянию пати SAFe 2002, где я предложил свою помощь в организации по амижной части, и её приняли. На следующий год уехал опять на SAFe 2003. Пожалуй это были мои лучшие два года, где я и работал и отдыхал в своё удовольствие. К слову сказать, я ещё успел съездить на Kidsoft 2003 в Воронеж.

Новая жизнь DiHalt

Настал 2004 год. Ситуация кардинально поменялась. SAFe закрылось из-за недостатка финансирования, а СС открылось. Во мне что-то щёлкнуло, и после SAFe я стал относиться к демопати более критично. Особой альтернативы не было, и я поехал на СС. Увиденное меня поразило. С одной стороны это был ужас из-за количества проколов в организации пати, с другой унылая безнадежность из-за понимания что это единственное кроссплатформенное пати

на территории СНГ. Высказав в Fido всё организаторам и другим читателям, я стал успокаиваться, пока не осознал, что другого пути как проведение альтернативного пати, лично для меня нет.

Это был исторический момент. :-)

Времени ждать не было, на дворе уже стояло лето, а в конце августа планировалось ЦЦ. Дата проведения была назначена по принципу "только не осенью". Сошлись на середине августа 13-14 числа. Место проведения оставили историческое - город Дзержинск.

На тот момент я работал в компании "Атяшево", занимающейся производством колбасы, и уговорил своего шефа помочь с финансированием. Пожалуй это был самый большой парадокс, вызывающий улыбку - мясокомбинат спонсировал демопати. :-)



Сделали сайт, начали искать других спонсоров, рассылать информацию по информационным порталам, подготавливали оборудование, приглашали сценеров. Времени оставалось мало. Получился эдакий экспресс-марафон :) который показал, что интерес к нашей альтернативе есть. Именно этого мы сами ожидали и надеялись. Многим понравился конкурс по метанию жёстких дисков, и было забавно наблюдать как победитель вслед за винчестером метнул свою мобилу.

В целом на этом пати мы попытались кардинально изменить подход в организации

и проведении демопати по сравнению с прошлым DiHalt 1999. И можно сказать, что это получилось. Была кросс-платформенность, был график проведения, были призы и дипломы, были даже матрацы для тех кто остался ночевать на патиплейс. :-) Конечно наплыва работ и посетителей мы не ждали, оно и понятно было. Подведя итоги пати было решено сделать продолжение в следующем году.

DiHalt 2006



К Новому 2006 году мы собрались вновь - обсудить будущий DiHalt.

Дату назначили на 29-30 апреля с расчётом на последующие майские праздники. Главным изменением была смена места проведения. Теперь это Нижний Новгород. Для начала мы сделали английскую версию сайта, чтобы привлечь интерес зарубежных сценеров. В ходе подготовки было сделано аж целых 5 инвйтейшнов (приглашений), что пожалуй рекорд для демопатей. Свои маленькие победы мы продолжали воплощать в жизнь. Ввели интернет на пати, впервые для всех добавили конкурс для новой звуковой карты TurboSound для Спектрума.

Народа на пати заметно прибавилось, да и работ тоже. DiHalt потихоньку начинает набирать обороты. Чтобы сплотить сообщество на hiddenparty даже устроили небольшую раздачу колбасы всё того мясокомбината "Атяшево". Еда была очень даже кстати, и марка "Атяшево" прочно вошла в рацион питания сценеров и я не удивлюсь, если со временем её возведут в ранг сценерской классики. :-)

DiHalt 2007

Подведя итоги, победы и просчёты... назревал DiHalt 2007. Дата 2-3 июня. Было решено внести ещё несколько инноваций в фестиваль. Во-первых, это семинары. Один про историю демосцены, второй про стандарт

сотовой связи CDMA и его отличия от GSM. Во-вторых, это были реалтайм конкурсы. В частности для художников на Спектруме и ПЦ, а также для программистов на ZX. Для Spessу, кстати, мы опять не побоялись и ввели отдельный конкурс для новой звуковой карты TurboSoundFM. Были и недостатки организации, правды ради нужно сказать, что из графика мы тогда сильно выбились, два конкурса пришлось перенести на второй день.



Работ и посетителей ещё удвоилось. Люди приезжали из самых разных уголков России. Мы даже попытались дать людям WiFi-доступ в глобальную сеть, но времени не хватало, связь постоянно ломалась. Тем не менее, невзирая на все трудности, пати уложилось в отведённые два дня, включая процедуру награждения.

DiHalt



Проводя пати из года в год, это перестаёт быть обычным хобби, перерастая в более профессиональную сферу. Уже с другим

интересом происходит поиск и подбор спонсоров, принцип организации мероприятия. Стараешься сделать пати брендом, узнаваемым хотя бы в пока ещё узких кругах. Огромный маховик наматывает кучу нюансов, постепенно усложняя весь процесс в целом. Не за горами глобальный фестиваль DiHalt, и наша цель - уровень Breakpoint и Assembly.

DiHalt 2008 состоится уже через несколько дней, 28-29 июня, всё там же в Нижнем Новгороде. Мы увеличиваем количество конкурсов, количество семинаров, ищем спонсоров, и самое главное, хотим повысить качество проводимого демопати. В первую очередь это конечно же за счёт неукоснительного соблюдения графика мероприятия. Новинка DiHalt, да и вообще на всей отечественной демосцене - живое выступление SandS & со., которые сыграют несколько музыкальных тем из демок, musicdisk'ов и игр. Из семинаров ожидаются темы по программированию звука и разработке фильтров, серьёзный 3D-редактор Maya от Autodesk, в котором делаются сейчас самые кассовые художественные фильмы, а также улучшенный семинар по истории демосцены, и новые технологии в стандарте сотовой связи CDMA EVDO.

Будет много интересного, участвуйте, приезжайте. Мы стараемся для вас!



Увидимся на DiHalt demoparty!



Представляю Вам интервью с одним из основателей фестиваля, большим другом нашей команды и просто хорошим человеком и спектрумистом.

Коротко расскажи о себе?

Солодков Александр Вячеславович
TmK^[deMarche]^[Di:HALT:Team]

Родился и вырос в городе Дзержинске Нижегородская область.

Когда и как ты познакомился со Спектрумом?

Это было где-то в начале 90-х. У нас в городе в магазинах открывались точки, где можно было поиграть. Тогда я еще не знал что это Спектрум. В основном там крутили Barbarian, Exolon. Тогда сильно раздражало и было непонятно зачем сидеть и смотреть на непонятные полоски бегущие по краям картинки.

Спектрумы и игры для них (тогда еще на кассетах) появились в продаже в магазинах. Один мой знакомый купил себе Спектрум, а потом и мне захотелось. Первым у меня был Дельта-48 если не ошибаюсь. Но долго он не прожил. Все входы/выходы были выполнены пятиштырьковыми круглыми разъемами, и в один прекрасный день питание было подано не туда :). Потом другой мой знакомый, который якобы разбирался в радиотехнике, сказал что починит. Так мой первый спектрум погиб навсегда (я тогда еще не знал, что отслоившиеся дорожки это плохо, но догадывался что что-то не так по лицу чудо-мастера). Чуть позднее я увидел у родственников в Москве Пентагон-128 с дисководом, а еще чуть позднее в Нижнем Новгороде на радиорынке купил плату Пентагона-48 с контроллером дисковода.

Начал изучать цифровую технику, методы обращения с паяльником... Потом я познакомился с ребятами из своего города, которые увлекались демосценой на Спектруме и также были членами групп по

созданию программ/демок (deMarche, который ранее был известен на спектруме как Master Home Computers Group; FACE OFF, которые не раз меняли название и честно говоря не помню точного первоначального названия, ныне EyeQ во главе с Organism`ом который мало что делает, только языком трещит), приобрел себе Пентагон-128, и каждый день перед школой крутил демки с, которые сильно меня поражали своей красотой и динамичностью.

Ну а далее, думаю, стандартная схема: изучение ассемблера, написание простеньких эффектов, поездки на пати.

Какие первые впечатления были при знакомстве с этим компьютером?

Маленькая серенькая коробочка, а сколько всего интересного и красивого она может.

Расскажи как ты познакомился с фестивалем DiHalt?

Хороший вопрос.

Впервые мое знакомство с фестивалем я думаю произошло тогда, когда в 1999 году под впечатлением от FunTop`98 (тогда уже было известно что в 1999 году чисто спектрумовского пати не будет), после очередного выпитого стакана боярышника Organism сказал "А давайте сделаем свое чисто спектрумовское пати".

Ну и понеслось - сбор техники, рассылка приглашений (тогда еще по обычной почте), знакомство с Ковровскими спектрумистами (превед Vel`у). Первый DiHalt проводился в Дзержинске в 1999 году, без спонсорской помощи и нормальной подготовки. К сожалению, одного желания оказалось недостаточно - фестиваль с треском провалился. Хотя некоторые положительные моменты все же есть. За день до проведения к нам приехала команда Syndrome, с которой мы мило пообщались.

А когда и как ты стал одним из организаторов фестиваля?

Ну, собственно, я стоял у его истоков.

Тут лучше наверное поднять вопрос о том откуда взялся DiHalt 2005, 2006. Когда я вернулся из армии, у меня было мало желания заниматься спектрумом и демосценой. Но как-то я зашел к Organism`у и узнал что есть такой человек Vinnny, и у него есть желание сделать фестиваль в нашей области. Знакомство с Владом произвело приятное впечатление. Также повлиял боевой настрой людей готовых принять участие в организации. И вот я снова в команде и мы делаем все чтобы каждый год провести очередной фестиваль.

Какие впечатления о последнем фестивале DiHalt-2007? Что было хорошего? А что не понравилось?

Подобные мероприятия всегда мне интересны. Много творческих людей, каждый со своими идеями и видением мира, интересное и познавательное общение. Впечатление о последнем фестивале не лучшие конечно. Из за нехватки людей в организационном составе были постоянные заморочки со сдвигами в графике проведения. То одно не успеваем, то другое. Да и общий доступ в сеть убивал.

Зря так построили сеть что любой мог зайти на любой комп и делать что угодно. Рабочие машины были постоянно загружены работой с сеткой, все что-то качали в рабочих машинах, все убийственно тормозило и порой, для нормальной работы приходилось отключаться от сетки, а работы перетаскивать на флэшках.

Еще и бессонная ночь между первым и вторым днем сказывалась (т.к. у нас в России по законам нашего менталитета все делается в последнее мгновение, многие работы были присланы к концу первого дня и нам с Владом пришлось в ночь между первым и вторым днем выкачивать, проверять и подготавливать к показу). Впрочем, загружены мы были по самое «не хочу».

Думаю, в этом году мы исправим допущенные ошибки. Основной фишкой фестиваля в 2008 году должно стать четкое

соответствие заявленному графику, а также своевременное размещение новостей и работ в сети. Также, если ребята успеют подготовиться, на фестивале будет живое выступление Нижегородской группы по тематике демосцены.

Считаешь ли ты интересной Спектрумовскую тематику на фестивале? Стоит ли поддерживать платформу "на плаву"?

Данная платформа интересна мне как минимум по двум причинам:

- я на ней вырос и впечатление от работ уровня 87-93 годов глубоко проникло в мое сознание.
- достаточная простота, ограниченность и тем не менее не до конца реализованный потенциал платформы (всегда можно придумать что-то новое).

И потом спектрумисты это отдельный слой населения со своим интересным мышлением, нам есть что вспомнить, что рассказать, мы были на заре развития демосцены...

Следишь ли за новостями компьютерного мира? Какие направления компьютерного андеграунда и арта тебе интересны?

Честно говоря, за сценой я слежу сейчас мало (работа и свои проекты отнимают много времени).

А сам что нибудь делаешь на ниве компьютерного творчества?

Готовлю две работы на DiHalt: TFM музыка и GIGASCREEN картинка. Также сейчас пишу cms для управления кланами в небезизвестной игре LineAge2.

Демку я на фестиваль сделать явно не успеваю.... а жаль...

Ну и традиционные пожелания нашим читателям?

Аффтар пиши исчо!

Ну а если серьезно, желаю всем творческих успехов в реализации своих проектов.

О больших программах, проблемах их наворачивания, отделении мух от котлет и конвейерах для этой цели



Автор *Alone Coder*⁶

О больших программах

Большая программа, даже понятно составленная и прокомментированная, не помещается в голове целиком. Её всегда приходится рассматривать по частям: сначала одни аспекты её работы (например, вывод на экран), потом другие (например, опрос клавиатуры). Нужно быть уверенным, что меняя одно место, мы не нарушаем успешную работоспособность другого.

Кроме того, в развитии программы (или её фрагмента) не исключены многолетние перерывы, в течение которых даже автор может забыть все детали устройства программы (или фрагмента). Что уж говорить о программисте, решившем навернуть чужой проект. В исходниках должно быть как минимум написано, как из них собрать программу!

О проблемах наворачивания программ

Перед созданием новой метки в малознакомом или существенно подзабытом исходнике нужно проверить наличие такой же. Это можно сделать в ALASM с помощью `COUNT metka` после компиляции.

Иначе можно нарваться на такую ситуацию:

- имеется метка, объявленная как `metka=$+1;`
- мы объявляем нашу метку выше этого места;

результат - наша метка испорчена!

При смене типа обращения к процедуре

(например, если параметры переехали в другие регистры) желательно эту процедуру переименовать. Тогда компилятор сам скажет, где обращения к ней. Если не переделать хотя бы одно обращение, то будут глюки. Если обращение спрячено в неактивной ветке условной компиляции, это тоже чревато, тем более что компилятор о таком обращении не сообщит. Поэтому находить обращения надо всё-таки поиском, но компилятор после переименования процедуры даст запасную долю уверенности, что вы ничего не пропустили.

Следует старательно избегать конструкций `CALL metka+3` и т.п., поскольку при редактировании тела процедуры в такой программе вы НЕ ЗНАЕТЕ, какие обращения такого вида есть по всему исходнику и насколько можно изменять процедуру. Если нужен вызов адреса `metka+3`, то достаточно поставить там ещё одну метку, назвав её, например, `metka3`. После этого основную метку переименовываем (для вящей уверенности, см. предыдущий абзац), ищем по всему исходнику выражения "`metka`" и исправляем их.

Похожая проблема возникает, если первый байт какой-либо процедуры на лету заменяется на `RET` с целью какого-либо переключения режимов. Проблема состоит в том, что вы не можете менять начало процедуры, не зная, где фигурирует значение её первого байта. Есть решение в виде `NOP` (потеря одного байта может окупиться в `«XOR A:LD (metka),A»` вместо `«LD A,byte:LD (metka),A»`). В любом случае, если метка используется как переменная, это должно быть помечено в исходнике! Лучший способ сделать это - объявление метки как `metka=$`. Весьма желательно в комментарии написать, что за

⁶ Email автора dmitry.alonecoder@gmail.com

числа (или команды) там бывают.

Например, так:

```
NOP ;/RET
```

Или так:

```
CALL proc1 ;/proc2
```

Такие комментарии особенно способствуют пониманию исходника отладчика STS :).

Если один и тот же адрес используется и как переменная, и для переходов/вызовов - делайте ДВЕ метки, будет гораздо понятнее и появится простая возможность перенести переменную в другое место в будущем.

Очень запутывает такой код:

```
metka LD HL,0
```

когда где-то далеко (например, в другом файле исходника) стоит:

```
LD (metka+1),HL
```

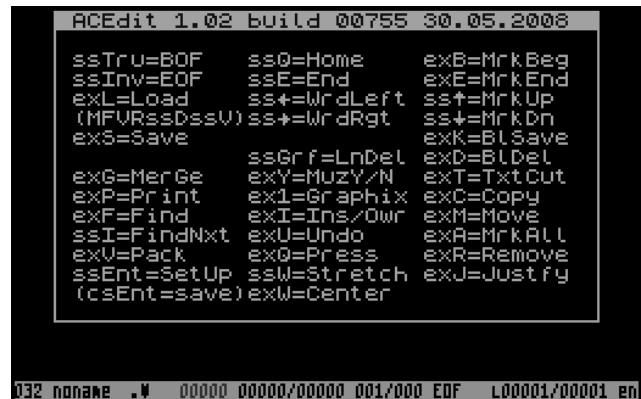
Когда читаешь первый участок, понятия не имеешь, что тут скрыта переменная! А это чревато глюконосными изменениями при доработках этого куска (особенно "оптимизациями").

Об отделении мух от котлет

Я не утверждаю, что каждая процедура должна уместиться в 2 экрана. Но всё же хотелось бы иметь достаточно понятные (с учётом того, что программист к ним вернётся, возможно, через годы, и не обязательно тот же самый программист) процедуры, делающие более-менее определённые действия.

Уже с первых версий AC Edit из процедур отдельных операций с текстом было выкинуто кое-что, не имеющее к ним отношения - перепечатка курсора. То же самое я сделал в Pro Tracker. А сделано это было не только и не столько для красоты исходника, а ещё и по той же вышеизложенной причине - чреватости глюками. За всеми вызовами перепечатывания не уследишь, а оставленные

на экране отпечатки курсора - пятна на репутации программиста. Поэтому, хотя в цикле опроса клавиатуры курсор виден, он стирается непосредственно перед вызовом обработчика клавиши (или клика в меню). После выхода оттуда в главный цикл работы программы курсор снова рисуется.



Пока что в AC Edit обработчики сами рисуют и стирают крест положения курсора, сами перепечатывают экран, сами смотрят буфер клавиатуры на случай повтора клавиши (Up, Down) или очищают этот буфер (PgUp, PgDn). Перепечатку экрана и крест можно перенести в главный цикл, при условии, что каждая клавиатурная функция, нуждающаяся в перепечатке экрана или рисовании креста, установит соответствующие флаги в определённой переменной. Это даст возможность автоматически выполнять длинные клавиатурные макросы (даже целые программы, если будет написан язык скрипта) очень быстро, не перерисовывая экран.

До версии 1.01 в отдельных обработчиках были даже куски опроса клавиатуры, отвечающие за заглавные русские буквы на цифрах и за ввод символа по коду. Это было убрано только тогда, когда был сделан конвейер клавиатурных событий.

О конвейерах для этой цели

Вообще говоря, опрос кнопочек (матрицы), автоповтор, русификация, переключение режимов и ввод символа по коду уже

изначально видятся как отдельные задачи. Проблема состоит в том, что между ними передаются не состояния, а события, и одно событие на уровне одной из этих задач не всегда вызывает ровно одно событие на уровне другой.

Проще всего сделать так, что в главном цикле будут опрашиваться кнопки, а по каждому изменению будут вызываться их обработчики. В этом случае нет никакой буферизации клавиатуры - нажатия во время долгих операций будут проглатываться. Кроме того, все обработчики сочетаний CS+цифра будут вынуждены вызывать процедуру ввода символа по коду, а та будет организована как автомат с кучей состояний. Раньше эта процедура обходилась без автомата с кучей состояний, а сама опрашивала порты клавиатуры, минуя общий опрос матрицы, но это не даёт возможности сменить тип клавиатуры или реализовать злосчастные макросы.

Рассмотрим же, какие данные откуда идут и куда поступают.

1. Процедура опроса матрицы опознаёт нажатия/отжатия и выдаёт эти события.
2. Процедура автоповтора повторяет нажатия, пока не пришло отжатие, отжатия она проглатывает. (Эта процедура не зависит от типа клавиатуры.)
3. Процедура ввода символа по коду глотает комбинации CS+цифра и выдаёт коды символов.
4. Процедура переключения режимов переключает режимы клавиатуры, проглатывая соответствующие события.
5. Процедура русификации (и прочей языкофикации) в зависимости от текущего языка и регистра превращает нажатия в коды символов.
6. Процедура буферизации кладёт события в буфер.

Всё это происходит по прерываниям, а основная программа берёт события из буфера (коды символов и управляющие клавиши) и вызывает соответствующие обработчики.

Чтобы, скажем, ввод символа по коду не подвесил всё, ожидая очередной цифры, все эти процедуры должны работать независимо, как отдельные потоки. Они могут, когда захотят, запрашивать данные (GET) и, когда захотят, выдавать результат (PUT). При обработке GET и PUT должен действовать планировщик, включающий тот или иной поток. Во всяких мудрёных операционных системах можно применять очереди сообщений или пайпы, но у нас память не резиновая, очереди на каждом уровне хранить негде. И что мы делаем?

Очевидно, что GET в случае отсутствия данных на входе должен замораживать текущий поток. Также очевидно, что PUT передаёт данные для GET следующего модуля в конвейере.

Поэтому вместо злой и страшной операционной системы достаточно написать:

```

PUT
    POP DE
    POP HL
    PUSH HL
    PUSH DE
    JP (HL)

GET
    POP DE
    POP HL
    DEC HL
    LD (HL), D
    DEC HL
    LD (HL), E
    RET

```

То есть PUT просто вызывает следующий модуль, а GET меняет свой адрес входа и возвращается в предыдущий модуль!!!

Список адресов модулей оформляется как «CALL mod1:CALL mod2:...:CALL modN». Передача идёт не побайтно - сразу передаётся весь набор регистров, кроме HL и DE.

Neo GS

Группа NedoPC решила сделать себе и Вам подарок в виде нового устройства. Это устройство планируется запустить в серийное производство в конце осени этого года.

Мы решили переработать и развить идею популярной звуковой карты General Sound для отечественных клонов ZX-Spectrum. Давайте посмотрим что у нас получилось.

Автор LVD⁷

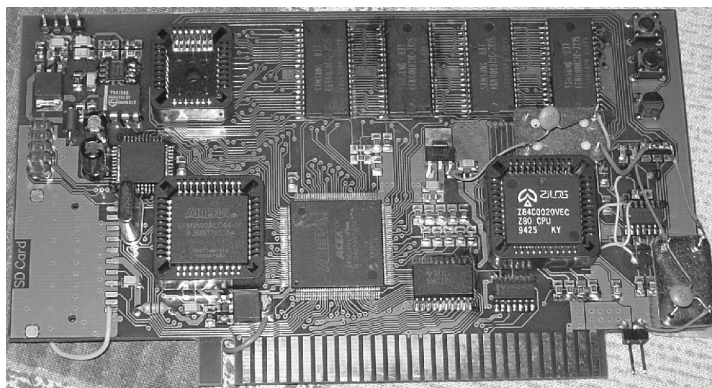
Краткие характеристики

- полная программная совместимость с General Sound (далее GS);
- гибкая архитектура, основанная на fpga
 - 2 мегабайта RAM
 - 512 килобайт flash-ROM
- расширенные способы адресации памяти: 2 окна проецирования, полное исключение ROM из адресного пространства;
- Z80 с переключаемой частотой (10, 12, 20 и 'турбо'-24 МГц);
- 8 аппаратных звуковых каналов с громкостями (в 2 раза больше, чем у GS);
- mp3-декодер (аналог vs1001) с аппаратным SPI для пересылки данных;
- интерфейс SD-card с аппаратным SPI для пересылки данных;
- DMA-доступ спектрума в память NeoGS (пересылка данных между спектрумом и NeoGS командами LDI, LDIR и подобными);
- возможность полностью автономной работы (проигрывание модулей и mp3-файлов с SD-карточки по плейлисту, при этом сброс спектрума не влияет на NeoGS).

Звуковые возможности по сравнению с GS значительно увеличены: теперь практически любой существующий 4-канальный модуль поместится в памяти NeoGS. Расширенное число аппаратных каналов и увеличенная частота Z80 позволяют проигрывать как 8-канальные модули, так и одновременно музыку и звуковые эффекты без отключения каналов музыки. mp3-декодер вместе с носителем SD-card позволят вывести озвучивание на качественно новый уровень. Наконец, NeoGS действительно (в отличие от GS) может работать как высокопроизводительный акселератор, имеющий широкий DMA-канал обмена данными со спектрумом. Кроме того, DMA-канал позволяет значительно сократить время загрузки звуковых данных в NeoGS.

Описание возможностей NeoGS с точки зрения программиста

Дешифрация портов со стороны спектрума, в отличие от GS, выполнена по полному младшему байту адреса, при этом сигнал IORQGE, опять же в отличие от GS,



выдаётся ТОЛЬКО на основе адреса и активен как при записи в порты, так и при чтении из них.

Порты **\$B3** и **\$BB** повторяют функциональность GS.

В карту портов добавлен порт **\$33** с функцией сброса NeoGS и выдачи процессору NeoGS немаскируемого прерывания: при выводе в него числа **\$80** происходит сброс NGS. При выводе **\$40** - процессору NeoGS выдаётся NMI.

Внутренние порты NeoGS (перечислены только отличающиеся от портов GS по функциональности, либо вновь добавленные):

\$40-\$FF - порты, использующиеся при начальной конфигурации FPGA, не должны использоваться при нормальной работе программ NeoGS.

порт **\$00**, запись - адресация расширена до 2 мегабайт (биты 5..0: 64 страницы по 32 килобайта)

порт **\$0F**, запись и чтение (того, что записано).

bit 0 - NO ROM bit:

- 1 - подключена только RAM,
- 0 - \$4000-\$7fff RAM, остальное FLASH со страницами такими же как RAM.

bit 1 - RAM RO bit:

- 1 - страница 0 RAM (абсолютные адреса 0..32767) защищена от записи (эмуляция ROM ogs)
- 0 - не защищена от записи.

bit 2 - 8 channels bit:

- 1 - 8 звуковых каналов
- 0 - 4 канала (как в GS).

bit 3 - extended paging mode bit:

- 1 - расширенный режим страниц (см. ниже)
- 0 - память переключается как в GS

bit 4,5 - z80 overclocking bits:

- 00 - Z80 работает на частоте 24МГц
- 01 - Z80 работает на частоте 12МГц
- 10 - Z80 работает на частоте 20МГц
- 11 - Z80 работает на частоте 10МГц

bits 6,7 - неиспользуемые: не определены при чтении, должны устанавливаться в ноль при записи

⁷ Email автора lvd.mhm@gmail.com

порт **\$10**, запись. Расширенный порт страниц, используется, когда extended paging mode bit = 1

Модель страниц в расширенном режиме такова:
порт **\$00** - страница (16кб), включенная по адресам \$8000-\$bfff

порт **\$10** - страница (16кб), включенная по адресам \$c000-\$ffff

Для 2 мегабайт памяти существует всего 128 страниц по 16 килобайт, пронумерованных числами от 0 до 127.

Имея номер такой страницы в А, делаем RRCA и выводим результат в порт \$00 или \$10, включая страницу с заданным номером в соответствующее окно проецирования.

порты **\$16-\$19**, запись. Установка громкости в каналах 5-8

Эти каналы существуют только в режиме, когда 8 channels bit = 1. В режиме 4 каналов при записи данных в каналы 1-4 эти данные автоматически копируются в каналы 5-8, т.о. громкость канала в 4-канальном режиме больше, чем в 8-канальном.

Как известно, в GS запись отсчёта в ЦАП канала происходит чтением этого отсчёта из памяти:

При чтении с адресов \$6000-\$60FF считанный байт идёт в ЦАП 1ого канала.

При чтении с адресов \$6100-\$61FF - в ЦАП 2ого канала

\$6200-\$62FF - 3его канала

\$6300-\$63FF - 4ого канала

\$6400-\$64FF - снова 1ого канала и так далее до \$7F00-\$7FFF - идёт в 4ый канал.

В 8-канальном режиме эта схема модифицируется очевидным образом:

\$60xx - 1ый

\$61xx - 2ой

\$62xx - 3

\$63xx - 4

\$64xx - 5

\$65xx - 6

\$66xx - 7

\$67xx - 8

\$68xx - 1

и так далее до \$7Fxx.

Распределение каналов по стереопанораме:

каналы 1,2,5,6 — левые каналы

3,4,7,8 — правые

MP3 u SD card

Для связи с MP3-декодером (ma8201, аналог vs1001) используется 2 канала spi. Первый работает только на передачу, передаёт непосредственно mp3-данные. Второй - управляющий, двунаправленный.

Для связи с SD-карточкой используется 1 spi-канал.

Передача байта осуществляется путём его записи в специальный порт передачи, ассоциированный с данным SPI-каналом, что инициирует цикл обмена. Принятый в этом обмене байт (для двунаправленного

канала SPI) доступен в порте чтения. Кроме того, возможно чтение с перезапуском обмена (приём сектора из SD-карточки будет выглядеть как INIR:INIR, передача - OTIR:OTIR).

DMA

Со стороны спектрума DMA-канал с NeoGS выглядит как кусок памяти, подставляемый NeoGS вместо ПЗУ спектрума при чтении или записи данных по адресам \$0000-\$3FFF при включённом ПЗУ (активен сигнал /CSROM на zx-bus).

Обмен происходит следующим образом:

На борту NeoGS есть адресный указатель, регистр чтения и регистр записи. Если при включенном режиме DMA спектрум читает из области \$0000-\$3FFF (конкретный адрес в этих пределах не имеет значения) при включенном ПЗУ, то происходит следующее:

1. ПЗУ блокируется сигналом RDRом zx-bus.
2. NeoGS выдаёт на ШД спектрума содержимое регистра чтения
3. NeoGS инициирует цикл DMA в собственную память выдачей /BUSRQ на процессор NeoGS; после получения /BUSAK на ША NeoGS выставляется содержимое адресного указателя и происходит чтение этого адреса, прочитанный байт сохраняется в регистре чтения, а адресный указатель инкрементируется.

Процессы 1-2 и 3 идут параллельно таким образом, что спектрум принимает байт, считанный NeoGS из своей памяти в предыдущий цикл. Однако если в момент чтения спектрумом байта NeoGS ещё не успел его прочитать из собственной памяти, то спектруму выставляется /WAIT.

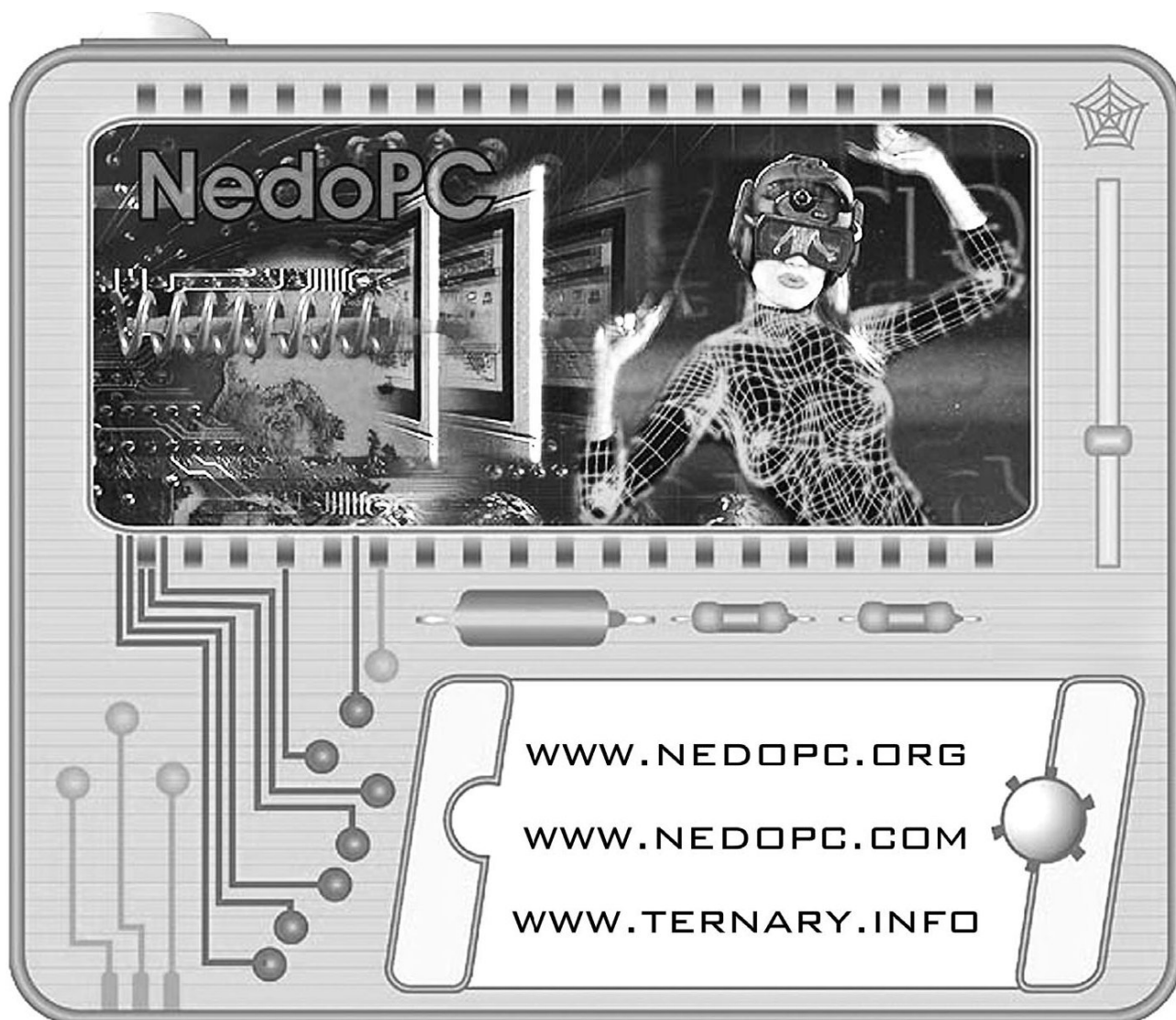
При записи спектрумом в адреса \$0000-\$3FFF происходит следующее:

1. Записываемый байт сохраняется в регистре записи.
2. Иницируется цикл DMA, в котором по адресу указателя записывается принятый со спектрума байт.

/WAIT аналогично выдаётся спектруму, если предыдущее значение регистра записи ещё не сохранено в памяти NeoGS.

При работе Z80 спектрума на частоте 3.5МГц, а NeoGS - на частоте 20 МГц /WAIT'ы выдаваться, как правило, не будут при любой последовательности чтения или записи (в том числе через стек).

Внимание: Вся приведённая информация является предварительной и может быть изменена без предупреждения. Конечные спецификации будут доступны в комплекте документации, поставляемой с готовыми устройствами NeoGS.



Copyright (c) 2002-2008 NedoPC team

ВСЕГДА В ON-LINE!

Сайты на домене NedoPC.COM :

- atmturbo.nedopc.com - сайт, посвященный компьютеру ATM Turbo2+ автор сайта Maksagor (max_timonin@mail.ru)
- pentagon.nedopc.com - сайт, посвященный компьютеру Pentagon1024 автор сайта КОЕ (KingOfEvil@yandex.ru)
- lvd.nedopc.com - сайт о современном «железе» ZX и Amiga автор сайта LVD (lvd.mhm@gmail.com)
- zxparty.nedopc.com - официальный сайт фестиваля ZX компьютерного творчества ArtField автор сайта КАСик (kidsoft-zx@yandex.ru)