



NedoPC group 2005

eXtra BIOS

Персональный компьютер TURBO 2+

V1.32
(LE/XT)

Описание прошивки
eXtra BIOS v1.32
для м/с ПЗУ
27512/271000
и выше

Москва 2005

1.ОГЛАВЛЕНИЕ

1. ОГЛАВЛЕНИЕ	стр. 2
2. ПРЕДИСЛОВИЕ	стр. 3
3. ОПИСАНИЕ ПРОШИВКИ eXtra BIOS v1.32LE/XT	стр. 4
Общие сведения	стр. 4
Значение страниц в стандартной прошивке	стр. 5
Значение страниц в xBIOS v1.32LE	стр. 5
Значение страниц в xBIOS v1.32XT	стр. 6
Описание верхней половины ПЗУ	стр. 6
Описание нижней половины ПЗУ	стр. 7
4.ОПИСАНИЕ СТАРТОВОГО МЕНЮ eXtra BIOS v1.32LE/XT	стр. 9
Стартовое меню eXtra BIOS v1.32LE	стр. 9
Стартовое меню eXtra BIOS v1.32XT	стр. 10
Описание низкоуровневых резидентных сервисов меню	стр. 12
5.ОПИСАНИЕ КОМАНД BIOS vTR-DOS 2.4 beta	стр. 13
Общая информация о функционировании vTR-DOS	стр. 13
Точки входа в керналь	стр. 13
0 - set_Hardware_init	стр. 14
1 - get_hardware_version	стр. 15
2 - set_vtrd_mount	стр. 16
3 - get_vtrd_mount	стр. 16
4 - set_vtrd_map	стр. 17
5 - get_vtrd_map	стр. 17
6 - set_page_hook	стр. 18
7 - get_page_hook	стр. 19
8 - set_nmi_hook	стр. 19
9 - get_nmi_hook	стр. 20
A - set_reset_hook	стр. 20
B - get_reset_hook	стр. 20
C - set_flag	стр. 21
D - get_flag	стр. 21
E - set_io_hook	стр. 21
F - get_io_hook	стр. 22
ПРИЛОЖЕНИЕ 1: изменения в процедуре проверки INTERFACE 1	стр. 23
ПРИЛОЖЕНИЕ 2: структура ОЗУ ATM-turbo 2+ по умолчанию	стр. 24
ПРИЛОЖЕНИЕ 3: краткая информация о структуре страницы #38	стр. 25
ПРИЛОЖЕНИЕ 4: работа xBIOS v1.32XT с ПЗУ объемом более 128Кб	стр. 25
ПРИЛОЖЕНИЕ 5: работа с резидентом в xBIOS v1.32LE/XT	стр. 28
ПРИЛОЖЕНИЕ 6: адаптация HONEY-Commander v4.00 к xBIOS	стр. 30
ИСТОРИЯ ВЕРСИЙ vTR-DOS	стр. 30
Наши адреса	стр. 32

2.ПРЕДИСЛОВИЕ

NedoPC group рада с гордостью представить вам плод многолетнего труда по совершенствованию ПЗУ ПК TURBO 2+. Работа над новой прошивкой была начата еще в далеком 2001 году, когда самой нашей группы еще не существовало, а ее нынешние члены даже не были друг с другом знакомы. Данная прошивка – плод совместных усилий множества активных пользователей АТМ и просто спектрумистов. Изначально совершенствование прошивки задумывалось с целью так изменить TR-DOS, чтобы заставить его «прозрачно» для основного множества программ работать с винчестером, чтобы сотни дискет можно было бы заменить на коллекцию образов на жестком диске.

Позднее эта идея была признана нерациональной и была выдвинута новая концепция эмуляции контроллера дисководов 1818ВГ93 при работе с верхней памятью, то есть с виртуальной дискетой, с RAM-диск. А образы в RAM-диск можно загружать и с винчестера, и вообще с любого другого внешнего устройства. Так что, по новой концепции, прошивка стала более универсальной.

Новая концепция, однако, потребовала и более рациональной организации имеющегося в наличии объема ОЗУ и ПЗУ, а также организации доступа к расширенным возможностям АТМ (диспетчер памяти, «скрытые» порты и т.д.). В результате работа затянулась, но все же, вопреки всему, была доведена до успешного завершения к концу июля 2005 года.

Однако, несмотря на выход прошивки, ее потенциал намного больше реализованных возможностей. Например, заложена возможность увеличения ПЗУ вплоть до 1024Кб. Это означает, что работа над прошивкой вполне вероятно будет продолжена. Но новые версии теперь будут опираться на созданную в данной прошивке базу, и будут обязательно совместимыми с версией v1.32 сверху вниз.

В завершение хотелось бы перечислить тех, кто все эти годы работал над прошивкой. В разное время в создании xBIOS так или иначе принимали участие следующие люди (перечисляются в хронологическом порядке подключения к разработке):

Maksagor/NedoPC group - Автор идеи о разработке нового ПЗУ. Организатор работы и ее информационного обеспечения. Соработчик первоначальной концепции. Автор ROM-trigger и адаптации BIOS CP/M для работы с ПЗУ, большим, чем 64Кб. Соавтор ядра vTR-DOS. Осуществил окончательную сборку xBIOS. Бетатестер.

TiM0N/AREAsoft - Активный соработчик первоначальной концепции. Информационное обеспечение. Автор процедур работы с винчестером. В дальнейшем отошел от активной работы.

ukms[z] - Автор новой концепции эмуляции 1818ВГ93 в верхней памяти ПК. Создатель на ее основе vTR-DOS – основного ядра xBIOS, без которого эта прошивка была бы невозможна. Взвалил на себя больше 90% всей «кодерской» работы. Соавтор ядра vTR-DOS.

Корсуни Юрий - Автор загрузчика iS-DOS с жесткого диска и адаптация BASIC-128 и TR-DOS для его вызова. Бетатестер.

SMT - Автор эмулятора Спектрума UNREAL SPECCY с поддержкой эмуляции ПК АТМ-turbo 2+. Благодаря данному эмулятору работа значительно ускорилась. Принимал участие в окончательной отладке и сборке xBIOS. Бетатестер.

3. Описание прошивки eXtra BIOS v1.32LE/XT

Общие сведения:

Новая прошивка eXtra BIOS v1.32 по сравнению со стандартной имеет целый ряд преимуществ и новшеств и бывает двух типов:

xBIOS v1.32 LE (Light Edition) – облегченный вариант xBIOS. Рассчитан на 64Кб ROM и сделан специально для тех, у кого нет возможности или средств раздобыть более объемную ПЗУ, зато стандартная ПЗУ так или иначе в наличии имеется, и ее можно перешить. В нем реализованы основные функции эмуляции ВГ93 и контроля за диспетчером памяти, однако из-за нехватки места упрощено стартовое меню, а также не реализована возможность подключения дополнительных внешних модулей (плагинов). Кроме того, развитие данной прошивки остановлено. Все дальнейшие версии будут выходить под маркой **XT(eXTended)** под ПЗУ объемом не менее 128Кб.

Системные требования: ROM 64Кб only, RAM – не меньше 1024Кб!!!

xBIOS v1.32 XT (eXTended) – рассчитана на (как минимум) вдвое больший объем ПЗУ против стандартного. Это позволило реализовать с ней намного больше разнообразных функций, которые будут описаны ниже, а также создать удобное интерактивное меню. Кроме того, еще осталось значительное количество свободного места, зарезервированного для дальнейшего развития прошивки, путем подключения дополнительных модулей.

Системные требования: ROM от 128Кб - до 1024Кб, RAM – не меньше 1024Кб!!!

В сравнении со стандартной прошивкой LE/XT-версии xBIOS выглядит так:

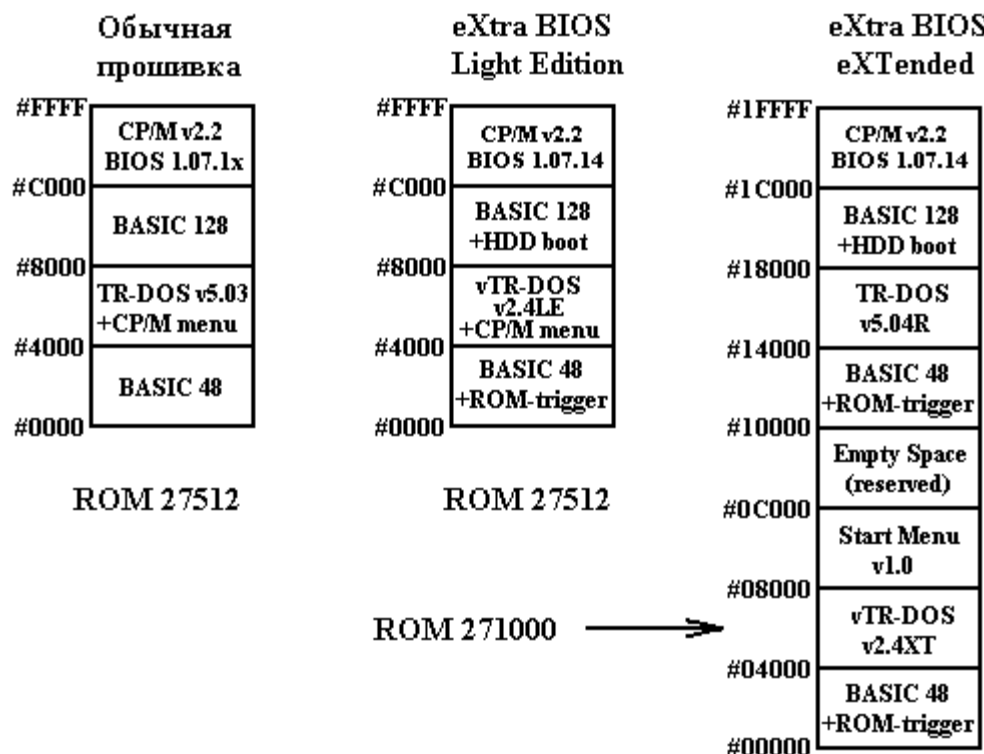


Рисунок 1

Значение страниц в стандартной прошивке –

BASIC 48 (#0000 - #3FFF) – стандартный бейсик для 48Кб с исправленной ошибкой обработки NMI.

TR-DOS v5.03 (#4000 - #7FFF) – ОС TR-DOS v5.03 со вставленными в свободное место **стартовым меню CP/M** и обработчиком резидента.

BASIC 128 (#8000 - #BFFF) – расширение стандартного бейсика для 128Кб

BIOS 1.07.1x для CP/M v2.2 (#C000 - #FFFF) – стандартная для ATM-turbo 2+ ОС CP/M v2.2. Именно с этой страницы стартует компьютер после RESET.

Прошивка бывает нескольких версий:

Версия **BIOS 1.07.12**, (вообще-то, изредка может попасться и старее, но эта уже давно является основной и базовой для всех последующих версий): базовая версия ПЗУ еще от МикроАРТ.

Версия **BIOS 1.07.13**, с исправленными ошибками в адресации RAM-диска.

Версия **BIOS 1.07.13a**, с подправленным Юрием Корсуниным драйвером винчестера, благодаря чему возможна работа с более широким кругом винчестеров.

Значение страниц в xBIOS v1.32LE –

BASIC 48 + ROM-trigger (#0000 - #3FFF) – тот же стандартный бейсик, что и в обычной прошивке. С одной лишь разницей: в нем присутствует т.н. ROM-trigger, переключатель между двумя половинками ПЗУ 271000. Но, так как в версии LE используется всего 64К-ПЗУ, то реального переключения половинок не происходит. Данная триггер реализован здесь лишь для совместимости софта, использующего с версией XT, где использование триггера имеет смысл. Подробную информацию о триггере смотрите в описании версии XT прошивки.

vTR-DOS v2.4LE (#4000 - #7FFF) – новейшая разработка ОС TR-DOS, в которой и заключается весь смысл eXtra BIOS. Практически ничем не отличается от такой же vTR-DOS версии XT кроме того, что в той версии стартовое меню вынесено в отдельную страницу, освободив дополнительное место. Поэтому более подробную информацию данной страницы смотрите в описании xBIOS XT.

BASIC 128 + HDD-boot (#8000 - #BFFF) – точно такое же расширение бейсика для 128Кб, что и в обычной прошивке, за исключением того, что в стартовом окошке доступна опция «HDD boot» которая осуществляет вызов загрузчика ОС iS-DOS, расположенного в TR-DOS v5.04R.

BIOS 1.07.14 для CP/M v2.2 (#C000 - #FFFF) - стандартная для ATM-turbo 2+ ОС CP/M v2.2; Версия BIOS 1.07.14. Именно с этой страницы стартует компьютер после RESET. Отличие от версии 1.07.13a заключается в подправленных процедурах вызова стартового меню и переключения различных страниц ПЗУ, что теперь позволило использовать ПЗУ любого объема вплоть до 1024Кб путем приращения дополнительного пространства снизу с автоматическим конфигурированием системы под новый объем ПЗУ.

Значение страниц в xBIOS v1.32XT –

Прошивка **Extra BIOS v1.32XT** – состоит как бы из двух половинок по 64Кб каждая. Из них верхняя (#10000 - #1FFFF) используется по умолчанию, а нижняя (#00000 - #0FFFF) включается программно по мере необходимости.

Описание основной (верхней) половины, используемой по умолчанию:

BASIC 48 + ROM-trigger (#10000 - #13FFF) – тот же стандартный бейсик, что и в обычной прошивке. С одной лишь разницей: в нем присутствует т.н. ROM-trigger, переключатель между двумя половинками ПЗУ 271000.

ROM-trigger имеет два входа:

#3C98 (15512) – на входе автоматически запрещаются прерывания, а на выходе разрешаются.

#3C9E (15518) – прерывания не отслеживаются. Их установка или запрещение – забота пользователя.

В обоих случаях портятся регистры А и ВС.

Суть триггера в том, что при его вызове происходит подмена страниц ПЗУ со страницы **#10000 - #13FFF**, содержащей BASIC 48 на страницу **#00000 - #03FFF**, где расположен точно такой же BASIC 48 с триггером. Но где триггер при вызове производит переключение обратно на страницу **#10000 - #13FFF**.

Таким образом, первый вызов триггера включает BASIC 48 из нижней половины ПЗУ, а вторичный вызов восстанавливает верхнюю половину. Смысл использования триггера заключается в возможности выбора таким способом одного из двух вариантов ОС TR-DOS –

TR-DOS v5.04R (далее - TRD) – обычная ОС (верхняя половина (стоит по умолчанию)

vTR-DOS v2.4XT (далее - vTRD) – новая разработка (нижняя половина, становится доступна при использовании ROM-trigger'а).

Вне зависимости от того, верхняя или нижняя половина ПЗУ выбрана, TR-DOS выбирается посредством перехода по адресам **#3Dxx**. Только в одном случае выберется TRD, а в другом – vTRD.

Определить, какая из половинок ПЗУ, верхняя или нижняя, сейчас активна, можно, считав байт по адресу **#3CBC (15548)** из BASIC 48.

Если он равен **#87(135)**, то активна верхняя половина.

Если он равен **#83(131)**, то активна нижняя половина.

Если он равен **#00(0)** или любому другому числу, то у вас прошита не eXtra BIOS и, следовательно, ROM-trigger отсутствует.

TR-DOS v5.04R (#14000 - #17FFF) – самая обычная ОС TR-DOS, с небольшими доработками, базирующаяся на версии 5.04T. суть доработок заключается во вставленных обработчике резидента, а также загрузчике ОС iS-DOS с HDD. Зато из этой страницы было убрано стартовое меню CP/M, которое было перемещено в другую страницу. Это высвободило много свободного места в TR-DOS, и вообще теперь можно свободно менять версии TR-DOS, **не забывая вставлять туда резидент и загрузчик iS-DOS**.

BASIC 128 + HDD-boot (#18000 - #1BFFF) – точно такое же расширение бейсика для 128Кб, что и в обычной прошивке, за исключением того, что в стартовом окошке доступна опция «HDD boot» которая осуществляет вызов загрузчика ОС iS-DOS, расположенного в TR-DOS v5.04R.

BIOS 1.07.14 для CP/M v2.2 (#1C000 - #1FFFF) - стандартная для ATM-turbo 2+ ОС CP/M v2.2; Версия BIOS 1.07.14. Именно с этой страницы стартует компьютер после RESET. Отличие от версии 1.07.13а заключается в подправленных процедурах вызова стартового меню и переключения различных страниц ПЗУ, что теперь позволило использовать ПЗУ любого объема вплоть до 1024Кб путем приращения дополнительного пространства снизу с автоматическим конфигурированием системы под новый объем ПЗУ.

**Описание дополнительной (нижней) половины,
вызываемой через ROM-trigger или через стартовое меню.**

BASIC 48 + ROM-trigger (#00000 - #03FFF) – тот самый обычный бейсик для 48Кб, с триггером восстанавливающим верхнюю половину ПЗУ, о которых было сказано выше.

vTR-DOS v2.4XT (#04000 - #07FFF) – новейшая разработка ОС TR-DOS, в которой и заключается весь смысл eXtra BIOS. Также как и в TR-DOS v5.04R, здесь вставлены резидент и загрузчик ОС iS-DOS с винчестера.

Список новшеств:

- 1) Программная эмуляция контроллера дисководов 1818ВГ93 в верхней памяти ATM-turbo 2+. При этом любое из устройств (А,В,С,Д) может быть или виртуальным (то есть работать через эмулятор ВГ93 на виртуальной дискете в верхней памяти, причем совсем не обязательно через точку входа #3D13!), либо реальным (работать с настоящим дисководом через реальную ВГ93).
- 2) Гибкая подстройка расположения виртуального диска в верхней памяти: изначально она, имея длину в **40(#28)** страниц, располагается в соответствии с RAM-диском оболочки HONEY-COMANDER 4.00(4.01), занимая первые его 640Кб. Однако система позволяет поменять расположение любой из 40 страниц виртуальной дискеты на другое место, при условии, что не будет использована страница ОЗУ **#38**, где располагаются дополнительные системные переменные новой TR-DOS.
- 3) Возможность программного перехвата кнопки MAGIC, RESET, а также системного порта **#xx77** и порта страниц **#xFF7**, с возможностью организации системы управляющих плагинов в верхней памяти.
- 4) Гибкая система опрвления всеми новыми функциями через единую керналь с тремя точками входа – **#3D3E(15678)**, **#3D42(15682)** и **#3D46(15686)**.

Полное описание функций vTR-DOS 2.4LE/XT смотрите в соответствующем разделе.

Для установки в тело TR-DOS в промежуток #3Dxx точек входа в керналь (сама керналь, как и эмулятор ВГ93, расположены в свободном пространстве прошивки TR-DOS), были сделаны небольшие изменения в самой прошивке, не отразившиеся на

работоспособности системы. Описание изменений в этом промежутке прошивки смотрите в **Приложении 1**.

Изначально планировалось выпустить прошивку с vTR-DOS только в рамках стандартной 64Кб-ПЗУ, заменив ею обычную TR-DOS v5.03. Однако внедрение в систему эмулятора контроллера 1818ВГ93 к большому сожалению, привела к, пусть и не большому, но, тем не менее, весьма досадному и неизбежному снижению совместимости с обычной TR-DOS до уровня 90-95% как при работе с виртуальной дискетой через эмулятор ВГ93, так и с реальными дисковыми устройствами, преодолеть которое программными средствами оказалось невозможно. Особенно это касается различных хитрых ксорок, обращающихся для получения ключа расксорки к содержимому ПЗУ TR-DOS, а также особо хитрых процедур опроса диска на предмет его наличия в дисковом.

Поэтому было принято решение выпустить 128Кб ПЗУ, где будут находиться две TR-DOS: обычная, полностью совместимая с другими TR-DOS, и новая с эмулятором ВГ93. Что и было сделано в данной прошивке.

Переключение между этими прошивками происходит либо путем выбора соответствующей опции в стартовом меню, либо посредством ROM-trigger'a, расположенного в ПЗУ BASIC 48. Включать vTRD наиболее удобно из среды других ОС, например, CP/M или iS-DOS посредством соответствующих утилит, на время использования виртуальной дискеты, за затем возвращать стандартное ПЗУ на место. Хотя ROM-trigger легко доступен даже из бейсика.

Стартовое меню CP/M (#08000 - #0BFFF) – Стартовое меню ATM-turbo 2+, ранее располагавшееся в свободном месте TR-DOS v5.03. Но на его месте теперь расположена TR-DOS 5.04R, а меню было перемещено сюда. Собственно данная страница в первых бета-версиях являлась копией TR-DOS v5.03 вместе с меню (где использовалось, естественно, только само меню), но в дальнейшем было оставлено исключительно стартовое меню, существенно переработанное, а точнее, переписанное с нуля. А освободившееся место использовано для дальнейшего расширения возможностей eXtra BIOS. Причем этого места еще осталось много, так что простор для дальнейшего развития xBIOS огромен. Особенность нового меню, использующегося, начиная с данной версии xBIOS v1.32XT, заключается в интерактивном интерфейсе, предварительной настройке системных переменных vTRD при входе с ZX-режимы, в переходе к загрузчику iS-DOS с винчестера. А также реализовано управление диспетчером памяти, возможность работы с дополнительными модулями, подключаемыми в соответствии со специальным форматом в свободные страницы ОЗУ или ПЗУ. Подробное описание функций стартового меню смотрите в соответствующем разделе.

Empty space (#0C000 - #0FFFF) – пустая страница, вся заполненная байтом #FF. Зарезервирована для дальнейших расширений в новых версиях eXtra BIOS. Пригодна также для размещения внешних модулей, распознаваемых стартовым меню по специальному заголовку (см. описание стартового меню). **Зарезервирована разработчиками! Для подключения внешних модулей используйте более объемные ПЗУ!**

Несмотря на то, что данная версия xBIOS рассчитана на ПЗУ 128Кб, она будет работать и с большим размером ПЗУ вплоть до 1024Кб. Для нормальной работы надо всего лишь, чтобы страницы xBIOS занимали бы самое верхнее положение в ПЗУ. То есть, новые страницы должны добавляться снизу, наподобие того, как на **рисунке 1** к стандартной 64Кб-прошивке ATM были добавлены страницы xBIOS. Подробнее об увеличении объема ПЗУ читайте в **приложении 4**.

4.Описание стартового меню eXtra BIOS v1.32LE/XT

Стартовое меню eXtra BIOS 1.32LE

Данное меню представляет собой список опций, пронумерованных в порядке возрастания. Опции выбираются путем нажатия на клавиатуре на цифру, соответствующую порядковому номеру опции в меню. Перед выводом данного меню хBIOS проверяет наличие в памяти резидента (если есть, запускает, минуя меню), а также наличие автозагрузчика (запускающего в автоматическом режиме, если не нажата клавиша SPACE, любую из опций меню). Существуют следующие опции:

- 1.**CP/M** – обычный запуск ОС CP/M.
- 2.**vTRD A** – включает защиту памяти в режим установленный опцией **7.STOP**, профиль памяти, монтирование рам диска на A и загрузка с A
- 3.**vTRD C** – включает защиту памяти в режим установленный опцией **7.STOP**, профиль памяти, монтирование рам диска на C, удаление всех пользовательских перехватчиков и карты памяти, и производит загрузку с A
- 4.**ZX 1mb** – отключает систему защиты/профилирования памяти и разворачивает доступную память до 1 Мб, производит горячий монтаж рам диска на букву C и выход в меню BASIC 128
- 5.**HDD** предназначен для загрузки с жесткого диска. включает защиту памяти в режим установленный опцией **7.STOP** и профиль памяти 1мб. при отсутствии загрузочного сектора на HDD включает профиль памяти указанный в опции меню, монтирует RAM-диск на C и производит загрузку с A.
- 6.**New RAM** – форматирует виртуальную TR-DOS дискету. А конкретно – очищает область каталога и устанавливает нужную служебную информацию в системном секторе. Перед форматирование требует подтверждения действия.
7. **FAST** – зелёный цвет – при входе в любой ZX-режим **turbo** включено, красный - выключено. Для CP/M всегда включено.
8. **AUTO** – предназначена для автоматического старта компьютера в соответствии выбранного основного пункта (зелёный цвет). не работает для CP/M в силу того что данные для авто старта сохраняются в странице **#38** которая используется для RAM-диска этой ОС. Для отмены автостарта при перезапуске компьютера необходимо нажать и удерживать пробел.
- 9.**STOP** – регулирует реакцию защиты памяти/системного порта. Аналог **.vendor_protect_hide**(функция **vTRD**) - красный цвет, **.vendor_protect_break**(функция **vTRD**) - зелёный цвет.

0.128/512/1024 используется для профилирования памяти в режимах vtrd по-умолчанию и ZX 1024 (профиль выбран, но отключен на момент входа в этот режим, для включения следует подать команду **set_hardware_init.vendor_protect_init**(функция vTRD))

Стартовое меню eXtra BIOS 1.32XT (Setup Utility v1.0)

Сделана возможность использовать новое загрузочное меню для атАТМ. Для этого была разработана программа **setup utility**. Она предназначена для управления некоторыми важными опциями системы dual tr-dos 128к.

После нажатия клавиши **RESET** производится поиск резидента в памяти и в случае его отсутствия - вывод информации о версии программы меню и проверка установки автозапуска.

В случае его отсутствия выводится сообщение и ожидается нажатие клавиши spase. После её нажатия выводится основной экран меню.

В случае обнаружения информации об авто-запуске, он выполняется и для его отмены необходимо нажать **SPACE**.

Навигация в меню осуществляется курсорными клавишами. изменение/выбор параметров производится с помощью клавиш '**ENTER**', '**PAGE UP**', и '**PAGE DOWN**'.

Функционально меню разделено на три группы - **boot**, **options** и **service**.

Первая группа (Boot) включает в себя варианты загрузки компьютера.

CP/M 2.2	- запуск CP/M
TR-dos boot	- автозапуск TR-DOS v5.04R с диска а с открытыми 128-ми страницами (т.е. эмулятор ВГ93 отключен)
vTR-dos hot	- автозапуск vtr-dos с диска а с настройками options. сохраняются пользовательские перехватчики.
vtr-dos cold	- автозапуск vTR-dos с диска а с настройками из Options и полной инициализацией.
ZX-spectrum menu	- обычное меню бейсика ZX-128 + vtTR-dos(т.е. эмулятор ВГ93 включен) с настройками из Options .
IDE hard drive master	- для запуска процедуры IDE-loader (загрузка iS-DOS)

Вторая группа (Options) включает в себя различные опции vTR-dos.

Auto Boot	- автозапуск после RESET выбранного ранее пункта меню из опции Boot.
Turbo mode	- включение/отключение режима TURBO.
Stop screen	- включение/отключение отображения специального стоп-сообщения при попытках доступа к защищённым портам и страницам памяти.
Swap drives	- меняет местами только реальные дисководы – A<>B и C<>D (только в vTR-DOS, естественно).
Protected mode	- включение/отключение защиты системных портов и страниц от несанкционированных посягательств для контроля целостности содержимого виртуальной дискеты. Таким образом, впервые в истории Спекти на нем реализована защита памяти!
Read ONLY mode	- включение/отключение режима запрета записи для виртуальной дискеты.
Virtual drive mode	- выбор расположения виртуальной дискеты в RAM или ROM (только для ПЗУ 1024Кб. В других случаях выбор ROM будет бессмысленным).
RAM drive mount to	- монтирование виртуальной дискеты соответственно к устройству A, B, C или D.
Extended memory profile	- объём доступной памяти в ZX режиме в защищенном режиме (выбор между 128Кб , 512Кб и 1024Кб . В первых двух случаях доступ к памяти выше указанной цифры для обычного ZX-софта запрещен).

Третья группа (Service) предлагает небольшой сервис.

Fast TR-dos FORMAT RAM disk	- очищает каталог виртуальной дискеты.
Extended ROM service	- вывод списка и запуск найденных в ПЗУ утилит (см. ниже).
Extended RAM service	- вывод списка и запуск найденных в ОЗУ утилит (см. ниже).
Save changes to RAM #38	- сохранение установок options в системной странице.

Чтобы перемещаться курсором по правому меню следует нажать клавишу **'right'**, а чтобы перемещаться курсором по левому меню необходимо нажать **'left'**.

Изменение параметров группы **Options** производится клавишами **'enter'**. Также можно использовать клавиши **'page up'** - уменьшение параметра и **'page down'** - увеличение параметра.

Описание низкоуровневых резидентных сервисов

В xBIOS 1.32XT сделана возможность вызова специальных программ прямо из стартового меню (опции **Extended ROM/RAM service** в группе **Service**). Эти программы именуются сервисными утилитами и их предназначение прежде всего в модульном расширении функций стартового меню. Таким образом, становится возможным наращивать функциональность ПЗУ АТМ без переделок самого меню.

При автостарте, программа проверяет первый мегабайт ОЗУ и ПЗУ на наличие в страницах специальным образом оформленных "корешков" сервисных утилит. Результат поиска запоминается и впоследствии предлагается пользователю в меню **Service - Extended ROM/RAM service**. Пользователь, выбирая нужный пункт меню, автоматически запускает соответствующую сервисную утилиту. На данный момент существует ограничение в количестве утилит, предлагаемых пользователю на выбор. Только первые 16 утилит, найденных в памяти будет предложено на выбор. Всего их 32 - 16 из RAM + 16 из ROM. Пока это связано с ограничениями вывода на экран.

Формат "корешка" по которому распознаются сервисные утилиты:

```
#0000          JP START
#0003          DB "utility name " ; всегда 25 байт
#001C  START  ...
```

Утилита запускается в расширенном режиме, запрещёнными прерываниями, с текстовым экраном со стандартной палитрой. Клавиатура находится в режиме ZX. Запуск производится в окне **#C000**. стек находится в окне **#8000** (в его конце – в последнем килобайте), 0-й экран (страницы 5 и 1) в **#0000** и **#4000** (символы и атрибуты соответственно). Возврат в меню - обычный RET. также следует иметь ввиду, что в 5-й странице находятся различные системные переменные ОС CP/M.

также **Setup Utility** предоставляет маленький сервис :

#8006 - print ; процедура печатающая строку из IX по координатам HL атрибутом A
#8009 - cls ; очистка экрана и закрашивание атрибутом из A

print выводит на экран цветную строку из IX по координатам в HL, атрибут строки в A. при этом в тексте допускаются следующие управляющие коды:

- 0** - конец строки
- 1** - перевод каретки на следующую строку и на исходную позицию по X
- 2** - отступ на 30 символов

"-", "0".."F" - управление цветом следующих символов и установка черного фона. например : **DB "-6hello -1world"** - будет напечатано на черном фоне сначала 6-м цветом, а потом - 1-м.

5. Описание команд BIOS vTR-DOS 2.4LE/XT

Общая информация о функционировании vTR-DOS

Первым делом необходимо заметить что в текущей версии vTR-DOS не поддерживается использование диспетчера памяти и верхней памяти в частности. Все попытки переключения страницы с адреса **#4000** немедленно пресекаются, а с адреса **#8000** всегда устанавливается 2-я страница при обращениях к виртуальному диску. Иными словами - если не использовать загрузку с виртуального диска (или использовать её осмотрительно) можно перепрограммировать две страницы - **#8000** и **#C000**.

Таким образом эмулятор ВГ может производить ввод/вывод только в следующую карту памяти :

#0000 - 48к бейсик
#4000 - 5-я страница
#8000 - 2-я страница
#C000 - любая из первых восьми страниц.*

Ввод/вывод при использовании реального диска возможен только в следующую карту памяти :

#0000 - 48к бейсик
#4000 - 5-я страница
#8000 - любая из страниц
#C000 - любая из первых восьми страниц.*

Страничное окно **#C000** предлагается использовать только через порт **#7FFD** для совместимости с будущими версиями vTR-DOS. Использование расширенной памяти через это окно не рекомендуется. Если использование необходимо, то следует учитывать что в будущих версиях vTR-DOS при обращении к её сервисам в этом окне будет устанавливаться нулевой восьмистраничный банк памяти (первые восемь страниц с выборкой через **#7FFD**).

* - на самом деле не важно какая там страница, т.к. в текущей версии это окно не используется.

Точки входа в керналь:

Числа в формате **#XX:#YYYY+#ZZ** используются для абсолютного указания адреса переменных в памяти 1024к здесь **#XX** - номер страницы, **#YYYY** - смещение в странице, **#ZZ** - смещение в блоке переменных

Существуют три входа в BIOS.

bios1 #3D3E (15678)
bios2 #3D42 (15682)
verify #3D46 (15686)

Первая точка принимает команду и параметры из трёх ячеек памяти А В и С адреса которых **23735(#5CB7)**, **23737(#5CB9)** и **23736(#5CB8)** соответственно.

Вторая точка принимает команды и параметры из одноимённых регистров А, В и С.

Третья точка позволяет определить, какое ПЗУ TR-DOS установлено. Если после вызова этой точки в регистре А и в ячейке памяти #5CEF(23791) находится число **#01**, то установлена одна из версий обычной TR-DOS, а кernalь отсутствует. Если там находится число **#00**, то установлена vTR-DOS со стандартной кernalью.

При каждом вхождении в BIOS происходит проверка на наличие системных переменных и их создание путём вызова **#3D21 (15649)**.

Команда передаётся в В
подкоманда или **параметр** передаются в С
дополнительный параметр передаётся в А

Для команд начинающихся с **get_**
на выходе **результат** выдаётся в ВС или С
в зависимости от длины

_not_implemented - код означающий что данная функция не реализована в этой версии и зарезервирована на будущее, числовое значение на выходе равно **#AA55**

set_Hardware_init Инициализация оборудования

*** код 0**

- подкоманда 0 .rst 0
= рестарт резидента, иначе рестарт TR-DOS

- подкоманда 1 .hot
= инициализация переменных необходимых для работы эмулятора и монтирование set_vtrd_mount.drive[a]

- подкоманда 02..7F
= **_not_implemented**

машиннозависимые подкоманды :

- подкоманда 80 .vendor protect hide

= тихий режим защиты. подключение запрещённых страниц игнорируется. попытки использования системного порта игнорируются. программы использующие верхнюю память могут работать неправильно. не рекомендуется.

- подкоманда 81 **.vendor protect break**

= обычный режим защиты. при обращении в запрещённую страницу памяти или к системному порту выполнение программы прекращается и демонстрируется сообщение.

- подкоманда 82 **.vendor protect page**

- дополнительный параметр = номер страницы.

= блокирование доступа к странице для всех программ.

- подкоманда 83 **.vendor unprotect page**

- дополнительный параметр = номер страницы

= разблокировать доступ к странице для всех программ. не работает для #38.

- подкоманда 84 **.vendor memory profile**

- дополнительный параметр = номер профиля 0..2

= установка профиля памяти **0 = 1024кб, 1 = 512кб, 2 = 128кб**. Обращение к памяти вне профиля игнорируется, системный порт тоже игнорируется. исключение - профиль 0 который отменяет профилирование памяти. включение профиля отменяет изменения сделанные командами **.vendor_protect_page** и **.vendor_unprotect_page**

- подкоманда 85 **.vendor protect init**

= инициализация подсистемы защиты памяти с установками по-умолчанию :

set_hardware_init.vendor_protect_(hide/break) значение указанное в меню при старте

set_hardware_init.vendor_memory_profile [значение указанное в меню при старте]

Выполняется при рестарте vTR-DOS.

- подкоманда 86 **.vendor protect stop**

= остановка системы защиты. не работает для #38.

- подкоманда 87 **..FF .vendor specific**

= **_not_implemented/резервировано** для машиннозависимого оборудования, определяются производителем.

get_hardware_version

Получение информации об установленном оборудовании

(в данной версии все возвращаемые данные не вычисляются, а "вшиты" намертво)

*** код 1**

- подкоманда 0 **.get bios**

= версия vTR-DOS в двоично-десятичном виде (#0204 для 2.4)

- подкоманда 1 **.get machine**

= идентификатор + версия платформы в двоично-десятичном виде
(#0007 - #00=ATM #70= 7.0 version)

- подкоманда 2 **.get ram**

= количество страниц памяти (#0040 для 1024к)

- подкоманда 03..7F

= **_not_implemented**

- подкоманда 80..FF **.vendor specific**

= определяется производителем

set_vtrd_mount

Монтирование виртуального диска на "букву"

Может быть произведено монтирование только на какую-то одну "букву".

Функция не меняет текущий диск и статуса **is_current_drive_virtual**.

Для подключения смонтированного диска в работу необходимо любыми средствами инициализировать "букву" (например если текущий диск А и монтирование производится на букву А необходимо повторно выбрать этот диск).

*** код 2**

- подкоманда 0..3 **.drive**

= монтирование на букву А..D соответственно

- подкоманда 04..FF

= **_not_implemented**

get_vtrd_mount

Получение информации о статусе и монтировании виртуального диска

*** код 3**

- подкоманда 0 **.drive**

= возвращает номер смонтированного диска #00..#03
для дисков А..D соответственно

- подкоманда 1 **.is current drive virtual**

= возвращает #01 если текущий диск виртуальный и #00 в противном случае

set_vtrd_map

Замена/восстановление стандартной разводки памяти под RAM-диск,

По умолчанию соответствующей используемой в **honey commander 4.0**. карта памяти берётся по адресу = (#80 or подкоманда)*256.

В этой версии карта памяти должна состоять из номеров страниц #00..#7F, все иные значения заменяются идентификаторами недействительной страницы, а так же иметь длину 256 байт. Так как карта памяти в длину обычно меньше 256 байт, то последние байты должны устанавливаться значениями недействительной страницы равным #FF.

Номера страниц выше #40 являются страницами ПЗУ и нумеруются в обратном порядке, т.е. #40 на самом деле является #3F страницей ПЗУ.

*** код 4**

- подкоманда 0 .reset

= устанавливает стандартную разводку памяти

- подкоманда #01..#7F

= устанавливает пользовательскую разводку памяти, действительную до следующего сброса/инициализации

- подкоманда #80..#FF

= устанавливает пользовательскую разводку памяти, сохраняющуюся после сброса/инициализации

get_vtrd_map

Получение в буфер текущей карты разводки памяти
подкоманды не используются

*** код 5**

- дополнительный параметр 0 .is current map user

= возвращает #00 если используется стандартная карта памяти и иное значение если наоборот.

- параметр = **младший адрес буфера,**

дополнительные параметр = старший байт адреса буфера

= загружает в буфер 256 байт текущей карты памяти

set_page_hook

Установка/удаление обработчика обращений к недействительным страницам

Общий размер обработчиков не может быть больше **#1FFF** байт <для компиляции> стартовый адрес **РАБОТЫ** кодового блока обработчиков = **#4200**

Блок обработчиков загружается в страницу **#38** и выполняется из неё в окне **#4000**, прерывания запрещены.

Подкоманды не используются.

* код 6

- дополнительный параметр <#80 .reset
= отключение обработчика

- параметр = младший байт адреса дескриптора,
дополнительный параметр = старший байт адреса дескриптора (>#80)
= загрузка блока обработчиков обращений к недействительным страницам

формат дескриптора:

addr dw - адрес кодового блока обработчиков должен быть больше **#8000**

size dw - размер кодового блока обработчиков должен быть меньше **#1F00**

inpt dw - относительный адрес процедуры обрабатывающей событие ввода информации из недействительной страницы

outp dw - относительный адрес процедуры обрабатывающей событие ввода информации в недействительную страницу

Относительные адреса указываются относительно начала кодового блока.

Отработка событий ввода/вывода из недействительной страницы:

Имеется возможность контроля ввода-вывода информации в страницы помеченные в таблице как **#FF**.

Это необходимо если данная страница не находится в данный момент в памяти, а к примеру на винчестере. так же есть возможность, установив пользовательскую таблицу и заполнив её только **#FF**, производить прямой редирект обращения эмулятора из памяти на винчестер.

Достоинством такого метода является полное освобождение рам диска от **vTRD** и размещение оно на жестком диске или на удалённой машине.

Процедуры, осуществляющие редирект, устанавливаются так же, как и перехватчик **magic**.

Есть две процедуры - одна отвечает за ввод информации с носителя, вторая процедура отвечает за запись на носитель.

На вход эти процедуры получают:

ввод - **BC**= адрес в таблице страниц (указывающий на недействительную страницу),
DE= смещение в странице,
Введённый байт должен быть помещён в переменную **#7FF2 (#38:#3FF0+#02)**
(регистр SP сохранять по адресу **#4004 (#38:#0004)**,
возврат через **#3D2F** на **#0F76**)

вывод - /то же самое что и ввод/,
Выводимый байт следует искать в переменной **#7FF2 (#38:#3FF0+#02)**
(регистр SP сохранять по адресу **#4004 (#38:#0004)**,
возврат через **#3D2F** на **#0F79**)

Стек необходимо создать свой, внутри кодового блока или в заранее выделенной странице (в этом случае придётся самостоятельно позаботиться о подключении и отключении требуемой страницы). Все используемые регистры перед выходом восстановить.

get_page_hook

Получить информацию о том, установлены ли обработчики ввода/вывода из недействительных страниц

Подкоманды не используются.

*** код 7**

= на выходе **V=#C3** если есть обработчик ввода, **S=#C3** если есть обработчик вывода; иначе - любое число

set_nmi_hook

Установка/удаление обработчика нажатия клавиши magic

Производится копирование кодового блока в страницу #38 откуда он и будет работать, стартовый адрес **#6200**
подкоманды не используются.

*** код 8**

- **дополнительный параметр <#80**
= удаление обработчика прерывания NMI

- параметр = младший байт адреса дескриптора,
дополнительный параметр = старший байт адреса дескриптора (>#80)

Формат дескриптора:

addr dw - адрес кодового блока обработчика (должен быть больше #8000)
size dw - размер кодового блока обработчика (должен быть меньше #1000)

Обработка немаскируемого прерывания:

при нажатии клавиши *magic*, управление будет передаваться
на адрес #6200 (#38:#2200) после чего нужно сохранить
содержимое регистра SP по адресу #400F (#38:#000F).
для возврата в эмулятор следует использовать
стандартный вход #3D7E с последующим вызовом #0F73,
не забыв, естественно, восстановить все использованные регистры

get_nmi_hook

Получение статуса обработчика nmi
в данной версии не реализовано

* код 9

= _not_implemented

set_reset_hook

установка перехватчика нажатия клавиши RESET и инициализации vTR-DOS

В данной версии всегда копируется все 16кб данных, из которых 1-й байт и 3 последних
изменяются соответствующим образом для работы резидента (см. приложение 6)

Подкоманды не используются.

* код A

- параметр = младший байт адреса кодового блока,
дополнительный параметр = старший адрес кодового блока
(должен быть больше #40 и меньше #C0)

get_reset_hook

Получение сведений об установленном перехватчике reset/инициализации
в данной версии не реализовано

* код В

= _not_implemented

set_flag Установка флагов

* код С

- подкоманда 1 .wr
- дополнительный параметр 0..1
= **установка флага произведённой записи.**

- подкоманда 0 .ro
- дополнительный параметр 0..1
= **установка атрибута read_only для виртуального диска.**

- подкоманда 02..#FF
= _not_implemented

get_flag Получение состояния флагов

* код D

- подкоманда 0 .ro
= возвращает #00 если сброшен и #01 если установлен.

- подкоманда 1 .wr
= возвращает #00 если сброшен и #01 если установлен.

set_io_hook

Установка/удаление перехватчика доступа к расширенным портам

кодový блок копируется в страницу #38, стартовый адрес обработчика #7200
подкоманды не используются

* код E

- дополнительный параметр <#80
= **удаление перехватчика доступа к верхней памяти**

- параметр = **младший байт адреса дескриптора,**
дополнительный параметр = **старший байт адреса дескриптора (>#80)**

Формат дескриптора:

addr dw - адрес кодового блока обработчика (должен быть больше **#8000**)
size dw - размер кодового блока обработчика (должен быть меньше **#0D00**)

Обработка событий вывода в порты:

Сделана возможность использования перехватчика команды out (c),a и out (c),d если эмулятор не распознает обращения через вышеуказанные команды к портам ВГ, он передаст управление на **#38:#3200 (абсолютный - #7200)**.

При этом необходимо будет сохранить значение регистра SP и, установив новый стек внутри страницы #38, сохранить в нём все регистры.

Прерывания запрещены, менять нельзя. Для возврата в эмулятор после обработки (значение для вывода в порт передаётся в регистре D, адрес порта - в BC) необходимо восстановить все использовавшиеся регистры, кроме регистра SP, старое содержимое которого перед входом в эмулятор нужно поместить по абсолютному адресу **#4004**.

Затем нужно произвести вход в эмулятор через стандартный вход **#3D7E** на точку **#0F70**.

Далее эмулятор производит вывод в порт, указанный в регистре BC, значение, указанное в A. Если установлен флаг C, то вывод в порт не производится.

get_io_hook

**получение информации об установленном перехватчике обращений
к расширенным портам**

В данной версии не реализовано.

*** код F**

= _not_implemented

ПРИЛОЖЕНИЕ 1

Изменения в процедуре проверки наличия INTERFACE 1

Как было в обычном TR-DOS:

```
15672    XOR  A
          OUT (247),A
          IN  A,(247)
          CP  30
          JR  15684
          CP  31
          RET  NZ           ;если INTERFACE-1 нет, то выходим
15684    RST  8           ;если же он есть, то устанавливаем
          DEFB #31       ;его переменные,
15686    LD  A,1         ;пишем 1 и выходим.
          LD  (23791),A   ;
          RET            ;
```

Как переделано в vTR-DOS:

```
15672    LD  A,1         ;укороченная "процедура" проверки
          LD  (23791),A   ;INTERFACE-1. Просто автоматом ставит
          RET            ;признак его отсутствия и выходит.
15678    NOP            ;вход в керналь номер 1
          JP  KERNEL1
15682    NOP            ;вход в керналь номер 2
          JP  KERNEL2
15686    NOP            ;процедура проверки "стандартности" ПЗУ:
          XOR  A         ;если на выходе в регистре А и в ячейке
          LD  (23791),A   ;23791 находится 0, то это "наша" ПЗУ.
          RET            ;Если там - 1, то это ПЗУ – стандартное
                          ;(сравните с первоначальной процедурой).
```

ПРИЛОЖЕНИЕ 2

Структура ОЗУ ATM-turbo 2+ по умолчанию, с установленной прошивкой xBIOS.

#38	#39	#3A	#3B	#3C	#3D	#3E	#3F
#30	#31	#32	#33	#34	#35	#36	#37
#28	#29	#2A	#2B	#2C	#2D	#2E	#2F
#20	#21	#22	#23	#24	#25	#26	#27
#18	#19	#1A	#1B	#1C	#1D	#1E	#1F
#10	#11	#12	#13	#14	#15	#16	#17
#08	#09	#0A	#0B	#0C	#0D	#0E	#0F
#00	#01	#02	#03	#04	#05	#06	#07

<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> <div style="width: 45%;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: white; margin-right: 5px;"></div> <div>Свободное пространство (по умолчанию)</div> </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: #cccccc; margin-right: 5px;"></div> <div>Область программ пользователя</div> </div> <div style="display: flex; align-items: center;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: #808080; margin-right: 5px;"></div> <div>RAM-диск ОС vTR-DOS (по умолчанию)</div> </div> </div> <div style="width: 45%;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: #404040; margin-right: 5px;"></div> <div>Область резидентных процедур пользователя</div> </div> <div style="display: flex; align-items: center;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: black; margin-right: 5px;"></div> <div>Страница системных переменных vTR-DOS</div> </div> </div> </div>

Рисунок 2

***по умолчанию** — означает, что данная область памяти устанавливается таковым образом после первоначальной инициализации xBIOS и vTR-DOS. В дальнейшем возможно ее перемещение любым образом в соответствии с функциями vTR-DOS.

Отсутствие такой надписи означает, во-первых, неизменность расположения данной области, а, во-вторых, то, что занятие данных страниц памяти другими, релоцируемыми областями, может сделать невозможным полноценное функционирование xBIOS и, в особенности, vTR-DOS. Однако вы можете использовать любую страницу памяти по своему усмотрению, при условии, что функции vTR-DOS использованы не будут, а сама vTR-DOS будет отключена (то есть нижняя половина ПЗУ через **ROM-trigger** или иными способами установлена не будет).

В этой конфигурации ПЗУ xBIOS будет просто соответствовать стандартной прошивке ПЗУ ПК TURBO 2+ и Spectrum128k.

P.S. В последней версии xBIOS страницы #35 и #37 также отведены под RAM-диск (по умолчанию), вследствие увеличения эмуляции количества цилиндров на виртуальном дисковом с 80 до 84. Однако, если предполагается загрузить в RAM-диск стандартные образы TRD или инициализировать заголовок стандартным образом, то данные страницы можно использовать по своему усмотрению.

ПРИЛОЖЕНИЕ 3

Краткая информация о структуре страницы #38.

Внимание! Данные сведения приводятся лишь для исчерпывающей осведомленности пользователей. Конкретные системные переменные не описываются, ибо в процессе дальнейшего совершенствования прошивки некоторые из них могут поменяться. Посему для использования возможностей xBIOS настоятельно рекомендуется использовать функции кернели vTR-DOS.

Страница #38:

- #0000 - #0010 -** Блок внутренних переменных N1 (необходимые для использования см. в описании функций vTR-DOS).
- #0011 - #01FF -** Таблицы расположения страниц в RAM-диске.
- #0200 - #3FEF -** Пространство для обработчиков (плагинов):
- a) недействительных страниц (адреса **#0200 - #20FF**),
 - (адреса **#2100 - #21FF** – таблица расположения страниц в ROM-диске (используется в 1024Кб ПЗУ))
 - b) MAGIC (адреса **#2200 - #31FF**),
 - c) доступа к расширенным портам (команды out (c),a и out (c),d)
- (адреса **#3200 - #3FEF**)
Подробно работа с перехватчиками освещена в описании функций vTR-DOS
- #3FF0 - #3FFF -** Блок внутренних переменных N2 (необходимые для использования см. в описании функций vTR-DOS).

ПРИЛОЖЕНИЕ 4

Работа xBIOS v1.32XT с ПЗУ объемом более 128Кб.

eXtra BIOS v1.32XT, как уже упоминалось, рассчитан на работу с ПЗУ, объемом не менее в 128Кб. Однако это только нижний предел. Возможности ATM-turbo 2+ позволяют адресовать ПЗУ вплоть до 1024Кб включительно, и xBIOS сполна поддерживает эту возможность.

Существуют несколько видов ПЗУ, объемом выше 128Кб. Это:

- ROM 27020 – 256Кб (16 страниц)**
- ROM 27040 – 512Кб (32 страницы)**
- ROM 27080 – 1024Кб (64 страницы)**

Прошивки к этим ПЗУ можно легко создать самостоятельно путем добавления к базовой 128К прошивке новых страниц, без всяких переделок самой xBIOS. При этом надо неукоснительно соблюдать одно правило: все новые страницы добавляются снизу, каждый раз вытесняя базовые 8 страниц xBIOS в самый конец ПЗУ.

Схематично принцип расширения ПЗУ представлено на **рисунке 3**:



Рисунок 3

На этой схеме хорошо видно, что если в базовом ПЗУ 27010 (128Кб) xBIOS занимает весь объем адресов #00000-#1FFFF, то в других ПЗУ его расположение будет выглядеть так:

- ПЗУ 27020 (256К) – адреса #20000 - #3FFFF
- ПЗУ 27040 (512К) – адреса #60000 - #7FFFF
- ПЗУ 27080 (1024К) – адреса #E0000 - #FFFFFF

При этом порядок расположения страниц внутри самой xBIOS не меняется!

В случае с ПЗУ 27020 и 27040 наиболее рациональным будет использовать дополнительное пространство для размещения дополнительных модулей в формате, освещенном в описании Setup Utility. В этом случае данные модули можно будет легко запустить посредством интерактивного стартового меню.

При использовании же ПЗУ 27080 (1024Кб) появляется возможность использования свободного пространства не только в качестве склада модулей, но и как виртуальную дискету в формате TR-DOS. То есть, другими словами, vTR-DOS может работать с виртуальной дискетой как в ОЗУ, так и в ПЗУ совершенно равноправно (но либо в ОЗУ, либо в ПЗУ, но не одновременно. Придется программно переключаться между ними), точно также монтирую ROM-дискету к устройствам А, В, С или D. Единственное понятное исключение в работе – данная дискета будет доступна только на

чтение. Расположение ROM-диска а также общая структура такой ПЗУ 27080 показана на рисунке 4:

#38	#39	#3A	#3B	#3C	#3D	#3E	#3F
#30	#31	#32	#33	#34	#35	#36	#37
#28	#29	#2A	#2B	#2C	#2D	#2E	#2F
#20	#21	#22	#23	#24	#25	#26	#27
#18	#19	#1A	#1B	#1C	#1D	#1E	#1F
#10	#11	#12	#13	#14	#15	#16	#17
#08	#09	#0A	#0B	#0C	#0D	#0E	#0F
#00	#01	#02	#03	#04	#05	#06	#07

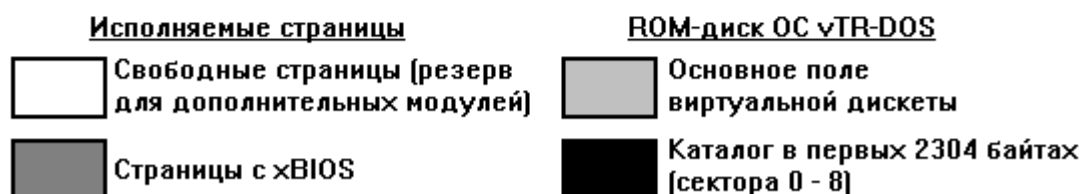


Рисунок 4

Сразу заметно, что ROM-диск занимает самые нижние 40 страниц ПЗУ, с каталогом в нулевой странице. Таким образом создание подобного ПЗУ на основе базовой прошивки xBIOS не представляет никакой сложности и происходит путем склейки трех файлов в указанной последовательности:

IMAGE.TRD + 256K.ROM +XBIOS.ROM

IMAGE.TRD – любой стандартный (скачанный или собранный скомпонованный) эмуляторный образ дискеты, которые легко можно найти в сети на ZX-сайтах.

256K.ROM – пустышка объемом в 256Кб: либо действительно пустая (в таком случае желательно заполнить ее кодом #FF), либо содержащая в себе модули пользователя.

XBIOS.ROM – собственно базовая прошивка xBIOS v1.32XT.

Однако ничто не может помешать вам использовать 1024Кб ROM целиком под свои модули. В этом случае попытка обратиться в ROM-дискету вызовет лишь стандартное сообщение TR-DOS «**disk error**», так как системой не будет найден каталог и системный сектор.

Подключить к АТМ-2+ ПЗУ объемом свыше 128Кб не сложно, но есть небольшая тонкость: как было сказано, диспетчер памяти позволяет адресовать до 1024Кб ОЗУ и ПЗУ. Однако на самой панельке под ПЗУ не разведено несколько дополнительных сигналов для «больших» ПЗУ. Их придется распаять вручную. Для этого надо:

У ПЗУ 27020 (D2) отогнуть пин 30 и припаять проводком к пину 11 микросхемы D75.

У ПЗУ 27040 отогнуть D2.30 и D2.31 и припаять **соответственно(!)** к D75.11 и D22.5.

У ПЗУ 27040 отогнуть D2.30, D2.31, D2.1 и припаять к D75.11, D22.5, D22.7.

ПРИЛОЖЕНИЕ 5

Работа с резидентом в xBIOS v1.32LE/XT

Понятие резидента

В ПЗУ xBIOS реализована так называемая функция резидента. Суть ее заключается в программном перехвате нажатия кнопки RESET или обращения к адресу #0000 в ПЗУ TR-DOS и vTR-DOS и передаче управления специальной резидентной программе, сохраненной особым образом в верхней памяти, ежели таковая, естественно, там обнаружится. И только в случае отсутствия в памяти резидентной программы, ПЗУ будет выполнять стандартную процедуру запуска стартового меню (в случае нажатия RESET), или автозапуска файла **boot ** в случае рестарта TR-DOS или vTR-DOS. Функция резидента не является нововведением именно xBIOS. Она была реализована в ПЗУ еще на ПК ATM-turbo 1, и только потом была перенесена на ATM-turbo 2/2+, где успешно применялась в стандартных ПЗУ 1.07.1x.

Самой известной программой, еще со времен ATM-1 использующей функцию резидента, по праву является файловая оболочка Honey Commander. При запуске какой-нибудь программы она сохраняет себя в качестве резидента в верхней памяти, а потом перезапускается автоматически по нажатию на RESET (что является очень удобным, учитывая тот факт, что большинство программ под ZX48/128 не имеют нормального выхода в систему), одновременно сохраняя скриншот последнего момента перед сбросом.

Однако, несмотря на давнее существование данной функции, более менее понятного описания ее использования с программной точки зрения так и не было создано. В результате желающие разобраться в ее работе с целью использования в своих программах, вынуждены были дизассемблировать ПЗУ и разбираться в процедурах обработчика резидента, что не всегда удобно, а для начинающих программистов слишком трудно. С выходом данной книги эта досадная историческая ошибка, наконец, исправлена.

Итак, резидент хранится в странице ОЗУ #1F. Когда-то давно, на компьютерах ATM-turbo 1 (v4.xx) и ATM-turbo 2 (v6.xx), у которых было всего 512Кб ОЗУ, это была самая последняя страница памяти. Но с появлением ATM-turbo 2+ (v7.xx) память увеличилась до 1024Кб и последней страницей ОЗУ стала #3F. Но резидент для совместимости со старыми программами править не стали (тем более, что не всегда на ATM-turbo 2+ ставили все 1024Кб. Часто по различным причинам обходились только 512Кб). Так и располагается сейчас резидент посреди всего ОЗУ в странице #1F.

Структура резидентной страницы

Чтобы программа, помещенная в страницу #1F была распознана как резидент, она должна быть оформлена специальным образом. В соответствие с этим, страница имеет следующую структуру (далее указывается смещение от начала страницы):

- Байт #0000** – код **#C3** (команда **JP nnnn**)
- Байты #0001-#0002** – адрес перехода для **JP nnnn** (рассчитывается по формуле: **#8000+относительный адрес начало программы в странице**).
- Байты #0003-#3FFC** – свободное место непосредственно под программу.
- Байт #3FFD** – контрольная сумма всей страницы (то есть, с **#0000** до **#3FFF**).
При создании резидента она рассчитывается путем сложения без учета переноса (то есть, по команде **ADD**) одного за другим всех байтов страницы. При этом, так как в процессе расчета контрольной суммы, она еще не занесена в байт **#3FFD**, он перед началом подсчета КС должен быть равен #00(!). И этот момент обязательно должен быть учтен при возникновении необходимости пересчета КС.
- Байт #3FFE** – всегда должен быть равен **#55**.
- Байт #3FFF** – всегда должен быть равен **#AA**.

Скомпонованная таким образом страница будет успешно распознана как резидентная и будет начата процедура запуска резидента, которая состоит из следующего:

- 1) По адресу **#C000** включается резидентная страница **#1F**.
- 2) Все содержимое резидентной страницы копируется по адресу **#8000** в уже установленную до этого (одновременно в двух картах диспетчера памяти) туда страницу **#02**.
- 3) Передается управление резиденту на адрес **#8000**, где должна располагаться команда резидента **JP nnnn**.

После передачи управления резиденту архитектура выглядит так:

Расположение стека (значение **SP**) – не определено. Остается на усмотрение резидента.

Карта памяти помимо адреса **#8000 - #BFFF** (с запущенным резидентом) – не определена. Также остается на усмотрение резидента.

Состояние палитры не определено. Также остается на усмотрение резидента.

Прерывания находятся в режиме 2. Включены. Вектор прерывания равен **#82FF (I=#82)**. Таким образом прерывания должны быть либо запрещены, либо, если работа с ними в короткий период работы резидента (пока он устанавливает основную программу) необходима, например, для установки палитры, то необходимо установить по адресу **#82FF-#8300** адрес подпрограммы обработчика прерываний (в резиденте **Honey Commander** там стоит указатель на ближайший **RET**).

Таким образом, программист при разработке резидента не связан никакими правилами и ограничениями, так как резидент не привязан ни к какой операционной системе и не обязан учитывать ее особенности. Какова будет конфигурация компьютера после рестарта в резидент, определяется самим резидентом.

Самое частое и перспективное применение резидента – сохранение с последующим восстановлением после рестарта различных ОС или оболочек. Однако в таких случаях объема одной страницы **#1F** может не хватить. Тогда для хранения избыточной информации используются соседние страницы. Например, **Honey Commander** использует

под резидент страницы #1F, #1E, #1D и #1C. А в vTR-DOS под резидент на всякий случай зарезервированы 5 страниц (исключены из RAM-диска): #1F, #1E, #1D, #1C и #1B.

ПРИЛОЖЕНИЕ 6

Адаптация HONEY-Commander v4.00 для работы с xBIOS.

Как известно, возможность использования ПЗУ, большего, чем 64Кб, была заложена изначально еще МикроАРТ - фирмой-создателем ПК АТМ-turbo 2+. Ими же был разработан стандарт расширения ПЗУ, по которому новые страницы присоединяются ниже уже существующих. Однако, в процессе разработки xBIOS выяснилось, что в одной из программных разработок МикроАРТ – HONEY-Commander v4.00 допущена ошибка. Вследствие этого, при переустановке карты памяти нарушается данный принцип. Это привело к тому, что HONEY может неправильно работать с прошивками, большими чем 64Кб. Ошибка проявляется при вызове функции резидента, нормальное действие которой становится невозможной.

Чтобы исправить этот недочет, надо в теле основного файла HONEY-Commander v4.00 сделать:

По смещению #1E38 заменить байт #9D на #81

По смещению #1E3C заменить байт #9F на #83

После проведенных изменений HONEY-Commander будет без проблем работать с ПЗУ любого объема от 64Кб до 1024Кб.

История версий vTR-DOS

vTR-dos 2.4SL (в разработке)

- адаптация для pentagon P-1024SL, особенности реализации :
- . рекомпиляция базовой версии vTR-dos 2.4XT
- . в этой адаптации не поддерживается работа с дисками ординарной плотности.

vTR-dos 2.5XT (в разработке)

- система driver management system (DMS)
 - обновлён мануал
-

vTR-dos 2.4XT

- релиз-версия для 128к пзу
- исправлен автозапуск с HDD
- устранена возникающая на недоработанных платах несовместимость короткой адресации порта FD
- новая подкоманда set_hardware_init.boot
- вместо гибридного управления защитой памяти от версии 2.4LE сделано полноценное
- ориентировано на работу с setup utility 1.0

- теперь в качестве пользовательских страниц рам диска можно указывать страницы пзу. сделана дополнительная карта ром-памяти.

- обновлён мануал

vTR-dos 2.4LE

- релиз-версия для 64к пзу

- исправлен недочет с автостартом vTRD A и возникающей при этом путаницей с профилем памяти

- развитие остановлено

- обновлён мануал

vTR-DOS 2.4

- версия "округлена" :)

- сделано новое загрузочное меню.

- исправлена проблема с неработающей клавиатурой в ПЗУ vTR-DOS, если во время запрещённых прерываний был нажат RESET.

- исправлена адресация на рам диске. теперь виртуальный диск можно адресовать до 84 цилиндра включительно. из таблицы удалена #38 страница с системными переменными. рам диск теперь находится до #36 страницы включительно.

- доработана команда **set hardware_init.hot**

- добавлена защита системной страницы.

- добавлена защита памяти от посягательств иных программ. К сожалению, **honey commander** не умеет "отчитываться" перед защитой памяти и при его использовании эту возможность необходимо автоматически отключать. Два режима работы : **ignore/abort**.

- добавлена блокировка системного порта. Два режима работы: **ignore/abort**.

- добавлена возможность блокирования верхней памяти. Введены три профиля : 128кб, 512кб, 1024кб.

- переделан set_io_hook

- создано новое меню

- обновлён мануал

vTR-DOS 2.03 beta

- пофиксены баги

фикс под honey commander 4.00 - более гибкая система определения прерываний

- добавлено

состояние прерываний запоминается в верхней памяти.

- известные глюки

если нажать сброс при запрещённых прерываниях, в дальнейшем может не работать опрос клавиатуры.

не работает real comander 1.96 (di:halt)

не загружается quick comander 3.05 (кривой загрузчик)

bestview 2.17 нормально работает только на виртуальной дискете :) не работает просмотр на реальном диске

vTR-DOS 2.02 beta

- пофиксены баги :

резидент

вход bios1

set_hardware_init.rst_0

- добавлено :

обязательное EI после отработки процедур биоса

подкоманда set_hardware_init.hot

- обновлён мануал

4.1.2005 ukms[z]/максагор

ukms@km.ru

vTR-DOS 2.01 beta

- пофиксены разные баги

- доделано большинство функций bios

- мануал

16.12.2004 ukms[z]

ukms@km.ru

Наши адреса

Интернет сайт: <http://www.nedopc.com>

Сайт поддержки АТМ: <http://atmturbo.narod.ru>

Электронные адреса:

chunin@infpres.com (Чунин Роман).

max_timonin@mail.ru (Тимонин Максим)

Автор благодарит за предоставленные материалы по резиденту Юрия Корсунина. А также выражает глубокую благодарность ukms[z] за разработку прошивки xBIOS, без которой не было бы и этой книги.

NedoPC group, Тимонин Максим, 2005 год.