ATARI XE

KMK/JŻ/IDEa

IDE Interface

User's manual

revised 10.11.2009

Table of Contents

Foreword

The KMK/JŻ IDE Interface allows you to attach an ATA (IDE) hard drive, a CF (Compact Flash) card or an ATAPI device (e.g. a CD-ROM drive) to your Atari 130XE computer. There are no special requirements on the device's manufacturer and model, the interface should work properly with most hard drives, CD-ROMs and CF cards. It should be taken into account, though, that the internal software – commonly known as IDE BIOS – that is burned into the EPROM and resides inside the Interface, knows only ATA devices and does not know ATAPI. This means that the system can only boot from a hard drive or a CF card (which are ATA devices), and to access CD-ROM or ZIP drive (which are ATAPI devices) you have to load additional software.

The general requirements to use the KMK/JŻ IDE Interface are as follows:

* a standard Atari 130XE or an Atari 800XE computer; a 65XE will also work, if the computer is equipped with the Enhanced Cartridge Interface (ECI) connector. Also more than 64k RAM is recommended.
* a floppy disk drive – this is only required for installation.
* an IDE hard drive (compliant to ATA-1 or later standard; at least ATA-2 is recommended).

Utilities, tools and the system software is being maintained actively and continuously developed. Software updates can be downloaded from the following URL:

http://drac030.krap.pl/en-pliki.php

We'd like to thank Larry White for many useful suggestions he had about this manual.

I. Preliminary information

The Interface's internal software (IDE BIOS) provides two modes of operation: native mode and emulation mode. The native mode uses a 512-byte physical block as a logical data sector, the emulation mode uses the physical block to store two 256-byte logical data sectors. Almost all existing DOSes require the emulation mode to work properly and limit the partition size to 16 MB. *The only one DOS, which can do sectors greater than 256 bytes, and is therefore capable of taking advantage of native-mode partitions larger than 16 MB (up to 32 MB per disk), is SpartaDOS X 4.39 (and later versions).*

Interface's firmware characteristics:

Maximum drive capacity: 16777215 physical blocks on each device (8 GB)

Maximum number of partitions: 16

Maximum capacity of a partition: 16777215 logical sectors (8 GB)

Logical sector length: 256 or 512 bytes; and 128 bytes as of BIOS version 1.8.

Average speed (depends on the drive and DOS used):

– 50-80 kilobytes per second (native mode, R/W), or

– 20-45 kilobytes per second (emulation mode, reading), or

– 7-12 kilobytes per second (emulation mode, writing).

16-bit data transfers.

The device is automatically found and initialized by the OS.

The device's operation is fully OS compliant, no patches are needed.

The interface's bus ID for the OS is configurable with a jumper.

Can boot from any partition. Independent partitions can be write-protected.

Note that there are four series of the KMK/JŻ IDE interface. They may differ in hardware characteristics, production quality and even components used, but all devices are compatible and can use the same software most of the time. Specifically, some interfaces feature the aforementioned jumpers, additional cartridge slots, XL adapters, while others do not. The greatest difference visible for software is that the oldest interfaces have 1,5k ROM and 512 bytes RAM, and newer interfaces have twice as much memory.

The utilities disk contains programs as follows:

1) KMKDIAG.COM – this is the diagnostic program you run as the very first program to see, if the interface and the drive work properly.

2) FDISK2.COM – the main utility to partition the disk and setup the interface, the FDISK II.

3) OLDFDISK.ARC – an archive that contains the old version of the above, the FDISK 1.40. It is included for your convenience, if you prefer it for any reason, we recommend however to use FDISK II instead.

4) BENCH.ARC – this archive contains two benchmark programs: RATEHD.COM, which tests the performance of the device itself (it talks directly to the disk, bypassing the operating system), and RWTEST.COM, which tests the speed of reading and writing files.

5) COPYWIPE.ARC – two utilities to duplicate and wipe the contents of a partition. Use with care.

6) FORCE.ARC – an utlity that performs media change.

7) HDB111.ARC – latest version of the IDE BIOS to burn into an EPROM.

8) LKUNLK.ARC – two programs to write protect and unprotect a partition.

9) MNT.ARC – a program that allows to "mount" any partition as D1:

10) NDCHK.ARC – lists bus IDs of the devices attached to the parallel bus.

11) SLAVE.ARC – an utility to attach slave disk to the system.

12) FL.ARC – File Loader, a program to execute binary games, demos etc. which can't be executed from DOS (a MyDOS-only program)

13) SDLOAD.ARC – SpartaDOS Loader, the function is similar to FL, just this one is a SpartaDOS-only program.

14) MKSDFS.ARC – a program to build SpartaDOS filesystems on partitions.

15) HDSC.ARC – a sector copy program (SpartaDOS only).

II. Basic setup (step by step)

First of all, please make sure, that the ECI/CARTRIDGE connector in your computer and the connector in the Interface are both very clean and not oxidized. Making sure, that the connection is solid, allows to avoid many odd problems you can experience due to contact issues between the Interface and the computer.

1) Make sure, that the power on your computer is turned off.

2) Insert the interface to the connector at the back side of the computer.

3) Connect the interface and a your IDE drive with the cable.

Caution: You may damage the drive if the interface cable is not connected properly. Make sure, that pin 1 on the cable (red line) is connected to pin 1 on the drive and on the interface (red dot on older versions of the interface – newer interfaces have a connector with „key" notch, and the pin one mark is stamped on the plastic flange of the IDE 40-pin connector of the interface).

4) Connect the power supply and the drive.

5) Insert the utilities disk into drive 1. The utilities disk contains MyDOS 4.53 on side A, and SpartaDOS 3.2d on side B, boot whatever you prefer of these two. Note that if you're a regular Atari user, it may be a good idea to prepare a boot floppy, that is properly configured for your setup (especially a ramdisk may be crucial), copy programs from our utility diskette onto that floppy, and boot from it.

6) Turn the power on the hard drive on and wait a while.

7) Turn the power on the computer on, holding down the SHIFT key. Make sure, if the drive is spinning up properly (most drives will start spinning up after you turn the computer on). If not, turn the power off, check the connections and try again. Note, that some very old drives (such as 3,5 inch/40 MB Seagate ST-157A) need **a lot** of power, so an ordinary Atari power supply may not be sufficient.

8) After the usual blue screen appears, the computer should boot the Utilities disk.

9) When the DOS is ready to use, execute the KMKDIAG.COM file. This is a diagnostic program, that may help you to make sure that both the interface and the drive work properly.

10) When the KMKDIAG.COM completes with no errors, quit it. **If it hangs at this point, just press RESET**: this happens with the first revisions of the IDEa interface (let us call that „IDEa revision A"). The IDEa rev. A interface is not fully compatible with its predecessor, KMK/JŻ IDE v.1.1, and a small patch needs to be done on the IDEa PCB. The patch cures this problem and simultaneously upgrades the IDEa interface to „revision B".

11) Execute the FDISK2.COM file.

12) A disk selection menu should appear, and an information about disk geometry at the boottom of the screen. If you see a message like **KMK/JZ IDE not found** instead (which means that the FDISK2 is unable to find the interface), please turn the power off, check the connection between the interface and the computer, then try again. Check the Master/Slave jumper position on the drive.

13) Select the drive to be partitioned. The bottom line marked **Sectors left** will indicate the total amount of sectors, that have not been allocated yet; **Phys size** indicates the size of a partition in physical (512-byte) sectors; **Log size** the size of a partition in logical sectors; **BPS** selects the logical sector size (as seen by DOS); **PSt** (Partition Status) indicates whether the partition is active or not, and **Atr** (Attributes) indicate additional status information (**B** – boot partition, **R** – D1: is redirected to this partition, **L** – the partition is write-protected).
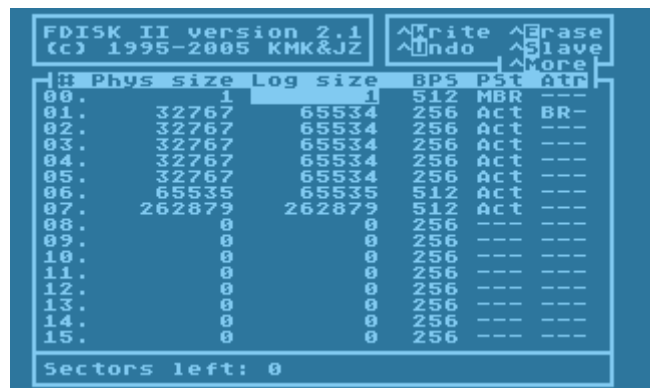
14) Use arrows to select a drive you want to be a partition.

15) Press the RETURN key and type in a number of logical sectors, that you want to be allocated to the first partition (number „01"). Note, that the first entry

(number „00") refers to the MBR (Master Boot Record), which is normally invisible to the system (it is only used at boot time to setup the interface and the disk). You normally don't need to change its size, although advanced users may want this.

16) If the logical sectors you prefer are smaller than 512 bytes, allocate an even number of them and don't go beyond 65534. Note that only SpartaDOS X 4.39 and later revisions will work with a 512-byte logical sector.

17) Press RETURN to confirm the number of sectors. Do not setup more partitions for now, just the D1: If you want to cancel the numbers you entered and exit the edit field, press ESC, then start again.

```
FDISK II version 2.1    ^Write ^Erase
(c) 1995-2005 KMK&JZ     ^Undo  ^Slave
                               ^More
# Phys size  Log size    BPS PSt Atr
00.         1          1  512 MBR ---
01.     32767      65534  256 Act BR-
02.     32767      65534  256 Act ---
03.     32767      65534  256 Act ---
04.     32767      65534  256 Act ---
05.     32767      65534  256 Act ---
06.     65535      65535  512 Act ---
07.    262879     262879  512 Act ---
08.         0          0  256 --- ---
09.         0          0  256 --- ---
10.         0          0  256 --- ---
11.         0          0  256 --- ---
12.         0          0  256 --- ---
13.         0          0  256 --- ---
14.         0          0  256 --- ---
15.         0          0  256 --- ---
Sectors left: 0
```

18) After you have allocated sectors for the first partition, position the cursor on its entry and hit Control/B and Control/R so that letters „B" and „R" appear in the rightmost column. This marks the D1: as bootable and redirects the D1: to itself (the interface allows you to „swap" D1: with any other partition, if it is needed).

19) SpartaDOS X and MyDOS: on FDISK2 version < 2.4 **mark the partition as *active* by hitting Control/A. If you do not do that, the partition will occupy the numer of sectors you selected, but will not respond to DOS commands.** As of version 2.4 the FDISK2 will do it automatically for you – the "Act" mark will appear immediately after you enter a nonzero number as partition size.

SpartaDOS 3.2d: mark the partition as *inactive*, i.e. hit Control/A until the „Act" mark disappears from the „PSt" column.

20) Hit Control/W to write the partition table out to the disk. **If you didn't boot from the hard drive, the partition table is only written to the disk, but it does not get automatically activated. This allows you to return to MyDOS (or any other DOS with non-resident DUP.SYS) flawlessly.**

21) Hit ESC twice to quit FDISK2. The remaining steps to do depend on the DOS you use.

In SpartaDOS 3.2d

22) Reboot the computer from the Utilities floppy. **This step is crucial**, because SpartaDOS 3.2 sets itself up differently when booted in the presence of a PBI device. If you just hit the RESET key to activate the IDE disk after you have set up it with FDISK, and then attempt to access a partition, SpartaDOS 3.2 will get horribly confused („Error – 139" syndrome)

23) Change the Utilities floppy number to 2 (i.e. change its number in APE, SIO2BSD or similar peripheral emulator, or insert it into floppy drive 2, or switch the floppy driver number).

24) Run FDISK2.COM again and enter the partition list.

25) Activate the first partition using Control/A so that the „Act" mark appear in the appropriate column.

26) Write the partition table out to the disk (Control/W)

27) Hit ESC twice to quit FDISK2

28) Hit RESET

29) Run MKSDFS.COM from the Utilities floppy (which should be still available as D2:), and give it „D1:" as a parameter. The command line you use should look like this: `D2:MKSDFS D1:`

30) After the program completes and quits to the DOS, a „DIR D1:" should list the directory of your partition.

31) Now do COPY D2:X32D.DOS D1:

32) When this succeeds, do BOOT D1:X32D.DOS

33) Reboot. Now SpartaDOS 3.2d should boot from the hard disk. The basic setup is complete.

In SpartaDOS X

22) When the command processor prompt appears, **hit the RESET key** to activate the partition table, and then type **FORMAT**. In the formatter, use the *B – Build directory* to write the filesystem structure to each of your partitions.

23) Exit the formatter and reboot. The basic setup is complete.

In MyDOS

22) **Wait for the DUP.SYS to load, and then hit RESET.** This should activate your freshly made partition table and make the disks accessible. To deactivate it back, if necessary, press RESET holding down the SHIFT key.

23) Use the **O** configuration command to tell MyDOS about the hard drive – the point is MyDOS must know how many sectors the drive has.

24) Use the I command with the **N** switch e.g. 1/N so as to soft format the directory on the drive.

25) Write the DOS file(s) to the BOOT partition. In MyDOS, select the **H** option from the DUP menu.

26) Reboot. The DOS will load itself from the drive – the basic setup is completed.

Note: Some IDE drives used to clear the BUSY and set the READY bits in their internal status registers **before** the spin-up process is finished – the drive looks to be ready, but isn't ready in fact and cannot execute any commands (very strange, by the way...). To prevent such troubles during power up, the internal software waits about 15-20 seconds before taking any action with the IDE controller. This delay is not necessary, if the cold boot was done without powering the system down. In such case the internal software uses a fast initialization method. However, if you turn the power switch off and on very quickly, the initial routines may not recognize this boot process as a real power up. As a result, the boot process will fail. To prevent such problems, after turning the power off, you should always wait 10-15 seconds before turning it on again. This time should be sufficient to invalidate internal flags, that have been located in RAM.

III. Advanced setup

More partitions

The interface allows you to setup up to 16 partitions, however, their exact number visible by the operating system, and the maximum size you may allocate for them depends on the DOS you use. Most DOS-es (including MyDOS and SpartaDOS 3.2d) will only see 8 partitions (D1: - D8:), up to 16 MB each. SpartaDOS X 4.20 can see one parition more, (D1: - D9:), but the size remains 16 MB.

Before proceeding, it is advisable to copy the FDISK2.COM program onto the first partition you just have created, i.e. D1:. Note that hard drives have priority over serial floppies, so that if you setup D2: as a partition, the corresponding floppy disk stops to be visible in the system. Copying FDISK2.COM onto the hard disk will save you a problem with enabling the floppy access back after that.

To setup a 16 MB partition, run FDISK2.COM, open the partition list, select the desired partition number, choose „256" in the **BPS** (bytes per sector) column, then geto into the **Log size** column and type in the desired number of sectors. Choosing an even number allocates the disk space optimally, thus the optimum here is 65534 sectors, which makes a 16 MB less 0,5 KB.

SpartaDOS X as of version 4.39 can access up to 15 partitions (D1: - D15: or rather A: - O:), up to 32 MB each. A 32 MB partition consists of 65535 sectors, 512 bytes each (which, to be specific, makes 32 MB less 0,5 KB). The steps to create one are similar as above, just the sector size choosen in FDISK2 must be „512", and the number of sector does not have to be even.

The disk 16 is not visible with any DOS, but it still can be used and the computer can even be booted from it. The easiest method is to temporarily redirect the D1: to D16: by positioning the cursor at „16" in the partition list and hitting Control/R. This way you may e.g. boot a game from D16: (also see below).

Note that FDISK2 before version 2.4 did not automatically activate the partition. If you use such a version, either hit Control/A to make the freshly created partition visible („Act" must appear in the **PSt** column), or better upgrade to a newer version.

Also note that FDISK2 allocates disk space consecutively. That means, that changing the size of a partition will cause all partitions with higher IDs to change location. Since the data are not moved physically, you may be badly surprised, when you discover, that, after you have changed the size of D1:, the contents of all other disks from D2: onwards is gone. To avoid that, you have to keep an eye on the sector number where the partitions start (Control/D in FDISK2 partition list will display these numbers). Experiment with it for a while before you decide to setup the disk permanently and copy software onto it.

Setting up boot disk

In FDISK2 position the cursor on a partition that you want to be your boot disk and hit Control/B. This is enough for SpartaDOS X version >= 4.30, after that it will boot from the drive you selected.

Older versions of SpartaDOS X will unfortunately not boot from anything else than D1:, you cannot change this. The best thing you can do is to copy CONFIG.SYS etc. onto partition D1: and change the number of the floppy drive to D2:. Alternatively you may redirect D1: to another drive, but the DOS will think that it is D1: anyway.

DOS-es which are entirely loaded from the disk, like SpartaDOS 3.2d or MyDOS, may also be booted from any disk. To do that, hit Control/C while in FDISK2 partition list. This enables the „Custom boot" routine. Its activity is indicated by a „C" letter appearing in the MBR attributes. Disable this, if you want to boot the DOS from D1:; the interface will then pass the control over the boot process to the operating system. Disable this option, if you have any troubles with the „custom" boot up – the interface, taking the control over the boot process, uses some tricks that may not work with some DOSes, cartridges or customized

operating systems. Also, this **never** works correctly with SpartaDOS X, no matter what version.

IV. Keystrokes

SHIFT/RESET disables the drive. The drive will remain spinning, but the partitions will not respond to operating system requests (error 138). To enable it again just press the RESET key.

SELECT/RESET forces the cold boot. This option is unfortunately removed in BIOS versions later than 1.5 and before 2.0 (apart of 1.10NC, which has this function built-in).

V. Write-protection

The IDE hard drive interface provides the write-protection feature to minimize a risk of accidental data damage caused by viruses, damaged software or children. When a partition is locked, there's NO POSSIBILITY to write data to this partition or unlock it by sending commands, causing a cold boot or turning the power off and on. Damaging a write-protected partition by writing accidental data to random memory locations is also practically impossible – the risk is very small.

You can mark a partition write-protected using the „Lock" option (Control/L) in FDISK2.

VI. Executing non-DOS software with the hard drive

Some software, especially games and demos, have their own disk formats and cannot be copied to a partition. However, the IDE software provides limited ability to execute such programs.

If you selected the „custom" hard drive boot, while partitioning the drive and your BOOT partition is not the D1:, you may run the non-DOS disk from the floppy drive. To do it, insert the disk into the floppy disk drive and reboot holding

down the SHIFT key. The internal software of the IDE Interface will pass its initialization routines by and your computer will boot up from the floppy. The hard drive will be „invisible" for the system.

If you want to execute such software from the hard drive, you must provide a small partition to use it in such a manner. The D10-D15 partitions are invisible for existing operating systems and cannot be accessed by DOS (except SpartaDOS X >= 4.39), they may be booted, though. When you are partitioning your drive, create a small (up to 520 physical blocks) disk, for example D10:. On BIOS >= 1.8 it may be useful to set the logical sector size to 128 BPS for such a partition (this improves compatibility with some software, especially DOSes). When you complete the installation and make the drive work, you have to use a sector copy (e.g. HDSC.EXE) to copy your floppy contents to the D10:.

If your interface has BIOS version earlier than 1.10, we recommend to do an upgrade. The 128 BPS emulation mode in BIOS 1.8 and 1.9 has a bug, that makes it not fully compatible with software that requires single or enhanced density.

Now execute the MNT.COM, mount the LUN 10 onto D1:, and reboot the system – the non-DOS program will load from the hard drive. To return to the previous configuration, insert the utilities disk to your floppy disk drive, reboot the system holding down the SHIFT key, then execute the FDISK, reset the D1: redirection and „Custom boot" to the previous state, write the new partition table, exit the FDISK and reboot the system.

VII. Dual drive configuration

If you want to connect two IDE hard drives to the interface, enable slave probing in FDISK2. This is accomplished by hitting Control/S. The slave probing state is indicated by „S" letter appearing in the MBR attributes. If you use only one disk, disable the slave probe. Notice that the FDISK tends to disable this option automatically, when no slave disk was detected at initial probing, and reversely it tends to keep this option enabled when it knows that the slave disk is present.

A slave drive can be setup in FDISK2 after its name is chosen from the initial disk selection menu. Note that every disk is setup just like it was the master disk, however, settings like boot drive number etc. are read only from the actual master (in other words, these settings on a slave disk will get taken into account once the slave becomes the master).

IDE BIOS versions as of 1.10 should initialize slave drives automatically at every RESET, if the slave probe is enabled in FDISK. *Earlier BIOS-es as of version 1.5 also contain the relevant function, but the code is faulty and doesn't setup the slave partitions properly.* Partition tables of both master and slave disk are merged, so that the slave partitions would appear past the last DOS drive number occupied by the master drive. For example, if the master drive partitions are assigned as DOS drive numbers from D1: do D5:, and the slave drive has two own partitions appearing as D1: and D2:, the final drive assignment will be as follows:

D1: - master partition nr 1

D2: - master partition nr 2

D3: - master partition nr 3

D4: - master partition nr 4

D5: - master partition nr 5

D6: - slave partition nr 1

D7: - slave partition nr 2

and so on.

For the BIOS versions 1.0-1.9 the slave drive must be "mounted" manually. You can do that using the program SLAVE.COM which can be found on our utilities disk. Before you execute the SLAVE.COM for the first time after the slave drive was powered on, it is strongly recommended to execute the KMKDIAG.COM file and run the drive test it offers.

VIII. Software development information

The IDE drive partitions operate as normal floppy drives or ramdisks and can be accessed via OS DISKINT ($E453) and SIOINT ($E459) routines. All the partitions recognize the following commands:

1) Standard subset

R – read sector – this command reads a specified logical sector from a specified partition.

In native mode it reads ALWAYS THE ENTIRE LOGICAL SECTOR, i.e. 512 bytes, regardless of the DBYT ($0308-$0309) value. In emulation mode BIOS versions up to (and including) 1.7 ignore the DBYT value too, and transfer 256 bytes. As of version 1.8 the DBYT value is taken into account in the emulation mode and, when DBYT is 128, 128 bytes are transferred, and 256 otherwise.

The computer's memory area at $D000-$DFFF is protected by the IDE BIOS, no direct (AKA burst) reads to these addresses can take place. Any attempt to do that will cause the interface to generate error 139 (NAK). Thus any read operations storing data to this area must be buffered by the loading software.

The sector number is a 24-bit value, the most significant byte (now called DAUX3) is located at $0307 (this byte was unused by the XL OS). Sector numbers less than $000001 or greater than maximum sector number for the specified partition are invalid and will cause error 144 (Write protected or bad sector) to occur.

P – put sector – writes data to a specified logical sector on a specified partition. There are the same restrictions, as mentioned above. This command will also return status 144 when attempting to execute on a write–protected partition.

W – write sector – the same, as "P" command.

S – read status block – transfers the 4–byte disk status to the memory. The bits of the first byte are as follows:

7 – enhanced density (set for partitions with 128-byte-sector emulation)

6 – not used

5 – 128 BPS (0) or more (1)

4 – not used

3 – the disk is write protected

2 – not used

1 – not used

0 – not used

The second byte contains reversed (eor'ed with $FF) value of the IDE controller error register. The bits are as follows:

7 – BBD – Bad block detected

6 – ECC – Error correction code (uncorrectable error)

5 – NUL – unused, always 1

4 – IDNF – ID not found (target sector could not be found)

3 – NUL – unused, always 1

2 – AC – Aborted command

1 – TK0 – Track 0 error (unable to find a valid track 0)

0 – DAMNF – Data address mark not found

The normal (default) value of this byte is $FF. The next byte has a dummy value $E0. The last byte of the status block is don't care.

N – read configuration – reads the 12-byte PERCOM block to the memory. The values returned by a partition are as follows:

0 – number of tracks in range from 1 to 34. This is the multiplier for the value stored in bytes 2,3,4.

1 – BIOS revision number ($10 = 1.0, $19 = 1.9, $1a = 1.10 and so on)

2 – total number of logical sectors, the middle byte

3 – total number of logical sectors, the low byte

4 – total number of logical sectors, the high byte

5 – additional information:

  bit 3 – the byte 4 contains the high byte of the size of the disk in sectors

  bit 2 – double density drive (always 1)

  bit 1 – 8 inch floppy disk drive (always 0)

6 – number of bytes per logical sector, high byte

7 – number of bytes per logical sector, low byte

8 – unused, always $FF

9 – don't care

10 – don't care

11 – don't care

For compatibility reasons, this commands causes error 139 (NAK) to be returned for partitions with 128-byte-sector emulation enabled.

2) Specific ones

$E6 – sleep drive – stops the drives and deactivates their internal controllers. See ALL RESET command for the DCB variables details. This command needs the number of any partition stored to the DUNIT($0301) (as of BIOS version 1.6 the DUNIT value may be 0 for this operation). This command doesn't do data transfer – the appropriate DSTATS ($0303) value is zero.

$E7 – all reset – resets, recalibrates and reinitializes both hard drives. It is the only way to exit the Sleep mode. This command needs the number of any partition stored to the DUNIT($0301) (as of BIOS version 1.6 the DUNIT value may be 0 for this operation). This command doesn't do data transfer – the appropriate DSTATS ($0303) value is zero.

$EE – force media change – forces the interface to re-read the partition table from the drive. This command doesn't do data transfer to the user specified buffer, so the appropriate DSTATS ($0303) value is zero.

$EC – identify drive – transfers the 512 bytes of data, that specify the drive's parameters. The fields are as follows (F = fixed value, V = variable, R = reserved, should be zero):

0 – vendor specific information, bits are as follows:

    15 – 0, reserved for non–magnetic devices (F)

    14 – vendor specific (F)

    13 – vendor specific (F)

    12 – vendor specific (F)

11 – vendor specific (F)

10 – vendor specific (F)

9 – vendor specific (F)

8 – vendor specific (F)

7 – removable media device, if 1 (F)

6 – removable controller and/or device, if 1 (F)

5 – vendor specific (F)

4 – vendor specific (F)

3 – vendor specific (F)

2 – vendor specific (F)

1 – vendor specific (F)

0 – reserved (R)

2 – number of cylinders (F)

4 – reserved (R)

6 – number of heads (F)

8 – vendor specific

10 – vendor specific

12 – number of sectors per track (F)

14 – vendor specific

16 – vendor specific

18 – vendor specific

20–39 – serial number, ASCII characters (F)

40 – vendor specific

42 – vendor specific

44 – number of ECC bytes transferred on LONG operations (F)

46–53 – firmware revision, ASCII characters (F)

54–93 – controller model number, ASCII characters (F)

94 – numbers of sectors/interrupt R/W multiples, bits:

   15–8 – vendor specific

   7–0 – $00 = READ/WRITE MULTIPLE not implemented (F)

$01–$FF = maximum number of sectors that can be transferred per interrupt on READ/WRITE MULTIPLE commands (F).

96 – reserved (R)

98 – capabilities, bits:

 15 – reserved (R)

 14 – reserved (R)

 13 – 1 = standard standby timer values are supported

   0 = standby timer values are vendor specific (F)

 12 – reserved (R)

 11 – 1 = IORDY supported (F)

   0 = IORDY may be supported (F)

 10 – 1 = IORDY can be disabled (F)

 9 – 1 = LBA supported (F)

 8 – 1 = DMA supported (F)

 7–0 – vendor specific (F)

100 – reserved (R)

102 – PIO data transfer cycle timing (F)

104 – DMA data transfer cycle timing (F)

106–511 – reserved

All values are in standard low/high convention. Some parameters are defined as a string of ASCII characters. For the string "Copyright", the character "C" is the first byte, "o" is the second byte etc. When such fields are transferred, the order of transmission is:

– the 1st character ("C") is on bits 15 through 8 of the 1st word

– the 2nd character ("o") is on bits 7 through 0 of the 1st word

– the 3rd character ("p") is on bits 15 through 8 of the 2nd word

– the 4th character ("y") is on bits 7 through 0 of the 2nd word

etc.

Note, that the DMA transfers, although may be supported by the drive itself, may not be supported by the current version of the interface's hardware. Please

also refer the SLEEP DRIVE command to get an information about the DCB variables.

All other commands will cause error 139 (Negative Acknowledge). Note, that the software does not provide a FORMAT DISK command – it hasn't been implemented to prevent an accidental data damage. The drive must be „formatted" using a separate program. Also note that IDE drives are preformatted at the factory and no physical formatting can in fact take place.

Operating system developers should note, that the internal software of the IDE Interface changes the DUNIT ($0301) to the BOOT partition number during boot up.

IX. Memory usage

The Interface's internal software uses the following RAM locations: $30–$35 and $38–$40. Older versions of the BIOS may also use the location $01. The PDVMSK ($0247), PDVRS ($0248) and DCB variables (especially DAUX3 $0307) should be used only in their proper functions (please DO NOT use them as temporary data registers!). The $0400-$06FF area should also remain intact during the cold boot.

X. Error messages

You can get the following error reports from the IDE drive:

138 – Timeout error – attempting to read or write data to a partition, that is physically allocated to the slave drive, while the slave drive is busy, not ready or does not exist at all; or attempting to assert the ALL RESET command, while the master drive is not present. It may occur, if you accidentally disconnect the master drive or disconnect the slave drive without reconfiguring your system. It may also indicate a damaged partition table – please reboot your system. If this action

doesn't cause any effect, you must use the FDISK to repair the partition table. See also section **VIII. Dual drive configuration**.

139 – Invalid command. The command is not known to the Interface.

144 – Device done error:

1) the software attempted to write data to a write–protected partition.

2) the software attempted to read or write data outside of the limits, which are valid for the partition (sector number below 1 or past the end of the partition).

3) there is a bad sector on the partition. Please send the status command to get the value of the internal IDE error register.

4) the interface's software is unable to handle your drive. Please run the KMKDIAG. If the test fails and you know, that your drive is in good condition (no bad blocks) for sure, please send the IDENTIFY DRIVE command (from BASIC, for example), copy the buffer to a file and send the file to us.

XI. Tuning

1) Bus ID

Oldest interfaces – these with BIOS versions 1.0-1.2 – and also the latest IDEa series have jumpers, that allow to determine the bus ID to be occupied by the interface. The bus ID is the number, that is displayed by the NDCHK.COM program in the "Dev #n" column - the digit 'n' shown there is the bus ID. You may want to allocate different bus ID for the interface to avoid conflicts with some other parallel bus device you possibly use.

You may like to know, that the bus ID you select may have an influence on the drive's performance. The XL OS scans the parallel bus devices starting from #0 and going towards #7. The sooner it finds the interface doing that, the faster is the disk's operation, so keeping the interface at bus ID #0 is the best option most of the time.

However, SpartaDOS scans the devices in reverse order – it starts at #7 and goes towards #0. So if you use this DOS, changing the bus ID to #7 will improve the general performance.

If your interface allows you to determine the bus ID, you can experiment with it for a while using the NDCHK.COM and RWTEST.COM, which setting is better for you.

2) Cartridge slot

Since the interface occupies the CART/ECI connector of the XE computers, simultaneous use of a cartridge may be a problem. Thus most interfaces have an additional cartridge slot either on the main PCB or (the IDEa series) on the CART/ECI connector.

Some interfaces however lack the additional cartridge slot, and you can't directly use a cartridge with them. This may be a problem, if you use SpartaDOS X or BASIC XE regularly. In this case you have to add the cartridge slot yourself. It is not difficult, because all interfaces without the cartridge slot do have a place on the PCB for the slot to be soldered. You can see it when you open the interface's chassis – the pinout is right near the CART connector. The slot must be soldered there, it is a simple task, but if you have no experience in doing such things, better ask someone else to do that for you.

XII. Troubleshooting

1.

Problem: The FDISK does not work, it only displays „KMK/JZ IDE not found" message at startup and exits.

Answer: Such a message means, that the FDISK hasn't detected the presence of the IDE Interface. You must carefully check the connection between the Interface and the computer. This problem is unrelated to the drive you used.

2.

Problem: The FDISK does not work, it only displays „Master drive does not respond" message at startup and then exits.

Answer: This message means, that the FDISK has found the Interface, but is unable to communicate with the drive; the drive is „invisible", just as if there was no drive attached at all. This may mean that the drive is not connected properly, or it is powered off, or it is damaged; this may be a symptom of unstable connection between the Interface and the computer; this may also mean, that the drive you attached is a Western Digital Caviar and the IDE BIOS is older than version 1.6. In this case you have to upgrade to a newer version of the BIOS (download it from the support page and burn the downloaded file into the Interface's EPROM).

3.

Problem: The FDISK detects the drive, but its name is improperly displayed, there are garbage characters and/or the string is cut down at the beginning (it says f.e. „TUM FIREBALL" while it should rather say „QUANTUM FIREBALL").

Answer: This is an alarming symptom of some hardware problem with data transfer between the disk and the computer. Such a setup is unlikely to work properly. Please run KMKDIAG.COM to find out what's wrong. You may try to fix the problem cleaning the connectors in the interface **and** in the computer with a good contact cleaner. If this does not help, you should contact the reseller for replacement or instructions.

4.

Problem: After I leave FDISK and want to return to MyDOS, the computer hangs.

Answer: This problem should be gone as of FDISK version 2.0. If you still use FDISK v.1.40 for a reason, here's a solution:

Check if the DOS sets up a ramdisk and copies the DUP.SYS into it. If not, you should first configure the MyDOS so that it does it (or use a copy of MyDOS that is already configured for your setup). After that, FDISK should exit to DOS flaw-

lessly. If you can't setup a ramdisk, try hitting RESET after you select the FDISK's „Exit to DOS" function. This should bring the DUP.SYS menu back to the screen.

5.

Problem: How do I build filesystems on partitions? When I use MyDOS's „Initialize disk" menu item, it reports an error.

Answer: Under MyDOS always do partition formatting using <drivenumber>/N. For example, if you want to format D2: partition, please type 2/N at MyDOS „drive number" prompt. This prevents the MyDOS from an attempt to physically format the disk (such an attempt must fail, because IDE drives are formatted at the factory and cannot be physically reformatted).

6.

Problem: The "custom boot" option does not appear to work. When I enable this, booting fails.

Answer: The FDISK II versions from 2.0 to 2.2 have a bug that causes them to install faulty booter in the MBR. To solve this problem, you must upgrade to FDISK II version 2.3 (or newer), and replace the booter. To do that it is enough to execute its "Write" function – the FDISK always replaces the entire MBR at that occasion, including the custom booter.

Other known problems:

1. Some CF cards (PQI for example) don't work properly with interface. We recommend using SanDISK CF cards. Note that some types of CF cards have a very low limit on the number of writes-per-sector (such as 100,000 or even 10,000 writes per sector). If you have such a card, we recommend purchasing a better one: some system areas in Atari filesystems – and in all other filesystems too, for that matter – are written to very often, thus you can exceed this limit rather quickly rendering the CF unusable. Another solution is to backup everything on a real hard drive as frequently as possible.

2. If you cannot bring interface to work, please check if there is a small delay (a few seconds) after turning on the Atari computer. If the usual blue screen appears as always (w/o delay) then your interface needs tuning of the timing circuit. It may be caused by too many expansions installed inside your Atari.

3. As it was said above, several different versions of the hardware was produced over the last 12 years. There were at least three designs, the actual production was licensed to different manufacturers, various parts were used, and so on. Thus different series of interfaces may present slightly different behaviour in the sense that some may be more tolerant to differences between disk types and between specific computers (such as between an older series 130XE and 800XE), while others are more sensitive. It must be taken into account that the interface actually interfaces an industry standard ATA drive controller to the 6502 bus. So it may be, that the specific series of interfaces doesn't work well with specific types of drives, while another series of the same interface does well with these drives, and other drives do well with both. For example, the IDEa interface is known to dislike certain types of Conner drives, while the stock KMK/JŻ IDE works very well with these. Thus on problems with communication stability it is usually advisable to try first a different drive type.