IDE Plus 2.0 Configuration Screen Guide

Copyright (c) 2011-2023 by JŻ and KMK

Revised 20/05/2023

Table of Contents

I. Basic function keys

Some of the interface's functions may be invoked by holding down certain keys or key combos during OS reset, either cold or warm. These keys are:

1) Start – this invokes the main configuration screen (explained below)
2) X – disable the SpartaDOS X module during cold restart (as of ver. 1.4)
3) Shift/Control/Return – this disables the entire device

You normally do not need to use the last combo. Its whole purpose is to force the OS to skip the device while initializing the entire system. It may only be necessary, when the interface's ROM went partially bad because of a damage. Also, it only works while cold reset (cold reboot) is taking place.

The first one, quite contrary, will probably be used by you very often.

If these combos have to take effect, the key(s) must be held down first, then you have to hit the Reset key or force a cold restart as necessary. After that, do not release the key(s) immediately, but wait a little (up to 3 seconds) while the OS initialization routine is executing.

II. The main configuration screen

The Start+Reset key combo invokes a menu, which may look like this:

The time information shown in the paenultimate line should be constantly changing, once a second.

The top line contains the device's name („IDE+2.0") plus copyright information. The most interesting bit in this line is the number enclosed in the square brackets: it is the firmware revision number.

As of v. 1.6 the BIOS will also display a single letter immediately following the „IDE+2.0" – it is the hardware revision of the interface: C (which also embraces revisions A and B), D, E or F.

The bottom line shows three shortcuts you may use to leave the config display:

1) **Esc** – this is „cancel+exit"; the changes you have made to the settings are discarded and the control is handed back to the OS (which continues its RESET routine afterwards).

The only exception are the date and time settings: any changes made here are immediately stored to the RTC circuitry, therefore are not discarded.

This is what you should select when you want to boot cassette or use anything else, which looks for the Start key state: just hold Start and hit Esc.

2) **^X** – that is, Control/X key combo. This is „save+exit". All settings are saved to the Non-volatile RAM and the control is handed back to the OS (which proceeds with the RESET routines as above).

3) **^B** – that is, Control/B key combo. This is „save+boot". As with Control/X, all settings are saved to the Non-volatile RAM, but instead of continuing to the OS RESET routine, a cold reboot is forced.

The rest of the screen presents the settings you can change to tune the interface's operation to your requirements. As you can see, certain characters are distinguished: you may change the settings pressing the keys corresponding to them on the Atari keyboard.

IMPORTANT: please never ever hit the Reset key while being in the Configuration Menu. This may corrupt the settings and you will have to setup all the options anew (this limitation is gone in hardware rev. F and later).

III. General remarks

The general rule is that no matter what you have done here, you will always be able to return to this display using the Start+Reset key combo (which is hardcoded in the IDE Plus ROM) and correct the settings, if something does not work as desired.

The interface contains several separate components, that is:

1) the ATA host adapter, which interfaces an IDE disk to your Atari
2) the SpartaDOS X ROM cartridge
3) a battery backed RTC (Real Time Clock)
4) high speed SIO routines in ROM
5) general PBI device management routines in ROM
6) a binary loader and an ATR loader, which share a common front-end

All these components may be configured independently. Therefore the configuration settings can be divided into six groups:

1) ATA (IDE) host adapter settings
2) General settings
3) Cartridge settings
4) Time settings
5) Default drive settings and
6) Other functions

IV. ATA (IDE) host adapter settings

`[I]`DE interface: on

Once the first one is switched to „off" and the configuration screen is left with Control/X or Control/B, the IDE disk stops being „visible" to the system.

`[S]`lave (IDE#1): off

When you use only one disk (i.e. the „master" device only), you may want to disable the ATA slave drive to prevent the firmware from attempts to communicate with it (which may for example add a 5 second delay to bootup time).

IMPORTANT: In BIOS-es prior to v. 0.8 slave drive access does not work, please either keep this option off, or upgrade.

V. General settings

[Q]`uick disable: off`

Quick disable allows to enable yet another „hot" key combo: Shift+Reset. Holding the Shift key and hitting Reset is equal to *IDE Interface: off* as described above. This is handy when you need to temporarily disable access to the hard disk and do not want to go to the configuration screen, nor want this setting be remembered by the Non-volatile memory. Pressing the Reset key alone enables the hard disk access again.

As of BIOS version 1.7 it is necessary to hit Shift/Reset (instead of just the Reset alone) to enable the disk access.

[P]`oweron delay: 0 s.`

Poweron delay adds the given number of seconds to the OS's initialization time at power-up time (not at cold reboot, however). It is handy when the spin-up time in your disk is a little bit longer than the IDE Plus ROM firmware expects by default, so that a time-out occurs while waiting for the disk to become ready (quite unlikely, by the way, as the firmware waits about 15 seconds).

It is also useful for the people who own the VBXE 1.2, which needs about 6 seconds to boot its core after power-up: 3 seconds added here usually prevent any possible trouble resulting from the Atari OS and DOS finishing initialization long before the VBXE is ready.

[R]`eboot hotkey: Help`

Reboot hotkey allows to choose the key combo which forces the machine to do a cold reboot. It is either Help+Reset (said to be a standard in PBI devices) and Select+Reset (which was used for this purpose in the older ATA interfaces: KMK/JŻ IDE and IDEa).

VI. Cartridge settings

```
SD[X] cartridge: off
```

The first one refers to the internal SpartaDOS X cartridge and is only effective, when the cartridge is physically enabled, i.e. when the switch is in its leftmost position (on an XE) or, to say this another way, when it is switched in opposite direction to the external cartridge port.

This is the only setting which is not held in the Non-volatile RAM. Storing it there would not make sense, because then, when set to *on*, it would cause the SpartaDOS X to be activated at every RESET, which in turn would make the command *COLD /N* ineffective, and that is not desired.

„SDX cartridge" state changes are only taken into account when the configuration screen is left with Control/B.

```
[E]xternal cart: on
```

External cart decides if the cartridge inserted in the external cartridge port is to be activated. It is only effective when „SDX cartridge" is set to off.

As of version 1.7 of the IDE Plus BIOS the binary loader respects this setting and starts with the external cartridge enabled, when this option was set to „on" while invoking the loader from the menu.

VII. Time settings

```
[Z]: CIO device: on
```

Z: CIO device: on installs a SpartaDOS 3.2-compatible „Z:" device in the OS's device handler table. This allows programs to access the on-board Real Time Clock in simple and standardized manner. The instructions on how to use that are to be found in *SpartaDOS Construction Set: R-Time 8 Owner's Manual and Supplement to SDCS Owner's Manual* by ICD (available on the net in PDF format). Most functions described there are implemented, except XIO 38 and XIO 39 (TD Line on and TD Line off).

```
[A]PE Time read: on
```

APE Time read: on enables the interface to emulate the APE Time protocol. It is useful, when a program which understands this protocol wants to access the

on-board Real Time Clock. If you prefer the time to be fetched from the real SIO (e.g. from the real APE program running on a PC), switch this off.

Note that SpartaDOS X as of version 4.44 does not really need this option enabled, because it has a dedicated driver (IDEPTIME.SYS), which accesses the time handler directly, bypassing the SIO.

```
Set [d]ate/[t]ime: Mon 20-Jun-11 18:23:03
```

Set date/time – this option is pretty much self-evident, it allows you to adjust the time (and date) which is being kept by the on-board Real Time Clock.

VIII. Default drive settings

```
[C]ONFIG.SYS: D0:
```

The *CONFIG.SYS* setting indicates the drive the SpartaDOS X will (attempt to) load the CONFIG.SYS file from, and which will become the default drive for the Command Processor. When SpartaDOS X is not used, this option has no effect. Otherwise, when SpartaDOS X *is* used, but the *CONFIG.SYS* is set to zero (as shown above), the *Boot drive* setting is used instead.

```
[B]oot drive: D0:
  D1: s[w]ap: off
```

Boot drive is the drive the system will boot from, except when SpartaDOS X is active (see above). When a zero is selected here, D1: is assumed. Separating these two options allows you to start e.g. MyDOS from one partition and SpartaDOS X from another one without a need to manipulate the settings every time you are changing DOSes.

The sub-option (introduced in BIOS v.1.3) changes the operation of the main option as follows: when it is off, the drive specified in „Boot drive" is visible with its proper drive number *and* as the D1:; when it is on, both drive numbers are swapped.

IX. SIO settings

```
[U]Speed SIO: off
   US on D[1]: off
   US on D[2]: off
```

```
US on D[3]: off
US on D[4]: off
```

These control the internal high-speed SIO routines. When *USpeed SIO* is set to *on*, the IDE Plus ROM enables the UltraSpeed protocol for serial devices (including disk drives) connected externally (to the regular SIO port). Up to 16 devices (including 15 disk drives) may be handled this way. In other words, you may now use high speed SIO transfers even without patching the ROM in your Atari.

As of BIOS version 1.0, when the UltraSpeed protocol is globally enabled (with *USpeed SIO: on*), you may selectively disable or enable it for each of the disk drives D1:, D2:, D3: and D4: by setting the respective sub-options to *on* or *off*. When such a sub-option is set to *off* for a drive, any requests made to that disk drive are passed on to be handled by the operating system. This is useful when a drive you have does not like UltraSpeed requests (such as LDW or Indus GT for example), or when one of the drive identifiers is occupied by a parallel floppy drive being a separate PBI device (such as the Karin Maxi floppy drive). In that latter case, setting the *US on Dx* to *off* for the specified drive allows the parallel floppy drive *x* to be properly handled, while the UltraSpeed protocol remains effectively enabled for all serial devices.

The *USpeed SIO* setting has a priority over the sub-options: when *USpeed SIO* is set to *off*, the sub-options have no effect.

With MyDOS all SIO requests will be handled according to these settings, either by the IDE Plus ROM („on") or the Atari OS ROM („off").

With SpartaDOS 3.2 this will only have influence on the calls made via the OS's SIOV entry.

With SpartaDOS X this depends on the DOS configuration, as different SIO drivers may be selected for individual drives. Refer to the SIO configuration sections in the SpartaDOS X User's Manual (especially SIO.SYS driver and these commands: MAP, SIOSET).

The IDE Plus's built-in high speed SIO routines can handle SIO transfers up to 126 kbps (HS Index 0). This has been practised with peripheral emulators (AspeQt and RespeQt on Windows 7, SIO2BSD on FreeBSD 8.2 and Ubuntu Linux) and with SIO2SD.

NOTICE: the IDE Plus's built-in high speed SIO routines do not work, if activated on the XL OS with an old version of Hias's patch applied for very high speed SIO transfers. The reason is that the patch changes the operation of a well-documented, RAM-based VBL vector (VVBLKI $0222/3, to be specific) in an incompatible way. Please use a newer version of the patch, where that problem was fixed.

X. Loading executables

`[L]`oad executable

Pressing [L] while in the main configuration screen, brings you to a loader program which is embedded in the IDE Plus 2.0 ROM BIOS as of version 0.8. A program of this kind is commonly called a „game loader", because people load games with it most of the time. But in fact, any standard executable file can be loaded here.

The loader supports internal drives (i.e. IDE Plus partitions) as well as serial drives connected via SIO. As of BIOS version 1.0 other PBI devices, attached to the PBI or ECI/Cartridge connector (such as Karin Maxi floppy drive for example), are also supported.

When the loader starts, it will change screen colour from green to blue, then attempt to load the main directory from the boot drive (the [B] setting above, chapter VIII).

At this point, instead of the directory listing, you may see a message like:

```
Unknown disk format!
Hit [1]-[9] or [A]-[O] to change disk
```

or simply:

```
Hit [1]-[9] or [A]-[O] to change disk
```

The loader supports Atari DOS, MyDOS and SpartaDOS file systems. The disk you selected is either physically unreadable or has a different format than the three mentioned ones. You are asked to change disk by pressing 1 or A for D1:, 2 or B for D2: and so on.

When the disk you selected has a known format, a directory listing should appear (unless the disk is empty). Directories are marked by the „|" character displayed to the left of the name, the remaining ones are files. One entry will be highlighted (displayed in inverse video). This is the cursor position, you may move it using the arrow keys (with or without the Control key depressed).

As of version 1.7 of the IDE Plus BIOS the directories are sorted before being displayed. They are sorted in ascending order by default: you can switch the sorting routine to descending order pressing the X key while in the loader.

The sorting routine is quite fast, but still, long directories may take few seconds to get sorted (you may expect sorting times such as 4-6 seconds to sort a 16 KB directory containing ~700 files on a stock Atari, the actual time depending on how badly unsorted the list was).[1] It may be advisable, then, sort such long directories permanently using SORTDIR.

As an extra feature, when the loader detects that a directory has already been sorted in the expected manner, the sorting procedure is skipped.

Use the [Return] key to enter a subdirectory the cursor is positioned on. When you hit [Return] while the cursor is standing on a file, and the file is executable, it will be loaded. While loading you may see a message like „Memlo: $0936" printed at the bottom of the screen. This informs you how much RAM the loader occupies: in this case from $0700 to $0936, which makes 566 bytes (including 512-byte sector buffer).

Many games (and many game cracks) do really weird things, therefore you will always be able to find a game, which does not work with this loader. But anyway, this loader runs more games than its disk-based version, mainly because of much lower RAM footprint. If a game refuses to load, it may be a good idea to cold reboot your Atari and try again.

A short help screen will appear when you press the [?] key while in the file list. Any key pressed there brings you back to the directory display.

Note that the Atari BASIC is disabled by default; when Control+B is hit while in the file list („toggle BASIC on/off„), a [B] appears in the top-left

---

[1]   The routine which is sorting directories in the loader underwent a performance test designed like this: 1) load to RAM a directory containing 735 entries (16.5 KB); 2) shuffle the data randomly; 3) start timer; 4) sort the data in ascending order; 5) stop timer; 6) steps 2-5 repeat 256 times. On a stock PAL Atari this procedure has returned 5.26 s. (min. 4.94, max. 5.74) as the average sorting time. Accelerated machines will take advantage of the faster CPU: at 20 MHz the same test shows average of 0.4 s.

corner of the screen – that is a flag indicating that the Atari BASIC ROM will be enabled during loading. It is useful when loading (rare) programs which require it. As of version 1.7 the loader will refuse to enable BASIC when another ROM already is occupying the same place (e.g. SpartaDOS X or an external cartridge).

XI. Loading ATR files

As of version 0.9 the IDE Plus ROM BIOS contains an ATR disk image loader. It is integrated with the binary loader's front-end described above, so the operation is quite similar: you just find an ATR file on the disk, highlight it, and press Return to execute.

After you hit the Return key, your Atari will boot off the selected ATR file just as if it was a physical disk. Therefore **do not forget to hold the Option key**, *if the program being booted requires that there are no cartridges (including Atari BASIC) in the memory. Note that the Control+B of the binary loader („toggle BASIC on/off") option is effective when loading regular binaries* **only!**

While the program is loading and running, your Atari „sees" the ATR file as a drive – the file is effectively „mounted" onto D1: at the SIO level. To unmount it, enter the main configuration screen (Start+Reset) and select the *Save+boot* function (Control+B). Alternatively, entering the binary loader unmounts any mounted ATR as well. Also powering your Atari down unmounts the file.

IMPORTANT: Starting an ATR file from the loader level locks out all real HDD partitions for safety. Unmounting it using the method described above restores access to the partitions. If the program started from the ATR does not force cold restart on every reset, hitting Reset while holding down the Start key will bring you to the main IDE Plus menu. There, enabling SpartaDOS X and hitting Control/X will bring you to the SpartaDOS prompt with the ATR file mounted as D1: and all the other disks being locked out.

The ATR loader supports APE extensions to the SIO2PC standard, specifically its write-protection flag. Even when this flag is cleared, you can still write-protect the image by setting the +P flag for the corresponding ATR file. Do that under SpartaDOS X. **Note that writing to a mounted and unprotected ATR file does not reset its „archived" bit!**

All ATR densities are supported: SD, ED, DD, „bad" DD, DD512. The ATR file's size is limited by the maximum size of a file on a SpartaDOS disk, which is 16 MB.

There are some limitations in comparison with the binary loader:

1) an ATR file can only be run from the hard disk: you can not load an ATR file from a serial drive (even if the binary loader generally allows to load regular binaries from a serial drive) or another parallel device.

2) an ATR file can only be loaded from a SpartaDOS-formatted partition: the MyDOS file system does not allow to access a file randomly, and that is the key feature here which is required.

3) prior to BIOS ver. 1.3 there was no possibility to „flip the disk", so certain programs, which occupy more than one ATR file, had still to be run the traditional way (i.e. from SIO).

As of BIOS ver. 1.3 it is possible to mount multiple ATR files to consecutive DOS drive numbers using the dedicated software (ATRM.COM and ATRU.COM) while at the SpartaDOS X prompt. Next you have to do a cold boot – the computer will then boot off the ATR file you have mounted as D1:. Pressing the IRQ button (available as of rev. D of the hardware) will then rotate the mounted images towards lower disk IDs, i.e. D2: then becomes D1:, D3: becomes D2: and so on, and D1: becomes whatever disk ID was last occupied by an ATR file. This mechanism allows to „flip disks" when necessary.

4) some programs use own serial I/O routines instead of calling the OS to perform the I/O – such programs will not work.

5) the I/O is much slower than with a real partition: you may expect ~25 KB/s reads and ~7 KB/s writes.

ATR files may also be created, mounted, accessed, unmounted from the SpartaDOS X level using the software bundled with the BIOS in the update archive. In this case, an ATR file may be mounted in addition to all the other HDD partitions, not blocking the access to any of them.

XII. Mounting multiple ATR files

As said above, as of BIOS ver. 1.3 it is possible to mount multiple ATR files onto drive letters of choice. However, it was only possible to mount multiple

ATR files from the SpartaDOS X prompt: the internal mounting mechanism lacked support in the binary loader front-end.

This support has been added as of BIOS ver. 1.7. To mount an ATR file from the binary loader:

1) highlight an ATR file that resides on a SpartaDOS partition,

2) hold Shift,

3) press a key 1-9 or A-O to select the drive number or the drive letter which is to be the mount point.

Note that on the Atari keyboard the key combination Shift+number will (for consecutive numbers 0, 1, 2, 3... etc. up to 9) produce the following characters:), !, ", #, $, %, &, ', @, (. If the layout of the keyboard you are using is different, the result of pressing Shift+number as described above may be unexpected.

The loader will visibly acknowledge that the selected file was mounted on the specified drive. When there is a mistake, Shift/0 unmounts everything and allows you to start over.

Example: say you have a program which occupies two ATR files, named SIDEA.ATR and SIDEB.ATR. To execute this program from your hard drive, please follow these steps:

1) If these files are on your PC, copy them first to one of the partitions of your HDD. Say that it is D4. Before doing that, make sure that the D4 is a partition which had been formatted with SpartaDOS file system (**not** Atari DOS file stystem). Use the copying procedure as for any other file you want to copy from the PC to your Atari.
2) Enter IDE+ menu.
3) Make sure the **[S]**DX option is set to *off*.
4) Enter **[L]**oader.
5) Press 4 to get the directory of D4 – you should see your two ATRs there.
6) Use cursor keys to highlight SIDEB.ATR.
7) Say you want it mounted at D5 – so press Shift/5 (or E). The loader should now visibly acknowledge that the file has been mounted on D5.
8) Now highlight SIDEA.ATR.

9) Press Return (holding down Option, if the program needs BASIC disabled).

10) Now the computer will boot off SIDEA.ATR.

After that, whenever you want to swap „sides", press the IRQ button on your IDE+ interface.

Note that the executed program will „see" only the ATRs, the „real" partitions of your HDD will not be visible to it for safety.

But when you mount several ATR files the way specified above including the last one, then instead of pressing Return you press Select/Reset to force a cold reboot, the system will boot off the drive D1: with all the ATRs you have selected mounted, but also with all the HDD partitions, which have not been masked out by these ATRs, visible in the system.