AspeQt-2k24 with 8Bit Cartridge

User Manual

Introduction | System Requirements | Installation | Configuration | Usage

Introduction

AspeQt is a cross-platform, free and open source Atari 8-bit serial peripheral emulator.

AspeQt emulates various Atari 8-bit peripherals like disk drives and printers via a SIO2PC cable. If you are familiar with software like SIO2PC, APE, Atari810, AtariSIO, AspeQt, etc., you probably won't have any problems getting used to AspeQt.

Even though AspeQt is not fully mature yet, it is easy to use and, despite its shortcomings, has many features that you may find useful, the highlights being:

- ❐ Cross-platform GUI with drag and drop support
- ❐ Atari Cartridge images to control disk and printer functions
- ❐ 15 emulated disk drives with support for 512 bytes per sector disk images
- ❐ High speed operation up to 6 times faster than the normal speed *(With compatible Atari OS / DOS)*
- ❐ Ability to use PC folders as emulated disks
- ❐ Disk image explorer for viewing and modifying disk image contents
- ❐ Ability to run Atari executable directly without using disc images and Doses
- ❐ Printer emulation with preview and ATASCII support
- ❐ Cassette image playback with custom baud rate
- ❐ ability to save and restore AspeQt sessions with its UI settings.
- ❐ Normal and mini UI modes

Up

System Requirements

AspeQt currently runs under Windows, Linux and Mac OS X. Other Unix-like operating systems shouldn't require too much effort. The absolute minimums aren't easy to tell but a modern Linux distribution with Qt 5 libraries or a Windows XP (or newer) is recommended.

Although AspeQt is reported to run on a Pentium 233 MMX, a modern processor and a decent amount of RAM is strongly recommended for smooth operation. It runs fine on 64-bit CPUs.

In order to do anything useful with AspeQt, you will need an Atari 8-bit computer and a SIO2PC cable. SIO2PC cable is an interface that connects the Atari's serial bus to the PC's serial port. Since many newer PCs lack a real serial port, you may need a UART card, or a high quality Serial-to-USB adapter *(FTDI chip based devices were tested successfully, others may not work as intended)* or a SIO2PC USB interface *(instead of serial)*, If you opt for the latter however, please note that the "AtariMax SIO2PC Universal Interface USB Edition" is not officially supported as it uses proprietary USB device drivers.

- **"Do It Yourself" SIO2PC build instructions - See: SIO2PC_Build_Instructions.pdf**

You don't need any real Atari disk drives or printers to use AspeQt.

Up

Configuration

In order to be able to use AspeQt, you need to configure it to suit your hardware. When it's running for the first time, a dialog will pop up asking you if you wish to open the configuration dialog. Click "Yes" to open the Options dialog. You can also access this dialog from the Tools/Options menu item.

On Linux, you have to choose a serial I/O backend first. If you have installed the AtanSIO package and you have a RS-232 port, the recommended way is to use the AtariSIO backend. If you have a USB adapter or you don't want to use AtariSIO for any other reason, choose the standard serial port backend. This is also the only available backend on Windows.

Configuring the standard serial port backend

First, enter a port name. On Windows, it should be something like COM1, COM2 etc. Check the device manager to see which one of them is installed. On Linux, it should be /dev/ttyS0, /dev/ttyS1 etc. for RS-232 ports or /dev/ttyUSB0, /dev/ttyUSB1 etc. for USB adapters.

Next, you will have to select a handshaking method. The handshaking method tells AspeQt which one of the RS-232 pins is connected to Atari's command line. There are 3 supported methods: RI, DSR, CTS and NONE. Check your SIO2PC interface's documentation to learn which one of them is used for your cable. Note that NONE *(no handshaking)* is highly experimental and not recommanded for day-to-day use, in this mode AspeQt will ignore Atari's command line (SIO7) and will monitor the COM port's RXD (Atari SIO5) line instead. No handshaking mode was included to experiment with wireless (bluetooth) serial interfaces.

The “High speed mode baud rate” combo box selects the transfer speed to be used for the high speed mode. Some OS /DOS for the Atari supports higher transfer speeds than the standard 19200 bits per second. Not all DOS /OS are capable of using 3x speed and some treat 2x as a special case for XF551 drives, so you may need some experimentation to find the best setting. 1x option is provided for unreliable connections, like when using a cheap USB adapter or when running AspeQt under virtualization software and/or on a slow CPU computer.

If your serial port supports arbitrary bit rates, you may check the “Use non-standard” speeds check box and select a POKEY divisor to be used in high speed mode, the lower the divisor, the higher the speed. Usually real COM ports and some VCP *(Virtual COM Ports)* don't support arbitrary baud rates, FTDI chip based serial-to-USB cables do support arbitrary baud rates, and are thus the recommended type of cable for use with AspeQt.

The exact formula for the nominal speed is:

baudRate = clock/ (2*(divisor+7))

Where dock is ~1,773,447 for PAL, and ~1,789,772 for NTSC, however, it’s not always possible to reach nominal speeds. So, 1x is divisor 40, 2x is divisor 16 and 3x is divisor 8. Divisor 0 is ~126kbits/secondn and that means approximately 6x.

Please note that very few OS/DOS will be able to function with such high speeds. Currently the only tested software that can reach divisor 0 with AspeQt is the hisio OS patch. As a final warning, some Atari 8-bit computers have capacitors connected to their SIO lines that can interfere with high speed transfers. In short, speeds beyond 3x are not guaranteed to work in every case.

The Complete/Error delay field selects the delay which is inserted after the Acknowledge response to a command by AspeQt, before AspeQt then sends the Complete response, or an Error response (whichever is appropriate). If you experience SIO errors or retries, increasing the value here should make the transfers more relible. Decreasing the value may **slightly** increase performance, but at the risk that you cause errors. There has to be a minimum delay from the last bit of the Ack to the first bit of Complete or Error, or the Atari might not recognize it. In most cases, this value does not need to be changed. The default values of 800µs and 300µs for OSX/linux/other and Windows, respectively, should be reasonable **if you have an FTDI based SIO2PC-USB**. If you have an RS232 SIO2PC connected to a real serial port, you can probably use a value closer to 300µs on linux/OSX/other than the default 800µs. See issue #2 on github for more information about why this discrepancy between the default values exists.

Configuring AtariSIO backend

Note: AtariSIO is available under Linux only!

After installing and running the AtariSIO module and making sure that you have the required permissions, you just need to enter a device name which should be /dev/atarisio0 under normal circumstances and select a handshaking method as described above for the standard serial port backend. Please refer to AtariSIO documentation on how to build and install AtariSIO.

Up

Usage

After making sure that you installed and configured AspeQt correctly, you can start using it with your Atari. If you used similar software before, it should be fairly straight forward.

AspeQt 2k24 - 8bit Cartridge

The AspeQt 8bit Cartridge can be run from most Atari cart emulators (UnoCart, Ultimate cart, etc.)

All AspeQt disk and most printer functions can be done using the cartridge.

Two versions of the cart are available:

MenuCart_BOOT_2k24.car - Boot D1: on power up
MenuCart_NoneBOOT_2k24.car - Nodisk boot on power up

Cartridge directory path set in:

Tools -> Options -> Emulation -> "RCL Client local path"

- used to list atr/atx/pro/exe files from the cartridge - Option "B - List Host Images"

Disk images

Instead of using real disks, AspeQt uses disk images. These are regular files that contain an image of an Atari disk. There are several formats, the most common being the .atr format, currently AspeQt only supports .atr, .xfd, atx *(still in development),* and .pro formats. Future releases may provide support for .scp, .dcm, .di and gzipped (.atz. atr.gz, .xfz and .xfd.gz) formats.

AspeQt emulates 15 disk drives. You have one slot for each of them, labeled D1 through DO *(letter O)*. D9 through DO are only supported by SpartaDOS X and compatible DOS *(see SDX manual for details on supported disk drive identifiers)*.

You can make AspeQt to show only the first slot, the first 8 slots, or all 15 slots using options in DISK and WINDOW menu items.

+ To hide/show drives D9 through DO, use menu item Disk/Hide drives D9-DO *(CTRL+H)*,

+ To toggle AspeQt mini mode use menu item Window/Toggle Mini Mode *(CTRL+M)*,in mini mode AspeQt shows only the first drive slot,

+ To toggle shade mode in mini mode use menu item Window/Toggle shade mode *(CTRL+S)*, in shade mode AspeQt shows a semi-transparent window which does not completely block the view of underlying UI objects,

Note that all drives are actually available for use with your Atari in all AspeQt display modes whether they are visible or not.

You can mount a disk image to an empty drive slot by

- ❐ Dragging and dropping a file to the desired slot
- ❐ Using the “Disk/Mount disk image” menu item *(the first available slot will be used)*
- ❐ Using the “Mount disk image” tool button available in each slot
- ❐ Right dicking on a slot and selecting the “Mount disk image” menu item
- ❐ Selecting a recently used disk image from the “Disk” menu
- ❐ By pressing Alt + 1 through Alt + O *(letter O)*
- ❐ Using RCL remote module on the Atari computer *(See RCL section of the this user manual for more details)*

You can see the result of your operation in the log display which is below the disk slots, if the operation is completed successfully, your Atari should be able to see the mounted disk image just like a real disk in a real drive.

You can use the tool buttons and context menu items to perform other operations like saving the disk image, enabling write protection for the image, reverting the image to its last saved state, ejecting (unmounting) the image, creating a new image and so on. You can also swap images using drag and drop.

Folder Images

Folder images provide a similar function to the features variously named as “PC mirror’, “Simulated disk”, “Share point” etc. by other peripheral emulators. This is basically a simulated Atari DOS2.5 disk. Instead of a disk image, you can mount a folder in your PC that contains some Atari files and Atari will see it as a disk with the same files in it.

Currently, the mounted folder will be seen as a standard DOS 2.5 disk and it’s *sequential access (Basic NOTE/POINT commands will not work as expected) and read-only* . Although Folder Images are simulated and behave like Atari DOS 2.5, there are some differences. The most important difference is that each file can be as large as 8MB in size and file sizes are shown as number of bytes as opposed to number of sectors (unlike Atari DOS). Maximum number of files in the directory is 64 and subdirectories are not supported. Folder Images are also compatible with MyDOS, all versions of Spartados and SpartaDOS X. There may be issues with others.

Folder images are also bootable as of version 0.8.5, but since the folder image is a simulation of a standard DOS 2.5 disk, you can only boot into a DOS that is compatible with AtariDOS disk structure. There are two exceptions to this, first one is SpartaDOS, AspeQt will allow you to boot into a 3.2G version of SpartaDOS but there are some limitations. AspeQt will also allow you to boot into the standard version of MyPicoDOS 4.05. See the following section for for details and limitations:

Folder boot details and limitations:

To boot your Atari with a Folder Image, first mount a PC folder to disk slot 1 (D1:). Once mounted, right-click on the Folder name and select *Folder Boot Options* from the menu, select the DOS you would like the boot and click *Apply*. make sure AspeQt is ready to receive commands from your Atari and finally turn your Atari ON, selected DOS will be booted. You must select a DOS boot option if you mounted a Folder Image for the first time and you want to boot your Atari from that folder whenever the folder is mounted. You normally only select the DOS option once for each Folder Image. The Folder will remain bootable with the same DOS between AspeQt sessions.

Warning: *AspeQt will copy the necessary DOS files into the mounted PC Folder to make it bootable. Do not keep any other DOS files (like dos.sys, dup.sys etc..) on that folder as they will be erased/replaced*.

Note that you don't need to supply any DOS files, AspeQt distribution supplies the files necessary to boot your computer. These DOS files are copyright of their respective owners, 13th Leaderand AspeQt distributes those files with the understanding that they are either freeware, abandonware or public domain and are widely available for download through the internet. If you are the copyright holder of one or more of these files, and believe that distribution of these files constitutes a breach of your rights please contact The 13th Leader. We respect the rights of copyright holders and won't distribute copyrighted work without the rights holder's consent.

The following DOSes can be booted from a Folder Image:

- **Atari DOS 2.5** AspeQt Folder images are fully compatible with this DOS, you can boot the DOS and load drivers and binary files *(autorun.sys)* during the boot process. AspeQt supplies the dos files *(dos.sys, dup.sys)* and the ramdisk driver *(ramdisk.com)*. You can add your own *autorun.sys* file by copying the file into the folder ***$bootata*** which is located in AspeQt application directory. You can also use DOS 2.0 instead of 2.5 by simply replacing the *dos.sys* and *dup.sys* in $bootata folder with the ones from a DOS 2.0 disk.

- **MyDOS 4.5** This DOS is disk structure compatible with AtariDOS, so everything that's said for AtariDOS 2.5 above is also valid for MyDOS. Boot files folder for MyDOS is ***$bootmyd***
- **MyPicoDOS 4.05** This game DOS is provided to support Folder Images which hold Atari games. You can quickly boot and start your games conveniently from a PC Folder. Only standard version of MyPicoDOS is provided and supported. When selecting MyPicoDos as your boot DOS there is an extra option on the menu to *disable high speed SIO* code built into this DOS. If checked, this option will instruct MyPicoDOS to run in normal speed. This may be necessary under certain configurations. One example is if you are using an Ultimate 1MB board or a PBI device like IDE Plus 2 on your Atari with high-speed OS enabled. MyPicoDOS used in high-speed mode will conflict with the high-speed OS on the Atari, so checking this option and booting and running MyPicoDOS in normal speed will solve this problem.

  AspeQt will also automatically generate a *piconame.txt* file during the Folder Image boot process. So if you have game files with long file names they will be displayed with their full names when MyPicoDOS menu appears. *piconame.txt* file will only be created/updated when you boot, so if you add more game files to your PC folder during your MyPicoDOS session they will not show with long names until you reboot.

  Boot files folder for MyPicoDOS is ***$bootpic***. it is not recommended to modify this folder, unlike AtariDOS and MyDOS there are no customizations you can make for MyPicoDOS.
- **SpartaDOS 3.2G** SpartaDOS is not compatible with AtariDOS. Its disk/file structure and boot scheme is completely different. So this DOS is only partly *(and I would say rather crudely)* supported. You will be able to boot SpartaDOS version 3.2G from a Folder Image with the following limitations:
  - To boot SpartaDOS from an AtariDOS structured Folder Image is an impossible task. So AspeQt uses some *(not so elegant)* tricks to coerce SpartaDOS into booting from an AtariDOS compatible Folder Image by giving it the impression that it's booting from a SpartaDOS formatted disk. Once the boot is completed AspeQt forces SpartaDOS

to re-detect the Folder Image as an AtariDOS formatted disk so that it can display and manipulate the files within the folder. The mechanics of this scheme necessarily limits the boot proccess in the following ways:

- ❐ You can not load drivers or run an *autorun.bat* file during the boot process
- ❐ Once booted from, a Folder Image won't be bootable again until you right-click on the Folder Image name, select *Folder Boot Options*, select SpartaDOS 3.2G and click *Apply*. This will reset the Folder Image boot files and will make the folder bootable once again.
- ❐ You can not modify/delete files in the boot files folder ***$bootspa***

- ❐ The restrictions and limitations may be lifted in the upcoming versions of AspeQt.

Running Atari executable

Most Atari programs floating around on the internet come as Atari DOS executables. These files may have .exe, .com, .xex or any other extension. Instead of messing with disk image software and DOS, you may directly run these files in your Atari using AspeQt.

You can either drag and drop a file with .exe, com or .xex extension into any slot or you can use the menu item “File/Boot Atari executable” to access this feature. A dialog with the necessary instructions will pop up and the file will be loaded and run. You can leave the dialog open to boot from the same executable more than once. A reload button is provided which will reload the executable into memory. This button is intended for atari software developers who may be developing on the PC and testing their software after making changes to it. Reload button will ensure the most recent executable is loaded from the PC, and as such is mainly a convenience feature for such developers.

The executable booter has an optional high speed code which will allow you to load programs a lot faster. You can enable/disable it with the “Tools/Options/Emulation/Use high speed executable loader” check box. The high speed code is not able to cope with higher speeds than divisor 3 so check your configuration before attempting to load a file in this way.

Please note that this feature is not compatible with every executable and, in practice, it’s not even possible to implement such a loader. High speed loader has even more issues. So there will always be some programs that you won’t be able to run with the executable booter but hopefully the number of the compatibility problems will decrease with future versions of AspeQt.

Cassette images

AspeQt can playback cassette images in .cas format. These are PC files that contain data extracted from an Atari cassette. You can either drag and drop a file with .cas extension into any slot or you can use the menu item "File/Playback cassette image" to access this feature. A dialog with the necessary instructions will pop up and the file will be played back.

The cassette emulator can be configured to ignore the baud rate that is embedded in the image file in favor of a custom one. This may speed up the load times but can cause compatibility problems. You can enable/disable it with the "Tools/Options/Emulation/Use custom baud rate for cassette emulation" check box. When enabled, you can use the spin box below to set the custom baud rate. The available values range from 425 through 875 bps. These values are the lowest and highest speeds that the Atari OS can process. The normal speed is 600 bps.

The cassette emulation does not support rewinding or seeking in the images. This may change in the future versions.

Printer output

AspeQt emulates a generic text-only Atari printer. It only emulates the first printer device, that is "P1:" You can view, save or print the emulated printer output using the "File/View printer text output". Support for ASCII and ATASCII is provided.

Running multiple instances of AspeQt (Sessions)

AspeQt allows you to save and load your disk sessions, that is, the order and names of the mounted images and their settings. You can access this feature from the "File/Open session" and "File/Save session" menu items. As of version 0.8.2 AspeQt fully implements multi session capability. This means you can now launch multiple instances of AspeQt using different session files and have different configurations for each session. This makes possible serving more than one Atari computer from one PC so long as you have more than one COM port and SIO2PC cable available. Simply create different sessions with different settings and save them to a permanent session file (a file with .respeqt file extension).

If you plan on serving more than one Atari computer simultaneously, make sure that the PC is fast enough to handle similtaneous SIO requests as Atari SIO is very time critical and can fail if the requests are not serviced in a timely fashion. Experiment with the multi-session capability and verify that it can be run reliably before you put it on serious use.

The following parameters can be configured for individual sessions *(stored in each session file)*

- Backend
- Atari SIO driver name
- Atari SIO handshaking method
- Serial port name
- Serial port handshaking method
- Serial port speed
- Serial port Pokey divisor
- Use of High speed exe loader
- Custom cassette baud rate
- Main window screen geometry
- Printer window screen geometry
- Preferred Language *(if you need to run different language sessions)*
- Mounted disk image file list
- Other session related parameters *(like showing/hiding certain windows etc...)*

The following configuration parameters are global and apply to all sessions *(stored in Windows registry)*

- First Time flag *(indicates it's the first time ever AspeQt was launched on that computer)*
- Last Disk image directory
- Last Folder image directory
- Last Cassette image directory
- Last Executable file directory
- Last Extract directory
- Last Saved Printer file directory
- Last Session file directory
- Minimize to tray option
- Recently mounted disk image file list

The following configuration parameters apply when AspeQt is launched without a session file *(stored in Windows registry)*

- Mounted disk image file list

To launch a session, create a shortcut *(Windows)* or a link *(Unix/Linux)* in a folder with the session file name as a command line argument:

An example of a shortcut for Windows would be:

"C:\Program Files\AspeQt\AspeQt.exe" C:\Program Files\AspeQt\session.AspeQt

Apetime

AspeQt supports ApeTime *(Date/Time downloader utility)* from the AtariMax APE package. The support code has been added back in from older versions of AspeQt.

Up