A8PicoCart

Instruction Manual

Revision 1 16/8/2023

Atari XL/XE multi-cart

Quick Start

The A8PicoCart is a cartridge emulator and XEX file launcher for Atari XL & XE machines, supporting cartridge images and files up to 128k in size. It is a port of the UnoCart design to the Raspberry Pi Pico hardware.

When plugged into a PC using a USB C cable, it will appear as a USB Mass Storage device allowing you to easily copy new Atari software to the cartridge. It has 15megs of flash storage available for Atari files.

ALWAYS REMOVE THE CARTRIDGE FROM YOUR ATARI BEFORE PLUGGING IN A USB CABLE

Copy XEX, CAR and ROM files to the cartridge from your PC. You can create folders to organize the files. Eject the USB storage device from your PC before disconnecting the cable.

Insert the cartridge in the Atari 8-bit. If the board is uncased, look for the text on the board indicating the correct orientation – the reset button should be facing forwards (Atari XL) or upwards (Atari XE). Once you've powered up the Atari, use the cursor up and down keys to select an item and press Return to start it.

Menu

The menu allows you to navigate the files on the built-in flash memory and select a title to use. Use the cursor up, down and [Return] keys to select items. Press [B] to go back to the parent directory. Press [ESC] to search for a file. Press [X] to disable the cartridge and boot normally (e.g. from a SIO device) – this allows you to leave the cartridge plugged in.

Reset Button

Press the reset button to reset the cartridge. This will generally cause the Atari to crash, and the reset button on the Atari can then be pressed to return to the cartridge menu.

Reset Filesystem

In the event the on-board filesystem becomes corrupted, you can reset it by holding the fire button on the first Atari joystick when you power on the Atari. Press Y in response to the prompt offering to reset the flash memory. The device will now be in a "factory" state when next used as a USB mass storage device.

CAR Files & ROM Files

CAR files and ROM files both contain the data for a cartridge. However CAR files have an additional 16 bytes header containing the cartridge type making it clearly identifiable to the A8PicoCart. In contrast, the cartridge type of a ROM file is ambiguous – e.g. an 8k ROM file could be a standard 8K cartridge, or an 8k OSS cartridge.

ROM Files

Plain ROM files are supported, but since these files are raw cartridge dumps, the A8PicoCart will decide what type of cartridge they come from based on the file size only. The table below shows what types will be used.

ROM File Size	Cartridge Type
8k	Standard Atari 8k
16k	Standard Atari 16k
32k	Atari XEGS 32k
64k	Atari XEGS 64k
128k	Atari XEGS 128k

Supported Cartridge Types (CAR files)

Description	CAR Type(s)
Standard Atari 8k	1
Standard Atari 16k	2
Atari XEGS 32-128k	12-14
AtariMax 1mbit (128k)	41
Switchable XEGS 32-128k	33-35
Megacart 16-128k	26-29
Bounty Bob	18
Williams 32k, 64k	8, 22
OSS 8k	44
OSS 16k (034M)	3
OSS 16k (043M)	45
OSS 16k (type B)	15
SIC!Cart (128k only)	54
SDX 64k	11
Diamond 64k	10
Express 64k	9
SDX 128k	43
Blizzard 16k	40
4k	58
Turbosoft 64k, 128k	50, 51
ATRAX 128k	17

Converting to CAR format

Utilities are available to convert ROM/BIN files into the CAR file format. The ROM2CAR utility available on the Ultimate cart website provides an easy drag and drop interface: https://github.com/robinhedwards/UltimateCart/

Problem Files

You may come across files with a .BIN extension. These are often just .ROM files and can be renamed. You may also find ROM files where the A8PicoCart menu gives the error "Unsupported ROM size". This happens when the ROM file is not an exact multiple of 1,024 bytes. In this case, the ROM dump may have already been converted to a CAR file (with a 16 byte header) but has been wrongly named and you can simply rename the file from .ROM to .CAR.

ATR Files (64k and above XL/XE only)

The cartridge also has simple support for reading and writing to ATR (floppy disk image) files. It supports both SD (128 byte sector) and DD (256 byte sector) images. DD images must be standard ATR files, with 128 bytes for the first 3 sectors and no padding. The device supports ATR files up to 15 megabytes in size.

When an ATR file is selected from the menu, a modified version of the Atari operating system (OS), based on the Altirra open-source LLE kernel, is installed as a "soft" OS in the 16k RAM under the OS – this allows the Atari's SIO requests to disk drive 1 (D1:) to be re-directed to the SD card in the cartridge.

The main use case for this feature is to allow you to boot a DOS formatted disk (e.g. DOS 2.5) and load and save BASIC programs to the disk. Note that the A8PicoCart cannot emulate a disk and cartridge at the same time.

Accesses to other disk drives (D2: upwards) are passed through to the SIO port as normal. This allows files to be copied between an ATR mounted on the cartridge (D1:) and other SIO devices (D2:+).

Hold down OPTION when you select the ATR file to disable BASIC. This is necessary for many games. MyPicoDOS ATRs will run, but SELECT should be pressed to disable high-speed SIO, and use the operating system SIO routines instead.

Important note – many ATRs will run correctly from the A8PicoCart. However, any programs that attempt to use the RAM under OS will not work, nor will any software that does not use the standard OS SIO calls.

The OS ROM is stored in the cartridge firmware, but an alternative UNO_OS.ROM file (16k) will be loaded and installed from the root directory of the SD card instead (if such a file is present).

Firmware re-programming (advanced)

To program a new version of the firmware to the cart, first unscrew the cartridge shell (if present) to reveal the PCB inside. There should be a single button on the purple Pico board, marked BOOTSEL. Press and hold this button as you connect the cartridge to your PC with your USB cable. The cartridge will mount as a mass storage device, but instead of showing up as a A8-PICOCART, it should instead show as a device labelled RPI-RP2.

You can reprogram the Raspberry Pi by dragging a UF2 file into the explorer window. When the new firware has been programmed the device will dissappear, and remount as an A8-PICOCART with all files still present.

The latest firmware is available on the project website:
https://github.com/robinhedwards/A8PicoCart/firmware

Microcontroller (advanced)

The A8PicoCart uses a clone of the Raspberry Pi Pico that differs from the official board by having all the RP2040 GPIO pins broken out to headers on the board. The official board has slightly too few GPIO pins available to fully connect to the Atari's cartridge port. If you want to develop your own firmware for the device, the following table details how the cartridge port is connected to the RP2040 microcontrollers GPIO pins.

RP2040 GPIO	**Cartridge Port**	**Function**
0-12	A0…A12	Input - Address bus
13-20	D0…D7	Input/Output - Data bus
21	/CCTL	Input - Cartridge control area access
22	PHI2	Input - 6502 Clock
23	R/W	Input - 6502 Read/Write
24	/S4	Input - Cartridge select
25	/S5	Input - Cartridge select
26 (A0)	RD4	Output – Cartridge present signal
27 (A1)	RD5	Output – Cartridge present signal
28 (A2), 29 (A3)		NC
RUN		Reset Button pulls low to reset RP2040

Credits

Original idea, hardware and firmware by Robin Edwards (electrotrains at atariage). Altirra LLE OS used with permission from Avery Lee (phaeron at atariage). XEX loader, OS modifications and many helpful discussions thanks to Jonathan Halliday (flashjazzcat at atariage).