This is an emulator for the Commander X16 computer system. It only depends on SDL2 and should compile on all modern operating systems.

Features

- CPU: Full 65C02 instruction set
- VERA
  - Mostly cycle exact emulation
  - Supports almost all features:
    - composer
    - two layers
    - sprites
    - VSYNC, raster, sprite IRQ
- Sound
  - PCM
  - PSG
  - YM2151
- Real-Time-Clock
- NVRAM
- System Management Controller
- SD card: reading and writing (image file)
- VIA
  - ROM/RAM banking
  - keyboard
  - mouse
  - gamepads

Binaries & Compiling

Binary releases for macOS, Windows and Linux are available on the releases page.

The emulator itself is dependent only on SDL2. However, to run the emulated system you will also need a compatible `rom.bin` ROM image. This will be loaded from the directory containing the emulator binary, or you can use the `-rom .../path/to/rom.bin` option.

> ***WARNING:*** *Older versions of the ROM might not work in newer versions of the emulator, and vice versa.*

You can build a ROM image yourself using the build instructions in the [x16-rom] repo. The `rom.bin` included in the *latest* release of the emulator may also work with the HEAD of this repo, but this is not guaranteed.

macOS Build

Install SDL2 using `brew install sdl2`.

Linux Build

The SDL2 development package is available as a distribution package with most major versions of Linux:

- Red Hat: `yum install SDL2-devel`
- Debian: `apt-get install libsdl2-dev`

Type `make` to build the source. The output will be `x16emu` in the current directory. Remember you will also need a `rom.bin` as described above.

WebAssembly Build

Steps for compiling WebAssembly/HTML5 can be found here.

Windows Build

Currently macOS/Linux/MSYS2 is needed to build for Windows. Install mingw-w64 toolchain and mingw32 version of SDL. Type the following command to build the source:

```
CROSS_COMPILE_WINDOWS=1 MINGW32=/usr/x86_64-w64-mingw32 WIN_SDL2=/usr/x86_64-w64-
mingw32 make
```

Paths to those libraries can be changed to your installation directory if they aren't located there.

The output will be `x16emu.exe` in the current directory. Remember you will also need a `rom.bin` as described above and `SDL2.dll` in SDL2's binary folder.

Starting

You can start `x16emu` / `x16emu.exe` either by double-clicking it, or from the command line. The latter allows you to specify additional arguments.

- When starting `x16emu` without arguments, it will pick up the system ROM (`rom.bin`) from the executable's directory.
- The system ROM filename/path can be overridden with the `-rom` command line argument.

- `-prg` lets you specify a `.prg` file that gets loaded after start. It is fetched from the host filesystem, even if an SD card is attached!
- `-bas` lets you specify a BASIC program in ASCII format that automatically typed in (and tokenized).
- `-run` executes the application specified through `-prg` or `-bas` using `RUN`.
- `-scale` scales video output to an integer multiple of 640x480
- `-rtc` causes the real-time-clock set to the system's time and date.
- `-echo [{iso|raw}]` causes all KERNAL/BASIC output to be printed to the host's terminal. Enable this and use the BASIC command "LIST" to convert a BASIC program to ASCII (detokenize).
- `-rom <rom.bin>` Override KERNAL/BASIC/* ROM file.
- `-ram <ramsize>` specifies banked RAM size in KB (8, 16, 32, ..., 2048). The default is 512.
- `-cart <crtfile.crt>` loads a cartridge file. This requires a specially formatted cartridge file, as specified in the documentation.
- `-cartbin <romfile.bin>` loads a raw cartridge file. This will be loaded starting at ROM bank 32. All cart banks will be flagged as RAM.
- `-joy1`, `-joy2`, `-joy3`, `-joy4` enables binding a gamepad to that SNES controller port
- `-nvram` lets you specify a 64 byte file for the system's non-volatile RAM. If it does not exist, it will be created once the NVRAM is modified.
- `-keymap` tells the KERNAL to switch to a specific keyboard layout. Use it without an argument to view the supported layouts.
- `-sdcard` lets you specify an SD card image (partition table + FAT32). Without this option, drive 8 will interface to the current directory on the host.
- `-fsroot <dir>` specifies a file system root for the HostFS interface. This lets you save and load files without an SD card image. (As of R42, this is the preferred method.)
- `-serial` makes accesses to the host filesystem go through the Serial Bus [experimental].
- `-nohostieee` disables IEEE API interception to access the host fs.
- `-warp` causes the emulator to run as fast as possible, possibly faster than a real X16.
- `-gif <filename>[,wait]` to record the screen into a GIF. See below for more info.
- `-wav <filename>[{,wait|,auto}]` to record audio into a WAV. See below for more info.
- `-quality` change image scaling algorithm quality
  - `nearest` : nearest pixel sampling
  - `linear` : linear filtering
  - `best` : (default) anisotropic filtering
- `-log` enables one or more types of logging (e.g. `-log KS`):
  - `K` : keyboard (key-up and key-down events)
  - `S` : speed (CPU load, frame misses)
  - `V` : video I/O reads and writes
- `-debug` enables the debugger.
- `-dump` configure system dump (e.g. `-dump CB`):
  - `C` : CPU registers (7 B: A,X,Y,SP,STATUS,PC)
  - `R` : RAM (40 KiB)
  - `B` : Banked RAM (2 MiB)
  - `V` : Video RAM and registers (128 KiB VRAM, 32 B composer registers, 512 B palette, 16 B layer0 registers, 16 B layer1 registers, 16 B sprite registers, 2 KiB sprite attributes)
- `-sound` can be used to specify the output sound device.

- `-abufs` can be used to specify the number of audio buffers (defaults to 8). If you're experiencing stuttering in the audio try to increase this number. This will result in additional audio latency though.
- `-via2` installs the second VIA chip expansion at $9F10.
- `-midline-effects` enables mid-scanline raster effects at the cost of vastly increased host CPU usage.
- `-mhz <n>` sets the emulated CPU's speed. Range is from 1-40. This option is mainly for testing and benchmarking.
- `-enable-ym2151-irq` connects the YM2151's IRQ pin to the system's IRQ line with a modest increase in host CPU usage.
- `-wuninit` enables warnings on the console for reads of uninitialized memory.
- `-zeroram` fills RAM at startup with zeroes instead of the default of random data.
- `-version` prints additional version information of the emulator and ROM.
- When compiled with `#define TRACE`, `-trace` will enable an instruction trace on stdout.

Run `x16emu -h` to see all command line options.

Keyboard Layout

The X16 uses a PS/2 keyboard, and the ROM currently supports several different layouts. The following table shows their names, and what keys produce different characters than expected:

Name	Description	Differences
en-us	US	[`] ⇒ [←], [~] ⇒ [π], [\] ⇒ [£]
en-gb	United Kingdom	[`] ⇒ [←], [~] ⇒ [π]
de	German	[§] ⇒ [£], [´] ⇒ [^], [^] ⇒ [←], [°] ⇒ [π]
nordic	Nordic	key left of [1] ⇒ [←],[π]
it	Italian	[\] ⇒ [←], [|] ⇒ [π]
pl	Polish (Programmers)	[`] ⇒ [←], [~] ⇒ [π], [\] ⇒ [£]
hu	Hungarian	[\] ⇒ [←], [|] ⇒ [π], [§] ⇒ [£]
es	Spanish	[|] ⇒ π, \ ⇒ [←], Alt + [<] ⇒ [£]
fr	French	[²] ⇒ [←], [§] ⇒ [£]
de-ch	Swiss German	[^] ⇒ [←], [°] ⇒ [π]
fr-be	Belgian French	[²] ⇒ [←], [³] ⇒ [π]
fi	Finnish	[§] ⇒ [←], [½] ⇒ [π]
pt-br	Portuguese (Brazil ABNT)	[\] ⇒ [←], [|] ⇒ [π]

Keys that produce international characters (like [ä] or [ç]) will not produce any character.

Since the host computer tells the Commander X16 via the emulator the *position* of keys that are pressed, you need to configure the layout for the X16 independently of the keyboard layout you have configured on the host.

Use the `MENU` command to select a layout, or set the keyboard layout at startup using the `-keymap` command line argument.

The following keys can be used for controlling games:

Keyboard Key	SNES Equivalent
X or Ctrl	A
Z or Alt	B
S	X
A	Y
D	L
C	R
Shift	SELECT
Enter	START
Cursor Up	UP
Cursor Down	DOWN
Cursor Left	LEFT
Cursor Right	RIGHT

Functions while running

Windows and Linux

- `Ctrl` + `F` and `Ctrl` + `Return` will toggle full screen mode.
- `Ctrl` + `M` will toggle mouse capture mode.
- `Ctrl` + `P` will write a screenshot in PNG format to disk.
- `Ctrl` + `R` will reset the computer.
- `Ctrl` + `Backspace` will send an NMI to the computer (like RESTORE key).
- `Ctrl` + `S` will save a system dump configurable with `-dump`) to disk.
- `Ctrl` + `V` will paste the clipboard by injecting key presses.
- `Ctrl` + `=` and `Ctrl` + `+` will toggle warp mode.

Mac OS

- `⌘F` and `⌘Return` will toggle full screen mode.
- `⇧⌘M` will toggle mouse capture mode.
- `⌘P` will write a screenshot in PNG format to disk.
- `⌘R` will reset the computer.
- `⌘Delete` aka `⌘Backspace` will send an NMI to the computer (like RESTORE key).
- `⌘S` will save a system dump (configurable with `-dump`) to disk.
- `⌘V` will paste the clipboard by injecting key presses.
- `⌘=` and `⇧⌘+` will toggle warp mode.

GIF Recording

With the argument `-gif`, followed by a filename, a screen recording will be saved into the given GIF file. Please exit the emulator before reading the GIF file.

If the option `,wait` is specified after the filename, it will start recording on `POKE $9FB5,2`. It will capture a single frame on `POKE $9FB5,1` and pause recording on `POKE $9FB5,0`. `PEEK($9FB5)` returns a 128 if recording is enabled but not active.

WAV Recording

With the argument `-wav`, followed by a filename, an audio recording will be saved into the given WAV file. Please exit the emulator before reading the WAV file.

If the option `,wait` is specified after the filename, it will start recording on `POKE $9FB6,1`. If the option `,auto` is specified after the filename, it will start recording on the first non-zero audio signal. It will pause recording on `POKE $9FB6,0`. `PEEK($9FB6)` returns a 1 if recording is enabled but not active.

Emulator I/O registers

x16-emulator exposes registers in the range of, from `$9FB0` - `$9FBF`, which allows one to control or toggle various emulator features from within emulated code.

When writing machine code that uses these registers, good practice is to read `$9FBE` and `$9FBF` and check for their return values. If the emulator is present, those memory locations will return the ASCII/PETSCII characters "1" and "6" respectively (`$31` and `$36` hex). After verifying that the code is running under the emulator, you can confidently use the features provided by these registers.

Several of the following registers are particularly useful for debugging. In particular, writing data to `$9FB9`, `$9FBA`, or `$9FBB` will output debug information to the console, terminal, or command prompt window from which you ran x16emu.

Register	Read Behavior	Write Behavior
$9FB0	Returns debugger enabled flag	`0` disables, `1` enables the debugger, overriding the absence or presence of the `-debug` command line argument.
$9FB1	Returns video logging flag	`0` disables, `1` enables logging of VRAM accesses to the console
$9FB2	Returns keyboard logging flag	`0` disables, `1` enables logging of keyboard events to the console
$9FB3	Returns echo mode	`0` disables, `1` enables raw echo, `2` enables cooked (`\Xnn` for non-ASCII), and `3` enables ISO (w/ conversion to UTF-8). When on, characters sent via the `BSOUT` KERNAL call will also appear on the console.
$9FB4	Returns save-on-exit flag	`0` disables, `1` enables save-on-exit. When this option is set and the program counter reaches $FFFF, the

		emulator outputs a dump of emulator state to `dump.bin` before exiting.
$9FB5	Returns GIF recorder state	`0` pauses, `1` captures a single frame, and `2` activates/resumes GIF recording. The path to the GIF file must have been passed to the `-gif` command line option in advance.
$9FB6	Returns WAV recorder state	`0` pauses, `1` enables WAV recording, and `2` sets up autostart. The path to the WAV file must have been passed to the `-wav` command line option in advance.
$9FB7	Returns emu command key flag	`0` allows, and `1` inhibits most emulator command keys. Setting this flag prevents the emulator from intercepting keystrokes such as Ctrl+V/⌘V or Ctrl+R/⌘R, allowing the Commander X16 application running inside to make use of them.
$9FB8	Latches the cpu clock counter and returns bits 0-7	Resets the cpu clock counter to 0
$9FB9	Returns bits 8-15 from the latched cpu clock counter value	Outputs `"User debug 1: $xx"` to the console with xx replaced by the value written.
$9FBA	Returns bits 16-23 from the latched cpu clock counter value	Outputs `"User debug 2: $xx"` to the console with xx replaced by the value written.
$9FBB	Returns bits 24-31 from the latched cpu clock counter value	Outputs the given character to the console. This is basically a STDOUT port for programs running in the emulator. Only printable characters are allowed. Non-printables are replaced with `&#xFFFD;`.
$9FBC	-	-
$9FBD	Returns the keymap index, based on the argument to the `-keymap` command line option	-
$9FBE	Returns the value `$31`/ASCII "1", useful for emulator presence detection	-
$9FBF	Returns the value `$36`/ASCII "6", useful for emulator presence detection	-

BASIC and the Screen Editor

On startup, the X16 presents direct mode of BASIC V2. You can enter BASIC statements, or line numbers with BASIC statements and `RUN` the program, just like on Commodore computers.

- To stop execution of a BASIC program, hit the `RUN/STOP` key (`Pause`), or `Ctrl+C` .
- To insert characters, first insert spaces by pressing `Shift+Backspace` or `Insert` , then type over those spaces.
- To clear the screen, press `Shift+Home` .
- To send NMI, similar to `STOP+RESTORE` on the C64, use Ctrl+Backspace/⌘Delete. On real hardware this is done with `Ctrl+Alt+RESTORE` (`Ctrl+Alt+PrtScr`) or by pressing the NMI button.

SD Card Images

The command line argument `-sdcard` lets you attach an image file for the emulated SD card. Using an emulated SD card makes filesystem operations go through the X16's DOS implementation, so it supports all filesystem operations (including directory listing though `DOS"$` command channel commands using the `DOS` statement) and guarantees full compatibility with the real device.

Images must be greater than 32 MB in size and contain an MBR partition table and a FAT32 filesystem. The file `sdcard.img.zip` in this repository is an empty 100 MB image in this format.

On macOS, you can just double-click an image to mount it, or use the command line:

```
# hdiutil attach sdcard.img
/dev/disk2              FDisk_partition_scheme
/dev/disk2s1            Windows_FAT_32                  /Volumes/X16 DISK
# [do something with the filesystem]
# hdiutil detach /dev/disk[n] # [n] = number of device as printed above
```

On Linux, you can use the command line:

```
# sudo losetup -P /dev/loop21 disk.img
# sudo mount /dev/loop21p1 /mnt # pick a location to mount it to, like /mnt
# [do something with the filesystem]
# sudo umount /mnt
# sudo losetup -d /dev/loop21
```

On Windows, you can use the OSFMount tool. Windows VHD files can also be created using the built-in Disk Manager. Careful attention should be paid to the settings when creating and formatting the VHD:

- The file must be at least 32MB and must be fixed size. Expanding VHDs are not supported.
- Use an MBR partition tables. The Commander X16 does not recognize GPT partition tables.
- You must format the VHD with FAT32. Other file formats are not supported.

This is a trick, since Fixed-size VHD files contain the data first, with the metadata in a footer at the end. Since the emulator does not read or edit that medatada, it will only work with fixed-size files that are fully populated.

Host Filesystem Interface

If the system ROM contains any version of the KERNAL, and there is no SD card image attached, all accesses to the ("IEEE") Commodore Bus are intercepted by the emulator for device 8 (the default). So the BASIC statements will target the host computer's local filesystem:

```
DOS"$"
LOAD"FOO.PRG"
LOAD"IMAGE.PRG",8,1
SAVE"BAR.PRG"
OPEN2,8,2,"FOO,S,R"
```

The emulator will interpret filenames relative to the directory it was started in. On macOS, when double-clicking the executable, this is the home directory. To specify a different path as the emulated root, you can use the `-fsroot` command line option.

To avoid compatibility problems between the PETSCII and ASCII encodings, you can

- use uppercase filenames on the host side, and unshifted filenames on the X16 side.
- use `Ctrl+O` to switch to the X16 to ISO mode for ASCII compatibility.
- use `Ctrl+N` to switch to the upper/lower character set for a workaround.

As of R42, the Host Filesystem interface (or HostFS) is the preferred method of accessing files. It does not require creating or managing an SDcard image, and it supports all of the CMDR-DOS commands. However, it is not cycle-accurate, since the emulator traps calls to DOS and performs the same actions in the host environment. If performance and hardware accuracy is required, you will want to perform final testing using an SD card image.

Dealing with BASIC Programs

BASIC programs are encoded in a tokenized form when saved. They are not simply ASCII files. If you want to edit BASIC programs on the host's text editor, you need to convert it to tokenized BASIC encoding from ASCII encoding before calling `LOAD` in the emulator.

- To convert the basic file from ASCII to tokenized BASIC encoding, reboot the machine and paste the ASCII text using `Ctrl + V` (Mac: `Cmd + V`) into the terminal. You can now run the program with `RUN` , or use the `SAVE` BASIC command to write the tokenized version to the host disk. Below is an example.

    1. Copy ASCII text from host basic file "PRG.BAS"
    2. Paste into new terminal session
    3. `SAVE"ENCODED.BAS`
    4. Now you can restart the emulator and load the encoded basic file with `LOAD"ENCODED.BAS"`
    5. Run with `RUN"ENCODED.BAS"`

- To convert BASIC to ASCII, start x16emu with the `-echo` argument, `LOAD` the BASIC file, and type `LIST` . Now copy the ASCII version from the terminal.

Using the KERNAL/BASIC environment

Please see the KERNAL/BASIC documentation .

Debugger

The debugger requires `-debug` to start. Without it, it is disabled.

There are 2 panels you can control. The code panel, the top left half, and the data panel, the bottom half of the screen. You can also edit the contents of the registers PC, A, X, Y, and SP.

The debugger uses its own command line with the following syntax:

Statement	Description
d %x	Change the code panel to view disassembly starting from the address %x.
m %x	Change the data panel to view memory starting from the address %x.
v %x	Display VERA RAM (VRAM) starting from address %x.
b %s %d	Changes the current memory bank for disassembly and data. The %s param can be either 'ram' or 'rom', the %d is the memory bank to display (but see NOTE below!).
r %s %x	Changes the value in the specified register. Valid registers in the %s param are 'pc', 'a', 'x', 'y', and 'sp'. %x is the value to store in that register.

NOTE. To disassemble or dump memory locations in banked RAM or ROM, prepend the bank number to the address; for example, "m 4a300" displays memory contents of BANK 4, starting at address $a300. This also works for the 'd' command.

The debugger keys are similar to the Microsoft Debugger shortcut keys, and work as follows

Key	Description
F1	resets the shown code position to the current PC
F2	resets the 65C02 CPU but not any of the hardware.
F5	close debugger window and return to Run mode, the emulator should run as normal.
F9	sets the breakpoint to the currently code position.
F10	steps 'over' routines - if the next instruction is JSR it will break on return.
F11	steps 'into' routines.
F12	is used to break back into the debugger. This does not happen if you do not have -debug
PAGE UP	is used to scroll up in the debugger.
PAGE DOWN	is used to scroll down in the debugger.
TAB	when stopped, or single stepping, hides the debug information when pressed

When `-debug` is selected the STP instruction (opcode $DB) will break into the debugger automatically.

Keyboard routines only work when the emulator is running normally. Single stepping through keyboard code will not work at present.

CRT File Format

The Commander X16 will support cartridge ROMs, including auto-booting game cartridges. On the Gen-1 Developer board, the first slot will be used for cartridges. On the Gen-2 console machine, there is only one slot. ROM carts should work on both systems.

This CRT format is intended for the emulator, and it is not required or used by the hardware. You can, however, use the MakeCart tool to convert between a single CRT file and BIN files that can be used to program a ROM burner. Also, note that this is different from the CRT format used the VICE emualtor, so files are not interchangable.

Commander X16 cartridges will occupy the same address space as the Commander's KERNAL and BASIC ROMs. You can control the active bank by writing to address $0001 on the computer. Banks 0-31 are the built-in ROM banks, and banks 32-255 will select the cartridge ROMs.

Header Layout

This is the cartridge header. The first 256 bytes are ASCII data and Human readable. The second 256 bytes are bank data; these are byte integers. Text fields are set to 16 or 32-byte boundaries for ease of formatting.

Location	Length	Description
00-15	16	ASCII text: CX16 CARTRIDGE\r\n
16-31	16	CRT format version. ASCII digits in format 01.02, space padded.
32-63	32	Name. ASCII text.
64-95	32	Programmer/Developer. ASCII text.
96-127	32	Copyright information. ASCII text.
128-191	32	Program version. ASCII text.
192-255	64	Empty.
256-287	32	Fill with zeros.
288-511	224	Bank Flags.
		00: Not Present. No data is present in the emulator or in the file.
		01: ROM: 16KB of ROM data. Data is write protected in emulator.
		02: RAM: No data in file. Bank is read/write in emulator.
		03: RAM: Data present: data is loaded from the file and discarded on shutdown. Useful for testing.
		04: NVRAM: No Data in file. Memory is writeable. Emulator saves data to NVRAM file.
		05: NVRAM: Data present. Memory is writeable. Emulator saves data to NVRAM file.
512-end		Payload data.
		16384 bytes per bank for types 1, 3, and 5.

		0 bytes for types 0,2, and 4.

For NVRAM banks: on shutdown, the emulator will write out an NVRAM file that contains the data of all of the NVRAM banks. The next time this cartridge is started, the NVRAM file will be loaded into any NVRAM bank. This overwrites any data present in NVRAM banks in the CRT file.

For types 00, 02, and 04: The file does *not* contain data for these bank types. Instead, the file skips straight to the next bank with initialized data (01, 03, or 05).

For all "No Data" banks, the data in RAM is *undefined*. While the emulator currently initializes RAM to 0 bytes, the hardware will have random values. In addition, unpopulated addresses will be "open collector" and will have unpredicatable results.

Vectors

X16 hardware, and thus the emulator, will only read 6502 vectors out of bank 0. This is done via the CPU's VPB pin being connected to the ROM bank latch reset pin. In the past specific vectors were recommended in cartridge ROMs, but this is no longer true. In cartridges, the addresses `$FFFA` - `$FFFF` are free to use for data.

MakeCart Conversion Tool

A conversion tool to pack cartridge data into a CRT file, `makecart`, is included in this release.

`-cfg <filename.cfg>` Use this file to pack the cartridge data. Config file is simply the command line switches, one per line.

`-desc "Name/Description"` Set the description field of the cartridge file. Up to 32 bytes of ASCII text.

`-author "Author Information"` Set the author information field of the cartridge file. Up to 32 bytes of ASCII text.

`-copyright "Copyright Information"` Set the copyright information field of the cartridge file. Up to 32 bytes of ASCII text.

`-version "version"` Set the version information field of the cartridge file. Up to 32 bytes of ASCII text.

`-fill <value>` Set the fill value to use with any partially-filled banks of cartridge memory. Value can be defined in decimal, or in hexadecimal with a '$' or '0x' prefix. 8-bit values will be repeated every byte, 16-bit values every two bytes, and 32-bit values every 4 bytes.

`-rom_file <start_bank> [<filename.bin> [<filename.bin>] ... ]`

Define rom banks from the specified list of files. File data is tightly packed -- if a file does not end on a 16KB interval, the next file will be inserted immediately after it within the same bank. If the last file does not end on a 16KB interval, the remainder of the rom will be filled with the value set by '-fill'.

Valid bank numbers are 32 - 255.

`-ram <start_bank> [<end bank>]` Define one or more banks of RAM. RAM banks are not included in the payload.

`-ram_file <start_bank> [<filename.bin> [<filename.bin>] ... ]` Define one or more banks of initialized RAM. Note that Initialized RAM banks are not saved to the NVRAM file at shutdown.

`-nvram <start_bank> [<end_bank>]` Define one or more uninitalized nvram banks.

`-nvram_value <start_bank> <end_bank>` Define pre-initialized nvram banks with the value set by '-fill'. Repeated payload bytes will be written to the file.

`-nvram_file <start_bank> [<filename.bin> [<filename.bin>] ... ]`

Define pre-initialized nvram banks from the specified list of files. File data is tightly packed like with -rom. If the last file does not end on a 16KB interval, the remainder of the rom will be filled with the value set by '-fill'.

`-none <start_bank> [<end_bank>]` Define one or more unpopulated banks of the cartridge. By default, all banks are unpopulated unless specified by a previous command-line option. These banks are not present in the payload and only popualte the bank header in the CRT file.

`-o <output.crt>` Set the filename of the output cartridge file.

All options can be specified multiple times, and are applied in-order from left to right. For -desc and -o, it is legal to specify them multiple times but only the right-most instances of each will have effect.

`-unpack <input.crt> [<rom_size>]` Unpacks the binary data from the cartridge file into `<rom_size>` slices. (for use with an EPROM programmer.) The ouptut files will be the same filename as the input file, with _### appended. This will also create a .cfg file that can be used to re-pack the files into a new CRT if needed.

The config file is just a series of command-line switches, with one item per line. This example assumes ladder.bin uses 3 banks, for a total of 48K, and that each level map is 4KB in size.

```
-o ladder.crt
-name "Ladder"
-author "Yahoo Software"
-copyright "(c) 1982, 1983 Yahoo Software"
-version "1.30TP"
-rom_file 32 ladder.bin
-rom_file 35 level_01.bin level_02.bin level_03.bin level_04.bin
-nvram 37
-fill 0
```

This would create file with

- 512 byte header
- 5 ROM banks
  - 3 for the 48K ladder.bin
  - 1 for the four 4KB level files.
- 1 empty NVRAM bank

Since the NVRAM bank is not initialized, it is not included in the file. This makes the file a total of 66,048 bytes long. (512 bytes, plus four 16KB banks.)

Web Site

https://commanderx16.com

Forum

https://cx16forum.com/forum

License

Release Notes

See RELEASES .

Releases

Release 45 ("Nuuk")

This is a minor release with respect to the emulator. The bulk of the changes are in the ROM .

- Features/Fixes
  - Revert VERA PSG amplitude resolution back to 6 bits. This was upped previously to match VERA firmware. It was subsequently reverted in VERA to make room for the FX feature. [akumanatt]
  - Intellimouse support added to the emulated SMC, partially implementing the new feature in hardware SMC firmware 45.1.0. [stefan-b-jakobsson]
    - Scroll wheel is supported (mouse device ID 3)
    - Not implemented: Extra buttons (mouse device ID 4)
  - New emulator debug register behaviors
    - Reading from $9FB8-$9FBB in this order returns the 32-bit CPU clock counter, snapshotted at the time $9FB8 is read. Previously the clock counter would remain in motion and the upper counter bits could roll over unpredictably.
    - Writing to $9FB8-$9FBB has new behaviors:
      - $9FB8: resets the cpu clock counter to 0.
      - $9FB9: prints a debug message to the console `User debug 1: $xx`.
      - $9FBA: prints a debug message to the console `User debug 2: $xx`.
      - $9FBB: prints the UTF-8 representation of the ISO character to the console. This can be treated like a debug STDOUT.
    - Before using any of these emulator debug registers, it's recommended to test for emulator presence first.
      - Read from `$9FBE` and `$9FBF`. When running under the emulator, the returned values should be `$31` and `$36` respectively. If any other values are returned, you can usually assume to be running on real hardware. While the stock machine doesn't have any I/O devices that

listen to the emulator I/O range, an add-on card could choose to use that same address space in the future for its own functions.

- New MCIOUT (blockwise write) implementation for HostFS, mirroring the feature in the kernal for use on SD card.
- New command key and capture behavior:
  - Ctrl+M/⇧⌘M is always processed by the emulator to toggle (mouse/keyboard) capture, regardless of the state of the "Disable Emulator Keys" flag.
  - Turning capture mode on disables all other emulator keys until capture mode is toggled off. While capture is on, the emulator also routes most OS shortcut key combos to the X16: for instance, Alt+Tab.

Release 44 ("Milan")

This is the third release of x16-emulator by the X16Community team

- Features/Fixes
  - Many changes to HostFS, including
    - Fix regression for loading "`:*`" from HostFS while also using an SD card image
    - Fix `-prg` and `-sdcard` options working together, which did not properly handle the case after emulator reset
    - Proper wildcard behavior in dir filters and OPEN string
    - Fix filetype directory filter command parsing
    - Add `$=L` long mode directory listing to emulate the new feature in the ROM.
    - Speed up directory filetype filter
    - Add CMD/SD2IEC style directory navigation: `CD:←` to enter parent directory
    - Partial emulation of case-insensitivity in filenames on case-sensitive host filesystems.
      - This works in `OPEN` strings and directory names in commands (`CD:` , etc.) which do not contain a `/` character.
      - It also works on the last path segment in relative or absolute paths in directory names or `OPEN` strings. In other words, if given a path specification containing one or more `/` characters, it will do only a case-insensitive search on the part after the final `/` .
    - Proper translation between UTF-8 filenames and their ISO representations.
  - Implement VPB behavior to match hardware [akumanatt]
    - Fix BRK to have VPB behavior
  - Support for additional keycodes (NumLock, Menu) [stefan-b-jakobsson]
  - Fix debugger to set the correct bank for breakpoints [gaekwad]
  - New `-fullscreen` CLI option
  - Proper cleanup when the emulator exits when in full screen [irmen]
  - FX emulation, which mirrors the features in the FX enhancement to the VERA firmware. See the VERA FX Reference for details.
  - Writing bit 6 and bit 7 together into VERA_AUDIO_CTRL now enables looping the PCM FIFO (and does not reset the FIFO). Any other write into VERA_AUDIO_CTRL disables looping.
  - New `-opacity` CLI option for window transparency [tstibor]
  - New support for screenshots (Ctrl+P/⌘P) [dressupgeekout]
  - Fix small memory leak caused by pasting into the emulator

    - Use relative mouse motion while in grabbed mode
    - Remove `-geos` CLI option [dressupgeekout]
    - New YM2151 audio core: remove old MAME core, replace with ymfm
        - Allows for IRQs from the YM, requires specifying `-enable-ym2151-irq` on the command line
    - Emulate hardware open bus behavior when reading from a device that doesn't exist in the `$9Fxx` space
    - Reset via I2C command: defer machine reset to the main loop, which allows the I2C write routine to return cleanly.
    - Fix 65C02 `BIT` immediate behavior [XarkLabs]
    - New NMI trigger emulator hotkey, emulates Ctrl+Alt+Restore on hardware (Ctrl+Backspace/⌘Delete) [XarkLabs]
    - Fix line artifact in application icon/logo
    - Grabbing the mouse with (Ctrl+M/⇧⌘M) now grabs the keyboard as well. It allows the emulator to receive keystrokes and key combinations which would otherwise be intercepted by the operating system.
    - Fix description of fill value in `makecart`
    - New features implemented in the ROM
- Build
    - Link-time optimization is now enabled by default
    - Portability enhancements [dressupgeekout]
    - Suppress clang warnings due to deprecated sprintf usage in ymfm lib [XarkLabs]

Release 43 ("Stockholm")

This is the second release of x16-emulator by the X16Community team

- **BREAKING CHANGE**
    - The keyboard protocol between the emulated SMC and the KERNAL has changed, thus x16-emulator version R43 requires x16-rom version R43.
    - This change also affects how the custom keyboard handler vector works (keyhdl). For details, see Chapter 2 of the Programmer's Reference Guide
    - **Your Keyboard will not work unless** you are running
        - R43 of both x16-rom and x16-emulator
- Features
    - Updates to support translation from SDL scancodes to new keynum encoding supported by KERNAL [stefan-b-jakobsson]
    - More granular support for RAM amount as argument to `-ram`
    - Minor HostFS bugfixes and enhancements, including tying the activity light to HostFS activity.
    - VERA updates: new support for 240p in NTSC/RGB modes. Chroma disable only works on NTSC.
    - Stepping the debugger now supports stepping over `WAI`
    - Debugger now shows the correct bank in the disassembly by default. [gaekwad]
    - Debugger breakpoints are now bank-specific [gaekwad]
    - Randomized RAM is now the default. New option: `-zeroram` [irmen]
    - Host's mouse cursor is now shown unless either the KERNAL mouse is enabled or the mouse cursor is captured (Ctrl+M/⇧⌘M).
    - Esc key is now Esc rather than STOP. Pause key sends STOP. (Ctrl+C is also recognized by the KERNAL as STOP)

    - SD card emulation now responds to CMD9
    - Emulated SMC can now assert NMI.
    - Add `-mhz` option to select a speed other than 8
    - When built with `TRACE`, the `-trace` output now shows the effective address for indirect and indexed opcodes and VERA data0/data1 reads and writes.
    - New comamnd line option `-midline-effects` that supports mid-line changes to the palette or tile/sprite data. R42 always had this behavior, which results in performance degradation for programs write to VERA heavily if the host CPU is not fast enough. This behavior is now disabled by default. `-midline-effects` restores this optional behavior.
    - New features implemented in the ROM
- Other
    - Release builds have link-time optimization enabled which seems to help performance.
    - Add git hash of build to `-version` string.
    - WebAssembly enhancements in the supporting html/js [Cyber-EX]
    - Fixed potential off-by one row with non-zero DC_VSTART.
    - Prevent laggy hostfs reads from causing the emulator to warp to catch up by translating the wall clock time to elapsed 6502 clocks. This effectively makes HostFS MACPTR behave like a DMA card, including the possibility that it prevents the CPU from executing instructions while interrupt sources may have been waiting for service.
    - Bugfix: Process multiple SDL events per frame. (Fixed choppy mouse movement if there were keystrokes in the keyboard buffer)
    - Audio resampling and ring buffer fixes [DragWx]
    - Build fixes on Mac

Release 42 ("Cambridge")

This is the first release of x16-emulator by the X16Community team

- Features
    - Added testbench mode [stefan-b-jakobsson, indigodarkwolf]
    - Added `-noemucmdkeys` option [jestin]
    - New `FIFO_EMPTY` flag in `PCM_CTRL` to reflect new VERA feature [ZeroByteOrg]
    - Added `-widescreen` option to simulate stretched 640x480 output at a 16:9 aspect ratio [jestin]
    - New `SCANLINE` VERA register behavior to reflect updated VERA feature [mooinglemur]
    - Added `-randram` and `-wuninit` command line arguments to randomize RAM at boot, and to emit a console warning when uninitialized RAM is read, respectively. [stefan-b-jakobsson]
    - Allow specifying non-power-of-2 argument to `-ram`, in increments of 8k [JimmyDansbo]
    - Added `-via2` option to selectively enable a VIA at $9F10. [akumanatt]
    - Added ROM cart loading with `-cart` and `-cartbin` [indigodarkwolf]
    - New `makecart` utility for building `.crt` cartridge files [indigodarkwolf]
    - Compressed SD card image support [indigodarkwolf]
    - Mouse grab mode, press Ctrl+M (Mac: ⇧⌘M) to toggle. [mooinglemur]
    - New `-fsroot` and `-startin` options to specify the root of the emulated host fs, and the host directory to start in respectively. [mooinglemur]
    - Many, many new features implemented in the ROM

- Other
  - PS/2 devices now connected via SMC via I2C, I2C pins have moved to match hardware [stefan-b-jakobsson]
  - Recognize middle mouse button [ZeroByteOrg]
  - Synchronized keymaps with ROM [megagrump]
  - Build fixes [irmen]
  - Show dialog when a `STP` instruction is encountered with debug turned off [akumanatt]
  - Improved emulated behavior of `WAI` [LRFLEW]
  - Clear D flag on interrupt entry [LRFLEW]
  - Update BRK length in debugger [indigodarkwolf]
  - IRQ/NMI entry clock cycles are now accounted for [mooinglemur]
  - Add reason string to memory dump output [irmen]
  - Clear sprite line buffer when disabling sprite layer [jestin]
  - Improved audio balance between VERA and YM2151. Much improved mixing routines to reduce stutters and clicking. [akumanatt]
  - To match hardware, VERA ISR bits are set at VSYNC, LINE, and SPRCOL regardless of whether their respective IEN bits are set [mooinglemur]
  - Changes to match Proto 4, including moving VIA1 interrupt pin to IRQ [akumanatt]
  - VERA mid-frame raster effects more closely match the timing of real hardware [mooinglemur]
  - Enabled and built out CI/CD build workflows [maxgerhardt, indigodarkwolf, mooinglemur]
  - Many host fs enhancements, bringing host fs very close to feature parity with SD card images [davidgiven, ZeroByteOrg, mooinglemur]
  - Many documentation updates and fixes [veganaize, irmen, tomxp411]

Release 41 ("Marrakech")

- allow apps to intercept Cmd/Win, Menu and Caps-Lock keys
- fixed `-prg` with `-sdcard`
- fixed loading from host filesystem (length reporting by `MACPTR` on EOI)
- macOS: support for older versions like Catalina (10.15)

Release 40 ("Bonn")

- Features
  - improved VERA video timings [Natt Akuma]
  - added Host FS bridging using IEEE API
  - added Serial Bus emulation [experimental]
  - added WAV file recording [Stephen Horn]
  - possible to disable Ctrl/Cmd key interception ($9FB7) [mooinglemur]
- Other
  - Fixed I2C (RTC, SMC)
  - Fixed RAM/ROM bank for PC when entering break [mjallison42]
  - LST support for -trace

Release 39 ("Buenos Aires")

- Switch to Proto2 Hardware
  - banking through zp addresses 0 and 1
  - modified I/O layout

    - modified VIA GPIO layout
    - support for 4 controllers
    - I2C bus with SMC and RTC/NVRAM
- Features
    - implemented VIA timers [Natt Akuma]
    - added option to disable sound [Jimmy Dansbo]
    - added support for Delete, Insert, End, PgUp and PgDn keys [Stefan B Jakobsson]
    - debugger scroll up & down description [Matas Lesinskas]
    - added anti-aliasing to VERA PSG waveforms [TaleTN]
- Bugs
    - fixed sending only one mouse update per frame [Elektron72]
    - fixed VSYNC timing [Elektron72]
    - switched front and back porches [Elektron72]
    - fixed LOAD/SAVE hypercall so debugger doesn't break [Stephen Horn]
    - fixed YM2151 frequency from 4MHz ->3.579545MHz [Stephen Horn]
    - do not set compositor bypass hint for SDL Window [Stephen Horn]
    - reset timing after exiting debugger [Elektron72]
    - don't write nvram after every frame
    - fixed write outside of line buffer [Stephen Horn]
    - fixed BRA extra CPU cycle [LRFLEW]
    - fix: clear layer line once layer is disabled
    - fixed BBSx/BBRx timing [Natt Akuma]
- Other
    - misc speed optimizations [Stephen Horn]

Release 38 ("Kyoto")

- CPU
    - added WAI, BBS, BBR, SMB, and RMB instructions [Stephen Horn]
- VERA
    - VERA speed optimizations [Stephen Horn]
    - fixed raster line interrupt [Stephen Horn]
    - added sprite collision interrupt [Stephen Horn]
    - fixed sprite wrapping [Stephen Horn]
    - added VERA dump, fill commands to debugger [Stephen Horn]
    - fixed VRAM memory dump [Stephen Horn]
- SD card
    - SD card write support
    - Ctrl+D/Cmd+D detaches/attaches SD card (for debugging)
    - improved/cleaned up SD card emulation [Frank van den Hoef]
    - SD card activity/error LED support
    - VERA-SPI: support Auto-TX mode
- misc
    - added warp mode (Ctrl+'+'/Cmd+'+' to toggle, or `-warp`)
    - added '-version' shell option [Alice Trillian Osako]
    - new app icon [Stephen Horn]
    - expose 32 bit cycle counter (up to 500 sec) in emulator I/O area
    - zero page register display in debugger [Mike Allison]

- Various WebAssembly improvements and fixes [Sebastian Voges]

Release 37 ("Geneva")

- VERA 0.9 register layout [Frank van den Hoef]
- audio [Frank van den Hoef]
  - VERA PCM and PSG audio support
  - YM2151 support is now enabled by default
  - added `-abufs` to specify number of audio buffers
- removed UART [Frank van den Hoef]
- added window icon [Nigel Stewart]
- fixed access to paths with non-ASCII characters on Windows [Serentty]
- SDL HiDPI hint to fix mouse scaling [Edward Kmett]

Release 36 ("Berlin")

- added VERA UART emulation (`-uart-in` , `-uart-out`)
- correctly emulate missing SD card
- moved host filesystem interface from device 1 to device 8, only available if no SD card is attached
- require numeric argument for `-test` to auto-run test
- fixed JMP (a,x) for 65c02
- Fixed ESC as RUN/STOP [Ingo Hinterding]

Release 35

- video optimization [Neil Forbes-Richardson]
- added `-geos` to launch GEOS on startup
- added `-test` to launch (graphics) unit test on startup
- debugger
  - switch viewed RAM/ROM bank with `numpad +` and `numpad -` [Kobrasadetin]
  - optimized character printing [Kobrasadetin]
- trace mode:
  - prepend ROM bank to address in trace
  - also prints 16 bit virtual regs (graph/GEOS)
- fixes
  - initialize memory to 0 [Kobrasadetin]
  - fixed SYS hex argument
  - disabled "buffer full, skipping" and SD card debug text, it was too noisy

Release 34

- PS/2 mouse
- support for text mode with tiles other than 8x8 [Serentty]
- fix: programmatic echo mode control [Mikael O. Bonnier]

Release 33

- significant performance optimizations
- VERA
  - enabled all 128 sprites
  - correct sprite zdepth
  - support for raster IRQs

- SDL controller support using `-joy1` and `-joy2` [John J Bliss]
- 65C02 BCD fixes [Norman B. Lancaster]
- feature parity with new LOAD/VLOAD features [John-Paul Gignac]
- default RAM and ROM banks are now 0, matching the hardware
- GIF recording can now be controlled from inside the machine [Randall Bohn]
- Debugging
  - Major enhancements to the debugger [kktos]
  - `-echo` will now encode non-printable characters like this: \X93 for CHR$(93), `-bas` as well as pasting accepts this convention again
  - `-echo raw` for the original behavior
  - `-echo iso` for correct character encoding in ISO mode
  - `-ram` to specify RAM size; now defaults to 512

Release 32

- correct ROM banking
- VERA emulation optimizations [Stephen Horn]
- added `-dump` option to allow writing RAM, CPU state or VERA state to disk [Nils Hasenbanck]
- added `-quality` option to change scaling algorithm; now defaults to "best" [Maurizio Porrato]
- output of `-echo` can now be fed into UNIX pipes [Anonymous Maarten]
- relative speed of emulator is shown in the title if host can't keep up [Rien]
- fix: 6502 BCD arithmetic [Rien]
- fix: colors (white is now white) [Rien]
- fix: sprite flipping [jjbliss]

Release 31

- VERA 0.8 register layout
- removed `-char` (character ROM is now part of `rom.bin`)
- GIF recording using `-gif` [Neil Forbes-Richardson]
- numpad support [Maurizio Porrato]
- fake support of VIA timers to work around BASIC RND(0)
- default ROM is taken from executable's directory [Michael Watters]
- emulator window has a title [Michael Watters]
- `-debug` allows specifying a breakpoint [Frank Buss]
- package contains the ROM symbols in `rom.txt`
- support for VERA SPI

Release 30

Emulator:

- VERA can now generate VSYNC interrupts
- added `-keymap` for setting the keyboard layout
- added `-scale` for integer scaling of the window [Stephen Horn]
- added `-log` to enable various logging features (can also be enabled at runtime (POKE $9FB0+) [Randall Bohn])
- changed `-run` to be an option to `-prg` and `-bas`
- emulator detection: read $9FBE/$9FBF, must read 0x31 and 0x36
- fix: `-prg` and `-run` no longer corrupt BASIC programs.
- fix: `LOAD,1` into RAM bank [Stephen Horn]
- fix: 2bpp and 4bpp drawing [Stephen Horn]

- fix: 4bpp sprites [MonstersGoBoom]
- fix: build on Linux/ARM

Release 29

- better keyboard support: if you pretend you have a US keyboard layout when typing, all keys should now be reachable [Paul Robson]
- `-debug` will enable the new debugger [Paul Robson]
- runs at the correct speed (was way too slow on most machines)
- keyboard shortcuts work on Windows/Linux: `Ctrl + F/R/S/V`
- `Ctrl + V` pastes the clipboard as keypresses
- `-bas file.txt` loads a BASIC program in ASCII encoding
- `-echo` prints all BASIC/KERNAL output to the terminal, use it with LIST to convert a BASIC program to ASCII
- `-run` acts like `-prg`, but also autostarts the program
- `JMP $FFFF` and `SYS 65535` exit the emulator and save memory into the host's storage
- the packages now contain the current version of the Programmer's Reference Guide (HTML)
- fix: on Windows, some file load/saves may be been truncated

Release 28

- support for 65C02 opcodes [Paul Robson]
- keep aspect ratio when resizing window [Sebastian Voges]
- updated sprite logic to VERA 0.7 – **the layout of the sprite data registers has changed, you need to change your code!**

Release 27

- Command line overhaul. Supports `-rom`, `-char`, `-sdcard` and `-prg`.
- ROM and char filename defaults, so x16emu can be started without arguments.
- Host Filesystem Interface supports `LOAD"$"`
- macOS and Windows packaging logic in Makefile

Release 26

- better sprite support (clipping, palette offset, flipping)
- better border support
- KERNAL can set up interlaced NTSC mode with scaling and borders (compile time option)

Release 25

- sdcard: fixed `LOAD,x,1` to load to the correct addressg
- sdcard: all temp data will be on bank #255; current bank will remain unchanged
- DOS: support for DOS commands ("UI", "I", "V", ...) and more status messages (e.g. 26,WRITE PROTECT ON,00,00)
- BASIC: `DOS` command. Without argument: print disk status; with "$" argument: show directory; with "8" or "9" argument: switch default drive; otherwise: send DOS command; also accessible through F7/F8
- Vera: cycle exact rendering, NTSC, interlacing, border

Release 24

- SD card support
  - pass path to SD card image as third argument
  - access SD card as drive 8
  - the local PC/Mac disk is still drive 1

    - modulo debugging, this would work on a real X16 with the SD card (plus level shifters) hooked up to VIA#2PB as described in sdcard.c in the emulator surce

Release 23

- Updated emulator and ROM to spec 0.6 – the ROM image should work on a real X16 with VERA 0.6 now.

Release 22

SYS65375 (SWAPPER) now also clears the screen, avoid ing side effects.

Release 21

- support for $ and % number prefixes in BASIC
- support for C128 KERNAL APIs LKUPLA, LKUPSA and CLOSE_ALL

Release 20

- Toggle fullscreen using `Cmd + F` or `Cmd + return`
- new BASIC instructions and functions:
    - `MON` : enter monitor; no more SYS65280 required
    - `VPEEK(bank, address)`
    - `VPOKE bank, address, value` example: `VPOKE4,0,VPEEK(4,0) OR 32` [for 256 color BASIC]

Release 19

- fixed cursor trail bug
- fixed f7 key in PS/2 driver
- f keys are assigned with shortcuts now: F1: LIST F2: <enter monitor> F3: RUN F4: <switch 40/80> F5: LOAD F6: SAVE" F7: DOS"$ <doesn't work yet> F8: DOS <doesn't work yet>

Release 18

- Fixed scrolling in 40x30 mode when there are double lines on the screen.

Release 17

- video RAM support in the monitor (SYS65280)
- 40x30 screen support (SYS65375 to toggle)

Release 16

- Integrated monitor, start with SYS65280 `rom.bin` is now 3*8 KB:
    - 0: BASIC (bank 0 at $C000)
    - 1: KERNAL ($E000)
    - 2: UTIL (bank 1 at $C000)

Release 15

- correct text mode video RAM layout both in emulator and KERNAL

Release 14

- KERNAL: fast scrolling
- KERNAL: upper/lower switching using CHR$($0E)/CHR$($8E)
- KERNAL: banking init
- KERNAL: new PS/2 driver

- Emulator: VERA updates (more modes, second data port)
- Emulator: RAM and ROM banks start out as all 1 bits

Release 13

- Supports mode 7 (8bpp bitmap).

Release 12

- Supports 8bpp tile mode (mode 4)

Release 11

- The emulator and the KERNAL now speak the bit-level PS/2 protocol over VIA#2 PA0/PA1. The system behaves the same, but keyboard input in the ROM should work on a real device.

Release 10

updated KERNAL with proper power-on message

Release 9

- LOAD and SAVE commands are intercepted by the emulator, can be used to access local file system, like this:

```
LOAD"TETRIS.PRG
SAVE"TETRIS.PRG
```

- No device number is necessary. Loading absolute works like this:

```
LOAD"FILE.PRG",1,1
```

Release 8

- New optional override load address for PRG files:

```
./x64emu rom.bin chargen.bin basic.prg,0401
```

Release 7

- Now with banking. `POKE40801,n` to switch the RAM bank at $A000. `POKE40800,n` to switch the ROM bank at $C000. The ROM file at the command line can be up to 72 KB now (layout: 0: bank 0, 1: KERNAL, 2: bank 1, 3: bank 2 etc.), and the RAM that `Cmd + S` saves is 2088KB ($0000-$9F00: regular RAM, $9F00-$9FFF: unused, $A000+: extra banks)

Release 6

- Vera emulation now matches the complete spec dated 2019-07-06: correct video address space layout, palette format, redefinable character set

Release 5

- BASIC now starts at $0401 (39679 BASIC BYTES FREE)

Release 4

- `Cmd + S` now saves all of memory (linear 64 KB for now, including ROM) to `memory.bin`, `memory-1.bin`, `memory-2.bin`, etc. You can extract parts of it with Unix "dd", like: `dd`

```
if=memory.bin of=basic.bin bs=1 skip=2049 count=38655
```

Release 3

- Supports PRG file as third argument, which is injected after "READY.", so BASIC programs work as well.

Release 2

- STOP key support

Release 1

- 6502 core, fake PS/2 keyboard emulation (PS/2 data bytes appear at VIA#1 PB) and text mode Vera emulation
- KERNAL/BASIC modified for memory layout, missing VIC, Vera text mode and PS/2 keyboard