

Notes on using the X16 Edit ROM version

December 8, 2021

1 Introduction

This document describes the steps required to install and use the X16 Edit ROM version:

- Prepare a custom ROM image
- Create startup code using the X16 Edit API

The methods described herein are tested on an emulator compiled from the GitHub master branch, what is expected to become version R39.

2 Prepare a custom ROM image

The first task is to prepare a custom ROM image that may be mounted in the emulator.

You got the standard ROM image when you downloaded the emulator. It's the file rom.bin, normally in the same folder as the emulator.

The standard ROM image uses banks 0–6 as follows:

0. KERNAL
1. Keyboard layout tables
2. CBM-DOS
3. GEOS KERNAL
4. BASIC interpreter
5. Machine language monitor
6. Charset data
7. Unused

The most user-friendly option would be to distribute a complete custom ROM image with banks 0–6 from the standard rom.bin, and with X16 Edit ROM code in bank 7.

The X16 Kernal is, however, not free software, and I am probably not allowed to distribute it in its original form or in an amended form. Consequently, the X16 Edit ROM version contains only the X16 Edit code (one bank, 16 kB). To get a usable ROM image, you need to append the X16 Edit ROM code to the standard rom.bin file.

In Linux and MacOS this may be done with the cat utility as follows:

```
cat rom.bin x16edit-rom-x.x.x.bin > customrom.bin
```

If you're using Windows, and cat is not available, you may try using the type command instead (not tested):

```
type rom.bin x16edit-rom-x.x.x.bin > customrom.bin
```

Select the custom rom on starting the emulator:

```
xemu -rom customrom.bin
```

3 Create startup code using the X16 Edit API

Running the X16 Edit ROM version requires that you enter a small startup routine in RAM.

The startup code must use the public API of the program, which currently consists of the two entry points described below.

3.1 Default entry point

Description:	The program's default entry point, opens the editor with an empty new buffer
Call address:	\$C000
Params:	.X = First RAM bank used by the program .Y = Last RAM bank used by the program

Example startup code entered into the builtin monitor:

```
.A 9E00 LDA $01          ;Get current ROM bank...
.A 9E03 PHA              ;... and store it on the stack
.A 9E04 LDA #$07         ;Store bank 7...
.A 9E06 STA $01          ;...in the ROM select
.A 9E09 LDX #$01         ;Set first RAM bank used to 1
.A 9E0B LDY #$FF        ;Set last RAM bank used to 255
```

```

.A 9E0D JSR $C000      ;Call editor entry point
.A 9E10 PLA            ;Retrieve original ROM bank...
.A 9E11 STA $01        ;...and write it to ROM select
.A 9E14 RTS

```

3.2 Load text file entry point

Description: Loads a text file into the editor on startup
Call address: \$C003
Params: .X = First RAM bank used by the program
.Y = Last RAM bank used by the program
R0 (\$02-\$03) = Pointer to file name, least significant byte first
R1 (\$04) = File name length

Example startup code entered into the builtin monitor:

```

.A 9E00 LDA $01        ;Get current ROM bank...
.A 9E03 PHA            ;... and store it on the stack
.A 9E04 LDA #$07        ;Store bank 7...
.A 9E06 STA $01        ;...in the ROM select
.A 9E09 LDX #$01        ;Set first RAM bank used to 1
.A 9E0B LDY #$FF        ;Set last RAM bank used to 255
.A 9E0D LDA #$23        ;Get filename AddressL...
.A 9E10 STA $02        ;...and store in $02
.A 9E12 LDA #$9E        ;Get filename AddressH...
.A 9E15 STA $03        ;...and store in $03
.A 9E17 LDA #$05        ;Get filename length...
.A 9E19 STA $04        ;...and store in $04
.A 9E1B JSR $C003      ;Call editor entry point
.A 9E1E PLA            ;Retrieve original ROM bank...
.A 9E1F STA $01        ;...and write it to ROM select
.A 9E22 RTS

.M 9E23 <FILE NAME DATA, 5 BYTES IN THIS EXAMPLE>

```

4 Final comments

The X16 Edit ROM version is currently a proof of concept, showing that the code is ready to be installed and run in ROM.

The installation and startup process is not very convenient for the end-user at this stage. The editor version is of most interest if you like to install your own custom ROM image or if the X16 project would like to officially include

the editor in the standard ROM. In the latter case, there could be a BASIC command to start the editor in a similar way that the built-in monitor is started.