



COLECOVISION  
MEGACART

# FAQ

**Version 1.02**

Last update : September 7<sup>th</sup> 2013

# Contents:

## **Section 1**

*Informations about the MegaCart*

## **Section 2**

*Using MegaCart with C*

## **Section 3**

*Using MegaCart with Assembler*

## **Section 4**

*Questions about MegaCart*

## **Section 5**

*Parts needed to assemble a MegaCart Pcb*

## SECTION 1

### *Informations about MegaCart*

The ColecoVision MegaCart works both as a standard 32K cartridge and an expanded bank-switching cartridge supporting up to 1MB.

The megacart has been developed and created by Bryan Edewaard in August 2005, but it's now a property of OPCODE Games since 2010

The MegaCart **do not** require any special device such as an Expansion Module  
It will run on stock ColecoVision system

If you want to use MegaCart for one of your project, contact OPCODE at : <http://www.opcodegames.com>

You can buy PCBs for your project(s)

The only thing asked if you plan to use the MegaCart, is you include the MegaCart logo somewhere on the game box, cartridge or manual

Bank switching on the megacart is just reading a specific address.  
SDCC doesn't have a good way to do a dummy read, so just read the address into a variable. Basically, you define the address as a volatile pointer to char (the volatile is important so the compiler doesn't optimize the seemingly useless read away).

Something like:

```
char dummyx = *((volatile char*)0xFFFF); // switch to bank 0
```

## SECTION 2

### *Using MegaCart with C*

Programming for the MegaCart in bank-switching mode:

The top 64 locations of the cartridge (CV FFC0h-FFFFh) are reserved for bank-switching. It is probably not useful to put any data in this area in your banks.

The selectable banks are 16K and appear at CV C000h-FFFFh (minus the reserved area).

The EPROM should be arranged starting with the first bank:

bank0 = EPROM 0000-3FFF  
bank1 = EPROM 4000-7FFF  
bank2 = EPROM 8000-BFFF  
and so on.

The first 16K of the cartridge (CV 8000h-BFFFh) is fixed and will always contain the highest/last 16K segment of the EPROM. Therefore, this data will appear twice (CV 8000h-BFFFh and C000h-FFFFh) if the last bank is selected.

It is recommended that the following addresses be read to select banks:

128K - 8 banks - FFF8h-FFFFh.  
256K - 16 banks - FFF0h-FFFFh.  
512K - 32 banks - FFE0h-FFFFh.  
1MB - 64 banks - FFC0h-FFFFh.

Remember that strobing FFFFh places the same data in both halves of the cartridge memory.

### SECTION 3

#### *Using MegaCart with Assembler*

For Princess Quest and Mecha-8 I'm using assembler, I've found useful the following definitions (TNIasm v0.44):

```
; For 128K Megacart (for 256K one I start on $fff0 upto SLOT_15 at $fff)
SLOT_0: equ $fff8
SLOT_1: equ $fff9
SLOT_2: equ $fffa
SLOT_3: equ $fffb
SLOT_4: equ $fffc
SLOT_5: equ $fffd
SLOT_6: equ $fffe
SLOT_7: equ $ffff

; At start of code
    ld hl,SLOT_0    ; Default slot of Megacart, but set again because user can do a Colecovision RESET
    call sel_slot

; My typical switching code
    ld hl,SLOT_3
    call sel_slot

; Slot selection
sel_slot:
    ld (slot),hl    ; Saves current slot, useful because in NMI routine sometimes I switch slots to play music from OTHER slot
    ld a,(hl)
    ret
;
```

In Princess Quest, I tend to keep all code for game in the most upper slot (always mapped in \$8000-\$BFFF) and some code/data in SLOT\_0 (default one, mapped at \$C000-\$FFBF), then code switches slots as needed to access other data (no code)

For Mecha-8 finally I've jumped the bridge and apart from basic SLOT\_7/SLOT\_0 containing 32K code/data, there is a separate intermediate slot were are contained various routines for bosses, SGM detection and so.

## SECTION 4

### *Questions about MegaCart*

#### ***How do I change banks?***

The last 64 locations will switch banks (FFC0-FFFF). If you have fewer than 64 banks, then the strobe addresses simply repeat where they end. If you have 16 banks, then bank 0 can be strobed at FFC0, FFD0, FFE0, or FFF0. I recommend using the highest set of addresses since some EEPROMs want their unused pins pulled high.

#### ***What memory ranges are banked?***

The bank switched area is C000-FFFF. The top 16K of the EPROM is always mapped to 8000-BFFF, so the top bank will never need to be selected, since it would simply duplicate the same 16K in 2 places.

#### ***Any limitations I should be aware of?***

The main thing is to use the uppermost set of address strobes (as stated above), since Eduardo discovered that some brands of EPROMs require it. The area from FFC0-FFFF can be read, but you'll be changing banks with each read, so you can say that each bank is 16K minus 64 bytes.

#### ***Why is there no 64K variation of the MegaCart?***

The PCB architecture doesn't really support it. If a game fits inside 64K, a 128K MegaCart will have to be used, which means a lot of available space on the MegaCart will remain unused.

#### ***What kind of equipment do I need to encode game data into the MegaCart?***

Any 40-pin EPROM burner should do the job nicely, as far as the ColecoVision software is concerned. It's really not that different from a regular ColecoVision PCB/EPROM setup.

#### ***I have no clue how to read specific area of a ROM to switch bank or origin the data in C. I don't know any other functions outside of coleco.h, getput.h and the cv\_programming.pdf.***

For generating the code, I used two things. First, use the Makefile to put the code into different segments:

```
bonus.o: bonus.c bonus.h
```

```
$(CC) $(CFLAGS) -c -mz80 --no-std-crt0 --codeseg bank2 --constseg bank2 bonus.c
```

"codeseg" names the segment to stick code in, and "constseg" names the segment to place constants in. Once you have all that, though, you need a modified linker sequence. I modified the SDCC linker to output the segment names in the hex file, then wrote my own final linker to read the hex file in and output a megacart.

Finally, a small modification to the crt0.s startup code to enforce which bank is switched in, because the Megacart powers up with an unpredictable bank loaded (BlueMSX does NOT emulate this). So if you have anything important in the top 16k, you want to set the right bank before you start C code. I only needed this because my main bank was all 32k, and that put all the C runtime functions in the switchable part. I would probably try to put them at the beginning in the future.

## SECTION 5

### *Parts needed to assemble a MegaCart Pcb*

To use as a standard 32K cartridge:

Install U1 and C1. Program a 27256/27C256 type EPROM with the game image and install in U3. Alternately, a 27128 can be used for 16K games.

To use as a bank-switching cartridge:

Install U1 and C1. Program a 20V8 type GAL (generic array logic) using file CVMC.JED and install in U2.

Select the EPROM based on the size of the game:

16K - 27128  
32K - 27C256 / 27256  
128K - 27C010  
256K - 27C020  
512K - 27C040  
1MB - 27C080

Program the EPROM and install in U4.

GAL CHIP IS NEEDED (Logic Chip) for PCB  
The chip is a 20V8 type

A PAL or GAL is a configurable logic chip. Without it, I'd need to have several smaller digital logic chips. The part's generic number is 20V8. I think the ones on the board I sent you are from Lattice, but I've also used Atmel ATF20V8B. The best price I found was from [www.jameco.com](http://www.jameco.com).

Suggested parts:

R1 4.7K resistor network: Digi-Key 4608-1-472LF-ND  
C1 0.1uF capacitor: Digi-Key BC1148CT-ND  
U1 74HCT21: Digi-Key 296-9262-5-ND  
U2 GAL20V8: Jameco <http://www.jameco.com/webapp/wcs/stores/se...productId=38658>  
U3,U4 Most brands of EPROMs should work, although I haven't tested to see what access time is required. I would probably buy speeds of 300-350ns or lower.

To assemble your own boards, you will need a device programmer for the GAL20V8 and the selected EPROM.

Stuffing guide:

Standard cartridge: C1, U1, U3

Bank-switching cartridge: C1, R1, U1, U2, U4

**FOR NON PROFITS USE ONLY**

If you have any question(s) about the MegaCart, please visit :  
<http://atariage.com/forums/topic/216354-colecovision-megacart-faq/>

MegaCart FAQ Created & Maintained by CollectorVision  
© 2013 CollectorVision. All rights reserved

MegaCart™ & MegaCart logo are registered trademarks of OPCODE Games.  
© 2005, 2013 OPCODE Games. All rights reserved