

Versa64Cart

User Guide

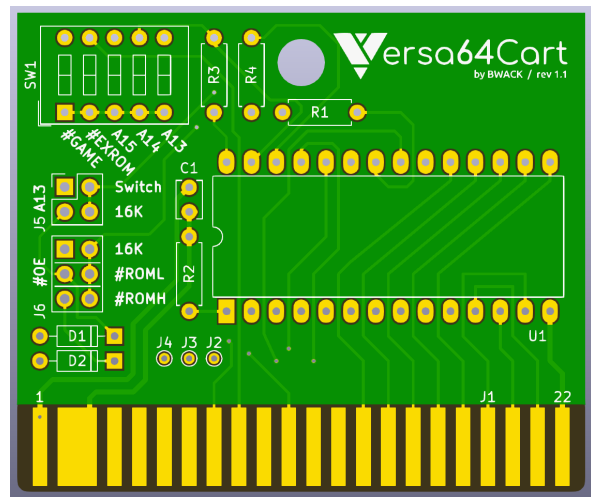
Introduction

Versa64Cart is designed to be a modern easy-to-use cartridge development board for Commodore 64 and Commodore 128 hobbyists and hackers. The versatile design of the PCB allows for flexible implementation of various cartridge types. Whether it's a simple cartridge backup of an 8k game or a switchable multi-program diagnostic cartridge, Vers64Cart has you covered.

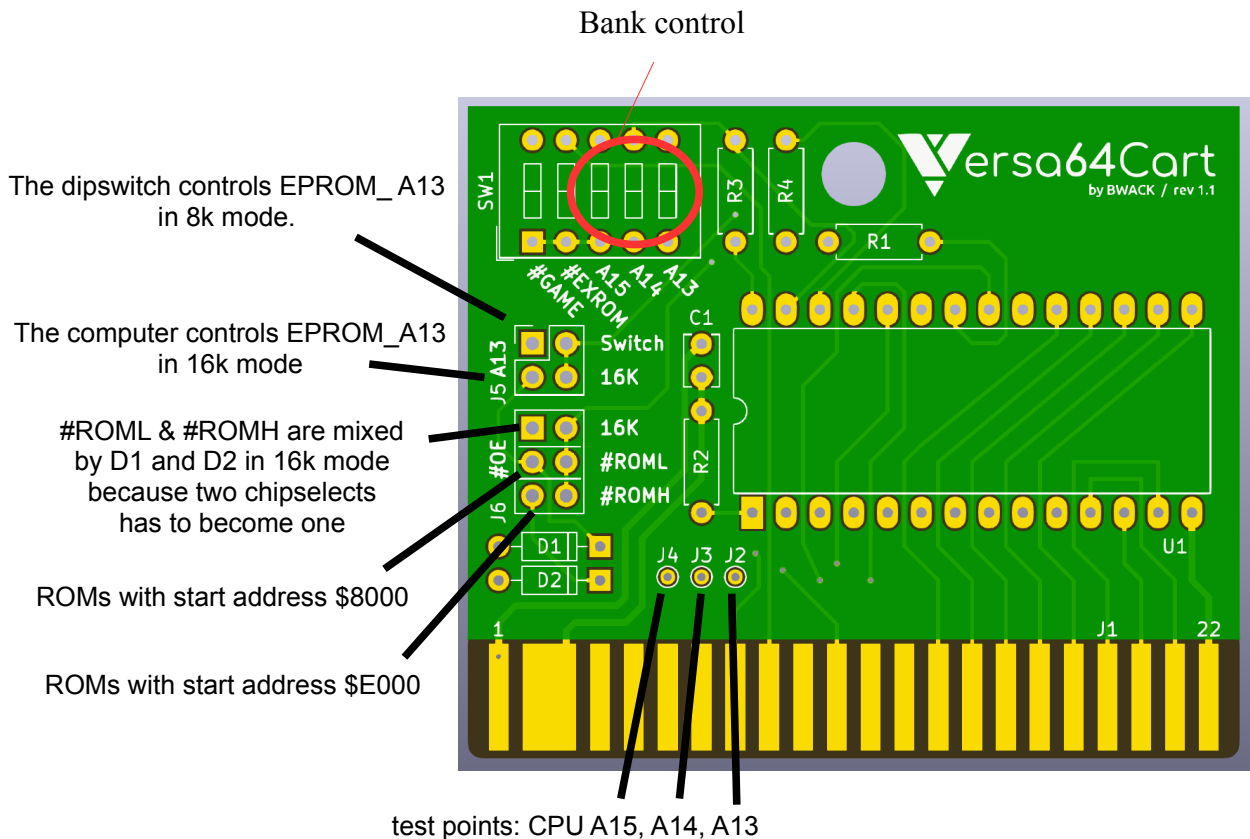
Overview

- Supported EPROMS:
27C64 (8k), 27C128 (16k),
27C256 (32k), 27C512 (64k)
- Ability to access multiple
program images by using optional on-board DIP switches(*)
- Rapid low-cost production of clean, single image cartridges via simple
solder pad bridging option
- PCB designed to fit inside any standard C64 cartridge shell, including
centre post versions
- Rounded corners to allow for easier cartridge insertion and reduced wear
to PCB

(*) CAUTION: Only make DIP switch changes when computer is power off. Never change DIP switch settings while cartridge is inserted and computer is powered on.



Jumpers and dipswitches



Technical note: The Versa64Cart makes it possible to run 2x8k (16k) ROM games on a single 16k EPROM device. Two chipselects has to become one, and the two 8k images has to be banked with A13.

Banking is easy, we let the computer control EPROM_A13. A jumper is put on J5:16k. The jumper now disconnects dipswitch A13 and reconnects #ROML to EPROM_A13. We have banking! The two chipselects are #ROMH and #ROML and are mixed with D1 and D2. A jumper on J6:16k routes the mixed signal to the EPROM_#OE.

Instead of #ROML on EPROM_A13 we could have used CPU_A13. It would have a better timing, because #ROML is a product of A13 from the PLA, but would it work in ultimax mode? In ultimax mode the two 8k banks are not placed adjacent (\$8000-9FFF and \$E000-\$FFFF).

Setting #GAME and #EXROM

Mapping the EPROM into memory is done with the #EXROM and #GAME jumpers. There are 4 modes:

8k: #EXROM=0, #GAME=1 : start \$8000-\$9FFF #ROML
8k ultimax: #EXROM=1, #GAME=0 : start \$E000-\$FFFF #ROMH
16k: #EXROM=0, #GAME=0 : start \$8000-\$9FFF (#ROML) + \$A000-\$BFFF (#ROMH)
16k ultimax: #EXROM=1, #GAME=0 : start \$8000-\$9FFF (#ROML) + \$E000-\$FFFF (#ROMH)

You can find the mode of a .CRT file using WinVICE's cartconv.exe.

```
C:\WinVICE-2.4-x86>cartconv -f Wizard_of_Wor.crt
CRT Version: 1.0
Name: Wizard of Wor
Hardware ID: 0 (Generic Cartridge)
Mode: exrom: 0 game: 0 (16k Game)

offset  sig  type  bank start size  chunklen
$000040 CHIP ROM   #000 $8000 $4000 $4010
```

Finding the mode for Wizard of Wor in a CRT file.

#EXROM and #GAME are routed to the PLA and the PLA is the address decoder inside the C64. The PLA controls and drives the ROML and ROMH signals used to put the EPROM onto the data bus.

Converting .CRT to .BIN

The EPROM must be programmed with a binary file. Use WinVICE's cartconv.exe tool to convert the .CRT file.

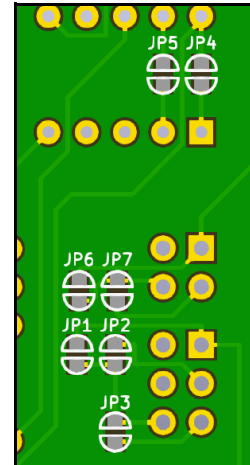
```
C:\WinVICE-2.4-x86>cartconv -i Wizard_of_Wor.crt -o wizardofwor.bin
Input file : Wizard_of_Wor.crt
Output file : wizardofwor.bin
Conversion from Generic Cartridge .crt to binary format successful.
```

Converting CRT to BIN for burning to EPROM.

Solder Jumpers

You don't need to attach jumpers and dipswitches if you just want a single game or a dead test cart. You can use solder jumpers instead.

solder jumper settings	#EXROM	#GAME	switch	16K	16K	#ROML	#ROMH
mode	JP5	JP4	JP6	JP7	JP1	JP2	JP3
8k	X		X			X	
8k ultimax		X	X				X
16k	X	X		X	X		
16k ultimax		X		X	X		



A Multi-Cart Example

An example with 16k + 8k + 8k ROM images on a 27C256 (32k) EPROM.

Jumpan Junior: 16k, #EXROM=0, #GAME=0

Dead Test: 8k ultimax, #EXROM=1, #GAME=0, start=\$E000 (#ROMH)

Jupiter Lander: 8k, #EXROM=0, #GAME=1, start \$8000 (#ROML)

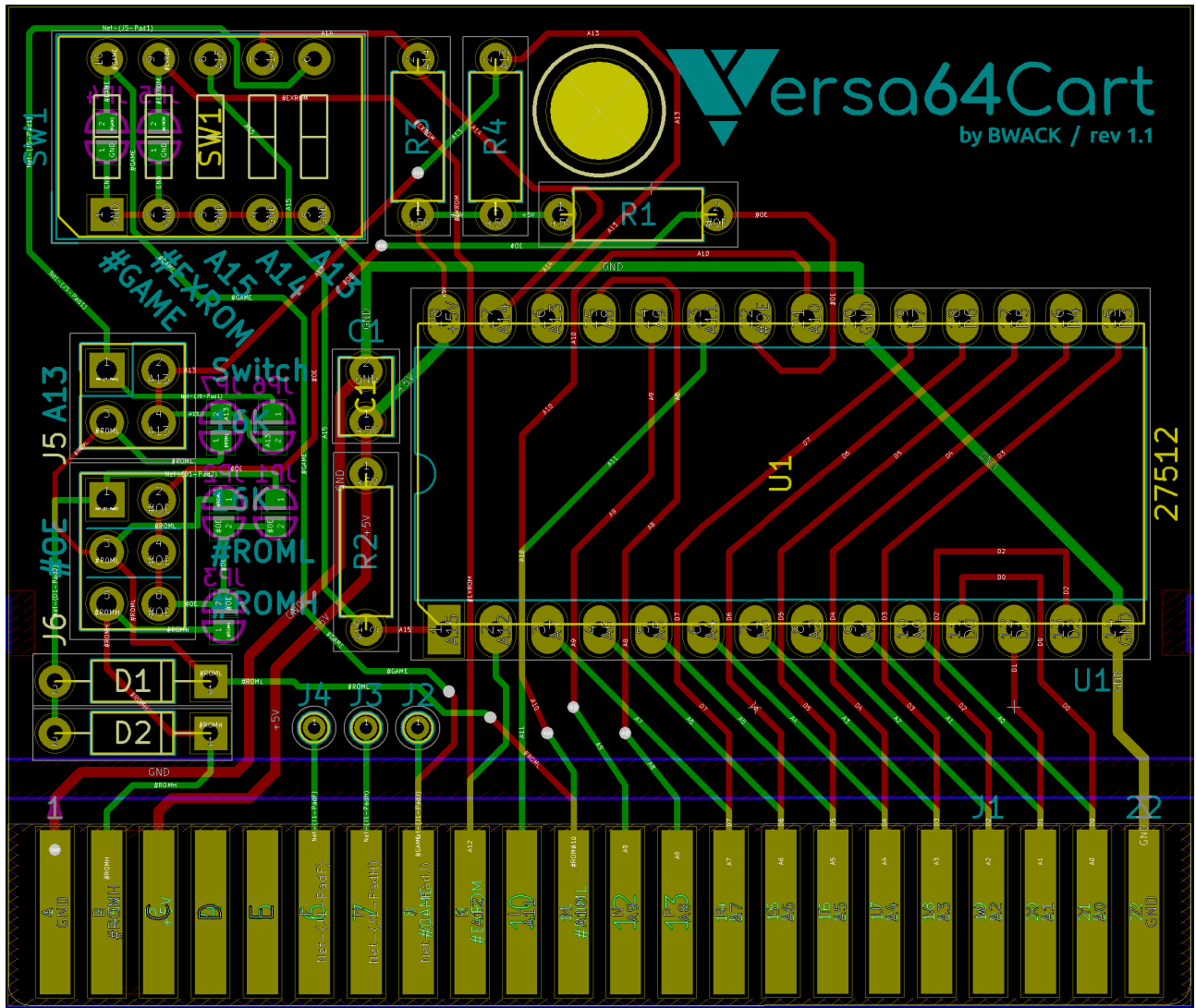
memory area	ROM image	dipswitch & jumpers
\$0000-\$1FFF	Jumpan Junior	↑↑x↑x, J5:16k, J6:16k
\$2000-\$3FFF		
\$4000-\$5FFF	Dead Test	↓↑x↓↑, J5:Switch, J6:#ROMH
\$6000-\$7FFF	Jupiter Lander	↓↑x↓↓, J5:Switch, J6:#ROML

dipswitch positions: ↑=0V, ↓=5V, x=don't care

now you should write down the settings because you will forget them.

Bill of Materials (BOM)

Board Layout



A screen shot taken in KiCAD Pcbnew

