

# Versa64Cart

## User Guide

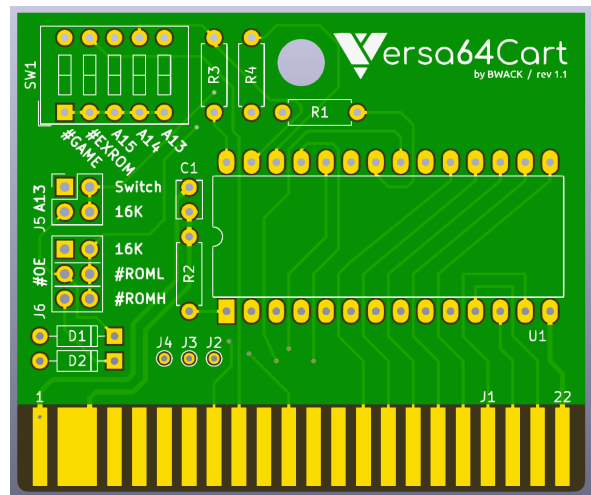
### Introduction

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, Versa64Cart has you covered.

### Overview

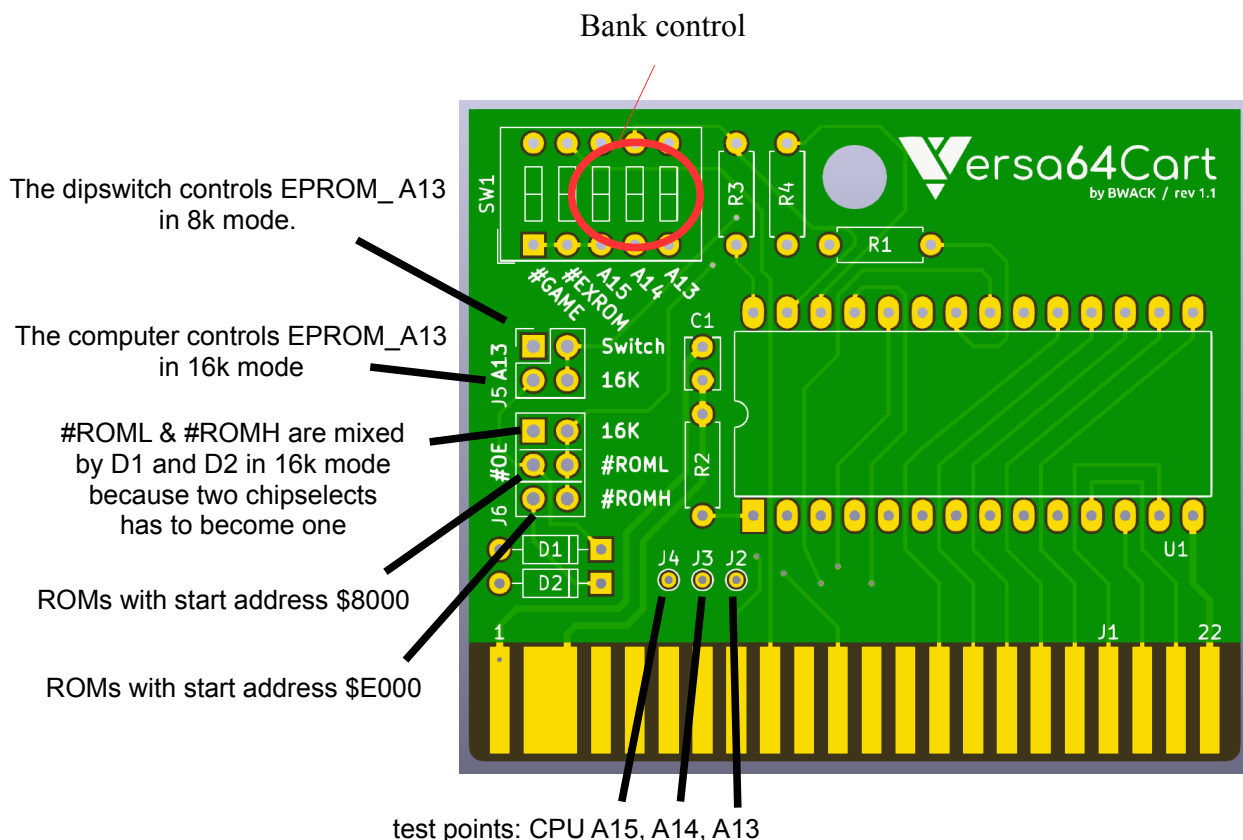
- Supported EPROMS:  
27C64 (8k), 27C128 (16k),  
27C256 (32k), 27C512 (64k)
- Ability to access multiple program images by using optional on-board DIP switches(\*)
- Rapid low-cost production of clean, single image cartridges via simple solder pad bridging option
- PCB designed to fit inside any standard 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.



## The board

### *Jumpers and DIP switches*



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.

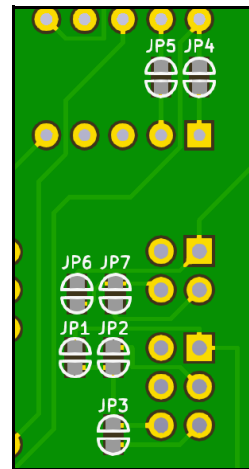
Banking is easy! We let the computer control EPROM\_A13. A jumper is put on J5:16k. The jumper now disconnects DIP switch A13 and reconnects #ROML to EPROM\_A13. We have banking! The two chipselects are #ROMH and #ROML and are mixed with D1 and D2. A jumper on J6:16k routes the mixed signal to the EPROM\_#OE.

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).

## Solder Jumpers

You don't need to attach jumpers and DIP switches 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		



## The ROM image

### 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.

## **Examples**

### ***Dead Test Cart example***

You must program a 27C64 EPROM with the dead test image. The Dead Test cartridge is in 8k ultimax mode (Please see "Setting #GAME and #EXROM"), and effectively replaces the C64 Kernel ROM. Solder in the programmed EPROM, and put a solder blob on JP4, JP6, and JP3 to configure as shown in the table above. Solder in the four 22k resistors. Solder in a 100nF cap on C1. Note: If you use 27C128/256/512, and you have burned the 8k image in the beginning of the EPROM, you must connect A15, A14 and A13 to GND on SW1. You can do that by soldering three wires from top to bottom of SW1 where the silkscreen says A15 A14 and A13.

### ***Multi-Cart example***

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

Jumpan Junior: 16k, #EXROM=0, #GAME=0, start=\$8000  
Dead Test: 8k ultimax, #EXROM=1, #GAME=0, start=\$E000 (#ROMH)  
Jupiter Lander: Same as Dead Test. Usually it is not ultimax though.

memory area	ROM image	DIP switches & 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:#ROMH

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

Solder in the programmed EPROM, the DIP switches, the 2x2 and 2x3 2.56mm pinheader, two 1n4148 on D1 and D2, the four 22k resistors, and a 100nF cap on C1. Use jumpers on J5 and J6.

This is the same example I have used in my Versa64Cart YouTube video. Now you should write down the settings because you will forget them. A 27C512 ROM will give you twice as much space. Eight 8k games, four 16k games or a mix of this.

## **Other notes**

Kernal ROMs on the cartridge does not work.

## **Board Manufacturing**

The board thickness is 1.6mm. The board is 58.0 x 48.9 mm. PCB prototyping houses take gerber files. Some companies can handle .kicad\_pcb files directly. Creating gerbers is beyond the scope of this document, however I have made a video about it on my [YouTube Channel](#):

- [KiCAD Gerber files and DirtyPCBs](#)

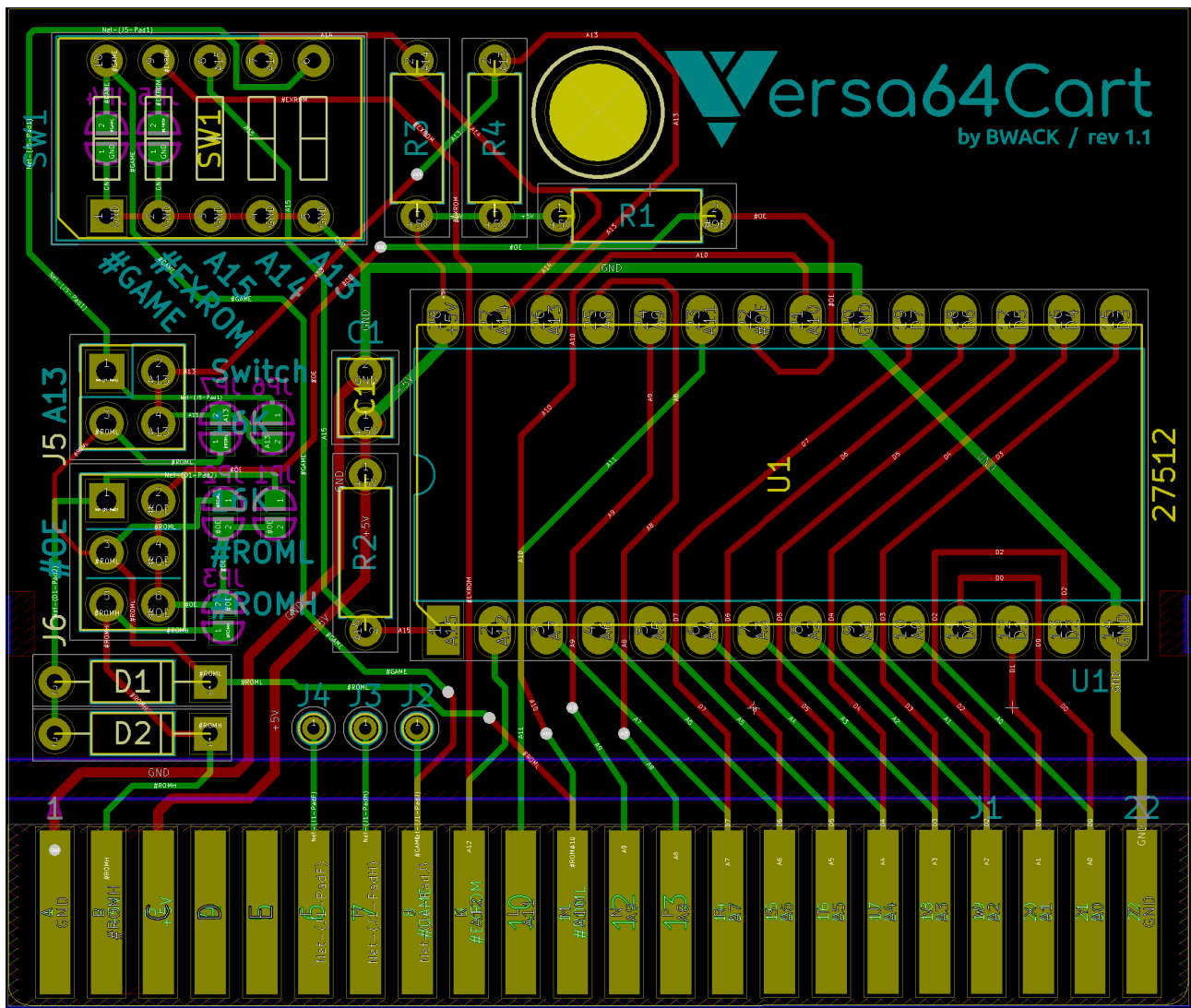
## **Enclosures**

The board is based on original cartridges and enclosures. The board width is 58mm. It fits original cartridge enclosures and Jim Brain's ones too. It does not fit stumpy cart. The standoffs are 1mm narrower. That can be tweaked. You can't use a ic-socket because then the eeprom hits the top.

## **Acknowledgements**

I would first of all like to thank MindFlare Retro for helping me out with this document. Also a thank you to Jim Brain for the Simple64Cart and its eagle design files. MindFlare converted the eagle .brd file to kicad. I took the cartridge outline (document layer) and imported it in my own design. From there it was super easy to design on top of that drawing. Also a big thanks to my viewers on YouTube for feedback on this project.

## PCBnew Screenshot



Zoom on the PDF to see the signal names on the traces.

## Schematics

