

How Cartridges Work

Daniel Bingamon
Batavia, Ohio

Have you ever wondered how a Commodore 64 cartridge starts up a game automatically without you giving any commands? The way it is done is fairly simple if you know a little machine language. Here we are going to show you how to use this cartridge control to make the computer start up with different colors.

When you start up your computer or reset it, it must have a place to go on start. This starting point is determined by the RESET vector. Upon reset the 6510 loads the program counter (a place in the microprocessor the computer uses to know where it is at) with the address that is in the reset vector. The 6510 "knows" that the reset vector is stored at \$FFFC and \$FFFD in ROM. Its contents is address \$FCE2.

The microprocessor transfers execution to \$FCE2, the RESET Routine. One of the first tasks performed is a subroutine at \$FD02 that tests for the presence of a cartridge. This test compares the characters 'CBM80' to the contents of locations \$8004 through \$8008. If 'CBM80' is present and the Most Significant Bit is set to one in the letters 'CBM', then the KERNAL ROM does an INDIRECT jump through a location \$8000 (32768 in decimal). Locations \$8000 and \$8001 are the first two bytes of the cartridge and are used to store the address of the start of the cartridge program to give the cartridge control.

The next two bytes, \$8002 and \$8003, are the NMI Reset address. This is the address the 64 will jump to if RUN/STOP-RESTORE is pressed, but again, only if 'CBM80' is present at \$8004.

Cartridge Simulation

Because there is RAM at location \$8000 (unlike the VIC-20 which is normally blank in its cartridge space), we can fool the C64 into thinking it has a cartridge plugged in. When there is a real cartridge in the computer, wiring is provided to turn off the RAM and connect the cartridge instead. This program will not work if you have a cartridge in the computer.

If you look at this program with a machine language monitor, notice the start-up code 'CBM80' at \$8004. It must show exactly like it is listed here in order to gain control of the computer:

```
.. 8000 09 80 2f 80 c3 c2 cd 38
.. 8008 30 Start Of Program. . .
```

The space marked 'Start Of Program' is where the cartridge program starts. In general, a cartridge program could start anywhere in the cartridge, but the program in our simulated cartridge will start here. Now when the computer is reset, it will go to location \$8009 and execute that program. There better be some-

thing there or you will probably get a non-recoverable crash, and turning off the computer will fix things - a reset button may not get you out of this one.

The Start Of Program begins with some initialization routines just like the ROM reset routine. First, the screen initialization at \$FF81 puts the screen in the right place. After that is done a call is made to \$FF84 which initializes I/O devices. Some cartridges never use this, but I find things run much smoother when you do use it. The next thing this program does is execute the routine called 'CLRCHN' to clear I/O channels and its address is \$FFCC. After all this is done I store the new colors to the graphics chip and do an indirect jump through \$A000 which starts up BASIC (ie. power up message, bytes free, etc.). However, this jump would be replaced by the cartridge program which would continue on from there. With this technique you can test your cartridge program in RAM, which later you can have 'burned in' on cartridge.

Also included in this program is a section that sets the NMI vector to change your screen color to the new default upon hitting the restore key and continue running any BASIC programs without stopping them, although the run/stop key by itself still works.

Listed with this article is the BASIC loader program that pokes this machine language into memory. You will be notified by the program if you have mistyped any DATA. To make this machine language to take effect type 'SYS64738' and upon restart you will notice something different. Please SAVE this program before typing the SYS statement.

Use SUPERMON or any other equivalent monitor to follow the flow of this program. Please type 'POKE 56,128:CLR' to protect the program from strings writing over it.

Program Listing:

```
5 printchr$(147) : print " cartridge simulator "
6 print : print " by:daniel bingamon
10 for i=32768 to 32838
20 read a : b = a + b : poke i,a
40 next
50 if b<>7693 then print " data error " : end
100 data 9, 128, 47, 128, 195, 194, 205
110 data 56, 48, 32, 129, 255, 32, 132
120 data 255, 32, 138, 255, 32, 204, 255
130 data 169, 18, 32, 56, 128, 169, 46
140 data 141, 24, 3, 169, 128, 141, 25
150 data 3, 108, 0, 160, 72, 152, 72
160 data 138, 72, 32, 204, 255, 32, 56
170 data 128, 104, 170, 104, 168, 104, 64
180 data 169, 11, 141, 32, 208, 141, 33
190 data 208, 169, 0, 141, 134, 2, 96
200 print:print " sys64738 will begin simulation "
```