

Introducing the VCC SDC simulator DLL

The Vcc SDC Dll is an attempt to simulate Darren Atkinson's SDC floppy drive emulator. It is not an emulator but is a simulation that mimics some of, but not all of, the features of Darren's emulator. The simulator is reverse engineered from information found in the CoCoSDC UserGuide and other sources as well as from tests of software designed to work with the SDC.

The simulator is a work in progress and some sdc features have been left out. Most of these are features used to create and manage sdcard files. These functions are necessary on a real Coco3 but are more conveniently managed using the Windows OS that Vcc runs on. Some implimented features may not work properly, these are usually not intentional but are due to a mis-understanding of how a read SDC works.

Some of the games found on the 'Coco SDC Image.zip' appear to be incompatible with the simulator. Some of these games are likely intended for Coco1 and Coco2, which may explain why they don't run on a Coco3. Also some games seem to overwrite portions of the SDC-ROM code and thus fail to load additionally needed virtual floppies.

Since I do not currently own a SDC I do not know if others are a problem with the simulation or a problem with the game running with a real SDC.

Floppy ports and port isolation

The SDC uses the floppy ports. The same ports are used to communicate with the floppy controller. If both the SDC and a floppy interface are installed in a Multi Pak Interface the SCS pack select signal must be used to choose which controller sees the floppy ports. SDC software relies on cart select to isolate the ports from the floppy controller.

An enhancement to Vcc's mpi.dll was required to selectively control port isolation. It is recommended the FD502.dll be in mpi slot 4 and SDC.dll be in mpi slot 3 with slot 3 selected. The Disable Cart Select checkbox should not be checked.

Virtual Disk Images

The VCC SDC emulator supports JVC and VDK images. It does not support SDF images, or any formats that contain per sector check sums or track specific information that might be found on real hardware. SDC.dll expects image files to contain a contiguous array of 256 or 512 byte sectors preceded by an optional header of less than 256 bytes. If a header is recognized it is examined to determine the number of sides on the virtual disk. If there is no header the number of sides is inferred from the image size.

Becker Interface

For performance reasons most emulators, including VCC, use a pair of unused floppy ports (the Becker interface) to communicate with a drivewire server via a TCPIP socket connection. Because VCC uses a separate MPI slot for becker.dll these ports can not be isolated like the other floppy ports. The Multi Pak DLL has modifications to allow these ports even with the SCS signal enabled.

However, SDC-DOS is designed to work with a CoCo and that uses the bitbanger to connect to a drivewire server. This means it does not interface to the Becker drivewire on Vcc. A custom SDC-DOS could be modified to use the Becker interface. This work is best done by those with experience in such modification.

Flash Banks

SDC.dll maintains a list of rom files (banks) that can be loaded using sdc flash control command. The content of a rom file is loaded into sdc's cart rom when selected. The roms can be accessed using the SDC-DOS 'RUN@n' command.

The bank active at startup can be selected from the SDC config menu. Normal SDC operation requires that the startup bank (default zero) contains the SDC-DOS rom. If this bank is left empty SDC.dll will automatically load SDC-DOS.rom from the VCC execution directory.

Using RUN@ to switch to disk11.rom will start the rom but will not set the MPI slot select, a poke is required after DECB starts up to allow floppy operations. The address to poke is 65407 (\$FF7F) and the value depends which MPI slot FD502.DLL is in. Poke 0 for slot 1, 17 for slot 2, 34 for slot 2, and 51 for slot 4.

The currently selected MPI slot can be seen on the VCC status line. It is displayed as MPI:x,x where x,x are digits showing slot numbers. The first digit shows which slot contains the ROM being run, the second digit shows which slot gets floppy control signals. The above pokes set both digits to the same slot.

SDC Explorer

SDC-DOS assumes a self centering joystick is being used. This means that "mouse" is not a good option for joystick emulation. It might be better to use "keyboard" or avoid using joystick mode if you do not have a WASD or self centering controller. Also if VCC is overclocked SDCX will be difficult to control.

It has been discovered that some games can damage the virtual disk SDCEXP.DSK if it is mounted in drive 0. Once SDC Explorer is configured the way you like it is recommended that SDCEXP.DSK be set to read only on the system, or to keep a backup handy.

Cloud9 clock

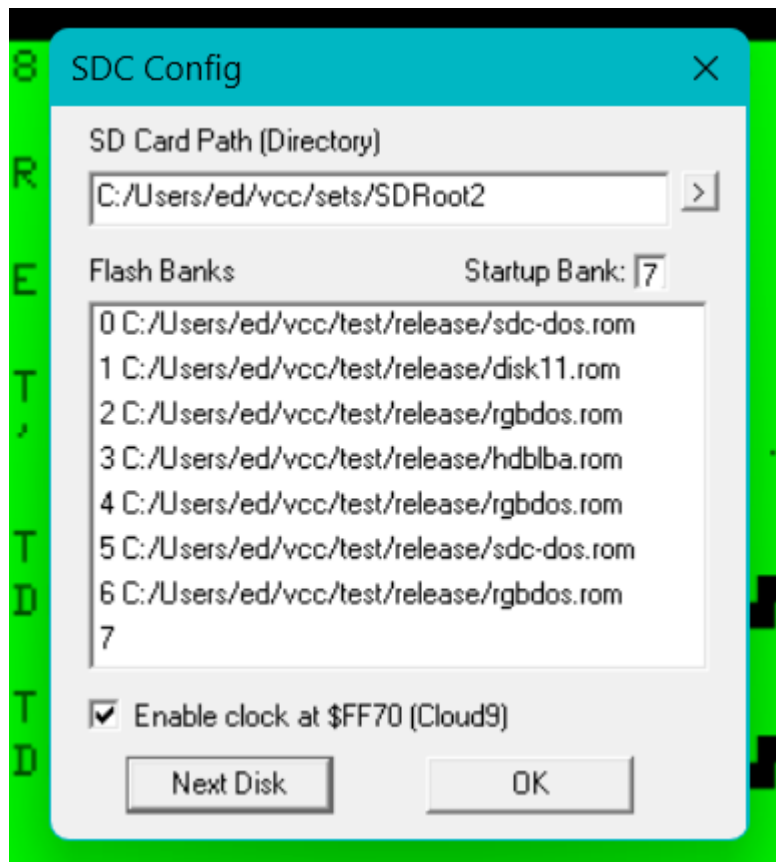
SDC.dll contains a Cloud9 RTC clock emulation. Software that can access this clock can use it to get the date and time.

Initial setup to try it out.

The setup I have used to test with is as follows.

1. In Cartridge/MPI config place FD502.DLL in slot 4, SDC.DLL in slot 3. Select slot 3 and make sure "Disable Cart Select Signal" check box is NOT checked.
2. Download a Coco SDC Image.zip from the Coco Archive. I have been using the 2025-04-30 version. Unzip the contents into a directory of your choice.
3. If you intend to use SDC Explorer's "RESTORE LAST SESSION" feature (set by default) in that same directory create an empty SDCX.CFG file to store session information. If "RESTORE LAST SESSION" is checked in the explorer's Config and this file does not exist the explorer will display an "SDC NOT READY" error and image loading will not work properly.
4. In Cartridge Cartridge/SDC config int the SD Card Path box enter or locate the path to the directory created. Leave everything else blank for now. Hard Reset to to start the simulation. (F9 twice or shift F5)
5. Type EXP to start the explorer if it does not auto start. If you hold the 'C' key down while it the explorer starts the configure screen comes up. Once the explorer is configured the way you want it is recommended that the SDCEXP.DSK be write protected at the system level (Set the file ReadOnly in Windows) to prevent accidental damage to it.

Configuration menu



The SDC Config menu is reached by selecting Cartridge/SDC Config.

SD Card Path: At the top you can enter the SD card path. This can be any Windows directory that you have write access to or it could be an actual SD card mounted on windows. If you click on the '>' button a dialog comes up that allows you to select the directory.

Flash Banks: This is a list of the rom files that simulate the SDC flash bank. Double clicking on one brings up a dialog that lets you select the rom. The bank used at VCC startup or hard reset can be set in the Startup Bank box. If SDC.dll is selected in MPI then the startup bank should contain SDC-DOS. If the startup bank is left empty the simulator (SDC.dll) will try to load SDC-DOS.rom from VCC's execution directory.

Enable Clock: This check box enables the Cloud9 clock. Replace the clock2 driver with clock2_cloud9 in OS9boot to use it. Alternately you can use the disto Clock in the fd502.dll. The Disto clock is subject to floppy bank isolation but should still work with Nitros9 *most of the time*.

Next Disk: The "Next Disk" button on the bottom of the config dialog mounts the next disk in a set to drive 0. This differs from the real SDC somewhat. If a disk is mounted using a wildcard or a partial match on the name Vcc's SDC simulator will get a list of matching disk images and will mount each in succession as "Next Disk" is pressed. Note that games that autostart when LOADM is used (eg Contras) do not work with the next disk button.

The "OK" button saves the settings to the Vcc ini file and exits the config dialog.

Known Bugs

nitros9 EOU does not appear to work with 6809 CPU

Some games do not work from SDC, most Sundog games do not work but most do with Vcc alone.

Loading games in sets in SDC Explorer will not work.