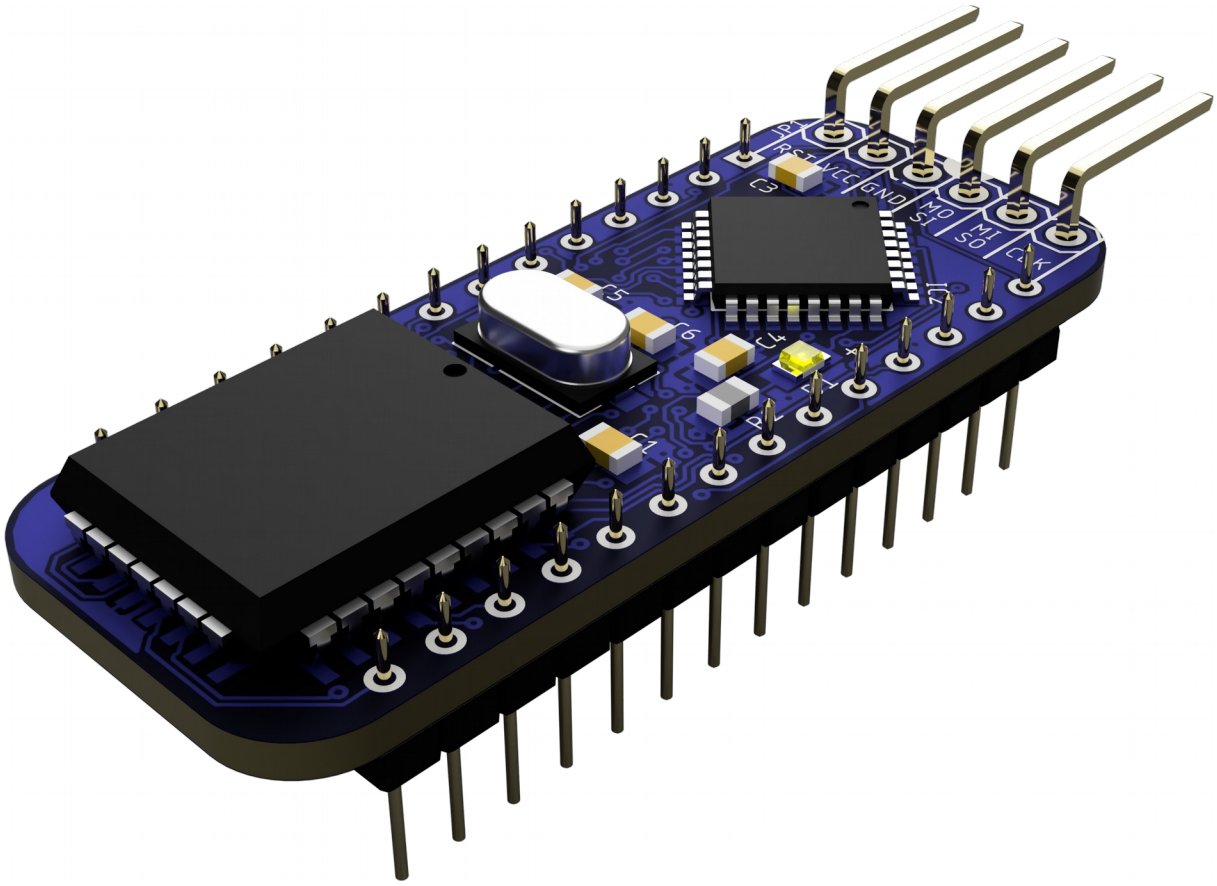
Switchless Multi-ROM for CBM 1541-II, 1571 & 1581 User's Guide



This microcontroller-based switchless ROM-switcher for the 1541-II, 1571 and 1581 disk drives lets you switch drive ROM between stock CBM DOS and JiffyDOS or other DOS version using simple basic commands. It's based on the switchless multi-ROM for 27128/27256 ROMs. Read more about it at: https://github.com/RetroNynjah/Switchless-Multi-ROM-for-27128-27256

Usage

The switcher is listening to the data bus for a predefined word and a number. The easiest way to send the word to the drive is to issue a load command to the drive from basic with the word as filename or to utilize a JiffyDOS wedge command.
The default magic word is #@RNROM where # is the number of the ROM image you want to switch to.
LOAD "1@RNROM",8 switches drive 8 to DOS image 1 and **LOAD "2@RNROM",8** switches to image 2 and so on. The default valid range of numbers is 1-4.
The drive will respond with a file not found error but the programmer will pick up the command, blink the multi-ROM LED to indicate which image was selected, switch the ROM by setting the high address pins accordingly and reset the drive using the newly selected ROM.

The selected ROM will be remembered and used at power-on.

If you have a JiffyDOS kernal in your computer you can select the right device using CTRL+D and issue the following commands instead: **@1@RNROM** or **@2@RNROM**

Theory of operation

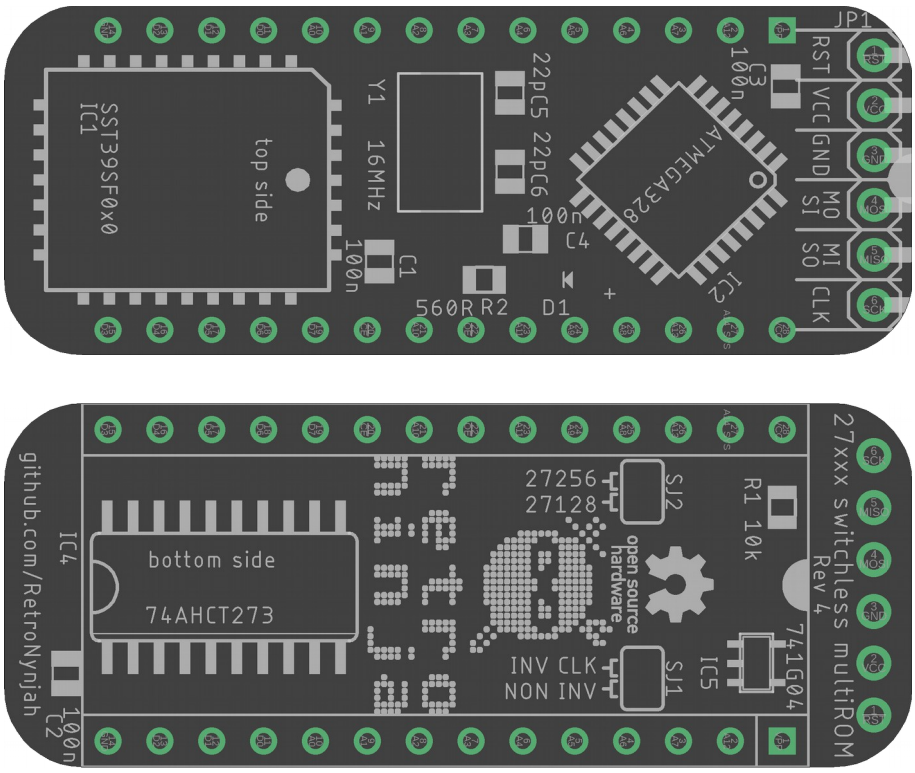
At power on, the microcontroller in the ROM switcher reads address 0 from its internal EPROM to find out which ROM it should select, sets the address A14-A16 or A15-A16 on the flash chip depending on ROM size and then resets the drive to make sure it's booted with the correct image. This can be noticed when the drive is powered on and the drive initializes and stops spinning and then spins up again briefly during the switch reset.
It then starts listening for a switch command.

Even at a clock speed of 20 MHz, reading and analyzing a 1-2MHz data bus can be a challenge so the data on the 1541-II data bus is buffered in the switch by a 74AHCT273 flip-flop that is clocked by the Write signal on the 6502 MPU in the disk drive. By clocking the data with the R/W signal we filter out irrelevant bus traffic and the flip-flop holds the data until the next write cycle. This gives the microcontroller enough time to read the data from the flip-flop.
The microcontroller listens for a predefined sequence of bytes *in reverse order* (MORNR@) and once the sequence is found, it treats the following numeric character as the target ROM image number. The reverse order of the characters on the bus is why the image number must be specified at the beginning of the command.
The microcontroller sets address pins A14-A16 or A15-A16 on the flash chip according to image selection and it stores the ROM number at address 0 of its internal EPROM for use at next power-on. It then resets the drive by toggling the IEC reset line.

Building the circuit

The switcher is small and a bit difficult to solder but it can be hand soldered. The reason for the small size is space constraints in many disk drives such as the 1541-II, 1571 and 1581.

The Flash chip is the trickiest part to solder and I recommend reflow soldering of the top side using stencil and solder paste. Then solder the bottom side by hand and finally solder the pin headers.



Parts

The switcher is designed for the SST39SF010A flash chip but the bigger SST39SF020/SST39SF040 could be used too if no SST39SF010A is available. Remember to duplicate the flash contents to fill all of flash to prevent floating pins from switching to empty flash areas.

The microcontroller can be pretty much any variation of ATmega48/88/168/328 as long as it can run at 20MHz.

The flip-flop needs to be a 74**AHCT**273. LS/HCT/AHC or any other variant will most likely not work.

The 74AHCT1G04 single gate inverter is needed for inverting the active low write signal from the 6502 to the active high clock input of the flip-flop.

The recommended crystal is a 20 MHz crystal (for 1571/1581) but a 16MHz works for 1541-II.

The LED on the board is optional but flashes a number of times indicating the number of the selected ROM when switching. It is only there for debugging reasons.

The pin headers that goes into the ROM socket should have round machined pins or be of that long flat stackable type that is used for the pass-through pin headers on Arduino shields. The flat ones are recommended as the profile gets even lower with those headers. There is almost no headroom in the 1541-II, 1571 and 1581 external drives. The internal drives in the 128D models have more room. Square pin headers destroy the ROM socket in the drive so don't use those.

Programming the microcontroller

The code for the controller was created using the Arduino environment and the device can be programmed in circuit from a sketch in the Arduino IDE using an ISP programmer.
I recommend the DIY AVR cores that can be installed using the Arduino Boards Manager (https://github.com/sleemanj/optiboot).

If you can't or don't want to use Arduino there are some precompiled hex files for the Kernal switch and along with fuse configurations and syntax examples for how to program it using avrdude. These can be found under the Applications folder in my GitHub repository at https://github.com/RetroNynjah/Switchless-Multi-ROM-for-27128-27256

How to use an Arduino as ISP or how to use an ISP programmer such as a USBASP will not covered by this document.
The firmware can also be programmed using AVRDUDE and the pre-compiled HEX file or using the popular TL866II programmer. The pin-out of the programming pins on the PCB matches the pin-out of the TL866II ICSP connector so a straight 6-pin cable can be used.

Programming the flash chip

Due to space constraints, there are no on-board programming headers for the flash chip so the flash chip must be programmed in a flash/EPROM programmer before soldering it to the board or desoldered for reprogramming.
It could also be possible to make a programming adapter to put on top of the flash chip to program it while it's still soldered to the board but this is not tested.

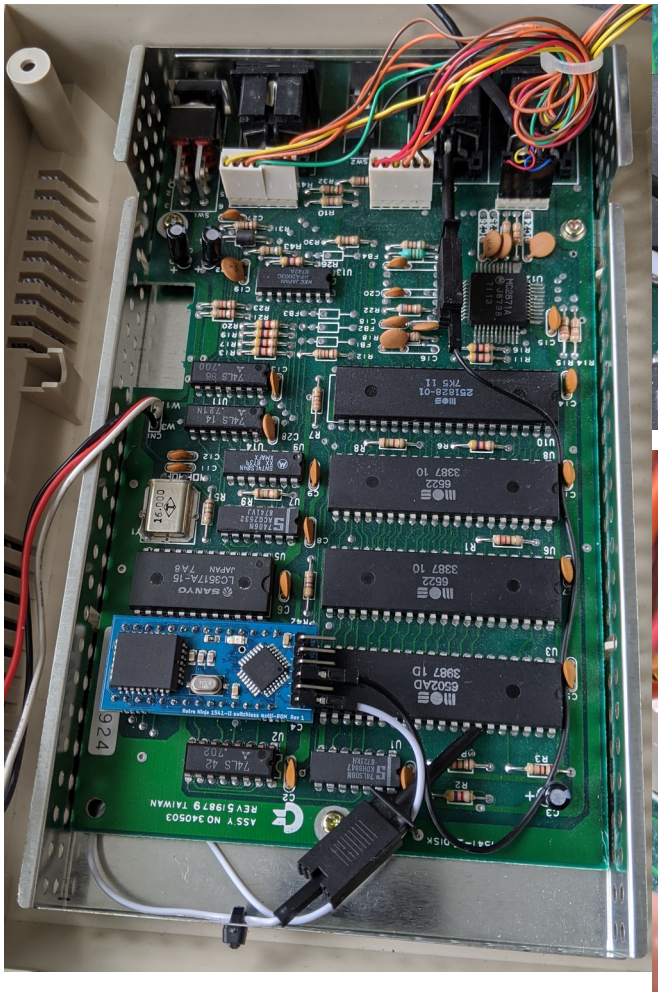
The ROM chip in the 1541-II is only 128 kbit (16 kilobyte) and the 1571/1581 drive ROMs are 256kbit (32 kilobyte) while the flash chip is 1 Mbit (128 kilobyte). It's highly recommended to fill the entire flash chip with working DOS images to prevent bricking of the drive by selecting a non-existing image. The switcher will not work without a working ROM in the drive that can receive the serial data and

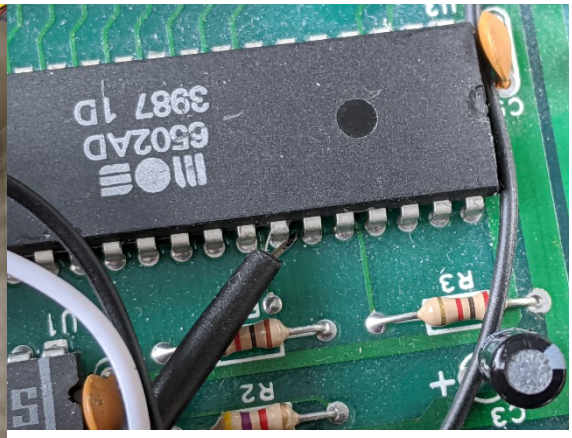
put it on the data bus.
If you just use two ROM images, fill the flash with the same two images over and over for safety.

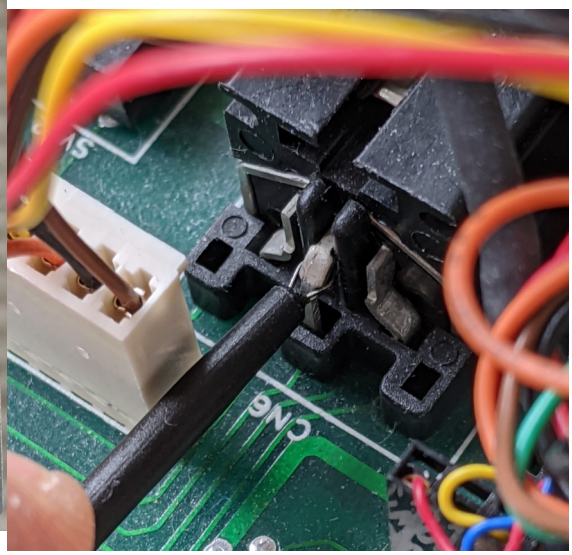
Installation

The circuit can be installed in the drive without soldering by using DuPont cables with test clips that attaches to the R/W pin on the 6502 MPU and to the reset pin on the back of any of the IEC connectors. However, it is recommended to solder the cables to the pins on the bottom of the drive PCB. That makes for a much more robust and transportable solution.
An even more stable solution is to solder the cables to the edge of the switcher PCB instead of using a pin header but that could make it harder to upgrade the firmware in the future.

The edge of the Multi-ROM PCB with the six ISP pins should face the notched end of the ROM socket (Pin 1).







The clock signal (CLK on the Multi-ROM) connects to the R/W pin (pin 34) on the 6502 MPU. MOSI on the Multi-ROM connects to IEC reset. That's the center pin on the back of the IEC connectors. The Multi-ROM works without a connection to IEC reset but a manual drive reset would then be needed after switching ROM.

The sketch and a pre-compiled firmware for the Multi-ROM can be found in the Applications directory in the repository at:
https://github.com/RetroNynjah/Switchless-Multi-ROM-for-27128-27256

Troubleshooting

If the device becomes bricked due to an incorrect DOS image, program the ATmega with a sketch that sets byte 0 to 0x00 or 0xFF in the internal EPROM. Then upload the original Multi-ROM sketch again and you should be back on track.
You can also set byte 0 of the EEPROM to 0x00 using AVRDUDE or using the Minipro software.

Switching will not work if you are some fast loaders such as Action Replay in your computer. Try switching the ROM without the fast loader. Switching ROM using JiffyDOS in the C64 is working fine though.

The switcher work in other CBM compatible drives that are using 27128 or 27256 ROMs. The Enhancer 2000 has been tested and works but unfortunately there's just not space enough in the OC-118 drive.

Change history

This is the change history for the user's guide.
For changes to the schematics, PCB or firmware, please refer to the GitHub repository at https://github.com/RetroNynjah/Switchless-Multi-ROM-for-27128-27256

Date	Version	Description
2021-05-15	1	Initial version
2021-05-22	1.1	Alternative ATmega MCUs
2023-01-10	1.2	Major update. Now supports 1571/1581. Updated recommendations for crystals and microcontrollers