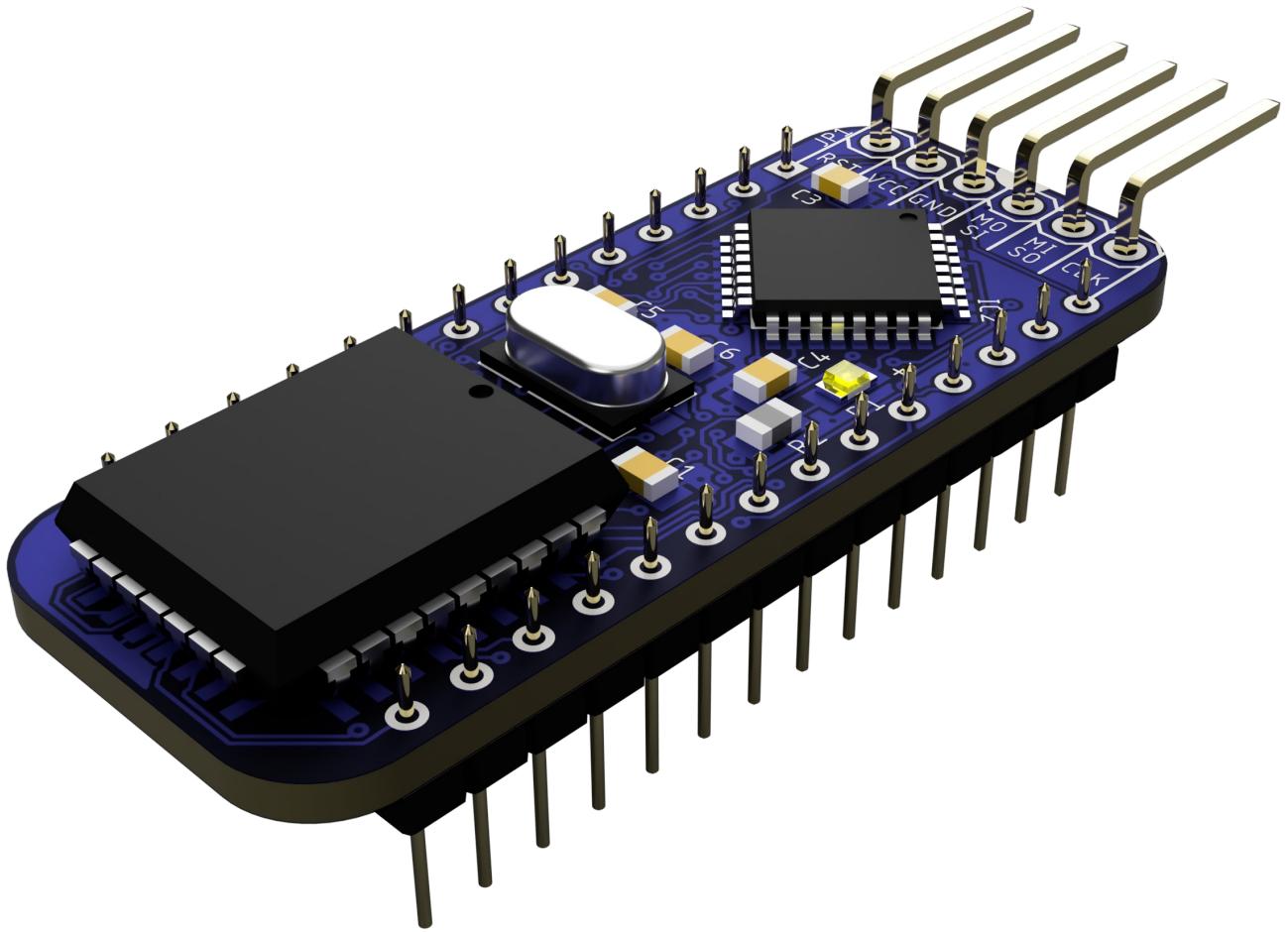


Retroninja Switchless Multi-Kernal for Commodore C16, C116 & Plus/4



User Guide

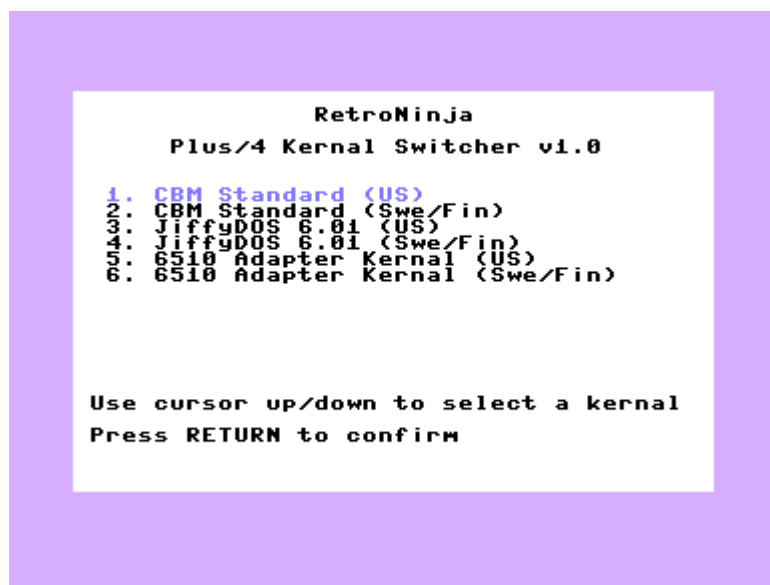
Table of Contents

Functional description.....	4
Installing the Kernal switch.....	5
Plus/4 PCB 310163.....	6
C16.....	7
C116.....	7
Using the Kernal switch.....	7
Upgrading the firmware.....	7
Programming the flash chip (ROM).....	7
Document revision history.....	7

Functional description

At power on, the MCU in the Kernal switch holds the reset line active while it reads address 0 from its internal EPROM to find out which Kernal it should select, sets the A14-A18 address pins on the flash chip accordingly and releases the reset line to make sure the computer is booted using the correct Kernal.

If at any time, reset is pressed and held for a few seconds the MCU will detect this and switch to the first Kernal in ROM (Kernal 0) which is a menu Kernal. For this application I created a custom Mini-Kernal for the C16/C116/Plus4 that displays a list of Kernals to choose from.



When the user selects a Kernal image from the menu, the Mini-Kernal writes a predefined switching command to RAM along with a byte that indicates the selected image number. It does this over and over. Example of the command: RNR0M+4#1 where 1 is a byte with value 0x01 that tells the switch that we want to switch to kernal image 1.

The Kernal switch now captures the command from the data bus, writes the selected kernal number to the MCU EEPROM for future use, then it switches the ROM address pins A13-A18 accordingly. Finally, it resets the computer using the new Kernal by pulling the reset line briefly. The computer will continue to start up using this new Kernal until a new choice is made.

The flash is populated with the Mini-Kernal followed by up to 10*16kB Kernals. It would in theory be possible to add up to 31 Kernals if both the Mini-Kernal and the MCU firmware was modified in some way to accommodate that many Kernal images.

Here's an example layout that fits in an SST39SF010A:

Mini-Kernal	CBM EN	CBM Swe	Jiffy En	Jiffy Swe	6510 En	6510 Swe	blank
16kB	16kB	16kB	16kB	16kB	16kB	16kB	16kB

Installing the Kernal switch

The switch will replace the kernal ROM chip (U24 on the Plus/4 or U4 on C16/C116).

Pay attention to the direction of the notch on the chip that indicates the location of pin 1. Pin one on the switch should face the same direction. Pin 1 on the switch is at the end that has a 6-pin header. It is also indicated by a printed notch on the bottom side of the switch.

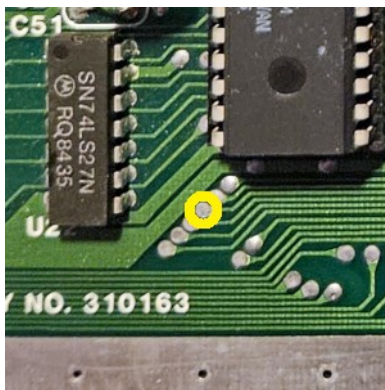
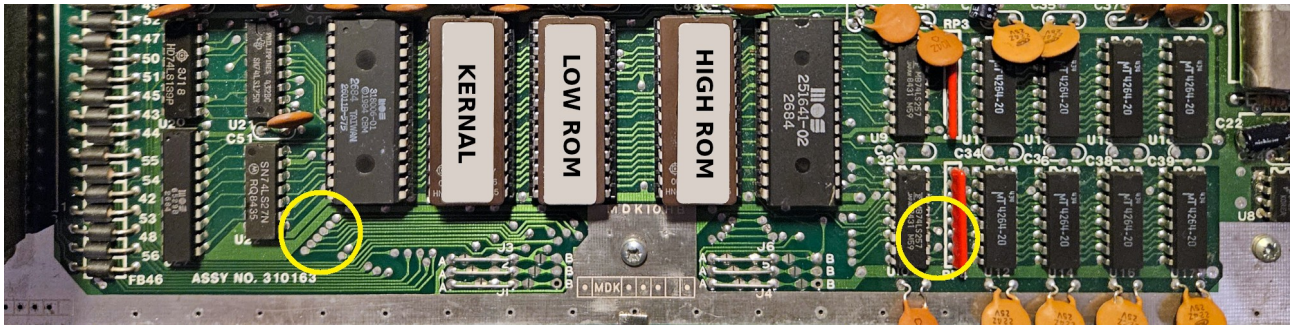
The below pins on the switch must be connected to the right signals on the motherboard.

Switch pin	Computer signal	Example of locations
CLK	R/W	7501/8501 pin 39, 6529 pin 1
MOSI	RESET	7501/8501 pin 40, 7551/8551 pin 4

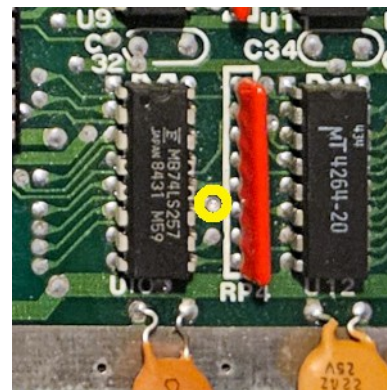
The connections can be made on the chips using test clips but it's recommended to solder pin headers to vias on the motherboard where available and connect the pin headers using Dupont wires or solder a cable to the motherboard for a permanent installation. If you want to solder pin headers to the vias you can try inserting them from the top side without taking the board out. Try heating the pins with the soldering iron while pushing them into the via using pliers but to get the best possible connection of the pins they should be soldered from the bottom of the motherboard.

The below pictures show examples of suitable vias for different motherboards.

Plus/4 PCB 310163



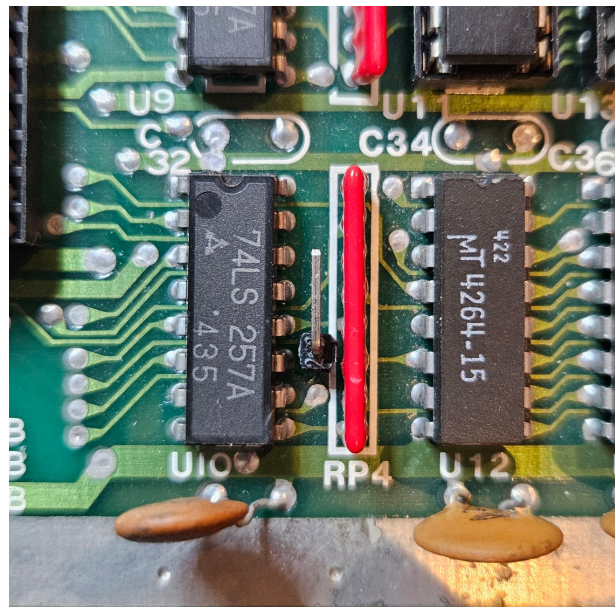
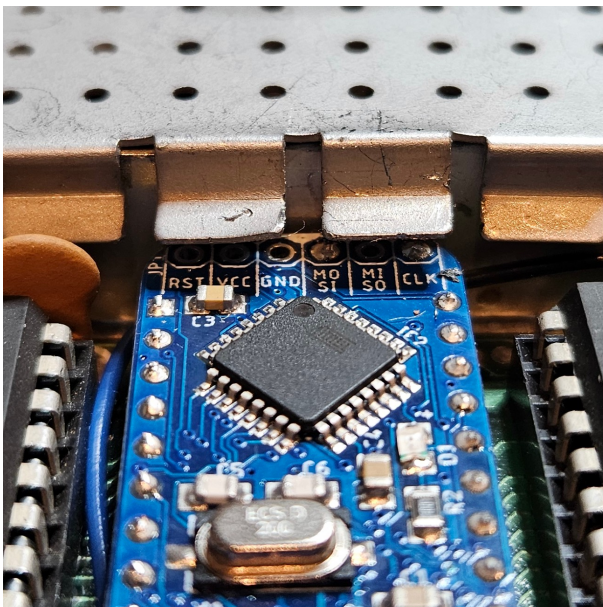
R/W



RESET

The ceramic disc capacitor C46 might be preventing the switch from being inserted properly. If it can't be bent out of the way enough to install the switch, it must be replaced by a smaller cap.

The connections on the switch should not be allowed to touch the lid of the TED can. If it does, bend the lid's fingers up and away from the switch PCB until none of them are touching the switch.



Due to the lack of headroom in the Plus/4, angled pin headers needs to be used on the motherboard.

C16

TBD

C116

TBD

Using the Kernal switch

Hold the reset key for about three seconds and then release it to activate the kernal menu. Once in menu you can use cursor up/down to move to the Kernal you want to switch to and press return. The switch resets the computer with the new Kernal and saves your choice for next power-on.

Upgrading the firmware

The firmware in the microcontroller can be upgraded using the 6 ISP pins. The source code is written for Arduino and programming the firmware can be done directly from the Arduino IDE using an ISP programmer.

If your device is using an Atmega328 it can be programmed as an Arduino UNO. The other variants require custom board definitions. I have used the Easy DIY Avr cores from <https://github.com/sleemanej/optiboot>. Board manager URL is

https://raw.githubusercontent.com/sleemanej/optiboot/master/dists/package_gogo_diy_atmega8_series_in_dex.json

If you can't or don't want to use Arduino I have some precompiled hex files for the Kernal switch and drive ROM switch along with fuse configurations and syntax examples for how to program them 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>

Programming the flash chip (ROM)

The flash chip needs to be programmed before soldering it. If you need to reprogram the flash later it must first be desoldered and then resoldered again after reprogramming.

The flash chip should contain the Mini-Kernal image followed by all switchable kernal images in correct order. All images needs to have a size of exactly 16kB.

Document revision history

1.0 Initial version

