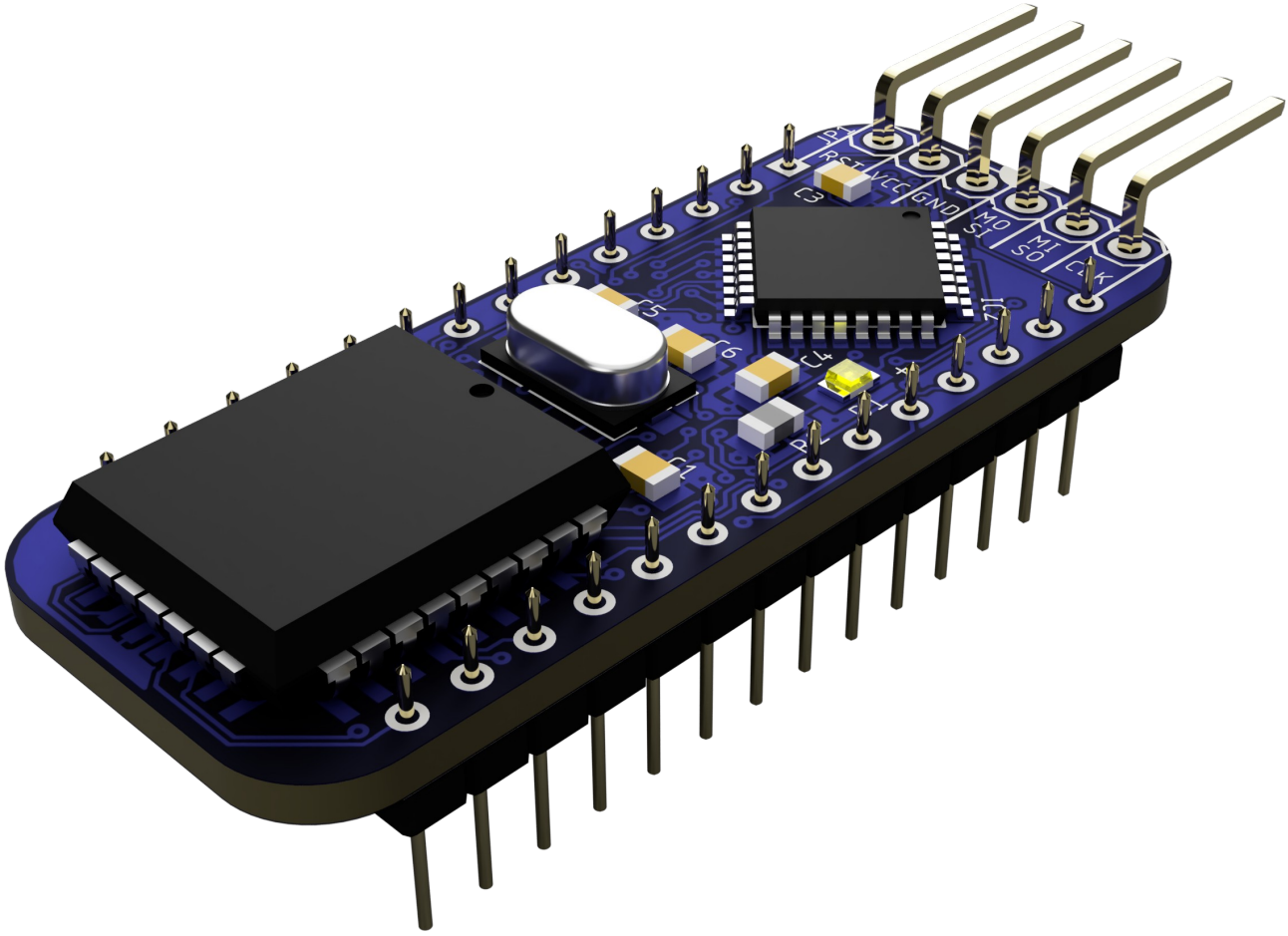
Retroninja Switchless Multi-Kernal for C64 longboard



User's Guide

Table of Contents

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, restore is pressed and held for a few seconds the MCU will detect this and switch to Kernal 0 which is a menu Kernal. For this application I created custom Mini-Kernals for the C64 and the C128 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: RNROM64#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 EPROM for future use, then it switches the address pins A14-A18 accordingly. It then resets the computer using the new Kernal by pulling the reset line. 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*16 KB 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 for a C64 layout that fills out an SST39SF010A:

Basic	Mini-Kernal	Basic	CBM Kernal	Basic	JiffyDOS US	Basic	JiffyDOS SE	Basic	JiffyDOS DK	Basic	JaffyDOS	Basic	DolphinDOS	Basic	SpeedDOS
8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB	8kB
16kB		16kB		16kB		16kB		16kB		16kB		16kB		16kB	

Example for C128 with an SST39SF010A:

Mini-Kernal	CBM Kernal	JiffyDOS US	JiffyDOS SE	JiffyDOS DK	Mini-Kernal	Mini-Kernal	Mini-Kernal
16kB	16kB	16kB	16kB	16kB	16kB	16kB	16kB

Example for C128CR/DCR with an SST39SF010A:

C64 Basic	64 Mini	128 Mini	C64 Basic	64 CBM krnl	128 CBM krnl	C64 Basic	64 Jiffy US	128 Jiffy US	C64 Basic	64 Jiffy SE	128 Jiffy SE
8kB	8kB		8kB	8kB		8kB	8kB		8kB	8kB	
16kB		16kB	16kB		16kB	16kB		16kB	16kB		16kB

As both C64 mode and C128 mode Kernals are in the same ROM a larger flash chip may be needed to accommodate all Kernals. If you have more C64 Kernals than C128 Kernals you still need to fill the C128 slots with working Kernals such as a stock CBM kernal or you will never be able to start the C128 into C64 mode either.

The C64 version of this Kernal switch only suits the C64 shortboards (PCB 250469) and the C64 part of the C128.
The C128CR/DCR doesn't have a separate ROM for the C64 part and doesn't need an separate switch for C64 mode but needs a special version of the switch.
C64 longboards have 24-pin 2364 type Kernal ROMs and I have another Multi-ROM version that is compatible with those at:
https://github.com/RetroNynjah/Switchless-Multi-ROM-for-2364

Installing the Kernal switch

The below pins on the switch must be connected to the right signals on the C64 motherboard.

Switch pin	Computer signal	Example of locations
CLK	R/W	8500 pin 38, 6526 pin 22
MISO	RESTORE	KBD connector pin 3
MOSI	RESET	8500 pin 40, 6526 pin 34

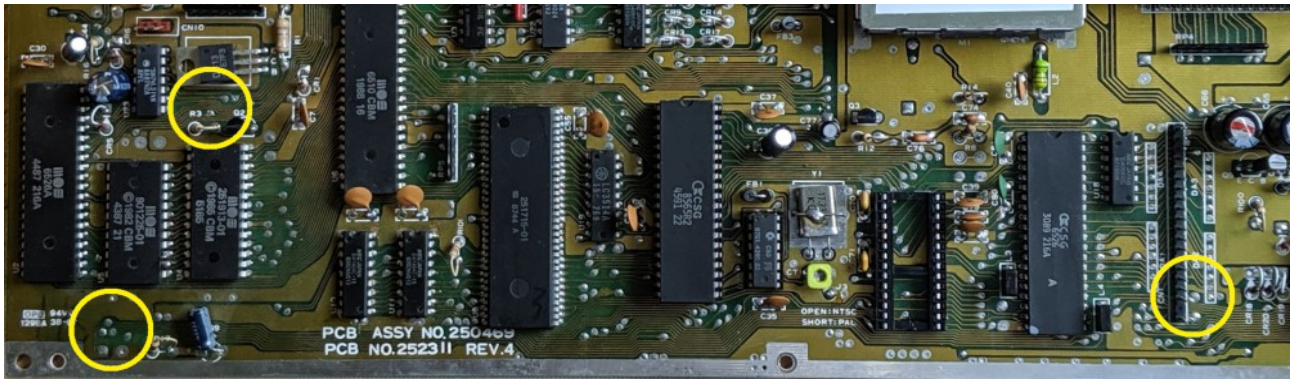
On the C128 motherboard the signals can be found at the below places.

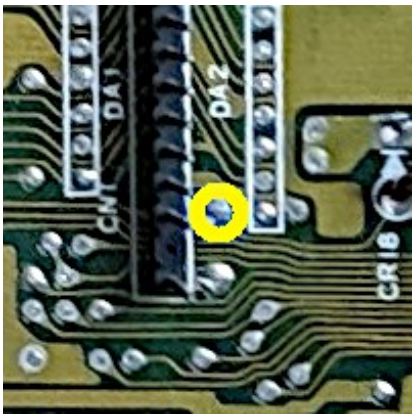
Switch pin	Computer signal	Example of locations
CLK	R/W	8502 pin 39, 6526 pin 22
MISO	RESTORE	KBD connector pin 3, U16 pin 9
MOSI	RESET	8502 pin 40, 6526 pin 34

The connections can be made on the chips using test clips but it's recommended to solder pin headers to vias on the motherboard and connect to the pin headers using Dupont wires or solder one end of Dupont cables directly to the vias 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 following pictures show examples of suitable vias for different motherboards.

C64 PCB 250469



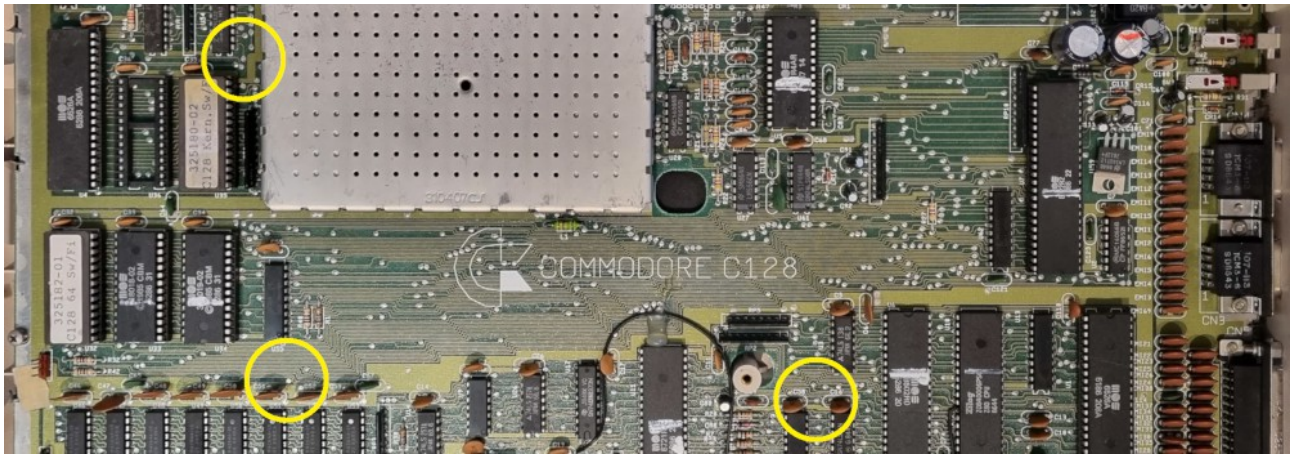


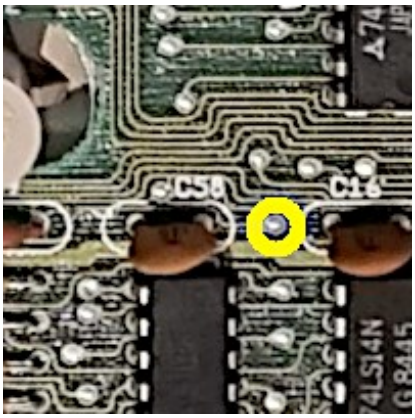
RESTORE

R/W

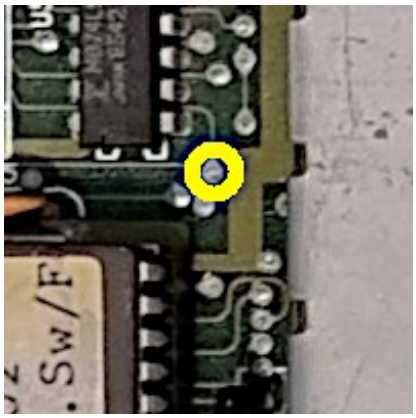
RESET

C128/C128D

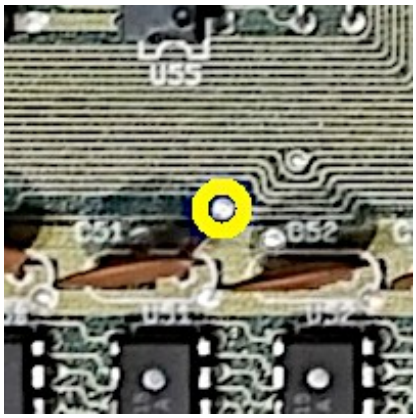




RESTORE



R/W



RESET

C128CR/DCR

To be added

Using the Kernal switch

Switching C64 and C128 Kernals

Hold the restore key for about three seconds until the kernal menu is activated. 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.

Switching C64 Kernal in a C128

When switching the C64 Kernal in a C128, hold the CBM and restore keys for about three seconds until the C64 kernal menu is activated. Keep holding the CBM key while selecting Kernal with the cursor up/down and return keys. While pressing return you should hold the CBM key until the reset is done to be taken to C64 mode. Otherwise you will end up in C128 mode after switching C64 Kernal and will have to do a GO64 or CBM+reset again.
The switch saves your choice for next power-on.
In the a C128CR/DCR

Upgrading the firmware

The firmware in the microcontroller can be upgraded using the 6-pin ISP header. 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 MiniCore from MCUdude (https://github.com/MCUdude/MiniCore) with default board settings.

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.

C64

In a C64 switch those 16KB are made up of 8KB Basic + 8KB Kernal.

C128/C128D

In the C128 the C128 kernal is 16KB and it does not have to be paired with a Basic ROM.
The C64 ROM in a C128 is the same as the one in a C64 so it needs basic+kernal pairs.

C128CR/C128DCR

In the C128CR/DCR the C128 kernal is 16KB and needs to be paired with the C64 Basic and Kernal. As the switch will switch both C64 and C128 kernal at the same time you will have to pay extra attention to how you mix the images to not get a confusing setup.