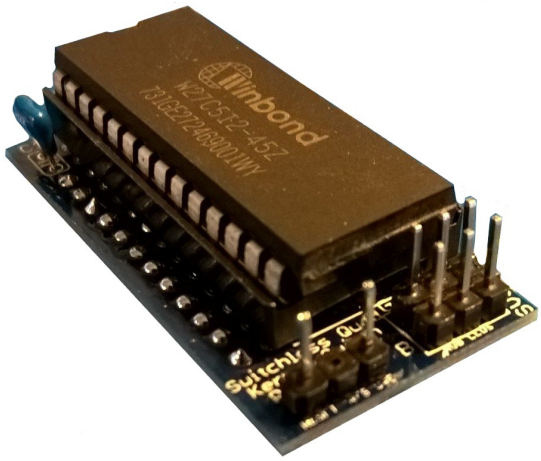
SKS64

Switchless Kernal Switcher

28 to 24 pin kernal ROM adapter and switcher for C64 longboard

User Guide

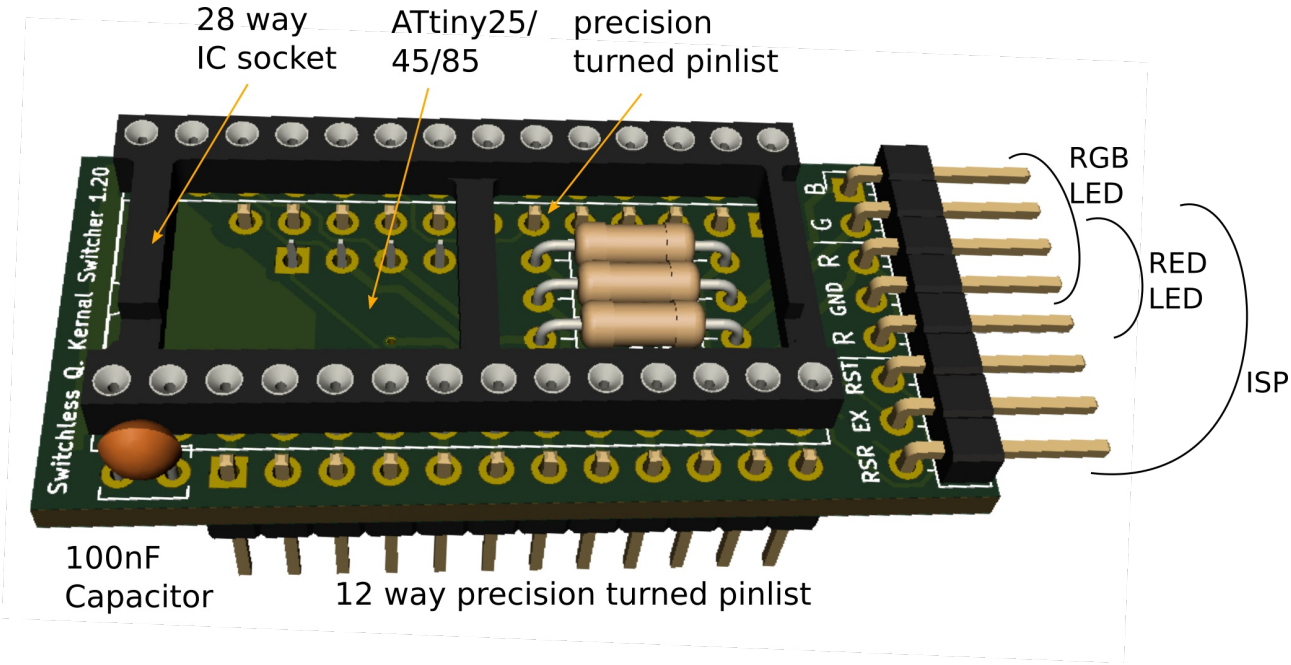


Introduction

Maybe you just need a 24-28 adapter, or you want four kernals in one chip. The Switchless Quad Kernal Switcher (SKS64) is just what you need.

There is also a C64C version, for people who are fortunate to have 250469 PCB boards. Note that some C64C looking cases may have C64 motherboards ! You have to open it up to see.

Board overview



Switching Kernals

The selection mode is entered by holding RESTORE for two seconds, and is indicated with a slight flash of RED on the RGB LED. Be quick and continue to tap RESTORE until the desired colour on the LED is shown. The computer reset after two seconds of no activity.

The timing can be changed in the source code:

```
#define PRESSTIME 20
#define MENUTIMEOUT 20
```

The RESTORE key still works as a normal for short presses. The C64 cannot read long RESTORE key presses anyway.

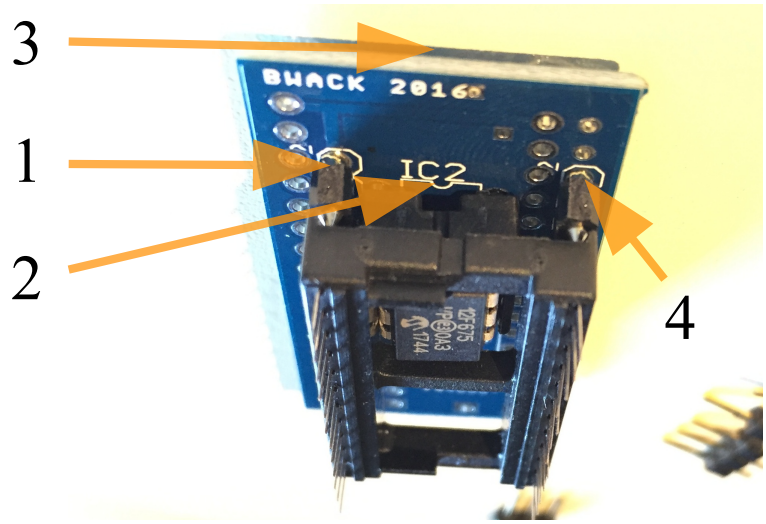
RESTORE as reset key

Hold the RESTORE key for two seconds then release. Wait two seconds for the reset.

Board Assembly

Because the board is so convoluted, the components need to be soldered in a special order to be able to reach all the solder pads. Using an IC socket for the ROM instead of precision turned socket rows is possible though time consuming, because you have to cut the solder pins flush that lay under the socket's footprint.

Soldering order:
- 1. rear left* pinlist
- 2. IC2 8pin IC Socket
- 3. IC1 28pin IC Socket
- 4. rear right pinlist
*** left as seen from bottom side.**



Here you see the board from the underside. I have inserted the IC socket that is later soldered into the C64 motherboard (longboard). 1. Start with the first 12-pin row on the left bottom side. 2. 8pin PDIP socket bottom side, 3. 28pin IC socket top-side. 4. Solder the last 12-pin row on the right bottom side. Tip! When soldering the last row of pins, attach the motherboard IC-socket (24pin) as shown in the image to align the last row of pins before soldering (no. 4).

Installation

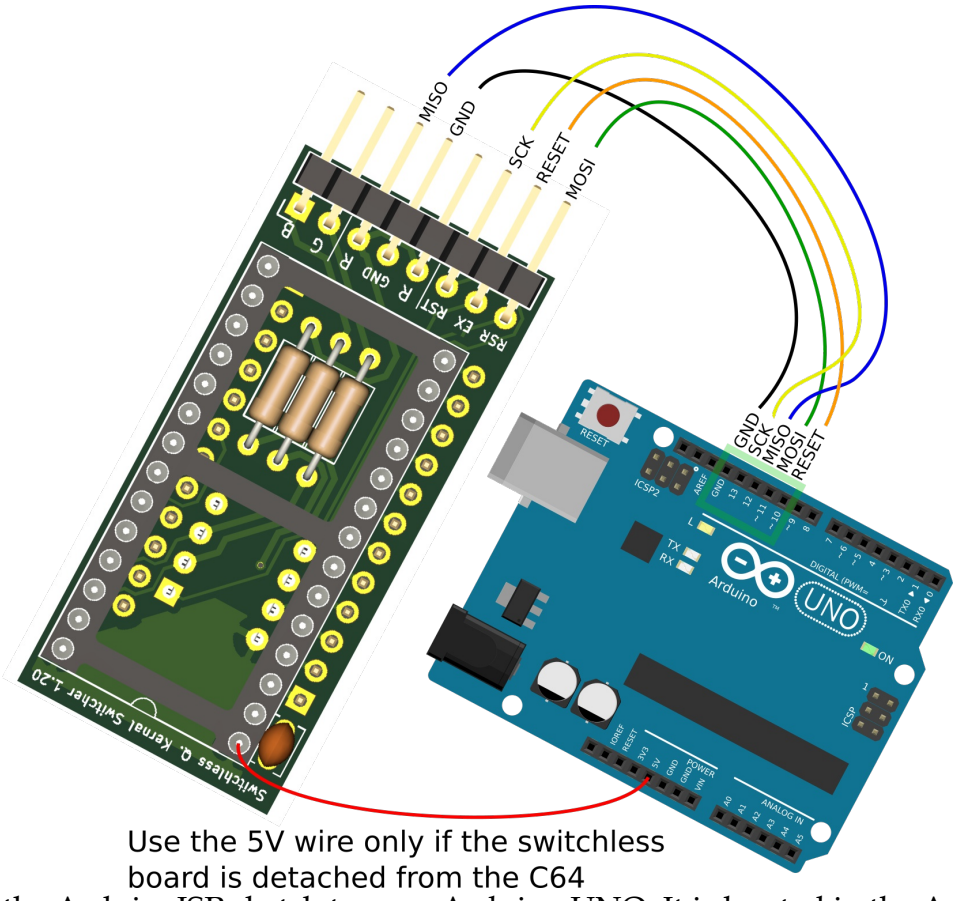
- Locations of the INTRES and RESTORE signals
- show images of different C64 motherboards and bubble-zoom closeups of where to attach intres and restore

TBD

Programming

In V1.20 the microcontroller is Attiny25/45/85. This enables Arduino development workflow. To work with the Arduino IDE you need to install the ATtinyCore libraries[1]. If you are not programming with the Arduino, you can load the hex file in for example Minipro and program it with the TL886.

Arduino as ISP



Upload the ArduinoISP sketch to your Arduino UNO. It is located in the Arduino IDE in the drop down menu File/Examples/11.ArduinoISP/ArduinoISP. Open the SKS64_SKETCH.ino file, and select: **Board** ATtiny25/45/85. **Processor** ATtiny85, and **Clock** Internal 1 MHz, **Programmer** Arduino as ISP. Connect all cables. If you power the SKS64 with Arduino 5V, you must not connect the SKS64 to the C64. Click **Burn Bootloader** ! This will write to the fuse bits, and there is no Arduino bootloader for the ATtiny family. Finally click **upload sketch**! The firmware gives a long flash and two short flashes on the RED LED if the fuse bits wasn't programmed.

USBasp as ISP

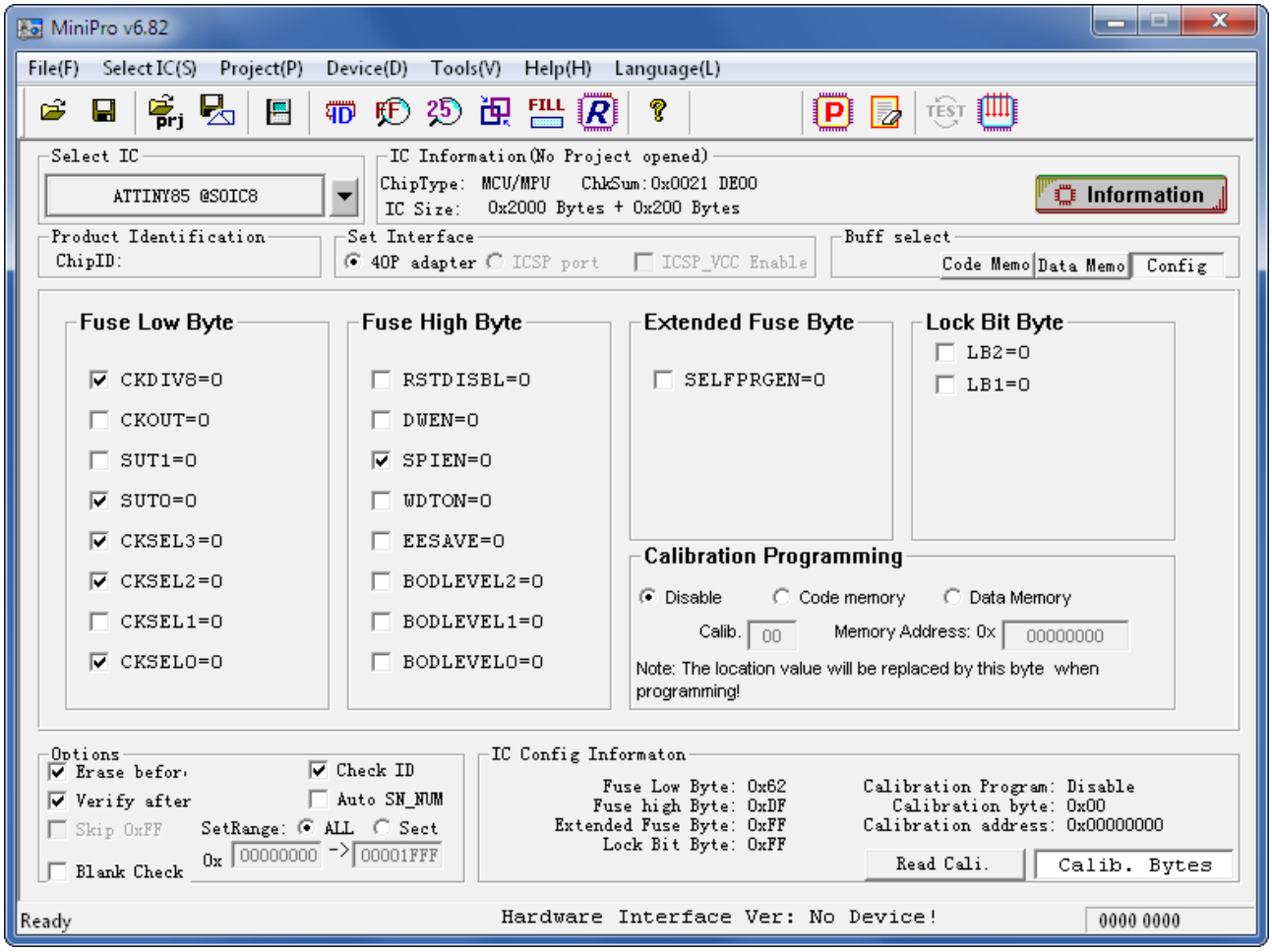
Similar workflow as Arduino as ISP. Select programmer: USBasp.

1 ATtiny library Installation instructions https://github.com/SpenceKonde/ATTinyCore

High Voltage programmer

The MCU must be popped out of the SKS64 and into the ZIF socket if using a HV programmer like for example a TL866CS/A. Unlike ISP programmers, HV programmers do not need MCU RESET. Activating #DISBLRST in the fuse bits will free up the last I/O of the ATtiny. The downside of #DISBLRST is that if you intend to further develop the firmware and program using ISP, you will need to use the HV programmer first to re-enable RESET!

Note: At power on, the SKS64 firmware reads the #DISBLRST fuse bit. If reset is disabled, the firmware will drive the EXROM signal. If reset IS enabled, the EXROM reset feature is disabled.



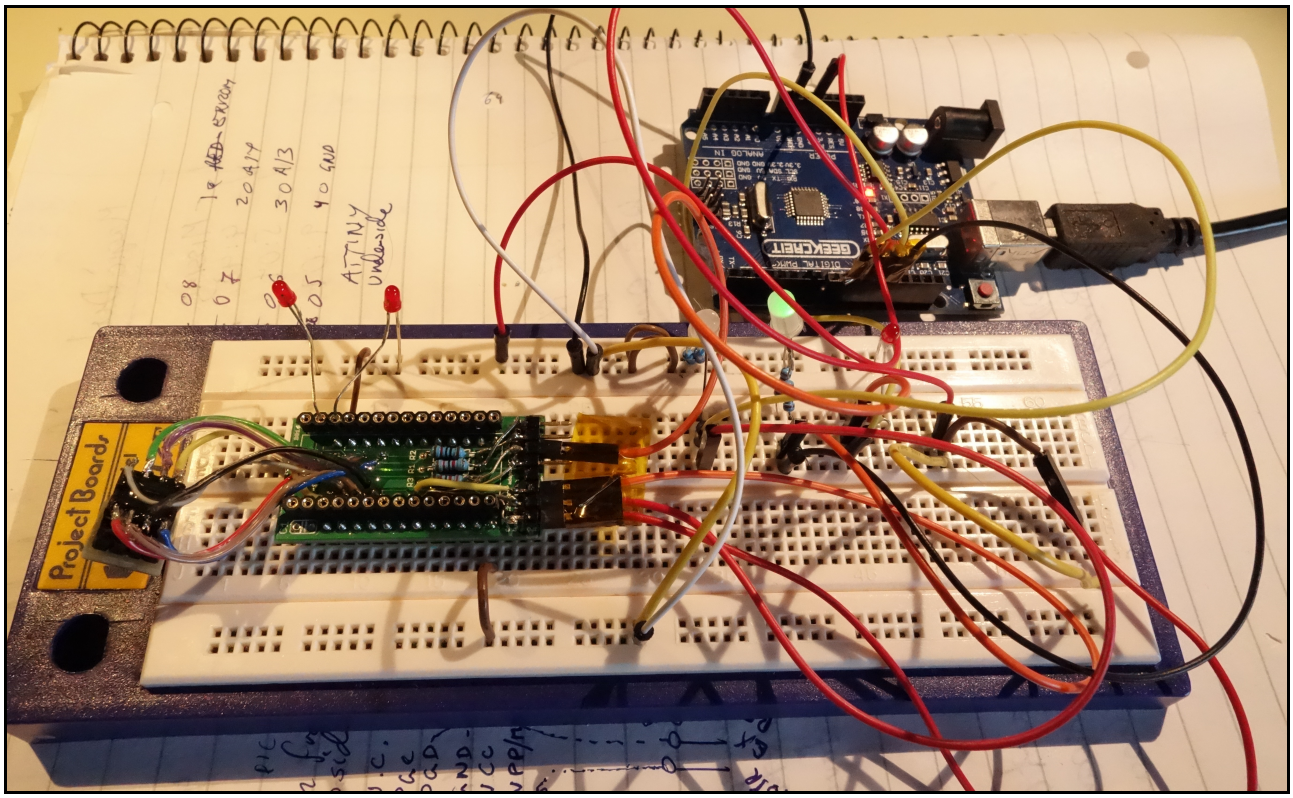
Make sure that Fuse Low Byte, high byte are 0x62 and 0xDF if reset is enabled, or 0x62 and 0x5F if disabled.

Pitfall: In MiniPro if you are in the Config window like above, and you try to load a hex file with File/Open, you will get an error message, but the fuse bits will be mangled. Writing the wrong fuse bits can lock you out of an ISP programmer.

Developing

While it is possible to program the SKS64 installed in the C64 using an ISP, the LEDs have to be removed before programming. This is exhaustive. Putting the SKS64 on a breadboard and hooking it up to an ISP (Arduino as ISP here), you can attach LEDs and a button to test the functionality of the switcher. The wiring below looks chaotic, but bear in mind it is the same circuit as shown in "Arduino as ISP" section, but with wires going from the SKS64 to the breadboard first, and then from the breadboard to the ISP.

The two LEDs to the left are connected to A14 and A13. Those LEDs have internal resistors! The next two LEDs are connected such that they light up when EXROM or INTRST (C64 reset) go low. The anode goes to +5V through resistor, and the cathode goes to the I/O. Note: The SKS64 pins R, G and B are already current limited, but if you don't use an additional resistor, you will swamp the programming signals MISO, MOSI or SCK and Arduino IDE will just fail and say "MCU id 0x0000000" or "0xffffffff". Use a resistor. The RGB LEDs are incredibly sensitive and you can still get enough light even at 20k Ohm. Same for the RED Power LED. Finally attach a switch to GND, and to the RESTORE input. Now you have a development setup. Attach power, the USB cable, and you should be able to upload programs to the SKS64 without rearranging the cabling for each program iteration!



The RGB LED Cable Assembly

The current selected kernal ROM image is indicated with an RGB LED. RED, GREEN, BLUE and CYAN. It must ba a common cathode type*. Use the LED pinout diagram as shown below to locate the negative, red green and blue LED connections. Attach wires and use heat shrink tubing. You can see an excellent realization of this in the pictures below done by **@thilographie_de** . Thank you for letting me use your pictures.

The other end of the cable goes into a 2x3 female pin header. The connections for R, G and B in the PCB are noted on the silk-screen (the white text on the PCB). You can solder the wires directly onto the pcb if you like, but it is nice to be able to separate top enclosure where RGB LED is clipped into from the rest of the computer.

* A common cathode means that all cathodes are joined together. The cathode is the "negative" side of the LED. I bought it on eBay, and the product title was: "*4PIN 5mm RGB LED - Tri-Colour 3 in 1 - Frosted Diffused Common Cathode*".