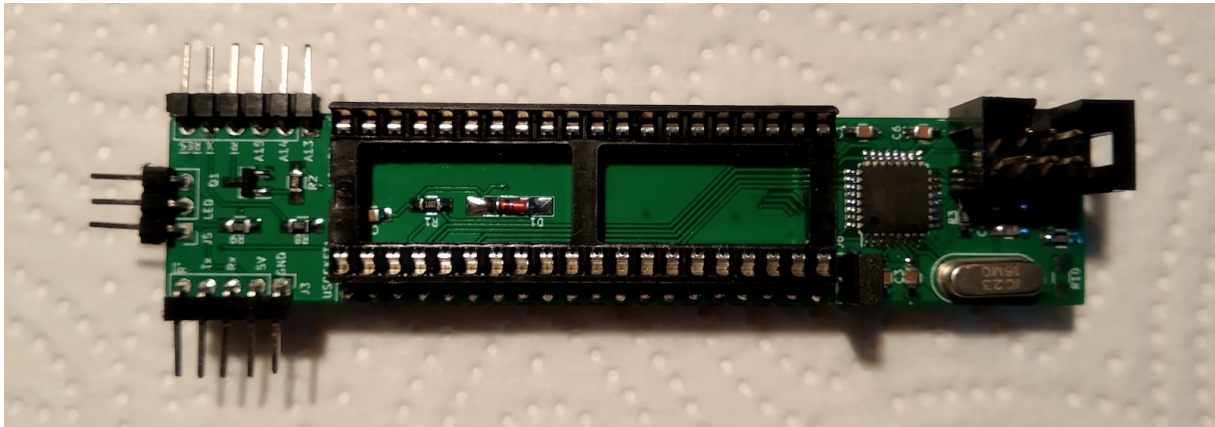


Project Documentation

Commodore C64 – Keyboard Controlled Kernal Switch 6526

Revision: 1.0

Date: 03.10.2020



C64 Keyboard Controlled Kernal Switch 6526 Rev. 1.0

General

This project is based on the C64 Keyboard Controlled Kernal Switch Rev. 1.0 from Sven Petersen (github: svenpertersen1965). For additional informations take a look at Sven Petersens github repository.

The original kernal switch ist directly connected to the C64 keyboard connector. In my C64 (bread box) with board Assy 250407 the Kernal Switch lies on the fastening screw and other components with the risk of an electrical connection. So I developed an alternative.

The kernal switch 6526 is an intermediate socket for the 6526 on U1.

Advantage:

- better mechanical stability
- SMD soldering pads for hand soldering

Disadvantage:

- additional line for querying the restore button
- if the 6526 is not socketed, it must be unsoldered

Connection points:

RESET → Userport pin 3



Figure 1: Reset Contact ASSY 250425 (identical to ASSY 250407) – Long Board

EXROM → Expansionport pin 9

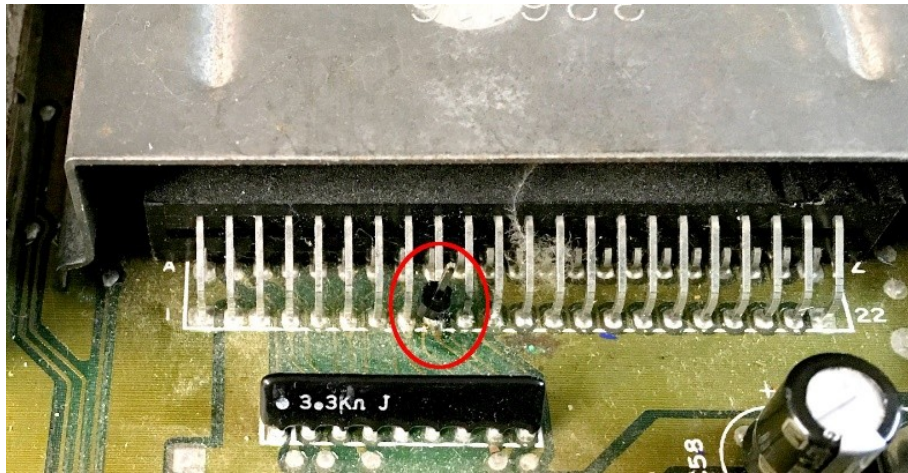


Figure 2: EXROM pin at the Expansion Port (Pin 9)

or

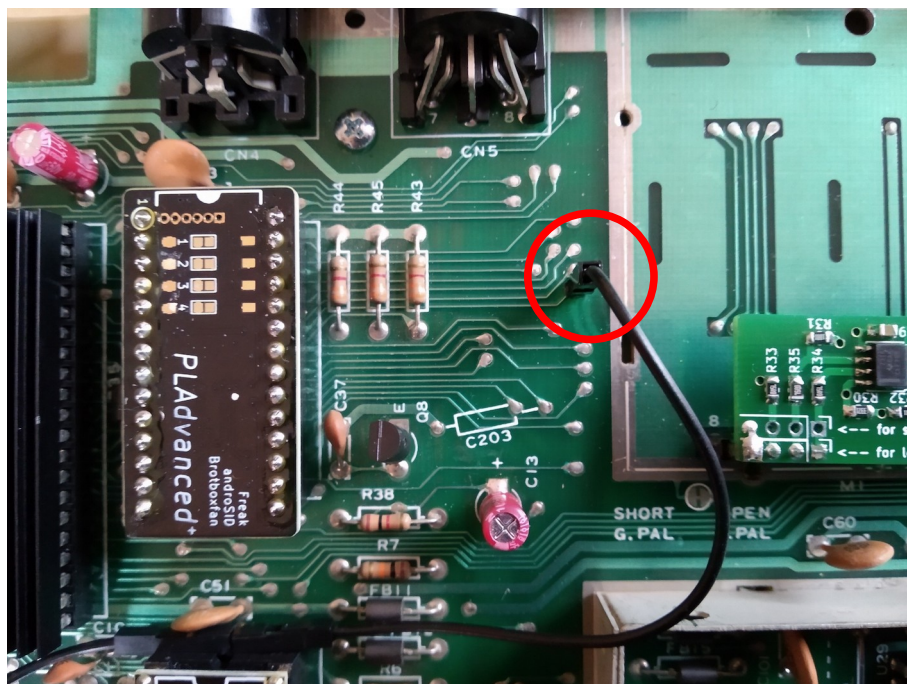


Figure 3: EXROM at the board 250407

RESTORE → east side of R41



Figure 4: EXROM at the board 250407

Wiring of ASSY 250407. The brown cable is the reset signal, which is attached to the RESET pin header that was installed previously. The black cable is EXROM. The brown cable with the yellow clamp is RESTORE.

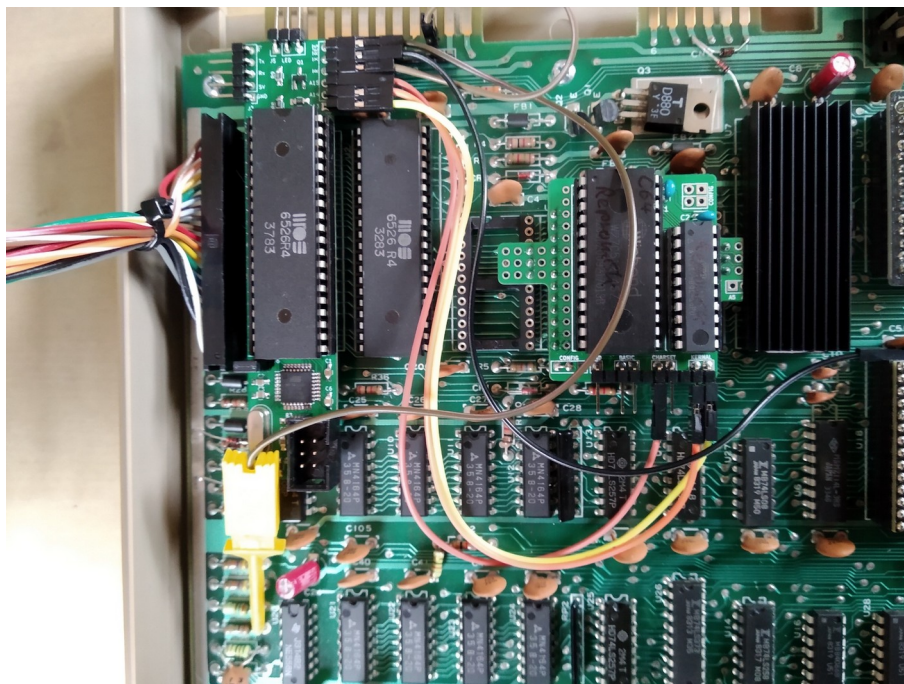


Figure 5: Wiring of ASSY 250407

Programming:

Program with an ISP-programmer like the Pololu USB AVR Programmer v2.1. With MacOS you can use the program AVRfuses.

It is not necessary to remove the Kernal Switch from the C64 for programming. **However, the C64 must be switched off.**

For programming, the ATmega 328P must be supplied with 5V from the ISP programmer.

The programming mode is selected with J6.

J6 open → programming mode

J6 closed → kernal switch mode

Set the fuses first when programming for the first time.

Fuses:

- EXTENDED: 0xFD
- HIGH: 0xDE
- LOW: 0xFF

After that you can program the firmware.