Project Documentation

C64 Kernal Adapter/Switch (Short Board)

Project number: 123

Revision: 0

Date: May the 4th, 2019

C64 Kernal Adapter/Switch (Short Board) Rev. 0

Module Description

Introduction

The board serves for adapting the BASIC/KERNAL ROM U4 (type 23128b) to a 27C512 (or 27C256 or 27C128) EPROM. The pin out of both ICs are slightly different and need adaptation. Furthermore, it allows to access (up to 7) different kernals, which can be selected via the pin-header on the module.

This pin-header is connected in a way, that the selection can either be accomplished with standard 2.54mm jumper bridges, DIP-switches, hex-encoding switches or a microcontroller like an Arduino etc.

Signal	Pin	Pin	Signal
A13	1	2	GND
A14	3	4	GND
A15	5	6	GND
+5V	7	8	+5V

Table 1: Jumper (JP1) for Bank Selection

The +5V pins are to provide supply voltage to a microcontroller.

Dimensions

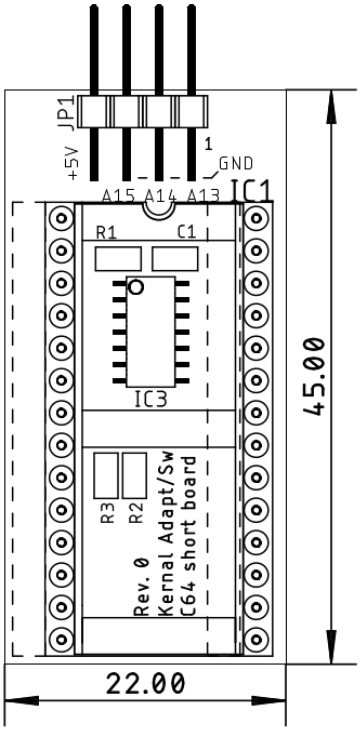


Figure 1: Kernal Adapter/Switch

Function

The short board (ASSY 250469) stores the BASIC and the KERNAL in the same ROM. Long boards (all earlier ASSYs) have a separate ROM for BASIC.

The BASIC is located in the C64's address space between 0xA000 - 0xBFFF, the KERNAL is located between 0xE000 – 0xFFFF. The highest three address bits have to be taken care to distinguish between BASIC and KERNAL.

Hex Digit	Binary [A15…A12]	ROM
A_{HEX}	HLHL	BASIC
B_{HEX}	HLHH	BASIC
E_{HEX}	HHHL	KERNAL
F_{HEX}	HHHH	KERNAL

Table 2: Encoding of the address bits

A15, A13 and A12 are encoded the same was for both the BASIC and the KERNAL. A14 makes the difference. Consulting the schematic of ASSY250469 (Document 252312), A14 of the CPU address bus is connected to A13 (pin 26) of the 16k ROM (U4/23128B). A LOW on A14 will select the lower 8k, a HIGH will select the higher 8k.

This has to be taken care of on the EPROM adapter. A LOW on A13 (which is connected to A14 of the CPU) has to select the lowest (BASIC) 8k block, a HIGH is addressing the (KERNAL) 8k memory block, which is selected by the jumper JP1. The trick is done with three AND gates (IC3). The outputs are connected to A13 … A15 of the EPROM. One input of each one is connected to pin 26 of the socket U4. In case this is LOW (BASIC selected) the output of all three gates is LOW, too. The lowest 8k memory block of the EPROM is selected. In case pin 26 is HIGH, the configuration set on JP1 appears at the output of the AND gates. This way, the selected 8k memory block of the EPROM is addressed.

Bank Selection

The desired KERNAL is selected at JP1. For the pinout refer to Table 1. The jumper is installed (vertically) in a way, that it connects the address line with the GND potential.

A15	A14	A13	8k Block	Addr. Offset
set	set	set	illegal	0x0000
set	set	open	#1	0x2000
set	open	set	#2	0x4000
set	open	open	#3	0x6000
open	set	set	#4	0x8000
open	set	open	#5	0xA000
open	open	set	#6	0xC000
open	open	open	#7	0xE000

Table 3: Selection of EPROM memory blocks

A set jumper corresponds to a LOW level (binary 0), an open jumper to a HIGH level. Do not confuse the C64 memory address and the EPROM memory address. They have the address Bit A0 to A12 in common, but the rest is different. Each of the 8k blocks appears between address $E000 and $FFFF of the C64. The setting marked illegal selects the lowest 8k, which are reserved for the BASIC. Do not use more than two jumpers to prevent this.

Sources for BASIC and KERNAL

The content of the BASIC ROM and the KERNAL ROM can be found here:

http://www.zimmers.net/anonftp/pub/cbm/firmware/computers/c64/index.html (64c.251913-01.bin)

This file includes the BASIC and the original KERNAL and have to be loaded to the lowest 16k of the EPROM. In case it is not desired to load the original KERNAL to 8k-Block #1, this memory block can be overwritten.

Separate BASIC and KERNAL ROMs and Scandinavian KERNAL ROMs can be found at the URL mentioned before or can alternatively be obtained from the Emulator Software VICE (C64/basic and C64/kernal).

Alternative kernals can be found elsewhere on internet. A reliable source is

https://csdb.dk/

The popular JiffyDOS is still a commercial product and can be acquired for little money from

http://www.go4retro.com/

JaffyDOS is a patch for JiffyDOS. The patch program can be obtained from World of Jani:

http://blog.worldofjani.com

Setting up an EPROM image

Combining the desired ROM (*.bin) files to one programming image works with the software of the programmer. This might a different with different model. Here it is demonstrated using the popular TL866. In case the programmer software does not provide the function, a Hex Editor like HxD will do the job.

First, the BASIC ROM has to be loaded into the buffer. This works like always:

File → Open → Select the BASIC ROM

Use default file load options (Figure 2):

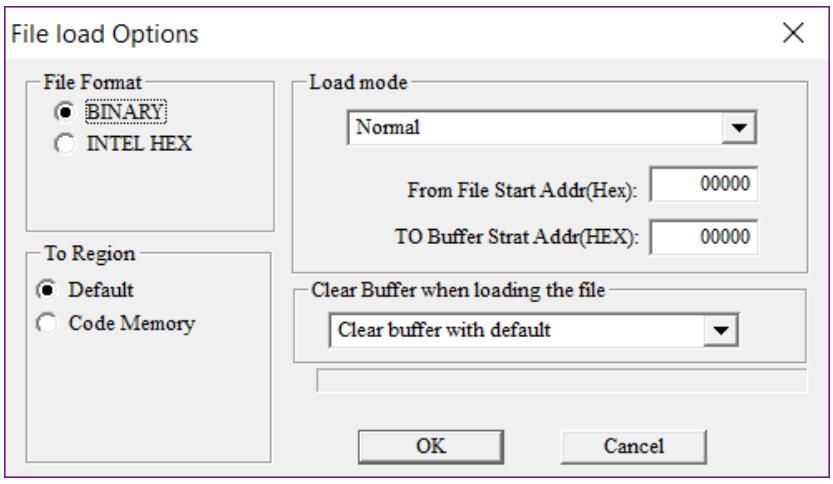


Figure 2: Default load options, clear buffer with default

Now the BASIC is loaded. It will occupy the addresses 0000 – 1FFF.

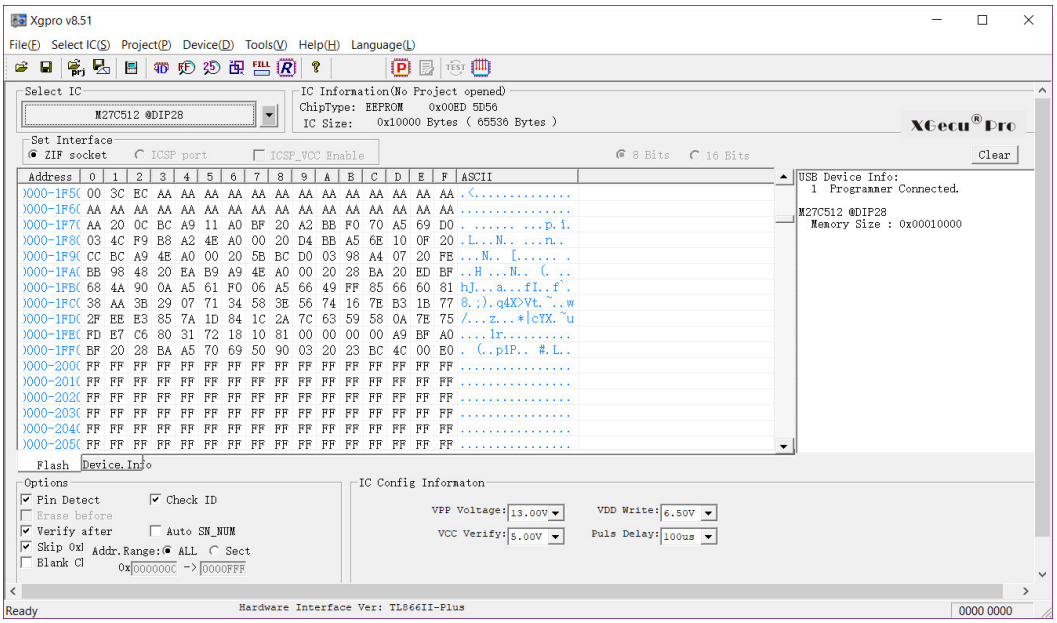


Figure 3: Buffer content after BASIC was loaded

The buffer above 1FFF is still empty (filled with FF, Figure 3).

The KERNAL has to be loaded to the next free 8k memory block in the buffer. Please refer to Table 3. Further, it is important, not to clear the buffer. Otherwise, the BASIC will be lost and the EPROM will not work. Refer to Figure 4.

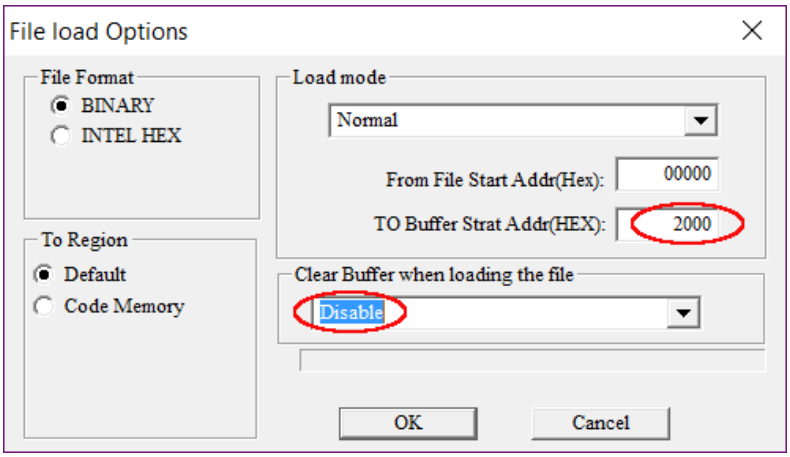


Figure 4: Load options for the first KERNAL

Clear Buffer when loading file is disabled, the address for the first kernal is 2000_{HEX}

Repeat this procedure with the appropriate buffer/EPROM address, until all desired kernals are in the buffer. Store the buffer for later use, insert a blank EPROM and program it. Insert it in the Kernal Adapter/Switch.

Adding More Kernals to an Already programmed EPROM

In case the unused 8k memory slots of the EPROM are filled with FF, it is still possible to program additional kernel images without erasing them before. FF is the content of an empty byte in an EPROM. It is possible to alter such a byte to every other byte content. Before the new memory is written, the prior content can be read into the program buffer, then a free 8k slot is determined and the kernel is loaded into that slot. Refer to chapter “Setting up an EPROM image”. Now, the

buffer content can be programmed. It is possible to program the complete buffer, the bytes already programmed will (usually) not be corrupted and after programming, the EPROM will verify ok.

Installation

Possibly, the BASIC/Kernal ROM (U4) has to be unsoldered and a socket (preferably round pin precision contacts) has to be installed, before the Kernal Adapter/Switch can be installed on ASSY250469. The notch of the EPROM is pointing towards the user port, when installed properly (Figure 5).

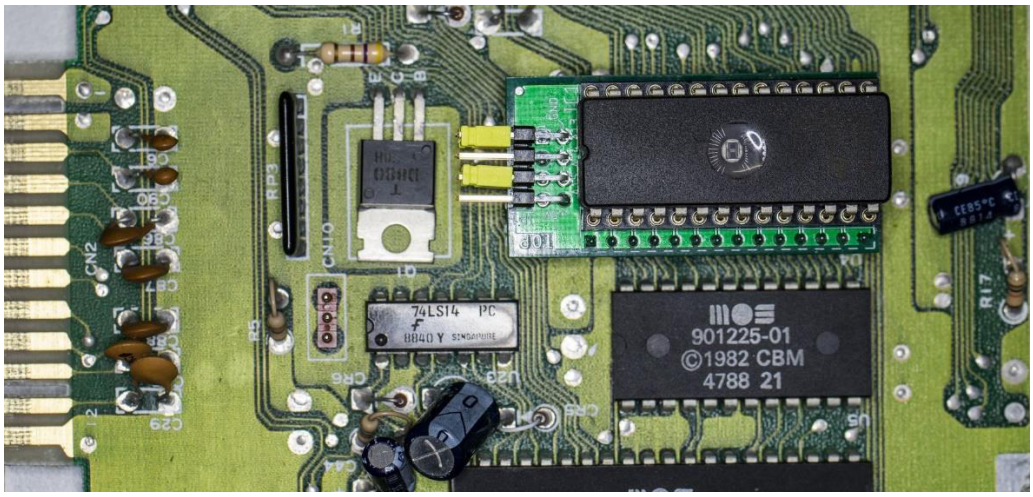


Figure 5: Installation of the Kernal Adapter/Switch

Usually, the board fits above the components placed next to U4, maybe those have to be bent backwards a bit. Prevent that the resistor touches the pin header of the adapter under all circumstances.

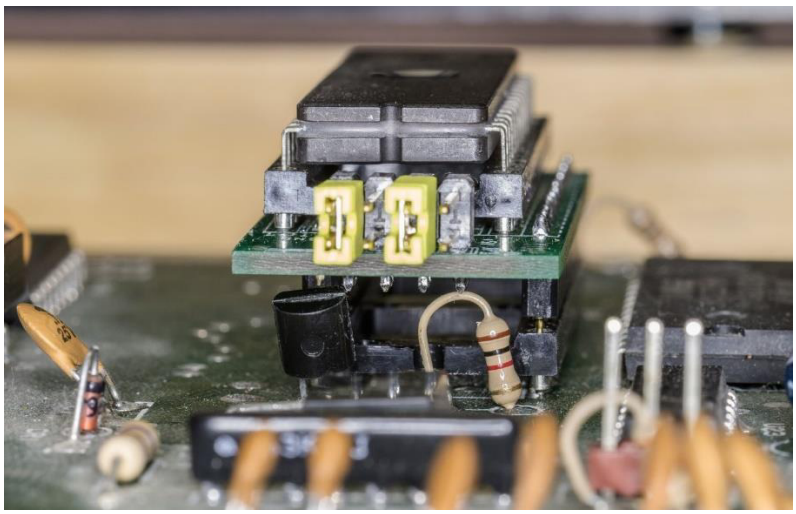


Figure 6: Parts under the adapter board

Compatibility of EPROMs

It is recommended to use 27C512 EPROMs only. The smaller EPROM Types 27C256 and 27C128 might conflict with accessing the BASIC 8k slot. The unused address signals will be tied

LOW while this access. This means, Vpp will be approximately 0V and /PGM will be low. This is a not recommended setting.

27C128									
	27C256								
		27C512							
			SOCKET						
Vpp	Vpp	A15	1	A15	VCC	28	VCC	VCC	VCC
A12	A12	A12	2	A12	A14	27	A14	A14	/PGM
A7	A7	A7	3	A7	A13	26	A13	A13	A13
A6	A6	A6	4	A6	A8	25	A8	A8	A8
A5	A5	A5	5	A5	A9	24	A9	A9	A9
A4	A4	A4	6	A4	A11	23	A11	A11	A11
A3	A3	A3	7	A3	/OE	22	/G/Vpp	/G	/G
A2	A2	A2	8	A2	A10	21	A10	A10	A10
A1	A1	A1	9	A1	GND	20	/E	/E	/E
A0	A0	A0	10	A0	D7	19	D7	D7	D7
D0	D0	D0	11	D0	D6	18	D6	D6	D6
D1	D1	D1	12	D1	D5	17	D5	D5	D5
D2	D2	D2	13	D2	D4	16	D4	D4	D4
GND	GND	GND	14	GND	D3	15	D3	D3	D3

Table 4: EPROM pin compatibility

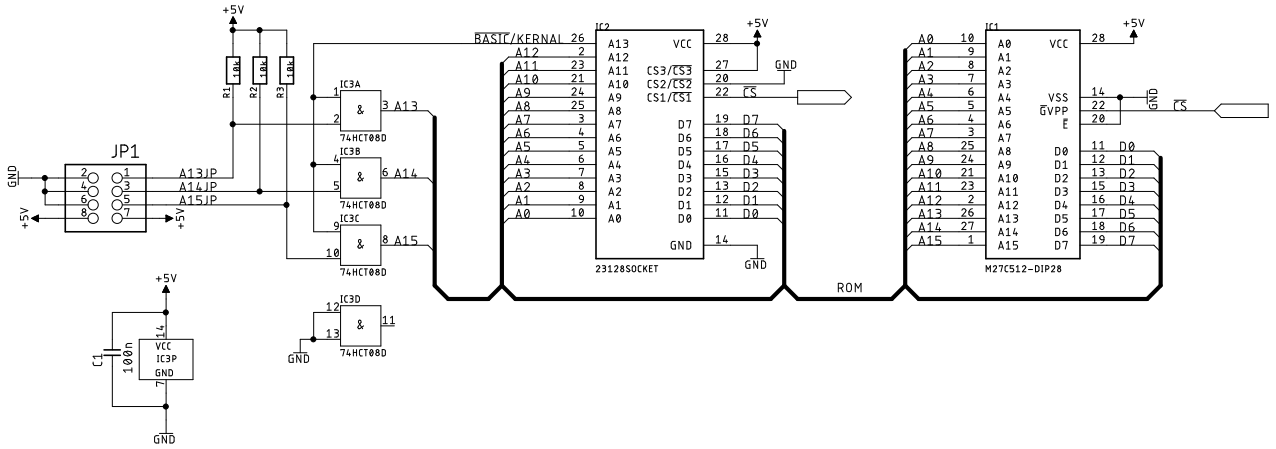
Startup and Trouble shooting

Before you insert the Kernal Adaptor/Switch into the socket of U4, you should make sure, that there are no fatal failures on it. In the worst case, it will produce a short circuit.

- Check the solder joints on the solder side
- Check the orientation of the socket. The notch is oriented to the side, where the jumper is located
- Make sure, that no jumper is installed horizontally. This could connect GND and VCC, which is a short circuit.
- After inserting the EPROM, check if the notch is at the same side like the notch of the socket. Check all pins are properly seated in the socket and not bent inwards or outwards.

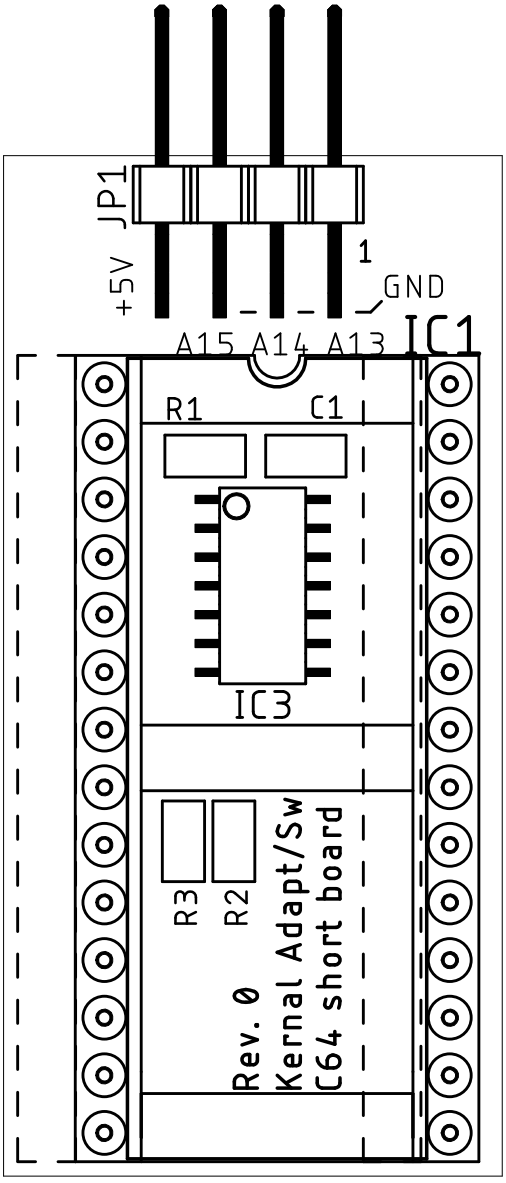
After all these points are correct, nothing really bad can happen to your C64 anymore. If you get a black screen you only have a configuration problem.

- Did you program the EPROM with a proper *.bin file? BASIC and one or more kernals? Maybe check the buffer content of your programmer software again.
- Did you set the address bits A13…A15 correctly? A set (closed) jumper means LOW
- You must not set all jumpers. This would put the BASIC in place of the Kernal, which will not work.

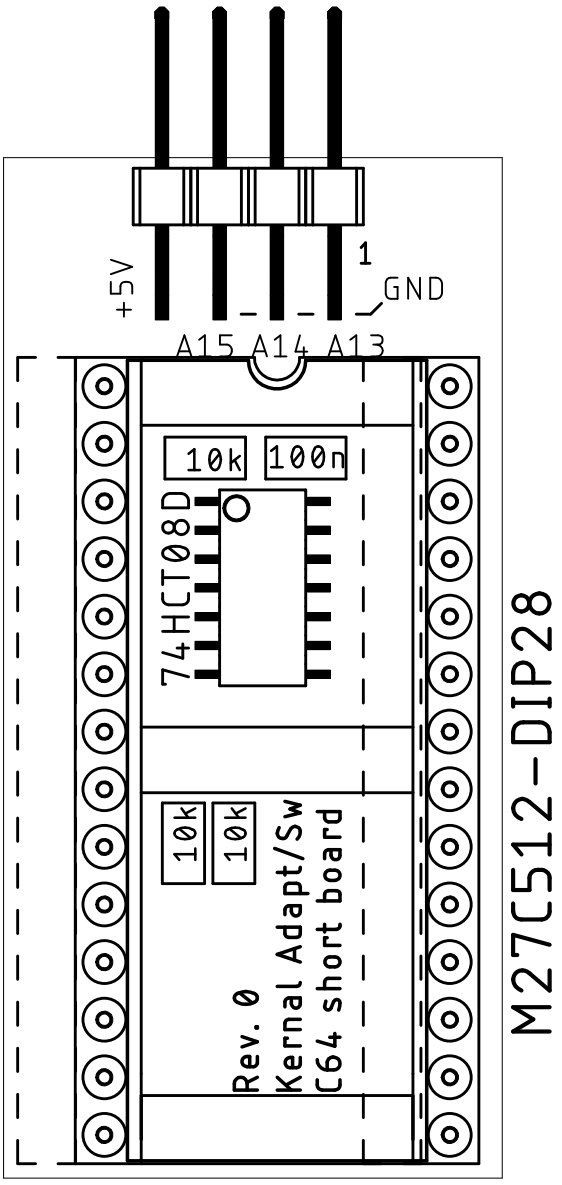


Title: C64 Kernal Adaptor/Switch for short board (ASSY 250469)		Doc.-No.: 123-1-001-00	
		Draft.: Sven Petersen	
Date: 05.05.2019 16:38		Rev.: 0	Page 1/1
File: C64_KernalSw_16k			
			A3

Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:38		Rev.: 0
placement component side		

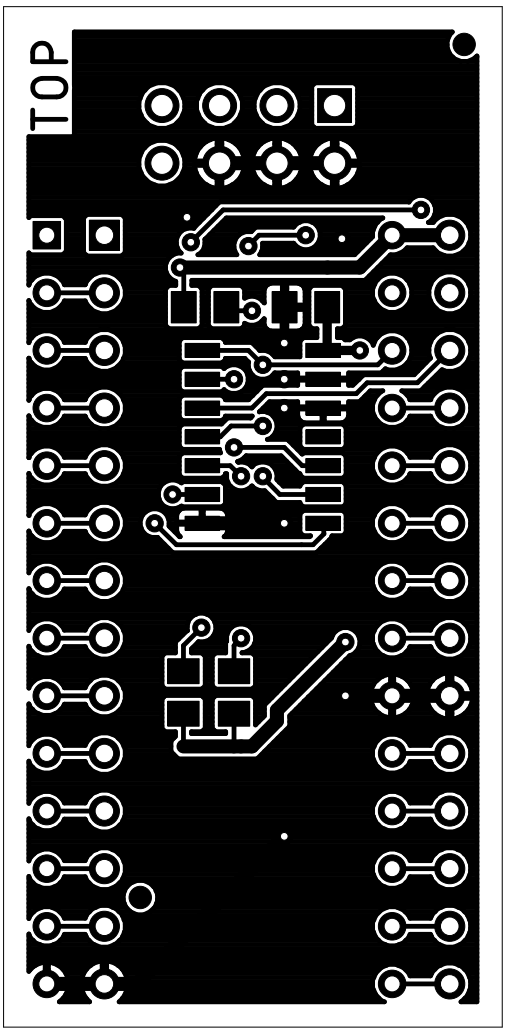


Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:42		Rev.: 0
placement component side		

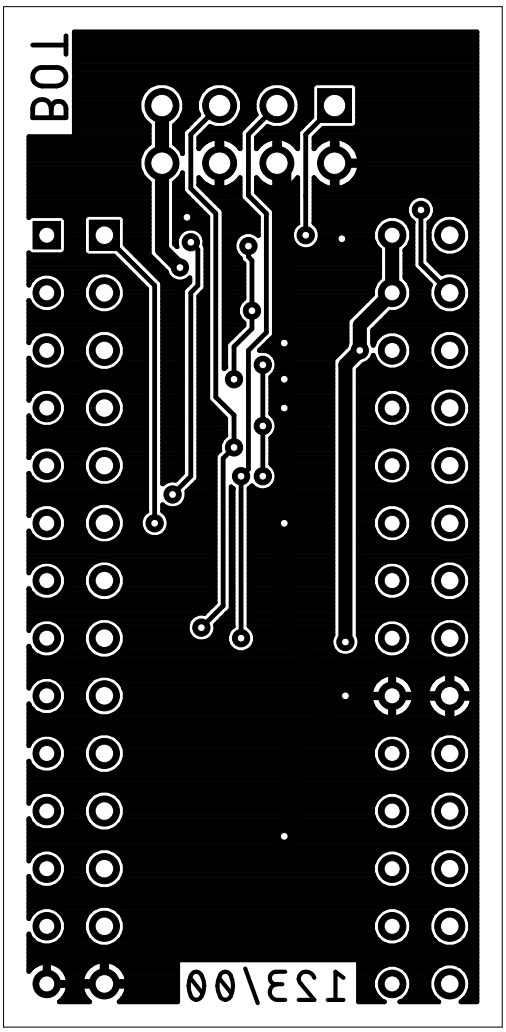


Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:42		Rev.: 0
placement solder side		

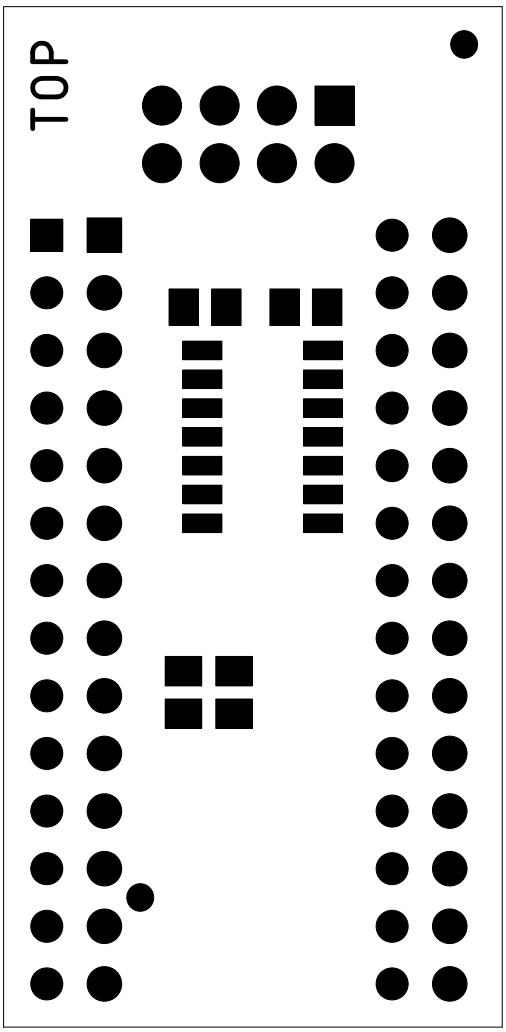
Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:42		Rev.: 0
top	[redacted]	



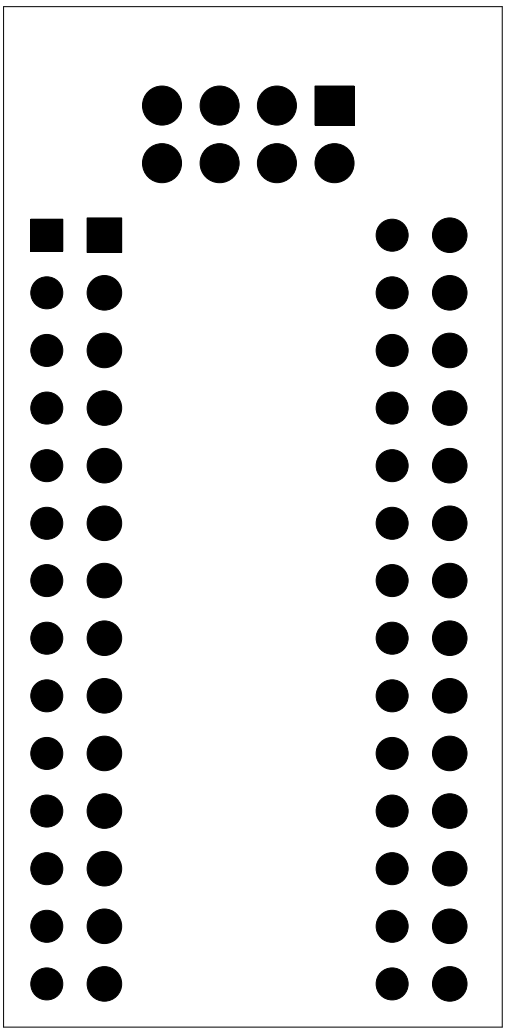
Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:42		Rev.: 0
bottom	[redacted]	



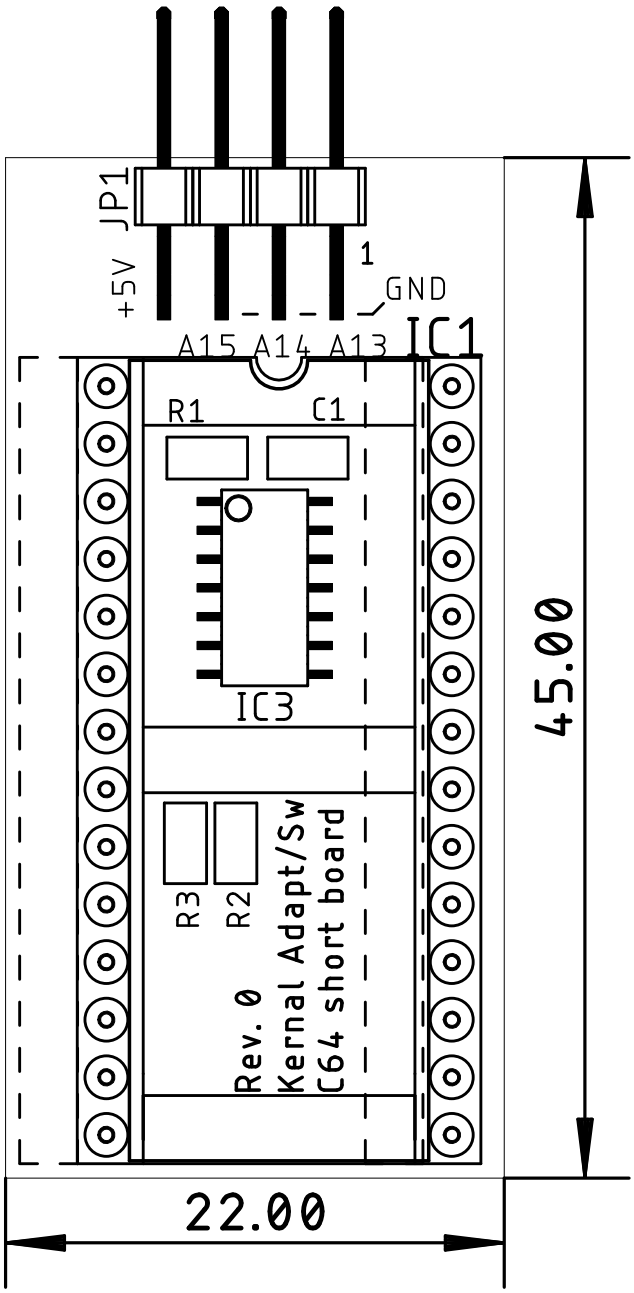
Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:42		Rev.: 0
stopmask component side		



Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:42		Rev.: 0
stopmask solder side		



Sven Petersen 2019	Doc.-No.: 123-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_16k		
05.05.2019 16:38		Rev.: 0
placement component side measures		



C64 Kernal Adapter/Switch (Short Board) Rev. 0

Testing

An image file for programming an EPROM was set up.

8k Block	Addr. Offset	Firmware
#0	0x0000	BASIC
#1	0x2000	Original Kernal
#2	0x4000	JiffyDOS
#3	0x6000	JaffyDOS
#4	0x8000	ExOS v3
#5	0xA000	SpeedDos
#6	0xC000	DolphinDos
#7	0xE000	TurboTape

Table 1: Firmware Setup

A M27C512 EPROM (ST, 100ns) was programmed using a XGecu TL866 II Plus programmer.

The EPROM was inserted into the module, the module was installed in the socket of U4 (BASIC/KERNAL) on an ASSY250469 Rev. 4 mainboard.

The jumper configured: A15 set, A14 set, A13 open.

The C64 was switched on. The commodore kernal booted, different software was loaded and executed: everything seems to be working.

The jumper setting was modified to start one alternative kernal after the other. The kernal all booted and a variety of software loaded and executed without problems.

Finally, the jumpers were configured for JiffyDOS and the C64 was used for a couple of days without any problem.

Conclusion: The C64 Kernal Adapter/switch is fully functional.

C64 Kernal Adapter/Switch for short boards Rev. 0
Bill of Material Rev. 0.0

Pos.	Qty	Value	Footprint	Ref.-No.	Comment
1	1	123-2-01-00	2 Layer	PCB Rev. 0	2 layer, Cu 35µ, HASL, 45mm x 22mm, 1.6mm FR4
2	1	2x04pin/90°	2X04_90_SERIES_088	JP1	90° pin header, 2.54mm pitch. E.g. Reichelt MPE 088-2-008
3	2	Jumper	2.54mm	(JP1)	Jumpers for address selection (in case it is intended to jumper the kenal selection)
4	1	100n	0805	C1	Ceramic cap, SMD
5	3	10k	0805	R1, R2, R3	SMD resistor
6	1	two Pinstrip, precision round pins, cut to 14 pins length	DIL28_SOCKET	(IC2)	Precision Round pins **mandatory**! E.g. Reichelt BKL 10120540 or
					10PCS Single Row 40Pin 2.54mm Round Male Pin Header machined
7	1	74LS08	SO-14	IC3	SMD, LS family recommended, HCT is probably working, but not tested
8	1	27C512	DIL28-6	IC1	EPROM 200ns or faster recommended
9	1	DIP28 socket	DIL28-6	(IC1)	Precision round pin is recommended