Project Documentation

C64 Kernal Adapter/Switch (Long Board)

Project number: 124

Revision: 0

Date: 05.05.2019

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

Module Description

Introduction

The board serves for adapting the KERNAL ROM U4 (type 2364) to a 27C512 (or 27C256, 27C128, 27C64) EPROM. The pin out of both ICs are slightly different and need adaptation. Furthermore, it allows to access (up to 8) different kernals, which can be selected via the pin-header on the module.

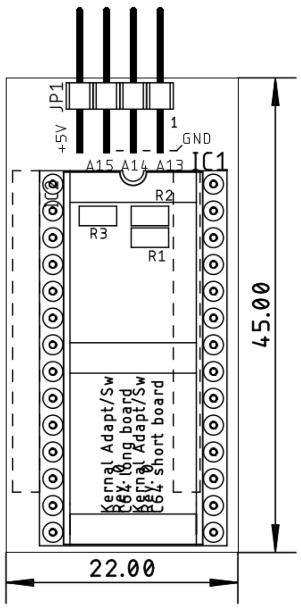
This pin-header is connected in a way, that the selection can either be accomplished with standard 2.54mm jumper bridges, DIP-switches, hex-encoding switches or a microcontroller like an Arduino etc.

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



JP1
+5V
1
GND
A15 A14 A13
IC1
R2
R3
R1
Kernal Adapt/Sw Rev. 0 C64 long board
Kernal Adapt/Sw Rev. 0 C64 short board
45.00
22.00

Figure 1: Kernal Adapter/Switch

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	#0	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 2: 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.

Sources for KERNAL

The content of the KERNAL ROM can be found here:

http://www.zimmers.net/anonftp/pub/cbm/firmware/computers/c64/index.html (kernal.901227-03.bin)

This file includes the original KERNAL (Rev. 3) and could be loaded to the lowest 8k of the EPROM. In case it is desired to have the original in bank#0. Also Scandinavian KERNAL ROMs can be found at the URL mentioned before or can alternatively be obtained from the Emulator Software VICE (./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/ or

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

http://blog.worldofjani.com

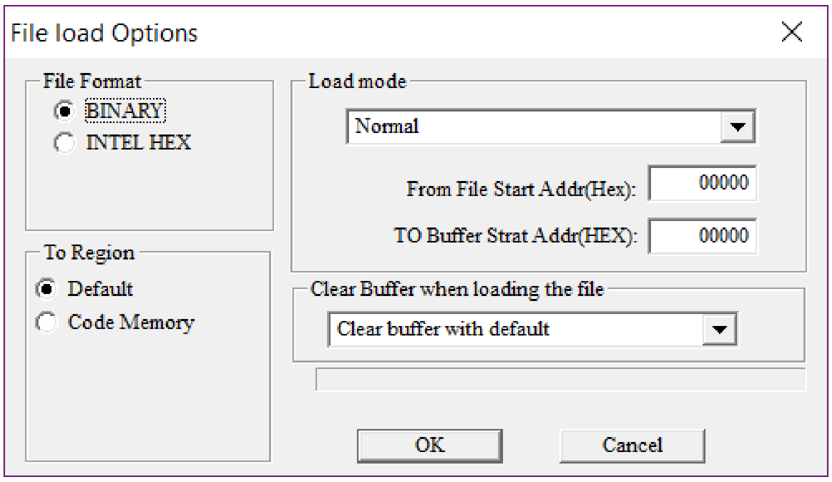
Setting up an EPROM image

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.

The first KERNAL ROM has to be loaded into the buffer like this:

File → Open → Select the BASIC ROM

Use default file load options (Figure 2):



File load Options

File Format: [✓ BINARY] INTEL HEX

To Region: [✓ Default] Code Memory

Load mode: Normal

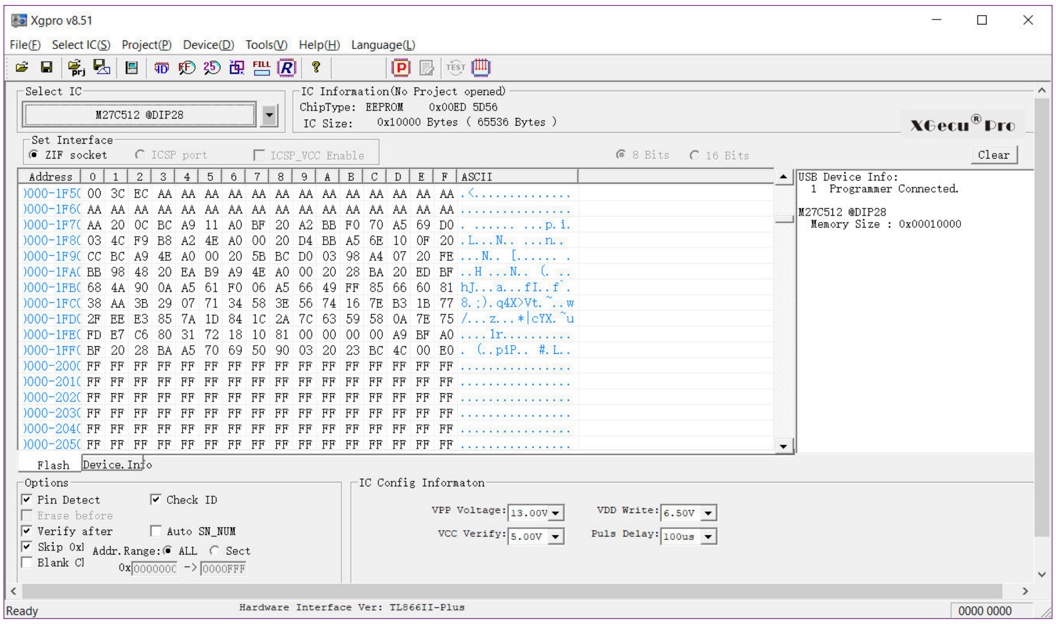
From File Start Addr(Hex): 00000

TO Buffer Strat Addr(HEX): 00000

Clear Buffer when loading the file: Clear buffer with default

Figure 2: Default load options, clear buffer with default

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



Xgpro v8.51

File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)

Select IC: M27C512 @DIP28

IC Information(No Project opened)
ChipType: EEPROM 0x00ED 5D56
IC Size: 0x10000 Bytes (65536 Bytes)

Set Interface: [✓ ZIF socket] ICSP port, ICSP_VCC Enable; [✓ 8 Bits] 16 Bits

```
Address  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  ASCII
)000-1F5( 00 3C EC AA AA AA AA AA AA AA AA AA AA AA AA AA .<..............
)000-1F6( AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA ................
)000-1F7( AA 20 0C BC A9 11 A0 BF 20 A2 BB F0 70 A5 69 D0 . ...... ...p.i.
)000-1F8( 03 4C F9 B8 A2 4E A0 00 20 D4 BB A5 6E 10 0F 20 .L...N.. ...n..
)000-1F9( CC BC A9 4E A0 00 20 5B BC D0 03 98 A4 07 20 FE ...N.. [...... .
)000-1FA( BB 98 48 20 EA B9 A9 4E A0 00 20 28 BA 20 ED BF ..H ...N.. (. ..
)000-1FB( 68 4A 90 0A A5 61 F0 06 A5 66 49 FF 85 66 60 81 hJ...a...fI..f`.
)000-1FC( 38 AA 3B 29 07 71 34 58 3E 56 74 16 7E B3 1B 77 8.;).q4X>Vt.~..w
)000-1FD( 2F EE E3 85 7A 1D 84 1C 2A 7C 63 59 58 0A 7E 75 /...z...*|cYX.~u
)000-1FE( FD E7 C6 80 31 72 18 10 81 00 00 00 00 A9 BF A0 ....1r..........
)000-1FF( BF 20 28 BA A5 70 69 50 90 03 20 23 BC 4C 00 E0 . (..piP.. #.L..
)000-200( FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
)000-201( FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
)000-202( FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
)000-203( FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
)000-204( FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
)000-205( FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
```

USB Device Info:
1 Programmer Connected.

M27C512 @DIP28
Memory Size : 0x00010000

Options: [✓ Pin Detect] [✓ Check ID] Erase before [✓ Verify after] Auto SN_NUM [✓ Skip 0xl] Blank Cl
Addr.Range: [✓ ALL] Sect 0x 0000000 -> 0000FFF

IC Config Informaton
VPP Voltage: 13.00V
VDD Write: 6.50V
VCC Verify: 5.00V
Puls Delay: 100us

Ready   Hardware Interface Ver: TL866II-Plus   0000 0000

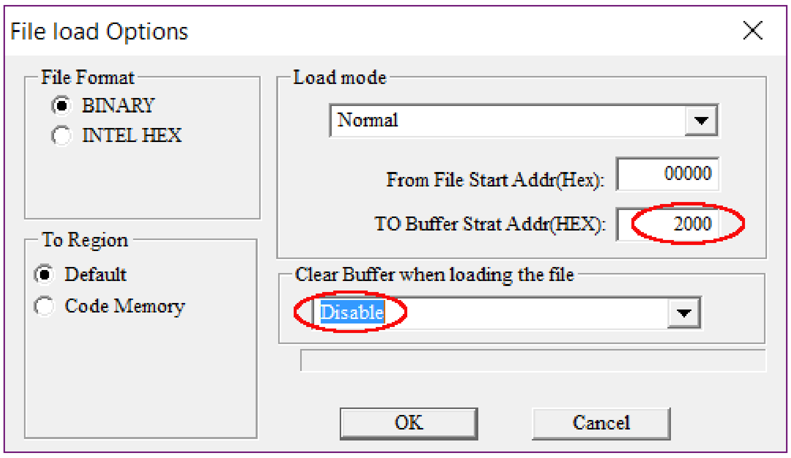
Figure 3: Buffer content after BASIC was loaded

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

The next KERNAL has to be loaded to the next free 8k memory block in the buffer. Please refer to Table 2. 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.

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.


File load Options

File Format: (● BINARY) (INTEL HEX)

To Region: (● Default) (Code Memory)

Load mode: Normal

From File Start Addr(Hex): 00000

TO Buffer Strat Addr(HEX): 2000

Clear Buffer when loading the file: Disable

Figure 4: Load options for the first KERNAL

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

In some cases, the 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 the Long Board. The notch of the EPROM is pointing towards the user port, when installed properly (Figure 5).

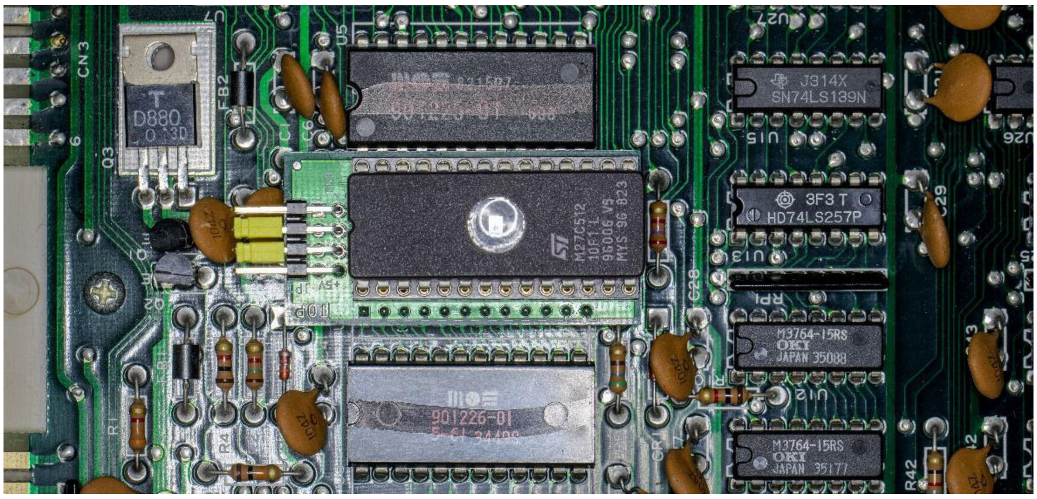


Figure 5: Installation of the Kernal Adapter/Switch

The capacitor under the board needs to be bent a bit backwards in order not to collide with it.

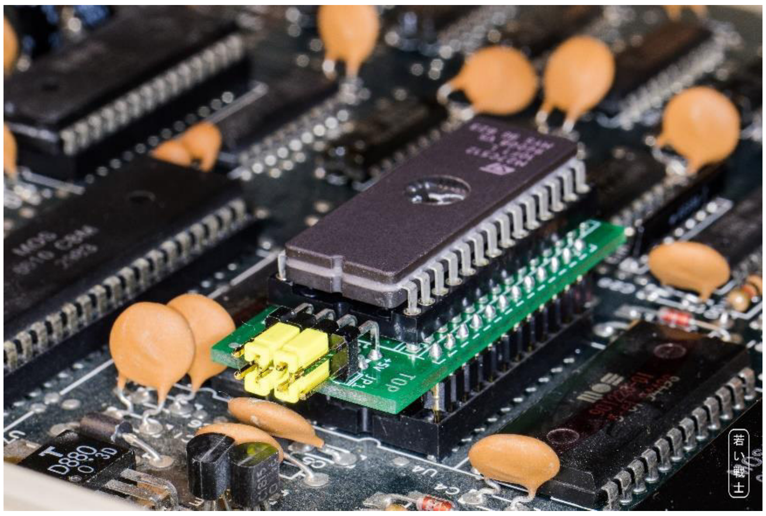


Figure 6: Parts under the adapter board

Compatibility of EPROMs

Although a 27C512 type EPROM is recommended, other types of EPROMs can be installed:

EPROM	Size	Capacity
27C64	8k	1x KERNAL
27C128	16k	2x KERNAL
27C256	32k	4x KERNAL
27C512	64k	8x KERNAL

Table 3: Capacity of EPROM types

Those EPROMs are pin compatible, the jumpers, that have no function, due to the size, have to stay open.

EPROM	Size	A15	A14	A13
27C512	64kx8	☑	☑	☑
27C256	32kx8	open	☑	☑
27C128	16kx8	open	open	☑
27c64	8kx8	open	open	open

Table 4: Settings per EPROM type

☑: The jumper can be open or closed, depending on the desired selection.

In case Vpp is located at a dedicated pin (pin 1), A15 has no effect anymore. A HIGH level is recommended, the corresponding jumper is open. The /PGM Pin should be set HIGH, this is accomplished by an open jumper for A14. "n.c." means not connected. Just leave the jumper open for it.

27C64	27C128	27C256	27C512	SOCKET				27C512	27C256	27C128	27C64
Vpp	Vpp	Vpp	A15	1	A15	VCC	28	VCC	VCC	VCC	VCC
A12	A12	A12	A12	2	A12	A14	27	A14	A14	/PGM	/PGM
A7	A7	A7	A7	3	A7	A13	26	A13	A13	A13	n.c.
A6	A6	A6	A6	4	A6	A8	25	A8	A8	A8	A8
A5	A5	A5	A5	5	A5	A9	24	A9	A9	A9	A9
A4	A4	A4	A4	6	A4	A11	23	A11	A11	A11	A11
A3	A3	A3	A3	7	A3	/OE	22	/G/Vpp	/G	/G	/G
A2	A2	A2	A2	8	A2	A10	21	A10	A10	A10	A10
A1	A1	A1	A1	9	A1	GND	20	/E	/E	/E	/E
A0	A0	A0	A0	10	A0	D7	19	D7	D7	D7	D7
D0	D0	D0	D0	11	D0	D6	18	D6	D6	D6	D6
D1	D1	D1	D1	12	D1	D5	17	D5	D5	D5	D5
D2	D2	D2	D2	13	D2	D4	16	D4	D4	D4	D4
GND	GND	GND	GND	14	GND	D3	15	D3	D3	D3	D3

Table 5: EPROM pin compatibility

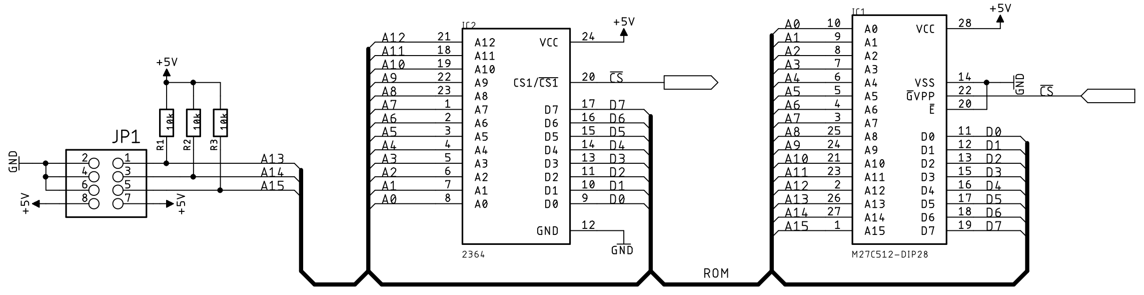
Startup and Trouble shooting

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.

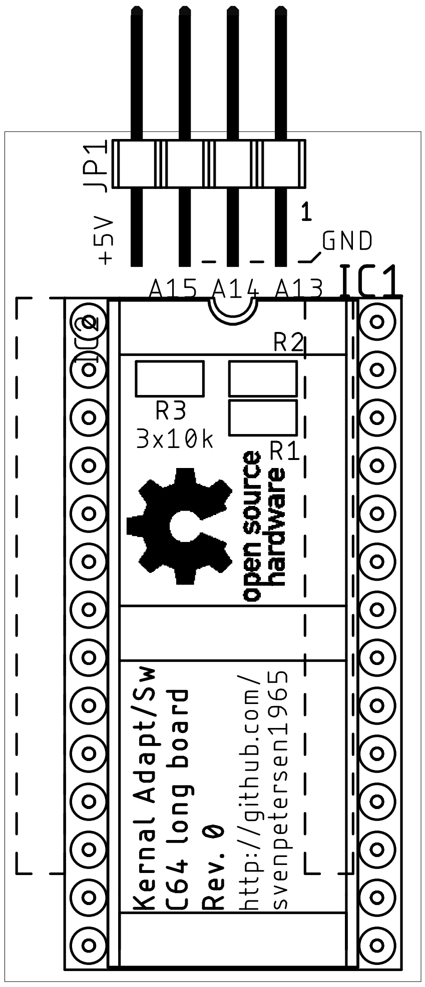
After all these points are correct, nothing really bad can happen to your C64 anymore. If you get a black screen you only have a configuration problem.

- Did you program the EPROM with a proper *.bin file? (*.rom) Maybe check the buffer content of your programmed software again.
- Did you set the address bits A13…A15 correctly? A set (closed) jumper means LOW

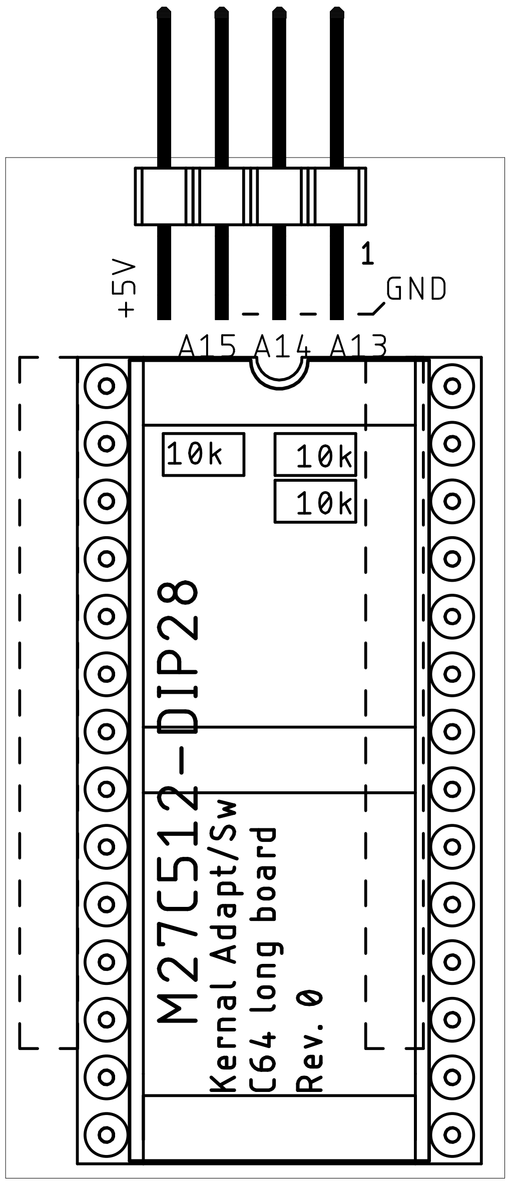


Title:	C64 Kernal Adaptor/Switch for long boards	Doc.-No.:	124-1-001-00
		Draft.:	Sven Petersen
Date:	30.07.2020 21:33	Rev.: 0	Page 1/1
File:	C64_KernalSw_8k		
	http://github.com/svenpetersen1965		A3

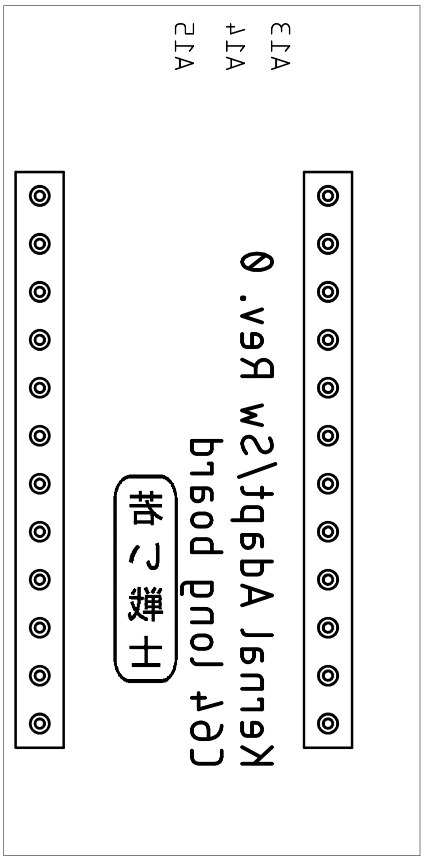
Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
30.07.2020 21:36		Rev.: 0
placement component side		



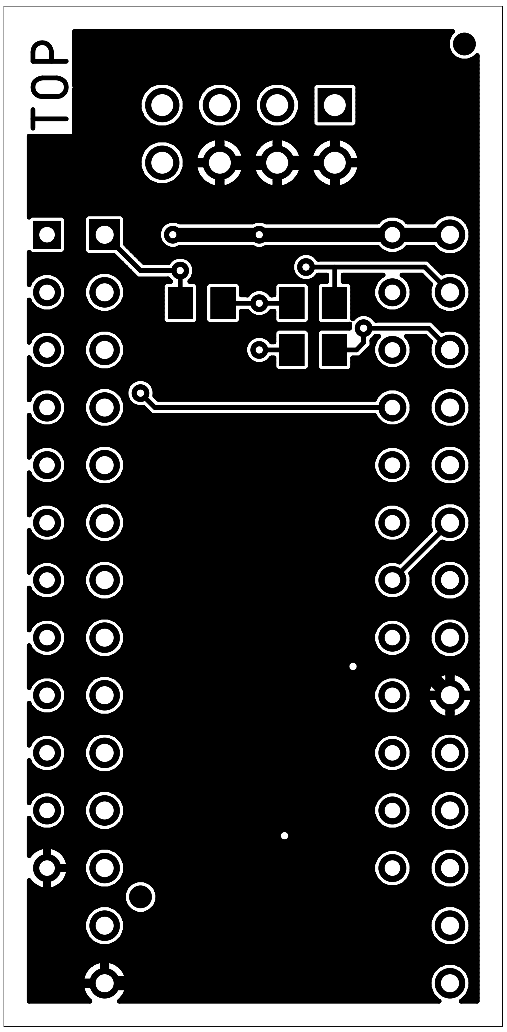
Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
05.05.2019 16:30		Rev.: 0
placement component side		



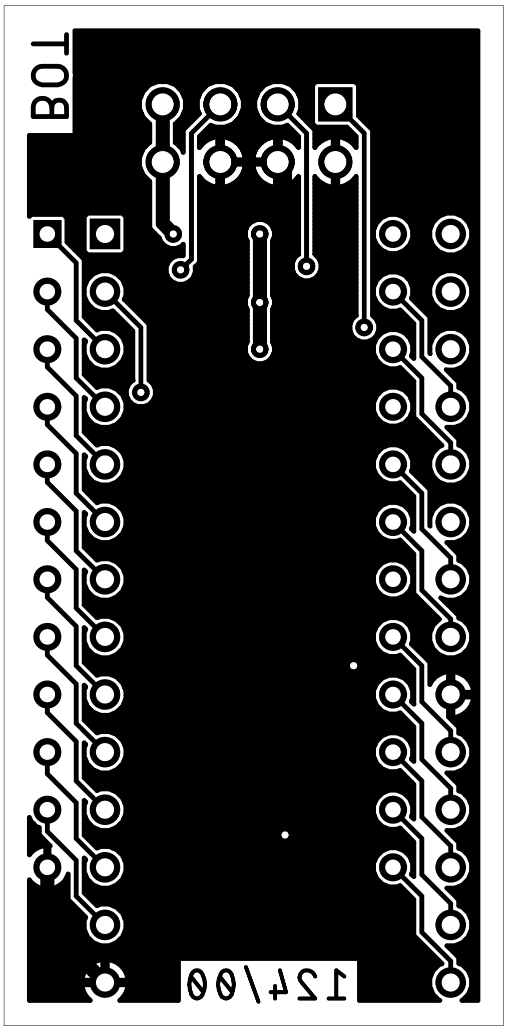
Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
30.07.2020 21:36		Rev.: 0
placement solder side		



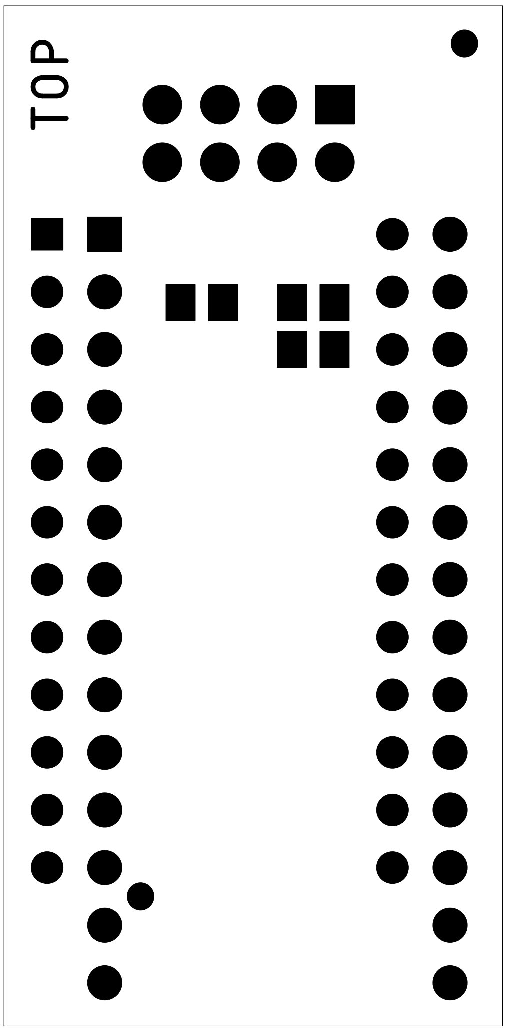
Sven Petersen	Doc.-No.: 124-2-01-00	
2019	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
05.05.2019 16:30		Rev.: 0
top	[redacted]	



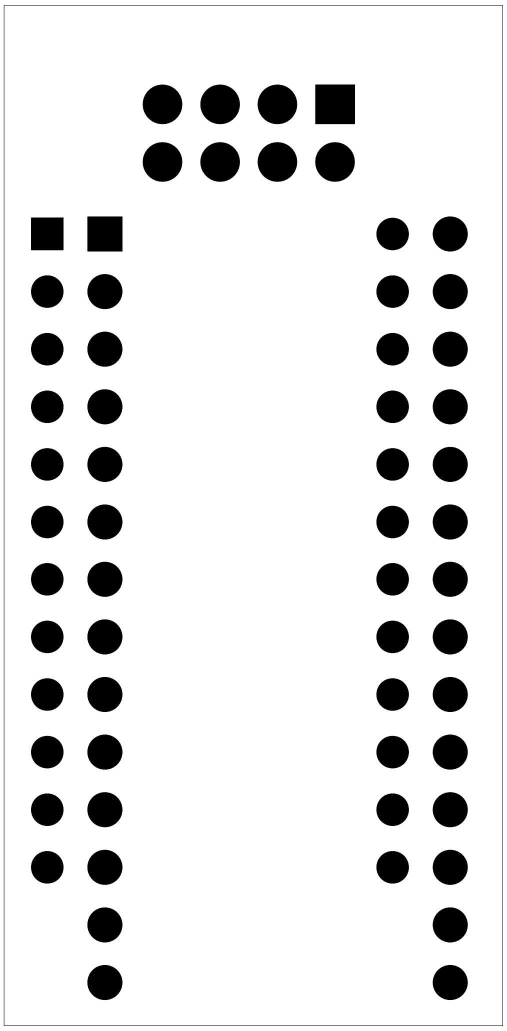
Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
05.05.2019 16:30		Rev.: 0
bottom	[redacted]	



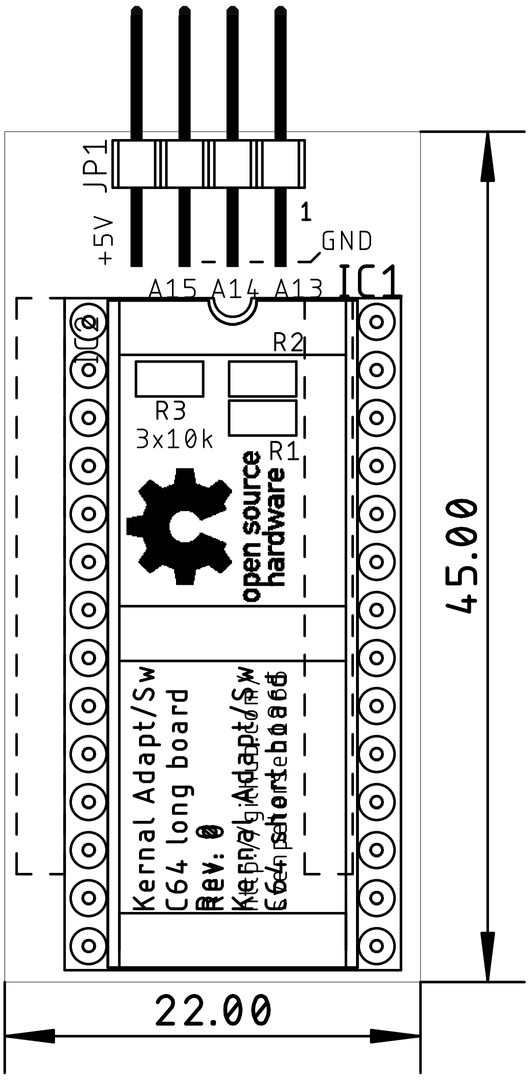
Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
05.05.2019 16:30		Rev.: 0
stopmask component side		



Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
05.05.2019 16:30		Rev.: 0
stopmask solder side		



Sven Petersen 2019	Doc.-No.: 124-2-01-00	
	Cu: 35µm	Cu-Layers: 2
C64_KernalSw_8k		
30.07.2020 21:36		Rev.: 0
placement component side	measures	



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

Testing

An image file for programming an EPROM was set up.

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

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 (KERNAL) on an ASSY250407 Rev. B mainboard.

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

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.

Since the pin compatibility with other EPROMs (27C256, 27C128, 27C64) is widely proved, it is assumed, that this kernal adapter works with them as well.

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

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

Pos.	Qty	Value	Footprint	Ref.-No.	Comment
1	1	124-2-01-00	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	3	Jumper	2.54mm	(JP1)	Jumpers for address selection (in case it is intended to jumper the kenal selection)
4	3	10k	0805	R1, R2, R3	SMD resistor
5	1	two Pinstrip, precision round pins, cut to 12 pins length	DIL24_SOCKET	IC2	Precision Round pins **mandatory**! E.g. Reichelt BKL 10120540 or
					[10PCS Single Row 40Pin 2.54mm Round Male Pin Header machined]()
6	1	M27C512	DIL28-6	IC1	EPROM 200ns or faster recommended, alternative sizes: 27C64, 27C128, 27C256 possible
7	1	DIP28 socket	DIL28-6	(IC1)	Precision round pin is recommended