

Commodore- Hardware- Retrocomputing

25 Jahre nach C64 & Co.:
C64 und A500 reparieren,
warten und erweitern

Peter Sieg

4. AUCHTALS
ERWEITERTE
AUFLAGE!

Sieg

Commodore-
Hardware-Retrocomputing

Peter Sieg

Commodore-Hardware- Retrocomputing

**25 Jahre nach C64 & Co.:
C64 und A500 reparieren, warten und erweitern**



Skriptorium-Verlag
Edition »Retrobooks«
34326 Morschen

www.skriptorium-vd.de
www.retrobooks.de

info@skriptorium-vd.de

Lesestoff für echte Computerfreaks

Bibliografische Informationen der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Verlagsanschrift:
Skriptorium-Verlag
Harald Horchler
Gartenstraße 4
34326 Morschen
Telefon: 056 64 / 93 99 46
Fax: 056 64 / 93 99 46
Internet: www.skriptorium-vd.de
Webshop: www.retrobooks.de
E-Mail: info@skriptorium-vd.de

Satz und Layout: Harald Horchler

ISBN 978-3-938199-13-8

Printed in Germany

Sieg, Peter: Commodore-Hardware-Retrocomputing. 25 Jahre nach C64 & Co.: C64 und A500 reparieren, warten und erweitern

4., erweiterte und ergänzte Auflage 2011

© Copyright 2008, 2009, 2011 Skriptorium-Verlag. Alle Rechte vorbehalten. Jegliche Vervielfältigung von Texten, Tabellen, Grafiken oder Bildern, auch in elektronischer Form, bedarf der schriftlichen Genehmigung des Verlags.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Waren-, Soft- und Hardwarebezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und der Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Gleiches gilt für die Wiedergabe von Logos, Schaltbildern und Markennamen. Trotz sorgfältiger Prüfung übernehmen weder Verlag noch Autor Gewähr für die inhaltliche Richtigkeit des Buchinhalts sowie für Schäden irgendwelcher Art, die durch die Nutzung der im Buch beschriebenen Schaltungen usw. entstehen. Das Nutzen der in diesem Buch enthaltenen Informationen erfolgt auf eigene Gefahr! Die bei Arbeiten mit Strom und elektronischen Bauteilen üblichen Sicherheitsbestimmungen sind unbedingt einzuhalten! Die Rechte an verwendetem Bildmaterial liegen bei den jeweiligen Rechteinhabern.

Inhaltsverzeichnis

Vorwort	7
Der Commodore 64	8
Wenn der C64 nicht mehr funktioniert	9
IEC-ATA V2 – IDE-Laufwerke am C64	15
C64-Bausteine (Chips)	16
Chip-Bezeichnungen	19
EPROM-Pinouts	23
C64-Ersatz-PLA auf 27512-EPROM-Basis	25
Ersatz-PLA auf CPLD-Basis	27
Platinenrestauration	28
C64-Restauration von Pentagon	32
Großumbau C64: Verringerung der Stromaufnahme, lebensverlängernde Maßnahmen	38
ROM auf EPROM	40
Ersatz-PLA	41
7805 auf Schaltregler	42
7812 auf Schaltregler	43
RAM-Umbau von 4164 auf 53c464-70	44
Gleichrichter auf Schottky-Dioden	46
Kühlkörper nachrüsten	47
C64-Netzteilenschutzschaltung	48
Netzteil nur für C64G/II, ASSY 250469 und VC20	51
PC-Netzteil-Adapter	52
BASIC und Kernel bei ASSY 250469 durch EPROM ersetzen	54
S-Video-Ausgang am C64	55
MMC2IEC-SD2IEC: SD-Card am C64	56
Neue Firmware sd2iec	64
Eagle-Layout für MMC2IEC	65
EasyFlash für C64	68
NeoRAM – batteriegepufferte RAM-Disk für C64	74
DCM – Draco Cart Maker	77
Speedmessungen verschiedener Lösungen	78
1541 Ultimate	80
SwinSID – SID-Ersatz auf AVR-Basis	82
Micro SwinSID88	88
Tiny-Eprommer-Nachbau	90
Floppy-»No Disk«-Hack für den Amiga	95
VC20-Multi-Cartridge	96
Final Expansion für VC20	99
Neigungssensor + AVR als Joystick	104
TFT-Kontrollmonitore für C64	110

Einfachst-PIC-Programmer (Ludipipo)	111
XS1541 – AVR zwischen PC- und CBM-Floppy	112
XU1541 – USB-Interface für CBM-Floppys	116
Kampf dem Gilb oder: Wie man vergilbte Plastikteile wieder hell bekommt	117
C64DTV	118
Easyflash	123
Amiga 500	124
Amiga 600	124
Amiga-Customchips	125
Amiga-Fehlercodes nach dem Einschalten	127
Amiga-Akku-Schaden	128
CF-Bootdiskette erstellen	131
ADF auf Floppy schreiben	132
PS/2-Maus am Amiga (und Atari ST)	133
PS/2-Tastaturadapter – nicht nur für A500	134
PC-Floppy für Amiga umbauen	136
Umbau A500+ auf 2MB Chip-RAM	142
A500 intern auf 1 MB RAM aufrüsten	143
A600-FastRAM-ERweiterung 4/8 MB	144
Amiga-500-Tastatur am CDTV	148
Floppy-Emulatoren	149
Elkos an SMD-Amigas tauschen	151
AT/X-Netzteil für A1000	153
TICK-Signal	155
Externe Floppy als DF0:	156
Verkaufszahlen Commodore Deutschland	157
Amiga am TFT anschließen	158
TV-Modulator A520 auf S-Video umbauen	160
Minimig	162
Minimig auf Altera-DE1-Board	163
PS/2-Tastaturbelegung	166
Micro KBD – Micro-Tastaturersatz für Minimig	166
AVR-Applikationen	170
Magic Light	172
Tiny BASIC im ATmega8 auf Pollin Eval. Board	174
AVR Nova Strike (Galaxian)	175
AVR-CP/M-Minirechner	177
Tools und Utilitys	183
Emulatoren	185
Weiterführende Webadressen	187
Kleine Bauteilkunde	189
Danksagungen	193
Über den Autor	194

Vorwort

Wir leben in einer schnelllebigen Zeit. Die technische Entwicklung schreitet mit großen Schritten voran. Wer erinnert sich heute noch an Musikkassetten oder Schallplatten – im Zeitalter des Internetradios, MP3 und Blu-Ray? Schnell geraten heute technisch obsolete Geräte in Vergessenheit. Und doch sind diese Geräte und die damaligen Entwicklungen die Wurzeln der heutigen Technik.

Dem »Vergessen« entgegenzuwirken haben sich nicht nur Museen verschrieben, sondern kleinere Gruppen – meist über das Internet verbunden – halten die damalige Technik am »Leben«.

In diesem Buch geht es im Besonderen um C64 und Amiga 500 – Heimcomputer der 1980er/90er Jahre. Dabei soll es eben kein Buch über die Historie dieser Computer »for the masses« werden, davon gibt es schon einige exzellente Bücher auf dem Markt, sondern es soll im speziellen auf einzelne, aktuelle Entwicklungen zu diesen Systemen hingewiesen werden. Auch die Restauration solcher »antiker« Rechentechnik soll hier und da aufgezeigt werden.

Deshalb auch eine Bitte: selbst defekte Geräte haben als Ersatzteilträger einen »Wert«. Nicht wegwerfen, sondern den entsprechenden Gruppen zugänglich machen!

Ich möchte Projekte und Kontaktadressen aufzeigen. Eventuell wird dadurch bei dem einen oder anderen Leser die Leidenschaft (wieder) erweckt, sich mit diesen Themen zu beschäftigen.

Durch die oben genannte Schnelllebigkeit können im Buch vermittelte Informationen inzwischen schon wieder veraltet, Internet-Links nicht mehr gültig sein. Dann sollte eine der aktuellen Internetsuchmaschinen helfen können, die entsprechenden Informationen und Webseiten wiederzufinden.

Alle hier aufgeführten Verfahren, Bauvorschläge etc. sind nach bestem Wissen und Gewissen wiedergegeben und zum Teil selbst nachgebaut worden. Eine Gewährleistung kann aber weder von Autor noch Verlag in keinerlei Hinsicht gegeben werden. Nachbau und Betrieb erfolgen auf eigene Gefahr! Die notwendigen Kenntnisse im Umgang mit elektrischem Strom sowie den entsprechenden Werkzeugen und Materialien werden vorausgesetzt, die üblichen Sicherheitsmaßnahmen sind in jedem Fall zu beachten.

Der Commodore 64

Abb. 1:
C64-
Einschalt-
meldung



Der Commodore 64 (kurz: C64, umgangssprachlich auch 64er) ist ein 8-Bit-Heimcomputer mit 64 KB (Kilobytes; 64 x 1024 Bytes) Arbeitsspeicher.

Äußerst populär war der von Commodore gebaute C64 von Mitte bis Ende der 1980er-Jahre sowohl als Spielkonsole als auch

zur Softwareentwicklung. Er gilt mit über 17 Millionen verkauften Geräten weltweit als der meistverkaufte Heimcomputer aller Zeiten.

Im Gegensatz zu modernen PCs verfügte der C64 über keine internen Massenspeichergeräte. Alle Programme mussten von externen Laufwerken wie dem Kassettenlaufwerk Datasette oder dem 5¼-Zoll-Diskettenlaufwerk 1541 geladen werden. Nach dem Einschalten wurde man mit einem sog. BASIC-Prompt begrüßt.

Der C64 bot viele Schnittstellen und war daher auch für Bastler und Hardwaretüftler interessant. Für die Spiele waren die – für damalige Verhältnisse – guten Grafikfähigkeiten und der auch heute noch großartige Sound über den SID-Chip verantwortlich.

Durch die sehr große verkaufte Stückzahl ist ein C64-System auch heute noch über Online-Auktionshäuser und Gebrauchtmärkte günstig zu bekommen und eignet sich daher ideal als Bastler- und Restaurationsobjekt, da sich bei eventuellen Fehlschlägen finanzielle Verluste in Grenzen halten.

Abb. 2:
meistver-
kaufter
Heimcom-
puter: C64



Wenn der C64 nicht mehr funktioniert

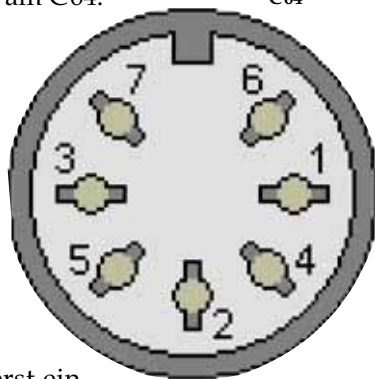
1. Zuerst einmal alles abziehen (IMMER im stromlosen Zustand!): Floppykabel, Module, Resettaster extern, Datasette, Joysticks, Maus etc. Auch das Netzteil, das sollten wir erst einmal separat prüfen.

2. Netzteil prüfen. Das Netzteil liefert 5 Volt Gleichspannung (DC) und 9 Volt Wechselspannung (AC). Die 5 Volt des externen Netzteils sollten mit einem Strom von mindestens 1,0 Ampere belastet werden können.

Die 9 Volt Wechselspannung dienen der internen Uhr als Referenzfrequenz und werden im C64 nach Gleichrichtung auch für die Spannungsversorgung des SID-Chips (9 oder 12 Volt) und des Motors für die Datasette (6 Volt) verwendet. Belegung der Buchse am C64:

Abb. 3:
Netzteil-
buchse des
C64

<u>Pin</u>	<u>Signal</u>	<u>Bemerkung</u>
1	GND	Masse
2	GND	Masse
3	GND	Masse
4	—	unbelegt, evtl. auch 5 Volt
5	+ 5V DC	5 Volt Gleichspannung
6	9V AC	9 Volt Wechselspannung
7	9V AC	9 Volt Wechselspannung



Benötigtes Werkzeug

- Schraubendreher
- Digitalmultimeter
- funktionierendes Spielmodul
- Logiktester (High/Low/Pulse) oder Oszilloskop
- Entlötwerkzeug (Pumpe, Litze und Kanüle)
- Lötstation geregelt
- IC-Sockel
- getestete Ersatz-Chips (ohne solche Chips kann man es vergessen! Dazu benötigt man einen C64, bei dem alle großen ICs und mindestens ein RAM-Chip gesockelt sind. An ihm können ausgelötete Chips dann getestet werden.)

Hier also erst einmal die Spannungen des Netzteils prüfen. Die 5V sollten im Bereich 4,95 bis 5,05 liegen. Andernfalls die Sicherung (schwarze Schraubverbindung) prüfen und ggf. ersetzen: 250 V/ Träge/200 mA (C64-I = 150 mA)

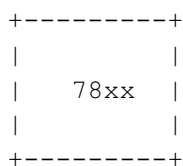
3. Wenn das Netzteil ok ist, kann man es mit dem C64 verbinden. Für die Überprüfung muss der C64 geöffnet werden. Beim C64-I gibt es an der Gehäuseunterseite vorn drei Schrauben, die gelöst werden müssen. Danach kann das Gehäuse ein-

fach nach oben aufgeklappt werden. Man findet eine Abschirmpappe, die mit Alufolie bezogen ist. Diese Pappe ist hinten rechts an einem Abschirmblech festgeklemt. Dort kann man die Klammer einfach nach vorn abziehen und die Pappe nach oben wegklappen. Nun wird der Blick auf die Sicherung an der rechten Seite des Boards frei.

Der C64C wird durch drei Kunststoffklammern an der Unterseite des Gehäuses zusammengehalten. Diese kann man einfach mit einem Schlitzschraubendreher auseinander biegen und dabei das Gehäuse aufziehen. Nachdem die Gehäuseoberhälfte nach oben entfernt wurde (Tastatur und LED einfach am Stecker abziehen), muss nun wieder die Abschirmpappe weggeklappt werden (Einige sind auch der Meinung, die sollte man ganz weglassen, um die Wärmeabfuhr zu fördern). Darunter befindet sich das Board mit der Sicherung auf der rechten Seite. Diese Sicherungen sind größer als die im Netzteil! Daten: 250 V / Träge / 1,5 A

Jetzt zuerst die große Sicherung herausnehmen und mit dem Multimeter auf Durchgang prüfen (Die Betriebs-LED leuchtet übrigens auch bei defekter interner Sicherung des C64!). Falls defekt, eine neue einsetzen! Jetzt einschalten und die Spannungen kontrollieren.

Es sind auf dem C64-Board noch zwei sog. 78xx-Festspannungsregler verbaut: 1x 7805 = 5 V und 1x 7812 = 12 V. Die Eingangsspannung sollte mindestens 2,5 V höher als die gewünschte Ausgangsspannung sein.



```

#  #  #
#  #  #
#  #  #

```

In GND Out

In = Ungeregelter Eingang +
 GND = Masse
 Out = Geregelter Ausgang +

Sollten die Spannungen nicht korrekt sein, so sind die entsprechenden Elkos und die Regler zu tauschen. (Auch hier sind einige der Meinung, Elkos und Regler sollte man in jedem Fall nach 15 bis 20 Jahren tauschen. Elkos altern!)

Auch an Pin 6 = +5 V und Pin 21 = GND der CPU 6510 messen.

Dann noch mal über Antennenanschluss/Monitor prüfen, ob jetzt ein BASIC-Einschaltbild kommt.

Ab hier wird es komplizierter

Ein weiterer Test ist das blinde Eintippen von Floppy-Befehlen. Dazu Floppy mit C64 verbinden und einschalten. Die Floppy sollte einen Einschalttest durchführen. Dabei leuchtet die Power-LED (grün) ständig, die rote LED leuchtet für 2-3 Sekunden und dabei läuft der Floppymotor. Danach sind Floppymotor und rote LED wieder aus.

Jetzt den C64 einschalten. Die Floppy bekommt dabei ein Reset-Signal und wiederholt obige Prozedur. Wenn bis hierhin ok, blind einen Befehl wie zum Beispiel `LOAD "$", 8` eintippen und `<Return>` drücken. Läuft die Floppy an und arbeitet für ein paar Sekunden? Wenn ja, könnte beim C64 »nur« die Bildwiedergabe gestört sein. Kabel zum Monitor/Fernseher prüfen, ggf. gegen ein getestetes Kabel tauschen. Kanal am Fernseher korrekt eingestellt? Ggf. Suchlauf starten. Ansonsten weiter ...

So. Sicherungen sind alle ok. Spannungen sind alle ok. Aber der C64 tut es trotzdem nicht.

Falls nur SID und VIC gesockelt sein sollten, kommt man um ein Auslöten und Sockeln der ICs nicht herum! Dazu sollte man sich erst einmal an einer defekten Platine im Auslöten üben. Man sollte einen funktionierenden C64 haben, bei dem alle großen ICs und am besten auch noch ein RAM gesockelt sind. Alle ausgelöteten ICs sollten darin getestet werden und falls sie defekt sind, bitte sofort aussortieren!

Wichtig ist,, dass zwar ein defekter CIA/SID sehr wohl den C64 lahmlegen kann, dass aber auch ohne diese drei ICs eine Einschaltmeldung kommt! Also erst einmal weglassen, bis man eine Einschaltmeldung bekommt, dann eins nach dem anderen wieder einsetzen. Ich hoffe, es muss an dieser Stelle nicht extra erwähnt werden, dass das Einsetzen und Entfernen von Bauteilen IMMER im stromlosen Zustand erfolgen MUSS!

Der VIC (Video-Chip) befindet sich unter der silbernen Abschirmung. Er ist fast immer gesockelt und hier kann man ein getestetes Exemplar ausprobieren.

In 90 Prozent aller Fälle ist eine defekte PLA Schuld. Man kann also direkt diese austauschen (auslöten, Sockel einlöten, Ersatz-PLA einstecken) oder weiter versuchen, den Fehler einzugrenzen bzw. zu lokalisieren.

ren. Symptome können zum Beispiel sein: schwarzer Bildschirm, Bildschirm komplett in einer Farbe

Reset prüfen: An Pin 40 der CPU 6510 mit Logiktester messen. Beim Einschalten für 1-2 Sekunden = Low, danach High. Siehe Symptome und Ursachen

Takt prüfen: An Pin 39 der CPU muss der Systemtakt anliegen (ca. 1 MHz).

Ein Modul einstecken (immer im ausgeschalteten Zustand!) und einschalten.

Kommt jetzt ein Bild? Wenn ja, könnte das auf ein defektes BASIC-ROM hindeuten.

Werden Chips beim Betrieb sehr heiß (so dass man sich verbrennen kann) – insbesondere die RAM-Chips? Das könnte auf deren Defekt hindeuten. Falls man beispielsweise eine Einschaltmeldung mit 0 Bytes free oder ähnlichem bekommt, deutet das auf defektes RAM hin.

Defekte RAMs kann man lokalisieren, indem man intaktes RAM per Huckepack aufsetzt (auf den eingelöteten Chip einfach einen intakten aufsetzen, so dass alle Pins guten Kontakt haben. Ggf. die Pins des aufzusetzenden RAMs zuerst etwas zusammenbiegen). Kommt dann eine Einschaltmeldung? Achtung: nicht nur ein RAM nach dem anderen auf diese Weise testen, denn es könnte ja mehr als ein Chip defekt sein. Also

Zum Thema Auslöten

Ich verwende eine starke Entlötpumpe. Platine mit Lötseite nach oben auf den Arbeitstisch legen und fixieren. Lötstelle erhitzen und wenn Lötzinn gut verflüssigt ist, LötKolben schnell wegziehen, gespannte Entlötpumpe sofort auf dem Lötauge platzieren und Schuss. Sollte einmal der Pin nicht frei sein, besser zuerst wieder etwas frisches Lötzinn nachführen und dann sofort erneut absaugen. Auch hat es sich bewährt, eine nicht freie Lötstelle (ohne Pin) zu erhitzen und dabei gleichzeitig eine gelbe Kanüle (Einwegspritze; Apotheke) zu erwärmen. Dann mit der Kanüle das Lötloch durchstoßen.

am besten alle 8 RAMs vom Typ 4164 per Huckepack-Methode überbrücken.

Ansonsten würde ich jetzt in dieser Reihenfolge weiter sockeln und testen:

- Kernel-ROM
- BASIC-ROM
- Char-ROM
- CIA 1
- CIA 2
- RAMs
- TTLs

Symptome und Ursachen

C64 arbeitet soweit, aber Zugriff auf Floppy klappt nicht? Dann könnte der 7406 invertierende Bustreiber defekt sein.

C64 hat keine Farb- sondern nur S/W-Darstellung. Ist es ein NTSC-Gerät? (VIC und Quartz prüfen) Ansonsten könnte die 8701-Clock-IC defekt sein (neben dem VIC).

Kommen nach dem Einschalttest einfach Zeichen, so als würde jemand die Tastatur betätigen? Evtl. mal den Joystick herausziehen oder Auto-Fire ausschalten. Ansonsten CIA1 (neben dem Tastaturstecker) prüfen.

Die Einschaltmeldung zeigt nur 30912 Bytes an. Der U13 (74LS257/ MOS 7708) könnte defekt sein. Generell sind bei der Anzeige von weniger RAM als normal U13 und U15 Kandidaten für eine Überprüfung – und natürlich die RAM-Bausteine selbst.

C64 zeigt Einschaltbild mit Rahmen und richtigen Farben, aber keine Schrift und kein Prompt. Ein Spielmodul funktioniert jedoch einwandfrei: Das BASIC-ROM könnte defekt sein.

Wenn das Resetsignal an Pin 40 der CPU nicht in Ordnung ist, dann kann etwas mit U20 (NE556) und U8 (7406) falsch sein. An Pin 9 des NE556 müsste das Resetsignal invertiert anliegen. Danach geht das Signal zu Pin 9 vom 7406, wo es invertiert, verstärkt und dann der Resetleitung zugeführt wird.

Blinkende Farben und/oder Zeichen

Zum Test des Problems dient folgendes Programm:

```
10 for x=0 to 15
20 poke 55296, x
30 print x, " = ", peek(55296) and 15
40 next x
```

Das Ergebnis sollte lauten: 0 = 0, 1 = 1, 2 = 2, ..., 15 = 15. Erhalten Sie andere Werte, ist das Farb-RAM (2114) defekt.

"2047 (weniger als 38911) Bytes Free". Wahrscheinlich sind ein oder mehrere RAM-Chips defekt. Zum Test folgendes eingeben:

A=2048:POKE A,0:?PEEK(A):POKE A,255:?PEEK(A)

Das bedeutet, der Wert \$0/\$FF (%11111111) wurde in den Speicher in Adresse A geschrieben und danach wird zum Beispiel \$F7 (%11110111) gelesen.

In diesem Fall ist das Bit 3 nicht gesetzt, was bedeuten würde, dass in einem C64 mit 8 RAM-Bausteinen das IC U10 defekt ist.

Tabelle:

C64 mit 8 RAM-Chips: RAM-Chip-Typ 64kbit x 1: e.g. 4164

Bit	Chip-Typ
0	U21
1	U9
2	U22
3	U10
4	U23
5	U11
6	U24
7	U12

C64 mit 2 RAM-Chips: RAM-Chip-Typ 64kbit x 4: e.g. 41464

Bit	Chip-Typ
0,1,2,3	U10
4,5,6,7	U9 / U11

Wenn bei einem C64 (alte Boards mit 8x RAM) der obige Fehler auftritt, aber beim Einschalten immer mal unterschiedliche RAM-Angaben kommen, so kann es sein, dass der Multiplexer auf U13 (74LS257 / MOS 7708) defekt ist.

IEC-ATA V2 – IDE-Laufwerke am C64

Links:

<http://www.djupdal.org/cbm/iecata/>

<http://www.dienstagstreff.de>

Das IEC-ATA beruht auf einer Entwicklung aus 2002 von Asbjørn Djupdal. Seit ca. 2004 hat sich die Fan-Gruppe des sog. Dienstagstreffs dieser Entwicklung angenommen und sie weiterentwickelt. Dies führte dann schließlich zur Version V2 mit einem Atmega162 und einem entsprechenden Platinenlayout. Ab ca. 2005 entwickelte Jochen Adler (NLQ) dann eine eigenen Firmware, die das FAT32-Format der IDE-Massenpeicher unterstützt. Somit können Daten zwischen C64 und PC problemlos ausgetauscht werden.

Als Besonderheit im Vergleich zum MMC2IEC/SD2IEC ist es hier möglich, den Atmega162 direkt vom C64 aus mit seiner Firmware zu versehen!

Über die Webseiten des Dienstagstreffs sind weitere Information auch zu anderen interessanten Projekten verfügbar. Von Zeit zu Zeit wird dort auch eine Sammelbestellung von Bausätzen organisiert.



Abb. 4:
IEC-ATA V2
mit IDE-CF-
Adapter und
128-MB-CF-
Karte

C64-Bausteine (Chips)

Platinenversionen

Der C64 wurde in mehreren verschiedenen Platinenversionen gefertigt. Die einzelnen Versionen werden im Folgenden abgebildet.

Abb. 5:
C64-
Urplatine
aus dem Jahr
1982



Abb. 6:
Die sog. KU-
Platine zählt
ebenfalls
noch zu den
C64-Urplatinen





Abb. 7:
Assy 250407
ca. aus 1983/
84. PLA ne-
ben CPU,
SID darunter



Abb. 8:
Assy 250425
ab 1984.
Achtung: ge-
genüber
Version
250407 ha-
ben PLA und
SID die
Plätze ge-
tauscht.
Taktteilung
über 8701

Abb. 9:
Assy 250466



Abb. 10:
Assy 250469



Chip-Bezeichnungen

901225	4k Char-ROM
901226	8k BASIC-ROM
901227	8k Kernel-ROM
906114	PLA-Ersatz durch 27512-Eprom und Adaptersockel möglich!
8701	Clock Converter IC
4164	64kx1 DRAM (acht ICs im C64)
7707	7406
7708	74LS257
7709	74LS258
7711	74LS139
7712	74LS08

CIA 6526

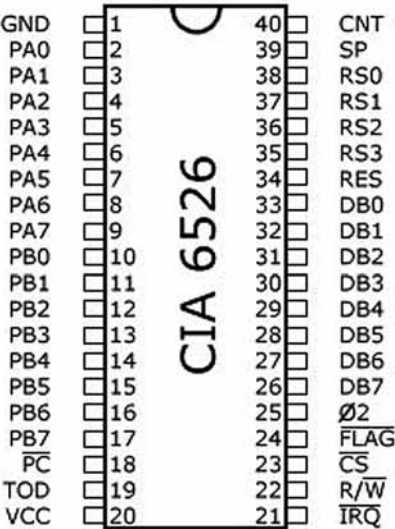
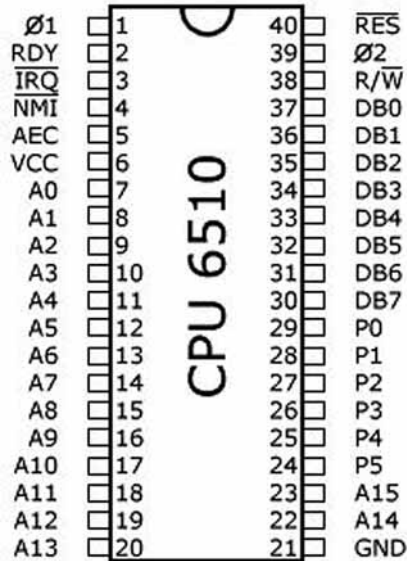


Abb. 11:
 (rechts oben)
 Pinbelegung
 der CIA 6526

CPU 6510



VIC 6567/9

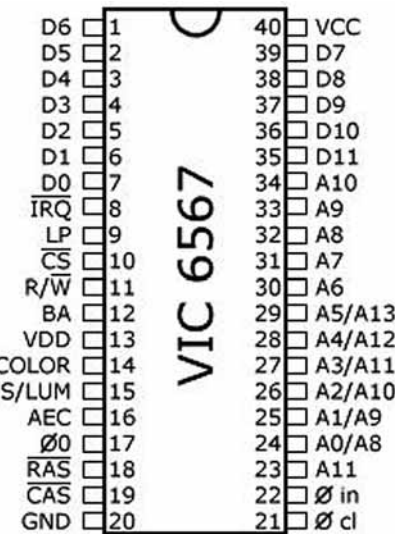
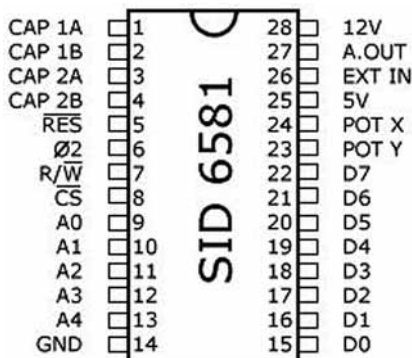


Abb. 12:
 (links unten)
 Pinbelegung
 der CPU
 6510

Abb. 13:
 (rechts un-
 ten)
 Pinbelegung
 des VIC 6567

Abb. 14:
Pinbelegung
des SID 6581

SID 6581



RAM 2114 (1kx4 SRAM)

+---+---+---+			
A6	1	+++	18 VCC
A5	2		17 A7
A4	3		16 A8
A3	4		15 A9
A0	5	2114	14 D1
A1	6		13 D2
A2	7		12 D3
/CE	8		11 D4
GND	9		10 /WE
+-----+			

Der SID des C64-II (8580) ist pin-kompatibel zum 6581. An PIN 28 sind dann aber nur 9 V und nicht 12 V anzulegen!

Wenn man einen 8580 dort betreiben möchte, wo eigentlich ein 6581 hineingehört, kann man einen ca. 137-Ohm-Widerstand zwischen PIN 28 des SID und PIN 28 der Fassung einfügen, um die 12 V auf ca. 9 V zu reduzieren. Dagegen kann ein 6581 problemlos in einem C64-C betrieben werden.

Hier gibt es ein Projekt, um den SID durch einen AVR zu ersetzen:
www.swinkels.tvtom.pl/swinsid

RAM 4116 (16kx1 DRAM)

Dieser Baustein benötigt -5, +5 und +12 V.

+---+---+---+			
VEE	1	+++	16 GND
D	2		15 /CAS
/WE	3		14 Q
/RAS	4	4116	13 A6
A0	5		12 A3
A2	6		11 A4
A1	7		10 A5
VDD	8		9 VCC
+-----+			

Verwendung eines 4164- statt eines 4116-Chips:

Pin 1 und 8 hochbiegen, so dass sie nicht mehr in die Fassung passen. Pin 8 und 9 mit leitendem Draht verbinden.

Pin 1 und 8 so abkneifen, dass keine anderen Bauteile berührt werden können.

Baustein dann so in die Fassung einsetzen.

RAM 4164 (64kx1 DRAM)

+---+---+---+			
	1	+---+ 16	GND
D	2		15 /CAS
/WE	3		14 Q
/RAS	4	4164	13 A6
A0	5		12 A3
A2	6		11 A4
A1	7		10 A5
VCC	8		9 A7
+-----+			

RAM 41256 und 41257 (256kx1 DRAM)

+---+---+---+			
A8	1	+---+ 16	GND
D	2		15 /CAS
/WE	3		14 Q
/RAS	4	41256	13 A6
A0	5	41257	12 A3
A2	6		11 A4
A1	7		10 A5
VCC	8		9 A7
+-----+			

Statt eines 4164 kann auch ein
41256 verwendet werden.
Dazu einfach ein Kabel zwischen
Pin 1 und 16 anlöten.

RAM 4464 (64kx4 DRAM)

+---+---+---+			
/OE	1	+---+ 18	GND
D0	2		17 D3
D1	3		16 /CAS
/WE	4		15 D2
/RAS	5	4464	14 A0
A6	6		13 A1
A5	7		12 A2
A4	8		11 A3
VCC	9		10 A7
+-----+			

RAM 44256 und 44258 (256kx4 DRAM)

+---+---+---+			
D0	1	+---+ 20	GND
D1	2		19 D3
/WE	3		18 D2
/RAS	4		17 /CAS
	5	44256	16 /OE
A0	6	44258	15 A8
A1	7		14 A7
A2	8		13 A6
A3	9		12 A5
VCC	10		11 A4
+-----+			

ROM

Der C64 (alte Bauform) besitzt 3 ROMs: U3 = 8k BASIC Interpreter, U4 = 8k Kernel und U5 = Character ROM. Dabei handelt es sich um masken-programmierte Chips mit diesen MOS-Nummern:

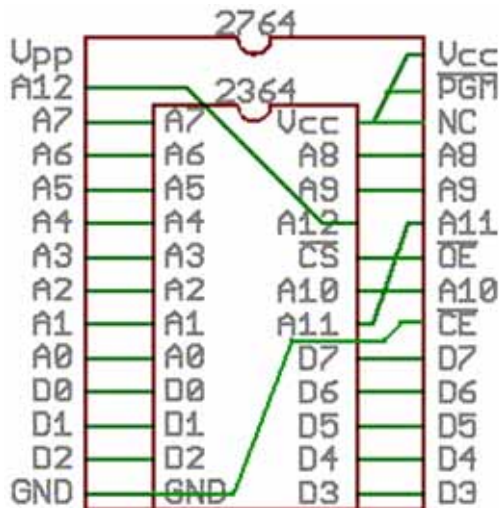
<u>IC</u>	<u>MOS-</u> <u>part</u>	<u>Memory</u> <u>region</u>	<u>Standard</u>	<u>ROM Type</u> <u>PROM part</u>
U3	901226-01	0xA000-0xBFFF	2364A	BASIC
U4	901227-03	0xE000-0xFFFF	2364A	Kernel
U5	901225-01	0xD000-0xDFFF	2332A	Character

EPROM als ROM-Ersatz

Hier eine Adapterschaltung, um Standard-2764-Eproms als ROM-Ersatz zu verwenden:

- vom 2364-Pin 24 = Vcc mit 2764-Pin 28, 27, 26 verbinden
- vom 2364-Pin 21 = A12 mit 2764-Pin 2 verbinden
- vom 2364-Pin 18 = A11 mit 2764-Pin 23 verbinden
- Pin 20 am 2764 /CE mit GND verbinden (Pin 14 am 2764)

Abb. 15:
ROM-
EPROM-
Adapter



EPROM-Pinouts

2764			
+-----\ /-----+			
1	- Vpp	Vcc -	28
2	- A12	/pgm -	27
3	- A7	nc -	26
4	- A6	A8 -	25
5	- A5	A9 -	24
6	- A4	A11 -	23
7	- A3	/OE -	22
8	- A2	A10 -	21
9	- A1	/CE -	20
10	- A0	D7 -	19
11	- D0	D6 -	18
12	- D1	D5 -	17
13	- D2	D4 -	16
14	- Vss	D3 -	15
+-----+-----+			

27128			
+-----\ /-----+			
1	- Vpp	Vcc -	28
2	- A12	/pgm -	27
3	- A7	A13 -	26
4	- A6	A8 -	25
5	- A5	A9 -	24
6	- A4	A11 -	23
7	- A3	/OE -	22
8	- A2	A10 -	21
9	- A1	/CE -	20
10	- A0	D7 -	19
11	- D0	D6 -	18
12	- D1	D5 -	17
13	- D2	D4 -	16
14	- gnd	D3 -	15
+-----+-----+			

```

                27256
            +-----\ /-----+
1  -|Vpp          Vcc|- 28
2  -|A12          A14|- 27
3  -|A7           A13|- 26
4  -|A6           A8|- 25
5  -|A5           A9|- 24
6  -|A4           A11|- 23
7  -|A3           /0E|- 22
8  -|A2           A10|- 21
9  -|A1   /CE   /pgm|- 20
10 -|A0           D7|- 19
11 -|D0           D6|- 18
12 -|D1           D5|- 17
13 -|D2           D4|- 16
14 -|gnd          D3|- 15
            +-----+

```

```

                27512
            +-----\ /-----+
1  -|A15          Vcc|- 28
2  -|A12          A14|- 27
3  -|A7           A13|- 26
4  -|A6           A8|- 25
5  -|A5           A9|- 24
6  -|A4           A11|- 23
7  -|A3   /0E-Vpp|- 22
8  -|A2           A10|- 21
9  -|A1   /CE-/PGM|- 20
10 -|A0           D7|- 19
11 -|D0           D6|- 18
12 -|D1           D5|- 17
13 -|D2           D4|- 16
14 -|gnd          D3|- 15
            +-----+

```

C64-Ersatz-PLA auf 27512-EPROM-Basis

Dieser Adapter basiert auf der Arbeit von Jens Schönfeld (Geschäftsführer der *Individual Computers GmbH*). Im August 1994 hat er als Zwischenschritt zur Logikanalyse des PLA-Chips einen Adapter gebaut, der das Auslesen mit einem EPROM-Programmiergerät ermöglichte. Die so entstandene Datei wurde im Z-Netz (einem Vorläufer der Internet-Foren) veröffentlicht. Später gab es auch von anderen Freaks Versuche zur Analyse. Im Sommer 2002 veröffentlichte Jens Schönfeld die Logik-Gleichungen der C64-PLA. Sie sind im Funet-Archiv zu finden (kostenlos im Netz oder im Handel auf CD). Dort sind auch weitere EPROM-Dateien zu finden, die mit dem von Schönfeld entwickelten Adapter angefertigt wurden – Commodore hat den Logikchip 82S100 mit jeweils unterschiedlicher Programmierung in vielen verschiedenen Computertypen eingesetzt.

Der hier vorgestellte Adapter nutzt die Datei, die ursprünglich als Zwischenschritt gedacht war, als Quelle für eine Ersatzschaltung. Dabei wird außer Acht gelassen, dass ein EPROM eigentlich nicht als Logik-Ersatz geeignet ist: Während der Zugriffszeit von bis zu 120 ns (also gut einem Viertel des CPU-Zyklus' des C64) ist der Zustand der acht Ausgangsleitungen nicht garantiert. Unter Umständen kann es vorkommen, dass zwei oder mehr Bausteine des C64 die Freigabe bekommen, den Datenbus des Rechners zu benutzen. In den meisten Fällen geht das gut, Langzeitschäden sind jedoch nicht auszuschließen. Die EPROM-Schaltung sollte daher nur als Übergang oder zur Diagnose eingesetzt werden. Für eine dauerhafte Lösung ist der Bau einer Ersatzschaltung auf Basis eines programmierbaren Logikchips dringend anzuraten.

Wenn der C64 (Ur-Modell, Bj. 1982-1984) den Geist aufgibt, ist zu 90 % die PLA platt. Der Baustein hat die Bezeichnung 906114 bzw. 82S100. Abhilfe schafft die »Bastel-PLA«, die aus einer Adapterplatine und einem 64K-EPROM besteht.

Benötigt wird dazu ein EPROM 27C512-45 ns, wobei bei mir 120 ns, 150 ns, 200 ns und 255 ns ohne sichtbare Probleme liefen – also einfach testen, ihr braucht nur eine Adapterplatine oder einen frei verlöteten Adaptersockel und natürlich die passenden Daten. Im Web: <http://peter-sieg.webng.com/pla/27512.zip>

Das sollte sich auch »fliegend« mit Zwischensockel aufbauen lassen.

Dazu folgende Erklärungen:

In den Sockel des C64-Boards kommt ein Zwischensockel (normaler IC-DIL28-Sockel). An den Pins 10, 19, 20 und 22 wird ein isoliertes Kabel bzw. eine Litze angelötet, so dass man das andere Ende später an die abgebogenen Kontakte des 27512-Eproms löten kann. Dann kommt das (programmierte!) 27512-EPROM auf

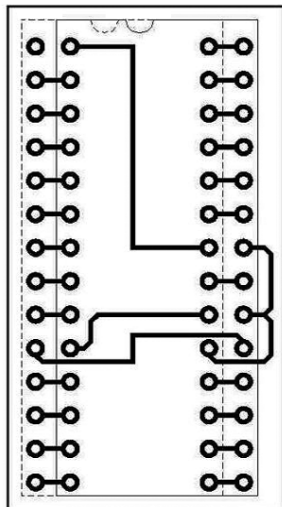


Abb. 16:
Verbindungen der Ersatz-PLA

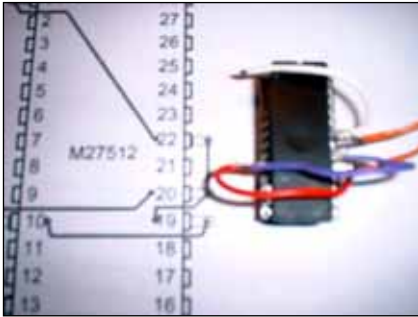
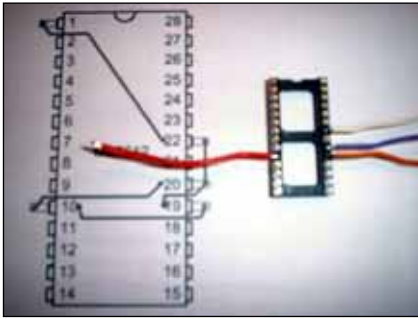


Abb. 17:
(oben links)
So werden
die Kabel
angelötet

Abb. 18:
(unten links)
fertig be-
stückter
Sockel mit
EPROM

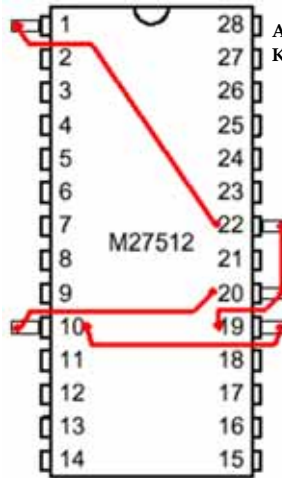


Abb. 16:
Kontaktskizze

den Zwischensockel, wobei am Eprom die Pins 1, 10, 19, 20 und 22 zur Seite gebogen werden, so dass sie keinen Kontakt mit dem Zwischensockel haben. Dann werden verbunden:

Zwischensockel <-> Eprom

10 <-> 19

19 <-> 20+22

20 <-> 10

22 <-> 1

Schritt für Schritt

1. Vier Kabel (ca. 5 cm) am Präzisionssockel an die Pins 10, 19, 20 und 22 anlöten
2. Ein 27C512 mit den korrekten Daten brennen
3. Die Pins 1, 10, 19, 20 und 22 abbiegen, so dass sie keinen Kontakt zum Sockel haben
4. Die vier Kabel wie oben ausgeführt mit den Eprom-Pins verbinden

Ersatz-PLA auf CPLD-Basis

Basierend auf den Arbeiten von Jens Schönfeld und Mark Smith hat Armin Läger die Logikgleichungen für die C64-PLA auf das CPLD-Board von Fpgaarcade portiert.

Das CPLD-Board besitzt den Formfaktor DIL28 und kann als Ersatz verschiedener ICs programmiert werden.

Referenzen

- http://vic20.de/html/eprom_pla_8296_und_c64.html
- <http://www.zimmers.net/cbmpics/cbm/c64/pla.txt>
- <http://www.zimmers.net/anonftp/pub/cbm/firmware/computers/c64/>



Abb. 19:
So sieht das
CPLD-Board
aus

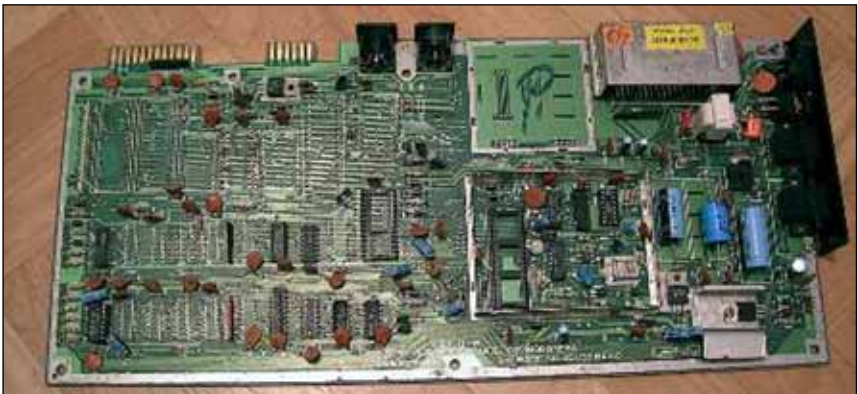
Platinenrestauration

Wiederbelebung eines Assy 250407

Diese großartige Wiederbelebung wurde ausgeführt und wird hier mit freundlicher Genehmigung wiedergegeben von Andreas Senk (Jerry Maus). Hier sein Bericht:

Ich habe diese Platine Ende der 1980er bekommen. Der vorherige Eigentümer hat versucht, das Kernel auszulöten und dabei mehrere Durchkontaktierungen und Leiterbahnen abgerissen. Gehäuse, Tastatur und viele der Bauteile sind über die Jahre hinweg anderen 64ern zugute gekommen. Da nun so ziemlich alles Brauchbare nicht mehr vorhanden war, wollte ich den Rest eigentlich entsorgen. So sah die Platine vor Beginn der Restauration aus:

Abb. 20:
Defekte
Assy 250407
vor der Res-
taurierung



Inspiziert durch Pentagons Bericht über die Rettung eines C64G habe ich es dann doch nicht getan, sondern mich zu einem Rettungsversuch entschlossen.

Nach Reinigen der Platine und Abschleifen der defekten Leiterbahnen mit einem Glasfaserpinsel war der Schaden gut zu sehen:



Abb. 21:
beschädigte
Lötstellen

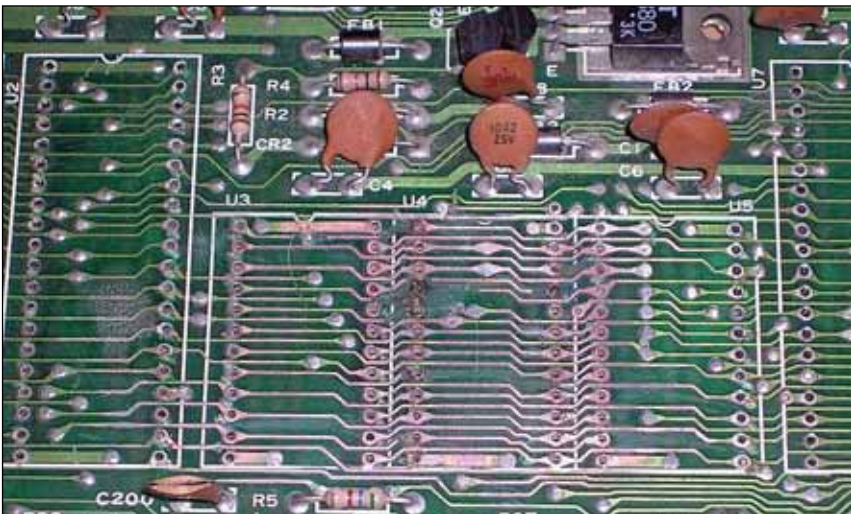


Abb. 22:
weitere
Schäden

Nach Durchmessen der neuen Verbindungen wurde das Ganze zwecks Isolation und Schutz vor Korrosion mit einem durchsichtigen Schutzlack überzogen. Nachdem er trocken war, habe ich den Sockel eingelötet. Im Zuge dessen wurden auch gleich die alten, noch vorhandenen Sockel durch Präzisionsfassungen ersetzt:

Abb. 23:
gesäuberte
Platine mit
Präzisions-
fassungen



Und so schaut nun die Rückseite nach dem Reinigen, Löten und Versiegeln aus:

Abb. 24:
gesäuberte
Platine von
unten



Die Bauteile habe ich aus meinem Fundus zusammengesucht, eine CIA und den SID jedoch vorerst einem anderen Rechner entliehen (siehe Abbildung folgende Seite).



Abb. 25:
die fertig
bestückte
Platine

Der anschließende Funktionstest fiel nicht so gut aus. Es hatte sich noch eine Unterbrechung bei den Speicherbausteinen eingeschlichen. Diese war aber schnell gefunden. Die Bauteile hatte ich natürlich vorher in einem anderen Cevie einzeln auf Funktion getestet. Und siehe da:



Abb. 26:
erfolgrei-
cher
Funktions-
test der
Platine

Ein brauchbares Gehäuse sowie eine Tastatur fehlen leider noch. Aber die Platine hat schon mal ein neues Leben vor sich.
(Andreas Senk)

C64-Restauration von Pentagon

Diese großartige Restauration wurde ausgeführt von Thomas Wirtzmann (Pentagon). Und hier kommt die Story:

Nach langer Zeit hat mich unser geschätzter Doc64 letzte Woche mal wieder zu Hause besucht. Wie es sich gehört, hat er mir als freundliches Gastgeschenk einen Commodore 64 mitgebracht. Es war ein wunderschöner C64G aus dem Jahr 1990. Und die Freude meinerseits war natürlich riesengroß. Na ja, nicht ganz: wie man sich denken kann, rückt der Doc nicht so ohne weiteres einen Cevie raus, oder? Leider hatte der Rechner diverse Probleme ...

Manfred (der Doc) war am Morgen zu einem Bekannten gefahren und hatte das Gerät dort im Keller gefunden, wo es zwischen Müllbergen, Mäusekot, Schimmel und Dreck sein trauriges Endzeitstadium fristete. Da ihm das gute Stück extrem leid tat, hat er es kurzerhand eingepackt und beschlossen, mir den Rechner als Geschenk mitzubringen, da ich immer auf der Suche nach Ersatzteilen bin.

Abb. 27:
Pentagons
Rechner vor
der Restau-
rierung



Als ich das Gerät dann in die Hände gedrückt bekam, war mein Entsetzen groß: So etwas hatte ich nämlich noch nie im Leben gesehen. Abgesehen von dem bestialischen Gestank von Moder und Kleintierexkrementen war das Teil

einfach nur ekelhaft. Mit spitzen Fingern haben wir dann das Gerät aufgeschraubt und wollten die Platine zum späteren Schlachten ausbauen. Nach dem Öffnen des Rechners kam der nächste Schock: Die Platine hatte einen Schaden durch eingedrungene Flüssigkeit, vermutlich Cola. Außerdem war noch eine Art stinkender klebriger Leim rund um den Modulport geschmiert, welche so ekelhaft gerochen hat. Es würde mich nicht wundern, wenn es eine Verwesung war.

Wir haben die Platine trotzdem ausgebaut und das ekelhafte Gehäuse in meine Restmülltonne befördert, denn es war einfach nur noch schrott-reifer Müll.

Die Platine wurde sofort bei uns ins Badezimmer gebracht, wo sie sogleich in 70 Grad heißes Wasser getaucht wurde. Es war ein Jammer, aber der Geruch war unter Wasser nicht mehr wahrzunehmen. Dann haben wir ein Schaumbad aus Spüli und Duschgel zubereitet und die Platine nach allen Regeln der Kunst mit Zahnbürste und Nagelbürste geputzt.

Nach knapp 40 Minuten Dauerschrubben war die Platine dann soweit sauber und mir war zum schwallartigen Erbrechen zumute. So was Ekelhaftes habe ich noch geputzt. Das Blech vom Modulport war so verrostet und angefressen, dass es große Blasen geworfen hat. Die Drosselpule war weggefressen, der Schalter ließ sich überhaupt nicht mehr richtig bewegen und stank selbst nach dem Bad noch ungemain.

Die Platine habe ich dann in einem Zimmer auf die Heizung gelegt, wir haben uns anschließend einen schönen Abend gemacht und 80er-Jahre Videos geschaut. Als Manfred dann später nach Hause fuhr, habe ich mir die trockene Platine angesehen und festgestellt, dass diese erst 17 Jahre alt war, also relativ jung für einen C64. Es waren nur CSG-Chips aus dem Jahr 1990 verbaut und das Herstellungsdatum war Kalenderwoche 40/1990.

Ich bin dann ziemlich müde ins Bett gekrabbelt und hatte eine sehr unruhige Nacht. Irgendwie hatte ich ein schlechtes Gewissen bekommen, dass ich das C64-Gehäuse so einfach in den Müll geworfen hatte. Meine

Frau hat mich für total bescheuert erklärt und sagte, dass dieses Ekelteil auf jeden Fall in der Mülltonne bleibt und entsorgt wird.

Früh morgens war es dann soweit. Meine Unruhe war so groß, dass ich mit meiner Schlafanzug hose und einem Pulli an die Mülltonne geschlichen bin.

Dann öffnete ich den Deckel der stinkenden Tonne und nahm das widerliche Gehäuse wieder raus. Im gleichen Moment kam mei-



Abb. 28:
das Innere
des Rechners



Abb. 29:
die un-
brauchbare
Tastatur

ne Nachbarin um die Ecke und hat mich mit dem Ding an der Tonne stehen sehen. Nachdem sie dann noch meine Schlafanzughose wahrnahm, hat sie den Kopf geschüttelt und ist fassungslos wieder zurück ins Haus gegangen. Ihre Gedanken konnte man förmlich am Gesichtsausdruck ablesen.

Nachdem ich das gute Stück dann mit ins Haus genommen hatte, habe ich es erst einmal in die Badewanne verfrachtet und auseinander genommen. Im oberen Gehäusedeckel war alles voll Müll, Dreck und Mäusekot.

Abb. 30:
Dieses Reinigungsbad
war absolut
notwendig!



Abb. 31:
das Gehäuse
des Rechners



Abb. 32:
erster Lohn
der Mühe:
eine funktionierende
Platine



Die Tastatur war total verklebt und ein schleimiger Brei klebte rund um die F-Tasten und über der Return-Taste. Das Ganze war so verharzt, dass man die Tasten nicht mehr bewegen konnte.

Nach dem Ausbau der Tastatur habe ich dann das Gehäuse in der Badewanne mit der chemischen Keule gereinigt. Ich habe einen Spezialschaum eingesetzt, welcher auch in die Poren einzieht und gleichzeitig leicht bleichend wirkt.

(Wasserpfeifenreiniger Marke »Schmand Weg« in lauwarmem Wasser soll hier gute Ergebnisse bringen.)

Nach dem Einsatz der chemischen Keule wurde das Gehäuse dann geschrubbt, und das erste Ergebnis war schon mal nicht schlecht – wenn auch noch lange nicht perfekt. Nach fast zweistündigem Schrubben, Putzen,

Bleichen, Trocknen und Polieren war ich mit dem Ergebnis soweit zufrieden.

Jetzt habe ich mich der Platine gewidmet. Die Platine war defekt und funktionierte überhaupt nicht mehr. Also habe ich den Schalter abgelötet, die Spule entfernt und das Modublech ebenfalls entfernt. Den verfaulten Sicherungshalter habe ich ausgelötet und ersetzt.

Der Schalter wurde aufgebogen und zerlegt. Im inneren befanden sich noch Wasser und klebriger Brei, welcher so zäh war, dass man ihn nicht entfernen konnte. Also habe ich den Schalter ausgetauscht und einen neuen eingelötet. Die Platine war soweit wieder sauber und sah aus wie gestern produziert. Die Spule habe ich neu eingelötet und die Sicherung ersetzt. Die gesockelten Chips habe ich gereinigt und neu eingesetzt.

Jetzt kam der große Augenblick: Ich habe die Hauptplatine am 1701 angeschlossen, eingeschaltet und: sie funktionierte wieder!

Es geht weiter

Ein Modulportblech befand sich auch noch in meinem Ersatzteilbestand, zwar vom C64E, aber egal: Hauptsache, es passt.

Sieht doch wieder sehr gut aus und funktioniert auch wieder einwandfrei. Der Modulport war mit Grünspan oxidiert und wurde daher mit dem Dremel und einem Zahnarztpolierer wieder vorsichtig auf Hochglanz gebracht. Steckmodule funktionierten auch wieder einwandfrei.

Jetzt konnte die fertige Hauptplatine wieder in die Unterschale eingebaut werden. Die Unterschale hatte auch einen Defekt: Beim Ausbau der Platine ist oben rechts eine Gewindeführung abgebrochen. Diese hing noch am Metallblech der Ports. Also habe ich sie mit Sekundenkleber wieder eingeklebt und die Naht mit einer kleinen Lötkolbenspitze verschweißt. Das hält auf jeden Fall wieder sehr gut und die Platine konnte einwandfrei verschraubt werden. Selbstverständlich habe ich die verrosteten Schrauben durch neue ersetzt.

Die Silberpappe für die Abschirmung war leider total verschimmelt und konnte nicht weiter verwendet werden. Also verzichtete ich eben darauf. Wird auch ohne Pappe funktionieren.

Jetzt habe ich die Tastatur in der Badewanne mit dem Schaumreiniger eingesprüht und einwirken lassen. Das war wirklich extrem ekelhaft: Beim Abspülen offenbarte sich der gesamte Dreck der letzten 17 Jahre. Dort waren tote Insekten, Würmer, Dreck, Mäusekot und Abfallteile

zwischen den Tasten. Es wurde immer mehr Dreck. Unglaublich, wieviel Dreck in so einer Tastatur vorhanden sein kann. Das war sowas von krass, dafür gibt's keine Worte.

Nach dem ersten Abspülen habe ich dann die Tastenkappen entfernt und die Tastatur komplett zerlegt. Dabei hätte ich mich wirklich fast übergeben müssen, denn im Inneren der Tastatur befanden sich kleine Würmer und winzige Käfer, die sich dort einen Lebensraum geschaffen hatten. Ich hätte das nie für möglich gehalten, aber diese Insekten haben dort auf der Platine lauter Eier abgelegt und – es lebte! Hier kommt der widerliche Beweis:

Die Tastenkappen habe ich dann in Oxycleanbad gegeben und über Nacht stehen lassen. Das Oxyclean löst den Dreck auf und bleicht die Tasten.

Leider war die Bleichwirkung so gering, dass ich zu härteren chemischen Mitteln greifen musste. Ich habe reine Chlortabletten mit kochendem Wasser aufgegossen und eine 70-prozentige Chlorklösung gebraut. Das Ganze noch mit einer Flasche Domestos aufgegossen, und fertig war die Chlorgasbombe.

Abb. 33:
eine »leben-
de« Tastatur!



Abb. 34:
Ist schon
eklig, oder?



Das Ganze habe ich natürlich draußen gemacht, denn der Gestank ist nicht nur unerträglich, sondern auch hochgiftig. Also bitte nicht zu Hause nachmachen!

Mein Nachbar war über den penetranten Schwimmbadgeruch ganz verwundert. Ich habe mir nichts anmerken lassen und innerlich gelacht. (Anmerkung: Vorsicht mit solchen chemischen Substanzen! Sicherheitshinweise beachten!)

Die Tasten haben es überlebt und sind schön gebleicht. Leider hat mir das Ergebnis trotzdem nicht ge-

fallen. Also habe ich mich entschlossen, einen Satz neuer Tastenkappen zu verwenden.

Nach Reinigung und Aufbereitung der Tastatur konnte diese wieder in das Gehäuseoberteil eingebaut werden. Die LED des Cevie war auch hinüber. Sie war so verrostet, dass sich der Rost bis ins Innere gefressen hatte. Da ich keine grünen LEDs im Haus hatte, musste ich notgedrungen eine blaue verwenden. Ich hoffe, man verzeiht mir das »Modding«. Die Kabel zur LED wurden ebenfalls ausgetauscht, da sie ebenfalls brüchig vor Korrosion waren. Neuer Schrumpfschlauch wurde ebenfalls »drumgeschrumpft«. Nach der Montage des Geräts sah es dann so aus: Das Gerät war nun nach langer Arbeit endlich fertiggestellt.

Nicht ohne Stolz habe ich es dann sofort angeschlossen und jetzt seit vier Stunden im Dauerbetrieb. Als meine Frau dann von der Arbeit nach Hause kam, wurde das Gerät sofort besetzt und mit großer Freude begrüßt.

Alles in Allem ist es doch sehr schön geworden.

Jemand anderes hätte das Ding mit Sicherheit im Müll gelassen. Aber wie man sieht, lohnen sich Ausdauer und Fleiß, und man bekommt ein gutes Ergebnis.

Vielleicht regt Euch das Projekt an, vielleicht haltet Ihr mich für bescheuert. Der Cevie funktioniert auf jeden Fall wieder.

HAPPY BIRTHDAY!

Pentagon



Abb. 35:
das
Endergebnis –
ein fast
neuer C64 ...



Abb. 36:
... der auch
noch funk-
tioniert!

Großumbau C64: Verringerung der Stromaufnahme, lebensverlängernde Maßnahmen

Dieser großartige Umbau wurde vorgenommen von Peter Trachte (AREA51HT).

Generell sollte man ein neues Netzteil am C64 einsetzen. Da hier aber mit Netzspannungen hantiert werden muss, sollte dies nur von einem Fachmann ausgeführt werden! Die Belegung der Netzteilbuchse/des Kabels ist aus der kleinen Reparaturanleitung zu entnehmen. Es werden +5 V, ca. 1,5 A und ~9 V ca. 1 A benötigt.

Der Umbau ist nicht für das seltene Sammlerstück gedacht, sondern für die Arbeitsmaschine, die viel läuft. Ziel der vielen kleinen Baustellen ist es, die Verlustleistung im Inneren des Gehäuses zu reduzieren, um die Betriebstemperatur der Chips abzusenken und dadurch die Lebensdauer zu maximieren. Was bringt der Umbau?

Stromaufnahme vor dem Umbau (Original):

$$9V / 0,65A = 5,85 \text{ Watt}$$

$$5V / 0,85A = 4,25 \text{ Watt}$$

10,10 Watt Gesamtleistung

Stromaufnahme nach dem Umbau:

$$9V / 0,326A = 2,94 \text{ Watt}$$

$$5V / 0,400A = 2,00 \text{ Watt}$$

4,94 Watt Gesamtleistung

Die Messergebnisse:

<u>Messaufbau</u>	<u>Luft</u>	<u>Core</u>	<u>Delta</u>	<u>Oberfläche</u>	<u>Innenluft*</u>
C64 mit Schirm	23,0	93,3	70,3	82	42
C64 ohne Schirm	22,0	85,3	63,3	76	39
C64 ohne Schirm mit losem Kühlkörper	21,5	72,5	51,0	63	38
C64 ohne Schirm mit Wärmeleitpaste	22,0	71,2	49,2	62	38
C64 ohne Schirm und Kühlkörper mit Low-Power-Umbau	21,0	72,0	45,0	--	27

* Core = Silizium-Die-Temperatur

Delta = Temperaturdifferenz zwischen Umgebungsluft außen und Silizium-Die

Luft = Lufttemperatur außerhalb C64-Gehäuse

Oberfläche = Oberflächentemperatur SID oder Kühlkörper

Innenluft = Lufttemperatur innerhalb des Gehäuses

Wir sind nun ohne Kühlkörper in dem Temperaturbereich, den wir vorher mit Kühlkörper erreicht haben. Die veränderte Wärmeentwicklung ist auf den Thermobildern gut zu sehen.

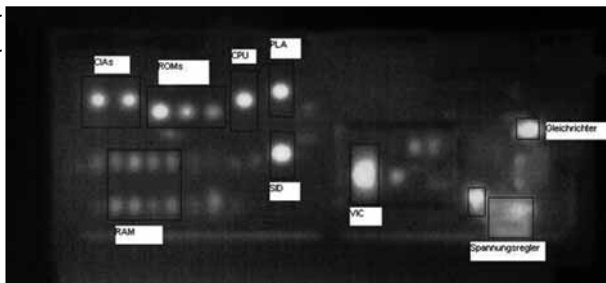


Abb. 37:
Wärmeentwicklung
vor dem
Umbau

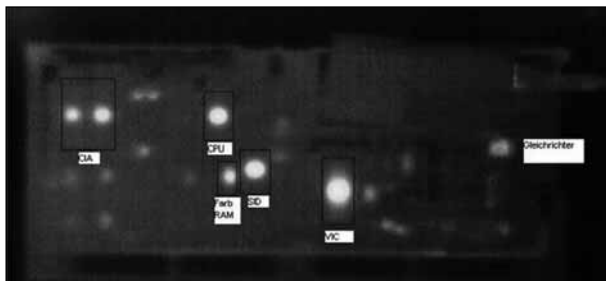


Abb. 38:
Wärmeentwicklung
nach dem
Umbau

ROM auf EPROM

Die ROM-Bausteine für Kernel, BASIC und Zeichensatz benötigen je ca. 65 mA. Ein EPROM 27CXXX hat lediglich eine Stromaufnahme von 4-7 mA. Durch das Ersetzen der drei ROM-Bausteine durch ein EPROM kann der Stromverbrauch auf 5 Volt Netzteilschiene um bis zu 270 mA reduziert werden, bei Verwendung eines 27C512-EPROMs ist auch ein Kernel-/BASIC-/Char-Umschaltung möglich.

Zuerst müssen die ROM-Bausteine ausgelötet und Präzisions-IC-Sockel verbaut werden. Anschließend werden in den mittleren Sockel U4 der Adaptersockel und die beiden Chip-Select-Leitungen auf Pin 20 des Nachbarsockels gesteckt.

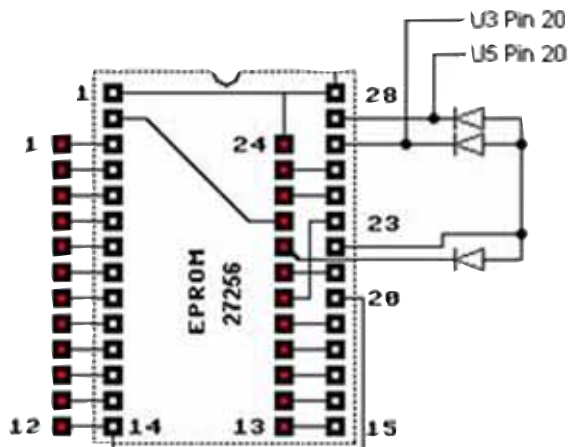
Speicheraufteilung im EPROM

-----32k			
Kernel	6000	bis	7FFF
-----24k			
Free	5000	bis	5FFF
-----20k			
Char	4000	bis	4FFF
-----16k			
Basic	2000	bis	3FFF
-----8k			
Free	0000	bis	1FFF
-----0			

Abb. 39:
Schaltbild
für den
EPROM-
Einbau

Benötigte Bauteile:

- 4x IC Sockel, 24-pol.
- IC Sockel, 28-pol.
- 3x Diode SB-140
- EPROM 27C256



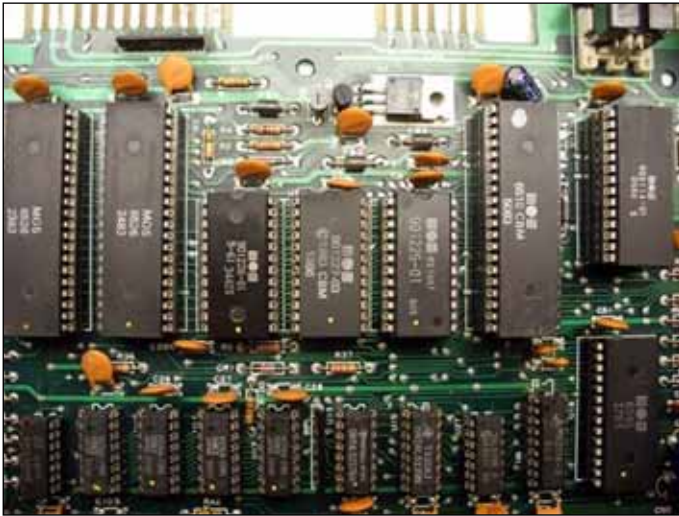


Abb. 40:
Platine vor
dem Umbau



Abb. 41:
Platine nach
dem Umbau

Ersatz-PLA

Die PLA im C64 zieht ca. 100mA, ein Umbau auf ein 27C512-EPROM reduziert die Stromaufnahme auf ca. 6mA. Es wird zuerst die PLA aus- und ein IC-Sockel eingelötet. Aus zwei IC-Sockeln wird der EPROM-PLA-Adapter zusammengelötet und das gebrannte EPROM eingesetzt. Eine detaillierte Anleitung gibt es im entsprechenden Kapitel.

7805 auf Schaltregler

Der Spannungsregler 7805 auf der ASSY 250407 wird mit einem Strom von ca. 300 mA belastet, die Eingangsspannung des Reglers sind die gleichgerichteten 9 Volt vom Netzteil. Der Regler hat nach der Glättung der 9 Volt ca. 10 Volt Eingangsspannung bei einer Ausgangsspannung von 5 Volt.

Der Wirkungsgrad des Reglers ist $10 \text{ Volt} - 5 \text{ Volt} = 5 \text{ Volt}$ Spannungsabfall am Regler, $5 \text{ V} : 10 \text{ V} = 50 \%$ Wirkungsgrad. Wenn der 7805 gegen einen pinkompatiblen Schaltregler ausgetauscht wird, steigt der Wirkungsgrad auf bis zu 94 %. Conrad Elektronik hat zum Beispiel die 154496-62 im Angebot. Einfach die Schraube am Kühlkörper lösen, den Regler auslöten und den neuen Schaltregler einsetzen.

Benötigte Bauteile: Schaltregler (Kosten ca. 10,16 €)

Abb. 42:
ASSY 250407
vor dem
Umbau (ori-
ginale 7805)



Abb. 43:
ASSY 250407
nach dem
Umbau



7812 auf Schaltregler

Beim 7812 auf der ASSY 250407 gilt das gleiche wie beim 7805: Strom ca. 100 mA bei 12 Volt, Eingangsspannung 17 Volt (Spannungsverdopplerschaltung mit Kondensator und Gleichrichter), Wirkungsgrad ca. 70 %. Durch Wechsel des Gleichrichters auf Schottky-Dioden und Reduzierung der Stromlast auf der 9-Volt-Schiene erhöht sich die Spannung am Eingang auf ca. 21 Volt, was dem Wirkungsgrad des Schaltreglers zu Gute kommt (Bis zu 97 %).

Benötigte Bauteile: Schaltregler (Kosten ca. 10,16 €, Bestellnummer 154484-62 bei Conrad)

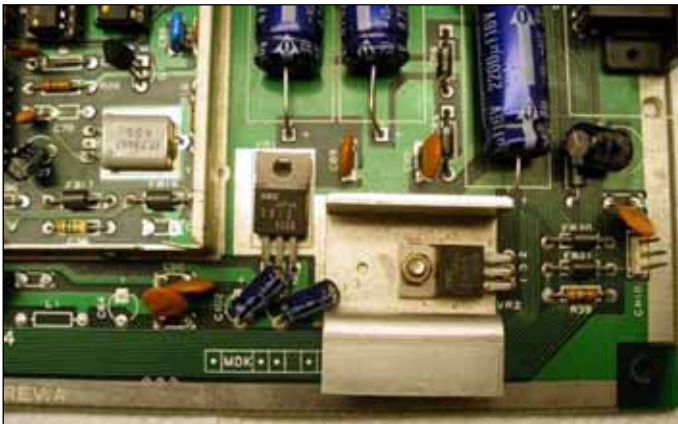


Abb. 44:
ASSY 250407
vor dem
Umbau (ori-
ginale 7812)



Abb. 45:
ASSY 250407
nach dem
Umbau

RAM-Umbau von 4164 auf 53C464-70

Der Arbeitsspeicher des C64 mit einer ASSY 250407 hat acht Bausteine à 64k*1, jeder Baustein benötigt zwischen 16 und 21 mA. Das bedeutet eine Gesamtstromaufnahme von 130 mA bis 170 mA.

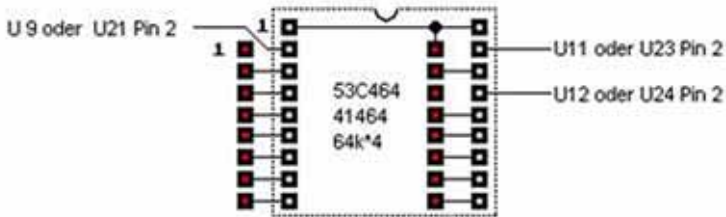
Commodore hat ja schon bei der ASSY 250466 gezeigt, dass ein Umbau von 4164- auf 41464- Bausteine möglich ist. Zuerst müssen alle acht RAM-Bausteine ausgelötet und die leeren Plätze mit 16-poligen IC-Sockeln bestückt werden. Nach dem Schaltplan aus zwei IC-Sockeln die RAM-Adapter bauen. Die 53C464-70 haben die geringste mir bekannte Stromaufnahme, ein -100 oder normaler 41464 braucht ca. 5-10 mA mehr.

Zu beziehen sind die Bausteine bei www.sh-halbleiter.de

Der Adapter sollte auf die Position U10 und U22 gesteckt werden, die Anschlussleitungen zu den leeren IC-Sockeln sollten so kurz wie möglich sein.

Stromersparnis des Umbaus: ca. 90 bis 140 mA

Abb. 46:
Schaltplan
für den
RAM-
Umbau



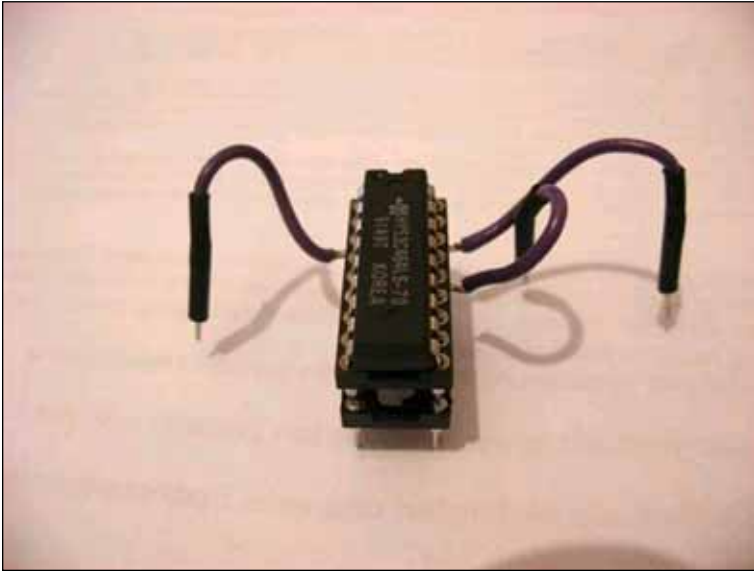


Abb. 47:
vorbereitetes
RAM (64x4)

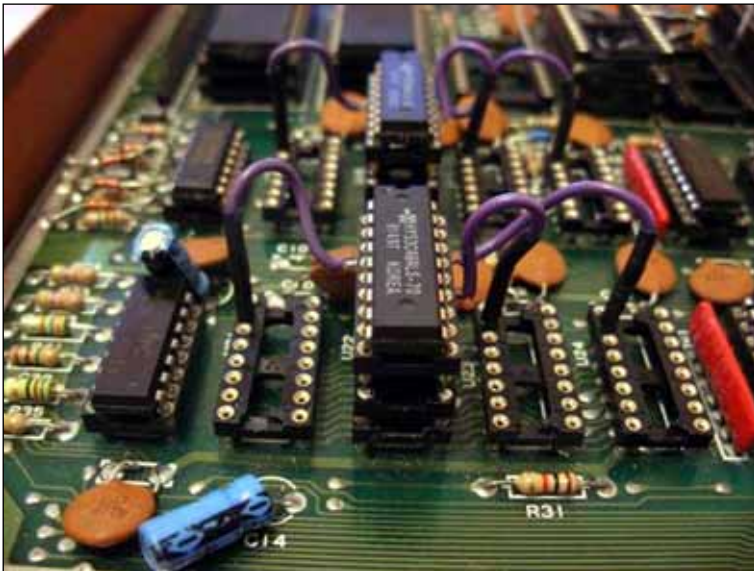


Abb. 48:
fertig
eingebaut

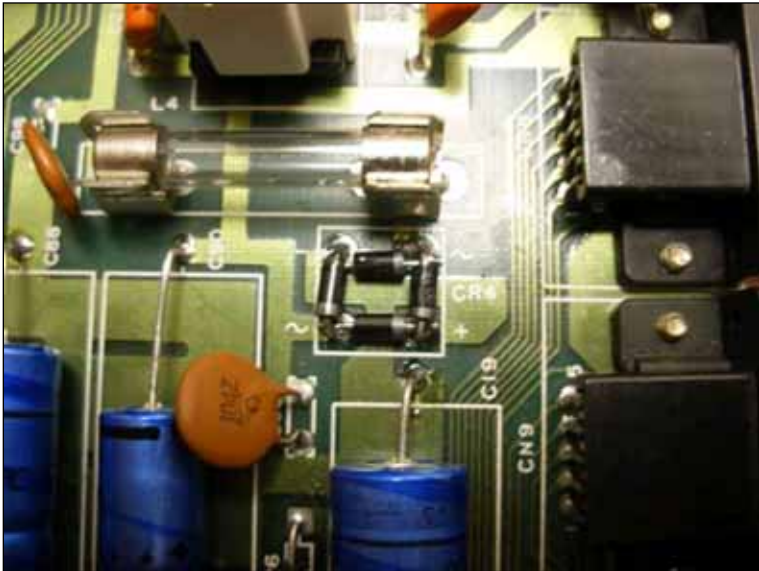
Gleichrichter auf Schottky-Dioden

Der Gleichrichter für die 9 Volt im Bereich des Netzschalters ist auch eine Wärmequelle: Spannungsabfall ca. $2 \times 0,7$ Volt bei ca. 600 mA Last = 0,84 Watt. Bei Schottky-Dioden reduziert sich der Spannungsabfall auf ca. $2 \times 0,3$ Volt = 0,36 Watt.

Empfohlener Schottky-Dioden-Typ: SB140

Der Umbau des Gleichrichters ist nur sinnvoll, wenn auch die Regler 7805 und 7912 umgebaut wurden! Ansonsten werden die 0,38 Watt bei den Reglern nutzlos in Wärme umgesetzt. Am Besten den alten Gleichrichter mit Entlötsaugpumpe auslöten, die neuen Dioden passend zurechtbiegen, in die Gleichrichterbohrungen stecken und verlöten (siehe Bild).

Abb. 49:
eingebaute
Schottky-
Dioden



Kühlkörper nachrüsten

Um die Wärme an den kritischen ICs abzuleiten, haben sich Kühlkörper bewährt. Die Kontaktfläche zwischen Kühlkörper und IC-Gehäuse ist dabei der kritischste Punkt: Der Wärmewiderstand sollte möglichst gering sein, Unebenheiten zwischen Kühlkörper und IC müssen ausgeglichen werden, und der Kühlkörper soll am IC haften.

Normale Wärmeleitpasten erfüllen den Zweck der Wärmeleitung und des Ausgleichs sehr gut, sind aber nicht für die Fixierung des Kühlkörpers geeignet. Da das IC-Gehäuse aus Zwei-Komponenten-Epoxydharz hergestellt wurde, bietet sich natürlich der gleiche Werkstoff auch für die Befestigung des Kühlkörpers an.

Die schlechtere Wärmeleitfähigkeit ist bei der geringen Wärmeleistung und Schichtdicke zu vernachlässigen. Als Vorbereitung ein Foto von der Leiterplatte machen, so ist ggf. nachträglich noch feststellbar, an welcher Stelle welches IC verbaut war.

Mit einem wasserfesten Stift bei der IC-Kerbe eine Markierung auf die Seite machen, da bei verklebtem Kühlkörper die Kerbe nicht mehr sichtbar ist. Wenn das IC gesockelt ist, lohnt es sich auch, die Typbezeichnung auf die Unterseite zu schreiben.

Für die eigentliche Verklebung werden nur 2-4 Tropfen 2K-Kleber benötigt. Man sollte aber mehr anmischen, weil dann das Mischungsverhältnis genauer eingehalten werden kann.

Die Klebstofftropfen mittig auf die Längsachse aufgetragen, den Kühlkörper auflegen und vorsichtig mit einem Hilfsgewicht anpressen. Überflüssiger Klebstoff darf nicht an den IC-Beinen herunterlaufen, überflüssigen Kleber im Notfall mit einem Wattestäbchen entfernen. Der Klebstoff härtet schneller aus, wenn man ihn auf ca. 40 °C erwärmt. Vor der Montage sollte man prüfen, ob der Kühlkörper genug Platz hat!



Abb. 50:
IC mit
Kühlkörper

C64-Netzteilenschutzschaltung

von Peter Trachte (AREA51HT)

Bei den inzwischen 25 Jahren alten Netzteilen kommt es leider immer häufiger zu Defekten. Problematisch ist dabei die Unterdimensionierung einiger Bauteile, insbesondere der Gleichrichterdioden (besser 3A-Dioden als 1A) und der Spannungs- und Temperaturbeständigkeit des Sieb-Elkos (anstatt 4700uF/16V besser 4700uF/25-35V und 105/125 Grad Celsius Temperaturbeständigkeit).

Im forum64 ist ein solcher Umbau beschrieben. An dieser Stelle möchte ich diesen nicht wiedergeben, da der Umbau doch etwas komplizierter ist und direkt mit netzspannungsberührenden Teilen gearbeitet wird. Stattdessen sollte man in Betracht ziehen, sich ein komplett neues Netzteil bauen zu lassen.

Um den C64 vor einem Netzteilausfall und der dann unter Umständen fließenden Überspannung auf der 5V-Zuleitung zu schützen, soll hier eine entsprechende Schutzschaltung vorgestellt werden.

Eigenschaften der Schutzschaltung:

1. Negative Spannungen werden kurzgeschlossen.
2. Spannungsdrifts durch langsame Ausfälle sollen erkannt und durch Notabschaltung des Netzteiles gestoppt werden.
3. Kurze Spikes (Spannungstransienten) sollen abgefangen werden.

Als Grundlage der Schaltung wurde eine Crowbar(Brechstangen)-Schaltung verwendet. Einfaches Konzept: Wenn die Spannung über einen gewissen Wert steigt, schließt die Schaltung die Spannung einfach kurz. Dabei soll die zusätzliche Gerätesicherung im Sekundärkreis auslösen (Schmelzsicherung) und die teure Schaltung und der C64 sind gerettet. Um vor negativen Spannungen im Verpolungsfall zu schützen, hat es sich bewährt, eine Leistungsdiode mit der Kathode (Ring) nach Plus in den Spannungseingang zu schalten. Im Fehlerfall wird die Spannung so ebenfalls kurzgeschlossen.

Die Erfahrung zeigt, dass die meisten Halbleiter ab ca. -0,7 bis -0,9 zerstört werden, deswegen kommt eine SB140 zum Einsatz, die unter Belastung ca. 0,3 bis 0,5 Volt Spannungsabfall hat und sehr schnell reagiert. Um vor Spannungsspitzen, die durch ESD-Endladung, Blitzein-

schläge in der Nähe oder auch Netzstörungen in die 5 Volt gelangen, zu schützen, kommt eine Suppressordiode zum Einsatz. Diese Diode ist eine Zenerdiode mit besonders steiler Kennlinie und sehr hoher Impulsleistung. Die verwendete Diode kann bis zu 600 W Peak vertragen, und durch die bidirektionale Ausführung ist auch der Verbau verpolungssicher.

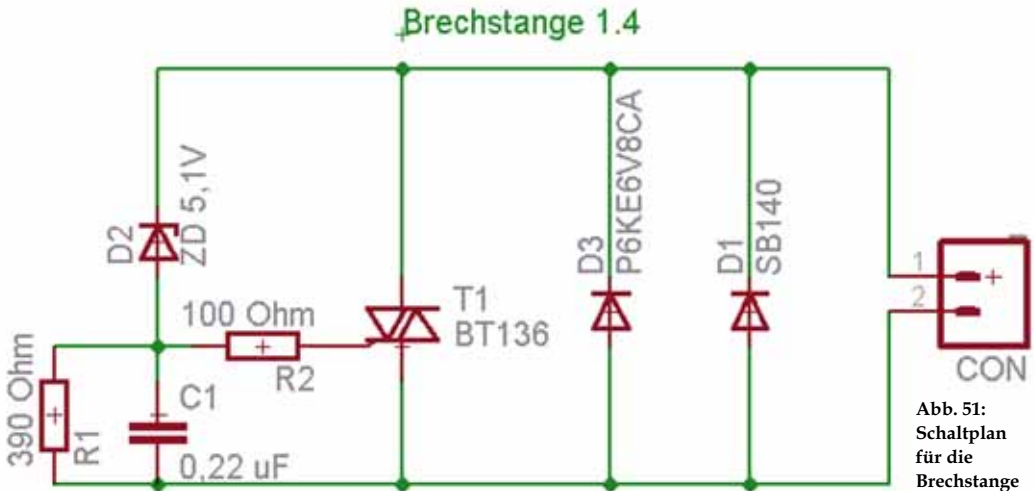


Abb. 51:
Schaltplan
für die
Brechstange

Nachteil der Schaltung ist, dass sie abgeglichen werden muss. Die verwendete Zenerdiode hat eine große Toleranz, so dass eine 4,7-Volt- oder auch eine 5,1-Volt-Diode in Frage kommt, am besten beide Typen beschaffen.

An einem Labornetzteil 5,00 Volt und die Strombegrenzung auf 500 mA einstellen. Die Schutzschaltung anschließen und den Strom beachten, es sollten weniger als 5 mA fließen. Nun langsam die Spannung erhöhen, bis die Crowbar auslöst und die Spannung unter 1 Volt zusammenbricht. Bei 20 °C sollte das bei ca. 5,3 bis 5,5 Volt geschehen.

Wenn die Schaltung zu früh auslöst, kann der Widerstand R1 verkleinert (unteres Limit ca. 180 Ohm) oder eine andere Z-Diode (zum Beispiel 5,1 V) getestet werden. Bei zu später Auslösung kann R1 auf ca. max. 560 Ohm vergrößert oder eine andere Z-Diode getestet werden. Vergrößern von C1 reduziert die Empfindlichkeit des Ansprechens der Schaltung.

Da die Bauteile, insbesondere die Z-Diode, Temperatur-Drift haben, ist auch ein Test unter erhöhten Temperaturen notwendig. Mit einem Haarfön oder besser einem regelbaren Heißluftfön die Schaltung auf zirka 50-60 °C erwärmen und den Versuch wiederholen. Die Auslösespannung darf nicht unter 5,25 absinken.

Wenn die Schaltung diese Tests überstanden hat, kann sie einfach in einen C64 verbaut werden. Dabei wird +5V und GND/Masse hinter dem Ein-/Ausschalter mit der Schutzschaltung über eine zusätzliche, superflinke Schmelzsicherung (0,8-1,0 A) verbunden. Erst über diese Zusatzsicherung im Sekundärkreis darf der C64 dann mit +5V versorgt werden! Im Störfall löst dann diese Sicherung aus.

Der Betrieb mit einem originalen C64 Netzteil empfiehlt sich nicht! Das Netzteil geht unter Umständen in einen Zustand, in dem keine Sicherung auslöst, maximale Last an Strom abgegeben wird und sich das Netzteil dabei bis zum Schmelzen (Brandgefahr!) aufheizt.

Generell alles immer nur hinter einer schaltbaren Steckdosenleiste unter Aufsicht betreiben und im Störfall sofort manuell die Stromzufuhr abschalten!

Netzteil nur für C64 G/II, ASSY 250469 und VC 20

Dem VC20 mit dem C64-Netzteilanschluss reichen +5V zum Betrieb. Die 9V~ werden nur für eventuelle Zusatzmodule benötigt. Daher kann man hier ein modernes 5V-Schaltnetzteil verwenden. Diese gibt es zum Beispiel als Restposten günstig zu erwerben.



Ich habe folgendes Netzteil von Pollin verwendet: Bestell-Nr. 350765. Es ist schön klein, liefert 2A an 5V und kostet nur 4,95 €.

Abb. 52:
Pollin-
Netzteil

Belegung der Netzteilbuchse am C64

Pin	Signal	Bemerkung
1	GND	Masse
2	GND	Masse
3	GND	Masse
4	-	unbelegt, evtl. auch 5 Volt
5	+5V DC	5 Volt Gleichspannung
6	9V AC	9 Volt Wechselspannung
7	9V AC	9 Volt Wechselspannung

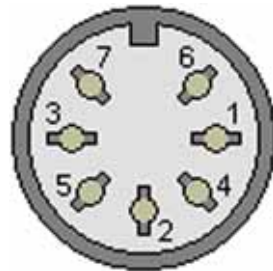


Abb. 53:
C64-Netzteil-
buchse

Ein C64G (ASSY: 250469) arbeitet so auch schon – allerdings ohne Sound, da der 8580-SID über die 9V~, Gleichrichter und 7809-Spannungsregler intern mit +9V versorgt wird. Abhilfe kann hier eine SID-Ersatzschaltung (siehe SwinSID) schaffen! Damit hat man am C64G auch Sound, obwohl man den Rechner nur mit +5V versorgt.

Bei den anderen C64-Platinenrevisionen reichen nur +5V als Stromversorgung leider nicht. Hier muss man noch die 9V~/ca. 1A über einen separaten Trafo zur Verfügung stellen.

Bitte bei Arbeiten mit Netzspannung die Sicherheitsbestimmungen beachten! Natürlich gehört obiges Netzteil in ein Kunststoffgehäuse und in den Primärkreis eine Schmelzsicherung! Ich empfehle den Betrieb von selbstgebauten Schaltungen und Retro-Hardware nur hinter einer schaltbaren Steckdosenleiste und nur unter Aufsicht!

PC-Netzteil-Adapter

Die AC-9V-Power-Lösung stammt von Tomas Pribyl (tomas.pribyl@ide64.org); Bilder von Josef Soucek (josef.soucek@ide64.org) mit freundlicher Genehmigung von Josef Soucek
(Link: <http://ide64.org/stuff.html>)

Die Schaltung gewinnt aus +5V und +12V DC die für den C64 benötigten +5V DC und ~9V AC.

Achtung: Der C64 sollte bei Verwendung des Schalters am PC-Netzteil oder einem Steckdosenschalter ein- und ausgeschaltet werden! Schalten am C64 reicht dann nicht mehr, da die 9V AC und die +5V eine gemeinsame Masse haben. Mit dem originalen C64-Einschalter läge so noch eine 9V-Phase am Gerät an. Auch hier sind ausreichende Erfahrungen mit elektrischen Schaltungen unbedingt erforderlich. Aufbau und Betrieb erfolgen auf eigene Verantwortung.

BASIC und Kernel bei ASSY 250469 durch Eprom ersetzen

Beim C64 mit ASSY 250469 sind BASIC und Kernel-ROM in einem 23128-DIL-28-ROM-Baustein kombiniert. Die Bezeichnung ist: 251913-01

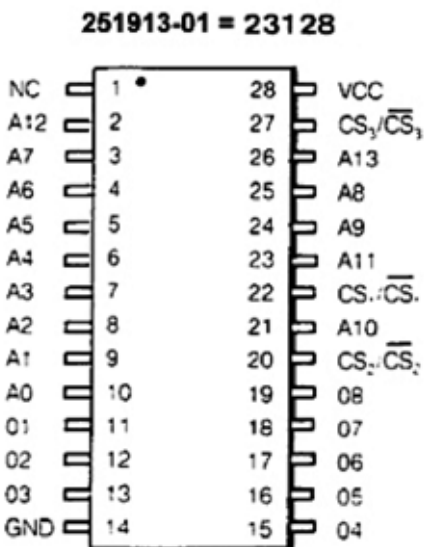
Dieser Baustein kann ohne jeden Adapter durch ein 27(C)128-Eprom ersetzt werden, da die beiden Bausteine pincompatibel sind.

Ein Eprom-Image kann unter Windows einfach aus den zum Beispiel dem VICE-Emulator beiliegenden BASIC- und Kernel-ROMs erzeugt werden. Dazu in einer Kommandozeile eingeben:

```
copy /b basic + kernal c64c.rom
```

Wichtig ist der Schalter /b für binäres Kopieren (keinerlei Ändern von Bytes). `basic` und `kernal` sind die beiden 8192 Bytes großen BASIC- und Kernel-ROMs. `c64c.rom` ist das 16384 Bytes große, resultierende Eprom-Image, welches dann in einem 27(C)128 gebrannt werden muss. Das originale ROM einfach auslöten, eine 28-pol DIL-Fassung einlöten, das gebrannte Eprom einstecken und fertig.

Abb. 55:
Pinbelegung
des 251913-01



S-Video-Ausgang am C64

Schaltung von Björn »JMP\$FCE2« Wieck

Was als Diskussion im Forum64 begann, den TV-Modulator auf der Assy 250469 zu entfernen, um Platz für sd2iec etc. zu haben, führte letztendlich zu einer Schaltung, die getrennte Chroma- und Luma+Sync-(S-Video)Signale zur Verfügung stellt. Diese Signale führen insbesondere bei TFT-Bildschirmen/-Fernsehern zu einer besseren Bildqualität als das FBAS-Signal des C64. An Bauteilen werden nur zwei Transistoren vom Typ BC547 (A-C), zwei Kondensatoren, eine Diode und ein paar Widerstände benötigt. Zwei Eingangssignale müssen direkt am VIC-Chip-Pin 14 und Pin 15 abgegriffen werden. Dazu zum Beispiel die beiden Pins des VIC aus der Fassung lassen und so hochbiegen, dass keine Verbindung zur Hauptplatine mehr besteht und man Kabel anlöten kann. Der FBAS-Ausgang funktioniert dann nicht mehr.

Die Schaltung lässt sich einfach auf Lochraster oder einer einseitigen Platine aufbauen.

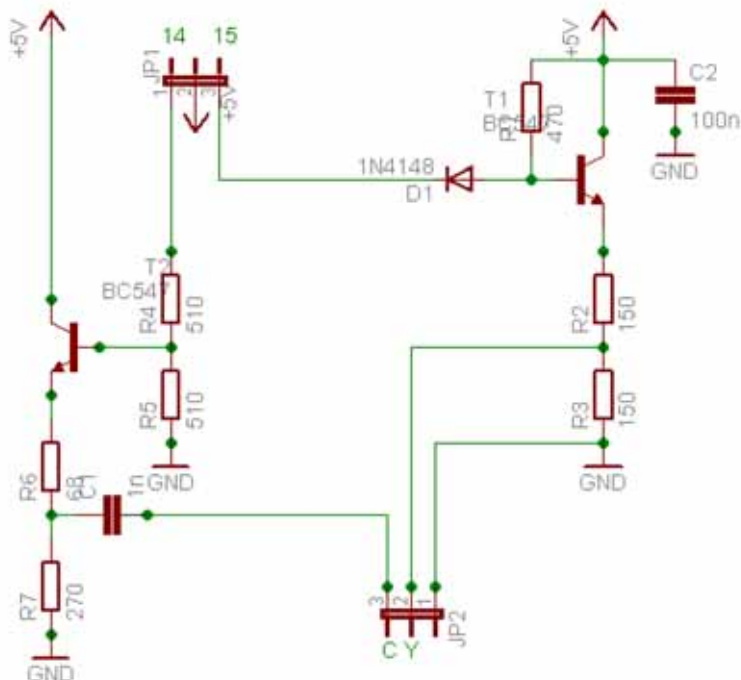


Abb. 56:
Schaltung

MMC2IEC-SD2IEC: SD-Card am C64

Historie/Hintergründe

Statt einer Floppy 1541, welche irgendwann mal das Zeitliche segnen wird, könnte man auch ein etwas moderneres und dabei deutlich kleineres Speichermedium verwenden. Aus dieser Idee entstand die 1541-III von Jan Derogee, welche mit einem PIC-Microcontroller arbeitet.

Pyrofer hatte bereits mit dem MMieC-Projekt etwas Ähnliches geschaffen. Als dann der C64DTV auf den Markt kam, hat sich Pyrofer zusammen mit Jussi Saily daran gemacht, die 1541-III und das MMieC umzubauen, damit das Resultat am C64DTV betrieben werden kann. Daraus entstand die 1541-III DTV – immer noch mit einem PIC.

Lars Pontoppidan hat dann die 1541-III DTV für seinen C64DTV-Mod auf einen AVR ATmega32 umgesetzt. So entstand sein MMC2IEC-Device. Er hat dann auch noch eine Version erstellt, welche wiederum an den C64 angeschlossen werden kann – damit hat sich der Kreis zur ursprünglichen 1541-II wieder geschlossen.

Das letzte, was Lars gebaut hat, ist ein MMC2IEC, welches sowohl am C64DTV als auch am C64 arbeitet – dieses hat er auch zum Verkauf angeboten. Basierend auf den Schematics von Lars gab es dann im Forum64 einige Nachbauten für den C64DTV.

Im Gegensatz zu Lars verwendete mancher dabei nicht den ATmega32 als TQFP (44-pin-SMD), sondern als DIP (40 pin), da dieser deutlich einfacher zu handhaben ist. AndyDTV benutzte dabei die L-Variante (3,3 V) für seinen Nachbau. Im Gegensatz dazu verwendete Shadowolf zwar den TQFP, änderte aber ein paar Ports. Die aktuelle Firmware SD2IEC passt schon nicht mehr in einen ATmega32, sondern der pinkompatible ATmega644 wird benötigt.

Mein Nachbau

Da mir der Nachbau mit SMD nicht so liegt, habe ich mich für einen Nachbau auf Lochraster entschieden. Programmiert habe ich mit dem Pollin Eval. Board.

Hier gibt es das HEX-File:

<http://petersieg.webng.com/mmc2iec/mmc2iec.zip>

Hier die »Fuses«-Einstellungen für *Ponyprog*:

Lochrasterschaltplan

Anstatt des LF33-3,3V-Reglers kann man auch zum Beispiel LP2950-3V-Regler verwenden. Der LF33 im TO-220-Gehäuse ist halt ziemlich groß. Auch zwei Si-Dioden in Reihe zur Spannungsreduktion funktionieren. Hier ein Foto des fertigen Aufbaus (SD-Slot wird noch mit Heißkleber fixiert). Wie man sieht, habe ich die drei IEC-Leitungen direkt an der rechten Seite des ATmega32 angeschlossen. Die beiden 560-Ohm-Widerstände zu den LEDs sind unterhalb des ATmega im Sockel »verstaute«:



Abb. 57:
Die »Fuses«-Einstellungen für *Ponyprog*

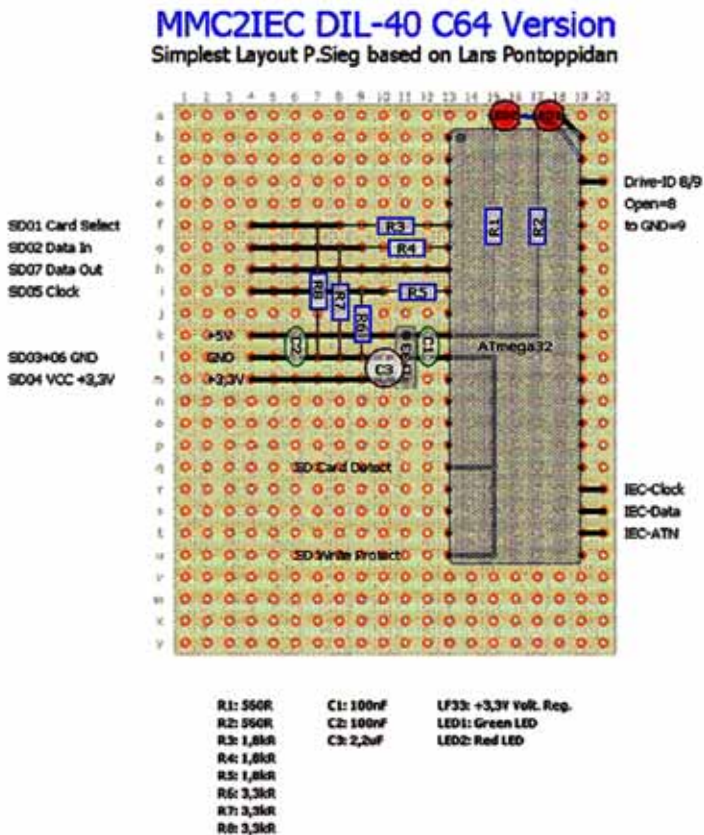
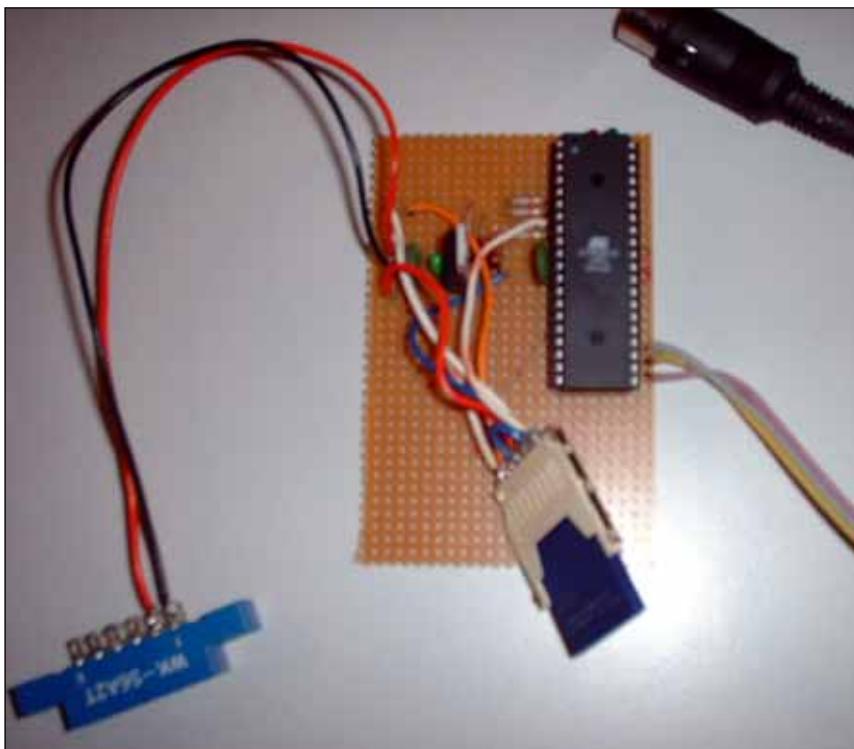


Abb. 58:
Lochraster-schaltplan

Abb. 59:
Der fertige
Aufbau



Hier ein weiterer Aufbau in einen USB Multicard Reader (Reichelt; 4,80 €).
Der USB Reader wurde »zerlegt« und der SD-Slot abgelötet:



Abb. 60:
Umbau in
USB Multi-
card Reader

Aufbau auf Lochraster
(noch ohne ATmega32).
Man sieht die beiden im
Sockel liegenden 560-Ohm-
Widerstände für die LEDs:

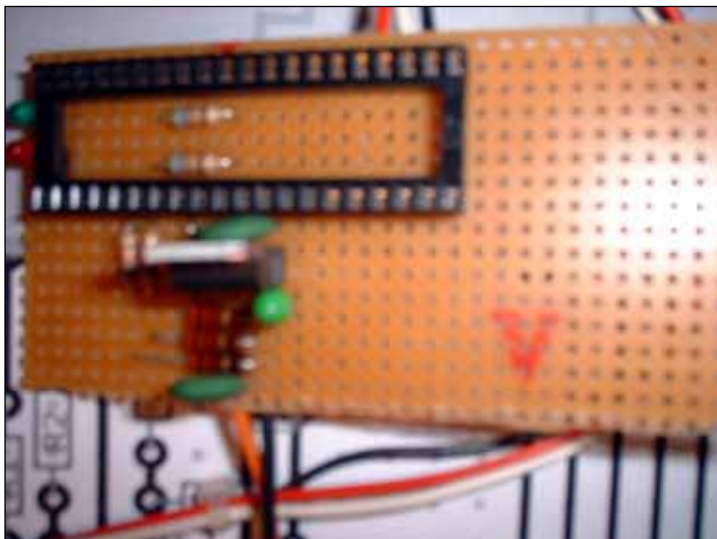


Abb. 61:
Aufbau auf
Lochraster

Abb. 62:
Lochraster in
den Gehäu-
seboden ein-
gepasst



Abb. 63:
Der Card
Reader in
Betrieb.
Für den
3,3V-Regler
im TO-220-
Gehäuse
musste ein
Loch in den
Deckel
gedremelt
werden.



A photograph of a Commodore 1541 disk drive. The drive is a light-colored, rectangular unit with a front-loading disk door. On the top surface, a custom printed circuit board (PCB) is attached, featuring a ribbon cable connected to the drive's internal components. A small label on the top left of the drive provides technical specifications. The drive is resting on a textured, light-colored surface.

Abb. 64:
Größen-
vergleich

Ich habe *SD Card Detect* und *SD Write Protect* fest auf GND gelegt. Falls der verwendete SD-Slot entsprechende Signale anbietet, kann man diese aber natürlich auch nutzen.

Dann sind die PINs mit den jeweiligen Slot-Anschlüssen zu verbinden und die GND-Verbindung ist zu trennen!

+5V kann man beispielsweise am Tapeport entnehmen:

1	2	3	4	5	6	
===				=====		
A	B	C	D	E	F	

(Ansicht, wenn man von hinten auf den Stecker schaut) Der untere Kontakt (A-F) ist mit dem jeweils oberen verbunden (A-1, B-2 ... F-6). +5V liegen an B-2 an, GND ist an A-1.

IEC-Buchse (in der Mitte ist Pin 6)

5	o		o	1	3 - ATN
		6	o		4 - CLK
4	o		o	2	5 - DATA
		3	o		

SD-Card

```
+-----  
| 8  
| 7 - Data OUT  
| 6 - GND  
| 5 - CLK  
| 4 - +3,3V  
| 3 - GND  
| 2 - Data IN  
| 1 - Select  
\  
 \ 9  
-----
```

LED

Das längere Bein ist + (Anode).

Nicht funktionierende SD-Karten laufen eventuell nach dem Formatieren mit diesem Tool wieder:

http://panasonic.jp/support/global/cs/sd/download/sd_formatter.html

(*Full Format* und *Adjust* einstellen)



Abb. 65:
So meldet
sich das
MMC2IEC

Neue Firmware sd2iec

Ingo Korb aus dem forum64 hat eine neue Firmware entwickelt! Diese ist JiffyDOS-kompatibel und unterstützt auch TurboLoad. Allerdings müssen für TurboLoad ein Quarz und zwei dazu passende Kondensatoren bestückt (XTAL1 + XTAL2) und die Fuses entsprechend programmiert werden (Low = 0xAF; High = 0xD9). Seit Version 0.7 wird auch der Fastloader der Final Cartridge III unterstützt.

Die jeweils aktuelle Firmware stellt Ingo Korb hier ins Netz: <http://www.sd2iec.de>

Darauf achten, dass die *lp für die LarsP-Pinbelegung verwendet wird! Im Unterschied zur originalen LarsP-Firmware erfolgt der Wechsel in ein Verzeichnis /M2I/D64 mit dem Befehl `open1,8,15,"CD:[name]":close1` und nicht über `load...`!

Abb. 66:
Lochraster-
aufbau mit
Quarz



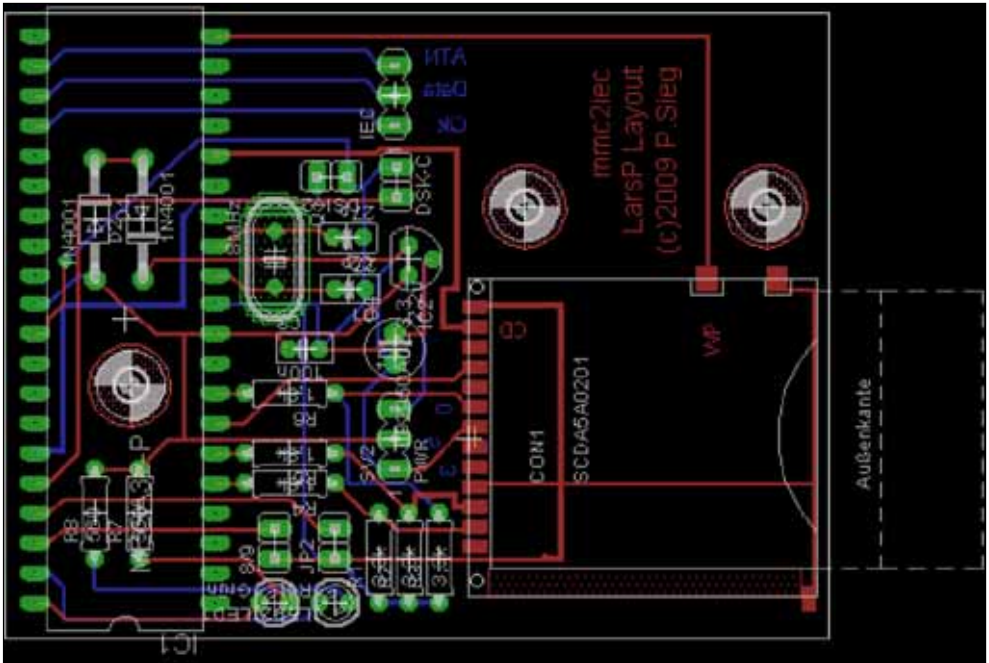
Abb. 67:
Speed-
Hardcopy
mit
TurboLoad



Eagle-Layout für MMC2IEC

Anfang 2009 habe ich mich darangesetzt, ein Layout in Eagle zu erstellen. Daraus wurde dann über einen Prototypen eine kleine Sammelbestellung im forum64.

Abb. 68:
Eagle-Layout



3D-gerendert
mit *eagle3d*:



Abb. 69:
MMC2IEC,
3D-gerendert

Abb. 70:
Platine
(Prototyp)

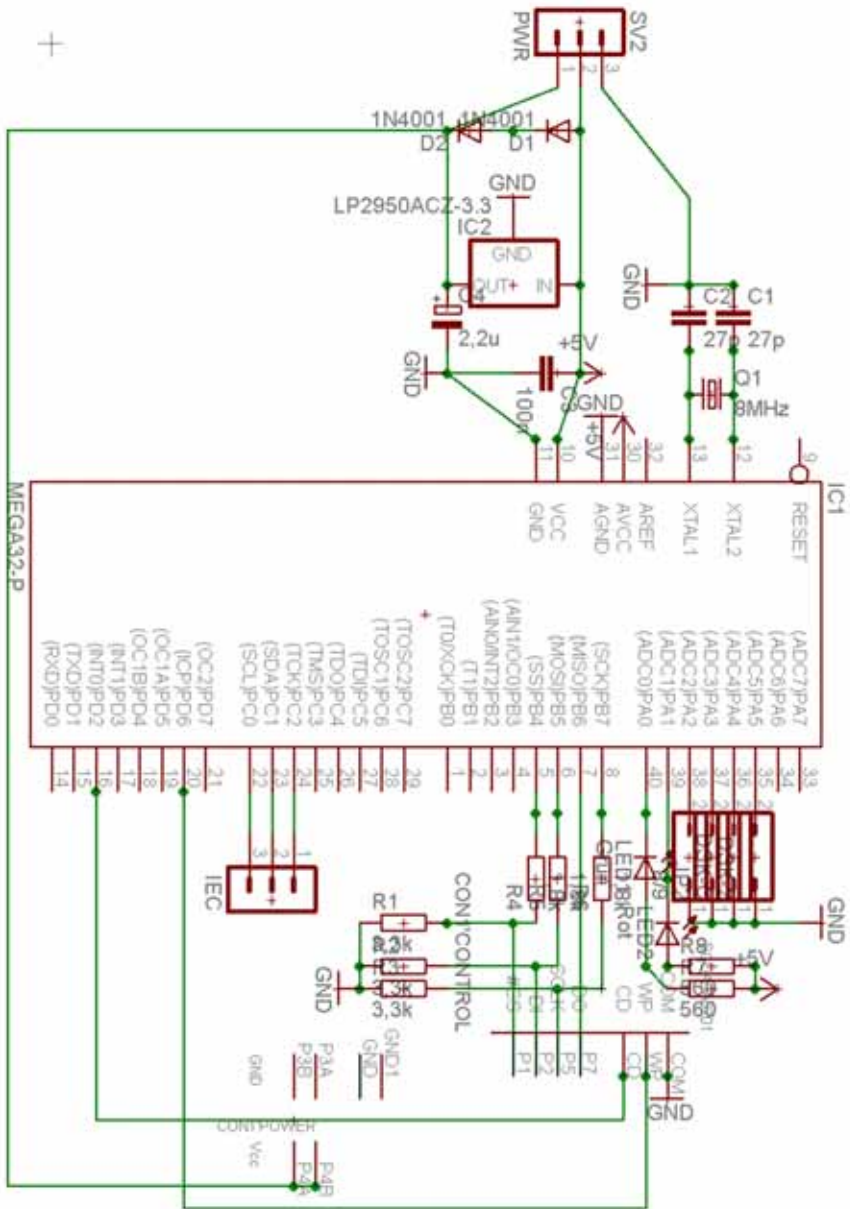


Abb. 71:
Professionelle Platine
(www.pcbcart.com)



Wichtig: Entweder 3V-Spannungsregler bestücken ODER die beiden SI-Dioden! Nicht beides! Zuerst den SD-Slot verlöten und alle Anschlusspunkte durchmessen.

Abb. 72:
Schaltplan



Easy Flash für C64

Von Thomas Giesel (skoe); Homepage: <http://skoe.de/easyflash>

EasyFlash ist ein Modul für den C64-Expansionsport. Dieses Modul lässt sich im Gegensatz zu klassischen Spielmodulen auch vom C64 aus beschreiben.

Damit kann man leicht diverse klassische Spielmodule, Programmsammlungen mit Menü oder sogar ein Diagnosemodul zur Fehlersuche herstellen. Alles, was man dazu braucht, sind ein C64, ein EasyFlash, die hier erhältliche Software und ein Abbild des Moduls (*.crt). Da solche CRT-Dateien recht groß sein können, ist eventuell ein großes Laufwerk wie eine FD-2000 oder ein sd2iec zum Transportieren der CRT-Datei ratsam.

EasyFlash ist kein Freezer-Cartridge wie zum Beispiel das Final Cartridge III oder das Retro Replay. Es ist auch kein Ersatz für ein 1541-Diskettenlaufwerk wie beispielsweise das sd2iec.

EasyFlash ist ein 1 MByte großes Flash-EPROM-Modul, das verschiedene Speicherkonfigurationen und Bänke unterstützt. Es benutzt keine traditionellen UV-löschbaren EPROMs, sondern Flash-Speicher. Dadurch kann man Module direkt am C64 »brennen«. Man braucht dafür kein EPROM-Programmiergerät.

Die EasyFlash-Hardware ist sehr einfach aufzubauen und wird durch ein umfassendes Software-Paket für Benutzer und Entwickler unterstützt. Damit ist es ganz einfach, sich sein eigenes Modul mit den Lieblingsprogrammen darauf zusammenzustellen. Für Entwickler ist es einfach, neue Software für das EasyFlash zu entwickeln oder vorhandene anzupassen.

EasyFlash unterstützt folgende Modulformate: Normal 8k, Normal 16k, Ultimax, Ocean Type 1, EasyFlash

Das Hardware-Design und die Dokumentation sind unter der »Creative Commons«-Lizenz (by-sa) veröffentlicht. Die Software steht unter der zlib-Lizenz. Das heißt, dass man damit fast alles machen kann.

Das sind die Teilprojekte von EasyFlash

- *EasyFlash*: Name der Hardware
- *EasyProg*: C64-Programm zum Schreiben von Cartridge-Images (*.crt) auf das Modul
- *EasyCRT*: PC-Programm zum Erstellen eigener Modul-Images
- *EasyLoader*: Menü, das zu eigenen Modulen hinzugefügt werden kann
- *EasySDK*: Sammlung von Dokumenten und Code-Schnipseln für Entwickler

Der erste Prototyp

Hier sieht man den ersten vollständig funktionierenden Prototypen. Er wurde auf einer Lochrasterplatine Rademacher Nr. 944 (Conrad 521250) aufgebaut. Ein Verdrahtungsstift oder ganz normaler Kupferlackdraht sind eine sehr einfache Art, ein EasyFlash-Modul zu bauen. Aber man braucht etwas Geduld :-). Eine perfekt passende Leiterplatte, an der man nicht rumzusägen braucht, ist das 8-Bit-Baby. Zur Zeit sind die ersten Platinen für einen Betatest bestellt.

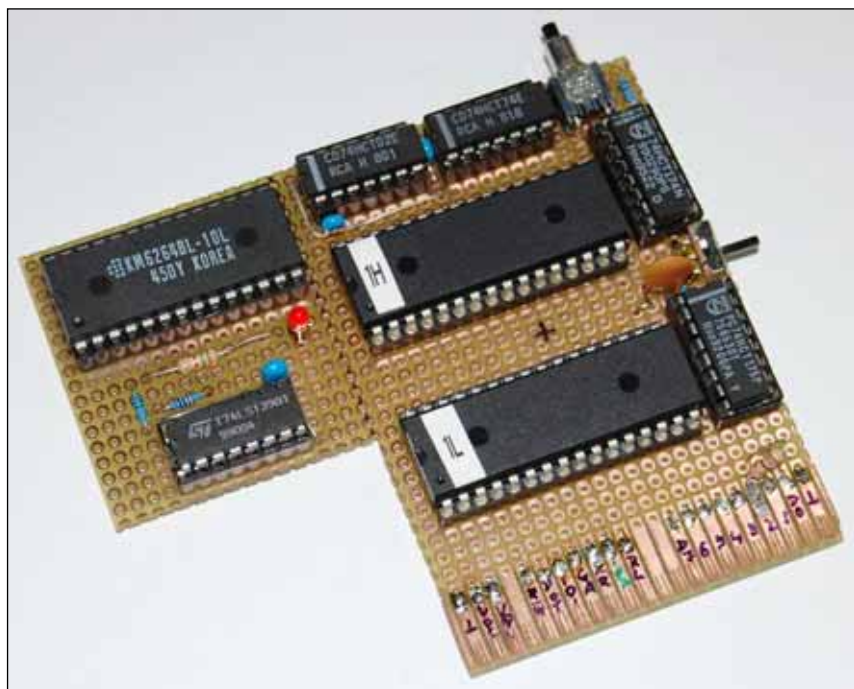


Abb. 74:
Easy-Flash-
Prototyp

Funktion

Die Adressleitungen A0...A12 für die Flash-Adressen können direkt vom Expansionsport verwendet werden, also von der CPU bzw. vom VIC-II. Damit können pro Flash 8 KiB adressiert werden. Die fehlenden Adressen BA13...BA18 kommen vom Register \$DE00, das durch U6 realisiert wird. Damit sind $64 \text{ Bänke} * 8 \text{ KiB} * 2 \text{ Flashes} = 1 \text{ MiB}$ möglich.

Nach dem Reset wird durch U5 die Leitung /GAME low und /EXROM high, der C64 aktiviert den Ultimax-Modus. Das bereits erwähnte Banking-Register U6 steht auf 0, so dass Bank 0 des ROMH-Flashs bei \$E000 ...\$FFFF sichtbar wird. Ein Reset-Vektor in diesem Speicherbereich kann für die weitere Initialisierung sorgen.

Durch Schließen von JP1 kann das Aktivieren von /GAME nach dem Reset verhindert werden. In diesem Fall startet der C64 mit unsichtbarem Flash-ROM. Dafür sorgt die Logik in U2A und U2B.

Die Speicherkonfiguration lässt sich über das Register \$DE02 (U5) mit beliebigen Kombinationen von /GAME und /EXROM konfigurieren. Sehr praktisch ist, dass trotz gesetztem Jumper die Programmiersoftware die Leitungen /GAME und /EXROM beeinflussen kann. Dadurch kann man die Flashes schreiben, ohne im Betrieb die Stellung des Jumpers bzw. Schalters ändern zu müssen.

/OE für die Flashes wird dann aktiv, wenn Phi2 low ist oder R/W high, das wird durch U1A und U2D erreicht. Diese Verknüpfung wird benötigt, weil auf dem C128 R/W auch low sein kann, wenn der VIC im Ultimax-Modus vom Modulspeicher lesen möchte. Das liegt daran, dass die CPU R/W schon im VIC-Zyklus setzt und dieses Phänomen dort nicht wie beim C64 durch AEC versteckt wird.

Die Ausgänge der Flashes werden nur dann tatsächlich durchgeschaltet, wenn zusätzlich zum gerade beschriebenen Signal /OE auch der Eingang /CE für einen Flash-Baustein low ist. Diese Eingänge sind deshalb mit den Leitungen /ROMH bzw. /ROML verbunden, die vom PLA im C64 für die richtigen Adressen enkodiert werden.

Die Erzeugung von /WE ist etwas komplizierter. Direkt nach der H-Flanke von Phi2 sind die Daten-, Adress- und Steuerleitungen von der CPU noch nicht stabil. Das liegt daran, dass der VIC-II erst mit dieser Flanke den Bus wieder freigibt. Deswegen müssen wir uns /WE zu einem späteren Zeitpunkt in diesem Zyklus selbst generieren. Die beiden Flipflops in U7 sind zu diesem Zweck zu einem 2-Bit-Grey-Code-Zähler zusammengeschaltet. Das erste Flipflop (U7B) wird durch Phi2 low im

Reset gehalten, das zweite (U7A) durch /OE. Wenn Phi2 high wird, sorgt die erste darauf folgende H-Flanke von Dotclock dafür, dass Q von U7B high wird. Damit diese erste Flanke von Dotclock garantiert nicht verpasst wird, benutzen wir für U7B direkt Phi2 als Reset-Signal. Warum das beim zweiten anders ist, wird gleich klarer werden.

Bei der zweiten H-Flanke kann Q von U7B auf Q von U2A durchgeschaltet werden. Die geschieht aber nur, wenn /OE zu diesem Zeitpunkt high ist, da dieses sonst U7A im Reset halten würde. Wir sehen also, dass U7 nicht nur für das Generieren des Write-Timings benutzt wird, sondern auch für das Verhindern eines /WE-Signals bei Lesezugriffen. Falls es sich nicht um einen Lesezugriff handelt, wird jetzt /Q von U7A low, also /WE aktiv.

Bei der dritten H-Flanke von Dotclock ändert sich nichts am Zustand der /WE-Leitung. Erst bei der vierten wird durch die Beschaltung als Grey-Code-Zähler /WE wieder high, rechtzeitig kurz vor dem Ende des Phi2-Zyklus'. Das so gewonnene Schreibsignal wird sowohl für die Flashs als auch für die beiden Register und den RAM benutzt.

U2C und U1B/C/D sorgen durch die Verknüpfung von /WE, /IO1 und der Adressleitung A1 dafür, dass die beiden Register bei Schreibzugriffen auf die richtigen Adressen beschrieben werden.

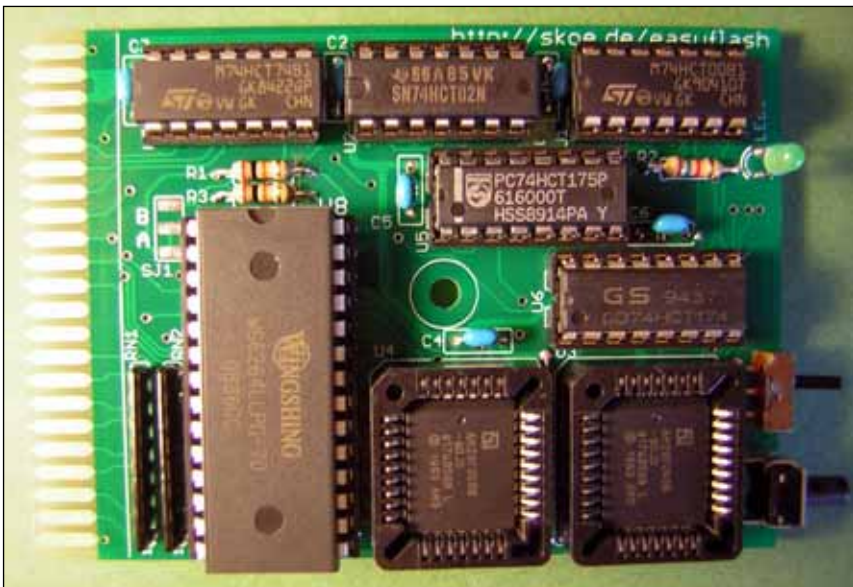


Abb. 75:
Das
Acht-Bit-
Baby

Vom RAM-Baustein U8 sind 256 Bytes an der C64-Adresse \$DF00 zu sehen. Dafür kann das Signal /IO2 direkt als /CS benutzt werden. Die bereits beschriebenen Signale /OE und /WE sorgen für die korrekte Steuerung dieses Bausteins.

Aus Sicht der Hardware wäre es naheliegend, die zusätzlichen Adressleitungen aus dem Bank-Register U6 für Banking des RAMs zu benutzen. Von dieser Möglichkeit wurde jedoch aus Gründen der Software-Entwicklung abgesehen: Wenn dieser Speicher für eine kleine bankübergreifende Leseroutine benutzt wird, darf er keinesfalls beim Umschalten der Bank ausgetauscht werden.

Es sei noch erwähnt, dass die Leitungen /ROML und /ROMH im normalen C64-Modus nur bei Lesezugriffen aktiv werden. Im Ultimax-Modus hingegen werden sie auch für Schreibzugriffe auf die entsprechenden Adressen auskodiert. Deswegen lassen sich die Flash-Bausteine nur im Ultimax-Modus beschreiben. Programmiersoftware wie zum Beispiel das Programm *EasyProg* muss diesen Umstand berücksichtigen.

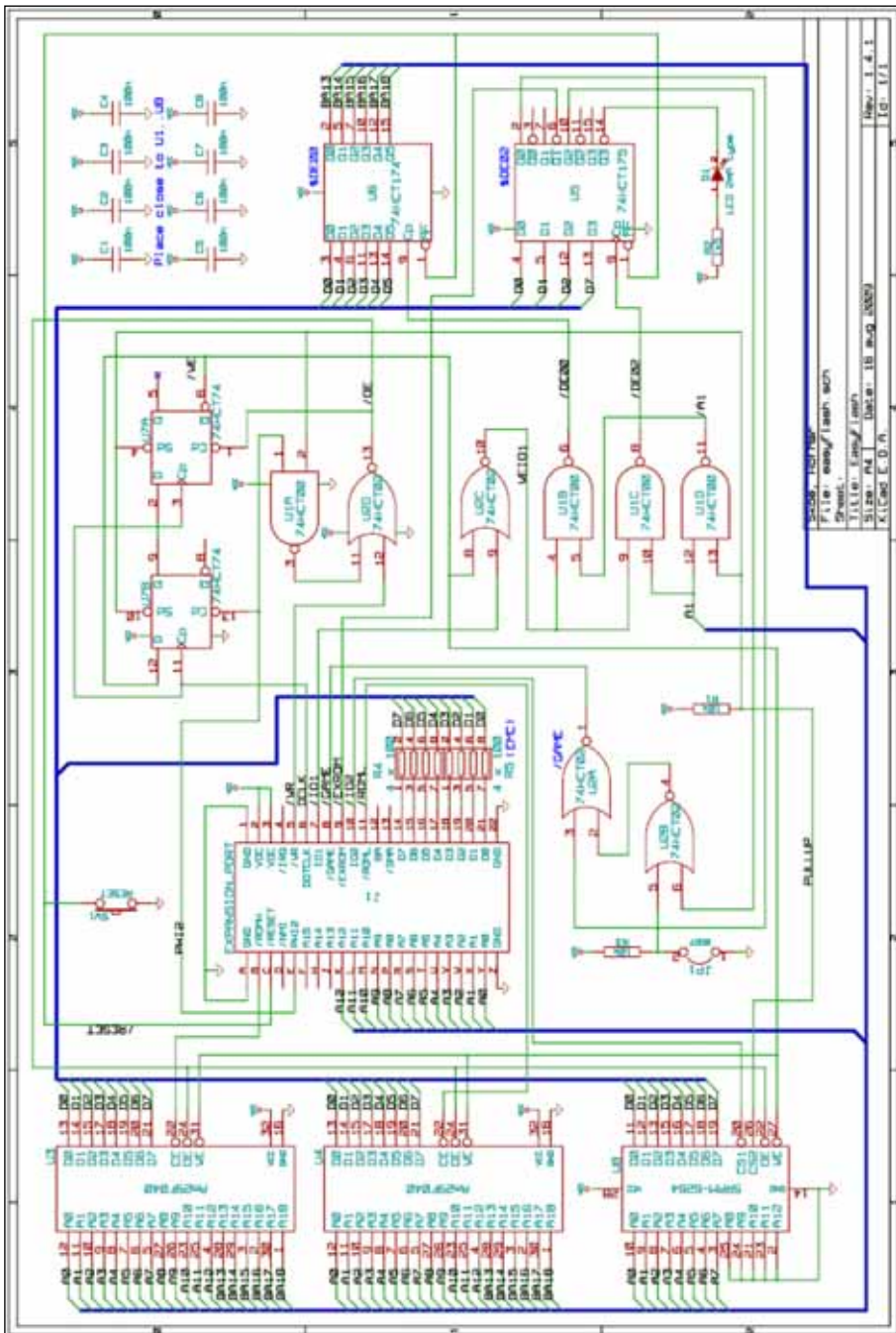


Abb. 76:
Schalt-
plan für
*Easy-
Flash*

NeoRAM – batteriegepufferte RAM-Disk für C64

Diese akku-gepufferte RAM-Karte wurde entwickelt von Nicolas Welte (x1541). Sie ist als Klassiker zu bezeichnen, weil sie ohne AVR, CPLD, GAL etc. auskommt und nur Standardtechnik und -bausteine verwendet. Die NeoRAM ist zur GeoRAM kompatibel. Der RAM kann in 512-KB-Schritten ausgebaut werden von 512 KB bis zu 2 MB.

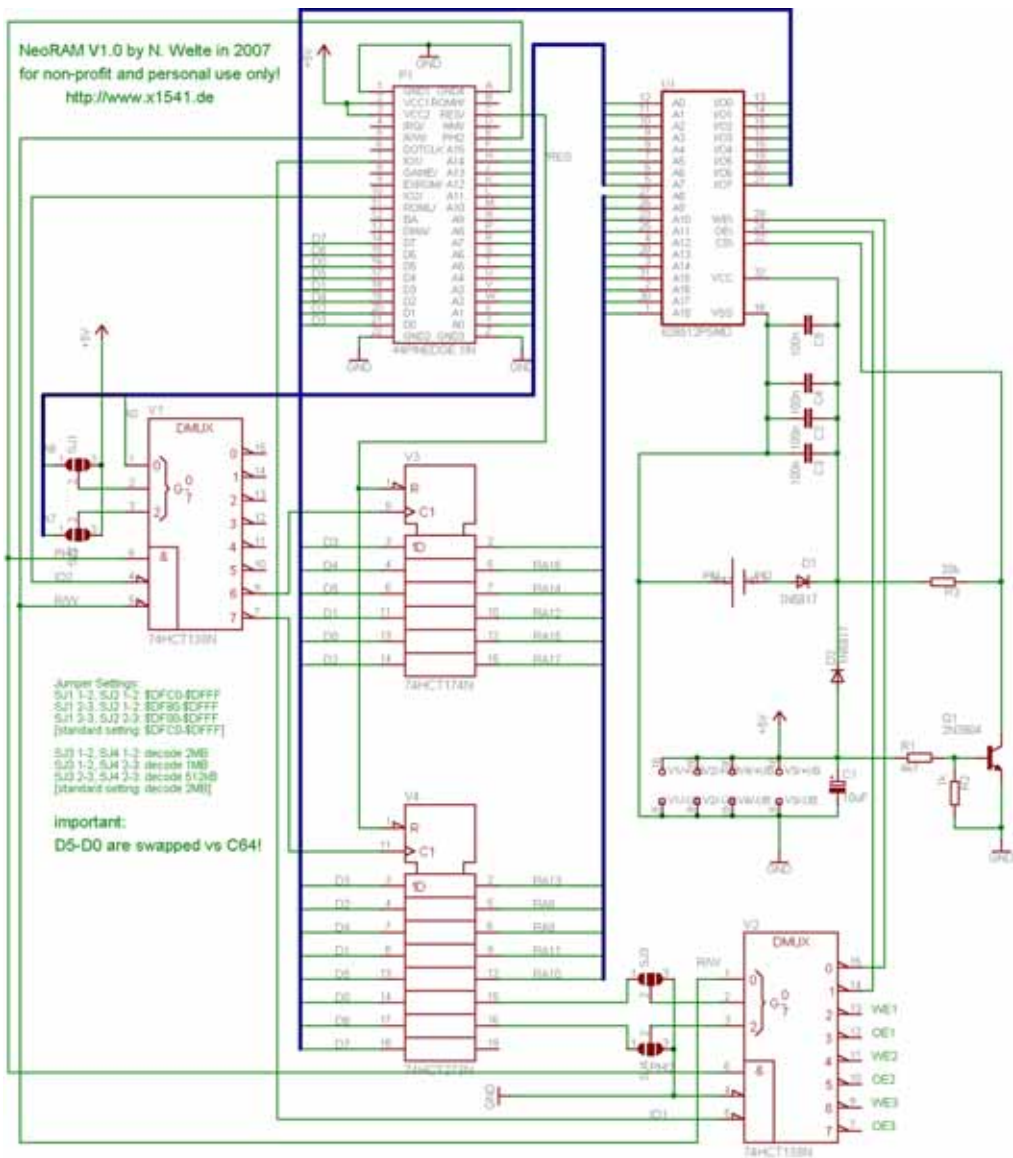
Durch die Pufferung mit einer Lithium-Zelle bleiben die Daten für ca. zwei bis drei Jahre erhalten. Wenn man die Batterie wechselt, während die RAM-Karte vom C64 mit Strom versorgt wird, sollten die Daten auch den Wechsel unbeschadet überstehen.

Mehr Informationen gibt es im Forum64 oder auf Nicolas' Homepage: www.x1541.de

Abb. 77:
NeoRAM
(hier mit
2 MB)



Abb. 78:
NeoRAM-
Schaltbild



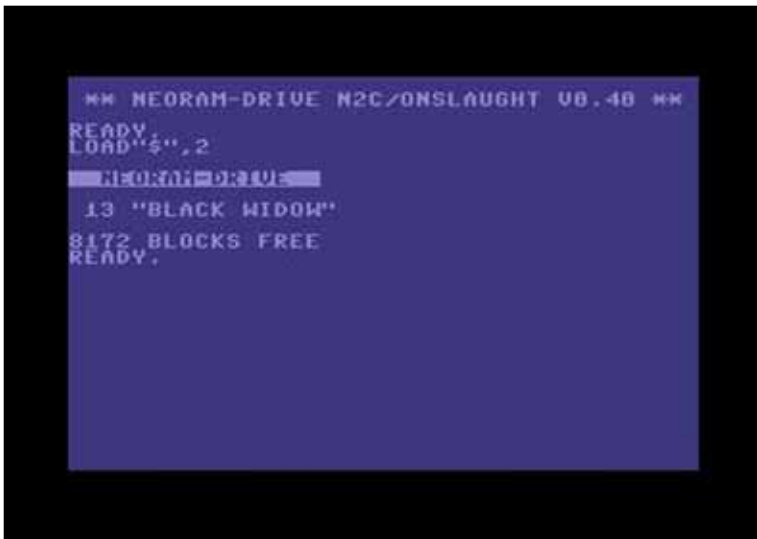
Für mich wurde die Karte durch den Ramdisk-Treiber von enthushi (Martin Wendt) aus dem Forum64 besonders interessant. Damit kann die Karte wie ein Laufwerk angesprochen werden. Die Ladezeiten (siehe Speedmessungen auf Seite 56) sind mit Faktor 20 ebenfalls erfreulich hoch. Das Tolle daran ist, dass man mit einer initialisierten RAM-Karte zu einem anderen C64 gehen kann und ohne Zusatzsoftware, einfach durch `SYS 57000` die RAM-Disk einbinden kann!

Schon kann man mit `load"$",2:list` die gespeicherten Programme sehen und natürlich auch laden. Super!

Den RAM-Disk-Treiber gibt es in der CSDB:

<http://noname.c64.org/csdb/release/index.php?id=61122>

Abb. 79:
NeoRAM im
Einsatz



DCM – Draco Cart Maker

Link: <http://www.sascha-bader.de/html/dcm>

Von Sascha Bader wurde ein Programm entwickelt, um sich einfach selbst eine Cartridgedatei für das Easyflash zusammenzustellen. Das Programm ist plattformübergreifend in Java geschrieben. Unter Windows dazu am besten eine kleine Batchdatei »dcm.bat« erstellen:

```
java -jar dcm.jar
```

Mit einem Doppelklick kann die Applikation gestartet werden:



Abb. 80:
DCM

Über »Load« kann eine vorhandene Cartridgedatei geladen werden.

»Save« sichert die Zusammenstellung in einer Cartridgedatei.

Über »+« können Programme hinzugefügt werden; über »-« werden sie entfernt.

Bei nicht alphabetischer Sortierung können einzelne Programme über »up« und »down« in den Programmplätzen verschoben werden (nach oben bzw. unten bewegt werden).

Speedmessungen verschiedener Lösungen

Die hier gezeigten Messungen wurde von unterschiedlichen Usern des Forum64 durchgeführt und sind hier nach den Programmladezeiten sortiert:

Testumgebung	PRG save	PRG load	SEQ save	SEQ load
C64 + IDE64 (Firmware V3.4)				
mit SCPU-Unterstützung	456,67	635,00	61,43	76,00
Ramlink mit Jiffy	47,24	115,45	14,58	13,33
Ramlink mit Fastload	47,24	105,83	16,86	16,17
CMD HD/1541-Partition	37,03	52,92	17,20	19,49
Dolphin 2 + Dolphin 2 1541	14,89	25,92	6,10	9,05
Vice 1.22 + 1541-Emulation	18,77	24,42	34,40	44,71
SX 64 + Professional DOS + 1541	13,05	23,09	4,73	4,29
Jiffy + sd2iec~0.5 + 1541-II	7,29	21,53	10,00	12,26
Jiffy + 1581	6,37	19,54	8,11	11,18
C64 + NeoRAM Ramdrive 0.3	18,77	18,68	2,54	1,63
SpeedDos + AR6 + SpeedDos 1541	8,51	18,41	1,67	3,49
Action Replay 4.2 Professional	8,56	18,14	1,22	1,16
Jiffy + AR6 + Jiffy 1541	8,84	18,14	2,78	2,74
Dolphin 2 + AR6 + Dolphin 2 1541	7,21	18,14	6,52	11,18
64'er V4 + AR6 + 64'er V3 1541	7,21	18,14	1,70	3,25
AR6 + CBM 1541	8,51	17,89	1,22	1,16
128D (64er-Mod.) + int. 1571 AR5 '88	8,62	17,89	1,22	1,16
AT 7.5 + Jiffy + 1581	11,30	16,70	7,89	11,34
Magic Formel 2.0 + 1541	14,89	16,28	5,70	5,51
AR 7.5 + 1581	11,00	16,00	2,40	1,34
AR 7.5 + 1571	7,00	15,50	1,00	1,00
AR 7.5 + Jiffy + 1571	7,65	15,49	2,30	6,91
FSD + AR6 + Jiffy 1581	8,95	15,49	4,94	6,03
Jiffy + AR6 + Jiffy 1581	8,95	15,49	7,29	11,34
AR 7.5 + 1541	7,00	15,30	1,00	1,00
AR6 + Jiffy 1581	8,78	15,12	2,61	1,54
FD2000 mit Jiffy	5,32	15,12	6,37	8,94
Jiffy + Jiffy 1581	5,07	14,77	6,19	9,38
Jiffy + 1571	2,77	14,43	2,35	6,18

Testumgebung	PRG save	PRG load	SEQ save	SEQ load
FSD + Jiffy 1581	3,69	14,43	4,22	5,03
Basic Jiffydos Fastload + FD-2000	20,45	13,96	6,24	9,16
Action Power Cartridge 8.1 DE	6,49	13,66	0,94	0,99
Turboload + sd2iec 0.5.0 (LarsP)	2,26	12,83	2,56	1,63
Beast System + CBM 1541	1,00	11,04	0,99	1,00
EXOS V3 + CBM 1541	1,00	11,04	0,99	1,00
128D (64er-Modus) int. 1571 mit RR	0,98	10,50	0,99	0,96
FC3 + Jiffy + 1541	2,60	10,40	2,20	5,31
FC3 + Jiffy + 1571	2,80	10,24	2,35	6,23
Jiffy + 1541	2,54	9,84	2,20	5,30
C64 + OC118 (beide mit Jiffydos)	2,50	9,77	2,13	5,17
Jiffy + Jiffy 1541	2,49	9,62	2,12	5,21
FC3 + 1541	5,40	9,34	1,00	1,00
FC3 + 1571	5,33	9,34	1,00	1,00
FC3 + 1541	5,76	9,07	1,00	1,00
FC3 + CBM 1541	5,71	9,07	1,00	1,00
SpeedDos + SpeedDos 1541	1,34	8,82	1,39	3,04

1541 Ultimate

Das MMC2IEC verhält sich gegenüber einem C64 wie eine 1541. Dabei wird die IEC-Schnittstelle abgebildet. Ansonsten hat MMC2IEC nicht viel mit einer 1541 gemeinsam. Diese hat eine eigene 6502-CPU, RAM und ROM – ist also ein eigener kleiner Computer.

Die *1541 Ultimate* von Gideon Zweijtzter emuliert eine 1541 fast exakt mittels Emulation der einzelnen Chips in FPGA.

Die 1541U ist ein Modul für die Commodore-8-Bit Computer. Es ist sowohl eine Floppy 1541 als auch ein Modul wie Action Replay oder Final Cartridge III. Das Herzstück der Platine ist ein mit 50 MHz getakteter FPGA. Zusätzlich befinden sich ein SD-Karten-Steckplatz, 2 MB Flash-ROM und 512 KB SRAM auf der Platine.

In der Plus-Version gibt es zusätzlich 32MB SDRAM, die unter anderem als RAM-Laufwerk (kompatibel zur REU) genutzt werden können.

Das Besondere der 1541 Ultimate ist, dass sie nicht nur das IEC-Protokoll der seriellen Schnittstelle von Commodore nachbildet, sondern eine echte 1541 emuliert. So können selbst Nachlader-Programme mit eigenen Schnelllader-routinen ohne Probleme von SD-Karte gestartet werden. Die 1541 Ultimate besitzt zwei Betriebsmodi:

Normalbetrieb

Hier wird die 1541U einfach in den Expansionsport des C64 oder C128 gesteckt und mit dem seriellen Kabel an der IEC-Buchse des Computers verbunden. Natürlich können gleichzeitig auch noch andere Laufwerke angeschlossen werden. Die 1541U hat drei Tasten. Es gibt eine Reset- und eine Freezertaste, die entsprechend dem emulierten Modul funktioniert. Zusätzlich gibt es eine dritte Taste, mit der man in den Datei-Browser gelangt. Hier kann man durch die Verzeichnisse der SD-Karte wechseln, Disketten-Images im D64-Format mounten, neue Verzeichnisse oder D64-Dateien erstellen und PRG-Dateien oder Programme aus einem D64-Image heraus per DMA in den Speicher laden und ausführen.

Bei der Benutzung mit einem C128 ist dabei zu beachten, dass der Datei-Browser im 40-Zeichen-Modus läuft. Falls man also gerade im 80-Zeichen-Modus arbeitet, muss man für den Datei-Browser kurzzeitig den Monitor auf 40-Zeichen-Darstellung umstellen. Die Bedienung ist sehr einfach und der Datei-Browser steht jederzeit zur Verfügung.

Wenn ein Programm zum Beispiel einen Diskettenwechsel verlangt, reicht ein kurzer Druck auf die Taste, dann wählt man die entsprechende D64-Datei aus und weiter geht es.

Stand-Alone-Betrieb

Hier wird die 1541U nur über das serielle Kabel mit einem Commodore-Rechner verbunden. Für die Stromversorgung benötigt man noch ein Netzteil mit einem Mini-USB-Anschluss. Man kann die 1541U auch einfach mithilfe eines USB-Kabels mit einem PC verbinden. Diese Verbindung dient nur zur Stromversorgung. Ein Datenaustausch über die USB-Schnittstelle ist nicht möglich.

In diesem Modus kann man die 1541U »blind« über die drei Taster steuern. Man kann mit den drei Tasten zur nächsten oder zur vorherigen Datei wechseln oder eine Datei auswählen.

Es gibt aber noch eine zweite Möglichkeit der Steuerung im Stand-Alone-Modus: Über eine eigene ID (standardmäßig ID 9) kann man mit DOS-Befehlen direkt auf die SD-Karte zugreifen. Wie im Dateibrowser kann man auch hier durch die Verzeichnisse wechseln, D64-Images mounten oder auch Programme direkt laden und starten.

Der Stand-Alone-Betrieb ist zwar nicht so komfortabel wie der Normalbetrieb, bietet aber den Vorteil, dass man die 1541U nicht nur am C64 und C128 betreiben kann, sondern auch an allen anderen Commodore-Rechnern, die über einen IEC-Bus verfügen (VC20, C16, ...). Auch wenn man einmal ein anderes Modul an seinem C64 oder C128 benutzen möchte, braucht man so nicht auf die 1541 Ultimate verzichten.

Abb. 81:
1541 Ultimate

Um die Aktivitäten der 1541 Ultimate sehen zu können, gibt es drei LEDs. Die eine zeigt den Zugriff auf die SD-Karte an, die zweite zeigt an, ob ein D64-Image geladen ist und die letzte zeigt den Betrieb der emulierten Floppy (wie die LED an einer echten 1541).

Mehr Informationen zu diesem Projekt gibt's hier: www.1541ultimate.net



SwinSID – SID-Ersatz auf AVR-Basis

Auf Basis des AVR ATmega 8515 konnte ein Ersatz für den schwer beschaffbaren Soundchip SID 6581 geschaffen werden. Die ersten Exemplare sind inzwischen von Forum64-Mitgliedern nachgebaut worden und die Ergebnisse sehr vielversprechend.

Schaltplan und Firmware für den AVR gibt es auf der Homepage:

www.swinkels.tvtom.pl/swinsid/

Abb. 82:
SwinSID mit
Original-SID
als Stereo-
SID



Abb. 83:
SwinSIDX2

Abb. 84:
Fuses-
Einstellun-
gen

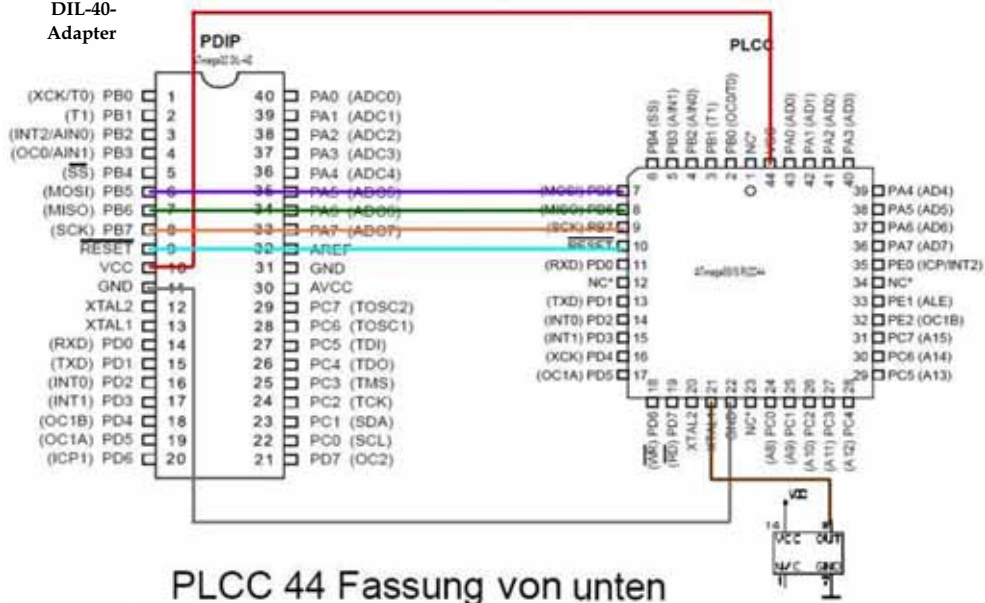


Abb. 86:
SwinSIDX2:
die fertige
Platine

Die Platine im Mono-Betrieb als SID-Ersatz. Slave-AVR kann dazu unbestückt bleiben und Leitung zur CPU A5 wird ebenfalls nicht benötigt. Sound-Out ist an Pin 3 der Audioleiste.

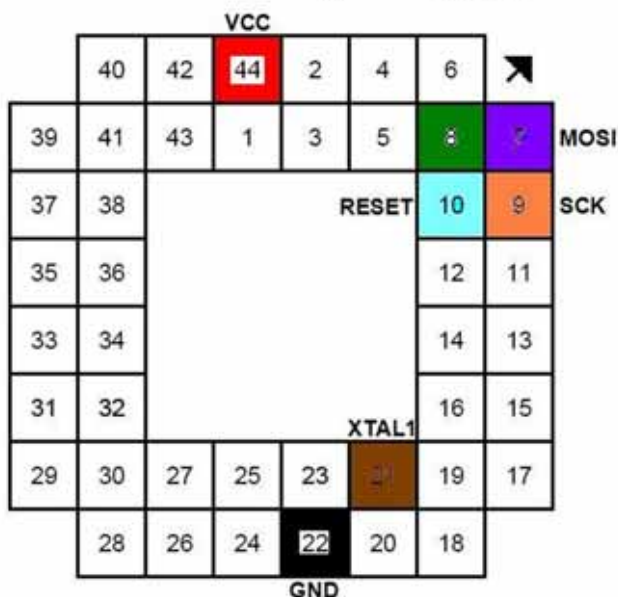
Zum Programmieren im Pollin-Board kann man folgenden DIL-40-Adapter für die Atmega32-Fassung bauen:

Abb. 87:
DIL-40-
Adapter



PLCC 44 Fassung von unten

Abb. 88:
PLC-44-
Fassung von
unten



Hier ein Bild eines aufgebauten Adapters (In der Atmega32-Fassung steckt natürlich auch eine Fassung, an der dann die Kabel angelötet wurden):

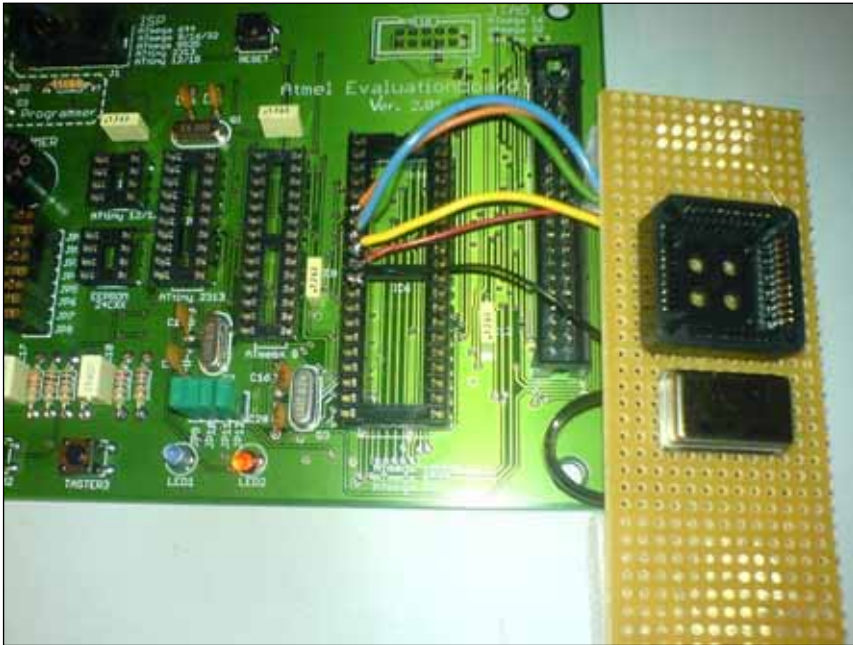


Abb. 89:
Aufgebauter
Adapter

Micro SwinSID88

Im Oktober 2010 geht ein kleiner Traum in Erfüllung: ein preisgünstiger SID-Ersatz auf AVR-Basis, der direkt in den SID-Slot gesteckt wird. Slawomir Musial (Swinkels) und Ingmar Weigel haben den TDA-Chip durch Nutzung von Pulsweitenmodulation (PWM) ersetzen können. Zum Einsatz kommt ein Atmega88PU, der über einen 32MHz-Quarzoszillator übertaktet wird.

Abb. 90: Fuses: lfuse:0xe0 hfuse:0xdf
Layout

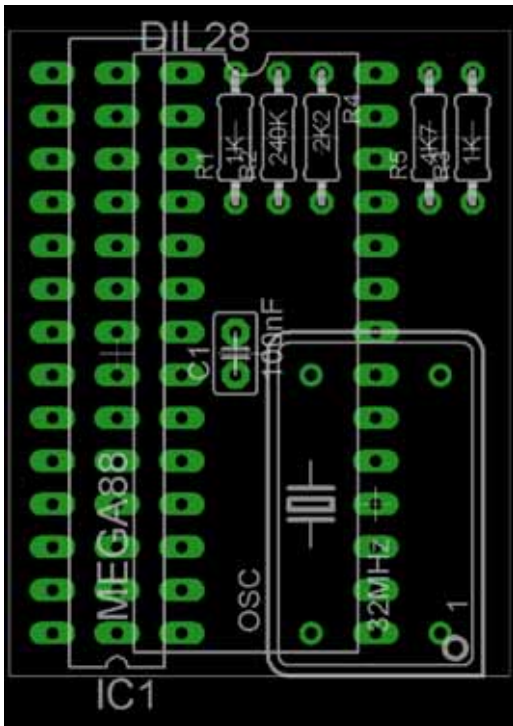
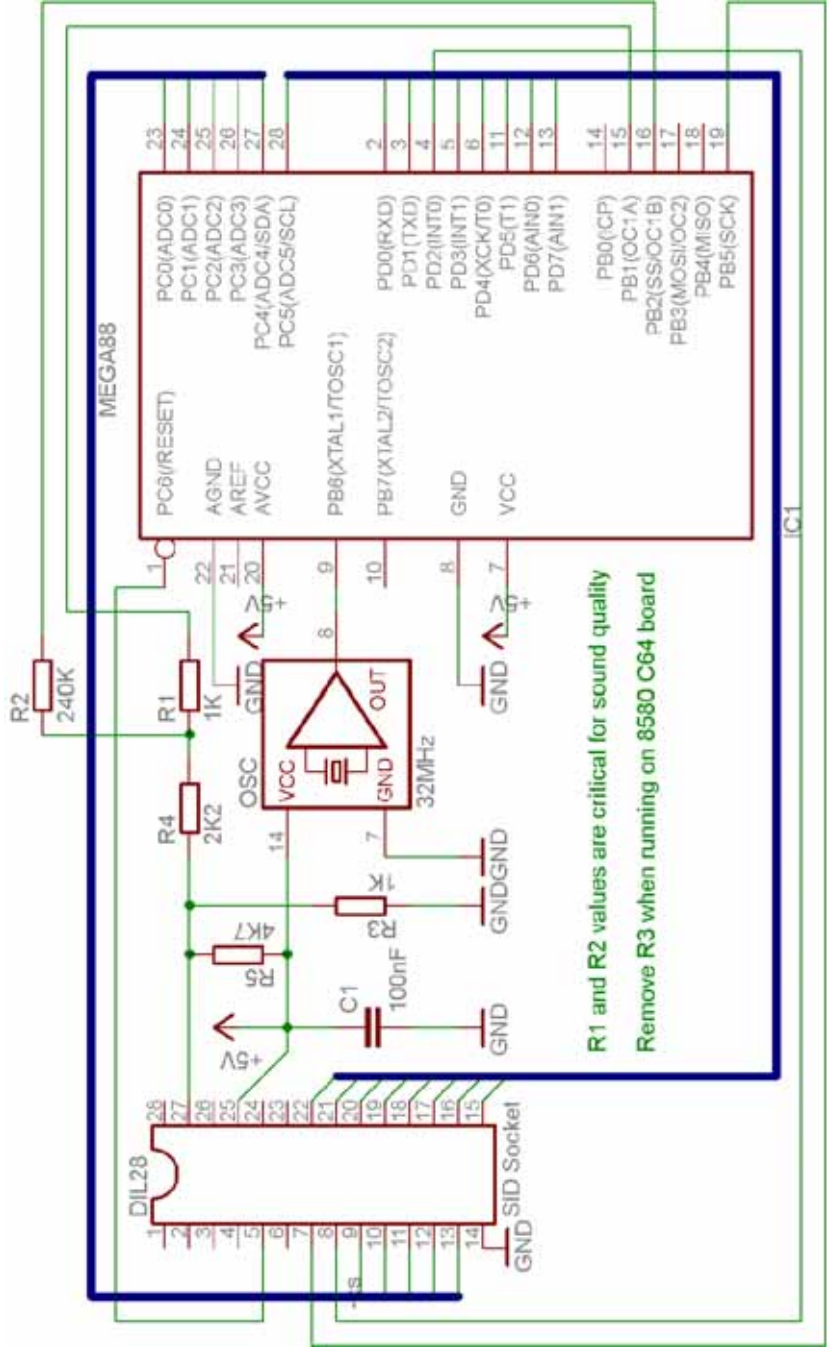


Abb. 91:
Erster
Aufbau mit
der neuen
1.1-Platine

Abb. 92:
Schaltplan



Tiny-Eprommer-Nachbau

Von: Björn »JMP\$FCE2« Wieck

Eprom-Brenner für den C64 gibt es ja haufenweise, und eigentlich arbeiten sie recht zuverlässig, wenn sie nicht gerade kaputt sind.

Bei den Alleskönnern aus dem Hause REX oder JANN bekommt man dann Probleme, weil sie Spezialchips haben, die nicht mehr zu bekommen sind. Der Tiny Eprommer aus der 64'er 1987 ist da eher angenehm simpel, weil er komplett aus Standardteilen besteht, die auch noch in 10 Jahren beschaffbar sein sollten. Leider hat auch der Tiny Eprommer gewisse Probleme, die sich aus der Einfachheit ergeben; so ist zum Beispiel die Brennspannungsregelung über Z-Dioden eher ungenau und einige Epromtypen lassen sich damit nicht brennen. Des weiteren können damit keine 64-KB-(27512)Eproms gebrannt werden, was aber damals über eine Zwischenplatine realisiert wurde. Das hat mich dazu gebracht, den Tiny Eprommer zu aktualisieren und einige Mängel zu beheben. Die Brennspannungen sind nun justierbar und damit die Z-Dioden Exemplarstreuungen obsolet. 64 KB (27512) lassen sich direkt brennen. Alle Bauteile sind wie vormals Standardtypen und daher leicht beschaffbar.

Der neue Tiny Eprommer hat zwei Schalter: Der eine schaltet zwischen Normalbetrieb und 27512, der andere für den 27512 zwischen LOW und HIGH Bank.

Abb. 93:
Prototyp auf
einseitiger
Platine

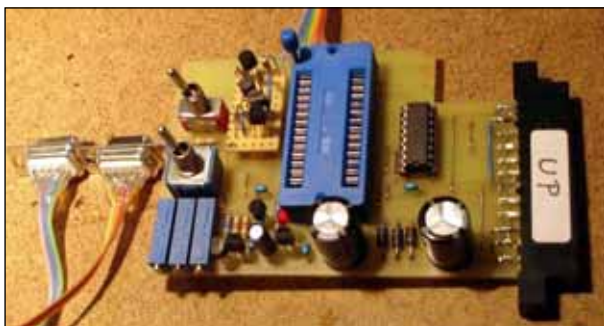
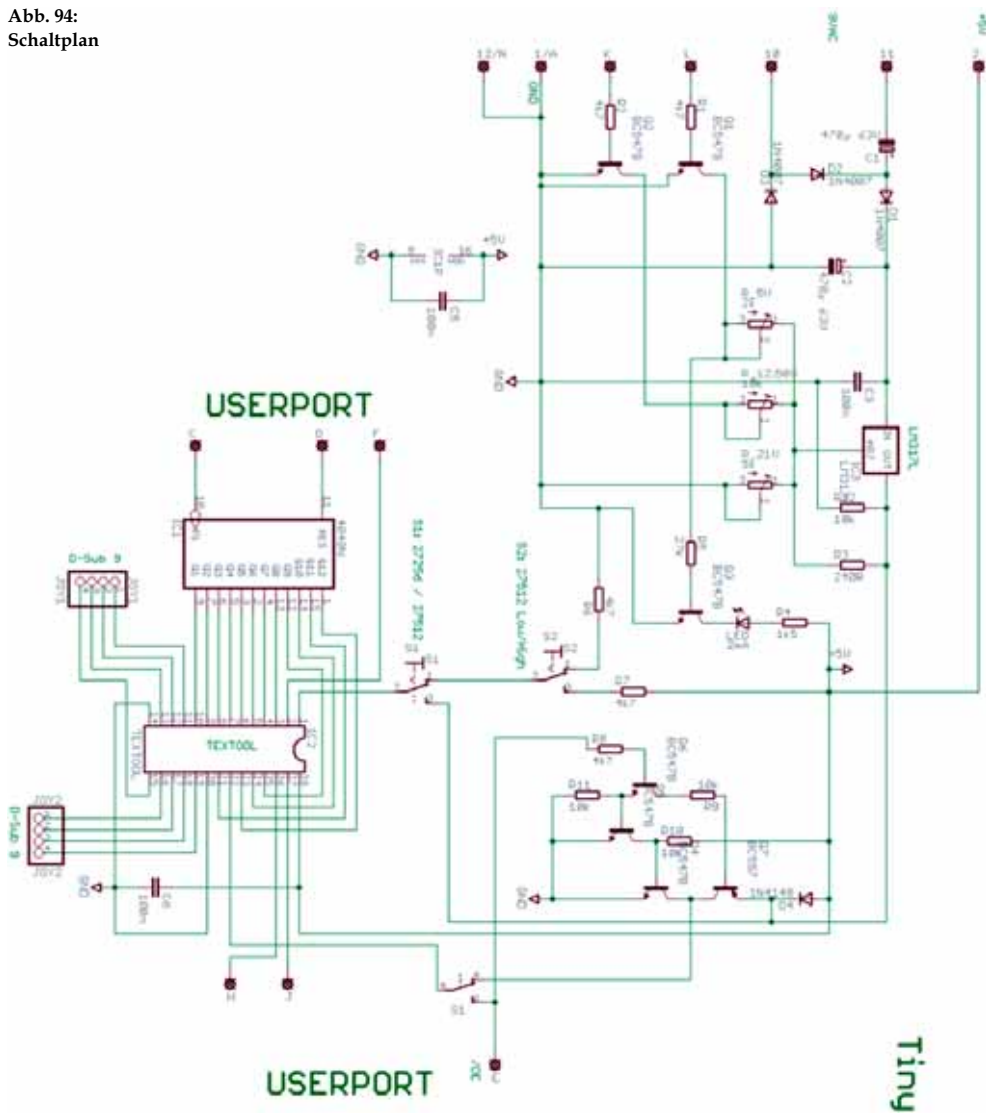


Abb. 94:
Schaltplan



Sotonic aus dem Forum64 hat das einseitige Layout für eine Sammelbestellung in ein zweiseitiges Layout umgewandelt.

Abb. 95:
Bild des
Aufbaus



Abb. 96:
Tiny
Eprommer
im Einsatz



Stückliste:

Bauteil	Wert	Device	Package	Reichelt Bestell Nr.
C1	470µ 63V	CPOL-EUE5-13	E5-13	RAD 470/63
C2	470µ 63V	CPOL-EUE5-13	E5-13	RAD 470/63
C3	100n	C-EU050-025X075	C050-025X075	KERKO 100N
C4	1µF	CPOL-EUE5-6	E5-6	RAD 1/63
C5	100n	C-EU050-025X075	C050-025X075	KERKO 100N
C6	100n	C-EU050-025X075	C050-025X075	KERKO 100N
D1	1N4007	1N4004	DO41-10	1N 4007
D2	1N4007	1N4004	DO41-10	1N 4007
D3	1N4007	1N4004	DO41-10	1N 4007
D4	1N4148	1N4148	DO35-10	1N 4148
IC1	4040N	4040N	DIL16	MOS 4040
IC2	TEXTTOOL	DIL28	DIL28-6	TEX 28
IC3	LM317L	LM317L	317L	LM 317 TO 92
LED	2mA	LED3MM	LED3MM	LED 3MM 2MA RT
Q1	BC547B	BC547B	TO92	BC 547C
Q2	BC547B	BC547B	TO92	BC 547C
Q3	BC547B	BC547B	TO92	BC 547C
Q4	BC547B	BC547B	TO92	BC 547C
Q5	BC547B	BC547B	TO92	BC 547C
Q6	BC547B	BC547B	TO92	BC 547C
Q7	BC557	BC557	TO92	BC 557C
R1	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R2	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R3	240R	R-EU_0207/7	0207/7	METALL 240
R4	1k5	R-EU_0207/12	0207/12	METALL 1,50K
R5	27k	R-EU_0207/7	0207/7	METALL 27,0K
R6	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R7	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R8	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R9	10k	R-EU_0207/7	0207/7	METALL 10,0K
R10	10k	R-EU_0207/7	0207/7	METALL 10,0K
R11	10k	R-EU_0207/7	0207/7	METALL 10,0K
R_5V	2K Lin	R-TRIMMT18	RTRIMT18	962-20 2,0K
R_12.75V	10k Lin	R-TRIMMT18	RTRIMT18	962-20 10K
R_21V	5k Lin	R-TRIMMT18	RTRIMT18	962-20 5,0K
S1		M9040P2	M9040P2	AS 500FPC
S2		M9040P	M9040P	AS 500APC
JOY1				D-SUB BU 09
JOY2				D-SUB BU 09

Kalibrieranleitung:

- Schalter S1 und S2 nach OBEN schalten
- Brenner mit dem Userport verbinden, die Joystickstecker müssen nicht angeschlossen sein
- Multimeter mit dem TexTool-Sockel verbinden, Plus an Pin 1 und Minus an Pin 14
- C64 einschalten. Das Multimeter sollte jetzt etwa 4,5V anzeigen.
- POKE 56579,255 : POKE 56577,0
- Das Multimeter zeigt jetzt irgendwas zwischen 1,25 und 23 Volt an. Die Brennspeisung ist jetzt eingeschaltet, die LED sollte leuchten.
- Mit dem 21V-Trimмер die Brennspeisung auf exakt 21 Volt einstellen
- Mit POKE 56577,64 ist die Brennspeisung für 12,5V aktiv und kann mit dem mittleren Trimмер auf genau 12,5 Volt eingestellt werden (im Schaltplan steht 12,75V, das ist noch vom Betatest ... 12,5V sind richtig).
- Zum Schluss noch POKE 56577,128
- Jetzt mit dem 5V-Trimмер genau auf 5 Volt einstellen. Die LED sollte jetzt aus sein.

Test der 64-KB-Funktion:

- POKE 56579,255
- POKE 56577,128
- Multimeter mit dem TexTool-Sockel verbinden, Plus an Pin 1 und Minus an Pin 14
- Schalter S1 und S2 nach UNTEN schalten, das Multimeter zeigt jetzt 0 Volt an.
- Schalter S2 nach OBEN schalten, das Multimeter zeigt nun ca. 5 Volt an
- Jetzt den Pluspol des Multimeters mit Pin 22 verbinden. Das Multimeter zeigt 0 Volt an.
- POKE 56577,132 > Multimeter zeigt 5 Volt an, LED ist aus.
- POKE 56577,68 > Multimeter zeigt 12,5 Volt an und die LED leuchtet
- POKE 56577,4 > Multimeter zeigt 21 Volt an und die LED leuchtet.

Damit sind Test und Abgleich abgeschlossen.

Floppy-»No Disk«-Hack für den Amiga

Wenn man auf die interne Floppy verzichten möchte, da man zum Beispiel eine interne (CF-)Festplatte eingerichtet hat und bei A600/1200 Daten mit dem PC über PCMCIA und CF-Karte austauscht, kann man natürlich trotzdem die interne Floppy eingebaut lassen. Aber zum einen schleppt man sie so immer mit herum, und zum anderen greift der Amiga ca. alle fünf Sekunden auf das Laufwerk zu, um zu prüfen, ob eine Diskette eingelegt wurde (hört man am leisen Klacken).

Die Floppy einfach auszubauen, spart zwar Gewicht, aber nun braucht der Amiga beim Booten noch länger.

Um das Warten zu vermeiden, kann man am Floppyanschluss eine Kabelbrücke zwischen Pin 2 und Pin 10 (im Bild in der unteren Reihe sind das Pin 1 und Pin 5 von rechts) einlöten. Dadurch »erkennt« der Amiga ein Floppylaufwerk ohne eingelegte Diskette und geht somit direkt zum nächsten Boot-Device, was dann in den meisten Fällen die interne Festplatte sein dürfte. Damit geht das Booten wieder so flott wie mit angeschlossenem Floppylaufwerk.



Abb. 97:
Der »No
Disk Hack«

VC 20-Multi-Cartridge

Ein defektes VC20-Modul kann man sehr einfach mittels eines größeren EPROMs in ein Mehrfach-Spielemodul umbauen. Es wird jeweils ein 8K-ROM-Bereich auf \$A000 über BLK5-Signal und Auswahl über die Adressleitungen A13 ... selektiert.

VC20-Expansionsport (von Rückseite gesehen):

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

A	B	C	D	E	F	H	J	K	L	M	N	P	R	S	T	U	V	W	X	Y	Z

1 - GND	A - GND
2 - CD0	B - CA0
3 - CD1	C - CA1
4 - CD2	D - CA2
5 - CD3	E - CA3
6 - CD4	F - CA4
7 - CD5	H - CA5
8 - CD6	J - CA6
9 - CD7	K - CA7
10 - !BLK1	L - CA8
11 - !BLK2	M - CA9
12 - !BLK3	N - CA10
13 - !BLK5	P - CA11
14 - !RAM1	R - CA12
15 - !RAM2	S - CA13
16 - !RAM3	T - I/O2
17 - VR/W	U - I/O3
18 - CR/W	V - SO2
19 - !IRQ	W - !NMI
20 - NC	X - !RESET
21 - +5V	Y - NC
22 - GND	Z - GND

27C256-EPROM und Beschaltung

27256	
+-----\ /-----+	
1 - Vpp	Vcc - 28
2 - A12	A14 - 27
3 - A7	A13 - 26
4 - A6	A8 - 25
5 - A5	A9 - 24
6 - A4	A11 - 23
7 - A3	/OE - 22
8 - A2	A10 - 21
9 - A1	/CE /pgm - 20
10 - A0	D7 - 19
11 - D0	D6 - 18
12 - D1	D5 - 17
13 - D2	D4 - 16
14 - gnd	D3 - 15
+-----+-----+	

+5V und Gnd sollte klar sein

D0-D7 auch; gehen direkt ans EPROM

A0-12 auch; gehen ebenfalls ans EPROM

VPP an +5V legen

/CE an BLK5 legen

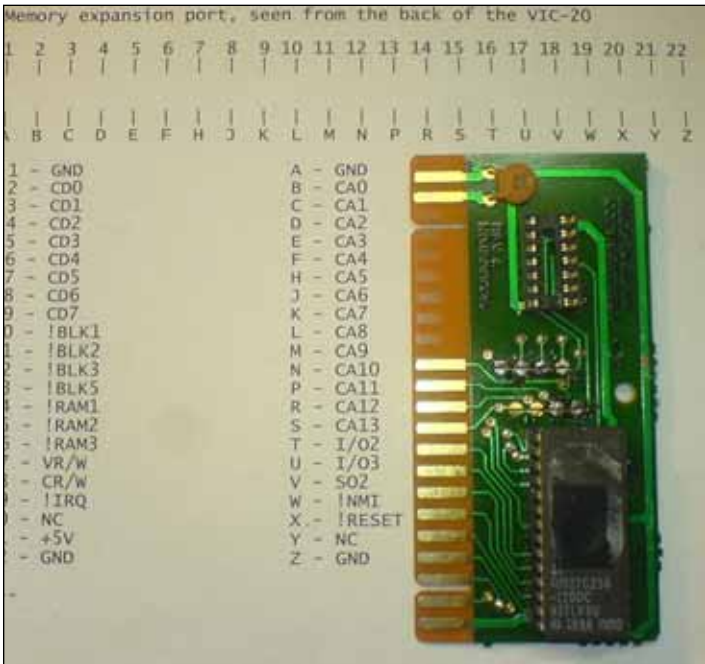
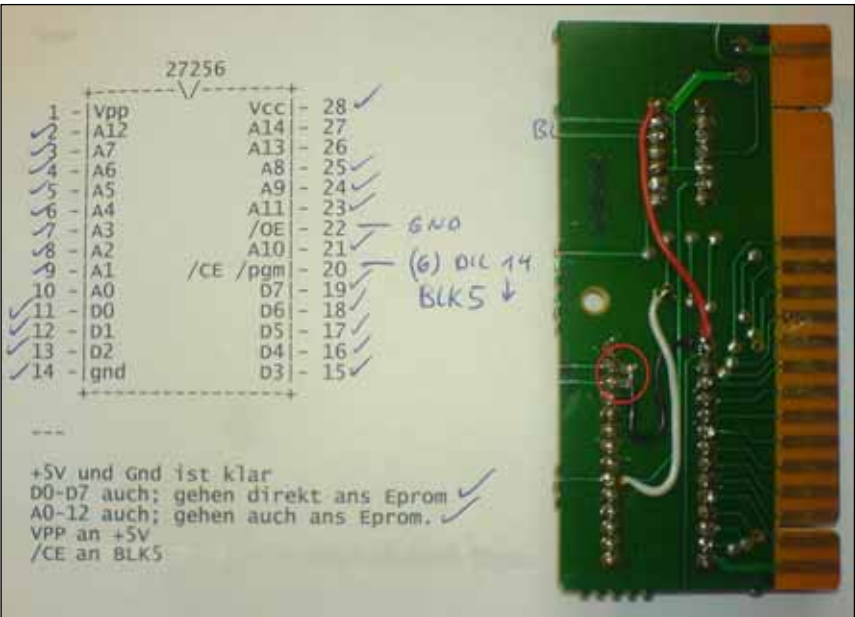


Abb. 98:
EPROM von
oben

Abb. 99:
EPROM von
unten



Im markierten Bereich sind die A13- und A14-Adressleitungen über 10k-SMD-Widerstände an +5V gelegt (Pullup). Über Jumper/Schalter an GND lassen sich so die 4 x 8k ROM-Bereich anwählen.

Final Expansion für VC20

Nach Ing. Thomas Winkler

Der VC20 war einer der ersten Heimcomputer, der in Masse produziert wurde. Keiner glaubte wirklich an einen Erfolg, Commodore produzierte ihn hauptsächlich, weil zu viele RAM-Chips auf Lager waren. Zudem steckte man viel Geld in die Entwicklung eines Grafikchips (VIC), aber keine Firma wollte ihn kaufen. Doch sehr schnell wurde der VC20 ein großer Markterfolg und bereitete den Weg für den C64 ...

Im Gegensatz zum C64 hat der VC20 nur sehr wenig RAM: nur 5 KB, und davon stehen nur 3,5 KB zur freien Verfügung. Deswegen wurden auch bald Speichererweiterungen verkauft. Erst eine 3-KB-, dann eine 8-KB- und eine 16-KB-Erweiterung. Man kann die Erweiterungen teilweise auch kombinieren, weil sie für unterschiedliche Speicherbereiche konzipiert sind. Es kamen einige Spiele und Programme für den VC20 auf den Markt, und leider benötigen viele der Spiele und Programme unterschiedliche Speichererweiterungen. Will man jede Software laufen lassen, so benötigt man alle Speichererweiterungen. Einige Programme laufen sogar nicht, wenn zuviel Speicher vorhanden ist.

Viele Spiele und Utilitys gibt es in Form einer ROM-Erweiterung (Cartridge). Manche belegen 8 KB Speicher, andere 16 KB. Einige dieser Cartridges laufen nur in einem ROM, weil sie in den eigenen Speicher schreiben. Lädt man diese Programme in eine Speichererweiterung, dann zerstören sie sich selbst.

Der VC20 kann wie der C64 exakt 64KB adressieren. Davon sind 16 KB belegt durch Kernel (8 KB) und dem BASIC-Interpreter (8 KB). Weitere 8 KB sind für IO-Ressourcen reserviert. Die restlichen 40 KB sind frei oder mit internem RAM belegt.

Auf Grund der Problematik der verschiedenen Speicherkonfigurationen wurde die Idee der *Final Expansion* geboren.

Der erste Prototyp der *Final Expansion* war handgefädelt und hatte zwei SRAM-Bausteine mit insgesamt 40 KB. Der zweite Prototyp der *Final Expansion* ist auf selbstgefertigter Zweifach-Layer-Printplatte (Dank an for(;;)) aufgebaut. Vor der Produktion einer Print kam im Forum64 die Idee auf, die FE mit der genialen SD2IEC-Platine zu kombinieren. Damit hat man zusätzlich Zugriff auf bis zu 2 GB in Form einer modernen SD-Karte und kann auf eine Floppy vollends verzichten. Dank JMP\$FCE2

wurde das extrem dichte Layout auf engsten Raum möglich, und das mit nur 2-fach Layer. Nun steht eine moderne Cartridge für den VC20 zur Verfügung:

- Erweiterung auf maximal 512 KB RAM (max. 40 KB sichtbar, Rest in Banks). Dabei sollen alle gängigen RAM-Ausbauten einstellbar sein: 3 KB, 8 KB, 11 KB, 16 KB, 19 KB, 24 KB, 27 KB, (35 KB)
- Jeder einzelne Speicherblock lässt sich gegen Überschreiben schützen, so dass er wie ein ROM aussieht (ladbare Cartridge).
- Jeder einzelne Speicherblock lässt sich deaktivieren, so dass die FE mit anderen Cartridges betrieben werden kann.
- Eine EEPROM-Erweiterung um 512 KB. Das EEPROM ist vom VC20 aus programmierbar.
- Bis zu 2 GB Massenspeicher in Form einer modernen SD-Karte als zuverlässiger Floppy-Ersatz. Dadurch einfacher Datenaustausch mit dem PC (und damit dem Internet) plus Starten von Programmen und Modulimages (die keineswegs erst in das Flash geschrieben werden müssen, sondern direkt von der SD-Karte geladen werden können)
- Zudem kann es extern durch ein LCD-Display erweitert werden. Ein eingebauter Schnelllader unterstützt die Turboroutinen im sd2iec.
- Zwei Register an einer IO-Adresse zur Konfiguration der FE per Software. Die Register lassen sich ausblenden.
- Eine batteriegepufferte Echtzeituhr

Mein Dank geht an alle Personen, die fleißig an dem Projekt mitarbeiten, insbesondere an Peter Sieg, JMP\$FCE2, for(;;), Wiesel, Unseen und Sign-Set vom Forum64.

Was kann man damit machen?

Nach dem Einschalten des VC20 (bzw. nach einem Hardreset) ist ein 8 KB großes Stück des EEPROM der *Final Expansion* aktiviert (sichtbar). Das Kernel des VC20 findet eine Cartridge und startet die Firmware der *Final Expansion*. Wenn keine Taste gedrückt wird, kommt ein Menü, mit dem man die *Final Expansion* konfigurieren kann:

- Speicherausbau wählen und VC20-BASIC starten. Man wählt eine RAM-Erweiterung und startet mit <Enter>.
- Eine Cartridge aus dem EEPROM direkt starten. Man wählt ein Spiel oder Programm und startet mit <Enter>.
- Ein Spiel oder Programm direkt von SD/Diskette starten. Das Verzeichnis wird angezeigt: Eintrag wählen und mit <Enter> starten.

Mit Zusatzprogrammen kann man die *Final Expansion* nach eigenen Wünschen konfigurieren: beispielsweise mit dem *Firmware Flasher* – damit kann man die Firmware der FE in das EEPROM schreiben: Cartridge oder Programm für den Cartloader in das EEPROM speichern.

Das Konzept der LOADER-Dateien

Diese Dateien enthalten neben der im Flash abgelegten Firmware die »Intelligenz« des Moduls. Dem Benutzer präsentieren sie sich als selbst veränderbares »Startmenü«, das dem Benutzer nur noch ausführbare Programme anzeigt und andere, für den Benutzer unwichtige Dateien verbirgt. In den LOADER-Dateien ist hinterlegt, ob das Programm ein BASIC-Programm, eine Cartridge oder ein Maschinencodeprogramm ist und welche Speicherkonfiguration das Programm benötigt. Einmal angelegt, muss sich der Benutzer um diese Details nicht mehr kümmern und sich auch nicht mehr merken, welches Programm welche Speicherkonfiguration benötigt.

Aufbau der *Final Expansion* 3.1

Es wird eine Printplatte gefertigt, die den Slot-Anschluss für den VC20 hat und Platz für den CPLD (PLCC-44 Sockel) sowie die beiden Speicherbausteine (DIL-32) bietet. Der CPLD ersetzt eine Menge TTL-Chips und ist für



Abb. 100:
Hauptmenü

die Adressdekodierung sowie für die softwaregesteuerte Konfiguration (Latch) zuständig. Der CPLD muss erst per JTAG für seine Aufgabe programmiert werden. Atmel bietet hierzu kostenlos eine Software an (WinCUPL), mit der man die Logik des CPLD »verdrahten« kann.

Neben dem CPLD und ein paar Bauteilen benötigt man noch den externen Speicher und einen Controller:

- ein 512-KB-SRAM (628512) als Erweiterung des Arbeitsspeichers
- ein 512-KB-EEPROM (29F040) als Speicher für die Firmware (Modul-Menüprogramm) und oft benutzte Programme bzw. Cartridges
- optional ein Atmega644 und eine SD-Karte als Massenspeicher bzw. als Floppy-Ersatz (SD2IEC)

Betriebsmodi der Cartridge

Start-Modus: Modus nach dem Einschalten bzw. nach einem Reset. Es ist nur ein 8 KB großes Stück des EEPROM an der Adresse \$A000 sichtbar, damit die Firmware startet.

ROM-Modus: Dieser Modus blendet das EEPROM ein. Das EEPROM ist in allen Speicherbereichen sichtbar (32 KB zur selben Zeit). Schreibzugriffe gehen in das SRAM, so kann man einfach vom EEPROM in das SRAM kopieren.

RAM-Modus: Speichererweiterung mit programmierbarer Größe und optionalem Schreibschutz

Super-RAM: In diesem Modus kann das ganze 512-KB-RAM in 32-KB-Stücke eingeblendet werden. Schreibschutz funktioniert hier nicht.

OFF-Modus: Cartridge ist »unsichtbar«, also ausgeschaltet (Reaktivierung nur durch Reset).

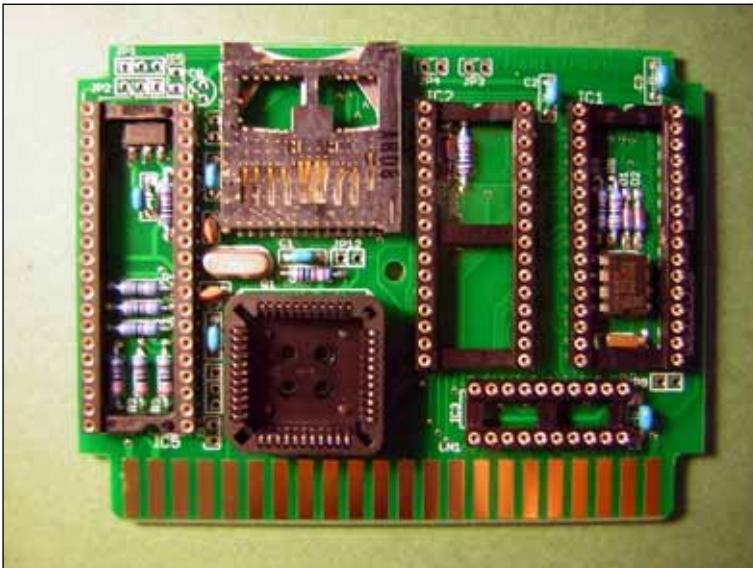
Program-Modus: In diesem Modus kann das EEPROM geflashed werden. Achtung: nur aktivieren, wenn man weiß, was man tut ...

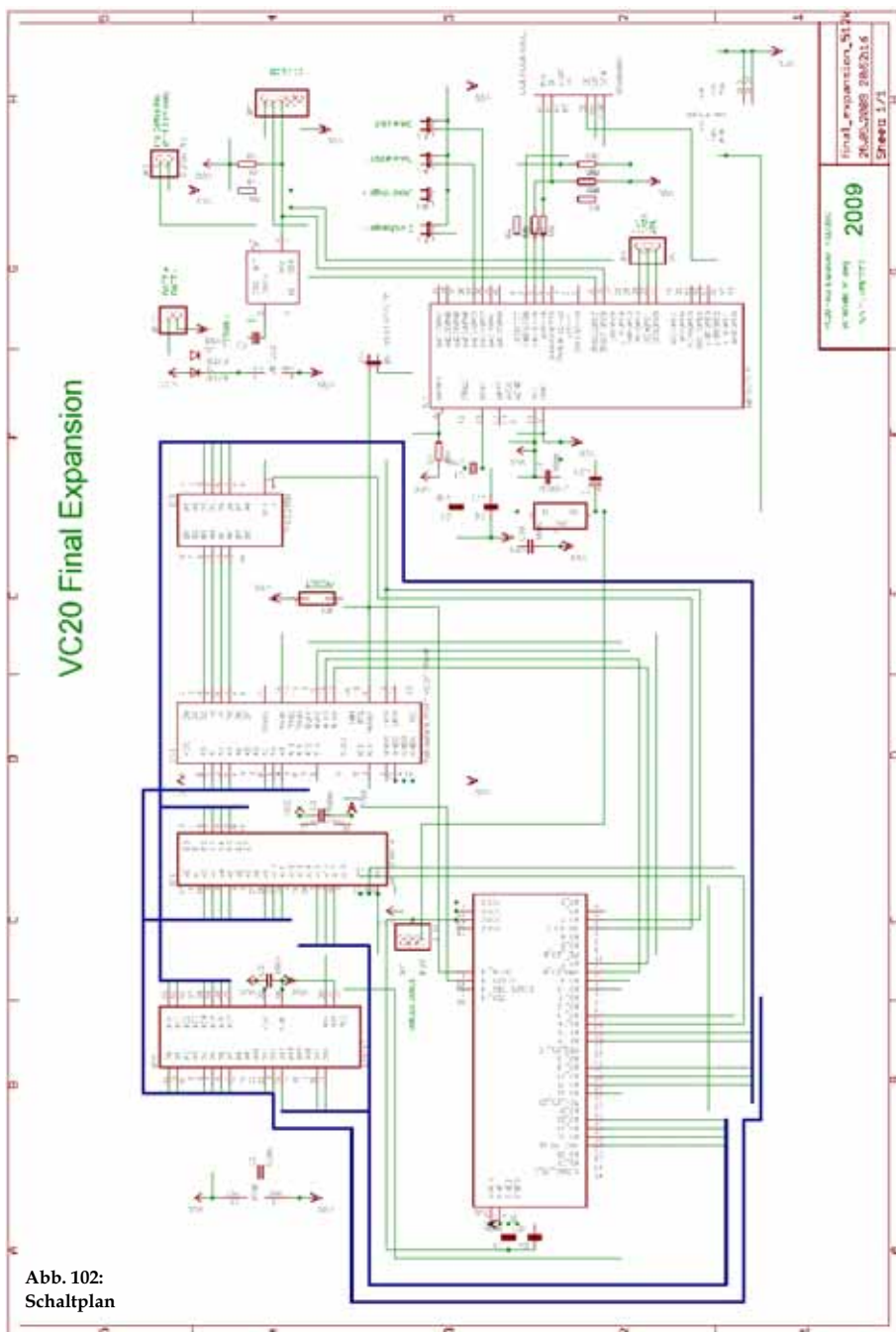
Links zum Projekt

<http://vc20final.t-winkler.net/index.html>

<http://www.for8bits.com/>

Abb. 101:
*Final
Expansion*
(ohne ICs
und Jumper)





Neigungssensor + AVR als Joystick

Nach Anregung von Till Harbaum:

<http://www.harbaum.org/till/palm>

Die wunderbare Seite von Till ist immer wieder einen Besuch wert! Die Programme sind GPL.

Ich hatte vor Jahren schon einmal diese tolle Seite besucht, aber der verwendete Sensor ADXL202 war mit zu teuer. 2009 kostet er bei segor ca. 26 €. Über Kongkong habe ich mir einen für ca. 20 € inkl. Versand bestellt. Dann habe ich durch weitere Internetrecherchen herausgefunden, dass dieser Sensor auch im Microsoft Sidewinder Freestyle verbaut ist! Und da kommt man heute nicht nur günstig dran, sondern es ist auch schon fast alles aufgebaut!

So sieht der Microsoft Sidewinder Freestyle (der mit dem »Bewegungssensor«) aus:

Abb. 104:
Sidewinder
mit Bewe-
gungssensor



So sieht der **Neigungssensor + AVR als Joystick** aus (umrandet der Sensor ADXL202):

Nach Anregung von Till Harbaum:

<http://www.harbaum.org/till/palm>

Die wunderbare Seite von Till ist immer wieder einen Besuch wert! Die Programme sind GPL.

Ich hatte vor Jahren schon einmal diese tolle Seite besucht, aber der verwendete Sensor ADXL202 war mir zu teuer. 2009 kostet er bei segor ca. 26 €. Über Kongkong habe ich mir einen für ca. 20 € inkl. Versand bestellt. Dann habe ich durch weitere Internetrecherchen herausgefunden, dass dieser Sensor auch im Microsoft Sidewinder Freestyle verbaut ist! Und da kommt man heute nicht nur günstig dran, sondern es ist auch schon fast alles aufgebaut!

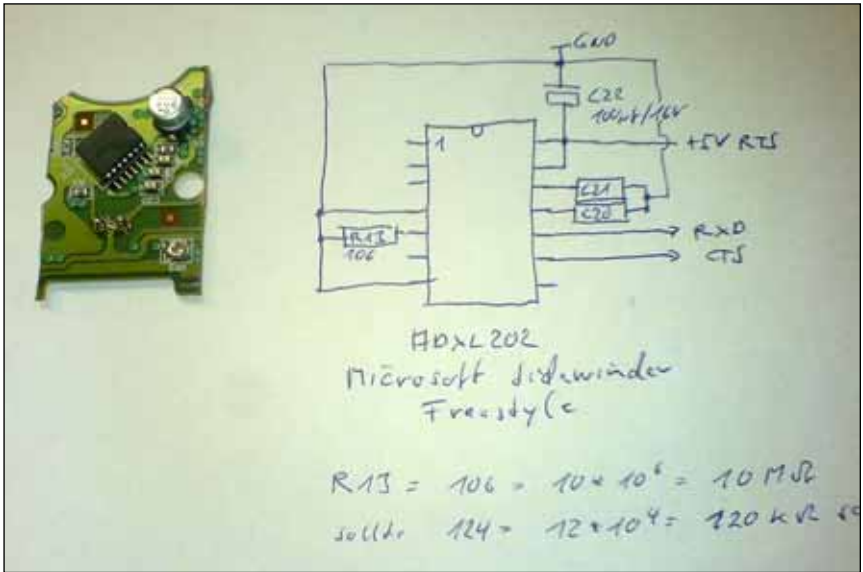
So sieht der Microsoft Sidewinder Freestyle (der mit dem »Bewegungssensor«) aus:



Abb. 104:
Sidewinder
mit Bewe-
gungssensor
Hier
Seite



Abb. 107:
Der abge-
trennte
Sensor



Per Multimeter und Nachmessen der Verbindungen, erkennt man, dass eigentlich alles schon korrekt drauf ist. Nur der Widerstand R13 ist mit $106 = 10 \times 10^6 = 10 \text{ MOhm}$ ca. 100x zu groß! Aber auf den restlichen Platinenteilen ist ein $104 = 10 \times 10^4 = 100 \text{ kOhm}$ -SMD-Widerstand zu finden. Den 106 entfernen und den 104 stattdessen einlöten! (Das übt man am besten zuerst an anderen SMD-Teilen auf den Reststücken.) Einfach beide Seiten mit dem Lötkolben erhitzen, bis sich der Widerstand von den Löt pads einfach wegschieben lässt. Sollte er am Lötkolben kleben, einfach mit einer Pinzette abnehmen. Es müssen dann auf einer Lochrasterplatine als Träger ein Sub-D-Stecker eingelötet und vier dünne Kabel zur Sidewinder-Platine verbunden werden (+5V und GND zum $100 \mu\text{F}$ -Kondensator, RTS+CTS zum ADXL202).

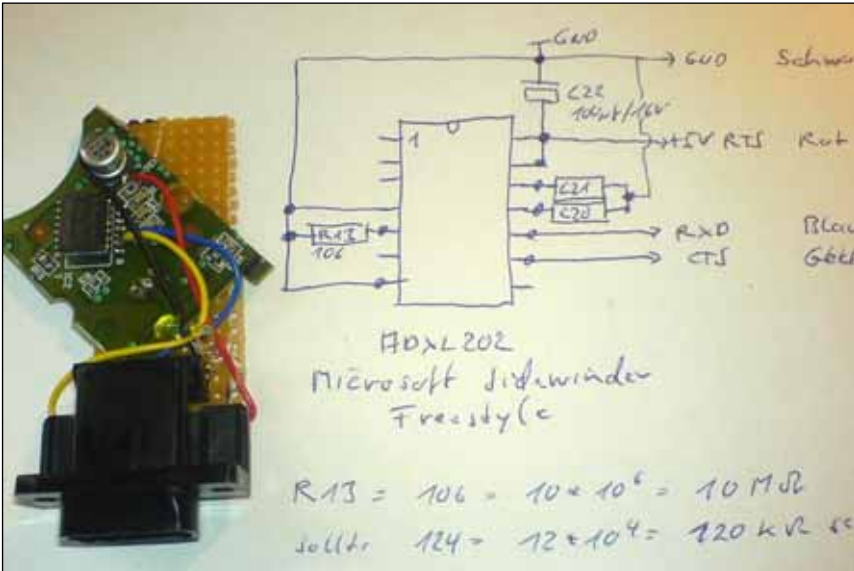


Abb. 108:
Auf Loch-
raster (mit
Heißkleber
fixiert und
mit Schalt-
planskizze

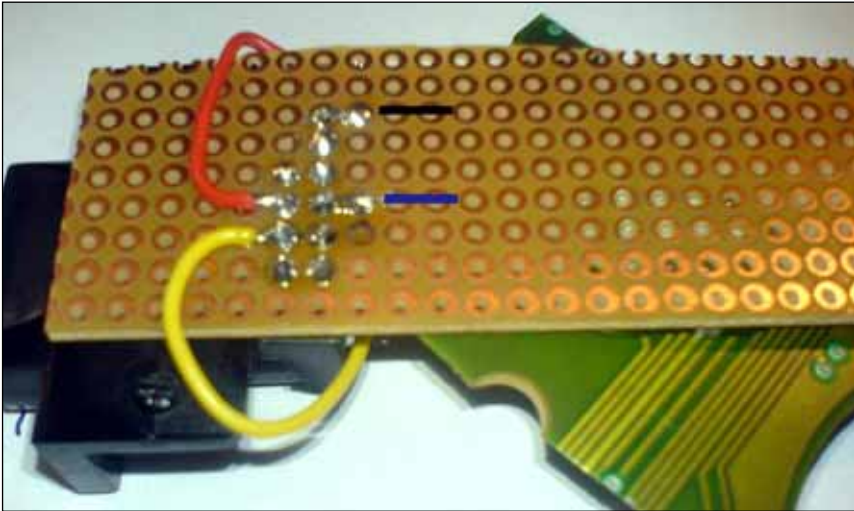
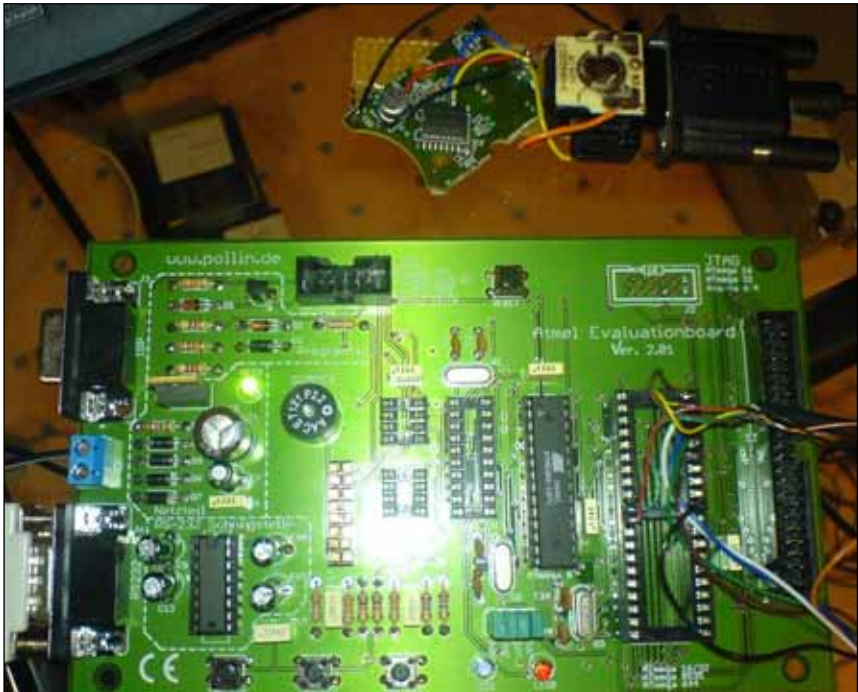


Abb. 109:
Ansicht von
unten

Jetzt muss der Sensor nur noch über einen AVR als Wandler mit einem Joystickport verbunden werden. Ab einer bestimmten Neigung wird dann der entsprechende Joystick-Pin auf GND gelegt (das Gleiche wie Joystickschalter geschlossen).

Hier der Testaufbau auf einem Pollin-Evaluation-Board mit Atmega8. Im Bild oben ist der Sensor. Anschluss über 9-pol Sub-D (damit das auch noch am Palm-PDA passt). Oben drauf der Feuerknopf. Die Signale habe ich an der ATmega32-Fassung über Adapter abgenommen.

Abb. 110:
Testaufbau
auf Pollin
Evaluation
Board



Das Bascom-Testprogramm

```
$regfile = "m8def.dat"      'definieren des verwendeten Chips
$crystal = 1000000          'interner 1MHz Takt (default)
$baud = 1200                'definieren Baudrate der RS232

Dim X As Word
Dim Y As Word

Config Pinc.2 = Output
Config Pinc.3 = Output
Config Pinc.4 = Output
Config Pinc.5 = Output
```

```

Main:
Do
  Pulsein X , Pinc , 0 , 1
  Pulsein Y , Pinc , 1 , 1
  Print "X=" ; X ; " - Y=" ; Y
  Waitms 100
  If X > 21 Then
    Portc.5 = 0
  Else
    Portc.5 = 1
  End If
  If X < 19 Then
    Portc.4 = 0
  Else
    Portc.4 = 1
  End If
  If Y > 21 Then
    Portc.2 = 0
  Else
    Portc.2 = 1
  End If
  If Y < 19 Then
    Portc.3 = 0
  Else
    Portc.3 = 1
  End If
Loop

```

C.0 = X-in
 C.1 = Y-in
 C.2 = Up
 C.3 = Down
 C.4 = Left
 C.5 = Right
 Fire wird direkt durchgereicht

Die serielle Ausgabe diente nur zur Kontrolle. Die Auflösung in Bascom bei internem 1-MHz-Takt ist ein wenig gering. In der Nullposition wird 20 zurückgeliefert. Der Wert geht dann jeweils um +/- 5 hoch bzw. runter, also 15-20-25.

Evtl. kann die Waitms auch noch raus bzw. kleiner, oder man lässt den AVR mit Quartz und höherem Takt laufen. Die PINs sollten besser Open Collector bei 1 sein! Also noch eine Menge zu optimieren.

TFT-Kontrollmonitore für C64

Der Betrieb eines C64 an einem TFT-Fernseher ist nicht unproblematisch. Zum Kauf sollte man seinen C64 mitnehmen und im Geschäft testen, ob es überhaupt geht und die Bildqualität ausreichend ist.

Für Testzwecke bzw. als Kontrollmonitore in der Reparaturrecke kann man auch die TFT-Displays der XBOX (Farbdarstellung und ohne Umbau direkt verwendbar) oder das der PS2 (Umbau nötig und nur S/W-Darstellung) verwenden. Insbesondere letzteres ist 2009 für wenige Euro verramscht worden.

Pin	Farbe	Funktion
1	braun	Ground
2	violett	Audio rechts
3	weiß	Ground
4	grau	Audio links
5	grün	S-Video Y
6	----	
7	----	
8	blau	Video Ground
9	rot	Blau
10	---	
11	gelb	Rot
12	orange	Grün

Für BAS:
Pins 5, 9, 11+12 verbinden = Signal

8=GND

Abb. 110:
PS2-TFT



Einfachst-PIC-Programmer (Ludipipo)

Die Idee zu dieser Schaltung stammt von Erik Herman und wurde zum ersten Mal am 12. Dezember 1995 von Ludwig Chatta veröffentlicht. Ludi = Kosenamen für Ludwig; pi = pic; po = Programmer. So entstand der Name *Ludipipo*.

Mittlerweile gibt es zahlreiche Varianten des *Ludipipo* oder auch JDM-Programmer. Der JDM-Programmer wurde entwickelt von Jens Dyekjær Madsen.

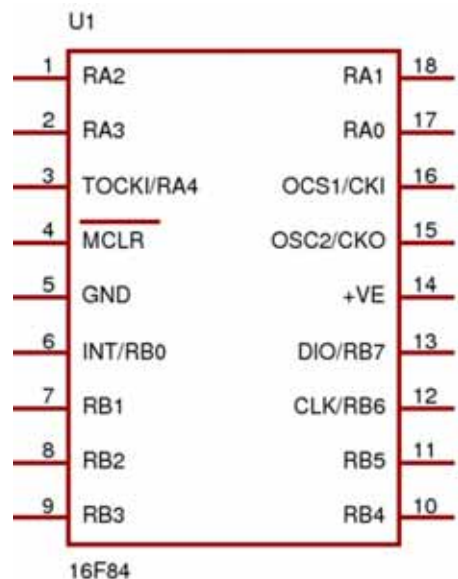
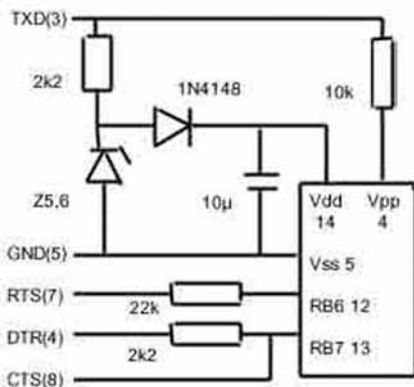
Diese einfache Schaltung arbeitet an einer seriellen Schnittstelle eines PCs und benötigt keine externe Spannungsversorgung. Die Pin-Nummern in Klammern sind die Anschlüsse einer 9-poligen Sub-D-Buchse. An Notebooks kann es zu Problemen kommen, wenn die Spannungen am seriellen Port zu niedrig sind.

Geeignet ist der PIC-Programmer für alle 18-poligen PICs (16F84/A, 16F627/8/A etc.). Als Programmiersoftware eignen sich *IC-Prog* und alle Programme, die einen JDM-kompatiblen Programmer unterstützen.

Generell empfehle ich den Blick in ein Online-Auktionshaus mit der Suche nach »JDM Programmer«, da kann man für zirka 10 € fertige Geräte kaufen.

Abb. 111:
Schaltplan
(unten)

Abb. 112:
Pinbelegung
(rechts)

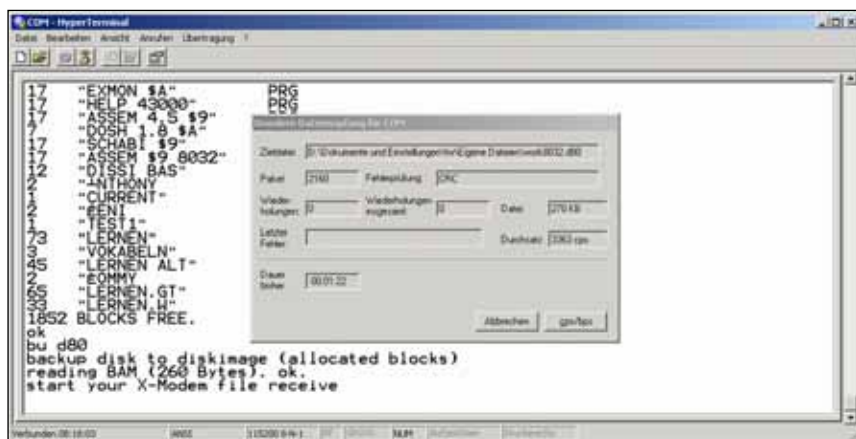


XS1541 – AVR zwischen PC- und CBM-Floppy

(nach Ing. Thomas Winkler)

Mit diesem neuen Wandler zwischen den Welten auf AVR-ATMega32-Basis lassen sich sowohl serielle IEC-Floppys als auch parallele IEEE-Floppys untereinander sowie mit einem PC verbinden. Dabei ist auf PC-Seite keine spezielle Software erforderlich, *Hyperterm* ist ausreichend (38400 8N1). Dateien werden zwischen PC und AVR über Xmodem übertragen.

Abb. 113:
Datenüber-
tragung mit
*Hyper-
Terminal*



Ich habe den XS1541 Adapter auf einem Pollin-AVR-Evaluation-Board aufgebaut. Dort ist schon alles drauf, was benötigt wird:

- ATMega32 mit Programmiermöglichkeit
- MAX232 als RS232-Pegelwandler an RX+TX
- 16-MHz-Quartzschaltung

Zum seriellen IEC-Anschluss (siehe mmc2iec-Kapitel) müssen anschließend folgende Verbindungen hergestellt werden:

AVR	PIN	Signal	J4-Anschlusspunkt auf Pollin-Board
pd7	21	clock	33
pd6	20	reset (nicht verbunden)	32
pd5	19	srq	31
pd4	18	data	30
pd2	16	atn	28
+gnd	–	Masse	35, 37, 39



Abb. 114:
XS1541, rea-
lisiert auf
dem Pollin-
Board

Firmware unter: <http://petersieg.webng.com/xs1541/xs1541.zip>

Befehlssyntax

Die Befehle können direkt im Terminal-Programm eingegeben werden und werden direkt vom Controller ausgeführt. Dadurch ist man komplett unabhängig von der Hardware. Das funktioniert auf jedem PC: unter Windows, Linux oder AppleOS – es muss nicht mal ein PC sein.

Einstellungen: 38400 Baud 8N1 – kein Hardware-Flowcontrol – kein Xon/Xoff

Einige Befehle kontrollieren direkt das Protokoll oder sogar die Portleitungen. Diese »Low level«-Befehle sind zum Test des Adapters bzw. für PC-Programme gedacht.

Jeder Befehl hat einen Mussteil und einen Kannteil. Der Kannteil kann ganz oder teilweise weggelassen werden und steht in rechteckigen Klammern. Manche Befehle haben mehrere Kürzel.

h[elp] Listet alle Befehle auf, die in der Firmware des Controllers implementiert sind. Es kann ›h‹, ›he‹, ›hel‹ oder ›help‹ eingegeben werden.

iee[e] Der aktive Bus wird zum parallelen IEEE-488-Bus geschaltet. Alle nachfolgenden Befehle (außer IEC) beziehen sich nun auf den IEEE-488-Bus. Dieser Modus ist die Standardeinstellung nach einem Controller-Reset.

iec Der aktive Bus wird zum seriellen IEC-Bus geschaltet. Alle nachfolgenden Befehle (außer IEEE) beziehen sich nun auf den IEC-Bus.

dev[ice] oder **#** Damit wird die aktive Gerätenummer des aktiven Busses festgelegt oder abgefragt. Wenn dem Befehl ein numerischer Parameter folgt, wird die Gerätenummer gesetzt. Der Befehl allein ohne Parameter listet die aktuelle Einstellung auf. Für beide Bussysteme ist die voreingestellte Adresse nach einem Reset 8.

@ Dieser Befehl sendet einen Befehl an die aktive Floppy (Befehlskanal 15) oder fragt den Fehlerkanal/Status ab, wenn der Befehl allein (ohne Parameter) gesendet wird.

dir[ectory] oder **ca[talog]** oder **\$** Listet das Verzeichnis der eingelegten Diskette auf. Als Parameter kann optional ein String mitgegeben werden. Das Format entspricht dem des Floppy-DOS (load "\$..." ,8). Bei Doppelfloppy sollte man als Parameter zumindest das Laufwerk (0 oder 1) mitgeben, sonst wird das Verzeichnis beider Laufwerke ausgegeben, was zu einem Fehler führt, wenn nicht beide Laufwerke formatierte Disketten eingelegt haben.

filec[opy] oder **fc** Kopiert Dateien von Floppy zu Floppy. IEC zu IEEE oder umgekehrt ist möglich.

filed[ump] oder **fd** Erstellt einen Hexdump von einer Datei.
bac[kup] Erstellt eine Image-Datei einer ganzen Diskette (d64, d72, d80, d81, d82). Die Image-Datei wird per X-Modem-Protokoll direkt auf die Festplatte des PC gespeichert.
res[tore] Schreibt eine Image-Datei auf eine Diskette zurück (d64, d72, d80, d81, d82). Die Image-Datei wird per X-Modem-Protokoll vom PC zum Adapter übertragen.
lo[ad] Lädt eine Datei zum PC (P00, T64). Die Datei wird per X-Modem-Protokoll direkt auf die Festplatte des PC gespeichert.
sa[ve] Schreibt eine Datei auf eine Diskette zurück (P00, T64). Die Datei wird per X-Modem-Protokoll vom PC zum Adapter übertragen.

XU1541 – USB-Interface für CBM-Floppys

Link: <http://www.trikaliotis.net/xu1541>
<http://opencbm.sf.net/>

Das XU1541-Projekt wurde ursprünglich von Till Harbaum gestartet. Ende 2007 übernahm Spiro Trikaliotis die weitere Betreuung.

Da inzwischen immer weniger PCs und insbesondere Notebooks noch Parallelports haben, ersetzt XU1541 die davor eingesetzten X*1541-Kabel und ermöglicht so den Anschluss von IEC-Floppys an den USB-Bus. Dabei wird eine CBM-Floppy allerdings nicht zum universellen Wechseldatenträger, sondern nur über spezielle Low-Level Befehle mit

spezieller Software über USB ansprechbar.

Die Softwareunterstützung für XU1541 bildet das OpenCBM-Projekt. OpenCBM stellt einige Kommandozeilentools zur Verfügung, zum Beispiel d64 copy.exe zum Transferieren von D64-Images von und zur Floppy:

Image schreiben: d64copy <image.d64> 8

Image lesen: d64copy 8 <image.d64>

Abb. 116:
XU1541



Kampf dem Gilb oder: Wie man vergilbte Plastikteile wieder hell bekommt

Wer kennt sie nicht, die hässlich vergilbten Plastikgehäuse unserer Schätzchen ... Im Forum64 und dem a1k.org-forum hatte ich einige Angaben und Tipps gelesen. Ausprobiert habe ich es dann mit Wasserstoffperoxid 9 %. Es geht auch 12 %. Ich habe dabei ein Produkt aus dem Färisorbedarf verwendet, welches eine cremige Konsistenz hatte.

Abb. 117:
Vor der
Reinigung:
Oberseite ei-
ner A2000-
Tastatur



Die Behandlung

Zuerst gut mit Wasser und Seife reinigen. Trocknen. Danach auf einer Unterlage, draußen an der Sonne, mit dem Mittel satt einpinseln. 3-4 Stunden unter Sonnenstrahlung einwirken lassen. Unter viel Wasser abspülen und trocknen. Je nach Bleichungsgrad ggf. die Behandlung wiederholen.

Abb. 118:
Nach der
Reinigung:
Oberseite ei-
ner A2000-
Tastatur



C64DTV

Das 64 DTV ist die Consumer-Inkarnation der erfolgreichen Bemühungen, einen C64 in FPGA/ASIC zu implementieren. Die FPGA-Entwicklungsstufe war der C-One.

Mit nur drei Chips, ASIC (mit CPU, CIAs, SID und VIC), 2 MB Flash-ROM (mit Kernel, BASIC, Char und Programmen) und RAM konnte ein kompletter C64 inklusive Programmen in einen nachempfundenen Competition Pro Joystick eingebaut werden. Die NTSC-Variante, welche sich vom Softwareumfang und auch in der Hardware von der PAL-Version unterscheidet, wird hier nicht betrachtet. In der PAL-Variante ist zum Beispiel *Alleykat* hinzu gekommen, während *World Games* weggefallen ist.

Das 64DTV (DTV=Direct To TV) – so haben findige Hacker schon bald herausgefunden – kann um so einiges erweitert und modifiziert werden. Man kann zum Beispiel eine PS/2-Tastatur anschließen. Durch Halten von <STRG> beim Einschalten gelangt man so direkt ins BASIC!

Man kann das DTV um einen IEC-Anschluss erweitern und so beispiels-

weise originale Floppys oder MMC2IECs anschließen.

Weiterhin können Joystickport-Anschlüsse angebracht werden, um bessere und austauschbare Joysticks nutzen zu können.

Darüber hinaus kann auch das 2-MB-Flash-ROM verändert werden – zum Beispiel, um weitere oder andere Spiele dauerhaft im Flash zu haben.

Für weitere Informationen helfen sicher Suchmaschinen oder die Links in diesem Kapitel.

Abb. 119:
Ein modifiziertes DTV



(von Daniel Svegert/MadModder)

DTV-Anschlusspunkte IEC + PS/2

Abb. 121:
Anschluss-
punkte der
DTV-Platine
für den
Anschluss
einer PS/2-
Tastatur

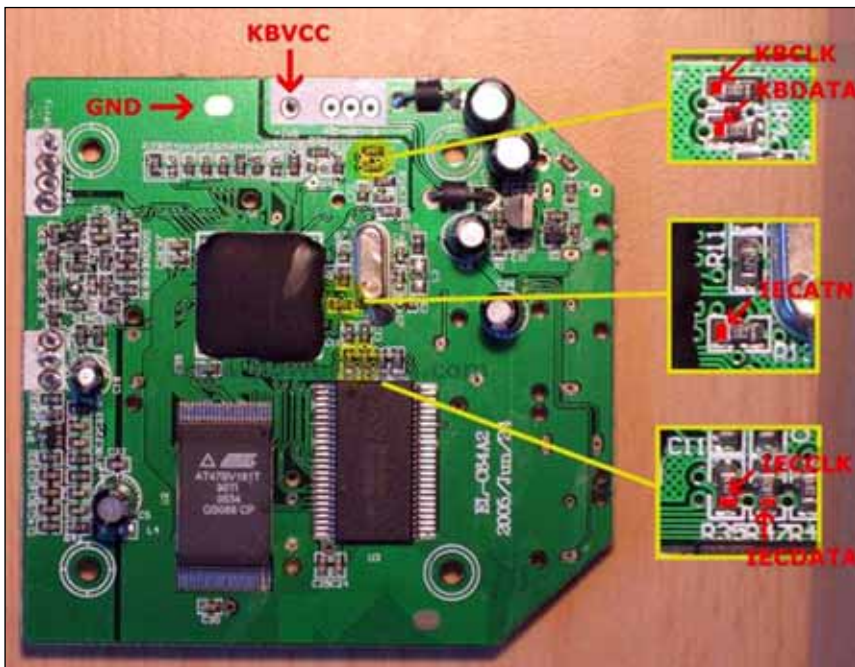
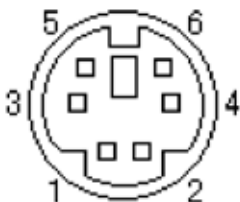


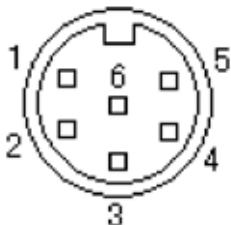
Abb. 122:
PS/2-Tasta-
turstecker



PS/2-Tastatur-Stecker: (Ansicht auf Steckerfront
(m) bzw. Lötseite Kupplung (f))

- | | |
|----------|----------|
| 1. DATA | 4. +5V |
| 2. N. C. | 5. CLOCK |
| 3. GND | 6. N. C. |

Abb. 123:
IEC-Stecker



IEC-(Floppy)-Stecker: (Ansicht auf Steckerfront
(m) bzw. Lötseite Kupplung (f))

- | |
|-------------------------|
| 1. SRQ (unbeschaltet) |
| 2. GND |
| 3. ATN |
| 4. CLOCK |
| 5. DATA |
| 6. RESET (unbeschaltet) |

Anschlusspunkte für Joysticks

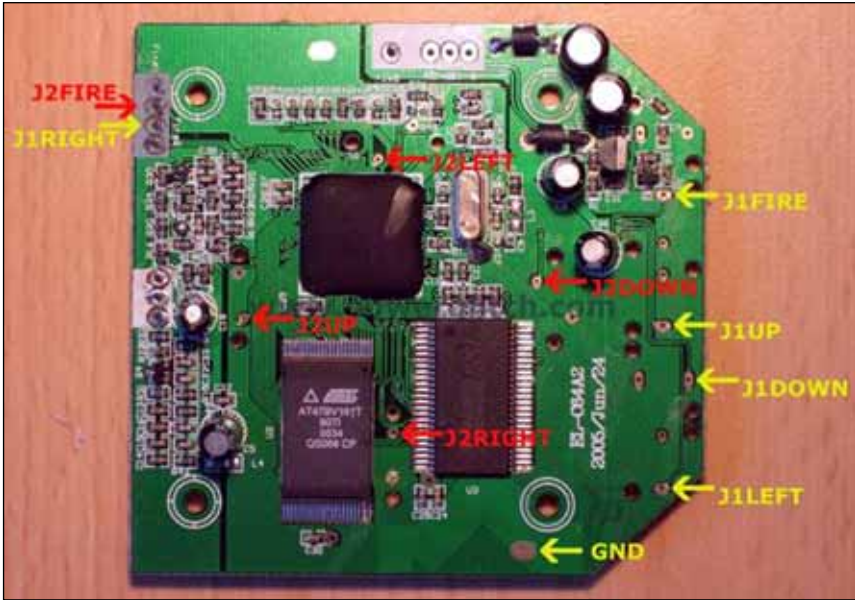


Abb. 124:
Anschluss-
punkte für
externe
Joysticks

Joystick-Stecker: (Ansicht auf Kupplungsfront (f) bzw. Lötseite Einbaustecker (m))

1. UP
2. DOWN
3. RIGHT
4. LEFT
5. (POTY)
6. FIRE
7. +5V
8. GND
9. (POTX)

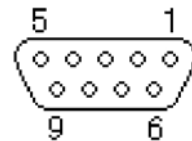


Abb. 125:
Pinbelegung
der Joystick-
buchse

Hier Bilder meines ersten Mods: Lochrasterplatine 160x100mm als Grundträger. Rechts: 5 V, PS/2, Joy2

Abb. 126:
Mein erster
Mod

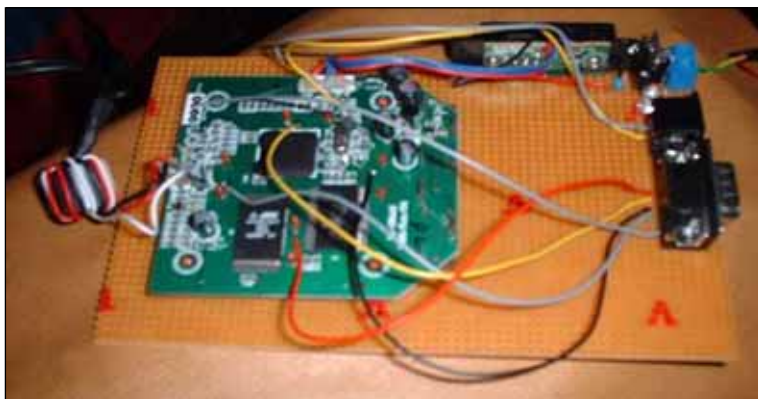


Abb. 127:
Mein erster
Mod (Seiten-
ansicht)

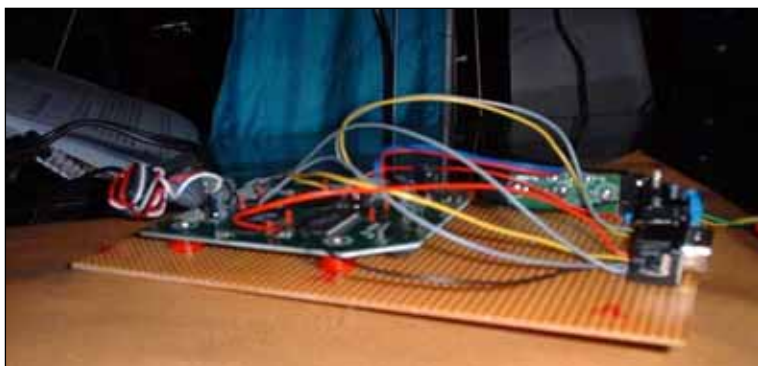
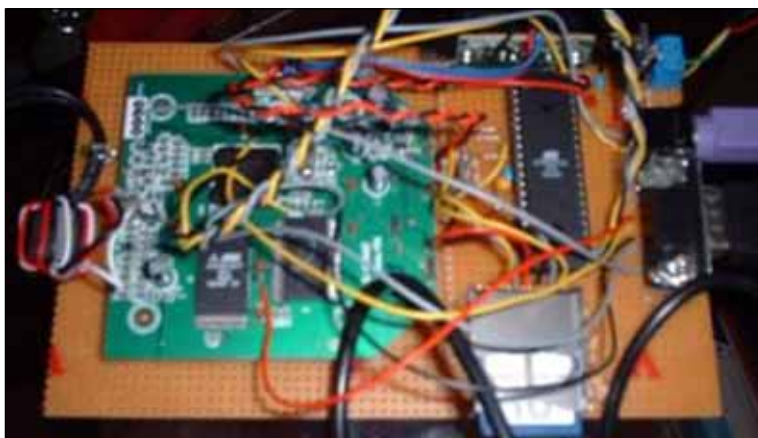


Abb. 128:
Der Mod mit
MMC2IEC



Easyflash

Dieses wunderbare Tool von Roland Tögel ermöglicht es auf sehr einfache Weise, sein DTV mit einigen neuen Spielen zu bestücken. Dabei bleiben alle originalen Spiele, Dateien und das originale Menü weiterhin zugänglich.

Es werden maximal 768 Blocks im hinteren, freien Bereich mit Spielen belegt und ein neues Menü wird installiert. Aus diesem Menü sind direkt die neuen Spiele und das Original-Menü aufrufbar. Außerdem sind die zeitintensiven Intros durch das neue Menü ebenfalls entfallen, so dass der Start erheblich beschleunigt wird. Ich habe mit diesem Tool selbst gearbeitet und alles hat wunderbar funktioniert. Es ist allerdings nur für die PAL-Version mit AT47...-Flashchip nutzbar.

Das Tool ist ideal, wenn man schon eine Floppy oder MMC2IEC am DTV angeschlossen hat. Dann das Tool und die gewünschten Programme in PRG oder codierter DTV-Form auf Floppy oder SD-Karte aufspielen, Tool starten und <L> zum Laden von Standard-C64-Programmen oder <E> für schon codierte DTV-Programme.

Zuerst den Dateinamen, so wie er auf dem Datenträger ist, eingeben. Nach dem Laden den Namen, den das Programm/Spiel im neuen Menü haben soll (wird mit dem Dateinamen vorbelegt, man kann also auch einfach <Return> drücken), danach die Startadresse oder einfach 0100 eingeben. Die vierte Eingabe kann leer bleiben. Das für alle Programme wiederholen, bis maximal der verfügbare Platz von 768 Blocks erschöpft ist. Dann erst <F> für das Flashen der neuen Programme und des entsprechenden Menüs eingeben.

Beim Flashen sollte man immer sicherstellen, dass es keine Unterbrechung der Stromzufuhr geben kann! Zudem sollte man am besten nur so genannte One-File-Programme verwenden. Das sind Programme, die selbst alles beinhalten, was zum Ablauf benötigt wird, die also keine separaten Dateien außer der PRG-Datei selbst brauchen. Das Tool gibt's hier: <http://noname.c64.org/csdb/release/?id=41413&show=review#review>

http://picobay.com/dtv_wiki/index.php?title=C64_DTV_Hacking_Wiki
www.powerglitch.com/index_dtv.html
www.geocities.com/dtvhacking
www.kahlin.net/daniel/dtv/http://galaxy22.dyndns.org/dtv/
www.geocities.com/homeofoscarvermeulen/dtv/dtv_pal.html

**L
i
n
k
s**

Amiga 500

Der Amiga 500 ist der meistverkaufte Amiga-Computer von Commodore. Er wurde 1987 auf der CeBIT zusammen mit dem Amiga 2000 vorgestellt und besaß ähnliche Leistungsdaten wie der 2000er. Zielgruppe für den Rechner waren vor allem Privatanwender mit kleinerem Geldbeutel.

Zulasten der Aufrüstbarkeit wurde auf ein Desktop- bzw. Towergehäuse verzichtet und stattdessen ein Design gewählt, das dem des Commodore 128 ähnelte. Der Amiga 500 war – mehr noch als der Amiga 2000 – zu seiner Zeit besonders als Spielecomputer beliebt, da Bild- und Tonqualität dem damals üblichen PC deutlich überlegen waren.

Abb. 129:
A500 mit TV-
Adapter A520
und Netzteil

Der Amiga 500 besaß eine mit 7,09 MHz (PAL-Variante) bzw. 7,14 MHz (NTSC-Variante) getaktete Motorola-68000-CPU und 512 KByte Arbeitsspeicher. Dieser konnte mittels eines Erweiterungsslots bis auf 2,8 MB



erweitert werden.

In den Jahren 1987 und 1988 wurde der Amiga 500 mit dem Amiga-Betriebssystem AmigaOS 1.2 (Kickstart) ausgeliefert, ab 1988 wurde dieses durch die wesentlich stabilere Version 1.3 abgelöst.

Wie beim Amiga 1200 und dem C64 muss an den Amiga 500 ein spezielles Netzteil angeschlossen werden – im Amiga 500 ist kein Netzschalter eingebaut.



Amiga 600

Der Amiga 600 war der erste Commodore-Rechner, der SMD-Technik nutzte und konnte daher viel kleiner als ein A500 gebaut werden. Als Besonderheit besaß er einen PCMCIA-Slot. Dank einer aktuelleren Kickstartversion und

Abb. 130: mittels PCMCIA-CF-Kartenadapter lassen sich mit ihm Amiga-Diskettenimages vom PC auf Amiga-Floppys zurückschreiben.

Amiga-Customchips

Hier als kleine Hilfestellung die häufigsten Amiga-Spezialchips im Überblick

Agnus – für Bildschirmaufbau (mit Denise)

8370 (NTSC) CPN: 318070-01

8361 (NTSC) CPN: 252125-01

8371 (PAL) CPN: 318071-01

8367R0 (PAL)

8372A (Fat Agnus) CPN: 318069-01

8375 (Fat Agnus) CPN: 390544-01

CPN: 318069-10 (A600)

8372B (Super Agnus) CPN: 318069-03

2048KB of memory

CIA – Complex Interface Adapter (Parallel, Seriell, Tastatur, Floppy)

8520 CPN: 391078-01/03

Denise – für Bildschirmaufbau (mit Agnus)

8362R5 (Denise)

8362R6 (Denise) CPN: 252126-01

8362R8 (Denise) CPN: 252126-02

8373R4 (Super Denise) CPN: 390433-01

CPN: 391081-10 (A 600)

Gary – I/O Prozessor (Bus Access, Floppy, IDE)

5719R2 (Gary) CPN: 318072-01

41-Fat Gary (Fat Gary) CPN: 390540-02

Paula – Sound, UART, Interrupt

8364	CPN: 252127-01
8364R7	CPN: 252127-02
	CPN: 391077-01

Im A3000/4000/1200 gab es weitere Spezialchips:

Alice – 8374	CPN: 391010-01
Amber	CPN: 390538-03
Budgie	CPN: 391425-01
Buster	CPN: 390539-11
DMAC	CPN: 390537-02
Gayle	CPN: 391424-02
Lisa	CPN: 391227-01
Ramsey	CPN: 390541-04

(CPN = Commodore Part Number)

Amiga-Fehlercodes nach dem Einschalten

Beim Starten führt der Amiga verschiedene Selbsttests durch und zeigt den Status über die Bildschirmfarbe und die Tastatur-LEDs an. Ändert sich der Bildschirm von Dunkelgrau über Hellgrau nach Weiß, ist alles in Ordnung.

Ohne bootfähige Diskette bzw. Festplatte am System sollte kurz danach ein Diskettensymbol am Bildschirm dazu auffordern, eine entsprechende Diskette einzulegen. Ansonsten können folgende Fehleranzeigen ggf. das Problem eingrenzen helfen. Generell muss ein entsprechender Baustein nicht gleich defekt sein! Oft sind es auch einfach nur Kontaktprobleme. In diesem Fall den Chip – wenn gesockelt – entfernen, Pins säubern und wieder einsetzen.

Farbcode	Diagnose
Dunkelgrau	Prozessor OK
Hellgrau	ROM-Prüfsumme OK
Weiß	Arbeitsspeicher (RAM) reicht aus (mindestens 256 KByte Chip-RAM und falls vorhanden, mindestens 256 KByte Fast-RAM) – das System startet ...
Rot	Fehler im Kickstart-ROM. Kontakte prüfen. Sitzt das ROM gut im Sockel?
Grün	Fehler in den unteren 256 KB Chip-RAM. Das können die RAM-Bausteine selbst, der Agnus (8371/2/5) oder sehr selten auch der Gary-Chip (5719) sein.
Blau	Fehler bei den Custom-Chips
Gelb	Ein Softwarefehler ist aufgetreten, bevor die Fehlerbehandlungsroutinen (»Guru Meditation«) installiert wurden

Blinkende LED der Tastatur

Anzahl Blinks	Diagnose
1	Fehler ROM-Prüfsumme
2	Fehler im RAM
3	Fehler interner Timer
4	Kurzschluss in der Tastaturmatrix

Amiga-Akku-Schaden

Hier wird die Reparatur eines Akkuschadens an einem Amiga 2000 in Bildern geschildert. Dank an Joachim Nemetz für diese gut gemachte Anleitung. Vom Prinzip her gilt die Beschreibung für alle derartigen Schäden, hervorgerufen durch auslaufende Akkus.

Achtung: Bitte die Akkus regelmäßig kontrollieren und tauschen, bevor es zu solchen Schäden kommt! Am besten danach keinen Akku mehr direkt auf der Platine einsetzen, sondern etwas abseits, so dass ein späteres Auslaufen zu keinen Schäden mehr an der Elektronik führen kann.

Sichtbare Schäden an der Platine und einigen Pins des 68000ers:

Abb. 131:
Defekter
Akku

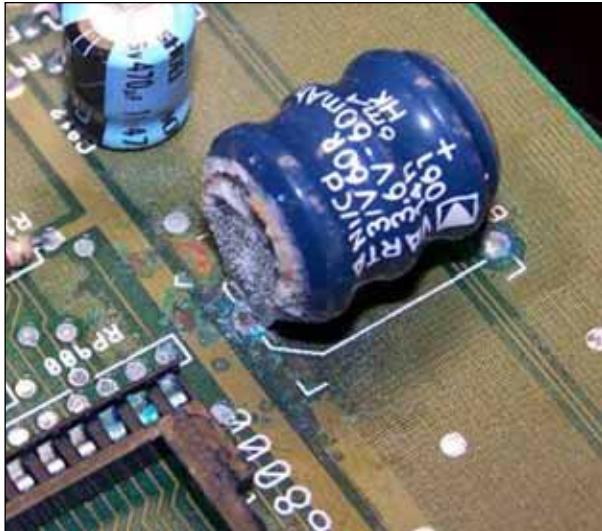




Abb. 132:
Platine ohne
Akku

So sieht die Platine nach dem Entfernen des Akkus aus.



Und das ist das, was man in den Pins des Sockels findet. Deshalb immer auch leicht sichtbare Schäden kontrollieren und lieber ein, zwei Teile mehr tauschen

Abb. 133:
Reste

Abb.
134-136:
Weitere
Arbeits-
schritte



Platine nach dem Abschleifen mit einem Glasfaserstift. Danach neutralisieren mit Essig und nachspülen, beispielsweise mit Spiritus



Danach besprühen mit Lötack, zum Beispiel SK10



Socket und Akku eingelötet,
Chip eingesetzt und – läuft wie-
der

CF-Bootdiskette erstellen

Man braucht dazu folgende Dateien: fat95, CF0 und compactflash.device. Diese Dateien gibt es unter anderem hier:

www.forum64.de/wbb3/index.php?page=Thread&threadID=22055

Man nehme eine normale Amiga-Diskette und formatiere sie auf 720 KB. Unter Windows muss man dazu auf *Ausführung* gehen und `format a: /t:80 /n:9` eingeben. Dann kopiert man die oben genannten Dateien auf die Diskette.

Jetzt legen wir die Diskette in den Amiga und starten crossdos. Alle drei Dateien erst einmal ins Ramdrive kopieren. Diskette herausnehmen und Workbench-Diskette einlegen. Evtl. muss man noch etwas Platz schaffen, um die drei Dateien aus dem Ramdrive auf die Diskette kopieren zu können:

- fat95 ins Verzeichnis L
- compactflash.device ins Verzeichnis Devs
- CF0 einfach ins Stammverzeichnis

Nach einem Neustart sollte das CF0 auch von der Workbench-Disk aus verfügbar sein. Bei eingesteckter CF-Karte sollte automatisch ein Laufwerk-Icon auf dem Desktop erscheinen.

ADF auf Floppy schreiben

Dazu das Programm `adf2disk` und die zu schreibenden `adf`-Dateien auf die CF-Karte kopieren, diese in den PCMCIA-Adapter und diesen in den A600 stecken. Mit vorbereiteter CF und Workbench-Bootdisk booten. Doppelklick auf das CF-Karten-Icon. Alle Dateien anzeigen lassen! Bootdisk entnehmen und zu beschreibende Floppy einlegen. Doppelklick auf `adf2disk`. In der Eingabezeile einen Leerschritt eingeben und dann den zu schreibenden `adf`-Dateinamen eingeben (der auch auf der CF-Karte sein sollte).

Schon wird das Image auf die Diskette geschrieben.

Abb. 137:
So zeigt
man bei der
Workbench
alle Dateien
an.



Abb. 138:
Kommando-
zeileneingabe



PS/2-Maus am Amiga (und Atari ST)

Tom Kirk hat folgende Adapterschaltung entworfen, um Standard-PS/2-Mäuse am Amiga oder Atari ST anschließen zu können. Die Schaltung besteht nur aus einem PIC 16F84 und wenigen weiteren Bauteilen. Ein Aufbau auf Lochraster sollte also problemlos möglich sein.

Abb. 139:
Schaltplan

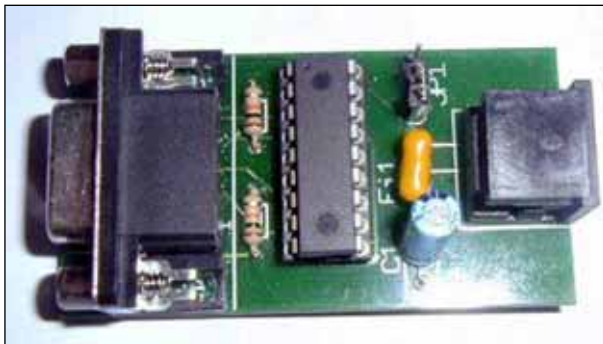
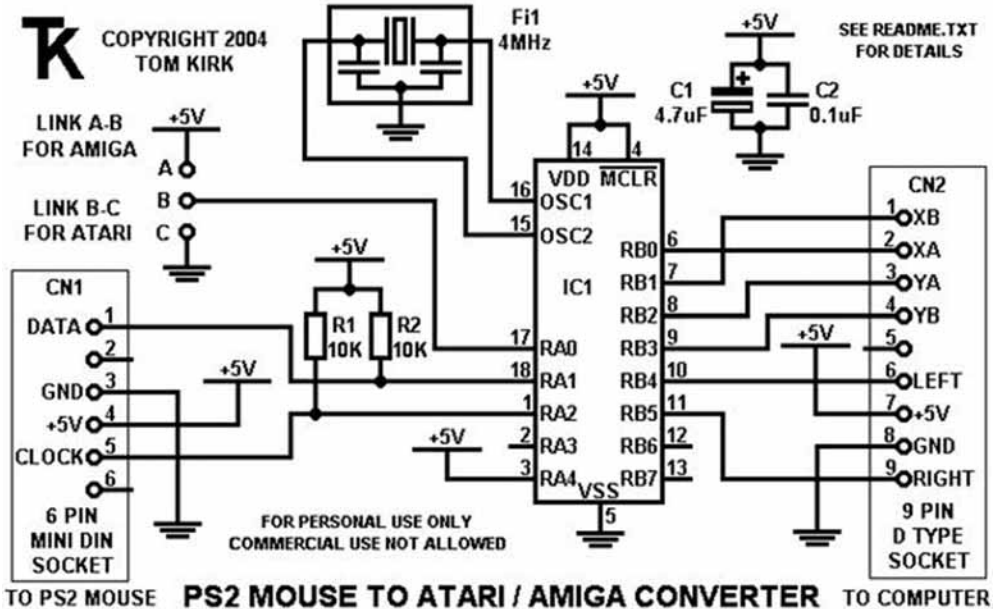


Abb. 140:
Platinen-
aufbau des
Adapters

Mehr Informationen und die Firmware des PIC 16F84 gibt es auf Toms Homepage: <http://members.aol.com/tgkirk/ps2mous.htm>

PS/2-Tastaturadapter – nicht nur für A500

Link: <http://www.rotgradpsi.de/mc/gadget/amiga.html>

Auf der Seite von Martin Clausen fand ich die Informationen, um mittels eines AT89C2051-Chips einen PS/2-zu-Amiga-Tastaturadapter zu bauen. Martin war auch so freundlich, mir programmierte Chips und Quarze zur Verfügung zu stellen. Leider fand ich erst nach der dritten Auflage dieses Buches die Zeit, den Adapter einmal auf Lochraster nachzubauen. Das Ergebnis: Der Adapter arbeitet prima!

Abb. 141:
Der
Lochraster-
aufbau

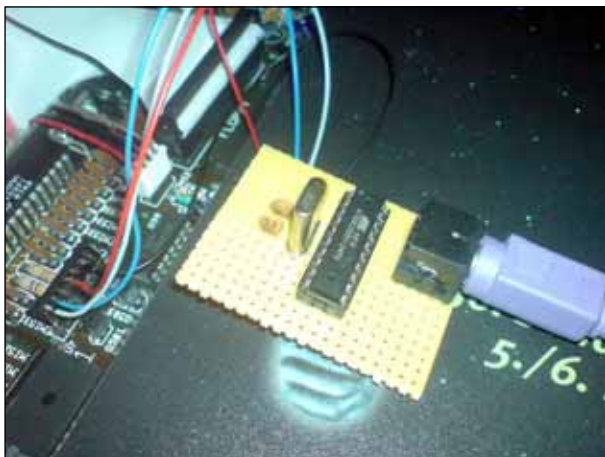
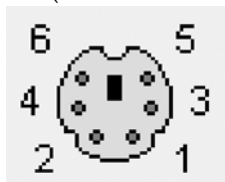


Abb. 142:
Tastatur-
schnittstelle



PS/2-Buchse (1=DATA; 3=GND; 4=+5V; 5=CLK):

Abb. 143:
PS/2-Buchse



Schaltplan

+5V an Pin20 und GND and Pin 10 des 89C2051 nicht vergessen. Zw. +5V und GND noch einen 100nf Kondensator schalten. C3 ist 1-2,2uF. C1+C2 sind 33pf.

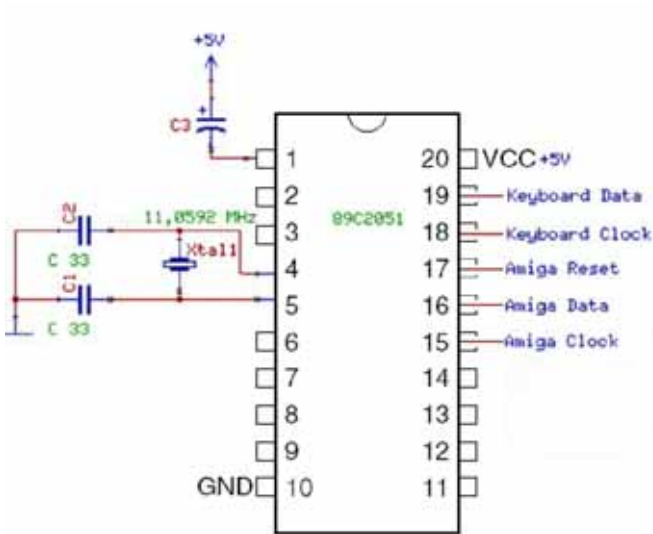


Abb. 144:
Schaltplan
des Adapters

PC-Floppy für Amiga umbauen

Von Stefan Pitsch, ^TePe^ und Stefan Hohendorf.

Unterschiede zwischen PC- und Amiga-Laufwerken:

1. Das DISK_CHANGE-Signal von Pin 34 entfernen
2. Das DISK_CHANGE-Signal auf Pin 2 legen
3. Das READY-Signal auf Pin 34 legen
4. Disk-ID von DS1 auf DS0 ändern (meist über Jumper/Lötbrücken)

Man benötigt also maximal zwei Kabel und eine Lötbrücke.

DISK_CHANGE ist das Signal des Diskwechsels, gerne auch /DC oder DCH abgekürzt.

READY wird gerne auch als /RY, RDY o.ä. gekennzeichnet.

DS1 und DS0 werden gerne auch mal nur S1 bzw. S0 genannt.

Das Problem ist nun, bei jeder Platine herauszufinden, wo diese Signale liegen.

Der Amiga braucht das READY-Signal eigentlich nicht, allerdings gibt es einige Programme (zum Beispiel XCOPY), welche es auslesen und ohne das Signal den weiteren Betrieb verweigern, weil sie meinen, es läge gar keine Diskette im Laufwerk.

Floppy-Typen, die erfolgreich umgebaut werden können (siehe auch www.pitsch.de/stuff/amiga/floppy.htm):

ALPS DF354H127F (oder DF354H090F)

ALPS(IBM) DF354H021E (no Disk Change)

Chinon FB 357

Chinon FZ357

Epson SMD 300

Epson SMD-300

Mitsubishi MF355-3250MG

Mitsubishi MF355F-3250MG

Mitsubishi MF355F-3250MG (anderes Board)

Mitsumi (Newtronics) D359T5 und D359T6

Mitsumi/NEWTRONICS D359T3
Mitsumi 5K0800G und 3J122JA
Mitsumi D359M3 (Typ 1)
Mitsumi D359M3 (Typ 2)
Mitsumi D359M3 / D359T3
Mitsumi D359M3D
Mitsumi D359T2
Mitsumi D359T3 (Typ 1)
Mitsumi D359T3 (Typ 2)
Mitsumi D359T5 (Typ 1)
Mitsumi D359T5 (Typ 2)
Mitsumi D359T6 (Typ 1)
Mitsumi D359T6 (Typ 2)
NEC FD1231H
NEC FD1231T / FD1231H
Panasonic JU-257A427P
Panasonic JU-257A604P und JU-257A656P
Samsung SFD-321B
Samsung SFD-321B
Samsung SFD-321B EF
Samsung SFD-321B KFUF
Sony MPF 920 E
Sony MPF 920 V
Sony MPF 920-E
Sony MPF 920-E
Teac FD-235HF 218U, 4430, 4291, 5291-U, 6291 und 7291-U5
Teac FD235HF 3240-U5
Teac FD235HF-217-U
Teac FD235HF-3240-U5
Teac FD235HF-4291-U
Teac FD235HF-5240
Teac FD235HF-6291
Teac FD235HF-7291
Y-E Data 702D
Y-E Data Typ 1
Y-E Data Typ 2

Beispiel: NEC FD1231H

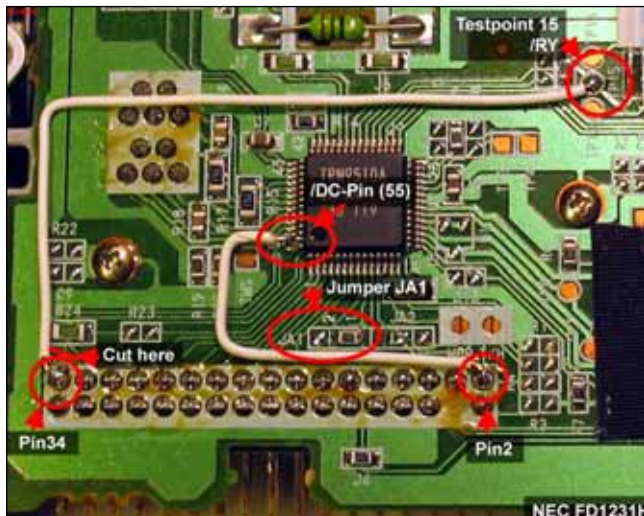
Hier gibt es wie so oft mehrere Boardvarianten. Meines sah zum Beispiel aus wie ein FD1231T und passte daher nicht zu der Anleitung aus dem a1k.org-Forum.

Der Aufbau:

1. Zunächst verlegt man am Jumper JA1 (mittig über der Steckerleiste) den 0-Ohm-Widerstand von der Stellung 2 nach rechts auf die Stellung 1.
2. Dann muss man oberhalb des Pins 34 die Leiterbahn durchtrennen (auskratzen).
3. Jetzt wird eine Verbindung zwischen dem TP15 und Pin 34 geschaffen – das ist /RY.
4. Nun muss man noch von der Leiterbahn, die vorher an Pin 34 angelegt war, eine Verbindung zu Pin 2 schaffen. Ich habe hierzu direkt an den dazugehörigen IC-Pin (Pin 55) angelötet.

HINWEIS: Ein so umgebautes Laufwerk funktioniert nicht ganz korrekt. Solange es das einzige Laufwerk am Rechner ist, ist alles in Ordnung, aber sobald man eine zweite Floppy anschließt, kommt es zu Problemen – meistens lässt sich die zweite Floppy gar nicht ansprechen. Das liegt daran, dass es sich beim /RY-Signal nicht um das echte /RY-Signal handelt. Im Zweifelsfalle sollte man also zu einem anderen Umbau greifen (zum Beispiel Mitsumi oder TEAC).

Abb. 145:
NEC
FD1231H



Beispiel: Mitsumi D359T5

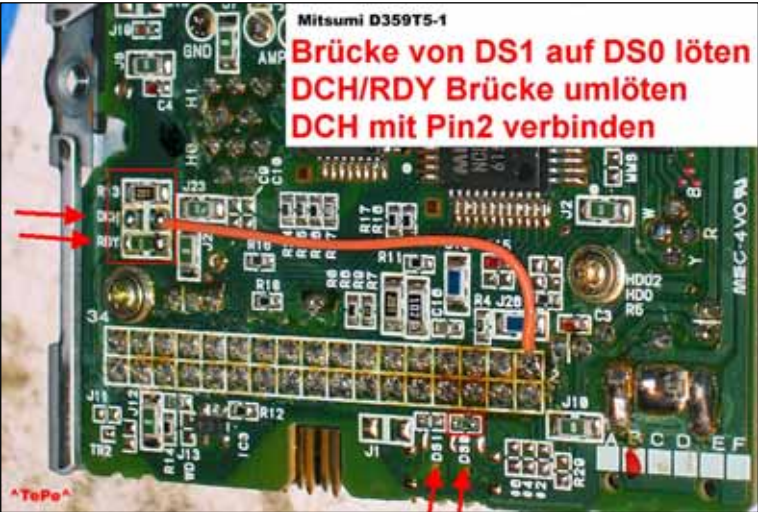


Abb. 146:
Mitsumi
D359T5-1

Etwas andere Variante:

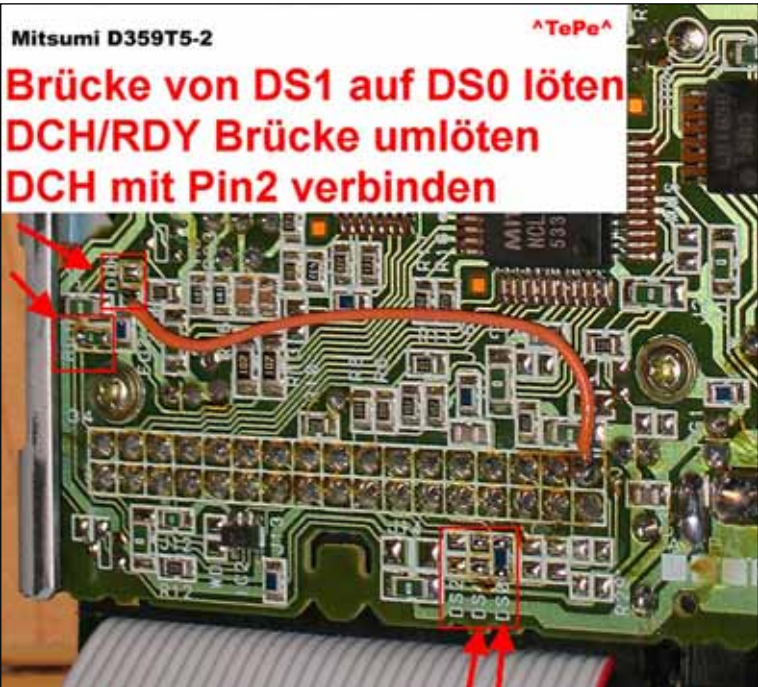


Abb. 147:
Mitsumi
D359T5-2

Beispiel: Samsung SFD-312B

Das Öffnen dieses Laufwerks gestaltet sich etwas schwierig:

- Frontblende abnehmen
- unten die vier Schrauben lösen, die das Gehäuseblech halten
- die eine Schraube lösen, die das obere Gehäuseblech hält
- dann zuerst das obere, danach das untere Gehäuseblech nach vorn wegziehen

Abb. 148:
Samsung
SFD-312B



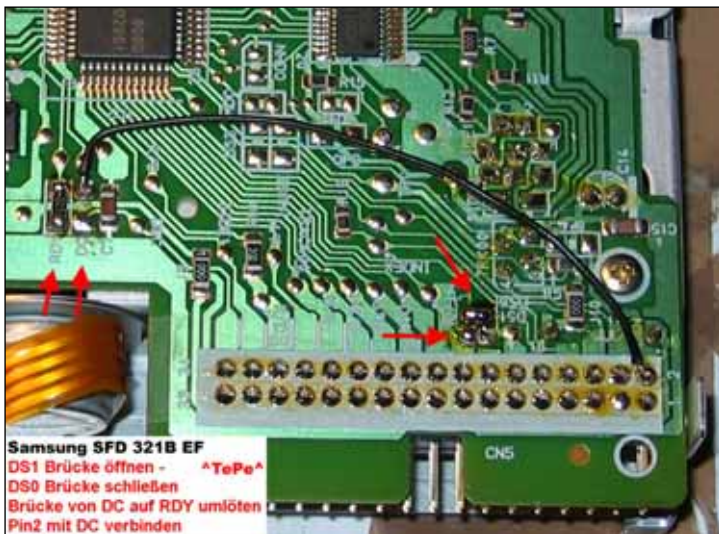


Abb. 149:
 Samsung
 SFD-321B EF

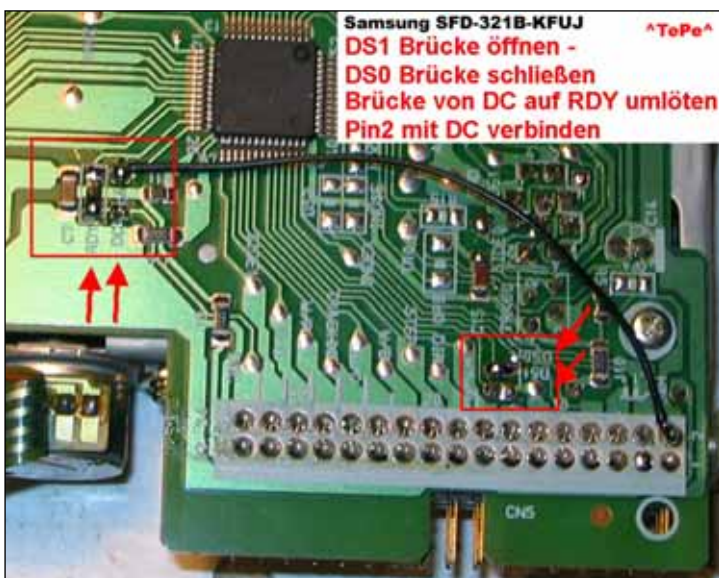


Abb. 150:
 Samsung
 SFD-321B
 KFUF

<http://www.torlus.com/floppy/>

<http://www.sensi.org/~tnt23/megadrive/>

Diese Floppy-Emulatoren-Projekte haben aber aus meiner Sicht noch keinen Stand zur einfachen und sicheren Nachbaubarkeit erreicht. Aber was (noch) nicht ist, kann ja noch werden.

**L
i
n
k
s**

Umbau A500+ auf 2 MB Chip-RAM

Nach jopower, basierend auf einer Anleitung von The_Killer

Benötigte Bauteile

- 8 RAMs 4*256K 120nS oder schneller (am besten der gleiche Typ wie bereits im Amiga vorhanden)
- isolierte Litze

Der A500+ ist bereits für 2MB Chip-RAM vorbereitet, was den Umbau sehr vereinfacht.

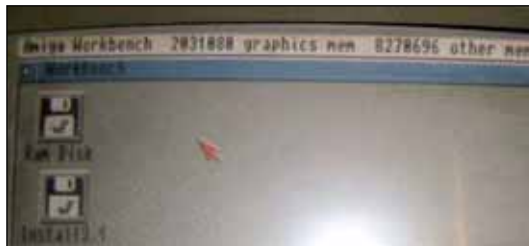
Bei den acht RAMs wird Pin 17 nach oben gebogen. Nun werden die RAMs huckepack auf die schon vorhandenen gesteckt und alle Pins (bis auf Pin 17) miteinander verlötet. Die aufgelöteten RAMs bekommen die gleiche Bezeichnung wie die RAMs darunter, nur mit einem »`« am Ende.

Alle Pins 17 von U16`, U17`, U20` und U21` werden miteinander und mit Pin 5 von U32 verbunden. Alle Pins 17 von U18`, U19`, U22` und U23` werden miteinander und mit Pin 11 von U32 verbunden. Damit ist der Umbau beendet, und der Amiga verfügt nun über 2 MB Chip-RAM.

Abb. 151:
2 MB Chip-
RAM beim
A500+



Abb. 152:
Die neue
RAM-Mel-
dung



Amiga 500 intern auf 1 MB RAM aufrüsten

Board-Rev. 6A

- JP2 von Mitte unten nach Mitte oben ändern
- JP4 auftrennen
- Fat Agnus 8371 gegen Fat Agnus 8372A austauschen (U2)
- Freie RAM-Plätze U21-U23 mit gleichen RAM-Typen nachbestücken (sockeln)

Achtung: Eine RAM-Erweiterung über die Steckkarte A501 ist dann nicht mehr möglich.



Abb. 153:
A500 intern
aufgerüstet

Die Rev. 5 und 3 lassen sich prinzipiell auch aufrüsten, das Verfahren ist aber komplizierter, weshalb ich hier nur die Erweiterung über eine 512K-RAM-Erweiterung im Erweiterungsschacht (sog. Trapdoor) empfehlen möchte.

A600-FastRAM-Erweiterung 4/8 MB

Autor: Vadim Akimov

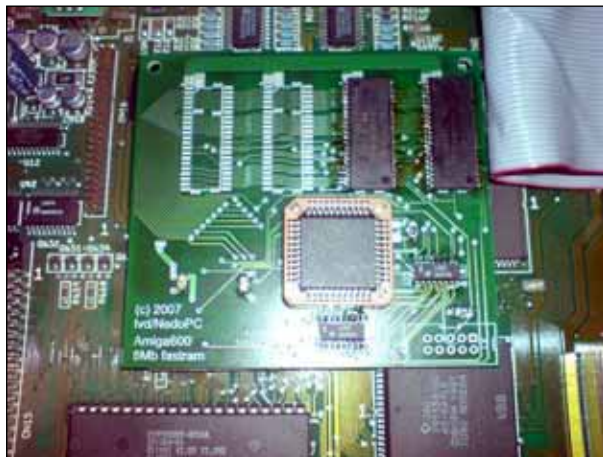
Homepage: http://lvd.nedopc.com/Projects/a600_8mb/index.html

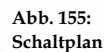
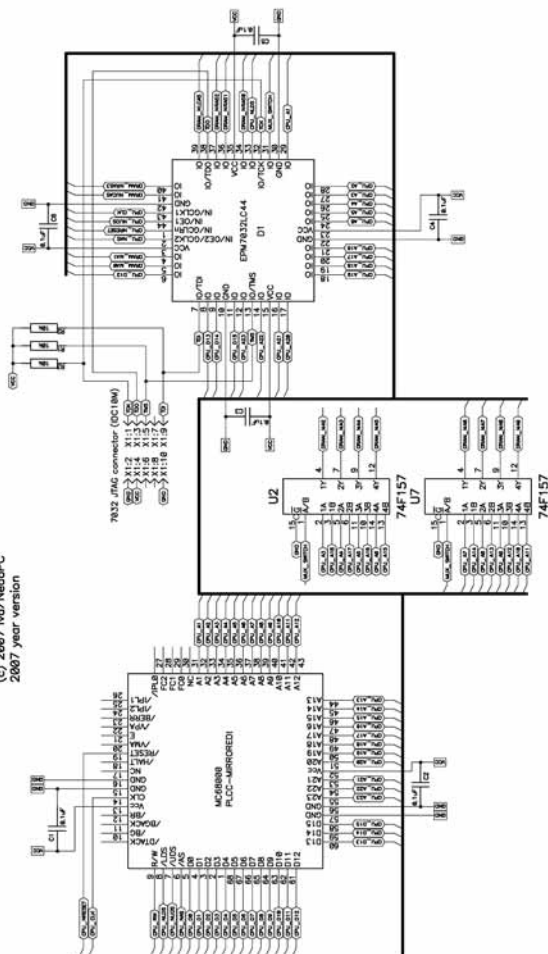
Hier wird ein Nachbauprojekt vorgestellt, das 4 oder 8MB (je nach CPLD-Konfiguration) FastRAM zusätzlich für den Amiga 600 zur Verfügung stellt. Bei der 4-MB-Version kann die PCMCIA-Funktionalität parallel genutzt werden. Bei der 4-MB-Version werden nur die beiden rechten RAM-Chips benötigt. Beim Auf-/Einbau zu beachten: 68-pol-PLCC-Sockel mit mindestens 2-3mm Abstand zur Platine auflöten. Quarzoszillator und Quarz auf dem A600-Board mit Isolierband o. Ä. isolieren!

Im a1k.org wurde in 2010 eine Sammelbestellung organisiert, um dieses Projekt nachzubauen. Board-Layout ist in P-CAD 2001. Platinen können zum Beispiel bei Conrad angefertigt werden. Die zentrale Logik ist in einem CPLD EPM7032SLC44-10 oder ATF1502AS10JC44 untergebracht. RAM-Chips sind 1 Megaword 16-BIT 42-Pins SOJ Dram's mit ≤ 60 -70ns.

Beispiel: NN5118160AJ-60, GM71C18160AJ7, TMS418160***-60 etc. Des weiteren werden 2x 74F157 SMD-IC, 3x 10k 1206 SMD-Widerstand und 2 + 3x je RAM 100nF 1206 SMD Abblockkondensatoren benötigt.

Abb. 154:
A600 auf-
gerüstet





Verbesserungsvorschläge (Dank an botfixer aus dem a1k-Forum): Hier sieht man eine 8-MB-Platine. Diese ist so zusammengebaut, wie es in der originalen Dokumentation beschrieben ist. Die Elkos sind auf beiden Platinenseiten verlötet und machen es erforderlich, den Platinenlack anzukratzen, um dort anzulöten. Das ist aber völlig unnötig, weil man die Elkos (die hier gezeigten Subminiaturelkos) oben montieren kann. Das PCB hat dazu sogar mehrere gut positionierte VIAs, wo man die Elko-Drähte einstecken und verlöten kann. Des weiteren ist für eine betriebs-sichere Funktion des JTAG-Connectors ein Pull-Down-Widerstand notwendig. Das PCB hat einen Fehler, so dass man diesen Pull-Down-Widerstand mit einem Stück Draht nach Masse verbinden muss, so wie auf dem Bild zu sehen:

Abb. 156:
Behebung
des Fehlers



Ich würde für die Elkos aber folgendes vorschlagen (Minuspole rechts):



Abb. 157:
Variante

Das ist eine weitere funktionsfähige Platine (4 MB), wo ich ein Elko als Beispiel eingesetzt habe.

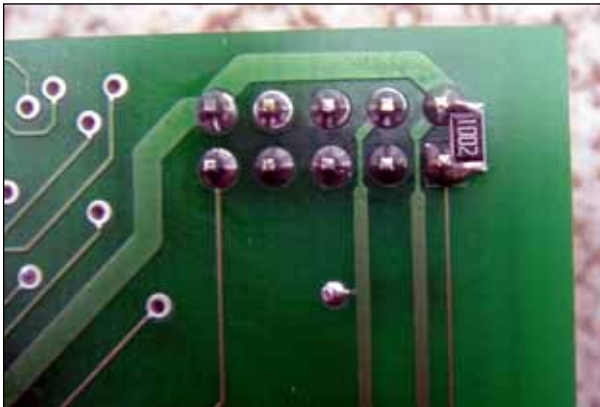


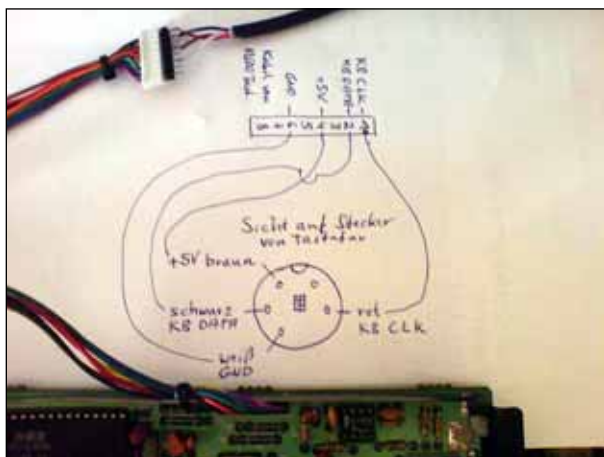
Abb. 158:
Weitere
Variante

Das hier ist die Rückseite des JTAG-Connectors. Wenn man den Widerstand so dort einlötet, hat er die gleiche Funktion und Verbindung, wie wenn man das Ganze wie auf der anderen Karte quer mit einem Kabel verdrahten muss. Ich finde, die beiden Verbesserungen sehen viel besser aus und lassen sich auch viel leichter umsetzen.

Amiga-500-Tastatur am CDTV

Als Stecker habe ich einen Din-6-Stecker mit Kabel von einer defekten PS/2-Tastatur genommen. Einen Pin (unten rechts – Draufsicht Stecker) abkneifen und auch der Mittel-Plastikpin ist zu dick, muss also auch entfernt werden. In die Buchse am A500-Tastaturkabel dann eine 8-pol-Lötpfostenreihe eingesteckt und über vier Kabel mit dem umgebauten PS/2-Stecker verbunden (+5V, GND, KB CLK und KB Data).

Abb. 159:
Schaltplan-
skizze



Floppy-Emulatoren

Links zu entsprechenden Projekten

<http://www.torlus.com/floppy/>

http://jeanfrancoisdelnero.free.fr/floppy_drive_emulator/index.html

<http://www.thesvd.com/SVD/>

<http://atariamiga.free.fr/sdiskemul.php>

<http://www.sensi.org/~tnt23/megadrive/>

Bei vielen Projekten ist leider keine allzu große Aktivität festzustellen.

Einzig das HxC-USB-Projekt hatte alle Voraussetzungen:

- Platinenlayout als Gerber/Eagle verfügbar
- Firmware verfügbar
- Schaltplan verfügbar
- Software verfügbar
- Getestet (20 Einheiten wurden gefertigt)
- Aktiv!

Nach langem Hin-und-Herüberlegen habe ich mich entschlossen, im forum64, a1k und CPC-forum nach Interessenten Ausschau zu halten. Letztlich ist eine Bestellung über 20 Platinen (pcbcart.com) entstanden.

Über das a1k-Forum meldete sich auch noch

jemand, der noch einen Restposten passender CP-LDs hatte und uns diese sogar programmiert zu einem guten Preis zur Verfügung stellen konnte.

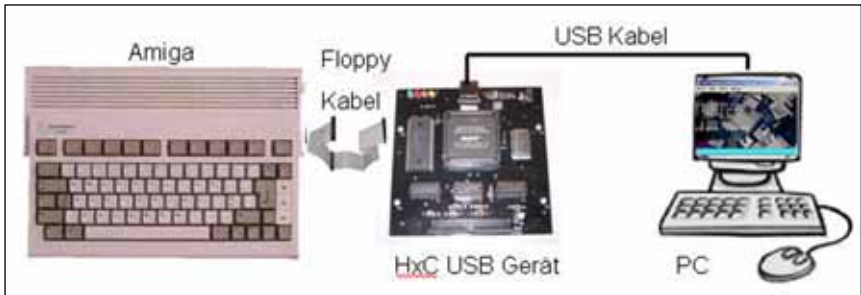
Insgesamt ist mit zirka 50 € pro Stück zu rechnen.

Abb. 160:
Floppy-Emu-
Platine



Und so wird das Gerät zwischen Amiga und PC/Notebook verkabelt:

Abb. 161:
Verbindung
Amiga/Emu-
lator/PC



Bedeutung der LEDs

D1 – Power on (Strom kommt vom USB-Anschluss)

D2 – Floppy-Aktivität (Wenn der Zielcomputer auf die Floppy zugreift)

D3 – Step (Wenn der virtuelle Lesekopf bewegt wird)

D4 – USB-Link

Noch ein paar Tipps zu Aufbau und Test

D5 kann weggelassen werden

L3 mit Drahtbrücke überbrücken

R6 ist 0 Ohm oder Drahtbrücke

Zuerst messen, dass zwischen 5V und Masse kein Kurzschluss besteht. Dann das Gerät über USB an den PC anschließen (jetzt muss der FT245-Treiber installiert werden). Dann die HxC-PC-Software starten. Es sollte die Meldung kommen, dass das Interface gefunden wurde! Unter dem »USB Stats«-Menüpunkt sollte ein Durchsatz von ca. 32768 Byte/Sekunde zu sehen sein.

Dann das USB Kabel wieder trennen, die Verbindung zwischen Zielcomputer und Gerät über Floppykabel herstellen, USB-Kabel wieder einstecken und ein Floppyimage einfach per Drag and Drop auf das Fenster des PC Programms ziehen.

Ein wirklich schönes Stück Hardware.

Elkos an SMD-Amigas tauschen

Von Sascha van Wahnem (AmigaGTI)

Elektrolytkondensatoren (Elkos) altern! Dabei können sie auslaufen und durch ihren ätzenden Inhalt auch ihr Umfeld zerstören! Aus diesem Grund sollten Elkos von Zeit zu Zeit einer optischen Kontrolle unterzogen und ggf. gegen frische ausgetauscht werden.



Abb. 162:
Ein ausge-
laufener
Elko

Ersatzteilliste für Elkos im Amiga 600

SMD-Elkos

Bezeichnung	Kapazität	Spannung	Best. Reichelt	Best. Kessler
C821	47 µF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C822	47 µF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C235	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C236	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C239	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C214	10 µF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C460	10 µF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C???	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C459	22 µF	35 V	SMD ELKO 22/35	ES 22/35-SMD
C324	22 µF	35 V	SMD ELKO 22/35	ES 22/35-SMD
C306	10 µF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C612	10 µF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C304	22 µF	35 V	SMD ELKO 22/35	ES 22/35-SMD
C303	22 µF	35 V	SMD ELKO 22/35	ES 22/35-SMD

Becher-Elkos

Bezeichnung	Kapazität	Spannung	Best. Reichelt	Best. Kessler
C307	470 µF	16 V	RAD 470/16	EST 470/16
C237	470 µF	16 V	RAD 470/16	EST 470/16
C408	1000 µF	10 V	RAD 1000/10	EST 1000/10
C811	1000 µF	10 V	RAD 1000/10	EST 1000/10

Ersatzteilliste für Elkos im Amiga 1200**SMD-Elkos**

Bezeichnung	Kapazität	Spannung	Best. Reichelt	Best. Kessler
C822	47 µF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C306	10 µF	35 V	SMD ELKO 10/35	ES 10/35-SMD
C303	22 µF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C304	22 µF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C???	22 µF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C324	22 µF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C???	10 µF	35 V	SMD ELKO 10/35	ES 10/35-SMD
C821	47 µF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C235	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C459	22 µF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C214	10 µF	35 V	SMD ELKO 10/35	ES 10/35-SMD
C239	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C236	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C409	100 µF	16 V	SMD ELKO 100/16	ES 100/16-SMD

Becher-Elkos

Bezeichnung	Kapazität	Spannung	Best. Reichelt	Best. Kessler
C811	1000 µF	10 V	RAD 1000/10	EST 1000/10
C237	470 µF	16 V	RAD 470/16	EST 470/16
C408	1000 µF	10 V	RAD 1000/10	EST 1000/10
C307	470 µF	16 V	RAD 470/16	EST 470/16

Das gilt analog auch für alle anderen Elkos/SMD-Elkos in anderen Systemen (Amiga 4000 etc.).

Webseite Reichelt: www.reichelt.de

Webseite Kessler: www.kessler-electronic.de

AT/X-Netzteil für A 1000

AT-Netzteilsteckerbelegung

Pin	Farbe	Bedeutung
1	Orange	Power ok +5V
2, 10, 11, 12	Rot	+5V
3	Gelb	+12V
4	Blau	-12V
5, 6, 7, 8	Schwarz	GND/Masse
9	Weiß	-5V

ATX-Netzteilsteckerbelegung

Pin	Farbe	Bedeutung
1, 2, 11	Orange	+3, 3V
3, 5, 7, 13, 15, 16, 17	Schwarz	GND/Masse
4, 6, 19, 20	Rot	+5V
8	Grau	Power ok
9	Violett	+5V für Standby
10	Gelb	+12V
12	Blau	-12V
14	Grün	PS ON (gegen GND = Netzteil EIN)
18	Weiß	-5V

Achtung: Sich nicht auf diese Standardfarben verlassen! Aufschriften auf dem Netzteil kontrollieren und immer auch selbst nachmessen!

Man muss Pin 14 (PS ON) mit einem Masse-Pin verbinden, um das ATX-Netzteil einzuschalten. Ich habe ein ATX-Netzteil mit Ein-/Ausschalter verwendet und PS ON dauerhaft auf GND gelegt. ATX-Netzteile benötigen eine Grundlast (also A1000/PC Mainboard oder CD-Laufwerk).

A1000 Power Connector J14

Von vorn gesehen. Pin 7/Grau liegt dem Modulator am nächsten.

Pin	Farbe	Bedeutung
7	Grau	TICK-Signal (50 Hz)
6	Rot	+5V
5	Rot	+5V
4	Schwarz	GND/Masse
3	Schwarz	GND/Masse
2	Orange	+12V
1	Violett	-5V

Benötigt werden also vom PC-Netzteil +5 V, +12 V, -5 V und GND.

TICK-Signal

Der A1000/2000 benötigt zwingend ein TICK-Signal, um stabil arbeiten zu können! Dabei handelt es sich um ein 50-Hz-(USA 60-Hz-)Rechtecksignal.

Ein solches Signal liegt an VSync an. Dieses ist an Pin 12 des Amiga-Videoports oder an Pin 38 des 8367-AGNUS-Chips verfügbar. Dieses einfach mit dem TICK-Pin des Power-Anschlusses verbinden.

Ohne weitere Belastung des Signals ist das so ausreichend. Beim A500 ist das auch so gemacht.

TICK von Pin 12 (links außen, oben – weißes Kabel) des Videoports:

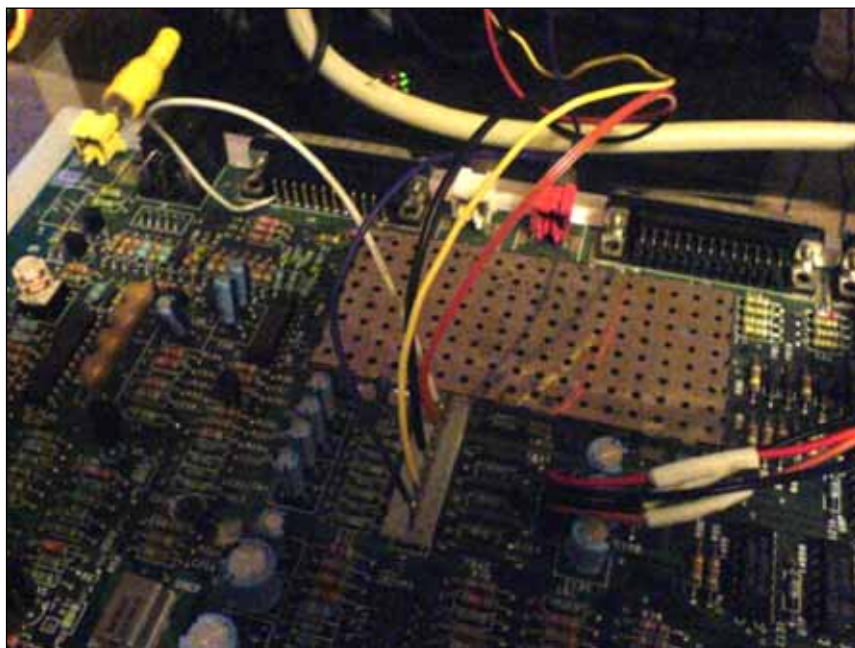


Abb. 163:
TICK vom
Videoport

Externe Floppy als DF0:

Um die externe Floppy als Bootfloppy (DF0:) nutzen zu können, muss ein 40-pol-Zwischensockel für CIA U6S angefertigt werden, wo SEL0 – Pin 13 und SEL1 – Pin 14 vertauscht werden (Kabel über Kreuz anschließen).

Wer möchte, kann dazu natürlich auch einen Umschalter entsprechend anschließen.

Für A500 ist das U8, für A2000 A ist es U11 und für den A2000 B ist es U301.

Abb. 164:
A1000
»Brett-
Edition«



Verkaufszahlen Commodore Deutschland

Gesamtzahl der in Deutschland verkauften Geräte; herausgegeben von der Marketing-Abteilung Commodore Deutschland:

System	Stückzahl (DE)
Amiga CD ³²	25.000
Amiga CDTV	25.800
Amiga 500	1.081.000
Amiga 500+	79.500
Amiga 600	193.000
Amiga 1000	27.500
Amiga 1200	95.500
Amiga 2000	124.500
Amiga 3000	8.300
Amiga 4000	11.300
C 128	284.300
C 16/116	286.500
C 64	3.050.000

Daraus ergeben sich die relative Seltenheit von A3000/4000 (dafür muss man heute sicher mehr als 200-300 € zahlen) und die doch noch gute und relativ preiswerte Verfügbarkeit der A500-Systeme auf dem Gebrauchtmarkt.

Amiga am TFT anschließen

Der Anschluss eines A500 über einen A520-TV-Modulator oder eines A600 über Video-Out und Scart-Adapter an einen Fernseher dürfte klar sein. Aber eventuell möchte man seinen Amiga auch an einem TFT-Monitor betreiben. Dazu sind eigentlich teure sog. Scandoubler und Flickerfixer erforderlich. Mit passendem TFT-Monitor, der schon bei ca. 50 Hz synchronisiert, und einem Video-zu-VGA-Konverter kann man dies aber unter Umständen und mit geringerer Bildqualität auch erheblich preiswerter erreichen.

Ich hatte mir dazu vor einigen Tagen den Video2VGA-Konverter von Tinxi bestellt (29,99 € + Versand = 36 €). Nach den bisherigen Erfahrungen bin ich von einer mittelmäßigen, für Spiele hoffentlich noch brauchbaren Bilddarstellung ausgegangen, aber 30 € sind ein guter Preis. Bestellen kann man dieses Gerät übrigens direkt unter www.tinxi.com, Artikelnr. 33003.

Nach dem Anschluss über Video-In muss ich sagen, ich bin doch positiv überrascht. Textverarbeitung sollte man damit nicht machen, aber für Spiele ist es eine durchaus gleichwertige Darstellung im Vergleich zum älteren Fernseher. Durch den VGA-In-Eingang kann auch beispielsweise mein Minimig DE1 oder jede andere VGA-Quelle parallel angeschlossen bleiben. Über einen 3,5mm-Stereo-Audio-Y-Stecker lassen sich auch Audio vom Amiga und Minimig/PC parallel oder ohne umzustöpseln betreiben.

Abb.165:
Video2VGA-
Konverter





Abb. 166:
Amiga 600
an einem
TFT-Monitor

C64 und DTV gehen leider so einfach nicht, da deren Signalqualität miserabel ist: Das Bild scrollt nach links weg und ist sehr unscharf.

Folgender Tipp kann für ein DTV für Abhilfe sorgen:

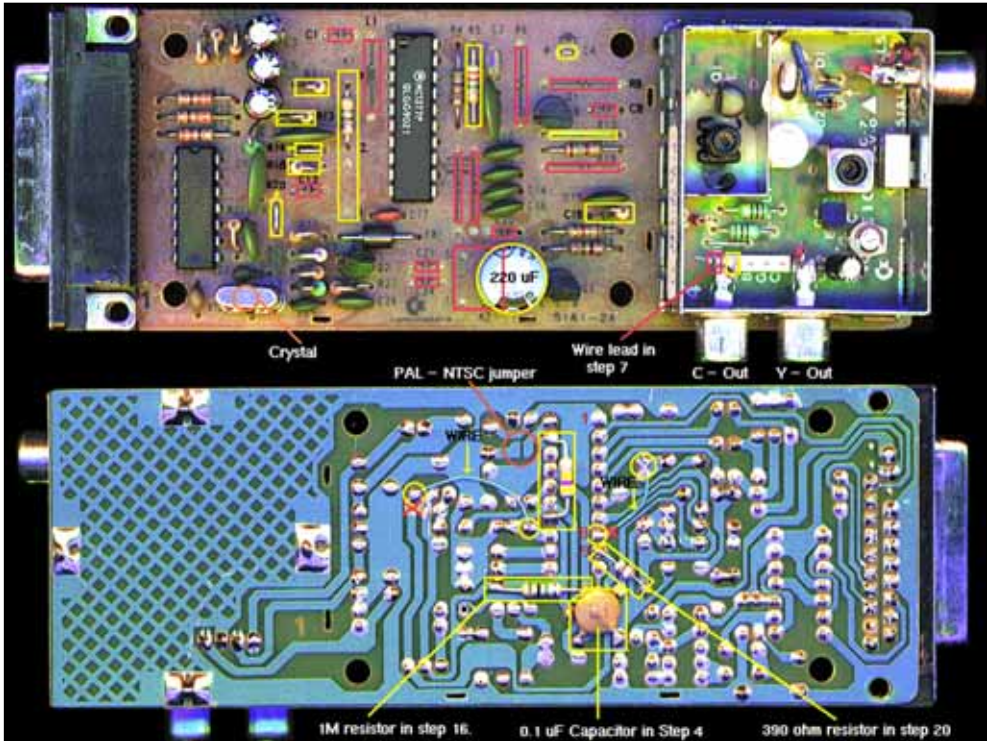
POKE 53311,1 : POKE 53326,2 setzt \$D04E/"Scan line timing adjust" auf 2 (0 ist der Standard).

TV-Modulator A520 auf S-Video umbauen

David Myers hat eine Umbauanleitung entwickelt, um einen A520-TV-Modulator so umzubauen, dass man ein S-Video-Signal erhält, das eine bessere Signalqualität als das FBAS-Signal hat.

Achtung: Als TV-Modulator ist der A520 nach dem Umbau nicht mehr zu gebrauchen! Das Bild wurde freundlicherweise von David selbst zur Verfügung gestellt.

Abb. 167:
S-Video-Mod
eines A520



Schritt-für-Schritt-Anleitung

- C1 und L1 entfernen (ist evtl. gar nicht bestückt)
- Spule K2 und C8, C13, C20, C21, C24, R6, R8, R15, R16, R19 entfernen
- K1 (delay line) entfernen und einen 470-Ohm-Widerstand zwischen Pin 2 und 3 einsetzen
- Einen 0,1- μ F-Kondensator zwischen Pin 10 vom IC und Masse anschließen (unter der Platine)

- C18 entfernen und durch einen 330-Ohm-Widerstand ersetzen
- Leiterbahn zwischen Basis (B) von Transistor Q1 (siehe rotes Kreuz auf Bild) durchtrennen
- Pin 13 vom IC mit der nun isolierten Basis (B) von Q1 verbinden (unter der Platine)
- Die Verbindung des Audio-Input-Cinch-Steckers entfernen (im Modulator – markiert mit dem roten Quadrat im Bild)
- Nun den Cinch-Stecker (Audio-Input) mit einem 100-180-Ohm-Widerstand (*) zum Punkt ›V‹ der weißen Anschlussreihe verbinden (siehe gelbes Quadrat)
- R14 durch eine Drahtbrücke ersetzen
- Leiterbahn zu Pin 8 vom IC durchtrennen (markiert mit Kreuz, Unterseite Platine)
- Pin 8 des ICs nun mit Pin 3 der ehemaligen K1 (delay line) mit einem Kabel verbinden (Pin 2 und 3 sind jetzt mit einem 470-Ohm-Widerstand überbrückt – siehe Schritt 3)
- R7 durch einen 180-Ohm-Widerstand ersetzen
- C4, R20, R13 entfernen und durch eine Drahtbrücke ersetzen
- R11 durch einen 1-KOhm-Widerstand ersetzen
- Einen 220-Ohm-Widerstand an der leeren R18-Position einsetzen
- Einen 1-MOhm-Widerstand von Pin 17 des ICs zu +12 Volt legen (Unterseite Platine)
- R5 entfernen (falls bestückt) und durch einen 2,2-MOhm-Widerstand ersetzen
- Einen 470-KOhm-Widerstand von Pin 14 zu Pin 19 des ICs löten (Unterseite Platine)
- Einen 220-µF-Kondensator zwischen Pin 3 und 4 der ehemaligen Spule K2 einsetzen (Pluspol an 4)
- Einen 390-Ohm-Widerstand von Pin 9 des ICs zu Pin 1 der ehemaligen K1 (delay line) legen (Unterseite Platine)
- Dann ein entsprechendes S-Video-Kabel auf 2x Cinch (Y/C) anfertigen. Fertig.

(*) In der originalen Anleitung ist hier nur eine Drahtverbindung vorgesehen. Dies kann aber unter bestimmten Umständen den dadurch ungeschützten Transistor Q1 zerstören (Ersatztyp zum Beispiel TIS90).

Minimig

Minimig (Kurzform für »Mini-Amiga«) ist eine Open-Source-Reimplementation eines Amiga 500 in FPGA (Field Programmable Gate Array). Minimig wurde von dem niederländischen Ingenieur Dennis van Weeren entworfen und gebaut. Auslöser waren nicht endende Diskussionen in der Amiga-Gemeinschaft, ob so etwas möglich wäre.

Die Entwicklungszeit ging von ca. Januar 2005 bis ca. Mitte 2007. Am 25. Juli 2007 wurden die Projektdaten (Sourcecode, Schaltplan) unter die GNU (General Public License), Version 3, gestellt und veröffentlicht.

Seitdem sind einige Minimigs in Eigenregie gebaut worden. Aber auch die Firma Acube Systems aus Italien lässt Minimigs bauen und vertreibt sie. In Deutschland werden diese durch Vesalia vertrieben.

Abb. 168:
Minimig



Minimig auf Altera-DE1-Board

Wie beschrieben ist der Minimig von Dennis van Weeren die Reimplementation eines Amiga 500 in FPGA. Von tobiflex (Tobias Gubener) wurde er mit einem 68000-Core erweitert und ist deshalb auf DE1-/2-Boards von Altera lauffähig. Es wird noch eine Kickstart-ROM-Datei mit Namen »kick.rom« und Größe 512 KB im Rootverzeichnis benötigt (Kick 1.3 = 256k; einfach mit `copy /b kick13.rom+kick13.rom kick.rom` zu einer 512KB-Datei zusammen kopieren.)

Im Core-Archiv ist die Datei »spihost.rom« enthalten. Diese muss ebenfalls auf die SD-Karte ins Rootverzeichnis. Dann noch ein paar ADF-Dateien drauf und schon kann's losgehen ...

Die Schiebeschalter SW 0-3 und SW9 müssen oben stehen.

SW[9] Monitorfrequenzumschaltung

SW[0] aus-an löst einen Komplett-Reset aus

SW[3..0] müssen an sein – damit lassen sich einzelne Teile des Cores resettet

SW[8..4] ungenutzt

Debugausgaben kommen über die RS232 mit 115000 Baud und 8N1.



Abb. 169:
DE1-Board
mit Monitor,
Tastatur und
Lautsprecher

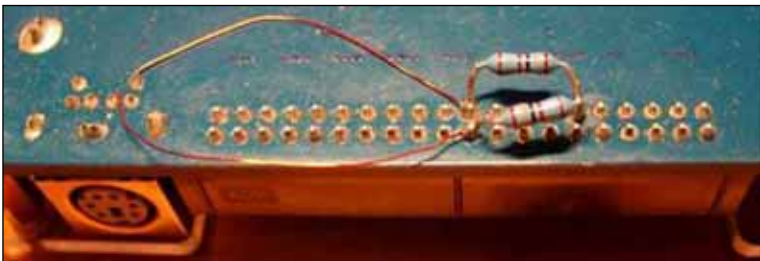
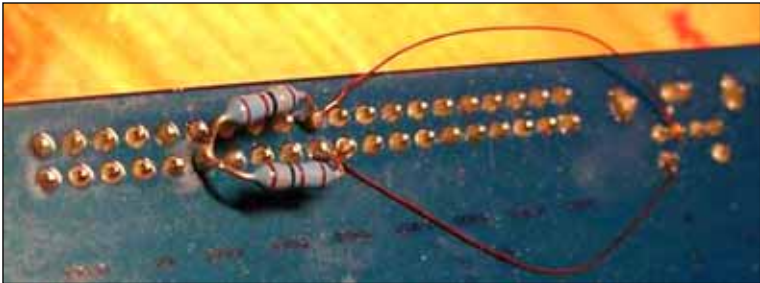
Die SOF-Datei dient dem einmaligen Konfigurieren des FPGA, nach dem Ausschalten ist sie wieder weg. Die Programmiersoftware muss dazu im Modus JTAG stehen und ein Häkchen bei PROGRAM/Configure muss beim geladenen SOF-File gesetzt sein.

Um die POF-Datei ins Start-Eeprom zu programmieren, muss der Schiebeschalter links oberhalb der LEDs in Richtung der LEDs geschoben werden. Der Programmiermodus muss dazu auf Active-Serial-Programmierung stehen. Der Vorgang dauert um Einiges länger als die normale Konfiguration. Haken bei PROGRAM/Configure nicht vergessen. Danach den Schiebeschalter wieder Richtung USB-Stecker.

Zum Laden der POF-/SOF-Datei benötigt man nicht die vollständige Quartus-II-Web-Edition (800 MB), sondern es gibt auch einen »Programmer Only« mit ca. 60 MB. Für letzteren benötigt man auch keine spezielle Lizenz, welche bei der Web-Edition beantragt werden muss (kostenfrei, aber mit zeitlich begrenzter Gültigkeit).

PS/2-Mauserweiterung an vorhandener PS/2-Tastaturbuchse (für Y-Kabel – Achtung: Anschluss umgekehrt zur Beschriftung! Tastatur an Mausbuchse; Maus an Tastaturbuchse):

Abb. 170
& 171:
2x 1K-
Widerstand



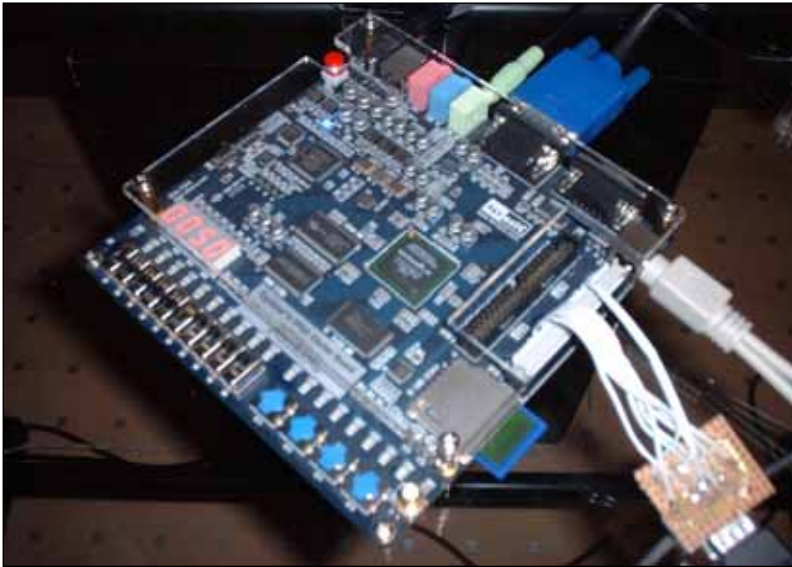
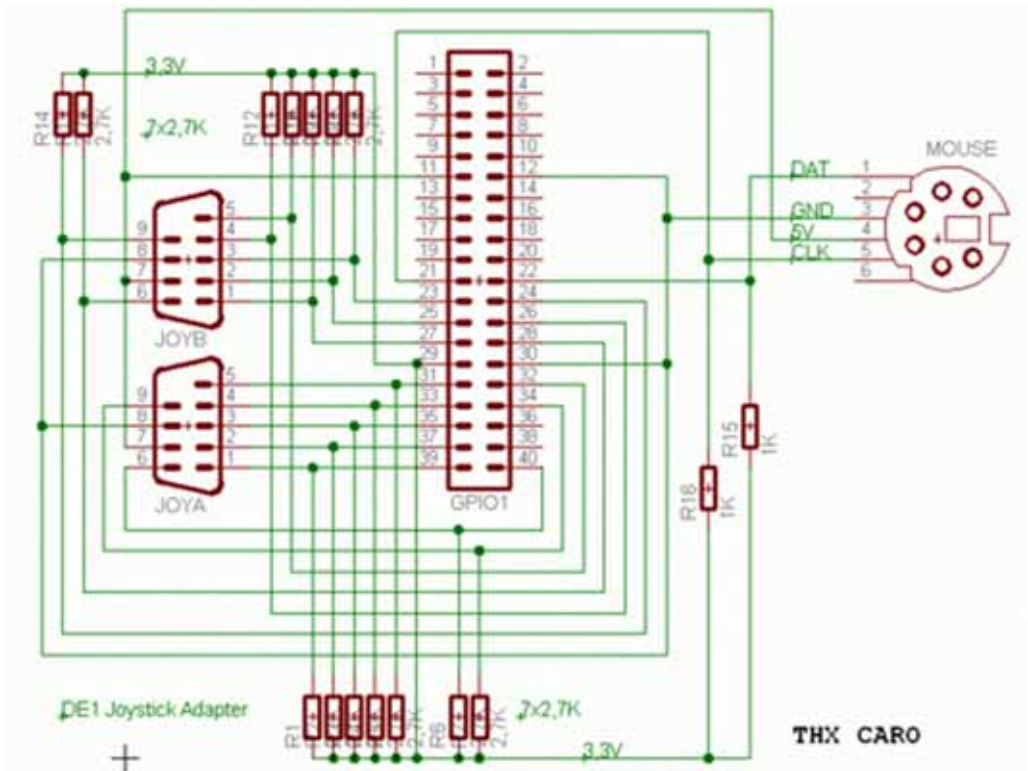


Abb. 172:
DE1 mit
Joystick-
adapter für
einen
Joystick

Abb. 173:
Schaltplan
für den
Joystick-
adapter



PS/2-Tastaturbelegung

F12 Minimig-Menü aufrufen/beenden
PgUp Menü hoch
PgDn Menü runter
Pos1 Auswahl

Micro KBD – Micro-Tastaturersatz für Minimig

Man nehme: Atmega8, 5 Taster, PS/2-Kabel und Stecker, Lochraster und Kabel, micro_kbd.hex und fertig ist der Miniatur-Tastaturersatz fürs Minimig ...

C0 = F12
C1 = PgUp
C2 = PgDn
C3 = Home/Pos1
C4 = F1
B0 = Clock-PS/2
D3 = Data-PS/2

GND und +5V zwischen PS/2 und AVR verbinden. Cn = PortC-Taster gegen GND

Sourcecode -> http://petersieg.webng.com/minimig/micro_kbd.zip

```
'-----  
'  
                                micro_kbd.BAS  
'                                (c) 2008 peter.siegl@gmx.de  
'                                PS2 AT Keyboard emulator for Minimig  
' NOTE THAT THE LIBRARY IS AN OPTIONAL ONE and by default is not included !  
'-----  
$regfile = "m8def.dat"  
$crystal = 1000000  
  
$lib "mcsbyteint.lbx"                                ' use optional  
lib since we use only bytes
```

```

Config Portc = Input                                ' port c for
taster
Portc = 255                                         ' pullups on

'configure PS2 AT pins
Enable Interrupts                                  ' you need to
turn on interrupts yourself since an INT is used
Config Atemu = Int1 , Data = Pind.3 , Clock = Pinb.0
'
'          ^----- used interrupt
'          ^----- pin connected to DATA
'          ^-- pin connected to clock
'Note that the DATA must be connected to the used interrupt pin

Waitms 500                                         ' optional delay
Do
    Waitms 150
    If Pinc.0 = 0 Then Sendscankbd F12              ' send a scan code
    If Pinc.1 = 0 Then Sendscankbd Pgdn            ' send a scan code
    If Pinc.2 = 0 Then Sendscankbd Pgup            ' send a scan code
    If Pinc.3 = 0 Then Sendscankbd Pos1            ' send a scan code
    If Pinc.4 = 0 Then Sendscankbd F1              ' send a scan code
    Waitms 150
Loop

F12:
Data 3 , &H07 , &HF0 , &H07

F1:
Data 3 , &H05 , &HF0 , &H05

Pgup:
Data 5 , &HE0 , &H7D , &HE0 , &HF0 , &H7D

Pgdn:
Data 5 , &HE0 , &H7A , &HE0 , &HF0 , &H7A

Pos1:
Data 5 , &HE0 , &H6C , &HE0 , &HF0 , &H6C

```

Abb. 174:
Das fertige
Micro-KBD

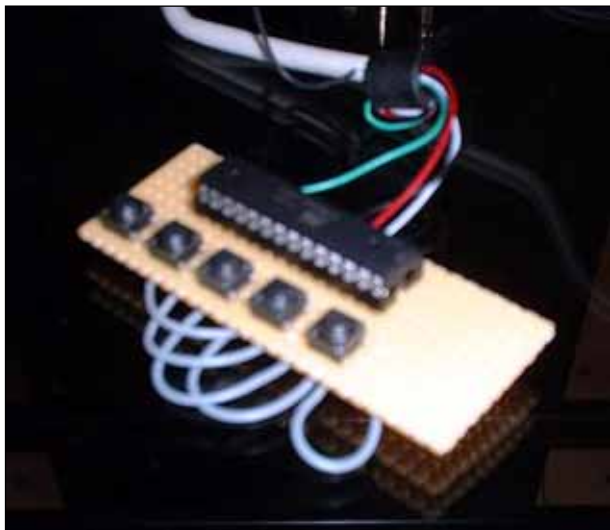


Abb. 175 &
176 (n. S.):
An einem
Benq FP71G
läuft der
Minimig/DE1
sehr gut.





Links

<http://home.hetnet.nl/~weeren001/http://gamesource.groups.yahoo.com/group/minimigt68/>

http://www.amiga.org/modules/newbb/viewtopic.php?topic_id=44173&forum=8&viewmode=flat&order=ASC&start=0

<http://home.freeuk.com/fpgaarcade/index.htm>

AVR-Applikationen

Einfachst-Programmer

Ich wollte mal den einfachsten AVR-Programmer nachbauen. Das Original-Schaltbild ist auf der PonyProg-Seite zu finden. Hatte ich nun gebaut, aber »Write failed« ...?

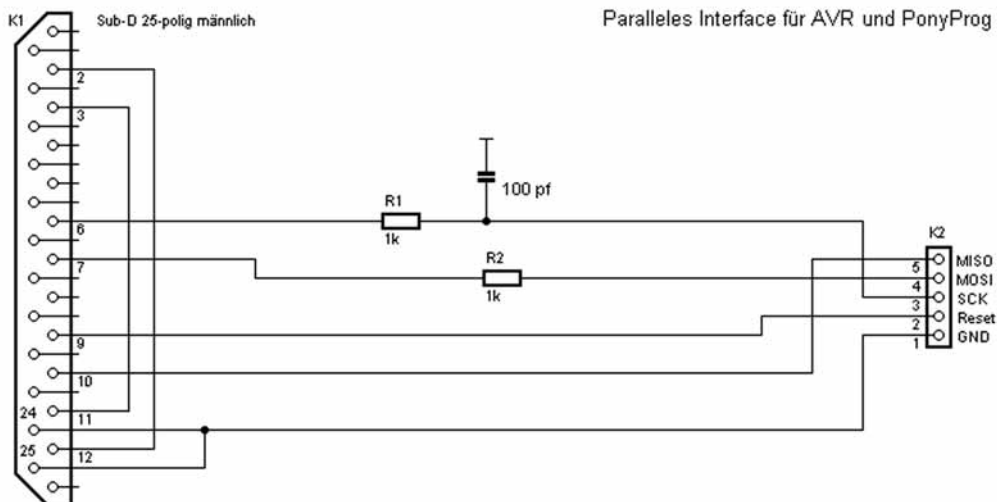
Habe dann gegoogelt und gute Hinweise gefunden. Die Leitungen sollten möglichst kurz sein! Gegen Leitungsschwingen sollte ein Abblockkondensator in die SCK-Leitung gegen GND rein. Printerport auf EPP/ECP, nicht auf SPP stellen. Habe ich gemacht und das ca. 80 cm lange Parallelport-Verlängerungskabel dazwischen raus genommen ... und siehe da, es funktionierte!

Wichtig ist, dass, sobald man auf externen Takt umstellt, danach den AVR nicht mehr hiermit programmieren kann, da ja hier kein externer Takt angeschlossen ist. Für mmc2iec ist das nicht wichtig, weil kein externer Takt verwendet wird.

Ansonsten kann man natürlich einen Quartz zwischen XTAL1 und XTAL2 plus 2x22pf gegen GND für einen externen Takt anschließen. Dieser Einfachst-Programmer soll STK200-kompatibel sein. Unter PonyProg sind Parallel-AVR ISP I/O und der richtige LPT Port einzustellen. Auch hier die Kalibrierung nicht vergessen!

Abb. 177:
Modifizierter
Schaltplan
für AVR

Geänderter Schaltplan



Lochrasteraufbau

Man sieht die beiden Kabelbrücken zwischen 2/12 und 3/11 direkt am Stecker. Auf der Unterseite sieht man den Abblockkondensator (hier nur 22 pf). Das helle Kabel zwischen Stecker und Platine ist GND.

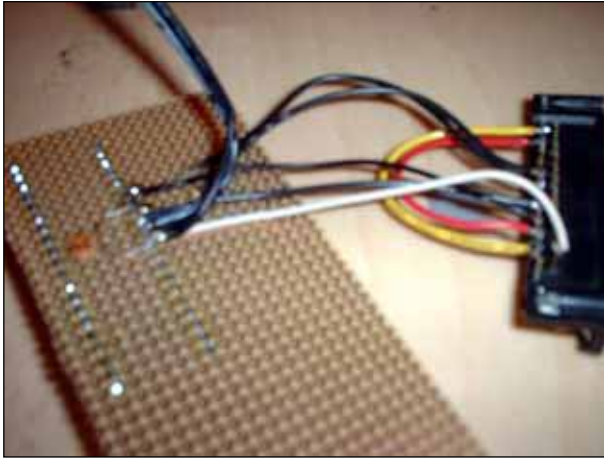
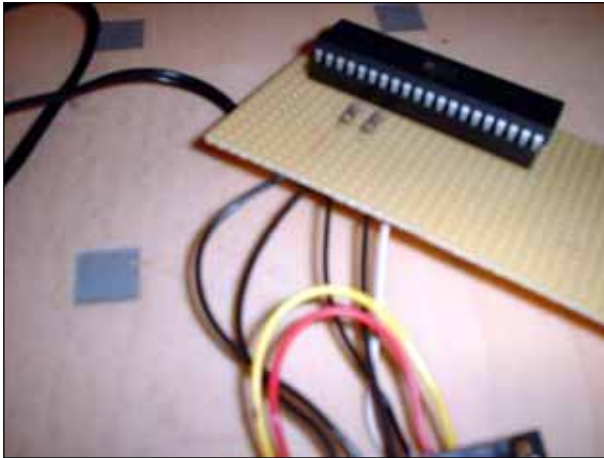


Abb. 178
& 179:
Lochraster-
aufbau von
unten und
oben.
Von oben
sind nur die
beiden 1K-
Widerstände
zu sehen.

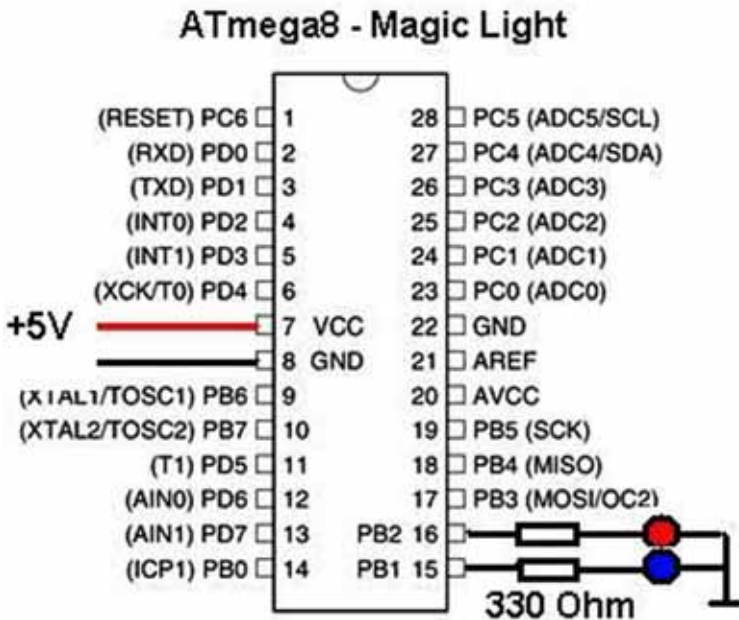


Magic Light

Simple Hello-World-Schaltung mit ATmega8. Einfache PWM-Bascom-Anwendung. Ideal als erste Nachbauschialtung. Lässt zwei LEDs abwechselnd heller und dunkler werden (Fading mittels PWM).

Um gute »Mischfarben« zu erhalten, am besten eine Mehrfarb-LED nehmen. Alternativ zwei einfache, möglichst helle LEDs. LEDs mit Schmirgelpapier anrauen, um die Lichtstreuung zu verstärken etc.

Abb. 180:
Schaltplan
für *Magic
Light*



Link

http://petersieg.webng.com/avr/avr_pwm.zip

Quellcode

```
$regfile = "m8def.dat"
$crystal = 8000000

Config Portb.1 = Output
Config Portb.2 = Output

Dim N As Byte

Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear
Up , Compare B Pwm = Clear Up , Prescale = 1

Do
  For N = 0 To 255
    Compare1a = N
    Compare1b = 255 - N
    Waitms 150
  Next N
  Waitms 3000
  For N = 255 To 0 Step -1
    Compare1a = N
    Compare1b = 255 - N
    Waitms 150
  Next N
  Waitms 3000
Loop

End
```

Tiny BASIC im ATmega8 auf Pollin Eval. Board

Basierend auf BSS02. Ohne TV-Ausgabe. Kommunikation über serielle Schnittstelle mit 9600 8N1. Läuft so direkt auf einem Pollin Eval. Board mit ATmega8, 16-MHz-Quartz. Quellcode, HEX Datei, BSS02 orig. Seite sind im ZIP-Archiv (Link unten).

Von der Firma Pollin (www.pollin.de) gibt es dieses preiswerte AVR-Entwicklungsboard für ca. 15 € als Bausatz. Zur Programmierung kann PonyProg verwendet werden.

Das Schöne an diesem Tiny BASIC ist, dass es auf einen Port von Li-Chen Wangs TinyBasic aus dem Jahr 1976 (Veröffentlicht im Doctor Dobb's Journal) beruht! Von Jeff Hunsinger wurde es in die AVR-Welt portiert. Ich habe nur ein paar kleinere Anpassungen für den ATmega8 gemacht. Außer 16-MHz-Quartzbeschaltung werden nur RX und TX be-

Abb. 181:
Pollin Eval.
Board mit
Tiny BASIC

Link: http://petersieg.webng.com/avr/avr_tinybasic.zip



AVR Nova Strike (Galaxian)

<http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2004/kfm9/index.htm>

Ein Space Invaders-/Galaxian-Styl-Spiel mit CodeVision C Sourcecode!
Zip-Archiv inkl. HEX-Datei (NTSC!):
<http://petersieg.webng.com/avr/NovaStrike.zip>

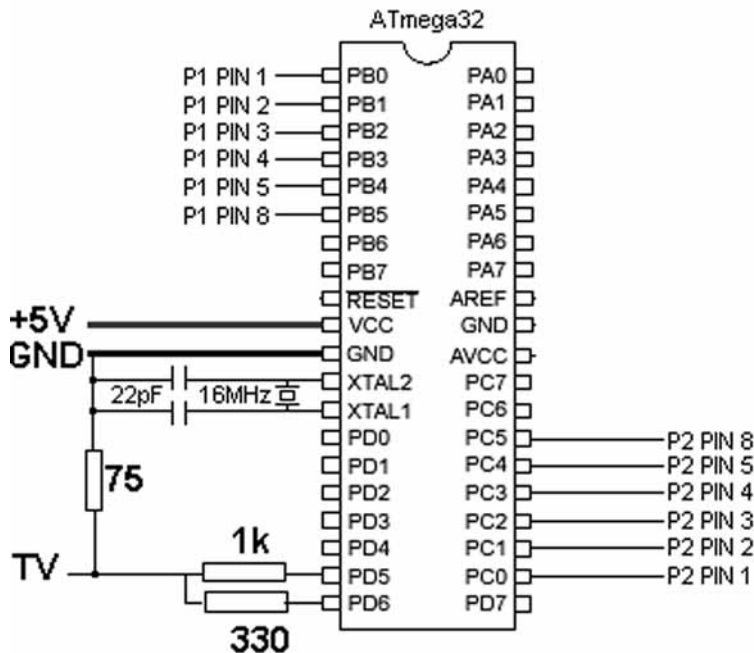


Abb. 182:
Nova-Strike-
Schaltplan

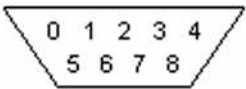
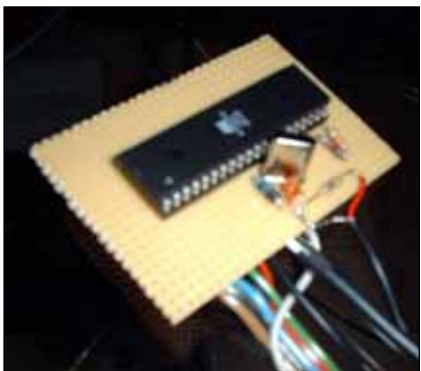


Abb. 183:
Lochraster-
aufbau

Abb. 184:
Nova-Strike-
Bildschirm-
foto



Abb. 185:
Fuses-
Einstellun-
gen

	Low	High	
Default:	0xB1	0x99	Apply default values
Current:	0xEF	0xD1	Apply user values

	Low	High
☐ BODLEVEL	☐ BODEN	☐ OCDEN
☐ SUT1	☐ CKOPT	☐ JTAGEN
☒ SUT0	☒ CKSEL3	☒ SPIEN
☐ CKSEL2	☒ CKSEL1	☒ EESAVE
☐ CKSEL1	☐ CKSEL0	☒ BOOTSZ1
☐ CKSEL0		☒ BOOTSZ0
		☐ BOOTRST

AVR-CP/M-Minirechner

Links:

<http://spritesmods.com/?art=avrcpm>
http://www.mikrocontroller.net/articles/AVR_CP/M
<http://www.mikrocontroller.net/topic/177481#1756046>
http://avr.cwsurf.de/?AVR_CP%2FM
<http://www.moria.de/~michael/cpmtools/>

Jeroen Domburg (Sprite_tm) hat als »Proof-of-concept« aus einem Atmega88-AVR, einem gerade verfügbaren 256x4-DRAM und einer SD-Karte einen minimalistischen CP/M-Rechner geschaffen. Dieses Projekt hatte mich sofort in seinen Bann gezogen, und ich musste es unbedingt nachbauen. Fast schon ungläublich schaute ich, als der Aufbau dann auch tatsächlich funktionierte! Im Mikrocontroller-Forum habe ich dann Gleichgesinnte gesucht, um das Projekt ggf. etwas weiter zu verbreiten und weiter zu entwickeln.

Das Mini-CP/M-System ist über serielle Konsole mit 38400 8N1 erreichbar (zum Beispiel Hyperterm/Minicom). Das Disk-Image wird ohne Filesystem einfach mit `dd if=diskimage of=/dev/sdf1 bs=512` auf eine SD-Karte gespielt (/dev/sdf1 ggf. durch anderen Bezeichner ersetzen).

Mit den CPMTOOLS von Michael Haardt lässt sich das Disk-Image ändern. Die Win32-Programme müssen zwingend im Wurzelverzeichnis entpackt werden, so dass \cpmtools entsteht.

Aufrufbeispiele:

```
cpmls -f avrcpm diskimage zeigt Inhaltsverzeichnis an
cpmcp -f avrcpm diskimage datei.com 0:datei.com kopiert
datei.com in das image
cpmcp -f avrcpm diskimage 0:datei.com datei.com kopiert
datei.com aus dem image
cpmrm -f avrcpm datei.com löscht datei.com aus dem image
```

Verbindungsliste:

<pre> .equ ram_oe = 2 .equ ram_a8 = 3 .equ mmc_cs = 4 .equ ram_a5 = 5 .equ ram_a6 = 6 .equ ram_a7 = 7 </pre>	
<pre> ;Port B .equ ram_a4 = 0 .equ ram_a3 = 1 .equ ram_a2 = 2 .equ ram_a1 = 3 .equ mmc_mosi = 3 .equ ram_a0 = 4 .equ mmc_miso = 4 .equ ram_ras = 5 .equ mmc_sck = 5 </pre>	<pre> ;Port C .equ ram_d1 = 0 .equ ram_w = 1 .equ ram_d2 = 2 .equ ram_d4 = 3 .equ ram_d3 = 4 .equ ram_cas = 5 </pre>

Schaltplan:

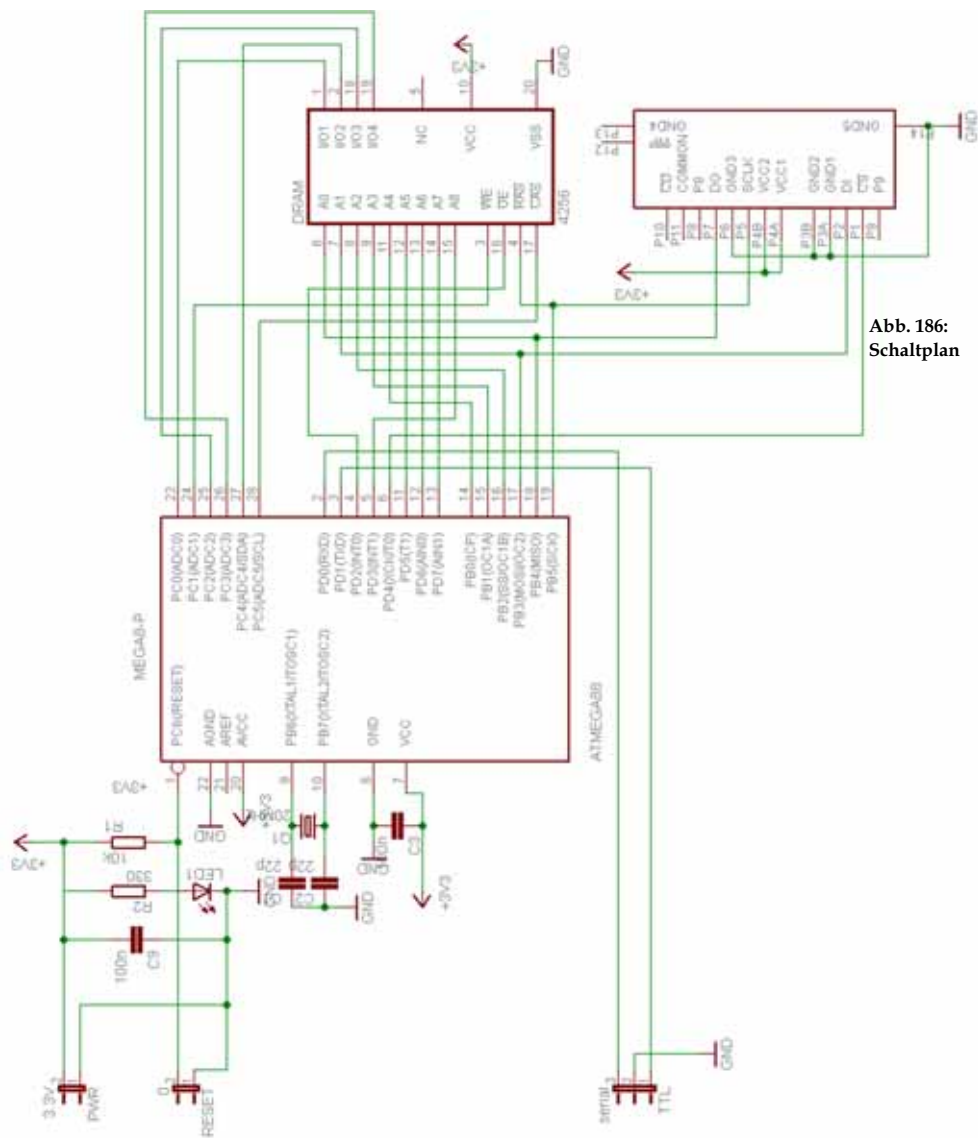


Abb. 186:
Schaltplan

Bauteile-Pinouts:

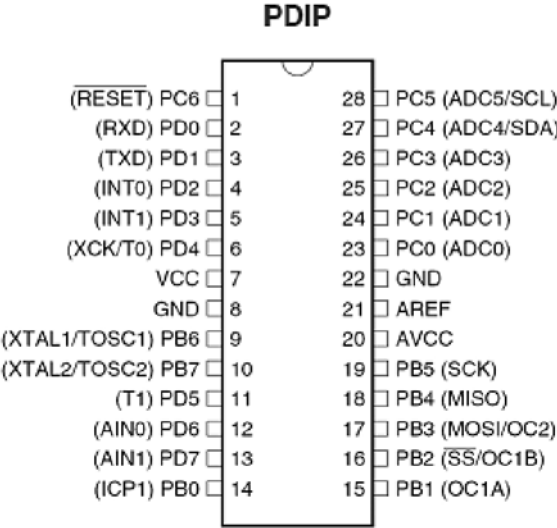
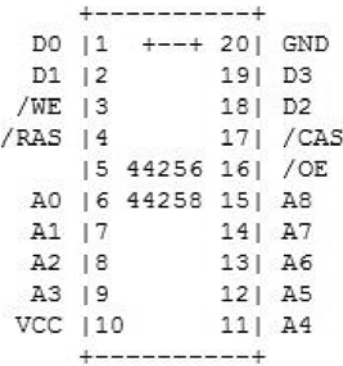


Abb. 189:
Pinouts für
Atmega88

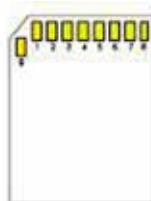
44256, 44258
256kx4 DRAM.

Abb. 190:
DRAM
254x4



SD-Karte:

Abb. 191:
Pinouts für
SD-Karte



SD Card

The Secure Digital card dimensions are: 24mm wide x 32mm long. The standard width is 2.1mm, while the Thin SD Memory Card has a width 1.4mm.

Secure Digital Pinout			
Pin #	Pin Name	SD Signal Function SD Mode	SPI Signal Function SPI Mode
1	DAT3/CS	Data Line 3	Chip Select/Slave Select [SS]
2	CMD/DI	Command Line	Master Out/Slave In [MOSI]
3	VSS1	Ground	Ground
4	Vdd	Voltage Supply [2.7v or 3.6v]	Voltage Supply [2.7v or 3.6v]
5	Clock	Clock	Clock [SCK]
6	Vss2	Ground	Ground
7	DAT0/D0	Data Line 0	Master In Slave Out [MISO]
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

Tools und Utilities

64'er Commander

Link: <http://c64.revivalteam.de>

Mit dem Programm lassen sich zum Beispiel D64-Diskimages öffnen, erzeugen, Dateien zwischen Images kopieren, verschieben, löschen etc. Außerdem lassen sich einzelne Programme aus Diskimages extrahieren.

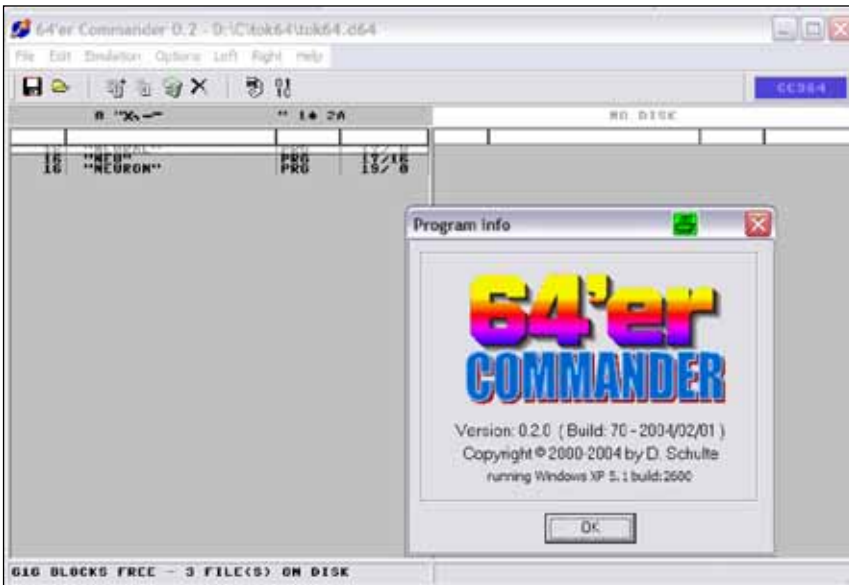


Abb. 192:
64'er
Commander

tok64

Copyright (c) 1996 Cris Berneburg <74171.2136@compuserve.com>

Link: <http://ourworld.compuserve.com:80/homepages/pcgeek/>

Zum Wandeln von BASIC-Programmen in ASCII-Text und umgekehrt

Abb. 193:
tok64

```

Eingabeaufforderung - tok64 /help
or: TOK64 /help

D:\C\tok64>tok64 /help

TOK64: Text <=> C64 Basic Converter
Version 1.4 6/15/96 Basic extensions. More special keywords.
Copyright (c) 1996 Cris Berneburg <74171.2136@compuserve.com>
http://ourworld.compuserve.com:80/homepages/pcgeek/
This software is FREEMWARE and is freely distributable.

TOK64 converts ascii text file representations of Commodore basic
program listings into program images suitable to run on a C64. It
also converts basic programs back into ascii text listings. TOK64
does NOT perform syntax checking.

*** Use the "/list" command line argument for a list of valid commands.

You may specify multiple commands and/or file names on the command
line in the order you want the commands to occur.

Default extensions are ".txt" for ascii text files and ".prg" for
Commodore Basic program files. That means you may specify extensions
but if you don't TOK64 will look for files with the defaults.

Hit <RETURN> to continue:
  
```

64Copy

Copyright by Peter Schepers 1994-2005

Hiermit lassen sich viele Formatumwandlungen durchführen. Also zum Beispiel: PRG in T64. Das Programm enthält noch viele weitere Funktionen und ist das Allzwecktool für Formatkonvertierungen.

Abb. 194:
64Copy



Emulatoren

Wer einfach mal ein bisschen Retro-Feeling auf seinem PC haben möchte, für den sind sicher auch entsprechende Emulatoren interessant.

VICE

Der VICE-Commodore-8-Bit-Emulator ist für mich einer der besten Emulatoren für PET/CBM, VC20, C64 und die 264er Reihe (C16, +4) von Commodore. Open Source und für viele Plattformen ist ein sog. Port verfügbar.

www.viceteam.org

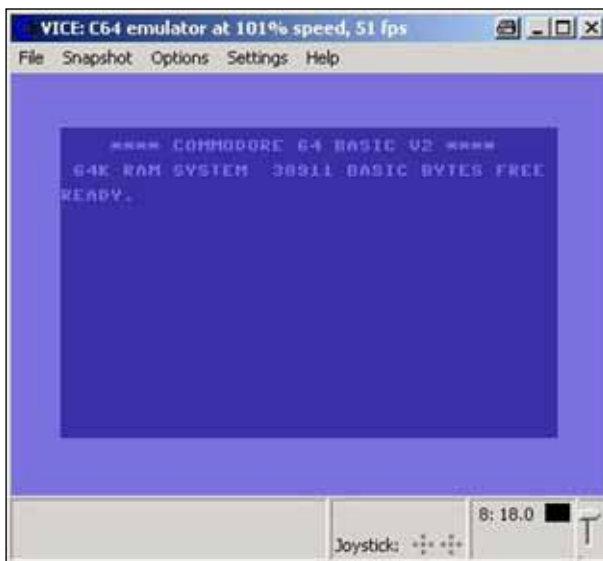
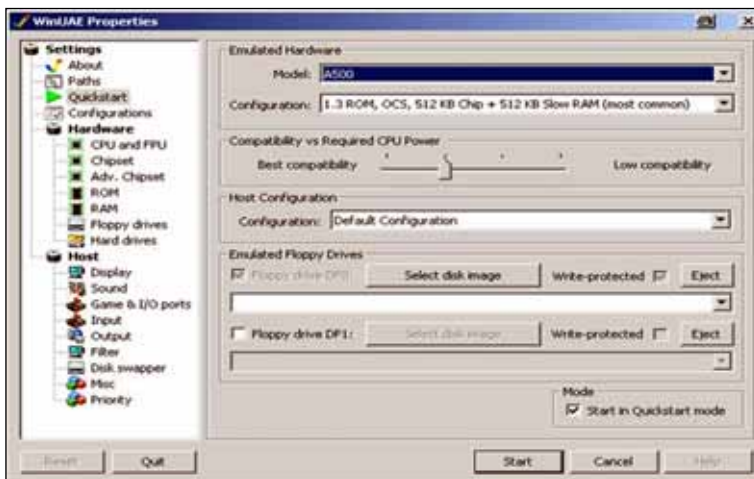


Abb. 195:
VICE

(Win)UAE

Universal Amiga Emulator: www.winuae.de

Abb. 196:
UAE



Wenn das Kickstart-ROM ausgewählt wurde, Floppy-Image (adf) wählen und Start anklicken:

Abb. 197:
UAE funktioniert
bestens.



Weiterführende Webadressen

Verein zum Erhalt klassischer Computer e.V.

www.classic-computing.de

Der Verein veranstaltet einmal im Jahr die sog. Classic Computing. Auf dieser Ausstellung kann eine Vielzahl von teilweise sehr seltenen Exponaten bewundert werden.

Eine Vielzahl der Systeme wird im funktionsfähigen Zustand gezeigt. Auch kann Mann/Frau einmal selbst Hand anlegen und einen der alten Klassiker an original Hardware »zocken«.



Abb. 198:
Webseite des
Vereins



Abb. 199:
Gruppenbild
der Vereins-
mitglieder
anlässlich
der Classic
Computing
2007

www.forum64.de

Abb. 200:
Forum64-
Webseite



www.a1k.org

Abb. 201:
A1K-
Webseite



Kleine Bauteilkunde

Widerstände

Farbe	Ziffer 1-3	Multiplikator	Toleranz
Silber		0.01	10 %
Gold		0.1	5 %
Schwarz	0	1	20 %
Braun	1	10	1 %
Rot	2	100	2 %
Orange	3	1k	
Gelb	4	10k	
Grün	5	100k	
Blau	6	1M	
Violett	7	10M	
Grau	8	100M	
Weiß	9	1G	



Beispiele:

Gelb - Violett - Rot - Gold = 4,7k Ohm 5 %

Braun - Schwarz - Orange - Gold = 10k Ohm 5 %

Abb. 202:
Widerstand

SMD-Widerstände

sind mit drei Ziffern beschriftet. Die ersten beiden Ziffern geben den Wert an, die dritte Ziffer den Multiplikator. Bei vier Ziffern geben die ersten drei den Wert an und die vierte Ziffer den Multiplikator.

Beispiele:

$$103 = 10 * 10^3 = 10k \text{ Ohm}$$

$$220 = 22 * 1 = 220 \text{ Ohm}$$

Elkos

Elektrolytkondensatoren sind gepolte Bauteile. Daher ist hier, wie bei allen gepolten Bauteilen, auf den korrekten Einbau zu achten! Der Minuspol ist bei diesen Bauteilen gekennzeichnet:

Abb. 203:
Elko



Tantal-Elkos

Hier ist der Pluspol mit einem »+« markiert.

SMD-Elkos

Abb. 204: Hier ist der Pluspol abgeflacht.
SMD-Elko



Dioden

Abb. 205: Bei diesen gepolten Bauteilen ist der Minuspol (Kathode) durch einen Strich markiert.



Leuchtdioden

Hier ist der Minuspol (Kathode) am Gehäuse abgeflacht. Das längere Anschlussbein ist die Anode (Pluspol).

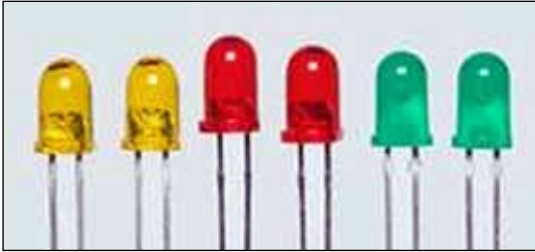


Abb. 206:
LEDs

Integrierte Schaltungen

Hier ist auf die Einbaurichtung der Kerbe zu achten. Fassungen sollten ebenfalls mit einer korrekten Position der Kerbe verbaut werden, so dass es später nicht zu einer falschen Bestückung kommt!

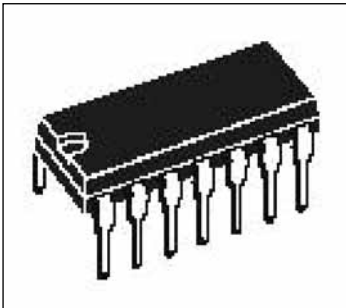


Abb. 207:
IC

Quarzoszillator im DIP14-Gehäuse

Pin 1 eines Quarzoszillators ist mit einem Punkt an der Gehäuseoberseite oder mit einer anders geformten Ecke gekennzeichnet. Elektrisch hat der Pin 1 keine Bedeutung.



Der Pin rechts daneben ist bei DIL 14 Pin 7 und bei DIL 8 Pin 4 (Zählung wie beim Standard-DIL-Gehäuse). Dieses ist der GND(Masse)-Pin.

Abb. 208:
Quarz-
oszillator

Der Pin über GND ist bei DIL 14 Pin 8 und bei DIL 8 Pin 5. An ihm liegt das Takt-Ausgangssignal an.

Der Pin oberhalb der markierten Ecke (links oben) ist bei DIL 14 Pin 14 und bei DIL 8 Pin 8. Er ist die Versorgungsspannung (meist +5 V).

Tipp

Noch ein genereller Tipp, falls auf einer Platine mal die Polung von Bauteilen nicht klar ersichtlich sein sollte: die Pins gegen einen bekannten Massepunkt (GND) auf Durchgang messen. Das ist dann der Minus-/Masse-/GND-Anschluss.

Danksagungen

Jogi (Joachim Nemetz) und *Jens Schönfeld* für die C64-Ersatz-PLA-auf-Eprom-27512-Schaltung und Angaben
JerryMaus (Andreas Senk) für seine C64-Assy-250407-Restaurationsstory
Pentagon (Thomas Wirtzmann) für seine C64-Restaurationsstory
Area51HT (Peter Trachte) für seinen Großumbau des C64
X1541 (Nicolas Welte) und *enthusi (Martin Wendt)* für NeoRAM + Treiber
Klaus Hegemann für die Pinout-Fotos und Angaben im DTV-Kapitel
Daniel Soegert/MadModder für sein Übersichtsbild der DTV-Anschlusspunkte
Bencao74 (Jochen Zurborg) für sein DTV-Mod-Foto
Roland Tögel für Easyflash – einfaches Flash-Tool für das DTV
Dennis van Weeren für seine tolle Leistung, einen Amiga 500 in FPGA zu realisieren
Tobias Gubener (tobiflex) für seinen 68000-Core und die Minimig-auf-DE1/2-Portierung
Jan Derogee für die 1541-III-SD-Karte am C64 auf PIC-Basis
Lars Pontoppidan für seine MMC2IEC-Umsetzung der 1541-III auf AVR-ATmega32-Basis
Ingo Korb für die Weiterentwicklung der Firmware zum SD2IEC
Gideon Zweijtzter für die Entwicklung der 1541 Ultimate
Cyberdyne (Jan Janssen) für Bild und Text zur 1541U
Tom Kirk für seinen PS/2-Mausadapter für Amiga und Atari ST
Stefan Pitsch, (^TePe^) und *Stefan Hohendorf (HONI)* für die PC-Floppy-Umbauten für Amiga
Richard Stallmann für »seine« GNU (General Public License) und die Open-Source-Bewegung
Ing. Thomas Winkler (Diddl) für die Entwicklung des XS1541-Adapters und der Final Expansion für den VC20
Thomas Giesel (skoe) für die Entwicklung von Easyflash für den C64
Wikipedia und dessen unzähligen Autoren und Helfern
Und last but not least meiner Familie: *Heike, Robin* und *Janis* für die Geduld und die Stunden, die ich mit alten Computern verbracht habe anstatt mit ihnen
Und danken möchte ich an dieser Stelle auch meinen Eltern *Edith* und *Reinold*.

Über den Autor

Mein Name ist Peter Sieg. Ich bin am 26.12.1963 in Boffzen geboren. Im Jahr 2008 bin ich somit 44 Jahre jung. An Homecomputern hatte ich früher mal ZX81 und Atari ST. Mit C64 und Amiga hatte ich zu damaligen Zeiten wenig Berührungen.

Nach der Ausbildung zum Chemikanten habe ich Dosiersysteme betreut. Schulungen in Assembler und Fortran auf DEC PDP-11 folgten ABAP unter SAP R/2, später R/3. Grundkenntnisse in Pascal, C, VB und Java sind inzwischen hinzugekommen.

Generell geht es mir allerdings gegen den Strich, noch brauchbare Sachen einfach wegzuerwerfen. Dabei hat es mir insbesondere alte Computertechnik angetan. Diese »alte« Technik ist hochinteressant, und durch die Spieleorientierung bleiben auch die eigenen Kinder nicht ganz außen vor.

Webseiten

<http://petersieg.kilu.de>

<http://petersieg.cwsurf.de>

<http://petersieg.webng.com>

<http://de.geocities.com/petersieg>

Abb. 209:
Ein Bild von
mir an mei-
nem Arbeits-
platz



...und wenn Sie mal nicht selbst löten möchten

- ✓ Adapter für optische Maus am C64/C128, Amiga, Atari ST
- ✓ Reader für moderne Speicherkarten am C64/C128
- ✓ Freezer-cartridge für C64/C128
- ✓ flimmerfreies Bild auf VGA-Monitoren für Amiga und C64
- ✓ Ersatz für Commodore-PLA Chips (C64, 264-Serie, C610, C710)
- ✓ Netzwerk für Amiga, C64, C128
- ✓ RS232 Schnittstelle für Amiga, C64, C128
- ✓ 4-Spieler Adapter für C64/C128
- ✓ Festplatten-Controller für Amiga
- ✓ Festplatten-Beschleuniger für Amiga 1200
- ✓ Adapter für PC-Tastatur zu Amiga

..oder einen Floppy-Controller für Ihren PC, der die Formate von C64 und Amiga lesen und schreiben kann. Damit können Sie ganz bequem und schnell D64- und ADF-Dateien aus dem Internet zu Ihrem Liebling bringen.

Gute Hardware für gute Computer. Seit 1994.

Individual Computers Jens Schönfeld GmbH
Im Zimmer 6
52152 Woffelsbach
kein Ladenverkauf!
Vertrieb über Vesalia, www.vesalia.de

 **INDIVIDUAL**
C O M P U T E R S
www.icomp.de

Commodore

Aufstieg und Fall eines Computerriesen

**Ein kurzer Streifzug durch die Firmengeschichte
mit Daten, Fakten und den Gründen, warum
der Computerpionier am Ende scheiterte**

Autor: Boris Kretzinger

ISBN 978-3-938199-04-6

Format 17x22 cm

108 Seiten

Mit 42 meist farbigen Fotos und Abbildungen

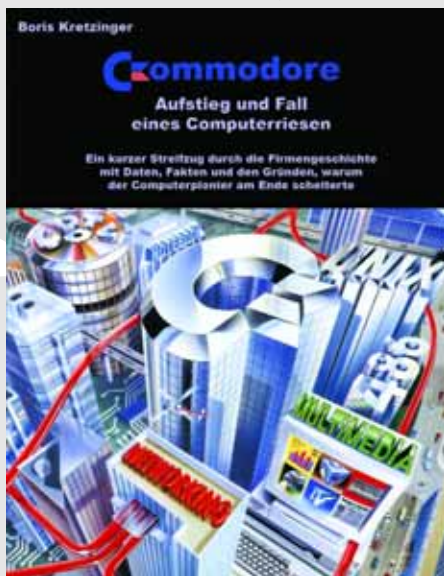
17,80 €

Der Computerhersteller Commodore gehörte zu den Pionieren der Branche. Mit dem PET wurde 1977 der erste »echte« PC entwickelt, der maßgeblichen Einfluss auf die weitere Entwicklung der Computerbranche haben sollte.

Viele weitere erfolgreiche und weniger erfolgreiche Commodore-Computer folgten: VC20, C64, C16/116/Plus4, C128, Amiga und IBM-kompatible Rechner.

Da Commodore und ihr Gründer Jack Tramiel (der später Atari übernahm) in der Literatur bisher geradezu stiefmütterlich behandelt wurden, widmet sich dieses Buch dem Phänomen Commodore.

Unternehmen Sie mit diesem Buch einen Streifzug durch die Firmengeschichte von Commodore und erleben Sie Erfolge, Katastrophen und interne Machtkämpfe mit. Sie erhalten Daten, Fakten und erfahren auch, warum der Computerpionier am Ende scheiterte.



Vor 25 Jahren hatten viele Teenager ihre erste Berührung mit einem Computer – meist in Form eines C64. Einige der damaligen Teenies halten ihm und seinem Nachfolger, dem Amiga, auch heute noch die Treue.

Peter Sieg ist einer von ihnen. In diesem Buch zeigt er auf, wie man defekte C64 repariert, wie man die Rechner fachgerecht säubert und von Gilb befreit, ROMs sinnvoll durch EPROMs ersetzt, SID-Karten an den C64 anschließt und gibt viele weitere Tipps, damit die Computer länger »leben«.

Ein besonderes Bonbon sind die Tipps, wie man TFT-Monitore und PC-Mäuse an einen Amiga 500 anschließt.

Tipps und Anregungen, was man mit dem Minimig und dem C64DTV alles anstellen kann, runden das Buch ab.

Die dritte Auflage wurde stark erweitert und enthält jetzt noch mehr tolle Informationen, Tipps und Anregungen.

17,80 €



Edition
Retrobooks