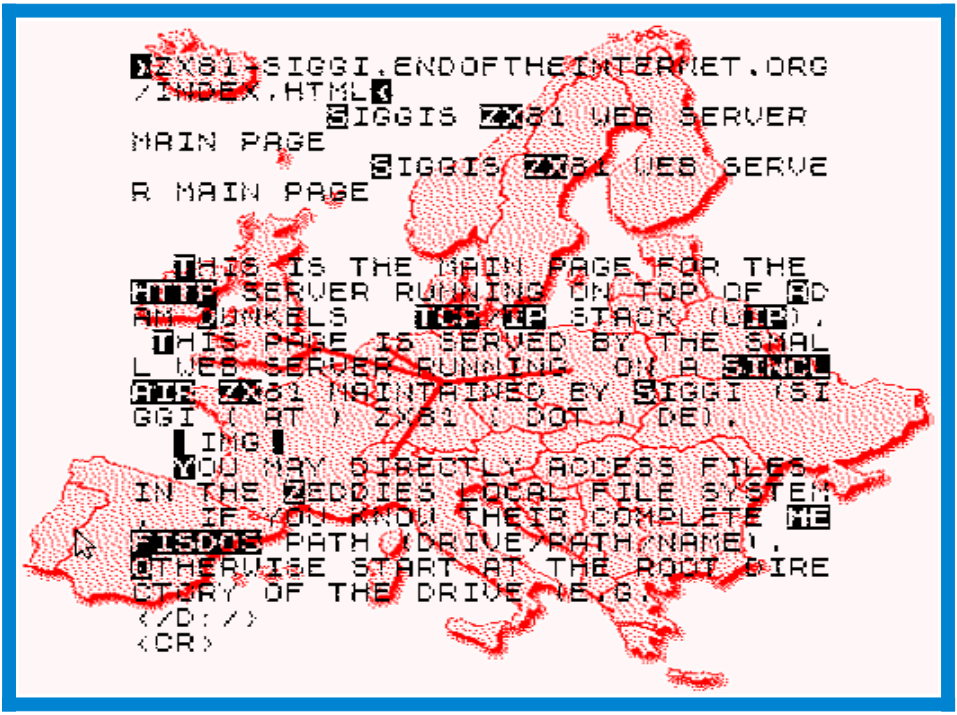
ZX-TEAM MAGAZIN

SONDERAUSGABE 2013



*** ZEDDY-INTERNET *** LEDS ** ** DIASHOW MIT ZXPAND * AX81 * * MK14 NACHBAU * ZX81XRAM **** *** 4MB-RAM-KARTE * 1K HRG *** **** UND VIELES MEHR *********

RUN 

Inhalt

Vorwort	2
Der ZX81-HRG-Bildbetrachter	4
ZeddyNet 1.0 – Der ZX81 im Internet	10
1K-HRG-Spiele: Sprechstunde mit Dr Beep	18
Grafikwettbewerb für den ZX81	22
Neues vonne Küste von Willi	24
LEDs nicht nur im Hintergrund	26
ZX81XRAM – externes RAM für ZX81	32
Ahnenforschung – Sinclairs MK14	36
„Läuft..." von Jens	40
ZX81CCB – Videomodul für ZX81	42
AX81 – Ein ZX81 für die Westentasche	43
Die 4 MB Speichererweiterung für den ZX81	48
„Die Zukunft hat begonnen" von Ali	52
Dies& Das	54

Dieses Magazin ist eine Sammlung von Beiträgen *von und für* Mitglieder des ZX TEAM, zusammengestellt und bearbeitet von

Oliver Lange oliver@zx81.de

Besonderer Dank für die tatkräftige Unterstützung gilt

Jens Sommerfeld
Klaus-Dieter Jahnke
Paul-Olaf Veltjens
Thomas Rademacher

Bitte insbesondere bei Nachbauten beachten: Die Autoren haben größte Sorgfalt walten lassen, dennoch gibt es ***keinerlei Zusagen oder Garantien bezüglich Eignung, Funktionalität oder Zulässigkeit des Dargestellten.*** Bitte unbedingt selber prüfen!

Bei Fragen und Anmerkungen ist auch das Forum http://zx81.tlienhard.com/ eine erstklassige Anlaufstelle, viele der Autoren sind im Forum aktiv.

25. März 2013

Vorwort



„Bitte Lächeln“

Die Computer-Fachpresse arbeitete schon immer gerne mit Attributen. Zu den häufigsten Begriffen im Zusammenhang mit unserem ZX81 zählte wohl stets das Wort „belächelt“. Daran ist auch nichts auszusetzen, immerhin ist das Lächeln ja eine schöne und positive Regung.

Die ZX-Gemeinde werkelt seit Jahren mit Eifer daran, dass sich zum Lächeln bei genauerem Hinschauen dann Erstaunen und Faszination gesellt: Jetzt ist der Zeddy vollends im Internet angekommen, es gibt Schaltungen und Programme, mit denen jeder Interessierte seinen ZX81 zum Chatten, Downloaden, und Browsen bringen kann.

Dauerthema bei den kleinen ZX-Computern sind die Grafikeigenschaften. Eine Diashow, die das Lächeln der lieben Mitmenschen auf den von einer Z80A-CPU generierten monochrom-Schirm bringt, stellen Thomas und Klaus im Heft vor.

Ein besonderes Lächeln ruft die Erinnerung an das letztjährige ZX-TEAM-Treffen in Mahlerts hervor. Angefangen von der wieder einmal perfekten Organisation durch Peter, der Bereicherung durch viele internationale ZX-Freunde, die bei der Anreise keine Mühen scheuten, bis hin zu den vielfältigen und spannenden Vorträgen. Ein paar Impressionen finden sich auf der rückwärtigen Umschlagseite. Für das Treffen 2013 lassen die Vorzeichen jedenfalls eine ähnlich famose Veranstaltung erhoffen.

Viel Spaß beim Lesen und frohes Zeddeln wünscht

Euer ZXTM Team

HISTORISCHE WERBUNG

Neu! Der Sinclair ZX81 Personal-Computer.

Als Bausatz DM 298,-.

Die Fertigversion DM 398,-.



Erschließen Sie sich umfangreiches Computer-Verständnis. In wenigen, faszinierenden Stunden.

1980 – das Jahr eines einzigartigen Durchbruchs: Sinclair präsentiert mit dem ZX 80 der Welt ersten Personal-Computer für unter fünfhundert Mark. Mit bisher unübertroffenen Leistungsdaten.

Weltweit konnten über 50.000 Exemplare verkauft werden, und auch in der Fachwelt fand das Gerät höchst positive Resonanz.

Jetzt kann Sinclair diesen technologischen Vorsprung weiter ausbauen. Mit dem neuen ZX 81.
Für nur DM 398,-.

Kleine Kosten-/ Nutzenrechnung.

ZX 81 – das ist nach wie vor eine der einfachsten Möglichkeiten, Computer zu verstehen und mit ihnen zu arbeiten. Aber jetzt mit erweiterten Kapazitäten! Denn der Z 80-Mikroprozessor wurde mit dem neuen, noch leistungsstärkeren „8k Basic ROM" kombiniert. Zur „trainierten Intelligenz" des Computers.

Dieser Chip mit Dezimalzahlen, logarithmischen und trigonometrischen Funktionen ermöglicht das Erstellen von Graphiken und legt bewegte Displays an. Weitere Vorteile sind z.B. die Speicherung von Programmen auf Kassetten, die Wiedergabe bestimmter gespeicherter Kassetten-Programme, sowie das Abrufen einzelner Programme per Keyboard.

Unerreichte Preis-/Qualitätsrelation.

Das Geheimnis heißt technologische Weiterentwicklung! Wo der ZX 80 bereits 40 Chips auf 21 reduzierte, braucht der ZX 81 nur noch ganze 4!

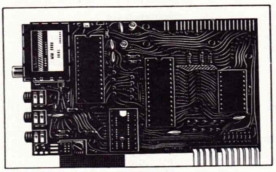
Die Lösung: der revolutionäre, neue Masterchip von Sinclair. Er ersetzt 18 Chips aus dem ZX 80!

Für den schnellen Kontakt: Service-Karten am Heftende

Als Bausatz oder die Fertigversion – wie Sie wünschen.

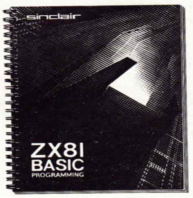
Die Abbildung veranschaulicht den unkomplizierten Zusammenbau des ZX 81. Einfach die mit den 4 Chips und den anderen Bauelementen bestückte Platine anlöten – fertig. Das passende Netzteil (600 mA bei nom. 9 V) ist bei Bausatz und Fertigversion im Lieferumfang enthalten.

Beide Versionen sind komplett ausgestattet mit allen Anschlußkabeln für TV (Farbe, s/w) und Kassettenrecorder.



Der ZX 81-Aufbau mit Mikroprozessor, neuem „8k Basic ROM", RAM – und dem einzigartigen Masterchip.

Das neue Basic-Handbuch. In deutscher Sprache.



Jedem ZX81 ist ein leicht verständliches, spezielles Handbuch beigefügt. Ihr kompletter Basic-Programmierkurs. Von der Einführung bis zu komplexen Programmen.

BILDBETRACHTER

Der ZX81-HRG-Bildbetrachter – ein Traum geht weiter

Der Bildbetrachter aus dem letzten Heft – jetzt verbessert und angepasst an andere Massenspeicher wie das ZXpand Interface

Von K.-D. Jahnke und Th. Rademacher

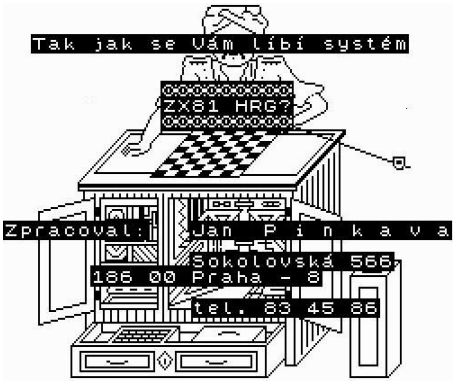
Die neuen Speichermedien am ZX81 (damit meine ich auch schon die 1541-Floppy und die MeFis-DOS-Lösung) bringen die Möglichkeit mit, Speicherbereiche (also zum Beispiel Grafik-Daten) zu sichern und zu laden, am ursprünglichen Kassetteninterface ist das noch nicht gegeben, allenfalls mit einer Maschinencode-Routine, doch dann vermutlich ziemlich nervenaufreibend wegen des auf Kassette nicht möglichen "wahlfreien Zugriffs". Richtig komfortabel (damit meine ich insbesondere schnell) ist es aber mit USB-Stick oder SD-Card geworden.

Im 2012er Sonderheft stellten Klaus und ich das hierauf basierende Diashow-Programm vor, dessen einziger Nutzer bisher möglicherweise immer noch Klaus ist, denn es ist auf seine recht exotische Hardware zugeschnitten.

Soft- und Hardware-Voraussetzungen

Daher habe ich mich an die Arbeit gemacht, das Programm für eine HRG (und Hardware-Basis) umzuschreiben, die verbreiteter ist. Meine Wahl des Grafiktreibers fiel auf die gute alte HRGCSFR.[1].

Sie bietet den Vorteil, daß keine RAM-TOP-Manipulation erforderlich ist: Die Routinen und auch die Grafikdaten sind in REM-Zeilen am Programmanfang untergebracht. Auch benötigt sie keinen RAM im Bereich von 8 bis 16K. Als Hardwarebasis setzt die neue Lösung ein ZXpand voraus, das im Team immerhin schon eine gewisse Verbreitung gefunden hat.

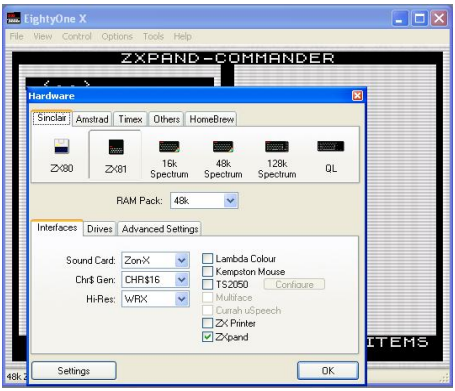


Setzt heute noch Maßstäbe: die HRGCSFR

Doch auch, wer über ein solches nicht verfügt, kann das Programm am PC unter der EightyOne-Emulation [2] nutzen, zu der Sir Morris eine Erweiterung [3] geschrieben hat, die das ZXpand nachbildet.

Mit kleinem Aufwand, Eintragen des betreffenden Ladebefehls in Zeile 6

und ggf. Korrigieren der RAM-Position in Zeile 10, läßt sich das Programm auch für andere Hardware, Commodore-Floppy, MeFisDOS, VDrive usw., anpassen.



Die erforderlichen Einstellungen für den EightyOne-Emulator

Mehr noch: Mit etwas konzentrierter Arbeit am Hexeditor läßt sich die HRG gegen die Semi-HRG von S. Schmidt austauschen. Diese Semi-HRG ist auch ohne Hardwareeingriff auf jedem Zeddy nutzbar, leider mit den bekannten Qualitätseinbußen.



Als Fahndungsfoto eher ungeeignet: ein Portrait des Autors in der Semi-HRG-Version des Diashow-Programms

Voraussetzung für das Diashow-Programm bleibt dennoch ein Ladebefehl für Speicherbereiche, also eine angeschlossene Hardware mit alternativem Massenspeicher.

Was ist neu?

Auf die in der ersten Version noch genutzte RESTORE/READ-Lösung [4] habe ich schweren Herzens verzichtet, es traten lästige und unerklärliche Effekte auf, weil das Programm zu komplex und/oder der Speicherplatz zu knapp wurde, nun liest eine herkömmliche Basic-Schleife die Namen aus der Liste aus. Dafür traten bisher keine Probleme mit "verbotenen Bildnummern" auf.

Wir haben auch den Gedanken verworfen, Matthias' POWER-Basic zu verwenden, hier hätte jeder Bildname in den DATA-Zeilen noch mit einem Paar Anführungsstrichen versehen werden müssen, was weitere 512 Bytes des RAM verschlungen hätte.

```
****** ZX81 HRG-DIA-SHOW *******
****  START MIT GOTO 9000  *****
*********************************

 K.-D. JAHNKE UND TH. RADEMACHER
          NOVEMBER 2012
    SYSTEMANFORDERUNGEN
    ZX81 MIT ZXPAND (M=H)
    SD-CARD ALS BILDSPEICHER
 BESTAND 27 VON 256 MOEGLICHEN
*********************************
<S>ELBE BILDER NACHEINANDER
<Z>UFAELLIGE REIHENFOLGE
<V>ORWAERTS BLAETTERN
<R>UECKWAERTS BLAETTERN
     <M>ENU
<N>AMEN EINGEBEN
<E>NDE
```

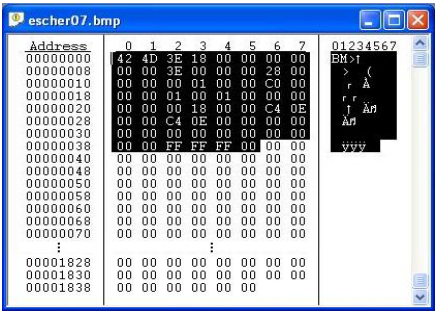
Neu: Anzeige der Bilderanzahl

Das Programm ist durch das ständig verfügbare Menu auch ohne viele REM-Zeilen weitgehend selbsterklärend, die Menu-Auswahl ist von Ziffern auf Buchstaben umgestellt. Die Anzahl der genutzten Bilder von den 256

möglichen wird am Anfang der Bilderliste mit gespeichert und vom Programm abgefragt, so daß die zufällige Wiedergabe nicht auf nicht vorhandene Bilder zuzugreifen versucht.

Woher kommen die .B-Dateien?

Wie gehabt wird vorausgesetzt, daß sich das Programm und die Speicherabzüge (.B-Dateien – für die Semi-HRG-Version .T-Dateien) im selben Verzeichnis befinden. Die .B- bzw. .T-Dateien müssen exakt acht Zeichen lange Namen ha-ben, was wir meist durch Auffüllen mit Nullen erreicht haben, also beispiels-weise ESCHER7.B -> ESCHER07.B. Eigene .B-Dateien lassen sich erstellen, indem ein (Schwarz-Weiß-) Bild von 256 Pixeln Breite und 192 Pixeln Höhe mit MS Paint auf dem PC erzeugt wird. Dann wird es invertiert und durch Spiegeln auf den Kopf gestellt. Wenn es nach dieser Umwandlung abgespeichert (als .BMP) wurde, muß von der erzeugten Datei der Header, die ersten 62 Bytes, entfernt werden, übrig bleiben die altbekannten 6144 Bytes [5].

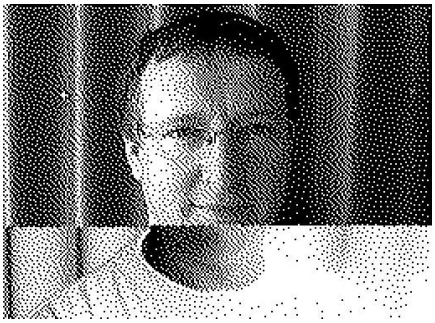


Diese Bytes müssen entfernt, danach die Datei als filename.B abgespeichert werden.

Die Diashow kann durch Nutzung des Menüpunkts <N>amen eingeben mit den .B-Dateien der eigenen Bilder ausgestattet werden. Zum Erzeugen der .T-Dateien aus diesen .B-Dateien dient das im vorigen Sonderheft erwähnte Programm UMWA01.P. [6].

Ist ein Programm jemals fertig?

Wer sich das Listing intensiver anschaut, entdeckt vermutlich, daß das Programm noch "in Bearbeitung" ist. Mir schwebten noch einige Features vor, zu deren vollständiger Einarbeitung mir aber die freie Zeit fehlt: Anhalten und Fortsetzen der Diashow, Invertieren des (angehaltenen) Bildes [vorbereitet als GOSUB 25], Schwärzen des Bildschirms [vorbereitet als GOSUB 19]. Wichtig ist, daß das Programm läuft - und nun auch ein paar Nutzer mehr erreicht.



Ein Portrait wird invertiert.

Die UDG (Schriftausgabe im Grafikbetrieb) funktioniert in Zusammenarbeit mit dem ZXpand leider nicht, sonst hätte vor dem Laden die Nummer und der Name des Bildes eingeblendet werden können. Auch das Flackern beim

Bildwechsel läßt sich nicht vermeiden - beim Bild-Laden springt die HRG in den Textmodus zurück und der Grafik-Modus muß erst wieder neu aufgerufen werden.

Januar 2013

Updates zu früheren Heften

Beim Arbeiten am neuen Diashow-Programm fand sich nebenbei die Antwort auf die Frage aus dem ***Team-Magazin 6/1992***, was der OVR-Befehl in der HRGCSFR bewirkt: Schriftzeichen (oder weitere Grafikelemente) können über eine Grafik geschrieben werden, ohne die darunterliegenden Pixel zu verändern, die Handhabung ist dann einfach, in die betreffende Befehlssequenz einzufügen: USR OVR, Ebenso wird der INV-Befehl verwendet: USR INV löscht dann also Punkte, statt sie zu setzen oder stellt Schriftzeichen invertiert dar.

```
  66 PRINT USR PRT,AT 1,17;"F■(Y
=COS (2*X)■)",USR OVR,USR PRT,AT
2,6;"F■",AT 10,5;"F■"
  68 GOSUB VAL "2E3"
  69 PRINT USR SCL,USR SCL,USR S
CL,USR SCL,USR SCL
  70 PRINT USR INV,USR PRT,"UM8M
 TAK9 KRESLIT - N7SLEDUJ8C8 ",US
R SCL,USR PRT,"JEDNODUCH6 PROGRA
M NAKRESL8 4ADU",USR SCL,USR PRT
,"■DOME3K0 JEDN8M TAHEM■:
  ";
```

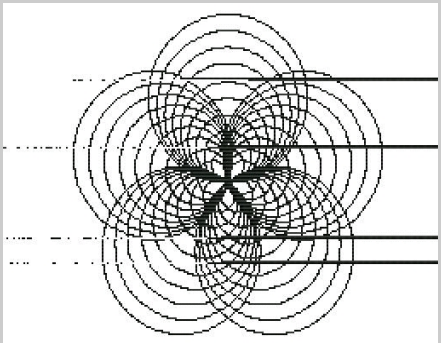
OVR- und INV-Befehl in der HRGCSFR

Diese, sozusagen, Befehlszusätze sind auch miteinander kombinierbar. Zum Invertieren eines vorhandenen Bildes ist der Befehl (also anders als in TM 4/92 angegeben) nicht verwendbar, dafür mußte ich eine eigene Routine schreiben: nach dem Laden einfach mal Menü mit E verlassen und GO-SUB 25 ausführen.

```
RAM :5DA0 87FAC25D30EE0E205E230608CB12CB13 ■...K.:4.7...>.<
RAM :5DB0 CB1ADBFB1F30FB7AD3FB10F00D20E9C1 .,..3K....(.$4..
RAM :5DC0 10C83E04D3FBCD0702C3155B21635F3E (.Y.......+.5..Y
RAM :5DD0 49BEC24B0DE9015700C52A1040CD9E09 .Y..$... .E(..2.
RAM :5DE0 C1213F5EEDB8C3155BB4B38F36000000 .5Z..S.+.ON?Q
RAM :5DF0 B42BAB8F36420000A831B88F364E0000 OFF?Q.  CLS?Q.
RAM :5E00 AE33BB8F36700000B43BB78F367E0000 INV?Q.  OVR?Q.
RAM :5E10 B828B18F368C0000B537B98F37D60000 SCL?Q.  PRT?R.
RAM :5E20 B531B98F39220000A937BC8F39340000 PLT?T6  DRW?TO
RAM :5E30 A835BE8F3B040000BA29ACBF3B980000 CPY?V.  UDG?V/
RAM :5E40 7600018604EABAA9AC7676FFFF000000 . ....UDG....
```

Die Befehle der HRGCSFR im Debug-Fenster des Emulators NO$ZX8.EXE

Im ***Sonderheft 2012*** *(S.30)* berichtete ich von Bildstörungen in der Memopak-HRG, wenn die Pixel den Code h76 ergeben. Auf dem 2012er Treffen konnte ich mit der realen Hardware arbeiten und mußte dort solche Effekte nicht beobachten: das Problem besteht offenbar nur unter der Emulation.



Zum Glück nur ein Effekt der Emulation

Quellen und Weblinks

[1] Vorgestellt in den Team-Magazinen 4/1992 S. 22 und 6/1992 S. 23 – unter dem Namen HRG-DEMO.P im Verzeichnis \ZX81\Programme\Peter auf der ZX-Team-CD zu finden

[2] http://www.aptanet.org/eightyone/downloads/archive/eightyone1.0.zip

[3] http://www.sinclairzxworld.com/viewtopic.php?f=3&t=982#p10249

[4] ZX-User Club Sammelband 1984 S. 59 "DATA für den ZX81" von G.W. Seefried
ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/ZXUserClub/Sammelband8401/Pages/ZXUserClubSammelband840100059.jpg

[5] Vgl. Team-Magazin 2/2006 S. 25 f.

[6] http://zx81-siggi.endoftheinternet.org/d:applic/hrg/basicod3/schmidt/umwa01.p

Programm

```
  0 REM HRG (C) PINK 84
```

Unterhalb Zeile 5 sind in versteckten Zeilen die HRGCSFR einschließlich der Grafikdaten und weitere Informationen untergebracht, deswegen sollte hier nichts gelöscht oder verändert werden.

```
  5 GOTO 9000
  6 LOAD "DIETER20.B;16640"
  7 RETURN
  8 REM BILDNAMEN IN LADEBEFEHL SCHREIBEN
  9 FOR R=1 TO 8
 10 POKE 27705+R,CODE A$(R)
 11 NEXT R
 12 RETURN
 13 REM BILDADRESSE IN READ-ROUTINE POKEN
 14 LET S=25378+9*P
 15 POKE 25361,INT (S/256)
 16 POKE 25360,S-256*INT (S/256)
 17 RETURN
 18 REM BILDSCHIRM SCHWAERZEN
 19 RAND USR 23296
 20 POKE 23344,113
 21 PRINT USR 23352,USR 23335
 22 POKE 23344,117
 23 RETURN
 24 REM BILD INVERTIEREN
 25 RAND USR 23296
 26 RAND USR 25362
 27 RETURN
 28 REM AUFBAU DES MENUES
 29 PRINT "****** ZX81 HRG-DIASHOW *******"
 30 PRINT "**** START MIT GOTO 9000 *****"
 31 PRINT "**********************************"
 32 PRINT
 33 PRINT " K.-D. JAHNKE UND TH. RADEMACHER"
 34 PRINT "           NOVEMBER 2012";
 35 PRINT
 36 PRINT "  ■■■ SYSTEMANFORDERUNGEN ■■■"
 37 PRINT "  ■■ ZX81 MIT ZXPAND (M=H) ■■"
 38 PRINT "  ■ SD-CARD ALS BILDSPEICHER ■"
 39 PRINT " BESTAND ";PEEK 25387;" VON 256 MOEGLICHEN"
 40 PRINT "**********************************"
 41 PRINT "<A>LLE BILDER NACHEINANDER"
 42 PRINT "<Z>UFAELLIGE REIHENFOLGE"
 43 PRINT "<V>ORWAERTS BLAETTERN "
 44 PRINT "<R>UECKWAERTS BLAETTERN "
 45 PRINT "  <M>ENU"
```

```
  47 REM "     <I>NVERTIEREN"
  48 REM "     <S>CHWAERZEN"
  49 REM "     <W>EITER"
  50 PRINT "<N>AMEN EINGEBEN"
  51 PRINT "<E>NDE"
  52 RETURN
  55 REM  NAME AUS LISTE LESEN
  56 LET A$=""
  57 FOR T=1 TO 8
  58 LET A$=A$+CHR$ (PEEK (S+T))
  59 NEXT T
  60 RETURN
  61 REM  PAUSE
  62 LET E$=INKEY$
  63 GOSUB 7*(E$="")+25*(E$="I")
+19*(E$="S")
  64 GOTO 66-4*(E$<>"W")
  65 REM  WEITER
  66 RETURN
  67 REM  NAMEN AUS LISTE LESEN
  68 FOR V=1 TO 8
  69 PRINT CHR$ PEEK (S+V);
  70 NEXT V
  71 PRINT
  72 RETURN
1000 REM  ANZEIGE ALLER BILDER
1005 FOR P=1 TO PEEK 25387
1006 GOSUB 14
1010 REM AT 0,0;
1015 REM A$="BEAUT500"
1020 GOSUB 56
1025 GOSUB 9
1030 GOSUB 6
1035 IF INKEY$="M" THEN RETURN
1040 PRINT USR 23296
1045 NEXT P
2000 REM  ZUFAELLIGE REIHENFOLGE
2005 REM USR 25326
2010 RAND 0
2015 REM A$="DIA-SHOW"
2020 LET P=1+INT  (RND*(PEEK 2538
7))
2025 GOSUB 14
2040 GOSUB 56
2045 GOSUB 9
2050 GOSUB 6
2055 IF INKEY$="M" THEN RETURN
2060 PRINT USR 23296
2065 GOTO 2020
3000 REM  BLAETTERN
3005 CLS
3010 PRINT "AB WELCHER BILD-NR.
? ";
3015 INPUT P
3020 GOSUB 14
3025 REM A$="DIA-SHOW"
3030 GOSUB 56
3035 GOSUB 9
3040 GOSUB 6
3050 PRINT USR 23296
3055 LET E$=INKEY$
3056 GOSUB 7*(E$="")+62*(E$="P")
3057 GOTO 4115-1055*(E$<>"M")
3060 LET P=P-1+2*(W$="V")
3065 LET P=P+PEEK 25387*(P=0)-PE
EK 25387*(P=PEEK 25387+1)
3070 GOTO 3020
4000 REM  NAMEN EINGEBEN
4005 CLS
4010 PRINT "BILD-NR. ? ";
4015 INPUT U
4016 PRINT " ";U;" ";
4020 IF U=0 THEN GOTO 4115
4025 IF U>256 THEN GOTO 4010
4030 LET S=25378+9*U
4035 IF U>PEEK 25387 THEN POKE 2
5387,U
4040 REM 15
4045 REM A$="DIA-SHOW"
4050 REM USR 25304
4055 REM U;" ";A$
4056 GOSUB 68
4060 PRINT "NEUER NAME: ";
4065 INPUT B$
4070 PRINT B$
4075 IF LEN B$=8 THEN GOTO 4095
4080 IF LEN B$=0 THEN GOTO 4115
4085 PRINT "BITTE EXAKT 8 ZEICHE
N,           LEERSTRING FUER ENDE
"
4090 GOTO 4060
4095 FOR V=1 TO 8
4100 POKE S+V,CODE B$(V)
4105 NEXT V
4110 GOTO 4005
4115 CLS
4120 GOSUB 29
4125 RETURN
9000 LET A$="DIA-SHOW"
9005 POKE 27721,28
9010 PRINT AT 0,0;
9015 RAND USR 25296
9020 GOSUB 29
9135 LET W$=INKEY$
9140 IF W$="E" THEN GOTO 9997
9150 GOSUB 7*(W$="")+1005*(W$="A
")+2005*(W$="Z")+3005*(W$="V")+3
005*(W$="R")+4005*(W$="N")
9220 GOTO 9135
9309 FOR R=1 TO 8
9310 POKE 27705+R,CODE A$(R)
9311 NEXT R
9312 RETURN
9314 LET S=25378+9*P
9317 RETURN
9356 LET A$=""
9357 FOR T=1 TO 8
9358 LET A$=A$+CHR$ (PEEK (S+T))
9359 NEXT T
9360 RETURN
9997 STOP
9998 SAVE "DIA-CSFR"
9999 GOTO 9000
```

ZEDDYNET

ZeddyNet 1.0 – Der ZX81 im Internet

Das scheinbar Unmögliche ist wahr geworden. Der ZX81 kann sich im Internet tummeln. Man kann mit dem Zeddy im HRG-Modus „live“ über den englischen ZX-Spectrum-Server „chatten“, Dateien aus Siggis TNF-Server hoch- und herunterladen sowie Mini-E-Mails senden und empfangen



Von Klaus

Der ZeddyFox Browser ermöglicht sogar das Surfen über das HTTP-Protokoll. Möglich wurde es durch Kombination modernster Mikroelektronikbausteine (dem WizNet-Baustein) und dem ZX-Know-How von Siggi Engel, Thomas Lienhard und Joachim Merkl.

Ich habe den Internetzugang mit dem Zeddy zum ersten Mal auf dem ZX81-Treffen in Mahlerts im März 2012 gesehen und mich kurze Zeit später mit der Hilfe des ZX-Teams an den Nachbau des ZeddyNet-Bausteins gemacht.

Ein Standard-16K-Zeddy reicht allerdings nicht aus, um die ZX-Internetfreuden zu genießen. Der Zeddy muss mindestens 48 kRAM und einen USB-Massenspeicherzugriff (z.B. TTL-V-drive-USB) haben und HRG-fähig sein, um alle Internet Tools voll auszukosten.

Hardwarevoraussetzungen

Read/Write RAM von Adresse 8192 bis 49152.

M1NOT-Hardwaremodifikation, die die Ausführung von MC-Programmen oberhalb Adresse 32768 ermöglicht.

HRG-Fähigkeit für das IRC-Chat-Programm.

Batteriegepufferter RAM für den USB-Treiber erleichtert das ZX-Leben und erhöht die Spielfreude an der Maschine.

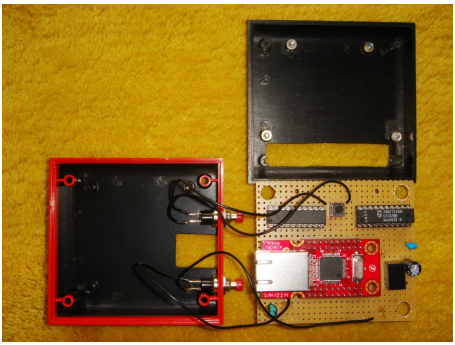
Externe ZeddyNet Karte mit dem „intelligenten“ WIZNet-Baustein D-Wiz-812MJV001 und programmiertem GAL-Chip (16V8D) und noch einigen anderen Standardbauteilen (siehe Pläne unten)

Bauplan für die Zeddy-Net-Karte

Ich habe den Siggi-Zeddynet-Bauplan (ZeddyNet-V4-WIZ812.pdf) mit dem Lochmaster-Programm (Version 3.0) so aufbereitet, dass er in das stilechte „Ferrari“-Gehäuse („Strapu Box“ von Conrad Electronic Nr. 526074) passt. Der ursprüngliche Siggi-Schaltplan wurde von Olli Lange für den TTL-56k-USB-Zeddy weiter vereinfacht.

Paul hat den GAL-IC (16V8D) mit der Steuerlogik für mich gebrannt und

„ganz nebenbei“ haben Siggi und Olli die M1NOT-Hardware-Modifikation für meinen 56kTTL-USB-Zeddy aus dem Ärmel geschüttelt (FE-Transistorschaltung mit 2N7000).



Damit nicht genug, Olli hat den USB-Treiber (VDR.BIN) für meine Maschine so umgeschrieben, dass er im Zeddy HighRAM ab Adresse 47616 läuft. Erst damit war die volle Kompatibilität zu Siggis Internet-Tools (NCLI, NFM, ZeddyFox) auf meiner sehr speziellen TTL-Vdrive-USB-56k-ZX81-Maschine gewährleistet.

Auf zwei Mini-ZXTEAM-Treffen 2012 in Nienburg und Geilenkirchen wurde weiter getestet und getüftelt bis die TTL-Maschine im Internet herumrasen konnte. Nachfolgend möchte ich Teilelisten, Bilder und Schaltpläne sprechen lassen für all diejenigen, die die Zeddy-Net-WizNet-Platine nachbauen wollen. Noch leichter als eine eigene Lochplatine zu löten, geht es mit der professionellen ZeddyNet-Platine von Joachim Merkl.

Danksagung

Siggi Engel, Thomas Lienhard, Joachim Merkl, Olli Lange danke ich für die vielen genialen Ideen, Schalt-pläne und Programme und dem „ZX-Mini-Team Nordwest“ (Paul, Jens, Olli) für wunderbare Treffen in Bremen, Nienburg und Geilenkirchen. Danke Euch allen für die Stunden!

Teileliste Zeddynet Baustein

WIZNet-Baustein: D-Wiz812MJV001

C5,C6: Elko 100µF, mindestens 16V

C7 100nF, der nicht auf dem Siggi-Originalschaltplan vorhanden ist

IC1: GAL 16V8D
(muss programmiert werden siehe Literatur)

IC2: 74HCT245

IC3: TRACO TSR 1-2433
(= Spannungswandler 3,3 V)

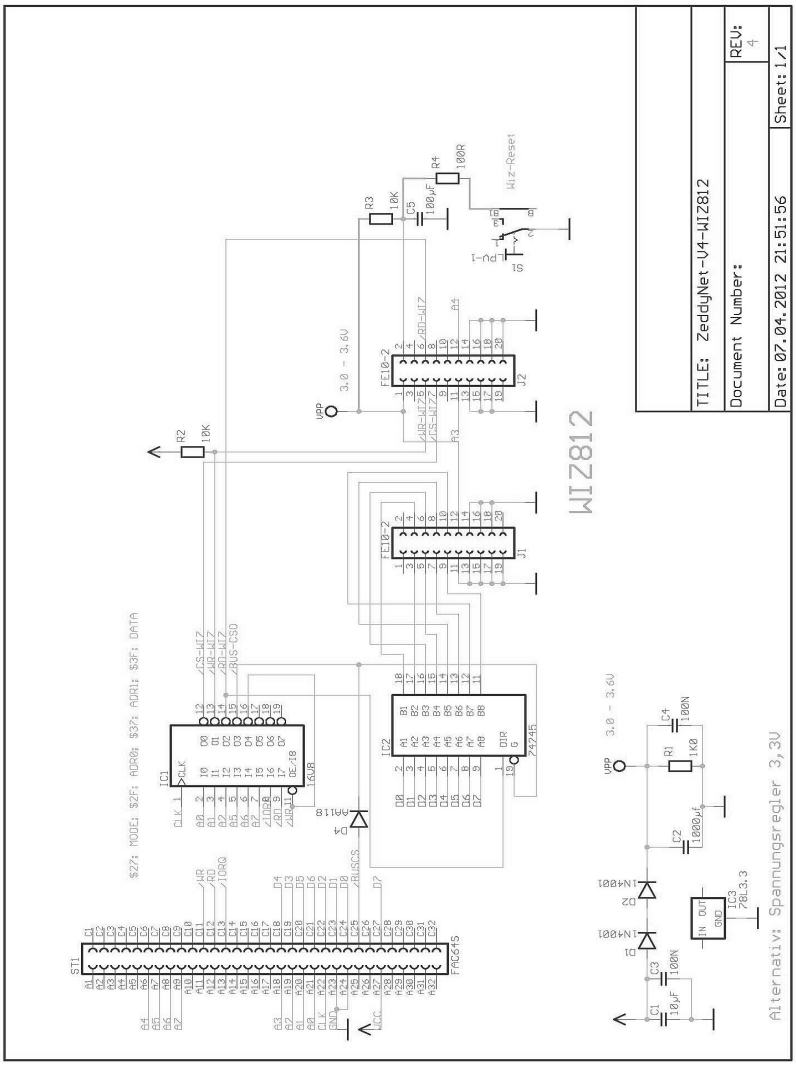
1 klassischer 44-poliger Zeddy-Bus-Stecker mit kurzen Kontakten

2 Pfostenbuchsen: je 20-polig zur Aufnahme des WizNet-Bausteins

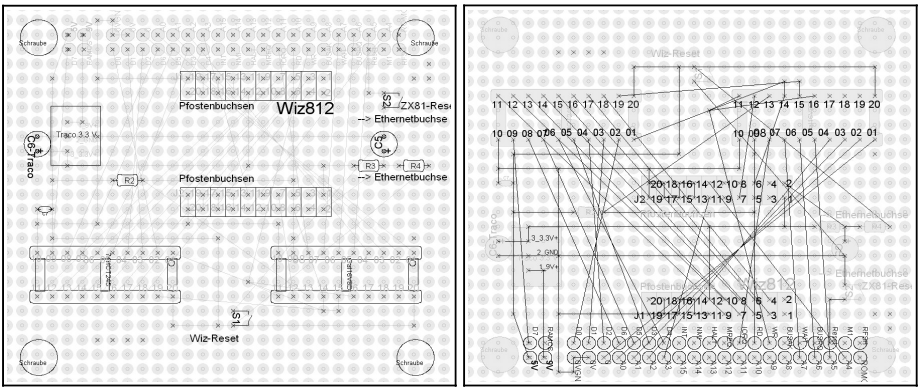
1 rot-schwarzes „Ferrari“-Gehäuse „Strapu-Box“ für die Lochplatine Conrad Electronic: Nr. 526074

2 Reset-Taster

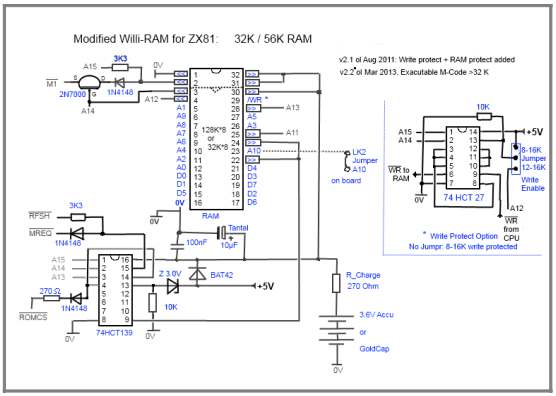
Bitte beachten: Die Dioden D1,D2,D3 und alle anderen auf dem Siggi-Originalplan genannten Bauteile werden nicht für den TTL-Zeddy benötigt.



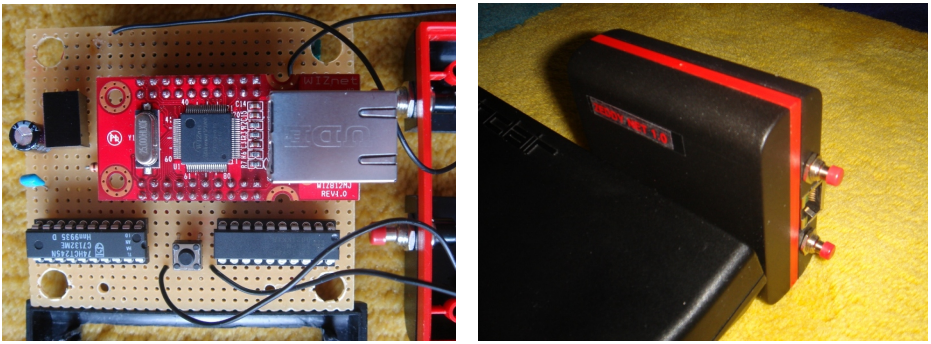
07.04.2012 21:52:22 D:\PRIVAT\ZX81\MEIN-ZX81\ZeddyNet-V4-WIZ812.sch (Sheet: 1/1)



Lochmaster-Lay-Out ZeddyNet: Bauteilseite (Busstecker oben) und Lötseite



Willi-RAM mit „M1NOT"-kompatibler High-RAM Ansteuerung

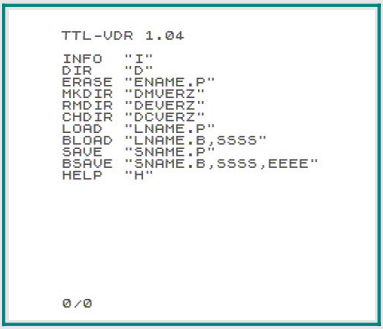
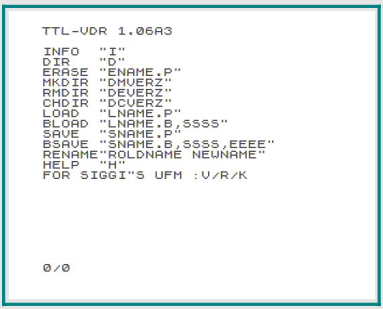
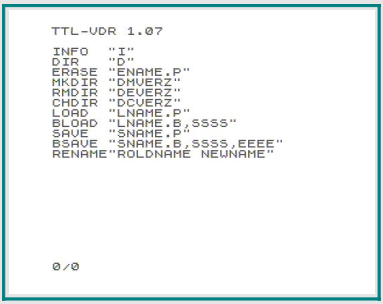


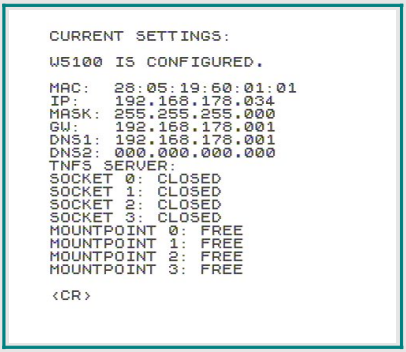
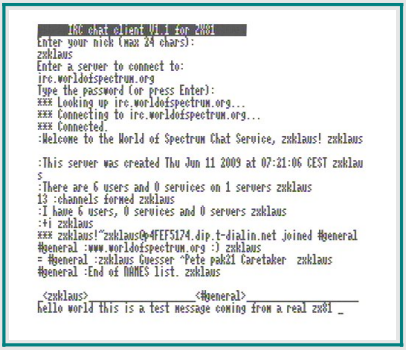
Details zum mechanischen Aufbau

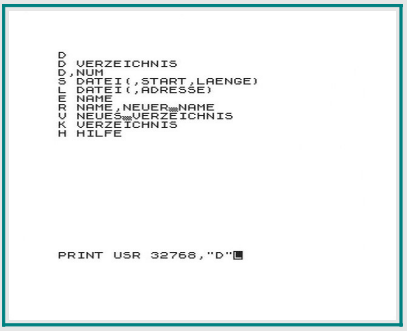
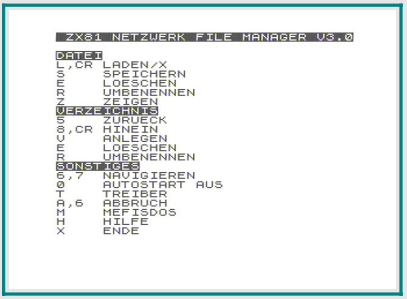
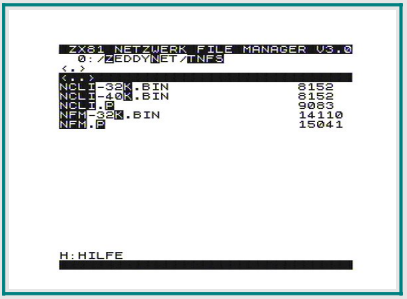
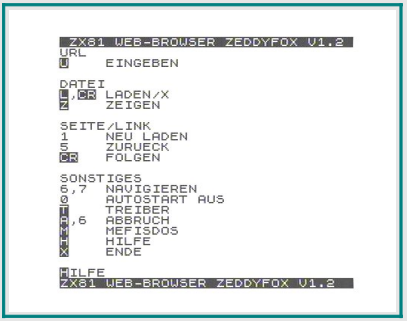
Software

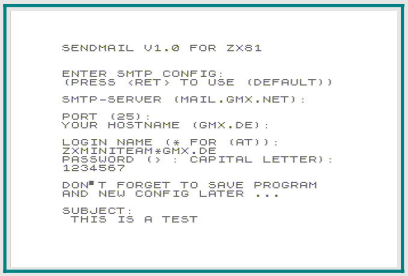
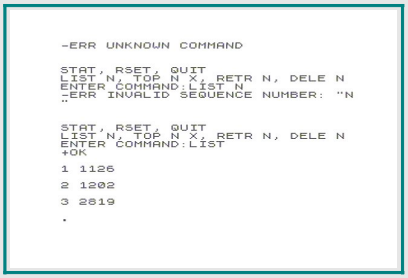
Unentbehrliche Treiber, Konfigurations- und Internet-programme die nötig sind, um den Zeddy ans Netz zu bringen siehe ZeddyNet Softwaretabelle.

Software-Tabelle ZeddyNet (Jahnke/2013-03-04)

Programm (Herkunft)	Kurzbeschreibung	Screen Copies
VDR.P TTL-Version 1.04 (O. Lange 2012)	USB-Treiber für TTL Vdrive befindet sich in REM-Zeile 0 in einem BASIC-Programm wird mit PRINT USR 16514, “<Befehl>” aufgerufen.	
VDR.BIN TTL-Version 1.06A3 (O. Lange 2012)	USB-Treiber für TTL Vdrive wird nach Adresse 8192 (hex2000) kopiert und von dort mit PRINT USR 8192, “<Befehl>” aufgerufen. **Hinweis:** Dieser Treiber ist auch mit Siggis USB-File-Manager UFM-32K.BIN kompatibel.	
VDRBA00.BIN TTL-Version 1.07 (O. Lange 2012)	USB-Treiber für TTL Vdrive wird nach Adresse 47616 (hexBA00) kopiert und von dort mit PRINT USR 47616, “<Befehl>” aufgerufen.	

SYSINFO.P (O. Lange, 2012)	Diagnose- und Testprogramm für das WIZNet und den Zeddy-RAM	
IPCONFIG.P (S. Engel, 2012)	Konfiguration des WIZNet-Bausteins zur Internet-kommunikation. Einfachste Konfiguration: DHCP = ON und TNF-Server IP-Adresse eingeben. Siggis TNF-Server: IP 62.89.167.32 ZX-Spectrum TNF-Server Winston UK IP 213.230.209.153 (enthält nur Spectrum-Programme)	
IRC.P (S. Engel, 2012)	IRC Chatprogramm läuft live im ZX-HRG-Modus. Zugangsdaten Server: irc.worldofspectrum.org Password: 16384 Weitere Kommandos: /join #general … /quit **Hinweis:** Das #-Zeichen wir auf dem Zeddy über SHIFT/SPACE erreicht.	

NCLI-32K.BIN (S. Engel, 2012)	Kommando-gesteuerter Network Client für TNF-Server (Spectrum und ZX81) TNFS = “Tiny Network File Server” Dieses Tool kann aus BASIC-Programmen heraus aufgerufen werden.	
NFM-32K.BIN (Net Work File Manager, S. Engel, 2012)	Network File Manager – Network Client für TNF-Server mit anwender-freundlicher Oberfläche ähnlich dem klassischen “NortonCommander” für DOS von 1986.	
NFM-32K.BIN (NetWorkFile Manager, S. Engel, 2012)	Die aktuellen ZeddyNet Tools auf Siggis TNF Server.	
ZF-32K.BIN (ZeddyFox, S. Engel, 2012)	HTTP/HTML Internet-Browser ZeddyFox wird nach Adresse 32768 geladen mit dem Befehl PRINT USR 47616, “LZF-32K.BIN, 8000”. Bei Benutzung überschreibt ZeddyFox den USB-Treiber ab Adresse 47616. Um auf USB-Stick speichern zu können, muss der USB-Treiber nach WizNet Konfiguration etc. ein zweites Mal nach Adresse 8192 geladen werden.	

SMAIL.P (Sendmail, S. Engel, 2012)	E-Mail Sender- Programm für Mini-Mails bis 1200 Zeichen. Es funktioniert nur mit POP3- bzw. SMTP-Serven mit **unverschlüsseltem** Datenzugang, z.B. dem GMX-Mail-Server und den Standardein-stellungen, die mit <ENTER> bestätigt werden.	
GMAIL.P (Getmail, S. Engel, 2012	E-Mail Empfänger- Programm für Mini-Mails. Es funktioniert nur mit POP3- bzw. SMTP-Serven mit **unverschlüsseltem** Datenzugang (siehe SMAIL.P) Empfehlung: GMX-Mail-Server verwenden.	

Literatur und Referenzen (Auswahl)

S. Engel 2012: Schaltplan ZeddyNet-WizNet und Programmierung GAL 16V8D - http://forum.tlienhard.com/phpBB3/viewtopic.php?f=2&t=273&start=100

S. Engel 2012: Network File Manager für NFM.BIN (Zeddynet V3.0) und die ZeddyNet Tools IPCONFIG.P, NCLI.BIN, IRC.P, SMAIL.P, GMAIL.P, ZF-32K.BIN (ZeddyFox) http://forum.tlienhard.com

TTL-56k-USB-Zeddy mit M1NOT-Schaltung
O. Lange 2012 (siehe aktuellen Schaltplan unten) FE-Transistor 2N7000

Bau eines TTL-USB-56K Zeddies K.-D. Jahnke: ZXTM Sonderausgabe 1, März 2012.

1K-HRG-Spiele: Sprechstunde mit Dr Beep

Im englischen Forum hat Dr Beep über die hohe Kunst des Programmierens von 1K-Spielen mit hochauflösender Grafik geschrieben. Hier die Übersetzung seines Originalbeitrags

Von Dr Beep - vom englischen ins deutsche übertragen durch Paul

Wie programmiert man 1K hires Spiele? Es ist kein kleiner Schritt Spiele zu programmieren wenn man auf eine Maschine mit nur 1K programmierbarem Speicher trifft, und dann noch Code zu schreiben mit dem man auf dieser Maschine HiRes anzeigen kann.

Die Idee dagegen ist revolutionär. Zu zeigen das HiRes auf dem 1K ZX81 möglich ist etwas anderes, als wirklich ein HiRes Spiel in nur 1K zu programmieren.

Also - welche Schritte sind notwendig um ein Spiel in 1K HiRes zu programmieren?

HiRes Bildschirm Modi

Das erste auf das du achten musst ist der Bildschirm den dein Spiel anzeigen soll. Das schwierigste ist es ein Bildschirmmodus zu definieren der nicht nur in 1K passt, sondern auch innerhalb der zur Verfügung stehenden Zeit (207 TStates) dargestellt werden kann. Wenn du ein bestimmtes Spiel kodieren möchtest so musst du dir eine Methode ausdenken wie das Bild ausgibst und wie du den Bildschirm im Speicher anordnest.

Suche nach Möglichkeiten die Größe des Bildschirms zu reduzieren. Einige Beispiele dies zu erreichen sind:

- Nutze einen kleinen Bildschirm. Ein 6x8 Character Bildschirm kostet dich schon 384 Bytes, könnte aber für ein einfaches Spiel ausreichen.

- Definiere nur sichtbare Linien. Falls dein Spiel nur wenige Bildzeilen hat kannst du diese Zeilen mit einem Marker versehen bevor du die Daten ausgibst. Du kannst fast einen vollen Bildschirm mit acht Zeilen in nur 256 Bytes unterbringen.

- Definiere nur sichtbare Zeilen sowie eine Default Zeile die du mehrfach wiederholst. Dies ist fast die gleiche Technik wie zuvor, nur statt leerer Zeilen werden Zeilen mit Daten angezeigt.

- Zeige Datenzeilen mehrfach an. Wenn du Zeilen zwei oder dreimal ausgibst so zeigst du mehr Bildschirm mit weniger Bytes.

- Spiegele Zeilen

Wenn deine UDG's symmetrisch sind brauchst du nur die eine Hälfte speichern und die zweite Hälfte durch spiegeln anzeigen.

Falls dein Spiel eine dieser Regel erfüllt (oder du dir eine andere Methode ausdenkst) dann könnte dein Spiel in 1K

passen.

Speicher-Spar-Tipps

Mit nur kleinem verfügbarem Speicher muss dein Programm klein sein aber funktionieren. Falls möglich musst du so viele Routinen wie möglich zusammen programmieren um Platz zu sparen. Alle meine Tipps folgen hier:

- Nutze deinen Bildschirmspeicher gleichzeitig als Spielfeld.

- Nutze deinen Zeilenpuffer (der Code im oberen Speicherbereich zum Anzeigen einer Zeile) gleichzeitig auch zum Speichern von Daten oder extra Code. Im oberen Speicherbereich werden in der Zeile Werte ausgelassen die Bit7 und/oder Bit6 nicht gesetzt haben. Daher kannst du dort Daten oder ausführbaren Code speichern die diese Bits gesetzt haben.

- Verstecke Befehle in anderen Be-fehlen. Ein relativer Sprung kostet dich zwei Bytes. Aber wenn der auszulassende Befehl nur ein oder zwei Bytes lang ist kann man ihn in anderen Befehlen wie LD HL,NN oder LD A,N verstecken. Dies spart ein Byte.

- Benutze FlagErgebnisse und Werte anderer Register. Wenn du das Ergebnis eines Befehls kennst kannst du es in deinem Programm benutzen. Wenn dein Programm NIEMALS das Carry Bit setzt brauchst du es vor SBC HL,DE nicht zurückzusetzen. Wenn du einen Wert in einem Register benötigst der bereits in einem anderen Register steht, dann lade den Wert in das andere Register.

Wenn du ein Register auf einen bestimmten Inhalt prüfen willst dann benutze XOR Wert. Das spart zwar kein Byte, aber A ist bei Gleichheit 0 was hilfreich sein kann UND das Carryflag wird NIEMALS GESETZT wenn A kleiner als der Wert ist.

- Benutze JR um abschließend einen größeren JP zu machen. Wenn JR nicht ausreicht, du aber an anderer Stelle auch zurück springst, so springe mit JR vor diesen Befehl und von dort dann nach oben.

- Verändere nur das untere Byte eines Registers wenn das höhere Byte nicht verändert werden braucht.

- Benutze selbst modifizierenden Code. Wenn man wenige Bytes ändert kann man eine Routine mehrfach benutzen.

- Versuche deine Daten zu komprimieren und vom komprimierten Modus zu lesen. In einem Labyrinth benötigst du nur zwei Richtungen (z.B. Links und hoch). Rechts und runter kann dann von den Nachbarfeldern gelesen werden.

- Der Aufruf des Maschinencode (die Basic Zeile) darf im Bildschirmspeicher stehen. Es wird nach dem Programmstart nicht mehr benötigt und kann dann überschrieben werden.

- Speichere Daten in den Systemvariablen. Nach dem Laden kannst du die Daten aus dem Bildschirm in die Systemvariablen kopieren. Außer LASTK und FRAMES sind alle Systemvariablen für Daten nutzbar.

- Du kannst deine Initialisierungen im Bildschirmspeicher ablegen und nach der Initialisierung löschen.

- Definiere deine UDG's so, dass ihre Eigenschaften im Code nutzbar sind. Wenn du ein UDG verschiebst und Bit 0 und 7 sind immer gesetzt, dann kannst du feststellen ob das UDG auf ein neues Byte verschoben wird.

- Setze alle zeitkritischen Befehle zwischen 4 und 40 T-States in sowenig Bytes wie möglich um Lücken im Display zu füllen.

- Benutze Befehle im oberen Speicher. . Für Timing und korrekte Bildausgabe kannst du Befehle mit gesetzten Bits 6 und 7 im oberen Speicher verwenden.

Hires Methoden von Spielen

Für alle meine bisherigen Spiele folgt nun eine kurze Beschreibung der verwendeten HiRes Methode. Alle Techniken können verwendet oder auch kombiniert werden um einen Bildschirm zu erzeugen.

Wiwo Dido and the case of Mazeddy's Castle: Dieses Spiel verwendet einen festen Bildschirm von 5x5x8 Bytes. Die Basis des Spiels war der 6x8x8 Bytes Bildschirm vom WRX1K. Die Full-Screen Idee wurde durch das Schieben des sichtbaren Bildes über den ganzen Bildschirm.

Blocky: Ein Bildschirm aus 8x32 Bytes wobei das erste Byte die y-Position angibt gefolgt von 31 Bytes Bildschirmdaten.

Return to Mazeddy's Castle: Ein 6x8x8 Bildschirm wird erst in absteigender Folge und dann in aufsteigender Reihenfolge dadurch wird eine 3D Ansicht erzeugt. Jede Zeile wird drei mal angezeigt um den Bildschirm zu füllen.

Ghost Hunt: Ein 2 Schirm Display mit 16x2 Bytes wie Wiwo Dido und ein 32 Byte Schirm wie Blocky aber mit Zeileninvertierung wenn die Bedinung der Y-Koordinate nicht erfüllt ist.

3D Random maze: Eine HiRes Methode bei der jede Zeile ein paar Bytes während der Darstellung verändert, nach der Hälfte des Displays umgedreht um den 3D Effekt zu erhalten. Von mir nirgends vorher verwendet.

Othello: 8 Zeilen mit 16 Bytes Daten. Jede Zeile wird 14 mal gezeigt. Zwei weitere Zeilen bilden ein Feld oben und unten.

Bowling: Ein 19x10 Bytes Bildschirm wie Blocky mit einer Default Zeile.

Spy vs Spy: Eine Technik wo UDG vor den Bildspeicher im oberen Bereich geladen werden. Zu Lasten von zwei sichtbaren Spalten werden die richtigen UDG geladen und angezeigt. Wo normaler Weise die Bildschirmdaten in einem Zeilenbuffer angezeigt werden, bilden hier 10 Zeilenpuffer die Daten.

Memory: Wie Spy vs Spy, aber mit mehr UDG zu Lasten weiterer Spalten. Dies kombiniert mit einer einzelnen Zeile Daten für einen Cursor wie bei Blocky.

Police Patrol: Wie Othello mit der Ausnahme das das Polizeiauto während der Anzeige in der HiRes Zeile gebildet wird.

Pong: 4 Zeilen für den Ball wie Blocky. Dies verbunden mit einem Test ob links, später auch rechts, ein Schläger angezeigt werden soll. Nur 4x30 Bytes und 2x40 Bytes Pointer für die Schläger.

Saloon City: Die Display Methode von Memory, aber dann mit einem selbst modifizierenden Zeilenpuffer um den Character auf der richtigen Position anzuzeigen.

Meteor Storm: Nur 8x22 Bytes Daten für den Turm und die Meteore und 3x21 Bytes für den Boden. Mit wenigen Zeilenpuffern wird der richtige Bildschirm während jedem Teil des Bildschirms angezeigt.

Car Race: Wie Bowling 3x8x8 Bytes werden für den Bildschirm und die 3 Autos benutzt. Besonderheit am Display ist die veränderliche Start Position innerhalb des Bildschirmpuffers und dann wieder zum Anfang bei Bedarf.

Outlaw: Die gleiche Basisroutine wie in Pong, aber anstelle eines festen UDG (der Schläger) wird hier unterschiedliche Daten (der Cowboy) angezeigt.

Sokob-One: Ein 7x4x8 Bytes Bildschirm wobei vier Zeilen angezeigt werden und dann wieder in umgekehrter Reihenfolge. Durch die symmetrische Form des UDG ist dies möglich und reduziert den Bildschirm um den Faktor zwei.

Pinball: Die Methode der Anzeige ist gleich mit Bowling, aber 2 Bildschirme werden bei der Anzeige zu einem kombiniert (merge). Beide Bildschirme habe die Technik von Bowling.

Fazit

Während man mit einem 6144 Bytes Bildschirmpuffer eine vordefinierte Anzeigemethode für alle Spiele verwendet werden kann, ist dies bei 1K HiRes Spielen nicht so.

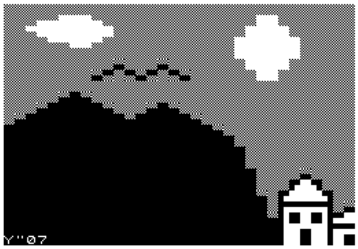
Jedes Spiel benötigt eigene Überlegungen für eine spezielle HiRes Anzeige Methode. Falls du planst den gesamten Bildschirm mit nur wenigen Zeilen Daten verglichen zur Gesamtgröße zu füllen ist die Methode mit der Y-Position und Anpassungen an den Daten ein guter Start. In 8 meiner erwähnten 17 Spiele findet diese Methode (angepasst an den Bedarf) Anwendung.

Auch doppelte oder dreifache Ausgabe der Zeilen kann einen besseren oder größeren Bildschirm bringen. Dies wird in 9 der 17 Spiele so gemacht. Die Bewegung wird dadurch allerdings weniger flüssig mit Sprüngen um 2 oder mehr Pixel.

Grafikwettbewerb für den ZX81

Im Juli 2007 startete Yerzmyey, ein polnischer Computerfreund, eine Aktion, um andere ZX81-User zu motivieren, sich wieder einmal mit ihrem Oldie zu befassen.
Auf ANDRE***s Poster als Vorbild verweisend rief er einen Grafikwettbewerb aus. Als Beispiel stellte er gleich selbst ein Bild ins Netz, das allein aus Zeichen des Standardzeichensatzes des "Türstoppers" zusammengesetzt ist. Obwohl es nicht als erster Beitrag gedacht war, vergaben auch hierfür User Punkte und es erreichte punktgleich (155) mit einem anderen Bild den 9. Platz.




MOUNTAINS by Yerzmyey/H-PRG 155

Ende September lief die Frist zur Punktevergabe für die eingereichten Beiträge ab und Anfang Oktober stand in Thomas Lienhards ZX81-Forum der link zur Auswertung.

Ein einziges Bild wurde in hochauflösender Grafik erstellt, dieses lief jedoch außer Konkurrenz, wurde aber mit veröffentlicht, um zu demonstrieren, daß der ZX81 auch zu HRG-Fähigkeiten gebracht werden kann.
Hier also auch für die Leser des ZX-Team-Magazins die Grafiken und Plazierungen:


out of the compo:
STAFFORD by XavSnap


1 AMYACKER by Russell Marks 299



2 CLIVE by Factor6 279



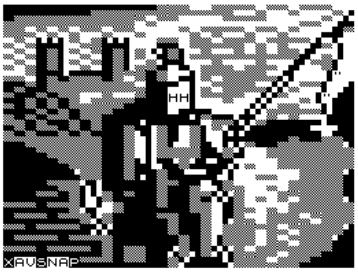
3 SPOCK by Russell Marks 227



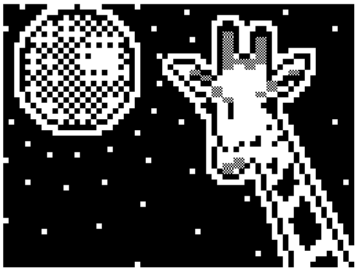
4 YUI-NYAN by Gasman/H-PRG 221



5 LAUREL by Andre 216



6 KNIGHT by XavSnap 198



7 DISCO STAR by DiamonDie 187



8 FACE by XavSnap 161



9 S7 by Russell Marks 155



10 ODIE by Thomas Lienhard 134

Thomas Rademacher · Januar 2008

PSYCHOLOGIE AKTUELL

Neues vonne Küste

Auf Umwegen erreichte uns der folgende Beitrag. Obwohl darin das schlimme A*-Wort benutzt wird, haben uns dennoch zur Veröffentlichung entschieden - als eindringliche Mahnung, wohin die übermäßige Beschäftigung mit nicht-Sinclair-Computern führen kann. Wir danken Willi für seinen offenen Umgang mit dem Thema und sichern ihm unsere mitfühlende Unterstützung zu.



Das Archivbild lässt die innere Zerrissenheit des bedauernswerten A-Fans erahnen*

Aus dem Atari-Forum,

Thema: Kfz.-Kennzeichen:

Liebe Freunde, ich möchte mich zu dem Thema auch melden. Wer denkt eigentlich über die ethnische Minderheit nach, in deren Brust zwei Herzen schlagen?? Ich bin seit Jahren



Fan von Sinclair-Rechnern und ATARI. Bei der Zulassungsstelle habe ich Hausverbot weil ich 800 XL durchsetzen wollte, mit Betteln, mit Bitten und auch mit Drohungen. Die sagen, das geht nicht, Zahlen und wieder Buchstaben. Sehr schweren Herzens (ich leide!) habe ich mich dann für den legendären ZX80 von Sinclair entschieden und ATARI mit einer Sonderlackierung bedacht. Ich muß nun Schelte von zwei Seiten ertragen. Bitte, seid gnädig mit mir!

Willi



P.S. Die Atari-Label sind Magnettafeln!! Die Kennzeichen echt!!

sinclair ZX80

Microcomputer.

Der programmierte Erfolg für Studium, Beruf und Freizeit.

NEU! FÜR NUR 498.– DM

Computer prägen in zunehmendem Maße nahezu alle Bereiche des täglichen Lebens. Und wer im Studium oder Beruf Erfolg haben will, muß sich mit ihrer Sprache und Bedienung vertraut machen.
Sinclair ZX 80 – der einfachste Weg zur Computertechnik. Der ZX 80 Microcomputer verarbeitet „Basic", die am weitesten verbreitete Computersprache. Seine Speicherkapazität, wichtig für universellen Einsatz, bietet schon in der Standardausführung 1 k-Byte RAM und 4 k-Byte ROM. Kombiniert mit einem einzigartigen Lernprogramm, speziell auch für den Anfänger entwickelt, führt Sie der ZX 80 Schritt für Schritt in das Gebiet der Datenverarbeitung ein. Systematisch. Gründlich. Und unkompliziert.
Das ZX 80 Basic-Handbuch. Die Grundlage Ihres programmierten Erfolges. Denn jedem theoretischen Kapitel folgt stets eine praktische Lektion. Von der Einführung bis zu schwierigen Programmen. Ein kompletter Basic-Kurs.
System ZX 80. Leicht zu bedienen. Alles, was Sie tun müssen, ist, den kompakten ZX 80 Microcomputer (nur 174 mm x 218 mm) an Ihren handelsüblichen Fernseher (UHF, Kanal 36) anzuschließen und mit einem ganz normalen Cassettenrecorder zu verbinden.

Peek und Poke ermöglichen die Eingabe von Maschinencode-Instruktionen. Der USR-Befehl bewirkt den Sprung zu einem in Maschinensprache geschriebenen Unterprogramm. Das Gerät bewältigt bis zu 26 Strings jeder Länge, die sich alle in Beziehung zueinander setzen lassen. Der einmalige Syntax-Check sorgt für absolut fehlerfreie Programmeingabe. Kein Wunder also, daß führende Fachzeitschriften dem Sinclair ZX 80 Microcomputer sehr gute Preis-/Qualitätsrelation, universelle Verwendbarkeit und hohen Leistungsstand bescheinigen. Wir wünschen viel Spaß beim Programmieren!



Das ZX 80 Basic-Handbuch.



Fertig ist Ihr persönliches Computer-Terminal. Sämtliche Verbindungskabel inkl. Netzgerät werden mitgeliefert. Das Arbeiten mit dem ZX 80 ist denkbar einfach. Zum Beispiel durch seine übersichtliche Tipptastatur. Sinnvolle Basic-Abkürzungen. Und besonders leistungsfähige, neue „LSI-Microchips" bzw. „Super-ROMS"! So ist der ZX 80 ein ebenso intelligenter, zuverlässiger Geschäfts- oder auch Schachpartner.
Ausbaufähig für Könner. Die Speicherkapazität läßt sich mit dem neuen „RAM-Modul" bis auf 16 k-Bytes erweitern. Ebenfalls interessant für Experten: Der Sinclair ZX 80 ist einer der schnellsten Basic-Computer der Welt.



System ZX 80. Leicht zu bedienen.

sinclair ZX80 EL 4/81

Science of Cambridge Ltd./Deutschland
Erlenweg 2, Postfach 17 10, 8028 Taufkirchen b. München
Telefon (0 89) 6121 793

Bitte senden Sie mir _____ Exemplar(e) ZX 80 Microcomputer (à DM 498,–) inkl. Zubehör.
und _____ Exemplar(e) 16 k-Byte RAM-Erweiterungsmodul (à DM 249,–).
Preise inkl. MwSt., Porto und Verpackung (6 Monate Garantie).
Summe insgesamt DM _____ Versand per Nachnahme oder Scheck im voraus.

Name _____
Straße _____ PLZ/Ort _____
Unterschrift _____ Datum _____

LED ANSTEUERUNG

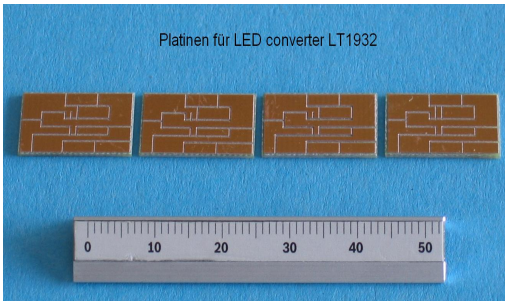
LEDs nicht nur im Hintergrund

Von Ulrich

Im Zusammenhang mit dem ZX2000 wurde die Hintergrundbeleuchtung des LC-Displays zum Thema im Forum. Der Einsatz von weißen Leuchtdioden ist mehr als nur der Ersatz einer defekten Leuchtröhre. Abhängig von Displaygröße, Umgebungslicht und persönlichem Empfinden sind dafür ca. 6-10 Leuchtdioden zur Flächenausleuchtung erforderlich. Dabei kann die Ansteuerung auf unterschiedliche Art erfolgen, je nach verfügbarer Spannung können die LEDs parallel oder seriell betrieben werden. Bei einer Betriebsspannung von 5 Volt ist die Parallelschaltung einfach zu realisieren, jede Leuchtdiode bekommt ihren eigenen Vorwiderstand, mit 120 Ohm stellt sich ein Strom von ca. 15mA ein. Hat man 12V zur Verfügung, kann man maximal drei weiße Leuchtdioden in Reihe mit einem Vorwiderstand (160 Ohm) betreiben. Je nach Ausleuchtungsbedarf werden dann mehrere Gruppen gebildet und parallel versorgt. Der Einsatz eines Konverters ermöglicht höhere Spannungen, 8-10 weiße LEDs können damit seriell betrieben werden. In Verbindung mit einer Konstantstromquelle (z.B. mit einem LM317 als Stromquelle nachgeschaltet) wird auch die Anzahl der Vorwiderstände reduziert .

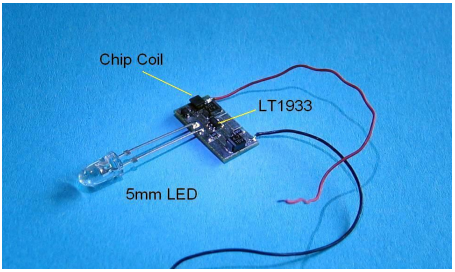
Die Produktpalette der Firma Linear Technology hat ein SMD-IC im Programm, das offensichtlich für den Handy-Markt geschaffen, sich hervorragend für die weißen LEDs eignet. Mit der Bezeichnung LT1932 ist ein Constant-Current DC/DC Baustein verfügbar, 6-beinig im SOT-23 Gehäuse. Das 16-seitige Datenblatt [1] zeigt verschiedene Anwendungsbeispiele, bis zu zehn weiße Leuchtdioden sind damit ansteuerbar. Ergänzend kann auch die Helligkeit variiert werden. Nicht irritieren lassen sollte man sich von der dort wiederholt angegebenen Betriebsspannung von 4.2 Volt, die absolute (!) max. Eingangsspannung beträgt 10 V.

Neugierig geworden, schließlich gibt es auch “vordergründige” Beleuchtungswünsche, habe ich mich an die praktische Erprobung herangewagt. Zunächst wurden Layout und Maßskizze erstellt. Nach diesen Vorlagen entstanden dann die Platinen im “Rapid Prototyping Verfahren” ☺, so wie ich es schon häufiger praktiziert habe [2].



Miniplatinen, hergestellt im „Handkurbelverfahren“

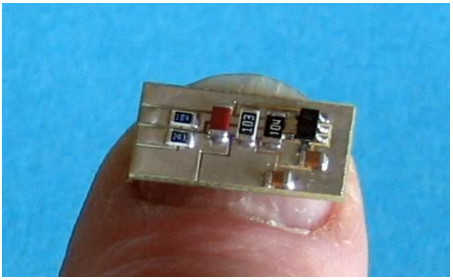
Ein Aufbau mit einer weißen Leuchtdiode sollte zeigen, dass bereits eine Eingangsspannung von 1V dafür ausreicht. Eine Mignonzelle, die in einer Uhr ihren Dienst versagt hatte, brachte dann tatsächlich den erleuchtenden Beweis!



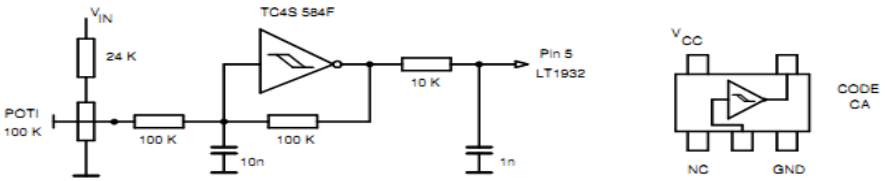
Als nächstes habe ich einen weiteren Aufbau für die Ansteuerung von 10 weißen LEDs mit einer Betriebsspannung von 5Volt erstellt. Die LEDs werden in zwei Gruppen aufgeteilt, jeweils mit 15mA bestromt (Datenblatt, Applikation auf S.16). Die beiden zusätzlichen Widerstände (RV=100 Ohm) sind auch auf der Platine zu finden (diese werden z.B. bei bis zu 8 LEDs durch 0 Ohm Widerstände ersetzt).

Auch dieser Aufbau funktionierte problemlos, die Stromaufnahme beträgt bei 5V Versorgung 140 mA. Die grell leuchtenden Dioden ließen aber sogleich den Wunsch nach einer Dimm-Möglichkeit aufkommen. Das Datenblatt liefert auch hierzu Anregungen, ich habe mich für eine Pulsweiten-Modulation am Shut-Down Pin (5) entschieden. Der Vorteil, der Stromfluß bleibt auch bei abgeschwächter Helligkeit konstant, die LEDs erstrahlen weiterhin im reinen Weiß ohne blaustichig zu wirken.

Der Pulsweiten-Modulator



Ein Impulsgenerator einfachster Bauart ist nun auf einer gleichgroßen Platine aufgebaut. Das variable Pulsbreitensignal wird dem LT1932 am Pin 5 zugeführt. Ein Schmitt-Trigger, aus der ‘Single-Gate-Logik’ Familie, bildet das Herzstück. Über das Poti wird die Triggerschwelle beeinflußt und damit das Puls-Pausenverhältnis.



PULS - WEITEN - MODULATOR

Die sich dabei ergebende Frequenzänderung ist subjektiv nicht wahrnehmbar. Das Lautstärkepoti aus einem DVD-Laufwerk ist aufgrund der logarithmischen Kennline nicht optimal zum Dimmen, evtl. mal die äusseren Potianschlüsse vertauschen.



Aus DVD-Playern, als Dimmregler weniger geeignet, aber schön klein!

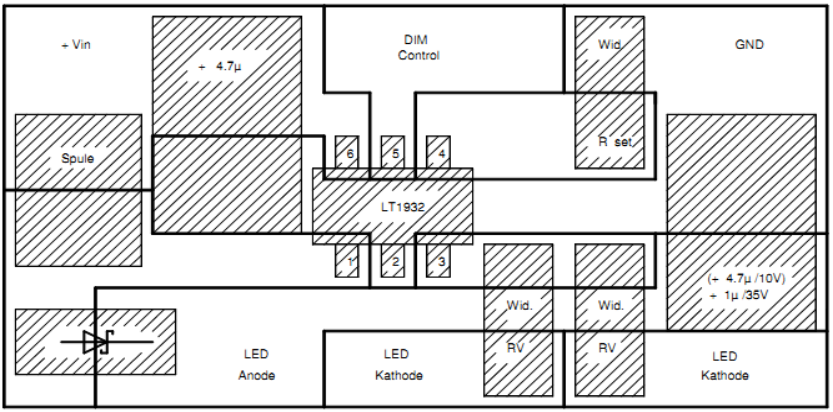
Beide Platinen können sowohl 'Rücken an Rücken' oder mit der Bauteilseite zueinander zusammengefügt werden. In beiden Fällen erhält man ein Modul mit den Abmessungen von 17.5 x 9.5 mm, mit einer Bauhöhe von knapp 5mm. Mit den Bauteilen auf der Innenseite hat man isolierte Außenflächen, andernfalls kann man auch die komplette Schaltung in einen Schrumpfschlauch einschrumpft, beliebig 'hinkleben'. Natürlich können die beiden Platinen auch bei halbierter Bauhöhe, z.B. mit doppelseitigen Tesa o.ä., nebeneinander positioniert werden. Das Dimmpoti wird über flexible Leitungen an der PWM-Platine angelötet und kann somit bedienungsfreundlich im Gesamtaufbau integriert werden.

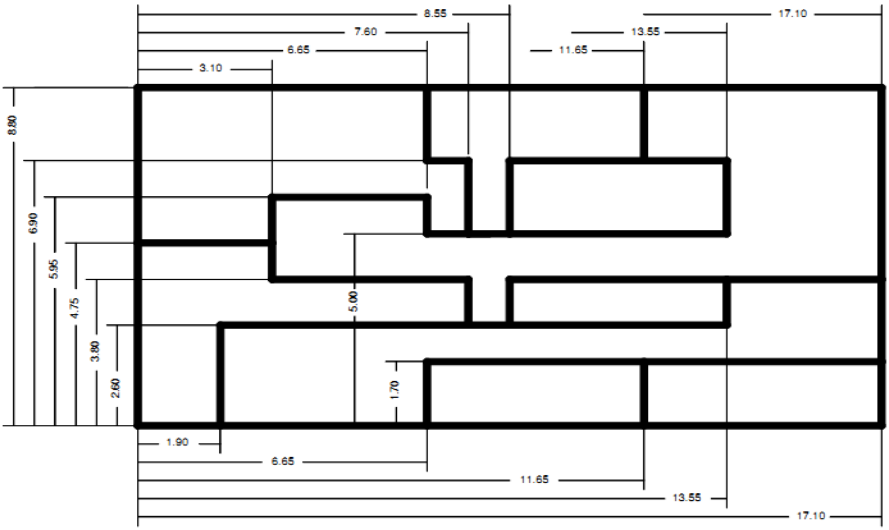
Wichtig: Der Betrieb darf nicht ohne angeschlossene LEDs erfolgen!!! Falls erforderlich, kann eine Zenerdiode (27V), angeschlossen zwischen Kathode D1 und Pin 3 des LT1932 zum Schutz bei offenem Ausgang, eingefügt werden, siehe Seite 7, Fig. 4 im Datenblatt.

Hinweise zum Nachbau

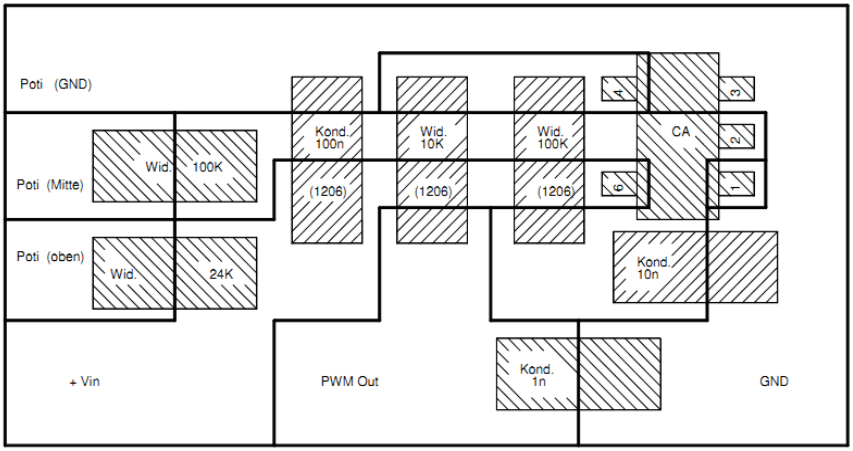
Bei meinen Aufbauten sind die SMD-Widerstände und –Kondensatoren überwiegend recycelt. Auch der SMD Schmitt-Trigger, (Toshiba, 1 Gatter Logik, Stempelung: EA entspricht der 74 HC Serie) stammt von Laufwerksplatinen. Bei Verwendung der CA-Version (Serie CMOS 4000, leichter erhältlich) muß evtl. der 24K Widerstand angepaßt werden, um den Regelbereich (= Drehwinkel am Poti), zu optimieren. Die beiden Tantal-Elkos in meinem Aufbau (1µ und 4.7µ) sollten durch keramische Ausführungen ersetzt werden (Vorteil: Kleinere Bauform (= reduzierte Bauhöhe), bessere Spannungsfestigkeit sowie polaritätsunabhängig.

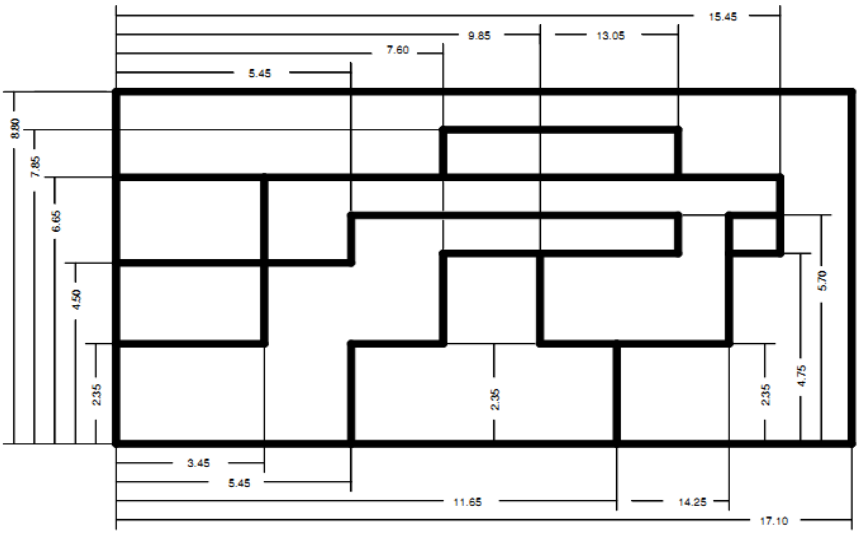
LED CONVERTER





LED PW-MODULATOR





Für Nachbauinteressenten, hier eine Liste für die erforderlichen Bauteile:

Reichelt	
LT 1932 ES6	(Constant-Current DC/DC LED Driver in Thin SOT)
LQH3C 10µ	MURATA Chip-Coil (in meinen Aufbauten verwendet)
X7R-G1210 1,0	(C1) MURATA High Caps 1µF
X7R-G1210 4,7	(C2) MURATA High Caps 4.7µF
NP0- G0805 1,0N	(PWM)
X7R- G0805 10N	(PWM)
X7R- G1206 100N	(PWM)
SMD- 0805 00	(Brücke zu Pin 3 v. LED-Kathode)
SMD- 0805 100	(2 Stck, S.16 Datenblatt)
SMD- 0805 xxx	(Rset, bestimmt den LEDstrom, Datenblatt s. Seite 3)
SMD 1/4W 24K	(s. Hinweise zum Nachbau)
SMD 1/4W 100K	(Poti, Schleifer)
SMD 1/4W 10K	(**CA** Pin 4)
SMD 1/4W 100K	(**CA** Pin 4)
BEL 160X100-1-8	(Platinenmaterial 0.8mm dick)
CONRAD	
Best.-Nr.: 171743-62	1Gatter CMOS Toshiba TC4S548F (SMD-Code: **CA**)
ZETEX- DIODE	Die Schottky-Diode **ZHCS400** mit besonders niedriger Schwell-spannung, bis 400 mA, und kleinster Bauform gibt es bei Farnell, Privatleute können sie bei [3] bestellen

Allen Nachbauern viel Erfolg!

Ulrich

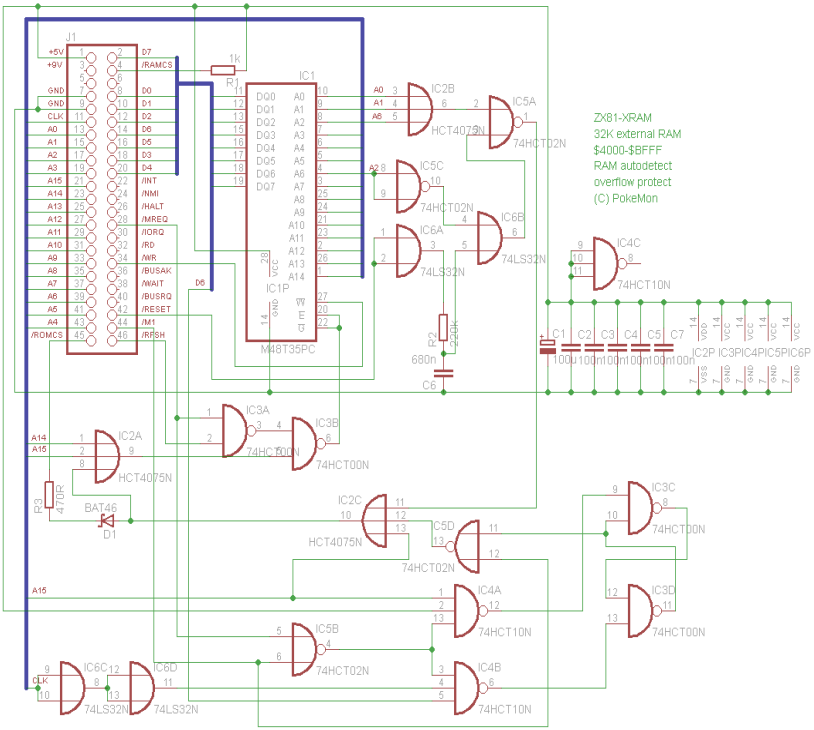
Quellen und Weblinks

[1] Datenblatt LT1932: http://cds.linear.com/docs/en/datasheet/1932f.pdf
[2] ZX-Team Magazin, Bericht: VDRIVE2 Nachlese
[3] Link: https://hbe-shop.de/DIODE-SCHOTTKYSOD-323-Typ-ZHCS400

ZX81XRAM

ZX81XRAM – externes RAM für ZX81

Von Karl-Heinz Dahlmann



RAM Erweiterungen gibt es reichliche Versionen für den ZX81. Nach einigen Recherchen im ZX Team Magazin der vergangenen Jahre habe ich mich mit allen Speicherlösungen beschäftigt. Der eigentlich Grund war der Ausfall meiner 16k RAM Erweiterung (Timex 1016), nachdem sie versetzt auf den Slotstecker aufgesteckt wurde, was zwar der Zeddy überlebte, nicht aber die Speichererweiterung.

Ziel war eine sehr einfache Speichererweiterung mit wenig Aufwand aber extern aufsteckbar wie das Sinclair RAM. Von den Warnungen aufgrund von Wobbles ließ ich mich nicht beirren, da diese nach meinen Recherchen nicht an der Steckverbindung über den Slotstecker liegen. Mit wenig Beschaltungsaufwand läßt sich eine 32k RAM Erweiterung für einen durchgehenden Speicher von $4000-$BFFF recht einfach realisieren. Ich habe mich aber dennoch für nachfolgende Zusatzfeatures entschieden, die den Schaltungsaufwand etwas erhöht haben und im

Folgenden erläutert werden:

=> Kein Öffnen des ZX81 notwendig (externes Modul)
=> Austauschplatine passend für das 16k Sinclair Gehäuse
=> HRG kompatibel
=> RAM autodetect
=> Video overflow protect

Die internen Umbauanleitungen für die Erweiterung des RAMs, Auftrennen von Leiterbahnen und Luftverkabelungen fand ich nicht zufriedenstellend. Eine Erweiterung zum Aufstecken war daher das Ziel, genauso wie die Sinclair oder Memotech Erweiterungen. Außerdem sollte die Platine exakt in das Gehäuse des 16k Block von Sinclair passen, damit man defekte RAM Erweiterungen mit der Austauschplatine fit machen kann inklusive Erhöhung des RAM auf 32k und den Zusatzfeatures. Zusätzlich habe ich die Schaltung zwecks einfachen Nachbaus mit leicht erhältlichen Bauteilen im DIL Format konzipiert, die auch von weniger Geübten gelötet werden können.

Das Schaltbild zeigt die komplette RAM Erweiterung. Die Aktivierung des RAM erfolgt wenn entweder A14 oder A15 high sind, also im Grunde von $4000-$FFFF wobei der Bereich $4000-$7FFF auf $C000-$FFFF gespiegelt ist, notwendig für die Videoausgabe. Normalerweise ist das ROM ebenfalls auf $8000 gespiegelt, was jedoch durch IC2c unterdrückt wird, wenn A15 auf high ist. Der Widerstand R3 ist nicht zwingend notwendig, reduziert aber die Belastung der ULA zusammen mit D1 auf maximal 3,5 mA, durchschnittlich etwa 2 mA. Das RAM wird sowohl mit /MREQ aktiviert als auch mit /RFSH, damit auch die meisten HRG Programme damit laufen.

Der nächste Schaltungstrick ist etwas komplizierter. Der ZX81 macht nach dem Einschalten oder einen Reset nur einen Speichertest von $4000 bis $7FFF und kann RAM größer als 16k nicht selbst erkennen. Klassisch teilt man dem System das durch POKE 16389,192 mit (für 32k) und macht anschließend ein NEW. Diesen Part wollte ich automatisieren und entwickelte eine Lösung, die automatisch einen RAM Test von $4000 bis $FFFF durchführt. Dabei wird das /ROMCS bei Auftreten der Adresse $0004 kurzzeitig deaktiviert. Der zweite Befehl nach dem Reset ist LD BC,$7FFF und in Byte 4 wird der Wert für das B-Register ($7F) gelesen. Durch das Deaktivieren des ROM greifen hier die Pullup Widerstände auf dem Datenbus, die dann anstelle des $7F ein $FF liefern und in der Folge wird der Speichertest bis $FFFF durchgeführt. Um den Schaltungsaufwand zu minimieren, wurde mit dem Adresskomparator IC5 realisiert, der nur 4 Bits benötigt und schlicht A0,A1,A2 und A6 einbezieht.

```
0000: [0000] D3 FD           L0000: OUT   ($FD),A
0002: [0002] 01 FF 7F               LD    BC,$7FFF
0005: [0005] C3 CB 03               JP    L03CB
```

Während des langen RAM Tests ist A6 auf high.

Der RAM Test selbst dauert etwa 10us (32 Takte) für das Füllen und 26us (84 Takte) für das Prüfen des RAM, was bei 32k und Test bis $FFFF etwa 1,3 Sekunden ausmacht. Hier wird das Reset Signal in Verbindung mit der langen Zeitspanne benutzt, um das ROM nur kurz nach einem Reset mit Adresse 4 (und A6 != 1) zu deaktivieren. Problematisch ist dabei den Reset zu erkennen, da es keine Reset Schaltung gibt und der Zeitpunkt der Erkennung des Reset (=1) der CPU nicht genau vorhergesagt werden kann. Die RC Kombination 220k und 1uF hat ein Tau von 220 Millisekunden. Dadurch bewirkt eine Differenz bei der Erkennung des high Pegels auf Reset gegenüber anderen TTL Bausteinen von 100 mV die CPU bereits etwa 4000 Befehle (in etwa 7 Millisekunden) ausgeführt haben kann. R2 und C6 verzögern daher das erkannte Reset Signal um etwa 70 bis 80 Millisekunden (was einem Anstieg von etwa 1 Volt entspricht), so dass ganz sicher der Startwert in das BC Register geladen wurde aber auf jeden Fall der Speichertest noch nicht beendet wurde.

Ebenfalls tricky war die Video overflow protection. In den Dokumentationen finden sich Hinweise darauf, dass es problematisch ist, wenn die Speichergrenze von $8000 mitten im DFILE liegt. Wenn das DFILE auf Adresse $7FF0 beginnt, wird der Videocode mit Setzen des Bit 15 ausgeführt, also mit Adresse $FFF0. Hier gibt es dann nach 16 Zeichen einen Überlauf von $FFFF auf Adresse $0000 was die folgenden Zeichen im ROM ausführen würde und die Bildausgabe durcheinander bringt. Um echte 32k BASIC Programme ohne Rücksicht auf die Lage des DFILE zu entwickeln wollte ich daher diese Hürde umschiffen.

Technisch erfolgt das mit einem Flipflop, gebildet aus IC3c und IC3d, welches bei Ausführen von Videocode gesetzt wird (/M1=low UND /MREQ =low UND A15=1) und beim Ausführen eines Befehls mit gesetztem Bit 6 (D6=high) als Ende einer Videozeile zurückgesetzt wird. Bei gesetztem Flipflop wird über IC2a das RAM aktiviert auch wenn A15 nicht (mehr) gesetzt ist und das ROM über IC2c deaktiviert (hier beim Überlauf des Adresszählers wichtig). Schaltungstechnisch wird hier das /RFSH Signal benutzt anstelle des /M1 weil die Abarbeitung der Videocodes immer in M1 Zyklen erfolgt und das Flipflop nur beim Ausführen eines Befehls mit gesetztem Bit 6 deaktiviert werden soll (z.B. bei HALT), nicht aber wenn ein Char aus dem ROM mit gesetztem Bit 6 gelesen wird.

Das Modul und auch fertige Platinen für Nachbauer werden voraussichtlich auf dem ZX Team Treffen im April 2013 verfügbar sein.

Karl-Heinz Dahlmann, München

WERBUNG



Wer die neuste Ausgabe des genialen MONOCHROME Magazins von YERZMEY noch nicht hat, sollte schnellstens im Internet nach dem PDF-Dokument fischen.

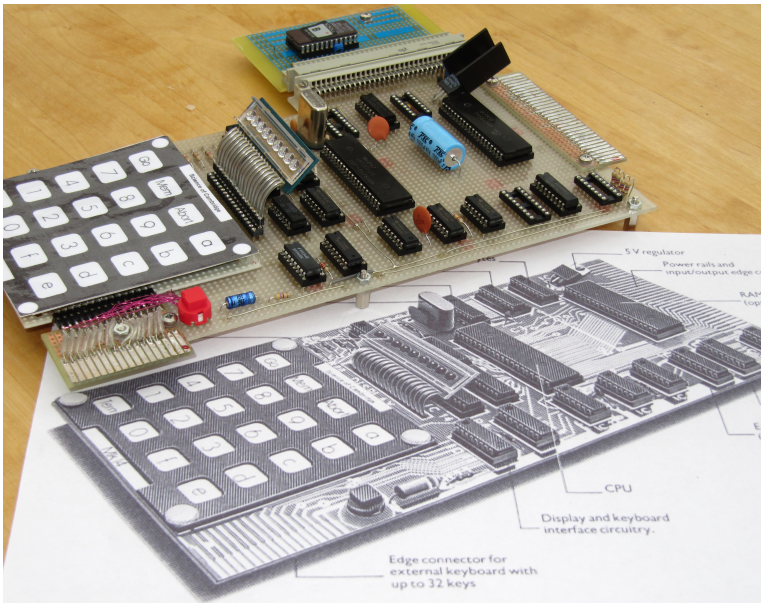
Als passende Hintergrundmusik zur Lektüre gibt es dort auch YERZMEYs ebenso geniale Eigenkompositionen, teilweise eingespielt auf einem original ZX81 mit 6-Kanal-Synthesizer. (http://yerzmyey.i-demo.pl/)

SINCLAIRS MK14

Ahnenforschung – Sinclairs MK14

Als erster Computer aus dem Hause Sinclair wird in verschiedenen Quellen der unter „Science of Cambridge“ vertriebene MK14-Bausatz beschrieben. Ein Nachbau mit alten Originalbauteilen als Reise zu den Wurzeln unserer ZXe.

Von Oliver



Ein halbdokumentarischer BBC-Film über die Heimcomputer-Ära der 80er Jahre macht den MK14 zum Ausgangspunkt der folgenden rasanten Entwicklung [1]: Mitte der 70er Jahre arbeitete *Chris Curry* für Sinclair und brachte in Eigenregie das MK14-Projekt ans Leben. Erst als Curry später dort gekündigt und die eigene Firma *Acorn* gegründet hatte, erwachte im „Analog-Elektroniker“ Clive Sinclair der Enthusiasmus für Mikrocomputer, und das ZX80-Projekt wurde gestartet.

Vom MK14-Bausatz wurden etwa 20000 Exemplare verkauft [2], die meisten davon auf den britischen Inseln, und sehr viele endeten später wohl ausgeschlachtet in Bastelkisten und im Schrott. Entsprechend selten sind überlebende Exemplare heute. Aber es gibt ja noch Schaltpläne und Stücklisten, sowie Lötkolben und Bastler-Ehrgeiz.

Bauteilbeschaffung

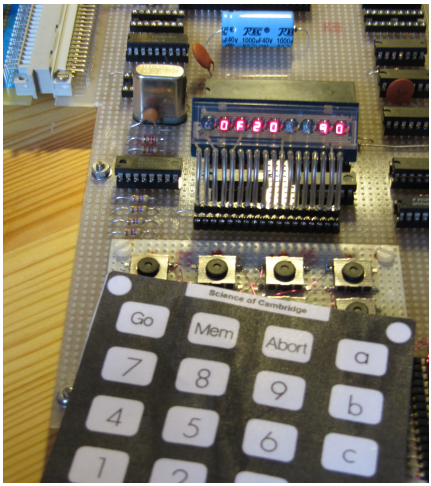
Echte Spezial-ICs (wie etwa die ULA des ZX81) hat ein MK14 nicht. Einige der eingesetzten Bauteile sind aber

heute sehr selten. Für den Mikroprozessor vom Typ SC/MP-II oder das LED-Display muss man heutzutage inflationsbereinigt wohl wieder so viel wie damals zahlen (je etwa 20 Euros), wenn einem nicht irgendwo ein Schnäppchen gelingt. Die restlichen Bauteile sind mit etwas geduldiger Sucherei aber sehr günstig zu bekommen.

Schwierig wurde es bei den eingesetzten 74S751 PROMS, die das Monitor-Programm halten. Alte PROMs sind durch ihre nicht reversible Programmierbarkeit eine endliche Ressource. Ich habe zwar einige unprogrammierte 74S571 des Herstellers Tesla sehr günstig ergattert, dafür ein Programmiergerät aufzutreiben oder selber zu bauen ist aber ein eigenes Projekt. So musste erst mal ein kleines 2732-EPROM herhalten und die PROM-Sockel bleiben vorerst unbestückt.

Bestandsaufnahme – die Hardware

Auffällig ist, dass sämtliche Halbleiter-Bauteile von einem einzigen Hersteller, nämlich von „National Semiconductor“ stammen. Das hat mit der Entstehungsgeschichte des NK14 zu tun. Die Schaltung ist in enger Kooperation mit National entstanden. Oder anders ausgedrückt – wurde nahezu komplett von einem National-Entwicklungskit übernommen [3]. Bei Sinclair hatte man durch die Herstellung von Multimetern und Taschenrechnern sicherlich schon Erfahrung mit den Siebensegment-Anzeigen. Die größten Beiträge von Sinclair beim MK14 lagen aber offenbar nicht in der Schaltung selber, sondern in der kostengünstigen Umsetzung eines 'massentauglichen' Bausatzes. Freilich ist der Tastatur die Verwandtschaft mit dem späteren ZX80 unrühmlich anzumerken. Berichten zufolge war sie beim MK14 nahezu unbrauchbar, ein Grund warum ich bei meinem Fädeldraht-Nachbau dann doch lieber auf mechanische Taster gesetzt habe.



Die MK14-Software

Nach dem Einschalten gelangt man direkt in das Monitor-Programm, mit dem man Speicherzellen auslesen sowie beschreiben und die Programmausführung starten kann. Zusätzlich befinden sich im PROM noch Routinen zum Laden/Speichern auf Kassette, und die Unterstützung für Einzelschritt-Betrieb. Die Programmierung erfolgt direkt im Maschinencode in Form von Hexadezimalzahlen. Einen Kassettenrekorder schließt man nicht direkt an, sondern man braucht zusätzlich einen kleinen externen Modulator/Demodulator.

Mit dem eingebauten Monitorprogramm kann der MK14 aber schon als „echter“ Computer gelten, denn er ist vom Benutzer komplett selbst und frei programmierbar (ein Privileg, das Besitzer moderner Smartphones trotz derer tausendfach höherer Leistungsfähigkeit nicht haben, dort gibt es Programme nur über App-Shops und Programmieren ist zumeist den Profis vorbehalten).

Der Mikroprozessor SC/MP

Der Befehlssatz des MK14-Prozessors ist sehr schön einfach und übersichtlich. Es gibt einen Akkumulator und ein Extension-Register, sowie drei Pointer-Register. Kleine Programme lassen sich damit relativ einfach schreiben, selbst wenn man die Umsetzung in Hex-Codes von Hand vornehmen muss.

Bei eingehender Beschäftigung mit dem National SC/MP wird jedoch auch klar, wieso diesem Prozessor kein großer Erfolg beschienen war. Sobald es über einfache kleine Programme hinaus geht wird es schwierig. So hat der SC/MP erst mal keinen Stack, dafür muss man eines der Pointer-Register nutzen. Auch einen CALL-Befehl gibt es so nicht, beim Aufruf von 16-Bit-Adressen muss mühsam über den Akku geladen werden und es geht wieder ein Pointer-Register dafür drauf. Interrupts – ein für alle Computer wichtiges Merkmal – gibt es nur auf Kosten eines weiteren Pointer-Registers. Kurzum: Komplexe Assembler Programme machen auf dem SC/MP keinen Spaß und sind selbst für 8-Bit-Maßstäbe seeehr langsam.

Der SC/MP ist von seiner Leistungsfähigkeit eher mit frühen Mikrocontrollern wie dem 8035 vergleichbar, im Gegensatz zu diesen hat er aber kein Onchip-Ram und keine Ports. Der SC/MP verschwand daher bald in der Versenkung - und ist heute recht selten.

Für Lehr- und Schulungszwecke war der SC/MP aber dennoch hervorragend geeignet.

Das MK14 Benutzerhandbuch stellt dann auch eine komplette Einführung in die Mikrocomputertechnik dar, geht auf den Prozessor ein, und liefert viele Beispielprogramme als Listings – vom Klassiker „Mondlandung“ über einen Reaktionstester bis hin zum selbstreplizierenden Programm.

Chris Curry hat bei Acorn später ein dem MK14 sehr vergleichbares modulares System gebaut (Acorn 1), dafür aber den 6502 eingesetzt. Sinclair setzte bekanntlich später auf den Z80.

Kleiner Tipp

Im Originalschaltplan ist der SC/MP-Eingang *SENSA* unbeschaltet. Ein 100KOhm Pull-Down-Widerstand von *SENSA* von nach Masse machte bei meinem Exemplar die Programmausführung zuverlässiger.

oliver@zx81.de

Quellen und Weblinks

[1] „Micro Men“, BBC-Fernsehproduktion 2009;
http://en.wikipedia.org/wiki/Micro_Men
[2] MK14-Info: http://www.old-computers.com/museum/computer.asp?c=961
[3] An interview with Steve Furber;
http://mags.acm.org/communications/201105?pg=37#pg37
[4] National Semiconductor Introkit: http://www.old-computers.com/museum/computer.asp?c=1147&st=1
[5] MK14 Benutzerhandbuch mit Schaltplan

HISTORISCHE WERBUNG



DER ZXILOSOPH

Läuft...

von Jens

Liebe User!

Wir haben einen neuen Papst, und der sagt, dass Traditionen wichtig sind und die Menschheit daran festhalten sollte. Da hat er Recht. Allerdings bin ich immer skeptisch, ob das Wort eines Mannes, der in Frauenkleidern herumläuft, auch immer zu 100% richtig sein kann...

Wie dem auch sei: Unsere Tradition, mit älteren Rechnern neue Dinge anzustellen ist eine feine Sache. Und wenn jeder Mensch an einer Sache festhalten würde, die er für erhaltenswert befindet, würde die Menschheit um einiges reicher sein. Wir sind da also absolut auf dem richtigen Weg.

Der ZX81 hat viele Möglichkeiten – die Grenzen setzen wir uns selbst. Im Prinzip ist der Mensch ja dazu geboren, neues zu entdecken und nach vorne zu schauen. Die Weitsicht dessen, was mit dem kleinen Rechner so anzustellen ist, hatte Sir Clive damals sicher nicht - schon gar nicht der Gestalt, das er das ZX-TEAM hätte kommen sehen können. Deshalb bin auch ich gespannt, was in den nächsten Jahren so kommen mag. Vielleicht eine Antigravitationsmaschine, die vom ZX81 gesteuert wird. Oder eine künstliche Intelligenz, die in 16kb passt. Zumindest aber viele neue Programme und Spiele. Ich warte noch darauf, das der ZX81 in einer Simpsons-Folge auftaucht. Oder im neuen Starwars VII.

Es gibt aber viele Dinge, die absolut nicht hätten erfunden werden müssen: zB der Schleudersitz für Hubschrauber, die aufblasbare Dartscheibe, der durchsichtige Sonnenschirm oder sogar die aufklappbare Computermaus für Frauen, inklusive Puder und Spiegel. Da lob ich mir meinen Roboterarm – der kann immerhin was, zB den Kaffee umrühren.

Wenn jetzt jemand glaubt, seine Erfindung mache keinen Sinn... keine Angst, irgend jemandem gefällt sie immer. Im TEAM wird das immer wieder deutlich. Man kann noch so unsinnige Dinge basteln – immer findet das jemand gut. Und das ist toll.

Was kann denn schöner sein, als

etwas zu erschaffen?! Mit dem Zeddy lassen sich so viele schöne Dinge tun.

Und das ist besser als Abends vor der Glotze zu hängen...



Ok, das war gemein, aber das liegt an RTL II. Und Hitler´s Helfershelfer der Helfer brauchen wir auch nicht – zu viel Tradition. Mannomann, ich bin gerade wieder in einer kritischen Phase. Wahrscheinlich Phase IV. Mein Vater hat immer gesagt, ich soll kritisch sein und nicht alles für gegeben hinnehmen. Aber mein Arzt sagt auch, ich soll viel trinken. Und wenn ich dann eine Kiste Bier wegzuppel, findet er das plötzlich doof. Versteh´ einer die Welt.

Der Zeddy macht ja auch was er will. Zumindest kann ich das gar nicht alles nachprüfen, was er so macht. Dafür ist er mir inzwischen zu schnell. In BASIC mag das ja noch gehen, aber bei Assembler ist´s vorbei. Das geht so schnell. Wir können froh sein, dass der Rechner so schnell rechnet. Wo kämen wir denn sonst hin. Ich bin ja schon froh, dass er überhaupt den Strom annimmt. Vielleicht möchte er den gar nicht und seine Ruhe haben? Das wäre doch vorstellbar. Aber ihm fehlt sonst die Wärme. Also lieber doch Strom. Aber welchen? Ökostrom? Windenergie? Solar? Vielleicht ist er ja Vegetarier und mag keinen Atomstrom? Und überhaupt... schneller ist Atomstrom auch nicht.

Ich persönlich glaube, der Zeddy braucht nur Liebe und ab und zu ein nettes Wort. Das baut ihn ungemein auf und er stürzt dann auch nicht ab. „Wer seinen Zeddy ehrt, lebt nicht verkehrt“, schrieb jemand im Forum. Ich sage:

Hauptsache, er läuft...

Zum Schluss gebe ich Euch noch den alten Irischen Segen mit auf den Weg:

Möge der Wind in Eurem Rücken nie Euer eigener sein!

:-)

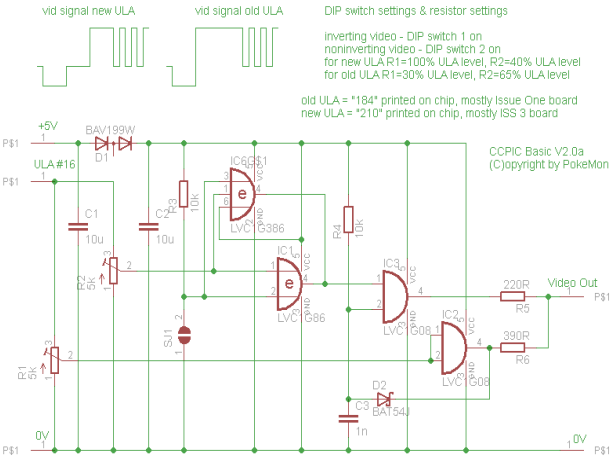
Läuft...

VIDEOMODUL

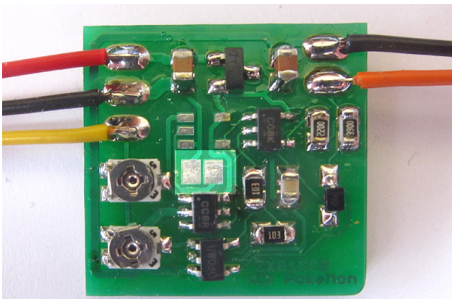
ZX81CCB – Videomodul für ZX81

Von Karl-Heinz Dahlmann

Das bereits bekannte und im letzten Team Magazin vorgestellte Modul ZX81CCP wurde überarbeitet und verfeinert und gibt es als ZX81SCP mit einem Standby Sync Generator, der auch ein stabiles Videosignal im FAST Modus und während des LOAD Befehls liefert und jetzt aktuell in einer preisgünstigen „BASIC“ Version ZX81CCB.



Das Modul bietet die gleiche knackig scharfe Bildwiedergabe durch Verwendung von superschnellen LVC Gates (Low Voltage) als platzsparende SMD Komponenten.



Das Modul wurde als Low Cost Modul konzipiert und daher auf einen DIP Schalter zur Umschaltung der Polarität des Videosignals (normal/invers) verzichtet und statt dessen eine Lötbrücke vorgesehen, im Foto hell markiert. Das Feature ist also noch vorhanden. Anstelle des XOR Gate 74LVC1G86 kann auch ein 74LVC1G386 verwendet werden.

Das ZX81CCB findet mit Abmessungen von 20 x 20 x 3 mm (Länge x Breite x Höhe) und Gewicht von nur 3 Gramm an vielen Stellen im ZX81 Platz. Auch der Stromverbrauch wurde deutlich reduziert durch Verwendung der stromsparenden LVC Technik und liegt bei etwa 10-12 Milliampere. Das Modul wird zum Preis von GBP 4,90 (entspricht EUR 5,90) bei sellmyretro.com angeboten und wird auch auf dem traditionellen Flohmarkt in Mahlerts käuflich zu erwerben sein.

Karl-Heinz Dahlmann, München

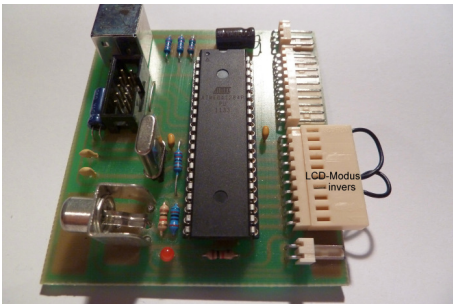
AX81

AX81 – Ein ZX81 für die Westentasche

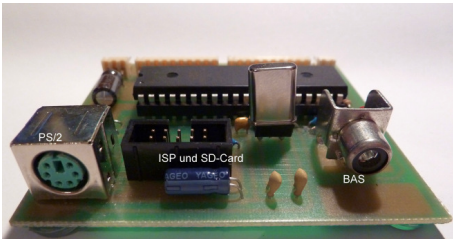
Ein moderner 8-Bit-Mikrocontroller emuliert einen alten 8-Bit-Computer. Dieses grandiose Projekt von Jörg Wolfram hat Joachim aufgegriffen und seinen Bericht im September 2012 ins Online-Forum eingestellt

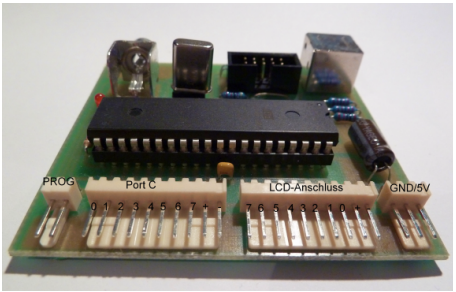
Von Joachim *(gekürzte Fassung des Artikels aus dem ZXTM-Forum [1])*

Im Forum [2] hat Jörg Wolfram seine Entwicklung, den AX81, vorgestellt. Er hat es geschafft, verschiedene Rechner, die mit einer Z80-CPU laufen, auf einem ATMEGA1248 zu emulieren.



Diese Rechner sind unter anderem ein ZX81 und ein ZX80. Zur Bildausgabe können ein BAS-Monitor, ein LCD oder ein VGA-Monitor verwendet werden. Zur Eingabe wird eine PS/2-Tastatur angeschlossen. Programme werden über eine SD-Card gespeichert oder geladen. Die neueste Entwicklung in diesem Projekt ist der AX81B, eine Erweiterung die nun auch zwei PIO-Ports hat. Außerdem ist noch ein Monitor eingebaut, der auch Snapshots erlaubt, sowie eine Datenübertragung über einen 1-wire-Port. Dieses Projekt habe ich nachgebaut und das ist dabei rausgekommen: Zunächst hatte ich mir eine Lochrasterplatine mit dem AX81 aufgebaut. Als ich dann gesehen hatte, wie zuverlässig die Schaltung läuft und als dann die Erweiterung zum AX81B kam, habe ich eine Platine entworfen, die ich beim 'Platinenbelichter' [3] fertigen ließ. *(Layout siehe [1])*





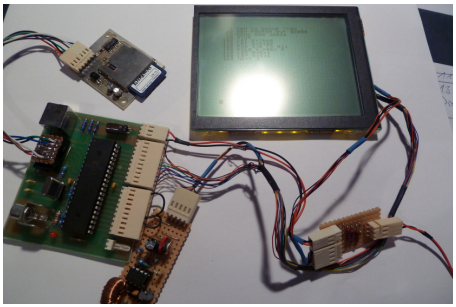
Anfänglich gab es Probleme mit der Übertaktung des ATMega. Bis zu 25 MHz waren ihm dann doch zuviel. Mit einem Oszillator wäre es gegangen, Jörg entschied sich aber, hier nachzubessern und heraus kamen, was den Takt betrifft, drei Varianten: 12 MHz, 14,756 MHz und 20 MHz in der

Version 1.33. Dabei ist zu beachten, dass bei der 20 MHz-Version die Bildausgabe nur über ein LCD möglich ist. Einen BAS-Monitor kann man damit nicht ansteuern. Die 12 und 14 MHz Version kann beides, leider ist da aber noch ein Bug drin, der den Jumper für die Invertierung des Bildschirmes nicht erkennt. Ich verwende derzeit die 20 MHz-Version mit einem LCD und der Firmware **ax81b_v133_cb2_20m.hex**. Der Schaltplan entspricht dem der Minimalvariante aus Jörgs Hardwaredokumentation.

Wer einen BAS-Monitor anschließen will, dem genügt die Platine, eine PS/2-Tastatur, eine SD-Card mit Leser/Schreiber, ein 5V-Netzteil sowie eine der Firmware-Varianten ax81b_v133_cfg2_12m.hex oder ax81b_v133_cfg2_14m.hex.

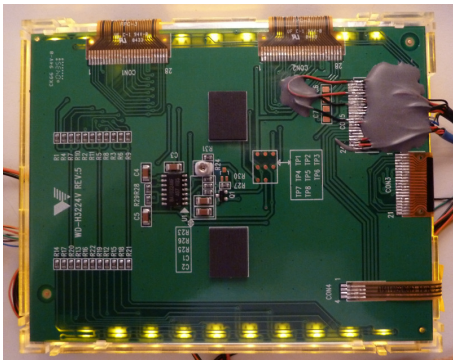
Wer ein LCD (in diesem Beispiel ein Wintek WD H3224) anschließen will, der braucht außerdem noch einen Step-up-Wander für die Kontrastspannung und eine Spannungsreduzierung auf 3,3 V für die Hintergrundbeleuchtung.

Dazu sieht mein Setup so aus:



Das LCD

Ich habe das Wintek WD H3224 gewählt, weil es das günstig bei einem Anbieter gab.



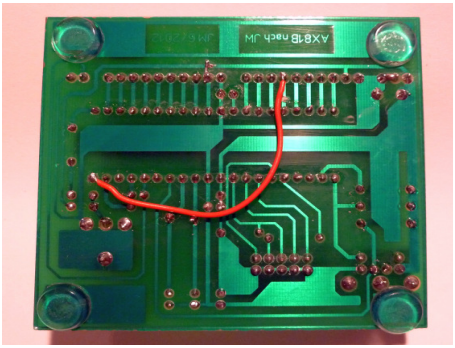
Die Anschlüsse könnte man über eine Flexprint-Buchse zum Port des AX81B führen, leider habe ich aber nirgends eine aufgetrieben. Deshalb habe ich mir Schaltlitze besorgt. Zuerst habe ich die Leiterbahnfolie direkt an den Lötpins auf dem LCD abgeschnitten. Dann habe ich den verbliebenen Rest mit einem Cutter zwischen den Leiterbahnen in kleine Stücke geschnitten und vorsichtig abgelötet. Die Platine kann einiges ab und somit war das kein Problem.

Dann habe ich die Litze angelötet und die Kabel ordentlich mit einem Zweikomponentenkleber gesichert.

(*Anschlussbelegung siehe [1]*)

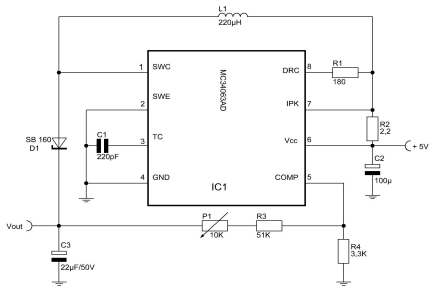
Zu beachten ist, dass das M-Signal mit PD6 verbunden wird. Da PA4 frei bleibt und ich nur einen Stecker zum Anschluss des LCDs verwenden wollte, habe ich die Leiterbahn von PA4 zum Stecker aufgetrennt und am Steckerpin PD6 angelötet. Vielleicht werde ich da

das Layout noch mit einer Brücke versehen.

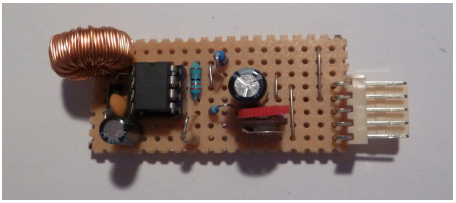


Step-up-Wandler

Entgegen dem Datenblatt haben sich nicht 32 V sondern 21,5 V als ideal herausgestellt. Der Step-up-Wandler reicht von ca. 20 bis 25 V und mit dem Trimmer kann man den gewünschten Kontrast einstellen. Ursprünglich war vorgesehen, da man annimmt, dass das Display erst nach dem Hochfahren des AX81B mit Spannung versorgt werden darf, mit einem Transistor die Spannungszufuhr über PA4 zu steuern. Da es aber offensichtlich ohne Schaden für das LCD auch ohne geht, habe ich den Transistor weggelassen.

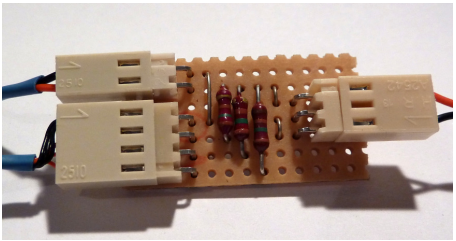


(*Lochmaster-Ansichten siehe [1]*)



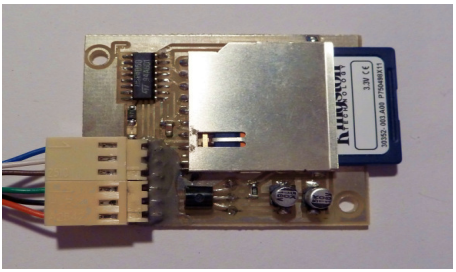
LED-Beleuchtung

Benützt man das Display mit hellem Hintergrund und schwarzer Schrift, ist es zu empfehlen, die eingebauten LED mit 3,3 V zu versorgen. Dazu werden in die drei 5V-Leitungen jeweils ein 150 Ohm-Widerstand gelötet.



Ich habe das auf einer kleiner Loch-rasterplatine gemacht, das die Hauptspannungsversorgung von 5 V auf die Widerstände legt und die 5 V für den AX81B durchschleust.

SD-Card



Als Leser/Schreiber für die SD-Card benütze ich die im ZX-Team bekannte Schaltung mit dem 4050. Das ganze

habe ich auf eine SMD-Platine gelegt und die sieht wie im Foto oben aus.

(*Sprint-Layout und Details zum Adapter-Stecker siehe [1]*)..

Die Programme in Form von P-Files kommen nun so auf die Karte: Wer mit Linux arbeitet, der kann die Programme von Jörg verwenden. Wer mit Windows arbeitet (ich habe es mit Windows 7 getestet), der muss die von mir neu compilierten Programme verwenden und wie folgt vorgehen:

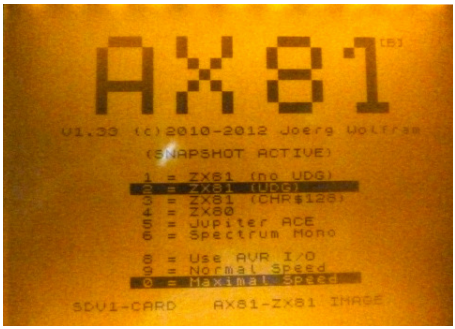
Die Karte muss zuerst mit FAT16 formatiert werden. Ich habe dazu das freie Tool SDFormatter verwendet. Dann werden die Files tape-install.bat und ax81-fsinstall.exe in das root-Verzeichnis kopiert. Nach dem Start von tape-install.bat werden alle Ordner und das Bin-File angelegt. Nun kopiert man in jeden Unterordner (z.B. TAPE000 oder TAPE001) die Programme ax81-read.exe und ax81-write.exe. Zuletzt kopiert man die ZX81-Programme, die man z.B. im TAPE000 haben will in den Ordner TAPE000 und startet in diesem Ordner das Programm ax81-write.exe. Jetzt werden die ZX81-Programme in ein Binärfile geladen. Auf dieses Bin-File greift der AX81B zu, wenn er auf die Karte schreibt oder von ihr liest.

Wenn man Programme im AX81B erstellt und auf Karte gespeichert hat, kann man diese in den PC laden. Dazu muss man auf dem PC in das Tape-Verzeichnis gehen, in dem das Programm gespeichert wurde (Z.B. TAPE001). Dort startet man das Programm ax81-read.exe. Nun werden alle Programme aus dem Bin-File für dieses Tape in den Ordner geschrieben. Fertig.

Betrieb

Will man das LCD nicht invers (weiße Schrift auf dunklem Hintergrund) dargestellt haben, muss PC4 von PortC auf low gelegt werden, zumindest solange, wie der AX81B hochgefahren ist. Dann spielt das Signal keine Rolle mehr. Dazu habe ich mir einen Stecker angefertigt.

Nach dem Einschalten erscheint das folgende Bild auf dem LCD (ohne Orangestich).



Will man mit der PIO arbeiten, muss man '8' (Use AVR I/O) eingeben. Ansonsten kann man sich noch die Geschwindigkeit und die Emulation aussuchen. Paßt alles, sorgt ein Druck auf die Space-Taste dafür, dass die Warteschleife abgebrochen wird. Wenn eine SD-Karte eingesteckt ist, wird deren Image-Namen (Binärfile) angezeigt und man weiß, dass die Karte erkannt wurde. Kurz darauf erscheint der bekannte ZX81-Cursor.

Die Port-Ausgabe hat Jörg genial gelöst:

`POKE 32295, n` definiert, welche Bits am Port C Ein- oder Ausgabeports sind (1 = Ausgabe, 0 = Eingabe) .

`POKE 32296, n` gibt den Wert n am Port aus. Das Einlesen habe ich noch nicht ausprobiert.

Die snapshot-Funktion sorgt dafür, dass ein Programm nach dem Start des AX81B automatisch in den Programmspeicher geladen wird. Dazu muss mit F12 zunächst der Monitor aufgerufen werden. Dann wird mit Shift S der snapshot aktiviert. Das Programm wird nun in den Flash-Speicher geschrieben.

Wird der Monitor und die Aktivierung des snapshot aufgerufen, während ein Programm läuft, startet das Programm nach dem Hochfahren das AX81B genau an dieser Stelle!

Die weitere Bedienung ist aus der Dokumentation von Jörg ersichtlich.

Speichermäßig ist der AX81B nur mit 15 KB RAM bestückt. RAMTOP liegt in der Emulation für einen 'normalen' ZX81 bei 31744 ($7C00) . Das heißt, dass Programme, die unter dem ZX81-Ramtop laufen angepaßt werden müssen bzw. Programme, die größer als das RAM sind, nicht geladen werden können.

Ein paar Änderungen wird Jörg sicherlich noch vornehmen. Ich denke aber dass man das Projekt im jetzigen Stadium schon nachbauen kann.

Ich wünsche allen, die sich dazu entschließen, viel Spaß!

Joachim

Quellen und Weblinks

Hinweis: Das Projekt wird ständig weiter vorangetrieben - Nachbauer sollten sich daher unbedingt im Forum den aktualisierten Stand holen.

[1] http://forum.tlienhard.com/phpBB3/viewtopic.php?f=7&t=755#p6400

Thema „ZX-Team Magazin“ (Anmeldung erforderlich)

[2] http://www.mikrocontroller.net/topic/ax81-zx81-im-avr

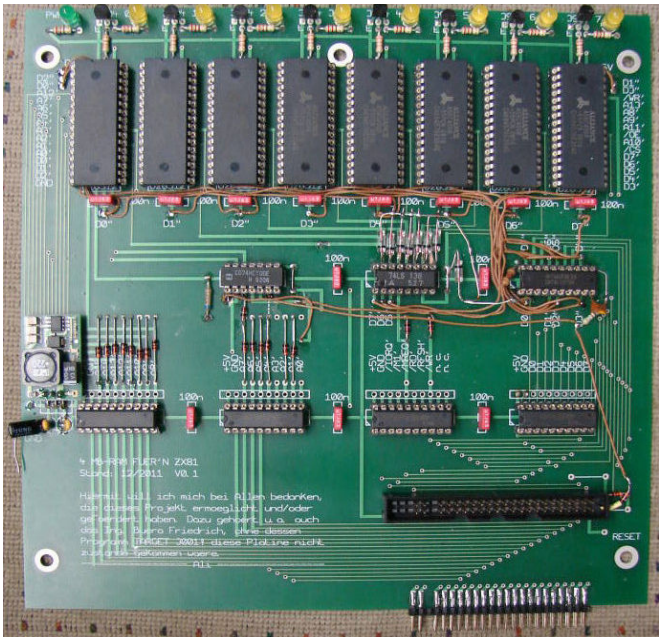
[3] www.Platinenbelichter.de

[4] www.abacom-online.de

Mehr geht immer
- Die 4 MB Speichererweiterung für den ZX81

Fast auf den Tag genau vor 20 Jahren wurde die erste Sensation im ZX-Team Magazin vorgestellt. Zwanzig Jahre? Zwanzig Jahre - das war im Heft 3/93, Männers wir schreiben inzwischen 2013!

Von Ali Schams und Kai Fischer



Auf dieser Schaltung von Gotthard Schulz basiert eine Steckkarte für den ZX96 im handlichen Euroformat von 160x100 mm. Diese läßt sich zwar mit relativ wenig Aufwand auch auf 4 MB aufbohren, aber das hat unseren Neuzugang Ali Schams aber nicht davon abhalten können, eine eigene Schaltung zu entwerfen. Mutig wie er ist, hat er auf einen Fädelaufbau verzichtet und gleich die Platine dazu entworfen und fertigen lassen - mit den stolzen Abmessungen von 200 x 200 mm! Das ist knapp dreimal soviel Platinenfläche wie vorher und damit ist alles auch recht luftig bestückt. Warum trotz des doppelseitigen Layouts dennoch reichlich Drahtbrücken verbaut werden müssen,

bleibt sein Geheimnis. Ebenso geheim bleibt, wie er ausgerechnet darauf kommt, mir den ersten Aufbau und die Inbetriebnahme anzuvertrauen. Die Platine war bereits zum Treffen 2012 fertig, den Artikel dazu hatte ich aber aus Zeitgründen leider nicht mehr geschafft.

Die Organisation der 4-MB-Karte ist identisch zu 1-MB-Karte. Es werden jeweils 16 kB große Speicherblöcke im Adreßbereich ab 49152 jeweils mit POKE 9 Befehl umgeschaltet. Mit 4 MB ist der POKE ausgereizt, denn das gibt exakt 256 Bänke. Die Schaltung ist im Detail jedoch etwas anders gestrickt, als unsere „alte“ Megabytekarte, insbesondere was die Dekodierung betrifft. Alle nötigen Adreß-, Steuer- und Datenbussignale werden jeweils mit einem 74LS245 gepuffert. An den Ausgängen ist die Bestückung einer Steckleiste möglich, die gepufferten Signale lassen sich hier für zusätzliche Erweiterungen abgreifen. Die Richtungsumschaltung des Datenbuspuffers erfolgt nur mit dem /RD-Signal, dahinter angesteckte Erweiterungen sind damit zunächst nicht HRG-fähig. Das läßt sich aber mit einer kleinen Diodenlogik nachrüsten.

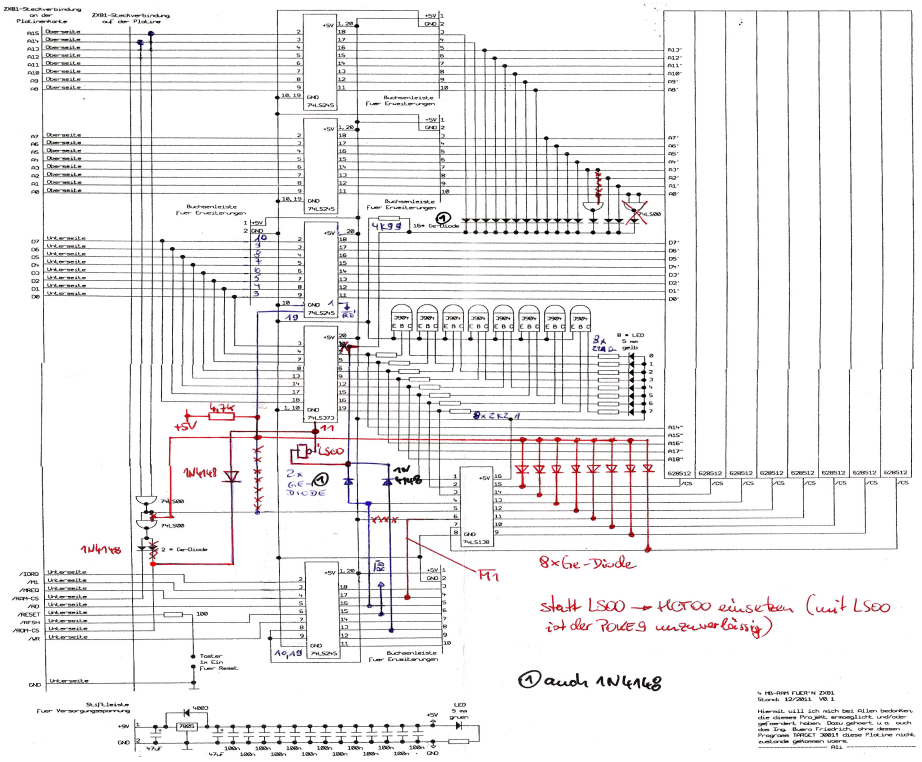
Für die Stromversorgung ist auf der Platine ein eigener 7805-Regler vorgesehen. Ich habe stattdessen die Schaltreglerplatine bestückt, womit auch das sperrige Kühlblech gespart wird. Das 5-Volt-Pin auf dem nach oben zeigenden Erweiterungsstecker hat Verbindung zum ZX-Bus. Diese Verbindung habe ich getrennt und auch die 5-Volt am Stecker mit auf den Schaltregler geführt. Im endgültigen Aufbau steckt hier noch das LCD-IF und damit ist der ZX-interne Regler komplett entlastet. Das gesamte System wird von einem originalen ZX81-Netzteil gespeist, hier sollte man möglichst das kräftigere mit 1200mA Belastbarkeit einsetzen. Aufpassen muß man bei der Nutzung dieses Erweiterungssteckers: hier ist nämlich die obere und untere Pinreihe vertauscht. Für meinen Aufbau war das ideal, weil ich so direkt mit Flachbandkabel und aufgepressten Steckern eine Verbindung zum LCD-IF herstellen konnte. Man kann jedoch keine ZX96-Karte von oben aufstecken!

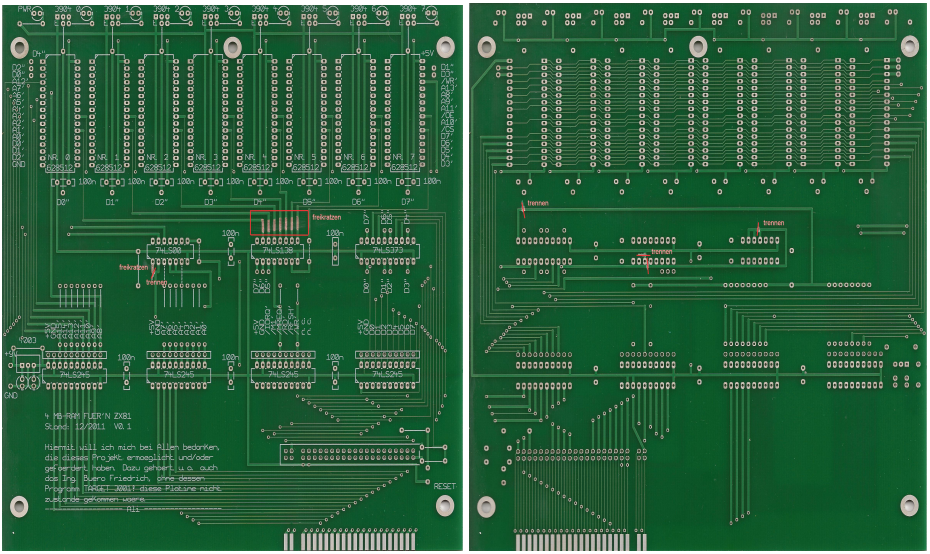
Die Dekodierung des POKE 9 wurde recht rustikal mit einer Diodenlogik gelöst. Anstelle der Ge-Dioden kann hier problemlos die 1N4148 eingesetzt werden. Auch wenn es „Pfennigbeträge“ sind - für eine einzige Ge-Diode bekommt man inzwischen einen ganzen Streifen 1N4148. Die teuren Ge-Dioden braucht man an anderer Stelle der Schaltung noch. Der POKE 9 wird mit einem 74LS373 gespeichert und der Wert optisch schön in binärer Form mit 8 Leuchtdioden angezeigt.

Nach dem Durchsehen des Schaltplans war mir klar, daß zur Inbetriebnahme doch einige Änderungen nötig sein würden. Dazu muß dann leider auf der fertigen Platine etwas herumgekratzt und fliegend verdrahtet werden. Mit den eingestrickten Änderungen funktionierte der Aufbau dann aber fast auf Anhieb. Getreu der alten Regel „wenn’s

auf Anhieb klappt, dann ist was faul“ offenbarte sich dann eine etwas wacklige Funktion des '373. Mal reagierte er gar nicht auf POKE 9, mal speicherte er den falschen Wert. Der '373 ist ein transparentes Latch. Solang der Takteingang ein H-Signal führt, folgen die Ausgänge den Eingangssignalen. Erst bei L-Signal wird der letzte Stand gehalten. Ich vermutete zuerst Störimpulse auf dem Takteingang, aber der Fehler lag in der Diodenlogik: Das 74LS00 Gatter vor dem Takteingang wird mit einem 4,7 k Widerstand auf Masse gezogen und durch die Dioden ein (N)OR Gatter gebildet. Dieser Widerstand ist schon zu groß, um den 74LS00 noch auf einen sauberen L-Pegel zu ziehen. Kleiner kann er aber auch nicht gemacht werden, da immerhin 17 Signale darauf OR-verknüpft sind. Bei 4,7 k macht das in der Summe ca. 17 mA, aber bei 1 k eben gleich 5 mA je Signal. Mit den insgesamt 100 mA verheizt der Widerstand dann ½ Watt! Der Austausch des 74LS00 gegen einen HCT00 führte dann zu einem zuverlässig arbeitenden POKE 9.

Die im Schaltplan blau eingetragenen Änderungen stammen noch von Ali, die roten Eintragungen stammen von mir.





Auf der Platine müssen zunächst ein paar Kratzereien ausgeführt werden. (Hochauflösende Versionen dieser Fotos finden sich im Online-Forum [1])

Der Umfang der frei einzustrickenden Verdrahtung wird auf dem folgenden Foto deutlich (noch mit einem kleinen RC-Glied vor Pin 11 des LS373, kann bei Einsatz des HCT00 entfallen):

Nach dem Aufbau der Platine empfiehlt sich zunächst ein Kurztest mit einem kleinen Basicprogramm, das jede Bank kurz anprüft:

```
10 FOR I = 0 TO 255
20 POKE 9,I
30 POKE 50000,I
40 NEXT I
50 FOR I = 0 TO 255
60  IF  PEEK  50000  <>  I  THEN
PRINT I
70 NEXT I
```

Das ausführliche Testprogramm MEMTEST.P hat mir Siggi schon 2012 in aller Eile gestrickt, es liegt im Forum zum download. Damit werden die gesamten 4 MB durchgeprüft. Ein Schleifendurchlauf ist auch möglich, um sporadische Fehler aufspüren zu können.

Quellen und Weblinks

[1] http://zx81.tlienhard.com/

Von Ali

- - - - - - - - Ein Science Fiction Roman - - - - - - - -

Die Zukunft hat begonnen

Heute ist einer der ueblichen heissen und trockenen Hochsommertage, wie man sie in den letzten Jahren haeufig erleben konnte. Mit +37 Grad Celsius ist es fuer einen Sommertag eigentlich noch zu kuehl. Kaum ein Mensch traut sich noch an die Erdoberflaeche, hier in dem Land, das vor ueber einhundert Jahren einmal Deutschland genannt wurde und in Europa lag. Grosse Teile des ehemaligen Kontinents sind heute wuestenaehnliches Brachland. Die starke Umweltverschmutzung, die mit der Entsorgungswelle der Windows-Computer gegen Ende des 20. Jahrhunderts eingesetzt hatte und im 21. Jahrhundert ihren Hoehepunkt erreichte, war zum Teil dafuer verantwortlich. Trotz der massiven Proteste von Windows-Benutzern und Pro-Windows-Verbaenden, die sich im 20. Jahrhundert gebildet hatten, wurde die Natur weiter stranguliert. Zu Beginn des 21. Jahrhunderts versuchte man die Katastrophe zwar aufzuhalten oder umzukehren, aber da war es bereits zu spaet. Insbesondere das ehemalige Microsoft hielt es fuer wichtiger, seine Entwicklungen anderen Firmen aufzuzwingen (meist in Form einer feindlichen Uebernahme), statt Gelder in den Umweltschutz zu investieren und friedliche Loesungen zu finden.

Selbst als Ueberschwemmungen, Wirbelstuerme und verheerende Waldbraende die Welt immer gewaltiger beutelten, wurden Milliardenbetraege fuer Windows-Computer verprasst; fuer Computer die mit anderer Software laufen konnten und dadurch fuer Viren, Wuermer und Trojaner unanfaelliger waren, gab es kaum Gelder.

Die Erde ist eine einzige, oede Wueste. Lediglich in der Aequatorregion kann man noch einigermassen an der Oberflaeche leben. Die wenigen Menschen, die es sich leisten koennen, haben sich dort eine neue Heimat gesucht. Durch die Katastrophe sind Milliarden Menschen ums Leben gekommen, Flora und Fauna sind nahezu vollstaendig ausgerottet. Pflanzen gibt es nur noch vereinzelt, hauptsaechlich in unterirdischen Gaertnereibetrieben. Die meisten Tierarten sind ausgestorben, das letzte Hausschwein in Europa starb vor knapp 25 Jahren an Altersschwaeche in einem Zoo.

Selbst der Mensch gehoert inzwischen zu den so genannten 'vom Aussterben bedrohten Tierarten'. Saeugetiere gibt es kaum noch, lediglich die Insekten haben es geschafft, sich den neuen Umweltbedingungen anzupassen und Pflanzen (ausser den von Menschen kuenstlich gezuechteten und daher bezeichneten "Kulturpflanzen") gibt es in der freien Natur nur noch einige Flechten und Sporenpflanzen. Nachdem die Oberflaeche der Erde eine einzige Wueste und Oednis ist, zwang es den Menschen, in unterirdische Hoehlen zu fluechten und dort neue Staedte zu errichten. Wie gesagt, nur die Aequatorregion ist noch oberflaechlich bewohnbar. Krankheiten und Hungersnoete, die nach dem letzten grossen Krieg zu Beginn des 22. Jahrhunderts wueteten, gaben der Menschheit den Rest.

2131 (150 Jahre nach Erfindung des ZX81 im Jahre 1981)

Zu den wenigen Computern, die noch funktionsfaehig sind, gehoert der ZX81 von Clive Sinclair.

Er steuert mittlerweile von der Modelleisenbahn bis zur Fraesmaschine, von der Discobeleuchtung bis zur Ampelanlage alles. Moeglich wurde dies erst nachdem die 4 MB-Speichererweiterung fuer Standard-Zeddys erfunden wurde. Diese ermoeglicht auch grosse Datenmengen fuer den Prozessor im Speicher vorraetig zu halten.

Ihre Vorteile sind ausserdem ultra-schneller Zugriff auf die Datenmengen und optische Signalisierung der jeweils aktiven Speicherbank.

Interesse an mehr ? Siehe Artikel "4 MB-RAM-Erweiterung" an anderer Stelle in dieser ZX-Magazin-Sonderausgabe !

```
+-----------------------------------------------+
| Als Vorlage diente das Kapitel "Prolog" aus   |
| dem Roman "Die Erfindung" von Armin Meiwes.   |
+-----------------------------------------------+
```

WAS MACHT EIGENTLICH ...



Auf einer Internetseite des ehemaligen Sinclair-Chefdesigners ***Rick Dickinson*** sind seine Sinclair-Produkte und – mindestens ebenso spannend – auch seine niemals umgesetzten Entwürfe aus jener Zeit zu sehen.



Das elegante Design hatte stets einen großen Anteil am Erfolg von Sinclairs Produkten, auf der Webseite von Dickinson kann man erahnen, wie viel Aufwand dahinter stand. Die hier gezeigten Ausschnitte stammen von seiner Seite. Dickinson verließ die Firma Sinclair im Jahr 1986 und betreibt seither sein eigenes Designbüro:

http://www.theproductdesigners.com/ZX80/zx80.html

NEU: C COMPILER FUER ZX80

Von Paul

Hallo Freunde des ZX80

Ich habe es geschafft *Stefano Bodrato*, den Autor der den Z88dk weiterentwickelt zu überzeugen auch den ZX80 zu unterstützen. Leider ist mit 1K nichts zu schaffen. Aber ab 4K RAM können C Programme sinnvoll eingesetzt werden. Dazu sollte eine Version ab v1.10 eingesetzt werden. Finale Tests auf der original Hardware stehen noch aus, aber in EO funktioniert alles sehr schön. Die Tests habe ich mir für Mahlerts vorgenommen.

Liebe Grüße Paul.

NEU: 8BIT-WIKI

Ein neue Datenbank für Heimcomputer hat ***Dieter Hucke*** aufgesetzt, und viele Fans alter 8-Bit-Computer haben schon Material beigesteuert. Die Internetseite ist als Archiv für die wichtigsten Ressourcen wie Anleitungen, Schaltpläne und ROM-Abbilder gedacht und konzentriert sich zunächst auf Z80-Computer, Sinclair und Verwandte

http://8bit-wiki.de.vu/home.html

'QL TODAY' WIRD EINGESTELLT

Siebzehn Jahre lang kam aus Duisburg das QL Magazin "**QL Today**" regelmäßig in die Briefkästen der Leser. In der aktuellen Ausgabe (Issue 3/Volume 17) wird nun das nahe Ende der Zeitschrift angekündigt.

Ein letztes Heft soll im September noch erscheinen, dann zusammen mit einer CD oder DVD, welche den Gesamtinhalt als Archiv enthält.

In der „QL-users" Mailingliste wurde die Nachricht mit Schock und großem Bedauern aufgenommen. Es gab zahlreiche Beiträge, die sämtlich dem Herausgeber Jochen Merz und seinen Mitstreitern für das langjährige wackere Engagement danken.

DIE GANZE WAHRHEIT ...

*...über **Sinclairs Microdrive** lässt sich ausführlich unter*

http://www.theregister.co.uk/2013/03/13/feature_the_sinclair_zx_microdrive_story/

nachlesen. Und die geht in Kürze etwa so:

Bei der Vorstellung des ZX Spectrum im April 1982 bereits angekündigt, setzte das Microdrive neue Preispunkte (50 Pfund im Vergleich zu 400 Pfund für ein Commodore-Floppy-Laufwerk). Anfangs noch als Microfloppy bezeichnet, wurde bald klarer, dass es sich bei Sinclair's Massenspeicher-Lösung um ein Bandlaufwerk handelt. Im Sommer 1983 schließlich starte die Auslieferung von Microdrive und Interface 1.

Es gibt den Bericht eines Ingenieurs, der sich 1974 bei Sinclair beworben

und im Job-Interview seine Idee einer Endlosband-Massenspeicherlösung präsentiert hatte. Ob Sinclair tatsächlich auf diese Weise an das Microdrive-Prinzip gelangte, oder ob es später unabhängig davon erfunden wurde, bleibt jedoch ungeklärt.

Mit der Software für das Interface 1 hat man einen freiberuflichen Programmierer, Ian Logan, beauftragt. Logan hatte zuvor in einem Buch das ZX Spectrum ROM analysiert, und kannte sich damit offenbar besser aus als die Sinclair-Mitarbeiter selber. Die zwei zentralen Spectrum-Experten, Vikers und Altwasser, hatten zu diesem Zeitpunkt nämlich schon eine eigene Firma und arbeiteten am Jupiter Ace.

Wie hinlänglich bekannt, haben die Microdrives einige Schwachstellen, teils prinzipieller Natur, teils aus Qualitätsgründen. Die Audiokassette konnten die ZX Microdrives jedenfalls nicht verdrängen. Entsprechend übertrugen sich die Probleme dann später auf den mit Microdrives ausgestatteten QL.

Fazit: Aus heutiger Sicht (und wenn man nicht auf die Zuverlässigkeit der kleinen Schnürsenkel angewiesen ist) ist Sinclairs Microdrive ein faszinierendes Stück Technologie, und in jedem Fall mal wieder völlig anders als der ganze Rest der Computerwelt.

DEMNAECHST: ZX80 GEHAEUSE

Die folgende Ankündigung für neu produzierte ZX80-Gehäuse kommt von **Ali**. In alter Sinclair-Tradition scheuen wir uns nicht, dies schon jetzt öffentlich zu machen: Sinclair hatte ja auch oftmals schon sehr frühzeitig Produkte anpriesen, obwohl sie noch nicht so ganz fertig waren...

Ich kuendige hiermit eine ZX80-Gehaeuse-Nachbildung an. Sie wird exakt so aussehen wie das Original-Gehaeuse, in beliebigen Farben (nicht nur weiss, rot, schwarz) und deutlich unter 100 EUR kosten.

Das erste Exemplar ist voraussichtlich Ende dieses Jahres fertig; ich komm vorher leider nicht dazu. Verfahren das ich verwenden werde: eine Art Gusstechnik auf Harz-Basis (mehr wird noch nicht verraten !) Ali

MAHLERTS 2012









