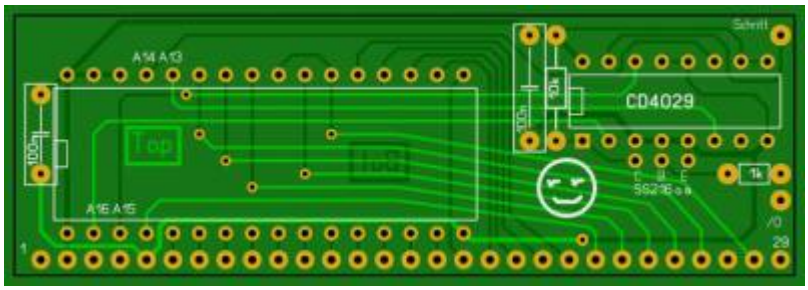
ROM-Bank

Die ROM-Bank ist ein Zusatzmodul für den TINY, ursprünglich entwickelt für die Ausbaustufe III und ES4.0.



Von Rolf Weidlich entwickeltes Modul mit Softwaresteuerung.

1. 128kByte ROM
2. 16 Bänke, umschaltbar per Binärzähler 4029
3. Betriebssystem muss um Routinen zur Bankumschaltung erweitert werden.



Leiterplatte von W. Hardwardt,
https://eb-harwardt.jimdofree.com/8-bit-technik/tiny-computer-6kb-aus-ju-te/

Es gibt auch eine größere Variante mit 512 kByte ROM:

1. 512kByte ROM
2. 64 Bänke, umschaltbar per Binärzähler 4024

s.a. https://eb-harwardt.jimdofree.com/8-bit-technik/tiny-computer-6kb-aus-ju-te/ weiter unten „Eprom Bank 2021 mit 512KB“

ES4.0

TINY 6K ROM-Bank für Ausbaustufe III und ES4.0

1. 128kByte ROM
2. ROM-Modul für Steckplatz 1 (auf 6K-Tiny!)
3. 2000h-3FFFh (8 kByte)
4. 16 Bänke, umschaltbar per Binärzähler 4029

5. Angepasstes Betriebssystem ES 40c mit Routinen zur Bankumschaltung etc.
6. Start des Moduls mit M (Mon), X (Bank 0, Sprung zu Adresse 2000h)

Mit zwei Leitungen ist dann das Modul mit der Hauptplatine zu verbinden:

```
/0 -> P3.5 Zählerreset
Schritt -> P3.7 weiterzählen (nächste Bank)
```

Die ROM-Bank wird mit dem Monitorkommando X gestartet:

```
M    (Mon)
X    X ohne Parameter zeigt den Inhalt der ROM-Bank an (DIR)
Xn   es wird gleich Programm Nr. n gestartet.
```

Downloads

- https://eb-harwardt.jimdofree.com/8-bit-technik/tiny-computer-6kb-aus-ju-te/, „Rombank zur Aufnahme von auswählbarer Software“, Eprombank_190603.pdf
  Beschreibung der Hardware und der Modulsoftware. Enthält auch Änderung des Systems zur Version 40c
- https://www.tiny-computer.de/about, dort unter „Weitere Programme für den JU+TE 6k“, „ROM-Bank Software“ (EPROM29_10_EMU.zip)
  orig. Modulsoftware, wie in der PDF beschrieben
  - rombank_es40.zip

  Meine ROM-Bank-Software, 29.07.2024, komplett incl. Quellcode und Programmen
  depack_z8b.asm ist der bitbuster_extreme-Entpacker im z8-Code.
  - rombank_roms.zip

  ROM-Bank, nur der ROM + Anleitungen

Beschreibung

alternative Software für „ROM-Bank 2000“ für Ju+TeC-6k, WeRo, buebchen



Ich habe 2021 eine alternative Software für die ROM-Bank geschrieben, um mehr Programme auf dem ROM unterzubringen und auch, um weitere Programme einfach hinzufügen zu können. Dabei kamen Erfahrungen und Software der Mega-Modulsoftware für Z9001 und Z1013 zum Tragen. Um Platz zu sparen, können die Programme vorab mit bitbuster_extreme gepackt werden. Aktuell sind 58 Programme in der ROM-Bank enthalten (tiny MPBasic, Maschinencode, Bilder).

ROM-Image aus obigem Downloadpaket: packedrom.bin 128 KByte

testen

1. jtcemu (www.jens-mueller.org/jtcemu/), Version 2.0 oder neuer
2. Konfiguration 6KByte ES4.0, 64K RAM, ROM-Bank, Inhalt der Rom-Bank packedrom.bin
3. beim Start muss sich das System als ES4.0c melden
4. Start der Modulsoftware mit M (Mon), X

Inhalt

Stand 19.9.2021. 58 Programme.

Bank, Adr., Zieladr., Endadr., Startadr., Typ, Name, Länge (gepackt), Quelle

```
1, 2000h, 0C100H, 0h, 0h, 'M', ASDIS, 8F0,
Software/kassette6k/asdis_6k.bin.pck
1, 2600h, 0C000H, 0h, 0h, 'M', Bitman, 60,
Software/kassette6k/bitman_6k.bin.pck
1, 2700h, 0F100H, 0h, 0h, 'M', KC-Trans, 300,
Software/kassette6k/kctrans_6k.bin.pck
1, 2A00h, 0E000H, 0h, E000h, 'B', EPROM-Br.6k, 1000,
Software/kassette6k/eprommer_6k.bin.pck
1, 3000h, 0D000H, 0h, D000h, 'M', BasicFloat, 1000,
```

```
Software/kassette6k/float_fuer_basic.bin.pck
1, 3A00h, 0D000H, 0h, D000h, 'M', Forth 6k, 1500,
Software/volker/forth_6k_vp_d000.bin.pck
2, 2A00h, 0C0F0H, C0F4h, C0F0h, 'M', ZG Z9001, 5,
Software/volker/zg_z9001.bin
2, 2B00h, 0E000H, 0h, E000h, 'B', HaseWolf 6k, 5F2,
Software/kassette6k/hase_wolf_6k.bin.pck
2, 2F00h, 0E000H, 0h, E000h, 'B', Kass-Info, 61E,
Software/kassette6k/kassetteninhalt_6k.bin.pck
2, 3400h, 0E000H, 0h, E000h, 'B', Master M.6k, 3CE,
Software/kassette6k/master_mind_6k.bin.pck
2, 3700h, 0F480H, 0h, 0h, 'M', S3004-Treib, 100,
Software/kassette6k/s3004_treiber.bin.pck
2, 3900h, 0E000H, 0h, E000h, 'B', Apollo, 95E,
Software/Perser/Apollo_E000.BIN.pck
3, 2300h, 08000H, 0h, 8000h, 'M', HANOI, 364,
Software/Rolf/Soft_8000/HANOI_8.bin.pck
3, 2700h, 08000H, 0h, 8000h, 'M', LADDER, A3D,
Software/Rolf/Soft_8000/LADDER_8.bin.pck
3, 3200h, 08000H, 0h, 8000h, 'M', LIFE, ABE,
Software/Rolf/Soft_8000/LIFE_8.bin.pck
3, 3D00h, 08000H, 0h, 8000h, 'M', MERKS, 5BD,
Software/Rolf/Soft_8000/MERKS_8.bin.pck
4, 2300h, 08000H, 0h, 8000h, 'M', NIMM, 2EB,
Software/Rolf/Soft_8000/NIMM_8.bin.pck
4, 2700h, 08000H, 0h, 8000h, 'M', SCHIEB AB, 39A,
Software/Rolf/Soft_8000/SCHIEB_8.bin.pck
4, 2B00h, 08000H, 0h, 8000h, 'M', TIC-TAC-TOE, 48C,
Software/Rolf/Soft_8000/TTT_8.bin.pck
4, 3000h, 08000H, 0h, 8000h, 'M', UHR, 862,
Software/Rolf/Soft_8000/UHR_8.bin.pck
4, 3900h, 08000H, 0h, 8000h, 'M', Wurmi, 699,
Software/Rolf/Soft_8000/Wurmi_8.bin.pck
5, 2000h, 08000H, 0h, 8000h, 'M', Testbild, 19B,
Software/Rolf/Testbild/testbild.bin.pck
5, 2200h, 08000H, 0h, 8000h, 'M', MINE, 716,
Software/Thomas/MINE_8_korr.bin.pck
5, 2A00h, 08000H, 0h, 8000h, 'M', SOKOBAN, 74F,
Software/Thomas/SOKOBAN_8_korr.bin.pck
5, 3200h, 08000H, 0h, 8000h, 'M', Steine, C18, Software/Thomas/6k-
steine_Kursor.bin.pck
5, 3F00h, 08000H, 0h, 8000h, 'M', TETRIS, 811,
Software/Thomas/TETRIS_8_korr.bin.pck
6, 2800h, 0E000H, 0h, E000h, 'B', 17UND4, 485,
Software/Wolfgang/17UND4.bin.pck
6, 2D00h, 0E000H, 0h, E000h, 'B', ALIEN, 629,
Software/Wolfgang/ALIEN.bin.pck
6, 3400h, 0E000H, 0h, E000h, 'B', ANGELN, 757,
Software/Wolfgang/ANGELN_neu.bin.pck
6, 3C00h, 0E000H, 0h, E000h, 'B', AUTOFAHRT, 6EE,
```

```
Software/Wolfgang/AUTOFAHRT.bin.pck
7, 2400h, 0E000H, 0h, E000h, 'B', BOMBER, 67B,
Software/Wolfgang/BOMBER.bin.pck
7, 2B00h, 0E000H, 0h, E000h, 'B', BONK, 642, Software/Wolfgang/BONK.bin.pck
7, 3200h, 0E000H, 0h, E000h, 'B', BOWLING, 4B2,
Software/Wolfgang/BOWLING.bin.pck
7, 3700h, 0E000H, 0h, E000h, 'B', BRUECKE, 5C8,
Software/Wolfgang/BRUECKE.bin.pck
7, 3D00h, 0E000H, 0h, E000h, 'B', COLOR-MASTERMIND, 5E9,
Software/Wolfgang/COLORMA.bin.pck
8, 2400h, 0E000H, 0h, E000h, 'B', ENTENJAGD, 6AB,
Software/Wolfgang/ENTENJAGD.bin.pck
8, 2B00h, 0E000H, 0h, E000h, 'B', FLIEGER, 40D,
Software/Wolfgang/FLIEGER.bin.pck
8, 3000h, 0E000H, 0h, E000h, 'B', FROSCH, 6B4,
Software/Wolfgang/FROSCH_neu.bin.pck
8, 3700h, 0E000H, 0h, E000h, 'B', JUMPER, 572,
Software/Wolfgang/JUMPER.bin.pck
8, 3D00h, 0E000H, 0h, E000h, 'B', LANDER, 60B,
Software/Wolfgang/LANDERV6.bin.pck
9, 2400h, 0E000H, 0h, E000h, 'B', OPA SINCLAIR, 607,
Software/Wolfgang/OPA_SINCLAIR.bin.pck
9, 2B00h, 0E000H, 0h, E000h, 'B', PAUL, 4D5, Software/Wolfgang/PAUL.bin.pck
9, 3000h, 0E000H, 0h, E000h, 'B', ROEMISCH, 22F,
Software/Wolfgang/ROEMISCH.bin.pck
9, 3300h, 0E000H, 0h, E000h, 'B', SKARABAEUS, 672,
Software/Wolfgang/SKARA.bin.pck
9, 3A00h, 0E000H, 0h, E000h, 'B', SKEET, 5BD,
Software/Wolfgang/SKEET.bin.pck
A, 2000h, 0E000H, 0h, E000h, 'B', WOCHENTAG, 21C,
Software/Wolfgang/WOCHENTG.bin.pck
A, 2300h, 0E000H, 0h, E000h, 'B', WUERFELN, 5D9,
Software/Wolfgang/WURF.bin.pck
A, 2900h, 0A000H, 0h, A000h, 'M', MINE 8 Raeume, 132B, Software/Thomas/MINE-
Bank/mine08_a000.bin.pck
A, 3D00h, 0A000H, 0h, A000h, 'M', MINE 74 Raeume, 22A4,
Software/Thomas/MINE-Bank/mine74.bin.pck
C, 2000h, 0E000H, E28Ch, E000h, 'B', Test BASERW, 28D,
Software/volker/testbild.bin
C, 2300h, 08000H, 0h, 8000h, 'P', ARTHUR, B55,
Software/Rolf/Bildbetrachter/Arthur.bmp.pck
C, 2F00h, 08000H, 0h, 8000h, 'P', Pixelgrafik, 9FC,
Software/Rolf/Bildbetrachter/pixelgrafik.bmp.pck
C, 3A00h, 08000H, 0h, 8000h, 'P', BUDDEL, A99,
Software/Rolf/Bildbetrachter/Buddel.bmp.pck
D, 2500h, 08000H, 0h, 8000h, 'P', DDR, 1059,
Software/Rolf/Bildbetrachter/DDR.bmp.pck
D, 3600h, 0E000H, 0h, E000h, 'B', Tropfen, 413,
Software/Wolfgang2/Tropfen_PLUS.bin.pck
D, 3B00h, 0E000H, 0h, E000h, 'B', SCHMUGGLER, 3CA,
Software/Wolfgang2/SCHMUGGLER_PLUS.bin.pck
```

```
D, 3F00h, 0E000H, 0h, E000h, 'B', NEUN GEWINNT, 2D2,
Software/Wolfgang2/Neun_PLUS.bin.pck
E, 2200h, 0E000H, 0h, E000h, 'B', INVASION, 5A8,
Software/Wolfgang2/INVASION_neu.bin.pck
```

Erweiterung

Eigene Programme können recht einfach in den ROM aufgenommen werden, indem die Binärdatei um einen kleinen 32Byte-Header ergänzt einfach in den ROM hinzugefügt wird:

Die Programme sind im ROM hintereinander abgelegt. Man hat quasi eine 120K-Rom-Diskette. Auf einer xx00h-Adresse im Rom steht ein Header (s.u.), und dann folgt die Binärdatei (JTC-Datei ohne die ersten 80h Bytes). Auf der nächsten xx00h-Adresse folgt das nächste Programm usw.

Der Header ist 32 Byte lang und liegt auf einer xx00h-Adresse. Nach dem Header folgt das Programm. Der Header hat folgenden Aufbau (packedrom.asm):

```
org xx00h         ; header
db  0FAh, 0FAh    ; +0 Kennbytes
dw  aadr         ; +2 Anfangsadresse im RAM
dw  eadr         ; +4 Endadresse im RAM
dw  sadr         ; +6 Startadresse im RAM (oder 0000h - nichtstartend)
db  packed         ; +8 Bit 7=1 -> binary ist mit bitbuster_extreme
komprimiert
db  0,0,0,0,0,0    ; +9 6 Byte Reserve
db  Dateityp    ; +15 'M'-MC, 'B'-Tiny-MPBASIC, 'P'-BMP-Bild
db  "NAME    "    ; +16 genau 16 Zeichen, mit Leerzeichen aufgefüllt
binclude    programm.bin
```

Beispiel Maschinen-Code-Programm „HANOI“, aadr 8000h, gepackt

```
00006200h: FA FA 00 80 00 00 80 00 80 00 00 00 00 00 00 4D ;
úú.€...€€......M
00006210h: 48 41 4E 4F 49 20 20 20 20 20 20 20 20 20 20 20 ; HANOI
00006220h: 00 31 10 6C F7 7C A0 82 06 00 09 30 2C 84 3C 4C ;
.1.l÷| ,...0,„<L
00006230h: 4C 84 00 5C 79 1C 28 82 02 92 04 00 A0 E2 A0 E4 ;
L„.\y.(,.’.. â ä
00006240h: 1A F6 B0 33 04 DC 0F D6 83 46 02 59 00 D6 82 96 ;
.ö°3.Ü.Öƒ F.Y.Ö,–
...
```

Um Speicherplatz zu sparen, können Programme mit bitbuster_extreme komprimiert werden (bitbuster_extreme.exe programm.bin, erzeugt programm.bin.pck). Im Header muss in packed dann das Bit 7 gesetzt sein (also 80h).

BMP-Bilder müssen im Format

- Format:BMP,

- unkomprimiertGröße:256×192
- PixelFarben:2 (schwarz-weiß)
- Größe:6206 Bytes

vorliegen, s.Bildbetrachter.pdf (WeRo07/2019)

Historie

23.08.2021 erste Version, auch mit packed depack_z8.asm

28.08.2021 neuer Typ P für Picture, 2 Bilder ergänzt; testbild.bin neu übersetzt, da orig. Adresse 2000 hier nicht funktioniert; wurmi_8 ersetzt durch neu gemailte Version

16.09.2021 Korrektur depack-Routine (Flags sichern in getbyte); Sprungverteiler am Bank-Anfang (s. Bank0.asm); neue Programmversionen STEINE,FROSCH,JUMPER; Z9001-Zeichensatz (kompatibel zur wero-Bank)

17.09.2021 neuer Start von BASIC-Programmen (Sicherung Stack); Entpackroutine an anderer Stelle im RAM, da STEINE etc. den vorher genutzten Bereich brauchen

18.11.2021 neu: Z8671 BASIC/DEBUG; Start nun mit Xn möglich, dann wird gleich Programm Nr n gestartet (ohne vorheriges DIR)

19.11.2021 neu: Lauflicht. Steine (die erweiterte Version) geht jetzt wieder. Ich nutz jetzt den Kassettenpuffer+Teile des Stackbereichs für die Entpack-Routine. Damit ist der ganze RAM nutzbar. Zuletzt war 9F00..9FFF belegt, aber genau dieser Bereich wurde für die erweiterte Steine-Version genutzt.

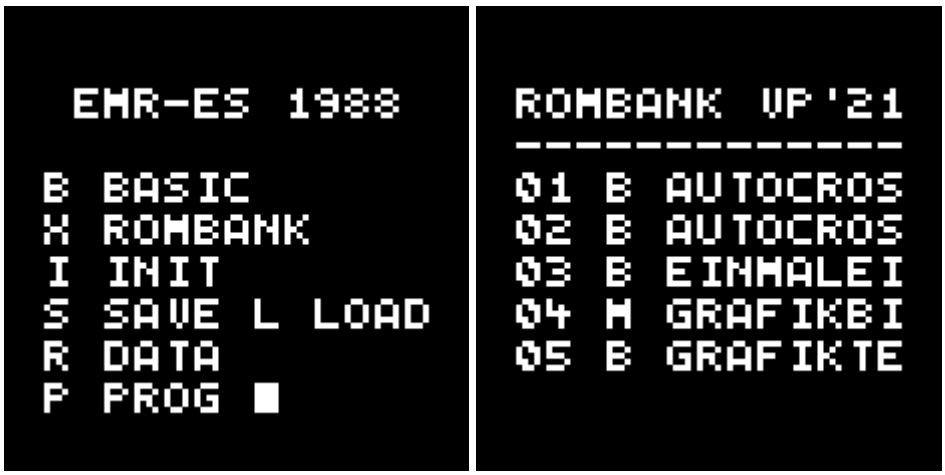
14.12.2021 Z8671-Basic um Assembler ergänzt

17.02.2022 OTHELLO und Mine-DEMO aufgenommen.

21.03.2022 FLOAT-Test aufgenommen. Jetzt ist die ROM-Bank 100% voll!

ES1988

Nach der Version für ES4.0 (s.u.) habe ich Software auch ans ES1988 angepasst.



1. Es wird das ES1988-System **mit 2 ROMs** erwartet (2. ROM auf Steckplatz 2, 2000h-27FFh) mit 8K RAM auf Steckplatz 1 (E000-FFFFh)
2. 128kByte ROM
3. ROM-Modul für Steckplatz 3
4. 4000h-5FFFh (8 kByte)
5. Angepasstes Betriebssystem ES 1988 mit Routinen zur Bankumschaltung etc.
6. Start des Moduls mit X (Bank 0, Sprung zu Adresse 4000h)

Mit einer Leitungen ist dann das Modul mit der Hauptplatine zu verbinden:

```
Schritt -> P3.5 weiterzählen (nächste Bank)
```

Start mit X<Nr> bzw. XD, s.u.

Hinweis: **JTCEMU** Die Version 2.0/2.1 unterstützt die ROM-Bank für ES1988 nicht! Ich habe zum Entwickeln eine gepatchte Version genutzt.

Downloads

- rombank_es1988.zip

Meine ROM-Bank-Software, 15.10.2021, komplett incl. Quellcode und Programmen.

- jtcemu-21b.zip

JTCEMU 2.1b Volker Pohlers mit Erweiterung für ROMBank im ES1988

Unterschiede zur ES4.0-Version

Gegenüber der nachfolgend beschriebenen Version für das ES4.0 gibt es einige Unterschiede:

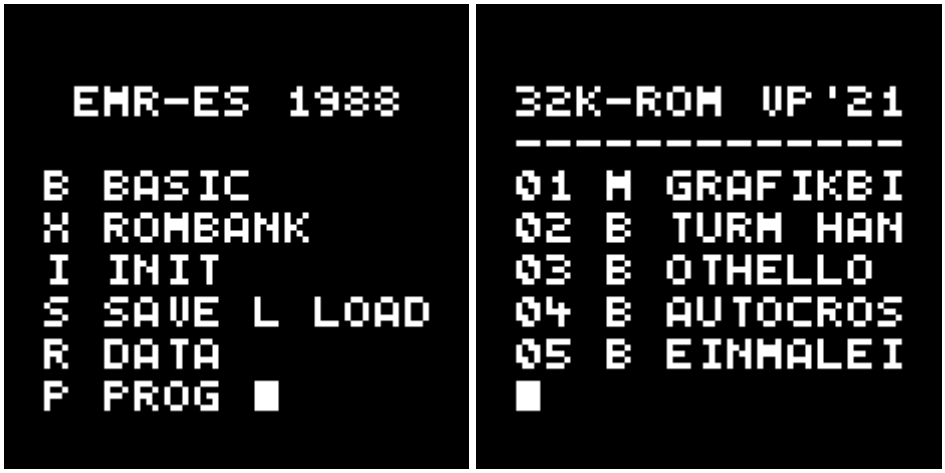
Der 2. System-ROM muss durch die im Paket beiliegende Version ersetzt werden. Die ROM-Bank wird im ES1988-Menü mit X gestartet. Wird anschließend D gedrückt, werden die in der ROM-Bank vorhandenen Programme zuerst aufgelistet. Anschließend wird die Nr. des zu ladenden Programms eingegeben. Anstelle D kann auch gleich eine Nr. angegeben werden, dann wird sofort geladen. Bei Basic-Programmen landet man im BASIC-Modus Mit RUN (+) wird das Programm gestartet. Maschinencodeprogramme wie FORTH starten direkt, andere wie die Grafikerweiterung werden nur in den RAM umgeladen.

Prinzipiell ist es möglich, Programme gepackt abzulegen. Aufgrund des geringen System-Speichers und der vorangigen Nutzung für die Grafikerweiterung wurde der Entpacker-Code in den Assembler-Quellen deaktiviert.

Bislang gibt es gerade mal 46 Programme für den TINY. Auch ungepackt ist der ROM nicht mal zur Hälfte gefüllt.

ES1988 32K-ROM

Die Software der ROM-Bank wurde erneut angepasst, um einen 32K-ROM (2x32k Speicher, e_mueller: JUTE COMPACT, TINY 2018, TINY MINI u.a.) mit Inhalt zu beleben. Auch wenn hier ROMBANK steht, ist obiges Modul nicht gemeint. Es wird kein extra Modul benötigt, es ist alles im 32K-System-ROM drin!



Von den 32K ROM stehen nur 18 kByte zur Verfügung, denn belegt sind 4K vom ES-System und 2K von TINY-BASIC, und der Bereich 6000h-7FFFh wird vom Adressbereich der Tastatur belegt. Dennoch konnten über 17 größere Programme (u.a. FORTH und Basic-Erweiterung sowie das Z8671-Basic) im ROM untergebracht werden:

ROM und Beschreibung
- rom_es1988_32k.zip

Quellcode etc. (10.12.2021)
- rombank_es32k.zip

Im Startmenü des ES1988 gibt es den neuen Buchstaben „X“ ROMDISK.

Nach „X“ wird die Eingabe der Programmnummer erwartet (z.B. aus tiny_es4k_32k.bin_inhalt.txt) Wird hier stattdessen „D“ gedrückt, werden die vorhandenen Programme gelistet und dann erst die zu ladende Nr abgefragt.

```
XD    Dir
Xnr   Lade Programm nr
```

Basic-Programme starten nicht automatisch, hier muss noch „+“/RUN gedrückt werden.

```
Nr. EPROM    Typ Name           aadr-eadr,sadr

  1 000012A0 M GRAFIKBIBL       FCA0-0000,0000
```

```
 2 00001320 B TURM HANOI       E000-0000,0000
 3 00001940 B OTHELLO          E000-0000,0000
 4 00002800 B AUTOCROSS+       E000-0000,0000
 5 00002A40 B EINMALEINS       E000-0000,0000
 6 00002CA0 B HASEWOLF+        E000-0000,0000
 7 00002F20 B KURZWECKER       E000-0000,0000
 8 00003000 B MAL-FIX          E000-0000,0000
 9 000031A0 B MONOPHON         E000-0000,0000
10 000032E0 B SCHIEBEPU.+      E000-0000,0000
11 00003580 B SOLO-HALMA       E000-0000,0000
12 00003820 B WOCHENTAG        E000-0000,0000
13 000039A0 B ZAHLENRATEN      E000-0000,0000
14 00003AA0 M FORTH            E000-0000,E000
15 00004A40 M TOR2             F800-0000,F800
16 00004D60 M STEINE           E000-0000,E000
17 000056C0 M Z8671BASIC       8000-0000,8000
```

Auf **1800h** steht die Basic-Erweiterung baserw2k

```
AT, CLS, INKEY, PRES, PSET, PTEST, READ, RESTORE
BASIC-Einbindung
1 PROC SETRR[8,%1800];REM Erweiterung einbinden
```

JTCEMU: Zum Testen im Emulator (www.jens-mueller.org/jtcemu/), Version 2.1, sind folgende Einstellungen nötig:

```
System: 4K ES1988
RAM: 8K (reichen) oder 32K (für das Z8671-Basic)
ROM: zusätzliche ROM-Bereiche: tiny_es4k_32k.bin auf Adresse %0000
```

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/tiny/erweiterungen/rombank?rev=1722320612

Last update: **2024/07/30 06:23**