

ES 4.0

Ausbaustufe III: großer Bildschirm, ES 4.0

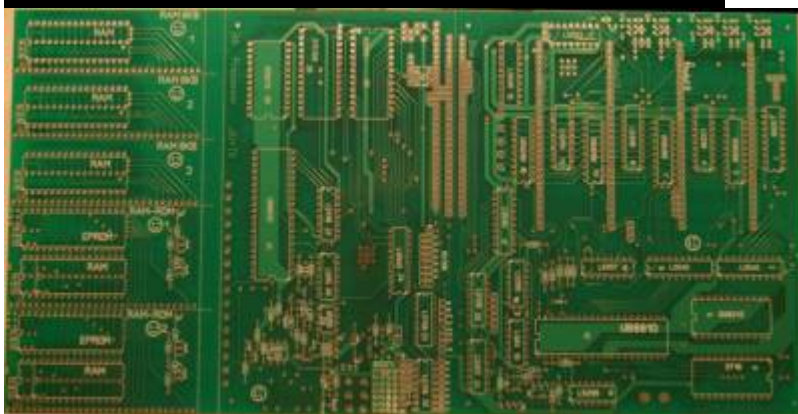
die höchste Ausbaustufe: Vollgrafik und Farbtüchtigkeit

- 192 Zeilen * 320 Punkte (24 Zeilen mit je 40 Buchstaben) monochrom
- oder 192 Zeilen * 640 Punkte (24 Zeilen mit je 80 Buchstaben) monochrom
- oder 192 Zeilen * 320 Punkte (24 Zeilen mit je 40 Buchstaben) mit 16 Farben je Bildpunkt.
- oder 192 Zeilen * 640 Punkte (24 Zeilen mit je 80 Buchstaben) mit 16 Farben je 2 Bildpunkte.
- 6K-Betriebssystem

Die Ausbaustufe III erweitert den TINY Ausbaustufe II um eine zusätzliche Grafikplatine mit eigenständiger Grafikerzeugung. Sie wurde in der Zeitschrift Jugend+Technik 6/1990 - 7/1990 vorgestellt. Die Bedienung ist in der Dokumentation

6k_es_40.pdf

beschrieben.



Zur Ausbaustufe III gibt es eine extra Website <https://www.tiny-computer.de>

Downloads

- Beschreibung
- ROM-Inhalte
- Quellcode System
- Anwendungen

originale ROMs

- 6k_es_40.zip

Beschreibung

- 6k_es_40.pdf

weitere Software:

- [ROM-Bank](#) Eine 128K-ROM-Bank mit allen Programmen zu diesem Computer
- [Basic-Erweiterung](#) Basic-Erweiterung (Grafik, Data, etc)
- [BASIC/DEBUG für ES4.0](#) Z8671-BASIC mit Assembler
- [FORTH](#) FORTH 83 (orig. H.Hoyer, korrigiert und angepasst)
- [Float für ES4.0](#) FLOAT für Basic (orig. H. Scheutzow, korrigiert und angepasst)

Links

- mehrfarbige Sprites: <https://github.com/haykonus/JU-TE-6K-Sprites>

Bedienung

J.Müller, Hilfe zu JTCEMU

Das Betriebssystem beinhaltet einen BASIC-Editor und einen Maschinenmonitor. Nach dem Einschalten, d.h. nach dem Starten des Emulators, und nach einem RESET meldet sich der BASIC-Editor.

BASIC-Editor

Ist als Prompt die Zeichenkette Edi zu sehen, befindet man sich im BASIC-Editor. Dieser kennt folgende Kommandos:

Kommando	Bedeutung
C	Setzt das BASIC-Programm nach einer STOP-Anweisung fort
E	Listet das BASIC-Programm zeilenweise auf, Mit der Leertaste wird die nächste Zeile angezeigt. Hinter dem E kann auch eine Zeilennummer angegeben werden. Dann wird das Programm ab dieser Zeilennummer angezeigt.

Kommando	Bedeutung
L	Lädt ein Programm von Kassette auf die Standardadresse für BASIC-Programme (%E000), Dazu muss die Emulation der Kassettenschnittstelle aktiviert werden. Wurde das Programm fehlerfrei geladen, meldet sich der BASIC-Editor mit einer Null zurück, sonst mit 255. (siehe auch Hinweise zum Laden von Kassette)
N	Löscht das BASIC-Programm aus dem Arbeitsspeicher
M	Ruft den Maschinenmonitor auf
R	Startet das BASIC-Programm
S	Speichert ein BASIC-Programm auf Kassette, Dazu muss die Emulation der Kassettenschnittstelle aktiviert werden.
?X	Gibt den Wert der Variable X aus
<Zahl>	Eingabe oder Löschen einer BASIC-Zeile, Wird hinter der Zahl nichts weiter angegeben, löscht der Editor die BASIC-Zeile mit dieser Zeilennummer. Anderenfalls wird eine BASIC-Zeile mit dieser Zeilennummer eingegeben. Dabei müssen die BASIC-Anweisungen in der Kurzform des komprimierten Speicherformats geschrieben werden.

Der Maschinenmonitor dient zum Anzeigen und Beschreiben von Speicherbereichen mit Hexadezimalzahlen und/oder ASCII-Zeichen. Die Tabelle zeigt die Kommandos des Maschinenmonitors, wobei alle einzugebenden Werte mit Ausnahme bei ASCII-Zeichen hexadezimal (groß geschrieben und ohne vorangestelltes Prozentzeichen) angegeben werden müssen.

Kommando	Bedeutung
,aaaa xx xx xx xx xx xx xx xx	Schreibt die max. 8 angegebenen Bytes in den Speicher ab Adresse aaaa
;aaaa cccccccccccccccc	Schreibt die max. 16 angegebenen ASCII-Zeichen in den Speicher ab Adresse aaaa
?aaaa bbbb	Berechnet aaaa+bbbb und aaaa-bbbb und gibt die beiden Ergebnisse aus
%aaaa	Gibt den vorzeichenbehafteten Dezimalwert der Hexadezimalzahl aaaa aus
#aaaa	Gibt den vorzeichenlosen Dezimalwert der Hexadezimalzahl aaaa aus
!rrxx	Schreibt den Wert xx in das Register rr
Faaaa nnnn xx	Füllt den Speicher ab Adresse aaaa mit nnnn Bytes xx
Haaaa	Zeigt ab Adresse aaaa 8 Bytes hexadezimal an
Jaaaa	Startet ein Maschinenprogramm auf Adresse aaaa mit einem CALL-Befehl, d.h., das Maschinenprogramm kann mit einem RET-Befehl beendet werden.
Laaaa	Lädt eine Datei von Kassette in den Arbeitsspeicher ab Adresse aaaa. Nach dem Laden wird die Anzahl der geladenen Bytes und der Rückgabecode (Null bedeutet kein Fehler) ausgegeben.
Q	Beendet den Maschinenmonitor
Maaaa bbbb nnnn	Verschiebt nnnn Bytes im Arbeitsspeicher von Adresse aaaa nach bbbb, Das funktioniert auch bei Überschneiden der Speicherbereiche.
Rrr	Gibt der Wert des Registers rr aus
Saaaa nnnn	Speichert nnnn Bytes ab der Adresse aaaa auf Kassette, Das zweite Argument gibt nicht die Endadresse sondern die Anzahl der Bytes an.

Es bedeuten:

aaaa bbbb nnnn 16-Bit-Hexadezimalzahlen
 rr xx 8-Bit-Hexadezimalzahlen
 c ASCII-Zeichen

Editierfunktionen Alle Eingaben erfolgen mit Hilfe eines Full-Screen-Editors. Dieser bietet komfortable Editierfunktionen, die in nachfolgender Tabelle gezeigt werden. In der ersten Spalte steht der hexadezimale Code der Funktion. Gibt man in einem Programm ein Zeichen mit diesem Code auf dem Bildschirm aus, wird die entsprechende Funktion ausgeführt. Für manche Funktionen gibt es auf der Tastatur des Emulator-Rechners keine speziellen Tasten. Diese Editierfunktionen können jedoch mit Hilfe einer CTRL- bzw. Strg-Tastenkombination benutzt werden (siehe letzte Spalte in der Tabelle).

Code	Name	Bedeutung	CTRL-Tastenkombination
01		Kursor links	CTRL-A
02		Kursor rechts	CTRL-B
03		Kursor hoch	CTRL-C
04		Kursor runter	CTRL-D
05	HOM	Kursor in linke obere Ecke	CTRL-E
06	SOL	Kursor an Zeilenanfang	CTRL-F
07	DEL	Zeichen unter Cursor löschen	CTRL-G
08	DBS	Zeichen links von Cursor löschen	CTRL-H
09	INS	Zeichen einfügen	CTRL-I
0A	LDE	Zeile löschen	CTRL-J
0B	LIN	Zeile einfügen	CTRL-K
0C	CLS	Bildschirm löschen	CTRL-L
0D	RET	Eingabe abschließen	CTRL-M
0E	ESC	Nächstes Zeichen nicht als Steuerzeichen interpretieren	CTRL-N

BASIC-Programme werden in der internen MPBASIC-Form eingegeben, d.h. für die Basic-Befehle werden keine Schlüsselwörter mehr verwendet, sondern nur Buchstaben. Damit kommt man aber nach kurzer Gewöhnung gut zurecht. Hinter dem Buchstaben (also Basic-Befehl) darf kein Leerzeichen stehen.

Hardware

Es gibt zwei Varianten:

- der normale TINY + zusätzliches Video-Modul
- die Kompaktversion, die beide Teile auf einer Leiterplatte enthält

benötigte zusätzliche Schaltkreise:

Außer dem 40poligen Einchip-Mikrorechner (Es eignen sich die Typen UB 8810, UB 8811, UB 8830, UB 8831, UB 8860, UB 8861, U 8611, UL 8611, U 8611/1 und UL 8611/1, da mit externem Takt und Test-ROM-Betrieb gearbeitet wird) und einem Speichermodul mit 8 Kbyte RAM enthält die höher auflösende Bildschirmsteuerung bei minimalem Ausbau (24 Zeilen mit je 40 Zeichen monochrom) endgültig folgende Schaltkreise:

1 x U 2716 C
 1 x DL 004 D
 2 x DL 257 D
 2 x DL 295 D

1 x DL 299 D
4 x DL 540 D
1 x DS 2805 D
1 x DS 8212 D
1 x DS 8286 D

Der Bildwiederholpeicher ist mit den bekannten Leiterplatten modular aufgebaut. Ein Speichermodul mit 8 Kbyte RAM (EPROM überflüssig) gestattet die monochrome Darstellung von 40 Zeichen (8 x 8 Bildpunkte) je Zeile. Zwei solcher Module lassen sogar 80 dieser Zeichen je Zeile erzeugen. Mit vier Modulen a 8 Kbyte RAM ist die volle Ausbaustufe der neuen Bildschirmsteuerung erreicht: Bei 40 Zeichen je Zeile ist jeder Bildpunkt einzeln in 16 Farben darstellbar.

Video-Modul

2022 aufgelegt:

eigenständiges Video-Modul von Boert:

- <https://www.robotrontechnik.de/html/forum/thwb/showtopic.php?threadid=19650>
- https://github.com/boert/JU-TE-Computer/tree/main/Erweiterungen/Videoerweiterung_2021

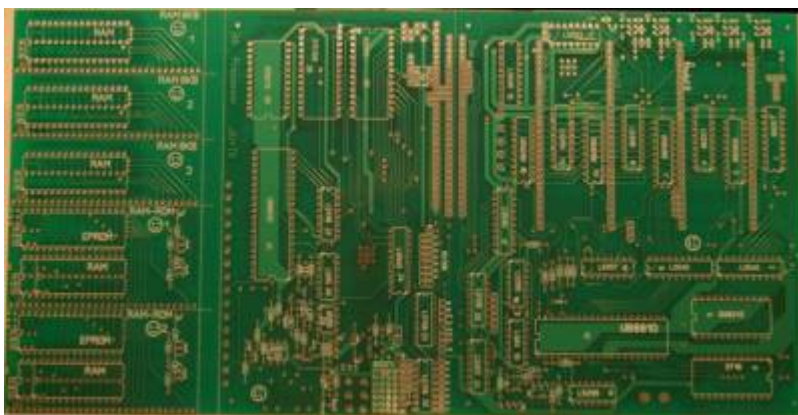
Kompaktversion

2012 gab es im [robotrontechnik-Forum](#) dank Leiterplatten von W. Harwardt neue Aufbauten des 6K-Tiny. Ich habe hier Informationen gesammelt und ein wenig sortiert

<https://buebchen.jimdo.com/selbst-gebaut-diy-homebrew-feito-por-mim/ju-te-computer/>

225 x 157 mm

Kleine Module für RAMs 6264



Nachbauhinweise

Unter Dokumentation zum Tiny findet man eigentlich alles. Dabei die Doku Teil 3 (3. von oben in der Liste), hier die Seiten von 40-51 und das Begleitheft zum ES4.0 findest du ganz unten in der Liste.

MB104 ist richtig und ein Optokoppler. Der Spannungsregler ist m.E. falsch beschriftet. Ist auch egal. Ich bestücke den Spannungsregler (nehme dabei einen 3A-Typ ; MC78T05) extern wegen der Kühlung. Denke daran, das der angegebene Brückengleichrichter für die gesamte Leiterplatte nicht ausreicht bzgl. Stromfestigkeit (sollte dann auch ein 3A-Typ sein). Steht aber auch in der Dokumentation alles drin ! Wichtig, zuerst durchlesen . Den Elko am Kollektor des SC308 (Batteriestützung) bestückst du erstmal nicht oder du musst beim Einlöten aufpassen wo du den Minuspol vom Elko anschließt. Auf dem Layout ist an dieser Stelle nämlich ein Durchkontakt zu 5P angegeben, doch der Elko muss einseitig an Masse. Ist aber ein Durchkontakt dafür in der Nähe zu finden. Und noch eins; die Kassettensteuerung so bestücken wie im Anhang vom Begleitheft zum ES4.0 angegeben und nicht wie es in der Schaltung steht. (Einige Werte ändern sich von R und C bzw. entfallen).

-

1. D11(DS8286)Pin 11 geht zu einem Widerstand der zwischen D10 und D8 endet und nicht wie bei dir an dem MODUL-B?

Das Richtungssignal kommt vom IC 21 dem DL257 und geht auf das dritte BUS-Pin das mit R/W bezeichnet ist. Auf der Bildwiederholpeicher Schaltung ist das Signal mit DIR bezeichnet. Also laut Schaltung in Ordnung!

2. Auf der Videoplatine fehlt für die Stromversorgung ein + und - (5V 1A)

3. Einige Bohrungen werden nicht passe zu klein (Audiobuchse)

Die Buchse wird über Drähte angeschlossen.

4. Keine Markierung der Modulsteckplätze mit Pin 1 , Module haben auch keine Bezeichnung des Pin 1

Beim Betrachten der Leiterführung sollte einem die doppelte Belegung der +5V Schiene auffallen. Für das auf dem Prozessorteil befindliche Modul kann über Lötbrücken direkte oder Batterie-Versorgung ausgewählt werden. Damit ist die Orientierung der Modulplatinen feststellbar.

5. D22 ist zu dicht an den rechts und links angrenzenden Steckkontakten dran.

Der 10polige Steckverbinder ist kurz vor der Fertigstellung der Platine dazugekommen. Damit kann das PS2 Interface von E.Müller für den Tiny 2009 angeschlossen werden. Es sind extra lange Pfostenfeldstecker nötig. Damit ist die auf der anderen Seite vorgesehene Pfostenfeldsteckerreihe überflüssig.

Details

(aus dem robotrontechnik-Forum)

Farbige Sprites:

Software-Sprites (sog. „Shapes“)

Bei der Darstellung von Pixeln wird je nach Belegung der „M“-Bits in der Sprite-Definition zwischen Farbe und Maske unterschieden.

Bit:	7	6	5	4	3	2	1	0
	---	---	---	---	---	---	---	---
Funktion:	M	M	M	M	R	G	B	H

Der Code ist komplette Eigenentwicklung von HeikoS. Pre-Shift wird im ES4.0 nicht verwendet für die einfarbigen Sprites. Das wird dort immer zur Laufzeit berechnet, was Zeit kostet, aber damals natürlich berechtigt war, da eher wenig RAM in den JuTe's vorhanden war.

Es gibt 4 Farbebenen, je 8K für R,G,B,H, eingeblendet in einen 8K-Slot im RAM (4000-5FFF) der Host-CPU.

Maske und Pixel-Farbe sind in einem Byte, damit ich das nicht getrennt ablegen muss. Die INIT-Routine testet beim Erstellen der pre-shifted MAPS einfach, ob min. 1 Bit des High-Nibbles 1 ist. Dann ist das ein Mask-Bit. Wenn das High-Nibble 0 ist, dann sind die unteren 4 Bits der Farbcode für die 16 Farben des Bits.

Man kann also mit dem High-Nibble keine Transparenz einstellen - kann der JuTe einfach nicht und war auch nie vorgesehen

From:

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

Permanent link:

<https://hc-ddr.hucki.net/wiki/doku.php/tiny/es40>

Last update: **2024/02/05 07:19**

