

# "ROM-Bank 2000" für Ju+TeC-6k

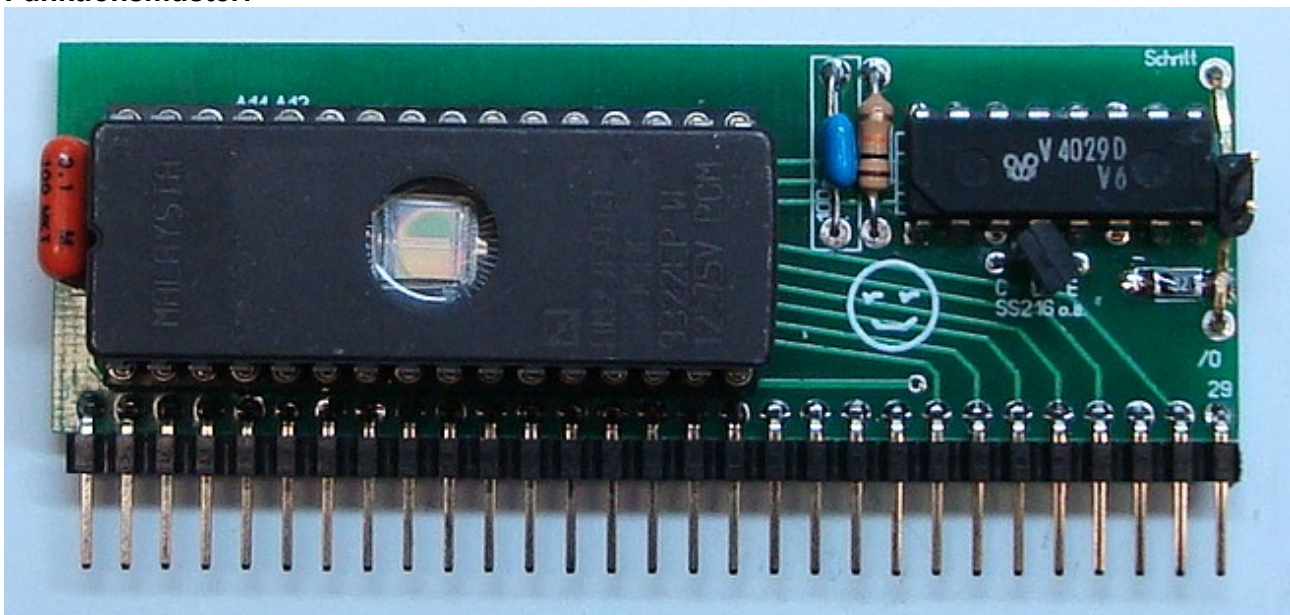
WeRo, V2.2, Stand: 03.06.2019

**Achtung: Benötigt Hard- und Softwareänderungen, auf JTC-Emulator nicht lauffähig!**

## Prinzip:

- Auf dem Steckplatz 1 ist ein ROM im Adressbereich %2000...%3FFF (8k\*8) einsetzbar.
- Die Idee besteht darin, statt des 8k-ROMS einen "großen" ROM (128k\*8)<sup>1</sup> zu benutzen, dessen obere Adressbits A13...A16 umgeschaltet werden. Damit sind 16 Bänke möglich. Die Umschaltung kann auf verschiedenste Weise erfolgen.
- Um den Steueraufwand zu minimieren, wurde zur Adresseinstellung ein Binärzähler direkt auf dem EPROM-Modul installiert. Mit zwei Leitungen ist dann das Modul mit der Hauptplatine zu verbinden:  
/0 → P3.5 Zählerreset  
Schritt → P3.7 weiterzählen (nächste Bank)
- Um also z.B. die 5. Bank einzustellen, sind ein Zählerreset und 5 Schrittpulse zu erzeugen. Dazu ist das ES4.0 etwas zu ergänzen (siehe Anlage). Zum Einblenden einer Bank ist deren Nummer in Register %54 zu laden und dann die Schalt-Routine mit CALL %092F aufzurufen.
- Einziger Nachteil: Es können direkt im EPROM auf %2000 keine Programme laufen, die o.a. Portleitungen benutzen.

## Funktionsmuster:



Die Platinenvorlage steht im Format "Sprint Layout" zur Verfügung. Einige wenige Musterplatinen sind bei "PC-OPA" erhältlich.

<sup>1</sup> EPROM (z.B. M27C010), EEPROM (z.B. AT28C010) oder Flash-ROM (z.B. W29C011)

## Software

Um eine bestimmte Bank zu benutzen (d.h. einen 8k-EPROM-Teilbereich auf %2000...%3FFF einzublenden), ist die Nummer der Bank (0...15) an eine Schaltroutine zu übergeben. Dazu wird das (bislang freie) **Register %54** verwendet. Die Schaltroutine ist in Form einer Erweiterung im ES4.0-EPROM untergebracht: **CALL %092F**.

Beispiele:

BASIC	Einblenden der gewünschten Bank	Assembler
<pre>... 100 OSETR[%54,n] 110 C%092F ...</pre>	n=Banknummer 0...15	<pre>... LD    %54,#n CALL  %092F ...</pre>

Je nach Inhalt der Bank sind nun verschiedene Möglichkeiten gegeben:

- a) Direktstart eines Maschinenprogramms (z.B. Bildbetrachter) in der Bank

BASIC	Direktstart Maschinenprogramm	Assembler
<pre>100 OSETR[%54,n] 110 C%092F</pre>	Bank-Auswahl	<pre>LD    %54,#n CALL  %092F</pre>
<pre>120 C%2000 ...</pre>	Sprung zum Anfang eines Bildbetrachters, der muss mit RET enden	<pre>CALL %2000 ...</pre>

- b) Ausführung eines in der Bank abgelegten BASIC-Programms

BASIC	BASIC-Programm ausführen	Assembler
-	<p>Die Anfangsadresse aaaa des BASIC-Programms in der Bank ist in R6/R7 einzutragen.</p> <p>Sind Adressbezüge vorhanden, so muss nach %E000 umkopiert und dann mit aaaa=%E000 gestartet werden</p>	<pre>... SRP    #0 LD     R6,#HIGH aaaa LD     R7,#LOW  aaaa CLR    R8 CLR    R9 SRP    #%10 CALL   %07FD ...</pre>

- c) Zugriff auf bestimmte Daten, z.B. einen Text. Um den aufzurufen und anzuzeigen:

BASIC	Text aus Bank anzeigen	Assembler
<pre>100 LA=%aaaa 110 X=GETEB[A] 120 PX 130 LA=A+1 140 ...</pre>	Anfangsadresse aaaa des Textes sowie Länge (oder eindeutiges Endezeichen) müssen bekannt sein	<pre>LD     R0,#HIGH aaaa LD     R1,#LOW  aaaa LD     %15,@R0 CALL   %0818 INCW   RR0 ...</pre>

- d) Aufbau eines Bankmenüs zur Auswahl eines Programms  
→ siehe dazu Prinzip-Beispiel in Anlage 4
- e) u.v.a. Möglichkeiten

Wird eine Bank durch RET im Code verlassen oder in ihrem Menü 'Monitor' aufgerufen, so bleibt sie immer noch eingeblendet und kann mit J2000 erneut aufgerufen werden.

#### Neues Mon-Kommando 'X':

Über einen neuen **Menübuchstaben 'X'** (im Mon) wird der Aufruf von Bank0 realisiert (Rücksetzen Bankzähler und Sprung zu %2000). Damit ist eine Funktion ähnlich "Paket-X" (AC1, LLC2) möglich. Aus Bank0 kann dann zu den einzelnen anderen Banken/Programmen verzweigt werden.

Achtung: Auf das Vorhandensein von "sinnvollem" Code auf %2000 wird nicht getestet! Ist gar kein EPROM-Modul auf %2000 gesteckt, so führt die Betätigung von 'X' i.d.R. zum Einfrieren des Rechners (RESET nötig).

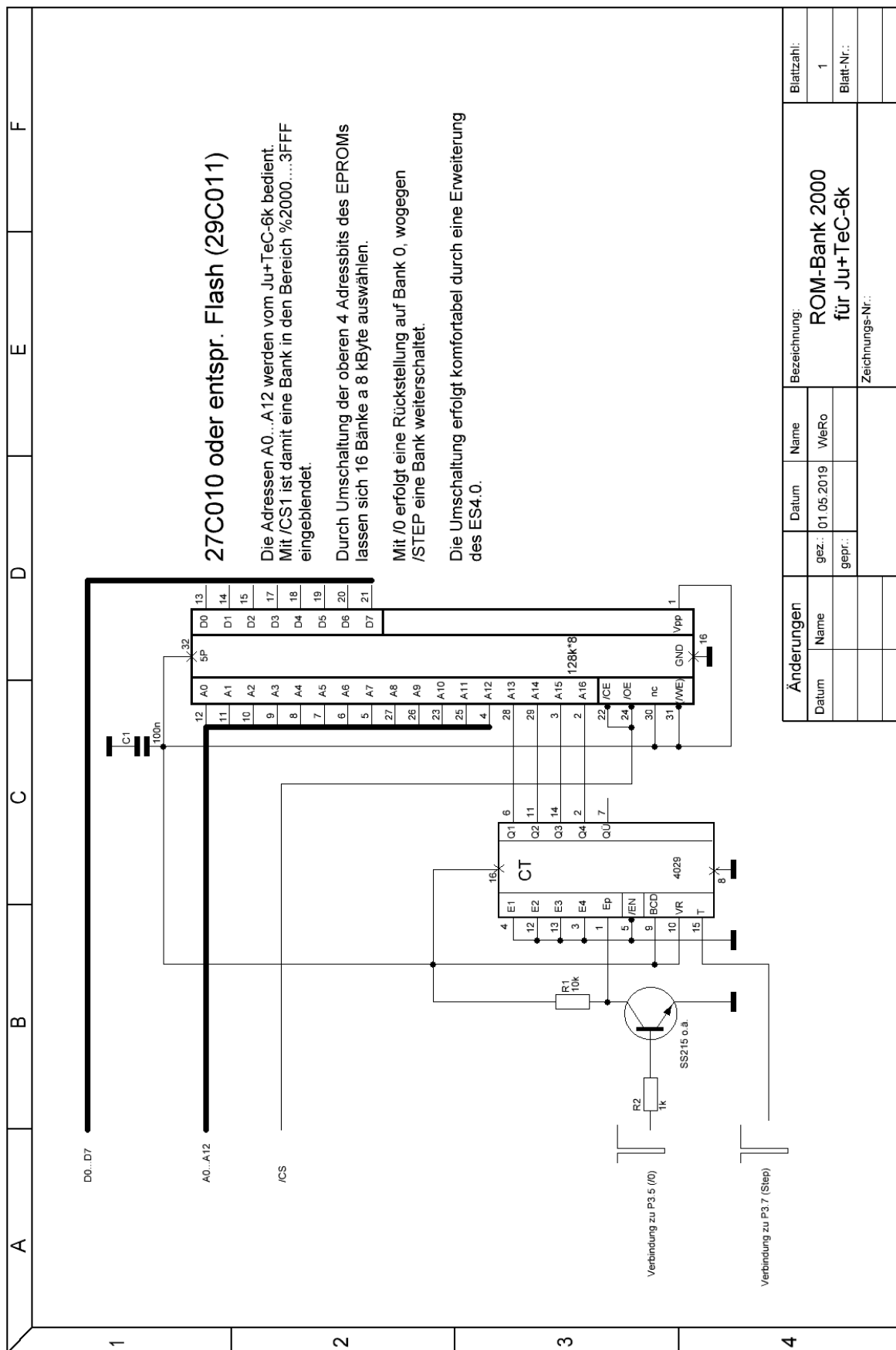
#### Hinweise zur Erstellung eigener Bankinhalte:

##### Bank0:

- stellt eine Übersicht mit einer Kurz-Beschreibung des jeweiligen Bankinhaltes dar
- verzweigt je nach Kennziffer zu Bank1...15. Prinzipiell müssen bei Änderungen anderer Banken hier nur die Texte der Bankinhalte angepasst werden.
- enthält darüber hinaus den leicht modifizierten Zeichensatz des Z9001, Aktivierung durch **CALL %0A4A** (erweiterter ES4.0-EPROM nötig, siehe gesonderte Beschreibung)

##### Allgemein:

- Anordnung der Banken im EPROM: Bank0=%0000, Bank1=%2000, Bank2=%4000,...
- Startadresse der Programme in jeder Bank ist immer %2000 (ORG %2000)
- Rückkehr zum Monitor mit "RET" oder auch Soft-RESET (Sprung zu %0812).
- Um BASIC-Programme aufzunehmen, gibt es zwei Möglichkeiten (siehe Beispiel):
  1. Umkopieren aus der jeweiligen EPROM-Bank nach %E000 und dann entweder Direktstart mit der vorgeschriebenen Methode (siehe Beispiel) oder nach Verlassen der Bank und Wechsel in Edi mit 'R' zu starten.
  2. Direktstart mit "verbogenem" Programmanfang (nicht möglich, wenn das Basic-Programm Adressbezüge ab %E000 aufweist)
- Müssen Programme/Daten >8kB aufgenommen werden, so belegen sie zwei oder mehr Banken. Die Daten sind daher durch geeignete Programmierung entsprechend aufzuteilen und nach Wechsel in die entsprechende Nachfolgebank von dort weiter zu kopieren.
- Dadurch, dass die eigentliche Bankumschaltung im ES-EPROM erledigt wird, kann diese sogar aus einer momentan aktuell eingeblendeten Bank heraus erfolgen.
- Ein Test der jeweiligen Bank kann (mit Ausnahme der Bankwechselfunktionen) auch mit dem JTC-Emulator erfolgen.



## Anlage 2: Erweiterung des ES4.0 für die ROM-Bank

Alle Adressangaben sind Originaladressen im Speicher. EPROM-Adressen: adr-%800

Adr	Code	Mnemo	Bedeutung
<b>Bankschaltroutine: %54 enthält Bank-Nummer = Anzahl Impulse</b>			
092F	56 03 DF	AND P3, #%DF	Impuls an P3.5: Zähler auf 0
0932	46 03 20	OR P3, #%20	
0935	A6 54 00	CP %54, #0	fertig?
0938	6B 0A	JR Z, %0944	ja!
093A	56 03 7F	AND P3, #%7F	Impuls an P3.7: Zähler weiter
093D	46 03 80	OR P3, #%80	
0940	00 54	DEC %54	nächsten Impuls
0942	8B F1	JR %0935	
0944	FF	NOP	kurz warten
0945	AF	RET	
<b>Neues Monitorkommando 'X' definieren: Aufruf Bank0</b>			
0C66	58	DB 'X'	Monitorbuchstabe
0C67	0D 87	DW %0D87	Startadresse
<b>Inhalt Kommando 'X':</b>			
0D87	B0 54	CLR %54	Bank 0 einstellen
0D89	D6 09 2F	CALL %092F	
0D8C	AC 20	LD R10, #20	Feste Sprungadresse %2000 (Ende dort mit RET!)
0D8E	B0 EB	CLR R11	
0D90	CF	RCF	Clear Carry
0D91	8B 61	JR %0DF4	(Zwischensprung, da kein Platz mehr)
hier weiter mit 'X'			
0DF4	E6 58 FF	LD %58, #%FF	wegen FSE (BASIC-INPUT!)
0DF6	8D 0D 45	JP %0D45	weiter in 'J'-Routine, Sprung zu...
<b>Bank Schalten+Starten</b>			
1C08	D6 09 2F	CALL %092F	Bank einstellen
1C0B	8D 20 00	JP %2000	und starten

### Anlage3: Inhalt des Muster-EPROMS

```
EPROMBANK 0: UEBERSICHT
=====
1=TOOLS: ASDIS/FLOAT/KCTRAM5/53004
2=
3=FORTH
4=BASIC1: TROPFEN/WUERFELN/ALIEN
5=BASIC2: BOWLING/FLIEGER/NEUN GEWINNT
6=BASIC3: HASE&WOLF/MASTERMIND
      REAKTIONSTEST/SCHMUGGLER
7=BASIC4: 17+4/SCARABAEUS
8=MC-SPIELE: MINE/TETRIS
9=MC-SPIELE: OTHELLO/SOKOBAN
A=MC-SPIEL : STEINE
B=BILD: ALBERT
C=BILD: ARTHUR
D=BILD: BUDEL
E=BILD: CHE
F=DEMOS: GRAFIK/FARBE/TESTBILD

M=MONITOR
KENNZIFFER? ■
```

Bank0=Inhaltsverzeichnis

Jede Bank kann (muss aber nicht) ein eigenes Menü enthalten.

FORTH, STEINE sowie die Bilder enthalten kein Menü.

- FORTH bzw. STEINE wird sofort gestartet.
- Bei den Bildern wird das auf %2000 befindliche Betrachterprogramm aufgerufen, welches das im Anschluss befindliche Bild anzeigt.

### Programm-Quellen:

- Software von Begleitkassette zu *Harun Scheutzows* Sonderdruck "Betriebssystem ES 4.0 für den JU+TE-Computer"
- BASIC-Programme aus der Feder von "PC-Opa" Wolfgang, MC-Programme von Thomas Singer
- ES4.0Erweiterung, ROM-Bank, Aufbereitung der Pixelgrafiken (aus dem Net) sowie Bildbetrachter dafür: WeRo

#### Anlage 4: Prinzip Basic-Programme mit Menü

```

ORG    %2000
CALL  %082D          ;PRISTRI
DB     %0C
DB     "EPROMBANK 4: BASIC",%0D
DB     "=====",%0D,%0D
DB     "T=TROPFEN FANGEN",%0D
DB     "W=WUERFELN",%0D
DB     "A=ALIENS",%0D,%0D
DB     "0=ANDERE BANK",%0D
DB     "M=MONITOR",%0D
DB     "AUSWAHL:",0

TAST:
CALL  %081E          ;WKEY → %13
CP     %13,#'T'       ;EINGABE PRÜFEN:
JP     Z,TROPFEN
CP...   ;ÜBRIGE PROGRAMME
CP     %13,#'0'       ;
JR     Z,NEUBANK      ;BANKWECHSEL
CP     %13,#'M'       ;
JR     Z,RAUS         ;ENDE
JR     TAST           ;FALSCH EINGABE
;-----
RAUS:  RET            ;ZURÜCK ZU MON
;-----
NEUBANK:
CLR     %54           ;BANK 0
JP      %1C08         ;AUFRUFEN
;-----
TROPFEN:
SRP     #0
LD      R6,#HIGH T_DAT ;ANFANG
LD      R7,#LOW  T_DAT
JP      START
;-----
WUERFELN:
;DIREKTSTART WIE OBEN MIT VERBIEGEN DES
PROGRAMMANFANGS AUF ADRESSE IM EPROM GEHT
NICHT, WENN BASIC-PROGRAMM ADRESSBEZÜGE
NACH %E000... ENTHÄLT. DANN UMKOPIEREN
NACH ADR. E000 NÖTIG!
SRP     #%70
LD      R0,#HIGH W_DAT ;QUELLE
LD      R1,#LOW  W_DAT

```

```

LD      R4,#HIGH (A_DAT-W_DAT) ;LÄNGE
LD      R5,#LOW  (A_DAT-W_DAT)
JP      UMKOPIEREN
;-----
ALIEN:
SRP     #%70
LD      R0,#HIGH A_DAT ;ANF.
LD      R1,#LOW  A_DAT
LD      R4,#HIGH (ENDE-A_DAT)
LD      R5,#LOW  (ENDE-A_DAT)
JP      UMKOPIEREN
;-----
UMKOPIEREN:
;QUELLE: VORHER IN IN R0/R1 ABLEGEN!
LD      R2,##E0      ;ZIEL IMMER %E000
LD      R3,#0
SCHLEIFE:
LDE     R6,@RR0      ;BYTE HOLEN
LDE     @RR2,R6      ;ABLEGEN
INCW    RR0          ;NÄCHSTE EPROM
INCW    RR2          ;NÄCHSTE RAM
DECW    RR4          ;LÄNGE-1
JR      NZ,SCHLEIFE ;
;BASIC-UMGEBUNG SETZEN:
SRP     #0
LD      R6,##E0      ;ANFANG
LD      R7,#0        ;ADR. E000
START:  CLR    R8      ;KEINE PROCTAB
CLR     R9
SRP     #%10
CALL    %07FD        ;INTERPRETER
;STARTEN
JP      RAUS          ;BASIC FERTIG
;-----
;ES FOLGEN DIE BASIC-PROGRAMME
T_DAT:
INCLUDE "TROPFEN.DAT"
W_DAT:
INCLUDE "WURF.DAT"
A_DAT:
INCLUDE "ALIEN.DAT"
ENDE:

```

Gelb markiert ist die "Autostart-Routine" für BASIC-Programme gem. Vorschrift. Ein Sprung nach Edi und 'R' ist damit nicht nötig.

Da der benutzte Assembler ZDS nur die Einbindung von Quelldateien ermöglicht, sind die BASIC-Programme zunächst in Daten umzuwandeln, z.B.:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	80	01	4D	2A	2A	20	4D	45	49	4E	20	45	52	53	54	4C	! .M** MEIN ERSTL
00000010	49	4E	47	53	57	45	52	4B	20	46	1F	52	20	44	45	4E	INGSWERK F.R DEN
00000020	20	4A	55	54	45	20	36	4B	2A	0D	80	02	4D	2A	2A	20	JUTE 6K*. ! .M**
00000030	57	4F	4C	46	47	41	4E	47	20	4B	4F	54	54	2C	20	41	WOLFGANG KOTT, A

→ DB %80,%01,%4D,%2A,...  
→ Abspeichern als name.dat und  
→ Einbinden in ASM-Quelltext mit .include "name.dat"