

**Dr. Helmut Hoyer**

## **JU+TE Computer selbst gebaut Teil III**



**Jugend+Technik 8/1989 - 2/1991**

## Inhalt

Schnelles EPROM-Programmieren.....	3
Höher integrierte EPROM?.....	7
Turm von Hanoi .....	8
U2716C auf Modulplatz 2 .....	11
Datenverlust trotz Batteriestütze .....	11
Betriebssystem -Varianten.....	11
Serielle Schnittstellen .....	13
Computer im Palast.....	16
Osziilograph mit dem JU+TE-Computer .....	17
Korrektur: Streifen beim JU+TE-Computer .....	18
JU+TE-Computer-Softwareip: Landstraße.....	19
A/D Wandler C520D am JU+TE-Computer.....	21
Leservorschlag: Meßvorsatz für JU+TE-Computer .....	24
EPROM-Brennen leichtgemacht .....	24
FORTH-83-System für JU+TE-Selbstbau-Computer.....	26
Software per Kasette.....	29
Höher auflösende Bildschirmsteuerung .....	29
Achtung! Computer-Treff! .....	30
Solo-Halma.....	31
Schiebepuzzle .....	32
Fertigtastaturen am JU+TE-Computer.....	35
Höher auflösende Bildschirmsteuerung.....	37
Erfahrungen mit dem EPROMMER.....	38
JU+TE-Computer als Taschenrechner .....	38
Sieb des Eratosthenes .....	38
Vollgrafik und Farbe .....	40
Selbstbau-Computer kompakt.....	48
TINY-BASIC lernt Rechnen .....	52
Leserbriefseiten .....	54

## **Schnelles EPROM-Programmieren**

In JU+TE 10/88, S. 786 ff. haben wir eine Schaltung vorgestellt, die bei entsprechender Steuerung durch einen Computer das Programmieren von EPROM (löschrare programmierbare Festwertspeicher) bis zu einer Kapazität von 128 Kbit gestattet. Das zugehörige BASIC-Programm (JU+TE 11/88) für den JU+TE-Computer beschränkt deren Anwendung jedoch auf den EPROM-Typ U 2716 C. Heute ergänzen wir diese Veröffentlichung mit einem Programm, das sich auch für die Typen U 2764 C und U 27128 C eignet. Das Standardverfahren benötigt 50 ms zum Programmieren jedes Bytes (1/4 Stunde für einen U 27128 C). Aus Zeitgründen wird verbreitet ein „intelligenter“ Algorithmus angewendet, der sich auf die tatsächlich nötige Programmierdauer beschränkt. Die Datensicherheit soll dabei nicht leiden, die Lebensdauer der Schaltkreise sogar verbessert werden. Das Prinzip besteht im Überprüfen des Programmiererfolges nach jeweils 0,5 bis 1 ms. Ist der gewünschte Inhalt auf allen acht Bitpositionen erreicht, wird zur Sicherheit noch ein Programmierimpuls der doppelten bis vierfachen Länge angeschlossen. Dabei verringert sich die insgesamt nötige Zeit auf etwa zehn Prozent (je nach Exemplar) der beim Standardverfahren üblichen. Das Erhöhen der Betriebsspannung Vcc des EPROM auf 6 V verstärkt den

Effekt, ist aber nicht unbedingt erforderlich. Um die schnellen Pegelwechsel bei diesem Verfahren zu gewährleisten, muß in der Schaltung aus JU+TE 10/88, S. 788 der 0,47 µF-Elektrolytkondensator am Anschluß A11/Vpp entfernt werden. Das gestattet die Anwendung des intelligenten Algorithmus auch zum Programmieren des Typs U 2716 C. Der Vollständigkeit wegen enthält unser Programmbeispiel auch die Möglichkeit, die für den U 2732 C nötigen Signale zu erzeugen. Dafür ist die veröffentlichte Schaltung jedoch nicht ausgelegt. Dem Dialog mit dem Bediener dient das in Abb. 1 dargestellte BASIC-Programm. Diese Sprache eignet sich hierfür gut, gestattet aber nicht die für den intelligenten Algorithmus nötige Rechengeschwindigkeit. Deshalb werden diese Aktivitäten mit Maschinenprogrammen realisiert. Allgemein erreichen wir mit einer solchen Aufgabenteilung zwischen den nutzbaren Sprachniveaus relativ übersichtliche und schnell zu entwickelnde Programme. Das Maschinen-Unterprogramm in Abb. 2 lädt die EPROM-Adresse aus den Arbeitsregistern r0 und r1 in die beiden Auffangregister DS 8212 D des EPROM-Programmierzusatzes. Da neben den höheren Adressen auch Steuersignale zur EPROM-Freigabe (/CE) und zum Durchschalten der Programmierspannung zum Anschluß A11/Vpp gespeichert werden, beginnt dieses Programm mit einer Typabhängigen Manipulation der höheren Adreßhälfte. Das

Unterscheiden erfolgt anhand des Arbeitsregisters r14, dessen Inhalt mit %08 einen U 2716 C, mit %10 einen U 2732 C, mit %20 einen U 2764 C und mit %40 einen U 27128 C kennzeichnet. Außerdem erwartet dieses Unterprogramm in r15 1-Belegung auf den Positionen, die zusätzlich im höheren Byte zu setzen sind (zum Programmieren von U 2716 C und U 2732 C). An der Marke AD3 wartet das Programm bei einem Pegelwechsel von A13/Vcc das Umladen des Stützelkos (0,47 µF) ab. Dem Lesen eines Bytes vom EPROM dient das Unterprogramm in Abb. 3. Nach Einstellen der Adresse werden Port 2 in Eingaberichtung initialisiert, die Ausgabe-Freigabe des EPROM (/OE) aktiviert und das gelesene Byte im Arbeitsregister r4 abgelegt. Die nop-Befehle lassen Zeit für das Einschwingen von /OE. Nach Rückstellen von Port 2 auf Ausgabe folgt der Rücksprung mit passiver Steuersignalbelegung. Das Programm in Abb. 4 blockiert die Bildinterrupts und setzt den Registerpointer auf #%20, so daß der die BASIC-Variablen A bis H speichernde Bereich als Arbeitsregistersatz vereinbart wird. Die Anfangsadressen von EPROM und RAM entnimmt das Programm den Variablen O und M (vgl. Abb. 1). Der Vergleich des höheren Teils der EPROM-Endadresse (Register %3E) mit dem Kennbyte von r14 setzt das C-Flag, wenn die Kapazität des Festwertspeichers nicht überschritten ist. Den Zusammenhang zwischen BASIC-Variablen und Registeradressen können wir uns leicht klarmachen, indem wir mit %20

# JU+TE Computerclub

beginnend den Variablen A bis Z je zwei Register zuordnen. Dem Weiterstellen der Adressen dient das Programm in Abb. 5. Durch Subtrahieren der aktuellen von der EPROM-Endadresse wird der Abschluß-Test ausgeführt. C=1 kennzeichnet Adressen innerhalb des zu behandelnden Bereichs. Das Unterprogramm in Abb. 6 bereitet das Einstellen der EPROM-Adresse mit Typ-spezifischen Belegungen der OR-Maske von r15 vor, damit die jeweils nötigen Steuersignalbelegungen entstehen. Nach Ausgabe der einzuspeichern Bitkombination über Port 2 eröffnet dieses Programm den Programmierimpuls. Das Initialisieren von Prozessor und Programmierzusatz erledigt die in Abb. 7 dargestellte Befehlsfolge. Sie wird von der

Zeile 10 des BASIC-Programms (Abb. 1) gerufen. Ab der Marke IN1 stellt der Bediener per Dialog das Kennbyte r14 ein. Dazu werden zwei Betriebssystem-Unterprogramme genutzt. Das Übertragen von EPROM-Inhalten in den RAM des Computers realisiert das Programm in Abb. 8 (von Zeile 80 gerufen) auf ganz einfache Weise. Bei Überschreiten der EPROM-Kapazität (Fehler bei Adresseingabe) wird das Schreiben in den RAM verhindert.

## 1 BASIC-Hauptprogramm

```
10 CALL %8DD;
   PRINT " SCHNELL-";
   PRINT " BRENNER";
   PRINT " TYP: U27";
   CALL %E1BE
20 PRINT "1=LESEN";
   PRINT "2=BRENNEN";
   PRINT "3=TESTEN";
   INPUT "AUSWAHL:" I
40 IF I>3
   THEN GOTO 10
50 INPUT "RAM-ANF:" M;
   INPUT "RAM-END:" N;
   INPUT "ROM-ANF:" O;
   LET P=N-M+O
60 IF I=2
   THEN GOTO 90
70 IF I=3
   THEN GOTO 100
80 CALL %E1F2;
   GOTO 100
90 PRINT "VPP PRUEFEN !"
   LET I=GTC;
   CALL %E202
100 CALL %E247;
   IF L=0
   THEN PRINT "ROM LEER"
110 ELSE ;
   PRINT K, " FEHLER"
120 PTH "CRC:" G;
   LET I=GTC;
   GOTO 10
```

## 2 UP Adresse laden

```
*ADR: E6 03 50 Vorauswahl
E132 48 E0 ROM-Adresse H
E134 56 E4 3F 00XX XXXX
E137 76 EE 18 2716 oder 2732?
E13A EB :AD4 wenn ja
E13C 76 E4 20 A13 besetzt ?
E13F 6B :AD1 wenn nicht
*AD4: 46 E4 40 A13/Vcc := 1
E144 A6 EE 08 U 2716 ?
E147 EB :AD1 wenn nicht
E149 46 E4 48 A11/Vpp := 1
*AD1: 56 E4 5F 0X0X XXXX
E14F 42 4F Steuerbits
E151 49 02 Ausgabe Port 2
E153 E6 03 10 Übernahmeakt
E156 E6 03 F0 Auswahl passiv
E159 19 02 ROM-Adresse 1
E15B E6 03 30 Übernahmeakt
E15E E6 03 70 Auswahl passiv
E161 56 E4 1F 000X XXXX
E164 42 41 A13 verändert ?
E166 EB :AD2 wenn nicht
E168 8C 23 Wartezeit
*AD3: 80 E8 Einschwingen
E16C EB :AD3 abwarten
*AD2: AF Rücksprung
```

## 3 UP Byte lesen

```
*BLN: D6 :ADR Adresse laden
E172 E6 F6 FF Port2: Eingänge
E175 E6 03 60 Vorauswahl
E178 E6 03 20 OE aktiv
E17B FF Einschwingen
E17C FF abwarten
E17D 48 02 Byte holen
E17F E6 03 60 Auswahl passiv
E182 B0 F6 Port2: Ausgänge
E184 AF Rücksprung
```

## 4 UP Anfangsadressen laden

```
*IAD: 8F Interruptsperr
E186 31 20 Registerpointer
E188 08 3C ROM-ANF H
E18A 18 3D ROM-ANF L
E18C 28 38 RAM-ANF H
E18E 38 39 RAM-ANF L
E190 A4 EB 3E Bereichstest
E193 AF Rücksprung
```

## 5 UP Adreßerhöhung und Test

```
*AET: A0 E0 ROM-Adresse +1
E196 A0 E2 RAM-Adresse +1
E198 48 3E ROM-END H
E19A 58 3F ROM-END L
E19C 22 51 - ROM-Adresse L
E19E 32 40 - ROM-Adresse H
E1A0 AF Rücksprung
```

# JU+TE Computerclub

Das Maschinenprogramm zum Programmieren des EPROM (Abb. 9, Aufruf in Zeile 90) prüft zunächst mit dem tcm-Befehl, ob alle Bits, die mit 1 belegt werden sollen, bereits auf 1 stehen. Beim Programmieren können nur 0-Belegungen erzeugt werden. Im Fehlerfall kommt ein Fragezeichen zur

Anzeige, der EPROM ist zu löschen. An der Marke PR7 beginnt der eigentliche Programmieralgorithmus, die Einzelimpulsdauer von 774 µs bestimmt die 256mal durchlaufene djnz-Schleife (PR3). Sobald das gewünschte Byte im EPROM steht, wird mit PR4 fortgesetzt. Das Arbeitsregister

r6 zählt die nötigen Einzelimpulse, deren Anzahl verdoppelt (r7) die Länge des abschließenden Programmierimpulses bestimmt. Beim Sprung auf PR5 wird mit Hilfe von r6 die Gesamtdauer auf etwa 48 ms begrenzt, um Beschädigungen des EPROM, z. B. bei einem Kontaktfehler, zu vermeiden.

## 6 UP Prog.-Impuls starten

```
*PIS: A6 EE 08 U 2716 ?
E1A4 EB :PI1 wenn nicht
E1A6 FC E0 OR-Maske
*PI1: A6 EE 10 U 2732 ?
E1AB EB :PI2 wenn nicht
E1AD FC 60 OR-Maske
*PI2: D6 :ADR Adresse laden
E1B2 FC 00 OR-Maske
E1B4 E6 03 40 Vorauswahl
E1B7 82 52 Byte aus RAM
E1B9 59 02 zum EPROM
E1BB B0 03 Impulsanfang
E1BD AF Rücksprung
```

## 7 UP Initialisieren

```
*INI: 8F Interruptsperre
E1BF E6 F7 01 P3M
E1C2 E6 03 70 Auswahl passiv
E1C5 B0 F6 Port2: Ausgänge
E1C7 31 20 Registerpointer
E1C9 0C 00 ROM-Adresse H
E1CB 1C 00 ROM-Adresse L
E1CD FC 00 OR-Maske
E1CF D6 :ADR Adresse laden
E1D2 EC 08 Kennbyte
E1D4 9F Int.-Freigabe
*IN1: 90 EE Kennbyte *2
E1D7 7B :INI bei Überlauf
E1D9 E6 5B 29 Kursor
E1DC 31 10 Registerpointer
E1DE 2C 00 Anzeigewert H
E1E0 38 2E Anzeigewert L
E1E2 D6 0A A3 Dezimalanzeige
E1E5 D6 08 24 Tastenabfrage
E1E8 31 20 Registerpointer
E1EA A6 5A 0D ENTER ?
E1ED EB :IN1 wenn nicht
E1EF E0 EE Kennbyte
E1F1 AF Rücksprung
```

## 8 UP EPROM lesen

```
*LES: D6 :IAD Anfangsadressen
E1F5 EB :FE1 bei Adreßfehler
*LE1: D6 :BLN Byte vom EPROM
E1FA 92 42 in den RAM
E1FC D6 :AET nächste Adresse
E1FF FB :LE1 wenn nicht Ende
*FE1: AF Rücksprung
```

## 9 UP EPROM Programmieren

```
*PRO: 06 :IAD Anfangsadressen
E205 FB :FE1 bei Adreßfehler
*PR2: D6 :BLN Byte vom EPROM
E20A 82 52 Byte vom RAM
E20C 62 45 programmierbar?
E20E 6B :PR1 wenn ja
E210 E6 5A 3F sonst löschen
E213 8D 08 27 Anzeige "?"
*PR1: D6 :AET nächste Adresse
E219 FB :PR2 wenn nicht Ende
E21B D6 :IAD Anfangsadressen
*PR7: 6C 14 Impulszähler
*PR5: D6 :PIS Impulsanfang
*PR3: 8A :PR3 768 µs warten
E225 E6 03 40 Impulsende
E228 D6 :BLN Byte vom EPROM
E22B A2 45 fertig ?
E22D 6B :PR4 wenn ja
E22F 6A :PR5 ächter Impuls
*PR4: 7C 15 Nachbrennzähler
E233 22 76 Impulszahl
E235 02 77 verdoppelt
E237 D6 :PIS Impulsanfang
*PR6: 8A :PR6 doppelte Zeit
E23C 7A :PR6 abwarten
E23E E6 03 40 Impulsende
E241 D6 :AET nächste Adresse
E244 FB :PR7 wenn nicht Ende
E246 AF Rücksprung
```

## 10 UP EPROM testen

```
*TST: D6 :IAD Anfangsadressen
E24A CC FF CRC H
E24C DC FF CRC L
E24E B0 34 Fehlerzahl H
E250 B0 35 Fehlerzahl L
E252 B0 36 Bytezahl H
E254 B0 37 Bytezahl L
E256 FB :FE2 bei Adreßfehler
*TS3: D6 :BLN Byte vom EPROM
E25B 82 52 Byte vom RAM
E25D A2 45 gleich ?
E25F 6B :TS1 wenn ja
E261 A0 34 Fehlerzahl +1
*TS1: A6 E4 FF leer ?
E266 B6 :TS2 wenn ja
E268 A0 36 Bytezahl +1
*TS2: B2 4C CRC-Algorithmus
E26C C8 E4
```

Der Rücksprung erfolgt nach Abarbeiten des gesamten vereinbarten Adreßbereichs. Die BASIC-Anweisung 100 ruft das Test-Programm (Abb. 10) zum Ermitteln der Anzahl programmierter (Variable L) und fehlerhafter Bytes (K) sowie zum Berechnen der Prüfsumme CRC (G). Bei Adreßfehler wird die normalerweise unmögliche Fehleranzahl -1 eingestellt, um diesen Irrtum mit wenig Aufwand zu kennzeichnen. Der CRC-Algorithmus realisiert mit einer Reihe von XOR-Verknüpfungen eine vom EPROM-Inhalt abhängige Zahl, die bei Verändern eines beliebigen Bits ganz anders ausfällt. Das gestattet spätere Kontrollen des EPROM-Inhalts. An dieser Befehlsfolge gibt es nichts zu verstehen. Schließlich wird die zum Schutz der Signalübertragung zum EPROM-Programmierzusatz unterbundene Annahme von Interrupts wieder freigegeben,

E26E	F0	E4	C
E270	56	E4	0F R
E273	B2	4C	C
E275	C8	E4	-
E277	F0	E4	A
E279	90	E4	l
E27B	58	E4	g
E27D	56	E4	1F o
E280	B2	40	r
E282	D8	E4	i
E284	48	E5	t
E286	E0	E4	h
E288	56	E4	F0 m
E28B	B2	4D	u
E28D	56	E5	E0 s
E290	B2	5C	
E292	C8	E4	CRC H
E294	D8	E5	CRC L
E296	D6	:AET	nächste Adresse
E299	FB	:TS3	wenn nicht Ende
E29B	E6	03	70 Auswahl passiv
E29E	9F		Int.-Freigabe
E29F	AF		Rücksprung
*FE2:	80	34	Fehlerzahl: -1
E2A2	80	36	Bytezahl: -1
E2A4	E6	03	70 Auswahl passiv
E2A7	9F		Int.-Freigabe
E2A8	AF		Rücksprung

die Bildausgabe fortgesetzt. Da das Steuern dieser Kommunikation auch P36 und P37 beeinflusst, begleiten jede Aktion entsprechende Töne und Strichmuster. Die beschriebenen Programme passen in das erste Kbyte ab %E000. Für die Markentabelle ist ein weiteres Kbyte erforderlich. Die Abb. 11 enthält die Maschinenprogramme als Hex Liste zur Eingabe im DATA-Modus des JU+TE-Computers, bei der keine Markentabelle benötigt wird. Die Bedienung dieses schnellen EPROM-Handlers ist Menü-gesteuert und damit sehr einfach. Zunächst muß der angezeigte Typ mit ENTER bestätigt oder mit CLR abgelehnt werden. Im zweiten Fall bietet das initialisierende Maschinenprogramm sofort den nächsten Typ zur Auswahl. Nach Festlegen der Betriebsart sind die Adressen des Speichers im Computer (RAM) und die Anfangsadresse des EPROM Bereichs (ROM-ANF) am besten hexadecimal einzugeben. Lesen oder Testen beginnen sofort. Vor dem Programmieren (Brennen) wird nach der Programmierspannung gefragt, bevor durch beliebige Tastenbetätigung der Algorithmus beginnt. Es empfiehlt sich aber, grundsätzlich vor dem Stecken des EPROM auf die richtige Programmierspannung zu achten und Falsch-

11 Hex-Listing															
E128:	43	3B	47	31	30	0D	00	E6							
E130:	03	50	48	E0	56	E4	3F	76							
E138:	EE	18	EB	05	76	E4	20	6B							
E140:	0B	46	E4	40	A6	EE	08	EB							
E148:	03	46	E4	48	56	E4	5F	42							
E150:	4F	49	02	E6	03	10	E6	03							
E158:	F0	19	02	E6	03	30	E6	03							
E160:	70	56	E4	1F	42	41	EB	06							
E168:	8C	23	80	E8	EB	FC	AF	D6							
E170:	E1	2F	E6	F6	FF	E6	03	60							
E178:	E6	03	20	FF	FF	48	02	E6							
E180:	03	60	B0	F6	AF	8F	31	20							
E188:	08	3C	18	3D	28	38	38	39							
E190:	A4	EE	3E	AF	A0	E0	A0	E2							
E198:	48	3E	58	3F	22	51	32	40							
E1A0:	AF	A6	EE	08	EB	02	FC	E0							
E1A8:	A6	EE	10	EB	02	FC	60	D6							
E1B0:	E1	2F	FC	00	E6	03	40	82							
E1B8:	52	59	02	B0	03	AF	BF	E6							
E1C0:	F7	01	E6	03	70	FC	66	31							
E1C8:	20	0C	00	1C	00	F0	0C	D6							
E1D0:	E1	2F	EC	08	9F	90	EE	7B							
E1D8:	E5	26	5B	29	31	10	2C	00							
E1E0:	38	2E	D6	0A	A3	D6	08	24							
E1E8:	31	20	A6	5A	0D	EB	E6	E0							
E1F0:	EE	AF	06	E1	85	FB	0A	D6							
E1F8:	E1	6F	92	42	D6	E1	94	FB							
E200:	F6	AF	D6	E1	85	FB	FA	D6							
E208:	E1	6F	82	52	62	45	6B	06							
E210:	E6	5A	3F	8D	08	27	D6	E1							
E218:	94	FB	EC	D6	E1	85	6C	14							
E220:	D6	E1	A1	8A	FE	E6	03	40							
E228:	D6	E1	6F	A2	45	6B	02	6A							
E230:	EF	7C	15	22	76	02	77	D6							
E238:	E1	A1	8A	FE	7A	FC	E6	03							
E240:	40	D6	E1	94	FB	D8	AF	D6							
E248:	E1	85	CC	FF	DC	FF	B0	34							
E250:	B0	35	B0	36	B0	37	FB	48							
E258:	D6	E1	6F	82	52	A2	45	6B							
E260:	02	A0	34	A6	E4	FF	6B	02							
E268:	A0	36	B2	4C	C8	E4	F0	E4							
E270:	56	E4	0F	B2	4C	C8	E4	F0							
E278:	E4	90	E4	58	E4	56	E4	1F							
E280:	B2	4D	D8	E4	48	E5	E0	E4							
E288:	56	E4	F0	B2	4D	56	E5	E0							
E290:	B2	5C	C8	E4	D8	E5	D6	E1							
E298:	94	FB	BD	E6	03	70	9F	AF							
E2A0:	80	34	36	E6	03	70	9F	AF							
E2A8:	AF	00	00	00	00	00	00	00							

polungen tunlichst zu vermeiden. Die angezeigte Prüfsumme CRC gilt stets für den behandelten EPROM-Inhalt. Nach Auswahl 3 kann das Gesamt-CRC bei entsprechender Adreßvereinbarung berechnet werden. Für den U 2716 C eignen sich der RAM Bereich von %0000 bis %07FF und der

# JU+TE Computerclub

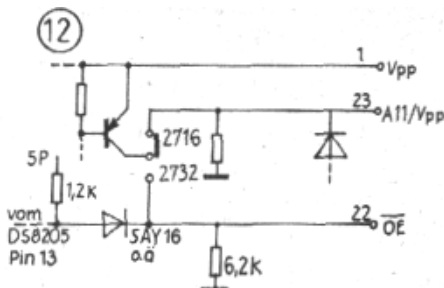
ROM-Anfang %0000. Bei der Anzeige des Testergebnisses ist auf Fragezeichen (EPROM löschen) und negative Fehlerzahl (falsche Adreßangabe) zu achten. Nach Lesen und Brennen muß die Fehlerzahl 0 erscheinen, nach Testen nur, wenn auf Übereinstimmung geprüft wird. Die Ausschrift „ROM LEER“ gilt wie die Prüfsumme nur für den behandelten Bereich.

Abb. 12 stellt eine Möglichkeit vor, den in Heft 10/88 vorgestellten Programmierzusatz mit einem Wechselschalter, einer Diode und zwei Widerständen zu erweitern, um das Programmieren von U 2732 C auch gerätetechnisch zu gestatten. Bereits vorhandene Bauelemente sind hier nicht bezeichnet. Beim Handhaben der Typen U 2764 C und U 27128 C spielt die Stellung des Wechselschalters keine Rolle. Das Lesen und Testen von U 2732 C ist auch ohne diese Hardware-Ergänzung möglich. Der Elko am Anschluß A11/Vpp ist jedoch für alle Anwendungen des schnellen EPROM-Handlers zu entfernen. Viel Erfolg beim EPROM-Brennen!

## Höher integrierte EPROM?

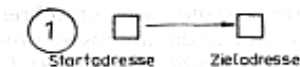
Viele JU+TE-Computer-Freunde mochten einen höher integrierten EPROM als den U 2716 C einsetzen, um das 4 K-Betriebssystem in einem Schaltkreis unterbringen zu können. Dadurch reicht natürlich ein Speichermodul auf Platz 1 aus. JU+TE vermittelt das Programmieren des 4K-Betriebssystems in die EPROM Typen U2732C, U2764C und U 27128 C. Dabei wird aber ein geschlossener Speicherbereich genutzt (%0800 bis 17FF). Deshalb verändern sich alle Startadressen der im Heft 5/1989 veröffentlichten allgemein nutzbaren Unterprogramme, die eine Startadresse über %2000 besitzen, auf den entsprechenden Wert ab 1000. Weitere Unterschiede bestehen nicht.

Dr. Helmut Hoyer



## Turm von Hanoi

Der Speicherplatzbedarf für dieses Programm beträgt 2 KByte. Die Spielidee besteht darin, den aus fünf Bausteinen unterschiedlicher Größe bestehenden Turm mit möglichst wenigen Versuchen vom linken auf den rechten Standort umzustapeln. Der mittlere Standort dient zum Zwischenlagern der Bausteine. Dabei ist es verboten, einen größeren auf einen kleineren Baustein zu setzen. Nach dem Starten baut sich der Turm selbständig auf dem linken Standort auf. Zu jedem nun folgenden Stapelschritt gehören zwei Eingaben, die Start und die Zieladresse, die

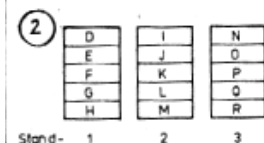


mit Hilfe der Zifferntasten 1 bis 3 nacheinander eingegeben werden. Der Cursor oben rechts zeigt an, welche der beiden Positionsangaben als nächste einzugeben ist (Abb. 1). Nach Eingeben der Zieladresse werden die Stapelaktion ausgeführt und der Cursor entsprechend rückgesetzt. Bis zur Eingabe neuer Start und Zieladressen bleiben die alten Positionsangaben in der Anzeige. Schummeln wird programmtechnisch verhindert. Bei Spielende ertönt eine Melodie. Danach erfolgt die Auswertung. Durch beliebigen Tastendruck kann ein neues Spiel gestartet werden.

### Programmbeschreibung

Zu Beginn erfolgt das Initialisieren der Parameter. In A wird

der kleinste Bausteinwert des linken Turmes gespeichert; in B der des mittleren und in C der des rechten Turmes. Die Parameter D bis R stellen die 15 möglichen Bausteinpositionen dar (Abb. 2). Sie speichern, welcher Baustein sich auf



welcher Position befindet. S lädt die zu löschende und T die zu setzende Bildpunktadresse, U speichert, welcher Baustein gerade versetzt werden soll, und in Abhängigkeit davon lädt V die entsprechende Bildinformation. W, X, Y sind Hilfsparameter, und Z ist der Versuchszähler. D=1 bis H=5 (Zeile 20) bereiten zusammen mit

T=%FF58 (Bildpunktadresse für den 1. Baustein) den Aufbau des kompletten Turmes auf dem linken Standort vor. Da U die Werte 1 bis 5 durchläuft, werden an dieser Stelle alle fünf Bausteine dargestellt. T=T+16 aktualisiert vor jeder Bausteindarstellung die Bildspeicheradresse und realisiert somit auch die Leerzeile zwischen zwei Bausteinen (siehe auch Abschnitt „Beschreibung der Graphik Prozeduren“). Die Zeilen 50 und 60 dienen der Erzeugung des Pfeils zwischen Start und Zieladresse in der rechten oberen Ecke. Auch sie werden später erläutert. Zeile 70 erwartet die Eingabe der Startadresse des auszuführenden Stapelschritts und positioniert sie vor dem Pfeil. Das Programm reagiert nur auf die Ziffern 1 bis 3 (1=linker,

2=mittlerer, 3=rechter Standort), wenn dort überhaupt ein Baustein liegt (A bzw. B bzw. C > 0). Nach Erkennen einer gültigen Startadresse wird in U gespeichert, welcher Stein versetzt werden soll, in den Zeilen 110 bis 270 die entsprechende Löschanzeige gesucht und in S geladen. Außerdem erfolgt die Aktualisierung der Parameter A, B, C bzw. D bis R.

Zeile 280 erwartet die Eingabe der Zieladresse. Auch hierbei sind nur die Ziffern 1 bis 3 er-

```

10  CALL %8DD;
    PRINT "TURM V. HANOI";
    PRINT "NR: "
20  LET A=1,B=0,C=0,D=1;
    LET E=2,F=3,G=4,H=5;
    LET I=0,J=0,K=0,L=0
30  LET M=0,N=0,O=0,P=0;
    LET Q=0,R=0,Z=1,U=1;
    LET T=%FF58
40  GOSUB 720;
    LET U=U+1,T=T+16;
    IF U=6
    THEN GOTO 40
50  PROC SETEB[%FE76,%FB1;
    PROC SETEB[%FE7E,%FD1;
    PROC SETEB[%FE86,0]
60  PROC SETEB[%FE8E,%FD1;
    PROC SETEB[%FE96,%FB1]
70  PROC SETR[%5B,%18];
    LET Y=GTC%A%7F;
    IF Y=%31
    THEN IF A>0
    THEN LET U=A;
    GOTO 110
    IF Y=%32
    THEN IF B>0
    THEN LET U=B;
    GOTO 170
    IF Y=%33
    THEN IF C>0
    THEN LET U=C;
    GOTO 230
100 GOTO 70
110 IF U=H
    THEN LET S=%FFD8,H=0,A=0
120 IF U=G
    THEN LET S=%FFB8,G=0,A=H
130 IF U=F
    THEN LET S=%FF98,F=0,A=G
140 IF U=E
    THEN LET S=%FF78,E=0,A=F
150 IF U=D
    THEN LET S=%FF58,D=0,A=E
160 GOTO 280
170 IF U=M
    THEN LET S=%FFDA,M=0,B=0

```



# JU+TE Computerclub

```

180 IF U=L
    THEN LET S=%FFBA,L=0,B=M
190 IF U=K
    THEN LET S=%FF9A,K=0,B=L
200 IF U=J
    THEN LET S=%FF7A,J=0,B=K
210 IF U=I
    THEN LET S=%FF5A,I=0,B=J
220 GOTO 280
230 IF U=R
    THEN LET S=%FFDC,R=0,C=0
240 IF U=Q
    THEN LET S=%FFBC,Q=0,C=R
250 IF U=P
    THEN LET S=%FF9C,P=0,C=Q
260 IF U=O
    THEN LET S=%FF7C,O=0,C=P
270 IF U=N
    THEN LET S=%FF5C,N=0,C=0
280 PROC SETR[%5B,%1C];
    LET Y=GTC$A%7F;
    IF Y=%31
    THEN GOTO 320
290 IF Y=%32
    THEN GOTO 390
300 IF Y=%33
    THEN GOTO 460
310 GOTO 280
320 IF H=0
    THEN LET H=U,T=%FFD8;
    GOTO 380
330 IF G=0
    THEN IF H>U
    THEN LET G=U,T=%FFB8;
    GOTO 380
340 IF F=0
    THEN IF G>U
    THEN LET F=U,T=%FF98;
    GOTO 380
350 IF E=0
    THEN IF F>U
    THEN LET E=U,T=%FF7B;
    GOTO 380
360 IF D=0
    THEN IF >U
    THEN LET D=U,T=%FF58;
    GOTO 380
370 GOTO 280
380 LET A=U;
    GOTO 530
390 IF M=0
    THEN LET M=U,T=%FFDA;
    GOTO 450
400 IF L=0
    THEN IF M>U
    THEN LET L=U,T=%FFBA;
    GOTO 450
410 IF K=0
    THEN IF L>U
    THEN LET K=U,T=%FF9A;
    GOTO 450
420 IF J=0
    THEN IF K>U
    THEN LET J=U,T=%FF7A;
    GOTO 450
430 IF I=0
    THEN IF J>U
    THEN LET I=U,T=%FF5A;
    GOTO 450
440 GOTO 280

450 LET B=U;
    GOTO 530
460 IF R=0
    THEN LET R=U,T=%FFDC;
    GOTO 520
470 IF Q=0
    THEN IF R>U
    THEN LET Q=U,T=%FFBC;
    GOTO 520
480 IF P=0
    THEN IF Q>U
    THEN LET P=U,T=%FF9C;
    GOTO 520
490 IF O=0
    THEN IF P>U
    THEN LET O=U,T=%FF7C;
    GOTO 520
500 IF N=0
    THEN IF O>U
    THEN LET N=U,T=%FF5C;
    GOTO 520
510 GOTO 280
520 LET C=U
530 LET Z=Z+1;
    GOSUB 700
540 IF Z>99
    THEN CALL %8DD;
    END
550 IF N<>1
    THEN GOTO 70
560 PROC SETR[%5B,%20];
    IF Z=32
    THEN PRINT " ";
    PRINT " R E K O R D";
    LET W=8,Y=0;
    GOTO 620
570 IF Z<37
    THEN PRINT " * G U T *";
    PRINT "ABER ES GEHT";
    PRINT "NOCH BESSER !";
    GOTO 610
580 IF Z<43
    THEN PRINT " ";
    PRINT " UEBEN !!";
    GOTO 610
590 IF Z<50
    THEN PRINT " WO BLEIBT";
    PRINT " DIE LOGIK ?";
    GOTO 610
600 PRINT " GEH' LIEBER";
    PRINT " MURMELN !";
    LET W=3,Y=10
    GOSUB 900;
    LET Y=GTC;
    GOTO 10
700 LET V=%FFFF,W=1
710 PROC SETEW[S,V];
    LET W=W+1;
    IF W<4
    THEN LET S=S+8;
    GOTO 710
720 PROC SETR[%5B,%13];
    IF Z/10=0
    THEN PROC PTC[#1B];
    GOTO 740
730 PROC PTC[Z/10+48]
740 PROC PTC[Z%M10+48]
750 IF U=1
    THEN LET V=%FC7F
760 IF U=2
    THEN LET V=%F83F
770 IF U=3
    THEN LET V=%F01F
780 IF U=4
    THEN LET V=%E00F
790 IF U=5
    THEN LET V=%0007
800 LET W=1
810 PROC SETEW[T,V];
    LET W=W+1;
    IF W<4
    THEN LET T=T+8;
    GOTO 810
820 PROC SETEW[%FFF8,%280];
    PROC SETEW[%FFFA,%540];
    PROC SETEW[%FFFC,%AA0]
830 RETURN
900 PROC SETRR[%F2,%23];
    PROC SETR[%F1,10]
910 LET X=240;
    PROC SETR[%F1,%8A]
920 PROC SETR[%F2,X];
    WAIT Y;
    LET X=X-20;
    IF X>0
    THEN GOTO 920
930 LET W=W-1;
    IF W>0
    THEN GOTO 910
940 PROC SETR[%F1,10];
    RETURN

```

# JU+TE Computerclub

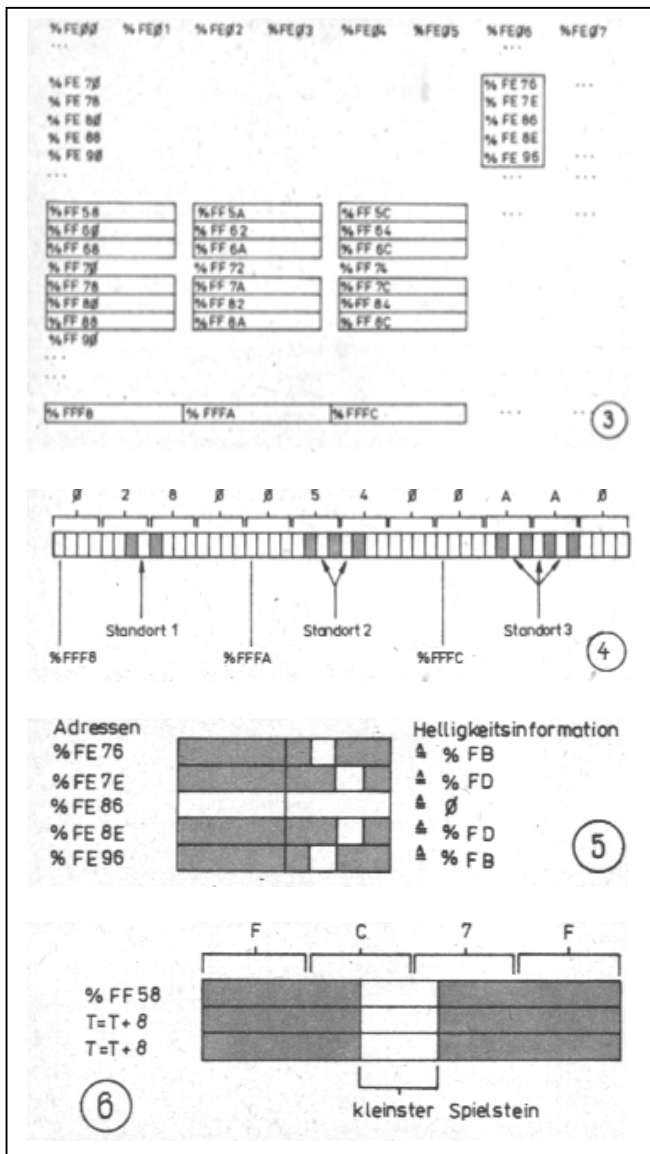
laubt. Bei richtiger Eingabe sucht das Programm die erste freie Position des gewünschten

Standortes, lädt in T die entsprechende Setzadresse, aktualisiert den Parameter A, B

bzw. C und springt in jedem Fall in Zeile 530, wo der Versuchszähler erhöht wird. Auf den Versuch, einen größeren auf einen kleineren Baustein zu stapeln, reagiert das Programm nicht, sondern erwartet eine neue Zieleingabe (Zeilen 370, 440 bzw. 510). Bei ungeschickter Spielweise kann es vorkommen, daß keine erlaubte Zieladresse möglich ist. In diesem Fall bleibt nur die Möglichkeit, als Zieladresse die Startadresse einzugeben; d. h., es erfolgt kein Umstapeln, aber ein Erhöhen des Versuchszählers. Ab Zeile 700 wird der Bildschirm aktualisiert: Zeile 700 löscht den gewünschten Baustein; Zeile 720 bis 740 gibt den Versuchszähler aus, und Zeile 750 bis 810 setzt den Baustein auf die neue Position. Zeile 820 erzeugt am unteren Bildrand eine Linie, die die drei möglichen Standorte des Turmes anzeigt. Ein sicheres Zeichen für das Spielende ist die Abfrage des Positionsparements N (Zeile 550). Bei Spielende (N=1) folgt ab Zeile 560 die Auswertung des erreichten Ergebnisses, verbunden mit der Ausgabe einer Melodie (Zeilen 900 bis 940). Durch beliebigen Tastendruck kann in Zeile 620 ein neues Spiel gestartet werden.

## Graphik-Prozeduren

Wir beschreiben nun, wie die Linie am unteren Bildrand erzeugt wird, der Pfeil zwischen Start- und Zieladresse, sowie den Aufbau der Spielsteine. Mit der Prozedur SETEW [A,B] bzw. SETEB [A,B] kann mit einer BASIC-Anweisung direkt ein Bild bzw.



# JU+TE Computerclub

ein Bildpunkt erzeugt werden, wobei A die Bildspeicheradresse und B die Helligkeitsinformation auf dieser Adresse darstellt. Voraussetzung ist die Kenntnis der Zuordnung der Speicheradressen zur Bildgeometrie (JU+TE 3-1988), die Abb. 3 darstellt.

Am einfachsten ist die Erzeugung der Linie am unteren Bildrand erklärt (Zeile 820). Die Bildspeicheradresse wird nacheinander auf %FFF8, %FFFA und %FFFC gestellt (Abb. 3). Die jeweilige Helligkeitsinformation ergibt sich wortweise nach Abb. 4, wobei zu beachten ist, daß eine 0-Belegung einen hellen Bildpunkt erzeugt. Nach demselben Prinzip wird der Pfeil erzeugt (Zeile 50, 60). Die Lage der Bildspeicheradressen ist in Abb. 5 dargestellt, wobei allerdings die byteweise Prozedur ausreicht.

Es folgt die Beschreibung des graphischen Aufbaus eines Spielsteines (Zeile 750 bis 810). In T sei z. B. die Bildschirmadresse %FF58 (aus Zeile 360) geladen (Abb. 3), und U sei 1 (d. h. kleinster Baustein). Dann wird auf diese Bildschirmadresse die Helligkeitsinformation %FC7F geschrieben. Dieser Vorgang wiederholt sich dreimal, wobei  $T=T+8$  praktisch eine Zeilenschaltung realisiert (Abb. 6).

Bernhard Piniek

## U2716C auf Modulplatz 2

Die Pin-Belegung der EPROM-Typen U 2716 C bis U 27128 C ist weitgehend gleich, so daß sich niedriger integrierte Schalt-

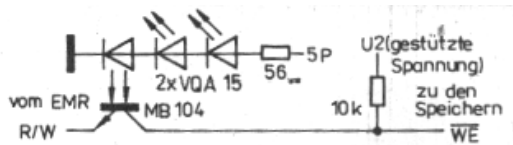
kreise auf Steckplätzen für höher integrierte

einsetzen lassen. Einige Besonderheiten sind den noch zu beachten. Der Programmierspannungseingang des U 2716 C ist bei den anderen Typen mit dem Adreßbit A11 belegt. Sicheres Lesen erfordert allgemein 1 Pegel am Programmierspannungseingang (Ausnahme: U 2732 C). Beim 4K-Betriebssystem des JU+TE-Computers belegt der erste EPROM den Adreßbereich von %0800 bis %0FFF. Da hier A11 mit 1 belegt ist, gibt es keine Probleme beim Einsatz des U 2716 C, auch wenn die Modulleiterplatte für U 2732 C oder U 2764 C ausgelegt ist. Der zweite EPROM wird auf %2000 bis %27FF angesprochen. Da hier A11 = 0 gilt, sollte beim Bestücken von Leiterplatten für höher integrierte EPROM auf Platz 2 mit U 2716 C dessen Programmierspannungseingang (Pin 21) von A11 getrennt und mit der Versorgungsspannung 5P verbunden werden.

Dr. Helmut Hoyer

## Datenverlust trotz Batteriestütze

Verschiedene Leserbriefe, für die wir uns sehr herzlich bedanken, haben auf Datenverluste in gestützten CMOS-RAM-Schaltkreisen hingewie-



sen. Entsprechende Untersuchungen ermittelten den TTL-Dekoder DS 8205 D auf Modulplatinen für U 224 D und U 6516 D als Ursache. Dieser Schaltkreis erzeugt trotz passiver Eingänge auf allen Ausgangspins den aktiven Nullpegel, wenn die Betriebsspannung zusammen bricht. Bei gleichzeitig aktiver Schreibfreigabe (während des Ein- und Ausschaltens durchaus normal) kommt es zum ungewollten Verändern des RAM-Inhaltes. Das zu verhindern gibt es zwei einfache Möglichkeiten:

1. Das Ersetzen des DS 8205 D auf den betreffenden Speicherplatinen durch einen U 74 HCT 138 DK (pinkompatibler CMOS-Schaltkreis) verhindert das Aktivieren der /CE-Eingänge der RAM.
2. Das Unterbrechen der Schreibfreigabe (siehe Schaltbild) schützt den Speicherinhalt trotz aktiver Schaltkreisauswahl.

Dr. Helmut Hoyer

## In eigener Sache:

## Betriebssystem - Varianten

Der JU+ TE Computer wurde mit dem Anliegen entwickelt, mit extrem wenig Hardware auszukommen. Daraus folgt auch, mit der Speicherkapazität für das Betriebssystem zu geizen. Das erste im Jahr 1987 veröffentlichte Betriebssystem

## JU+TE Computerclub

*hat deshalb einen Umfang von nur 2 Kbyte und gestattet das Programmieren in TINY-MP-BASIC. Als Voraussetzung für das Nutzen der hohen Rechenleistung folgte Anfang dieses Jahres die Erweiterung auf 4 Kbyte, wo zusätzlich das Programmieren in Maschinensprache mit ausgebauten Test- und Korrekturhilfen und das bei Kleincomputern übliche Magnetkassetteninterface hinzukommen. Das einfache Grundkonzept veranlaßte einige Computerfreunde, ihre Ideen in veränderte oder ganz neue Betriebssysteme einfließen zu lassen. Es ist allerdings wegen der Konsequenzen für die Anwendersoftware nur dann möglich, sie zur allgemeinen Nutzung anzubieten, wenn sie in hohem Maße aufwärtskompatibel sind (die bisherigen Eigenschaften nicht einschränken) und mit einem deutlichen Qualitätssprung Langlebigkeit erwarten lassen. Wir freuen uns trotzdem über das Interesse auch auf diesem schwierigen Gebiet und gratulieren allen, denen solch ein Projekt gelang. Für die zweite Hälfte des kommenden Jahres haben wir die Veröffentlichung einer neuen Bildschirmsteuerung mit 320 x 200 Bildpunkten geplant, die einen eigenen Prozessor enthält. Das entlastet den bisher alles erledigenden Einchip-Mikrorechner erheblich und versiebenfacht dessen effektive Rechenleistung. Das dafür nötige neue Betriebssystem soll wie bisher BASIC enthalten und das jetzige Maschinenprogramm-Entwicklungssystem zum Assembler erweitern. Als höhere Programmiersprache ist FORTH als transidente (von*

*Kassette zu ladende) Komponente geplant.*

*Um dabei möglichst vielen Interessenten gerecht zu werden, bitten wir alle Computerfreunde, ihre weitergehenden Wünsche, Anregungen oder gar Angebote an die Redaktion einzusenden. Was in diesem Jahr eingeht, läßt sich sicher noch berücksichtigen. Auf diese Weise wird die Attraktivität des JU+TE-Computers im Ergebnis einer echten Gemeinschaftsarbeit am besten gesteigert. Wir freuen uns schon auf Eure Ideen und bedanken uns im voraus sicherlich auch im Namen aller JU+TE-Computerfreunde.*

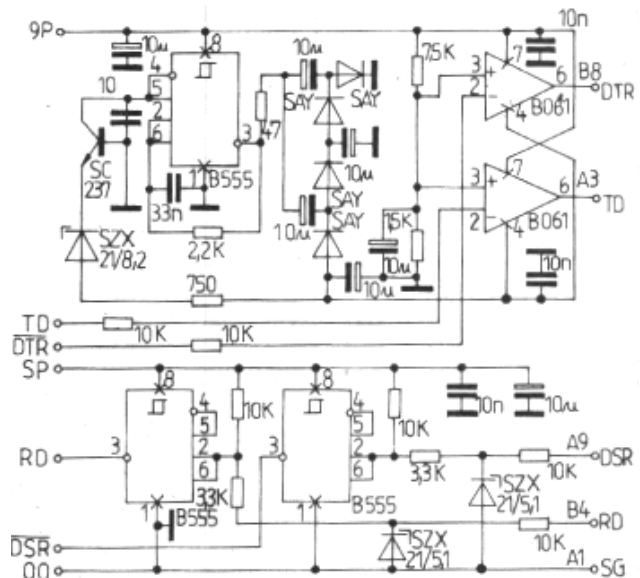
JU+TE

## Serielle Schnittstellen

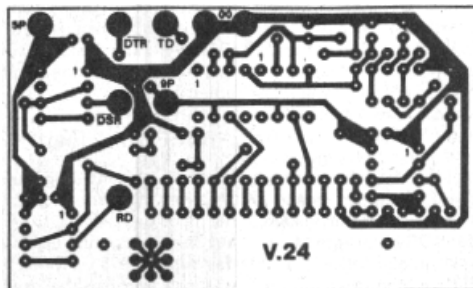
In diesem Beitrag wollen wir mit den verbreiteten seriellen Schnittstellen-Standards V.24 und IFSS bekanntmachen und einfache Ergänzungsbaugruppen für Computer, denen solche Anschlüsse fehlen, also auch für den JU+TE-Computer, vorstellen. Diese Baugruppen werden gebraucht, um Peripheriegeräte (z. B. Drucker) anschließen zu können und Rechner zu koppeln.

Im Unterschied zu parallelen übertragen serielle Schnittstellen alle acht Datenbits nacheinander über eine Leitung. Das spart gerätetechnischen Aufwand und gestattet das Überbrücken größerer Entfernungen. Von den unterschiedlichen Möglichkeiten, die serielle Übertragung zu organisieren, hat sich das asynchrone Datenformat zum üblichen Standard entwickelt.

Den Ruhezustand kennzeichnet der Sender mit 1-Pegel (Linienstrom). Das Übertragen eines Zeichens beginnt mit Nullsignal (Startbit), dem die sieben oder acht Datenbits folgen. Einem eventuellen Prüfbit (Paritätsbit) folgt wieder Linienstrom (mindestens 1 bis 2 Stoppbits). Für jedes Bit ist mit der Übertragungsrate eine bestimmte Zeitdauer festgelegt. Der Computerclub im Heft 5/89 enthält auf Seite 378 eine graphische Darstellung des asynchronen Datenformats. Das Empfangen solcher „Te-



Seriell Interface V.24: Schaltbild



Leiterplattenlayout (1:1)

legramme" setzt voraus, daß die eingestellte Übertragungsrate von der des Senders nicht mehr als 2 Prozent abweicht, was bei Quarztakt kein Problem darstellt. Gewöhnlich kann der Sender Daten schneller bereitstellen, als der Empfänger sie zu verarbeiten in der Lage ist. Um Datenverluste zu vermeiden, haben sich zwei unterschiedliche Mechanismen für

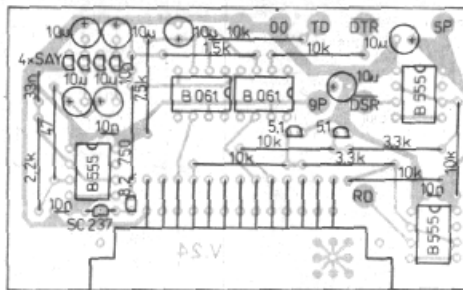
die Rückmeldung des Empfängers verbreitet. Der erste ist das sogenannte „Software-Protokoll“, auch als DC1/DC3 oder Xon/Xoff bezeichnet. Hier gibt der Empfänger mit dem Senden von Steuerzeichen (ebenfalls im asynchronen Format) Auskunft über seinen Betriebszustand. Der Code %11 (Xon) bedeutet, daß Daten empfangen werden können,

# JU+TE Computerclub

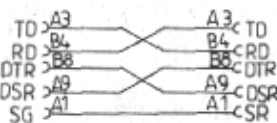
%13 (Xoff) kennzeichnet dagegen fehlende Empfangsbereitschaft. Das Zeichen % kennzeichnet die hexadezimale Schreibweise.) Das Software-Protokoll setzt serielle Datenleitungen in beiden Richtungen voraus, benötigt jedoch keine Steuersignale. Das „Hardware-Protokoll“ verwendet für die Rückmeldung ein statisches Signal (/DTR), das mit 0-Pegel die Bereitschaft des Empfängers kennzeichnet. Der Sender darf solange Daten anbieten, bis dieses Steuersignal 1-Pegel annimmt. Auch hier werden zwei Leitungen gebraucht: die Datenleitung vom Sender zum Empfänger und die Steuerleitung in umgekehrter Richtung.

**V.24**

Die Interface-Empfehlung V.24 (USA: RS 232 C) wurde für den Anschluß von Modulatoren/Demodulatoren (MODEM) zur Datenfernübertragung (z. B. per Telefonverbindung) an Computer entwickelt. Den 1-Pegel kennzeichnet eine Spannung zwischen -3 V und -15 V, 0-Pegel zwischen +3 V und +15 V. Neben den seriellen Datenein- und -ausgängen enthält V.24 einige Steuersignale. Zum Anschluß von Peripheriegeräten und zur Rechnerkopplung werden keine (bei Software-Protokoll) oder nur zwei davon benötigt. Üblich ist, DTR (data terminal ready, Daten-Endstelle bereit) zum Kennzeichnen der Empfangsbereitschaft und auf der anderen Seite /DSR oder /DCD zum Abfragen der Bereitschaft zu verwenden. Selten wird statt dessen das Signalkonzept RTS/CTS genutzt. Die Broschüre „Mikrorechner in der Amateurnetztechnik“ von And-



## Bestückungsplan



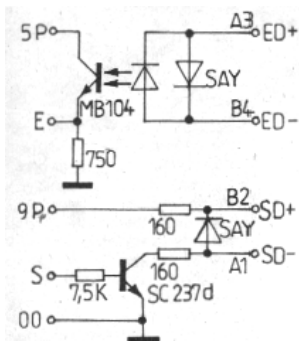
### Anschlußplan V.24-Kabel

reas Bogatz (Militärverlag der DDR 1988) beschreibt diesen Standard und die Bedeutung der einzelnen Steuersignale ausführlich. Zur Rechnerkopplung muß das Verbindungskabel Sende- (TD) und Empfangsdaten (RD) zwischen den beiden Steckern über Kreuz verbinden. Bei Anwendung des Hardware-Protokolls ist auch das betreffende Steuersignalkaabel (z. B. DTR/DSR im Kabel zu vertauschen. Auf diese Weise verbindet jede Ader einen Ausgang des V.24-Interface auf der einen Seite mit einem Eingang auf der anderen Seite. Unser Schaltungsvorschlag verwendet im Amateurhandel leicht erhältliche Schaltkreise. Die B 555 D werden als Bastelbeutel 11 angeboten. Statt der beiden B 061 D können auch B 081 D eingesetzt werden. Doppeloperationsverstärker eignen sich schlecht, da ein Kurzschluß beider Ausgänge,

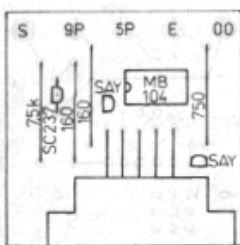
den das Interface ohne Schaden überstehen muß, eine Überschreitung der maximalen Verlustleistung zur Folge hätte. Die Spannungsversorgung der Operationsverstärker, die als

Ausgabetreiber dienen, erfolgt aus dem Ladekondensator des 5-V-Netzteils. Hier liegt eine Gleichspannung von etwa 9 V (höchstens 15 V!) an, aus der eine Schwingschaltung mit B 555 D auch die negative Betriebsspannung erzeugt. Eine überlagerte Wechselspannung an 9 P (Restwelligkeit) stört nicht. Die Operationsverstärker sind minimal beschaltet. Eingangsspannungen (TD und /DTR) unter 1,5 V werden als 0-, darüber als 1-Signal bewertet. Als Leitungsempfänger dienen zwei B 555 D. Die Widerstandsbeschaltung bestimmt die Triggerschwellen der V.24-Eingänge von -2,7 V und +1,5 V. Diese Werte verändern sich entsprechend beim Einsatz abweichender Widerstände. Als Kondensatoren können bedenkenlos auch Bauelemente mit doppelter oder halber Kapazität eingelötet werden. Für die Dioden eignen sich alle Typen von SAY 12 bis SAY 42. Zur Anpassung an Kabel für ein Hardware-Protokoll mit anderen Steuersignalen kann die Leiterzugführung am Steckverbinder leicht verändert werden, ohne Bauelemente anders anzuordnen. Die Signalbezeichnungen, die auf der V.24-Seite wie

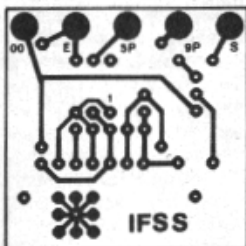
# JU+TE Computerclub



**Serielles Interface IFSS:**  
**Schaltbild:**



### Leiterplattenlayout (1:1)



## Bestückungsplan

allgemein üblich keine Negationsstriche haben, werden wie folgt verwendet:

TD: Sendedaten

RD: Empfangsdaten

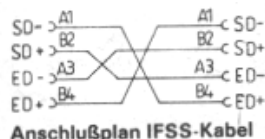
DTR: Bereitschaftsausgang

DSR: Bereitschaftseingang

SG: Masse

**IFSS**

Der Interfacestandard V.24 eignet sich nur für relativ kurze Verbindungen (bis 15 m) und hat den Nachteil, daß die gekoppelten Geräte auch galvanisch verbunden werden. Daraus resultieren bei hohem schaltungstechnischen Aufwand einige Schwächen, darunter auch eine gewisse Störanfälligkeit. Wesentlich stabiler sind Verbindungen mit potentialtrennenden Stromschleifen. Der wichtigste Standard mit diesem Prinzip ist IFSS (Interface seriell sternförmig). Die Schaltungstechnik für IFSS basiert auf dem Optokoppler MB 104. Seinen Eigenschaften entspricht die Zuordnung von 0 mA bis 3 mA für 0-Pegel und 15 mA bis 25 mA für 1-Pegel. Bei Stromquellen mit genügend hoher Leerlaufspannung (über 10 V) können Entfernungen bis 500 m überbrückt werden. Typisch sind 4-Draht-Verbindungen für je eine Stromschleife in beiden Übertragungsrichtungen. IFSS verwendet das Software-Protokoll. Üblich ist, mit Auswahlaltern festzulegen, ob die Stromquelle einer Schleife auf der Sender- oder Empfängerseite angeordnet wird. Unser Schaltungsvorschlag setzt grundsätzlich aktiven Sender und passiven Empfänger (senderseitige Stromquelle) voraus. Die Potentialtrennung erfolgt am Empfänger, so daß nur hier ein Optokoppler nötig ist. Der Widerstand des Verbindungskabels darf 150 Ohm nicht überschreiten, da als Stromquelle nur die Ladekondensatorspannung 9 P mit Vorwiderstand (2 x 160 Ohm) wirkt. Unter diesen Bedingungen minimalen Hardwareaufwandes lassen sich jedoch



immerhin 200 m überbrücken. Das Kabel muß jeweils SD+ der einen mit ED- der anderen Seite und gleichermaßen SD- mit ED+ verbinden. Achtung! Diese Signalbezeichnungen sind in der Literatur zum Teil widersprüchlich angegeben. Sie stehen für Sende- und Empfangsdaten mit der Polaritätsangabe (Betrachtung als Stromquelle). E und S bezeichnen die entsprechenden Signale in dem Mikrorechner angepaßten TTL Pegel. Mit der vorgestellten einfachen Schaltung lassen sich Übertragungsraten bis zu 4800 bit/s realisieren (vgl. Abb. 23 in JU+TE 11/88 S.874). Die Leiterplatte ist wie die des V.24-Interfaces so konstruiert, daß sie von den abgewinkelten Anschlußdrähten des am Gehäuse angeschraubten EFS-Steckers gehalten wird. Übrigens können wahlweise 10-polige (V.24: 26-polige) oder teilbestückte Steckerleistungen mit halbiertem Kontaktanzahl zum Einsatz kommen.

## Software

Am einfachsten ist das Verwalten der seriellen Interfaces mit seriellen Ein-/Ausgabe-Baugruppen (U 856 D oder SIO-Baugruppe innerhalb von Einchip-Mikrorechnern). Als erstes Beispiel dient ein Programm für den Anschluß eines Druckers mit Software-Protokoll an einen Einchip-Mikrorechner, wobei P37 als Signal S (IFSS)

# JU+TE Computerclub

oder TD (V.24) und P30 als E (IFSS) oder RD (V.24) zur Anwendung kommen. Beim Initialisieren werden die Übertragungsraten von 1200 bit/s eingestellt und mit dem Setzen der Interruptanmeldung IRQ4 die Geschwindigkeitsanpassung der Ausgaberroutine vorbereitet:

```
di
srp  %%F0
ld r10, %%10
ld r7,  %%41
ld r5,  %%37
ld r4,  %%04
ld r1,  %%03
```

Zum Übertragen eines Zeichens, das im Arbeitsregister r1 stehen muß, ist das Unterprogramm SEN zu rufen:

```
SEN: tm  %FA, %%10
jr  z, SEN
cp  %F0, %%11
jr  nz, SEN
ld  %F0, r1
and %FA, %%2F
ret
```

Zunächst wird die Bereitschaft des SIO-Senders abgewartet, die mit IRQ4=1 gekennzeichnet ist. Danach erfolgt der Test, ob als letztes Zeichen %11 (Xon) empfangen wurde. Nur, wenn beides gilt, folgen das Laden des Sendeschieberegisters mit dem Inhalt von r1 und das Rücksetzen der Interruptanmeldung IRQ4.

Als zweites Beispiel. nun für ein Hardware-Protokoll, dient der Anschluß eines Thermodruckers K 6304 mit programmtechnischer Erzeugung des asynchronen Datenformats. Über P35 (TD) werden die Daten mit 1200 bit/s ausgegeben, wenn die Rückmeldung DTR des Druckers über /DSR an P31 weitergeleitet aktiv ist. Das für jedes übertragene Zeichen gerufene Unterprogramm ist selbstinitialisierend und paßt zum Betriebssystem des JU+TE-Computers. Der

Algorithmus entspricht dem Anschluß der Schreibmaschine 3004 (siehe JUTE 5/88 S. 376 ff.), arbeitet jedoch ohne Interruptserviceroutine.

```
DRU: tm  3, %%02
jr  nz, DRU
djnz r9, DRU
di
ld  r8, %%0B
DR5: and  3, %%DF
DR6: ld  r9, %%B6
DR2: nop
     djnz r9, DR2
     dec  r8
     jr  z, DR3
     cp  r8, %%03
     jr  c, DR4
     rr  r1
     jr  nc, DR5
DR4: or  3, %%20
     jr  DR6
DR3: ei
     ret
```

Das Programm benutzt r1 als Datenregister, r8 als Bitzähler und r9 als Zeitzähler. Zunächst wird das Hardware-Protokoll durch Testen von P31 realisiert. Zur sicheren Störunterdrückung geschieht das gleich 256mal, was jedoch wegen der hohen Rechengeschwindigkeit keinen spürbaren Zeitverlust zur Folge hat. Nach Voreinstellen des Bitzählers beginnt die serielle Ausgabe mit einem Nullbit (Startbit) ab Marke DR5. Über DR6 (Initialisieren des Zeitzählers) und das Abwarten der Zeitdauer für ein Bit (DR2) kommt das Programm zum Verringern des Bitzählers r8. Zunächst werden die acht Datenbits aus r1 mit dem Schiebefehl rr über das C-Flag ausgewertet und entsprechend über DR4 mit 1-Pegel oder über DR5 mit 0 Pegel am Datenausgang P35 widerspiegelt. Bei r8 = 2 und r8 = 1 folgen zwei Stoppbits und zuletzt nach Interruptfreigabe der Rücksprung (DR3). In beiden Beispielen wird ein Quarztakt von 8 MHz voraus-

gesetzt. Um Störungen durch die Bilderzeugung auszuschließen, erfolgt im zweiten Beispiel eine zeitweilige Interruptsperre, die bei Verwendung der SIO-Baugruppe wie im ersten Beispiel nicht nötig ist. Beim JU+TE-Computer gibt es hier jedoch die Schwierigkeit, daß der serielle Ausgang P37 vom Betriebssystem für die Ausgabe der Synchronsignale genutzt wird. Deshalb empfiehlt es sich, serielle Schnittstellen am JU+TE-Computer wie im zweiten Beispiel zu realisieren. Das Koppeln zweier JU+TE-Computer erfordert übrigens kein Standard-Interface. Durch Verbinden von P36 des Senders mit dem Magnetbandeingang des Empfängers über einen Widerstand (ca. 20 kOhm) lassen sich die SAVE- und LOAD Routinen für diesen Zweck nutzen.

Dr. Helmut Hoyer

Zeichnungen: Krause Layout: Hoyer

## Computer im Palast

*Traditionell bieten wir Euch wieder während des Solitages der Jugendmedien der DDR unseren JU+TE-Computer-Service an. Wir treffen uns am 26. November 1989 im Palast der Republik in Berlin, Erdgeschoss, Platzseite, gegenüber der Post, von 10 bis 17 Uhr. Dazu laden wir Euch herzlich ein. Wir brennen EPROM, überspielen Software (darunter schon den FORTH-Compiler), verkaufen Leiterplatten und geben Tips. Der Autor des JU+TE-Computers beantwortet Eure Fragen.*



## Oszillograph mit dem JU+TE-Computer

Ein Oszillograph ist ein nützliches Hilfsmittel zur Fehlersuche in elektronischen Schaltungen, aber leider nur selten vorhanden.

```
E168: E6 20 08 E6
E16C: 15 4C 10 21
E170: FB 03 E6 15
E174: 48 D6 08 18
E178: 00 20 EB EF
E17C: AF FF FF FF
E180: 70 FD 31 F0
E184: 8F 6C FF 1C
E188: C0 7C 01 8C
E18C: B2 31 00 FF
E190: 29 20 29 21
E194: 29 22 29 23
E198: 29 24 29 25
E19C: 29 26 29 27
E1A0: 29 28 29 29
E1A4: 29 2A 29 2B
E1A8: 29 2C 29 2D
E1AC: 29 2E 29 2F
E1B0: 29 30 29 31
E1B4: 29 32 29 33
E1B8: 29 34 29 35
E1BC: 29 36 29 37
E1C0: 29 38 29 39
E1C4: 29 3A 29 3B
E1C8: 29 3C 29 3D
E1CC: 29 3E 29 3F
E1D0: 29 40 29 41
E1D4: 29 42 29 43
E1D8: 29 44 29 45
E1DC: 29 46 29 47
E1E0: 29 48 29 49
E1E4: 29 4A 29 4B
E1E8: 29 4C 29 4D
E1EC: 29 4E 29 4F
E1F0: 31 F0 8C 92
E1F4: 7C 08 1C 03
E1F8: 50 FD 9F AF
E1FC: FF FF FF FF
```

Der JU+TE-Computer kann durch eine einfache Schaltung auch als digitaler Speicheroszillograph benutzt werden. Mit 4 MHz Abtastzeit und einer Abtastzeit von 96 µs sind schon Messungen am JU+TE-Computer und an ähnlichen Geräten möglich. Da es sich bei Computern um Digitalschaltungen handelt, bei denen auf den

Signalleitungen nur zwei meßbare definierte Zustände auftreten, erkennt dieser Oszillograph auch nur H- oder L-Pegel.

Für ihn nutzen wir die Geschwindigkeit des Prozessors mit Hilfe eines Maschinenprogramms voll aus. Die gelesenen Werte werden in den Registern %20 bis %4F (BASIC-Variable A bis X) gespeichert, um eine maximale Abtastfrequenz von 500 kHz bei einer Datenbreite von 8 bit zu erreichen. 500 kHz sind für den Computerservice noch zu wenig. Deshalb verwenden wir nur ein Bit, das mit 4 MHz abgetastet wird. Den Takt liefert der entsprechend programmierte P36-Ausgang des EMR. Das Schieberegister DL299 sammelt jeweils 8 Abtastwerte, die dann als Byte über Port 2 übernommen werden. Dabei ist Bit7 das zuerst gelesene Bit und Bit0 das letzte eines Bytes. Die 1 kOhm Widerstände schützen den EMR, falls das Port 2 noch auf Ausgabe programmiert war.

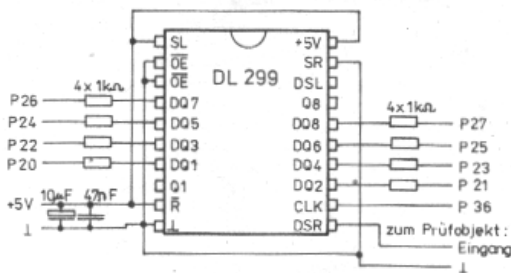
Das BASIC-Programm realisiert den Dialog mit dem Nutzer und die Bildschirmdarstellung der Daten, die das Maschinenprogramm (Startadresse %E180) eingelesen hat. Während des Laufs des Maschinenpro-

gramms (nur 120 µs) wird der Bildschirm abgeschaltet. Eine zweite Maschinenroutine bringt jeweils das Byte aus Variable A zur Anzeige (%E168).

Das Programm bietet zwei Funktionen an: Durch Eingabe von 0 wird eine Abtastserie gestartet. Die Werte werden danach aus den Registern in den Speicher von %E200 bis %E22F transportiert. Die Eingabe einer Ziffer von 1 bis 8 bewirkt das Anzeigen von jeweils 48 aufeinanderfolgenden Abtastwerten. Ein Buchstabe entspricht dabei einer Zeitdauer von 250 ns des jeweiligen Pegels.

Die ordnungsgemäße Funktion des Oszis kann man testen, indem der Eingang an Pin 4 des DS 8205 auf der Grundplatte des Computers gelegt wird. Nachdem mit 0 ein Abtastvorgang gestartet wurde, muß bei Anzeige des Blockes 1 der Abtastwerte folgende Pegelfolge auftreten: LLLHHLL. Danach stehen immer die Folgen HLLHHLL.

Auf einige Besonderheiten sei noch hingewiesen: Impulse am Eingang, die kürzer als 250 ns (entspricht 4 MHz) sind, werden entweder nicht erfaßt oder als 250 ns lang dargestellt. Da die Informationen erst komplett



# JU+TE Computerclub

eingelassen und dann auf dem Bildschirm angezeigt werden, kann man sie in Ruhe betrachten, allerdings entsprechen sie dem Signalverlauf der untersuchten Schaltung zum Abtastzeitpunkt, und nicht wie auf normalen Oszis dem Signalverlauf zum Betrachtungszeitpunkt.

Zuerst wird das BASIC-Programm eingegeben, danach mit DATA das Maschinenprogramm ab Adresse %E168. Anschließend wird mit SAVE E000 E1FF der BASIC-Teil zusammen mit dem Maschinensprachteil abgespeichert. Später kann das Programm ganz normal von BASIC aus geladen werden. Das Abspeichern muß aber immer wie oben beschrieben erfolgen, da andernfalls das Maschinenprogramm nicht mit gespeichert wird.

*Harun Scheutzw*

s. Nachtrag Seite 54

## Korrektur: Streifen beim JU+TE-Computer

Die Reaktionen zahlreicher Leser haben uns veranlaßt, die in JU+TE 7/1989 veröffentlichte Schaltung zur Streifenbeseitigung zu überprüfen. Tatsächlich halbiert sie nur die Anzahl der Streifen auf dem Bild und erfüllt ihren Zweck damit nicht. Wir bitten für diesen Fehler um Entschuldigung. Er entstand, weil die Schaltung mit einem anderen Betriebssystem getestet wurde. Eine Lösung für das 4K-System, die alle Streifen (auch beim Scrollen) beseitigt, aber keine reine Hardwarelö-

sung ist und deshalb nur unter BASIC funktioniert, stellen wir Euch heute vor. Für Interessierte eine nähere Erklärung der Funktionsweise:

Das Betriebssystem liest vor dem Schreiben eines Zeichens immer aus dem Bildwiederhol-speicher, da ein Zeichen nur 5 bit belegt und die anderen Bits eines Bytes ihren alten Wert behalten sollen. Oft werden die 5 Bit sogar auf 2 Byte verteilt. Also ist das R/W (Read/W rite) Signal des EMR nicht verwend-bar, um die Dekodierung so zu ergänzen, daß das Video-Schieberegister nur bei der Bildausgabe geladen wird. Die bereits eingebaute Schaltung (vgl. JU+TE 7/1989) ist jetzt nicht sinnlos, sondern es wird der Anschluß, der ehemals an Pin7(R/W) des EMR lag, mit Pin10(P35) des EMR verbunden. Das vorher vom Betriebssystem nicht benutzte P35 übernimmt die Aktivierung der Dekodierung zum Schieberegisterladen. Ist P35 auf L-Pegel, so sind alle Ausgaben aus dem Computerspeicher über das Schieberegister auf dem Bildschirm gesperrt. Bei H-Pegel kann das Schieberegister zur Bildausgabe geladen werden. Das Umschalten von P35 übernimmt eine Bildausgabe-Interruptroutine, die sich von der Routine im EPROM nur in 2 Byte unterscheidet. Der EPROM wird aber nicht umprogrammiert, weil damit der Einsatz des bereits vorgestellten S3004-Moduls gestört würde. Deshalb wird die Interruptroutine aus dem EPROM durch ein kleines BASIC-Programm in den RAM kopiert und dort entsprechend modifiziert. Dann wird der Interruptzeiger auf diese neue

Routine gestellt. So kann man jederzeit wählen, ob eine S3004 angeschlossen wird oder die Streifen verschwinden sollen. Im Beispiel wird ab Adresse %FB00 RAM vorausgesetzt, und zwar mindestens 144(=%90) freie Bytes. In Zeile 1 kann bei B=%FB00 auch eine andere RAM-Adresse eingetragen werden, ab der 144 Byte frei sind. Zeile 2 bis 4 kopieren die Routine aus dem EPROM, und Zeile 5 modifiziert diese. Zeile 6 sperrt die Interrupts, stellt den Zeiger auf die modifizierte Routine im RAM und gibt die Interrupts wieder frei. Das Sperren ist notwendig, da sonst ein Interrupt den erst halb geänderten Zeiger als Adresse benutzen könnte und abstürzen würde. Wenn dieses Programm schon einmal lief, dann reicht es aus, daß nach RESET (falls gestützte RAM verwendet werden, auch nach dem Einschalten), nur Zeile 6 abgearbeitet wird, die den durch Reset in den EPROM zurückgestellten Zeiger wieder auf die im RAM noch vorhandene Routine setzt. In dieser einzeln verwendeten Zeile 6 muß aber die Variable D durch den Hexadezimalwert ersetzt werden, der in Zeile 1 bei B= ... eingetragen war. Diese Lösung wurde mit dem veröffentlichten Betriebssystem erprobt.

*Harun Scheutzw*

```
1 LET A=%ADA,B=%FB00,
  C=%90,D=B
2 PROC SETEW[B,GETEW[A]]
3 LET A=A+2,B=B+2,C=C-2
4 IF C>0THEN GOTO 2
5 PROC SETEB[D+%15,%A0];
  PROC SETEB[D+%5B,%5F]
6 PROC SETR[%FB,%10];
  PROC SETRR[%7C,D];
  PROC SETR[%FB,%90]
7 END
```

## JU+TE-Computer-Software tip: Landstraße

### Spielidee

Landstraße ist ein Würfelspiel, das mit drei Würfeln gespielt wird. Ziel des Spieles ist es, die auf dem Bildschirm angegebene Zahlen zu streichen. Die Schwierigkeit besteht darin, daß die Zahlen unbedingt der Reihe nach zu streichen sind. Das kann auf zwei Arten geschehen:

1. Wertung eines der drei Würfelwerte oder
2. Wertung durch Addition mehrerer Würfelwerte. Beispiel: Es wird in der Landstraße die Zahl 5 gesucht. Möglicher Wurf: 3, 5, 1  
Wertung: 5  
Möglicher Wurf: 1, 4, 3  
Wertung: 1 + 4 = 5

### Spielablauf

Nachdem auf dem Bildschirm die Zahlen 1 bis 12 - die Landstraße - erschienen sind, werden drei Zufallszahlen erzeugt und als Würfelwerte angezeigt. Wenn man diese zur Wertung verwenden kann, so drückt man auf der Tastatur die entsprechende Zahl und schließt den Vorgang mit ENTER ab. Sofort wird diese Zahl auf der Landstraße gelöscht. Können noch andere Würfelwerte oder deren Addition zu einer weiteren Wertung herangezogen werden, so kann dies geschehen, ohne daß ein neuer Wurf ausgeführt wurde. Das Betätigen der Taste 0 löst einen neuen Wurf aus und schaltet den Wurfzähler weiter. Wenn alle Zahlen der Landstraße abgeräumt sind, erscheint das Spiel-

ergebnis. Da nach kann ein neues Spiel folgen.

### Programm

Nachdem in den Zeilen 10 und 20 die Variablen B (Bestwert), L (erste Ziffer in der Landstraße) und Z (Wurfzähler) geladen und der Zufallsgenerator initialisiert wurden, folgt die Anzeigeroutine (GOSUB 300). In den Zeilen 30 bis 50 werden die drei Würfelwerte P, Q, R erzeugt und mit Hilfe des Unterprogramms 150 auf ASCII korrigiert angezeigt. Anschließend erfolgt die Eingabe eines zu wertenden Würfelwertes S (Zeile 60).

Durch Eingabe einer Null wird ein neuer Wurf ausgeführt. Die Zeilen 70 bis 130 prüfen alle erlaubten Würfelkombinationen und vergleichen sie mit dem eingegebenen Wert S. Fällt die Prüfung positiv aus, so werden über Zeile 220 mit GOSUB 300 die Landstraße aktualisiert und der als nächstes zu würfelnde Wert ermittelt.

Ist die Landstraße abgeräumt, so folgt der Ergebnisausdruck. Das Unterprogramm 300 unterscheidet aus Gründen der Geschwindigkeitserhöhung zwischen ein- und zweistelligen Ziffern. Für die Ausgabe einstelliger Ziffern wird lediglich zur ASCII-gerechten Null der Laufindex M addiert. Es folgt die Ausgabe eines Bindestriches (Zeile 330). Bei den zweistelligen Ziffern wird für die Zehnerstelle mit PTC [%31] eine 1 erzeugt und anschließend die Einerstelle berechnet und ausgegeben (Zeile 340). Nach der Zahl 12 wird der Bindestrich durch Rücksprung ins Hauptprogramm unterdrückt.

Die Melodieerzeugung (Zeile 400 bis 440) geschieht in bekannter Weise (siehe JU+TE 4/89. S. 311). Das Programm benötigt knapp 1 Kbyte im BASIC RAM.

*Tobias Piniek*

```
10 LET B=100;
CALL %8DD
LET L=1,Z=0;
PROC SETRR[%F2,%FF0F];
PROC SETR[%F1,10]
30 GOSUB 300;
LET Z=Z+1;
LET W=GETR[%F2];
LET P=W$M6+1,W=W/6
40 LET Q=W$M6+1,W=W/6;
LET R=W$M6+1;
PROC SETR[%5B,%30];
PRINT "WURF: "Z;
PRINT " "
50 LET X=P,Y=%40;
GOSUB 150;
LET X=0,Y=%42;
GOSUB 150;
LET X=R,Y=%44;
GOSUB 150
60 PROC SETR[%5B,%50];
PRINT " ";
INPUT "WERTEN: "S;
IF S=0
THEN GOTO 30
70 IF S=P
THEN GOTO 200
80 IF S=Q
THEN GOTO 200
90 IF S=R
THEN GOTO 200
100 IF S=P+Q
THEN GOTO 200
110 IF S=P+R
THEN GOTO 200
120 IF S=Q+R
THEN GOTO 200
130 IF S=P+Q+R
THEN GOTO 200
140 GOTO 60
150 PROC SETR[%5B,Y];
PROC PTC[X+48];
RETURN
200 IF S=L
THEN LET L=L+1;
GOTO 220
210 GOTO 60
220 IF L<13
THEN GOSUB 300;
GOTO 60
230 CALL %8DD;
GOSUB 400;
PRINT "VERSUCHE: ";
PRINT Z
240 IF B>Z
THEN PROC SETR[%5B,%30];
PRINT " * REKORD **";
LET B=Z
250 PROC SETR[%5B,%40];
```

# JU+TE Computerclub

```
PRINT "BESTWERT:";
PRINT B;
PRINT "1-NEUES SPIEL";
INPUT "2-ENDE "U
260 IF U=1
    THEN GOTO 20
270 END
300 PROC SETR[%5B,0];
    PRINT "LANDSTRASSE";
    PRINT " ";
    PRINT " "
310 PROC SETR[%6D,0];
    PROC SETR[%5B,%10]
320 LET M=L;
    IF M>9
        THEN GOTO 340
330 PROC PTC[%30+M];
    PROC PTC[%2D];
    LET M=M+1;
    IF M<10
        THEN GOTO 330
340 PROC PTC[%31];
    PROC PTC[M$M10+48];
    LET M=M+1;
    IF M>12
        THEN RETURN
350 PROC PTC[%2D];
    GOTO 340
400 LET E=3
410 LET F=10,G=260
420 PROC SETR[%F1,%8A];
    PROC SETR[%F2,G];
    LET F=F-1,G=G-10;
    IF F>0
        THEN GOTO 420
430 LET E=E-1;
    IF E>0
        THEN GOTO 410
440 PROC SETR[%F1,10];
    PROC SETR[%6D,0];
    RETURN
```

## A/D Wandler C520D am JU+TE-Computer

Der mitunter auch als Basteltyp N 520 D angebotene Schaltkreis C 520 D wurde für die digitale Anzeige von analogen Spannungswerten entwickelt. Er hat den großen Vorteil, nur eine Versorgungsspannung (+5 V) und wenig äußere Beschaltung zu benötigen. Eingangsspannungen von 0 bis 999 mV ordnet er eine entsprechende dreistellige dezimale Ausgabe am Multiplexbetrieb zu. An den vier Datenausgängen A, B, C und D wird eine Dezimalziffer binär codiert ausgegeben, während einer der drei Auswahlausgänge (/LSD: niederwertige, /NSD: mittelwertige und /MSD: höchstwertige Stelle) die zugehörige Anzeigenposition mit 0-Pegel festlegt. Indem für jeweils etwa 2,5 ms zuerst die Messung (keine gültige Ausgabe), dann die Ausgabe der höchstwertigen, anschließend der niederwertigen und zuletzt der mittleren Ziffer erfolgen, kann über eine Multiplex-Anzeige ein scheinbar stehendes Bild entstehen. Die Übernahme des so codierten Meßwerts in einen Mikrorechner erfordert das Abwarten und Auswerten der drei Ausgabeschritte. Da der C 520 D eigenständig schwingt, kann sein Zyklus von außen nicht beeinflusst werden. Der Rechner muß sich anpassen. Von der Gerätetechnik erfordert das nicht mehr, als die sieben Ausgänge (open collector) des C 520 D im TTL-Pegel einem Eingabeter (Port) zuzuführen. Die in den Abbildungen darge-

stellte Lösung realisiert das mit minimalem Aufwand. Die Eingangsbeschaltung gestattet das Messen von Gleichspannungen von 0 bis 10 V bei einem Lastwiderstand von 200 k $\Omega$ .

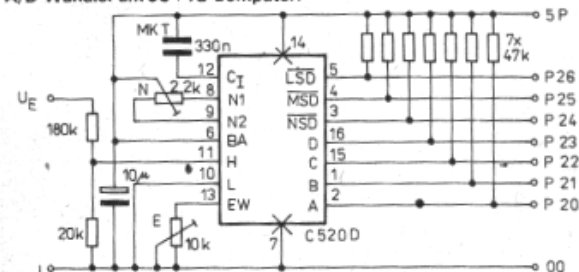
### Hardware-spezifische Unterprogramme

Während des Übergangs von einem Ausgabezustand des C 520 D zum anderen können für höchstens 2  $\mu$ s Schwingungen und damit unter Umständen auch ungültige Belegungen entstehen. Das erfordert einen geeigneten Störschutz. Wenn

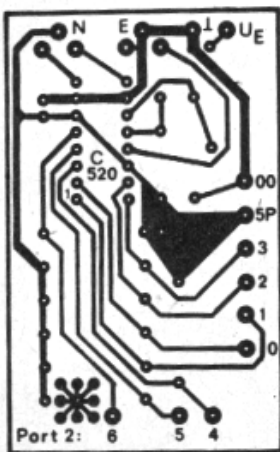
im Abstand von wenigstens 2  $\mu$ s übernommene Daten übereinstimmen, sind sie auch verlässlich.

Das Maschinen-Unterprogramm 1 realisiert solch eine Tastung. Der eingefügte nop-Befehl gewährleistet eine Zeitdauer von 3  $\mu$ s zwischen der Eingabe in Arbeitsregister r0 und in r1. Der anschließende Vergleich läßt ungültige Belegungen erkennen, die über den Sprungbefehl zur wiederholten Datenübernahme (Abtastung) führen. Eine gültige Tastung wird mit

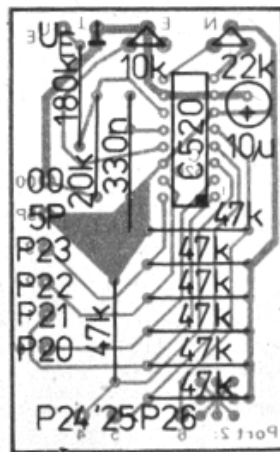
A/D-Wandler am JU+TE-Computer:



a Schaltbild;



b Leiterzugbild (1:1);



c Bestückungsplan

# JU+TE Computerclub

Bit-Manipulationsbefehlen so zerlegt, daß in r0 nur die Auswahlssignale /MSD, /NSD und /LSD, die den Zustand des C 520 D kennzeichnen, verbleiben und in r1 die BDC-Information (A, B, C und D) zum ASCII ergänzt. Auf diese Weise macht das Unterprogramm 1 dem Rechner die gerade vom C 520 D getätigte Ausgabe zugänglich. Das Erlassen eines vollständigen Meßwertes erfordert das Übernehmen der drei zugehörigen Ziffern in der gleichen Reihenfolge, wie sie der C 520 D anbietet. Hierzu eignet sich das Maschinen-Unterprogramm 2. Es nutzt das erste zum Tasten und legt die drei Ziffern des Meßwertes in den Arbeitsregistern r2 (höchste), r3 (mittlere) und r4 (niedrigste) als ASCII ab.

## Verarbeitungsorientierte Unterprogramme

Das Unterprogramm 3 paßt das zweite an den BASIC-Interpreter des UB 8830 D und das Betriebssystem des JU+TE-Computers an. Es verbietet Interrupts, damit die Bildausgabe nicht die Datenübernahme vom C 520 D stört, und setzt den Registerpointer auf %20. Damit werden die Register, die sonst die BASIC-Variablen A, B und C speichern, als Arbeitsregister festgelegt. Vor dem Rücksprung mit dem Meßwert in den genannten Registern wird die Interruptannahme wieder freigegeben. Damit entsteht eine kurze Unterbrechung der Bildzeugung während der Meßwertübernahme. Auch das Speichern einer Meßwertfolge ist mit einfachen Mitteln möglich. Das Maschi-

nen-Unterprogramm 4 belegt für 64 Meßwerte je drei Speicherzellen ab der Adresse %E200. Dabei dienen die Arbeitsregister r5 dem Speichern der Registeradresse, r6 und r7 der Speicheradressierung, r8 als Meßwertzähler und r9 als Zeitzähler. Um die Eigenschaften des C 520 D optimal zu nutzen, wird dessen Zykluszeit als Zeitbasis verwendet. Im Arbeitsregister 11 (niederes Byte der Variablen F) erwartet das Speicherprogramm die Anzahl der C-520-D-Zyklen (von 1 bis 256), die zwischen jeweils zwei Tastungen (hier Meßwertspeicherungen) vergehen sollen. Die zwischenzeitlich erfaßten Meßwerte werden einfach ignoriert. Das Speichern erfolgt mit dem hierfür gut geeigneten Idei-Befehl, der automatisch Register- und Speicheradresse erhöht. Da die nötige Interruptsperre und das Einstellen des Registerpointers enthalten sind, läßt sich auch das Programm 4 direkt von einem BASIC-Programm rufen. Es werden die Variablen A, B, C, D und E genutzt. Das Maschinen-Unterprogramm 5 holt einen gespeicherten Meßwert aus dem RAM wieder in die Arbeitsregister r2, r3 und r4. Die Speicheradresse wird mit Hilfe der Variablen X gewonnen, die mit einer Zahl zwischen 0 und 63 angibt, welcher Wert geholt werden soll. Auch hier leistet der Idei-Befehl gute Dienste. Dieses Programm läßt sich mit der BASIC-Anweisung CALL %E356 rufen. Eine Interruptsperre ist hier nicht nötig, da der C 520 D nicht abgetastet wird. Das letzte Maschinenprogramm (Abb. 6) dient der Anzeige aus

den Arbeitsregistern r2, r3 und r4. In der linken oberen Bildschirmecke stellt es den Meßwert mit Komma (entsprechend der Eingangsbeschaltung des C 520 D) und nachgestelltem V dar. Außerdem wird aus den beiden höheren Dezimalstellen eine Zahl zwischen 1 und 50 abgeleitet, die in der BASIC-Variablen Y gespeichert auch eine graphische Anzeige vorbereitet. Die Hex-Liste (Abb. 7) faßt alle Maschinenprogramme für die Eingabe mit DATA zusammen. Auch der PROG-Modus läßt sich nutzen. Als Anfangsadresse der Markentabelle eignet sich %E1FF. So passen alle Programmbeispiele in das erste Kilo von %E000 bis %E3FF.

## BASIC-Programmbeispiele

Das Programm Messen (Abb. 8) löscht den Bildschirm sowie die Variable X und ruft anschließend die Unterprogramme 3 (Wert holen) und 6 (Wert anzeigen). Die graphische Ausgabe in der Zeile 30 (CALL %FCA0) setzt die Punktgraphik-Unterprogramme im Speicher des JU+TE-Computers voraus. Mit LET G=GTC wird eine Tastenbetätigung abgewartet, bevor der nächste Meßwert zur Anzeige kommt. Dieses extrem einfache Programm läßt sich auch zum Einstellen von Nullpunkt und Endwert des C 520 D nutzen. Dabei darf der Stellwiderstand N (22k) nicht an einen der beiden Anschlüsse gedreht werden. Ein Doppelpunkt in der Anzeige kennzeichnet Meßwerte unter Null, ein Semikolon dagegen über 9,99 V (Meßbereichsende). Die Nullpunktstellung gelingt bei offenem oder besser kurzgeschlossenem

# JU+TE Computerclub

nem Eingang. Anschließend kann nach Anlegen einer bekannten Spannung (optimal: 9V) mit dem Endwertregler (E) die entsprechende Anzeige auf dem Bildschirm erzeugt werden. Günstig für das Abgleichen ist das Ersetzen der Tastenabfrage durch WAIT 50. Das BASIC Programm Speicheroszi (Abb. 9) ruft das Unterprogramm 4 zum Speichern von 64 Meßwerten. Die Tastzeit ( $1 \mu = 10 \text{ ms}$ ) wird zuvor in die Variable F eingegeben. Nach dem Messen kommt ab Zeile 30 der erste Wert zur Anzeige. Zum Erleichtern der Orientierung erscheint neben dem Zahlenwert die aktuelle Belegung der Variablen X. Die Zeile 40 erzeugt nach kurzem Blinken die graphische Anzeige und ruft die dynamische Tastenabfrage mit dem Ergebnis in G. Keine Betätigung ( $G=0$ ) schließt den Programmzyklus zur Zeile 40. Bei ENTER wird abgebrochen, bei + der nächste und bei - der vorherige Meßwert angezeigt. Die Zeile 90 beschränkt X auf den Bereich von 0 bis 63. Die Hexliste (Abb. 10) enthält die Punktgraphik-Unterprogramme für alle Leser, die die betreffenden Artikel im JU+TE-Computerklub nicht mehr zur Verfügung haben. Alle hier veröffentlichten Programme sollen als Beispiel dienen und das Umsetzen eigener Meßprobleme in geeignete Software anregen. Zum Messen von Gleichspannungen bis 100 V kann ein zusätzlicher Vorwiderstand  $R = 1,8 \text{ M}\Omega$  verwendet werden (Vorsicht bei Spannungen ab 60 V: Unfallgefahr!). Die Spannungsmessung in mV ist bei Ersetzen des 180-k $\Omega$ -Widerstandes durch eine

Brücke möglich. Mit dieser Brücke und einem 1-Q-Widerstand statt der 20 k $\Omega$  können auch Ströme gegen Masse (00) in mA erfaßt werden. Das Messen mit feinerer Auflösung und von Wechselgrößen erfordert einen entsprechend beschalteten Operationsverstärker vor dem C 520 D.

*Dr. Helmut Hoyer*

Layout:Hoyer, Zeichnungen: Liebig

## UP C520-Abtastung 1

```
*TAS: 08 02      C520-Ausgänge
E302 FF          Warten
E303 18 02      2. Testung
E305 A2 01      gleich ?
E307 EB :TAS    wenn nicht
E509 56 E0 70   nur Zustand
E30C 56 E1 0F   nur BCD
E30F 46 E1 30   ASCII
E312 AF         Rücksprung
```

## UP Analogwert holen 2

```
*ADU: 06 :TAS    C520-Tastung
E316 A6 E0 50    MSD ?
E319 EB :ADU     wenn nicht
E31B 28 E1       r2:=MSD-ASCII
*AD1: D6 :TAS    C520-Tastung
E320 A6 E0 30    LSD ?
E323 EB :AD1     wenn nicht
E325 48 E1       r4:=LSD-ASCII
*AD2: D6 :TAS    C520-Tastung
E32A A6 E0 60    NSD ?
E32D EB :AD2     wenn nicht
E32F 38 E1       r3:=NSD-ASCII
E331 AF         Rücksprung
```

## UP Wert holen (für BASIC) 3

```
*GET: 8F         Int.-sperre
E333 31 20       Registerpointer
E335 D6 :ADU     Wert holen
E338 9F         Int.-Freigabe
E339 AF         Rücksprung
```

## UP 64 Werte speichern 4

```
*STO: 8F         Int.-sperre
E33B 31 20       Registerpointer
E33D 6C E2       RAM-Adresse H
E33F 7C 00       RAM-Adresse L
E341 8C 40       Zähler := 64
*ST2: 98 EB      Tastzahl aus F
*ST1: D6 :ADU     Testwert holen
E348 9A :ST1     F mal
E34A 5C 22       Registeradr.
E34C 93 56       MSD speichern
E34E 93 56       NSD speichern
E350 93 56       LSD speichern
```

```
E352 8A :ST2     64 mal
E354 9F         Int.-Freigabe
E355 AF         Rücksprung
```

## UP Wert vom Speicher holen 5

```
*LAD: 31 20      Registerpointer
E358 6C E2       RAM-Adresse H
E35A 78 4F       L-Byte von X
E35C 02 77       *2
E35E 04 4F E7    *3 als Adr. L
E361 5C 22       Registeradr.
E363 83 56       MSD holen
E365 83 56       NSD holen
E367 83 56       LSD holen
E369 AF         Rücksprung
```

## UP Anzeige eines Wertes 6

```
*ANZ: 31 20      Registerpointer
E36C 80 5B       Cursor
E36E 29 5A       MSD
E370 D6 08 27    Anzeige
E373 E6 5A 2C    Komma
E376 D6 08 27    Anzeige
E379 39 5A       NSD
E37B D6 08 27    Anzeige
E37E 49 5A       LSD
E380 D6 08 27    Anzeige
E383 E6 5A 56    V
E386 D6 08 27    Anzeige
E389 08 E2       MSD
E38B 18 E3       NSD
E38D 56 E0 0F    nur MSD-BCD
E390 56 E1 0F    nur NSD-BCD
E393 D0 E1       NSD/2
E395 02 10       +MSD
E397 90 E0       MSD*2
E399 90 E0       *4
E39B 02 10       NSD/2+5*MSD
E39D 1E          +1
E39E 19 51       L-Byte von Y
E3A0 AF         Rücksprung
```

## Hex-Listing C520-UPS 7

```
E300: 08 02 FF 18 02 A2 01 EB
E308: F7 56 E0 70 56 E1 0F 46
E310: E1 30 AF D6 E3 00 A6 E0
E318: 50 EB F8 28 E1 D6 E3 00
E320: A6 E0 30 EB F8 48 E1 D6
E328: E3 00 A6 E0 60 EB F8 38
E330: E1 AF 8F 31 20 D6 E3 13
E338: 9F AF 8F 31 20 6C E2 7C
E340: 00 8C 40 98 EB D6 E3 13
E348: 9A FB 5C 22 93 56 93 56
E350: 93 56 8A EF 9F AF 31 20
E358: 6C E2 78 4F 02 77 04 4F
E360: E7 5C 22 83 56 83 56 83
E368: 56 AF 31 20 B0 5B 29 5A
E370: D6 08 27 E6 5A 2C D6 08
E378: 27 39 5A D6 08 27 49 5A
E380: D6 08 27 E6 5A 56 D6 08
E388: 27 08 E2 18 E3 56 E0 0F
E390: 56 E1 0F D0 E1 02 10 90
E398: E0 90 E0 02 10 1E 19 51
E3A0: AF 00 00 00 00 00 00 00
```

## BASIC-Programm Messen 8

# JU+TE Computerclub

```

10  CALL %8DD;
    LET X=0
20  CALL %E332;
    CALL %E36A;
30  CALL %FCA0;
    LET G=GTC
40  LET X=X+1;
    GOTO 20

```

## BASIC-Programm Speicheroszi 9

```

10  CALL %8DD;
    PRINT "+ VOLTMMETER + ",
    INPUT "TASTZEIT:" F
20  CALL %E33A;
    CALL %8DD;
    LET X=0
30  CALL %E356;
    CALL %E36A;
    PRINT " X"X
40  CALL %FC80;
    CALL %FCA0;
    PROC SETR[%6D,0];
    CALL %C56;
    LET G=GETR[%6D]$A%7F
50  IF G=0
    THEN GOTO 40
60  IF G=%D
    THEN END
70  IF G=%2B
    THEN LET X=X+1
80  IF G=%2D
    THEN LET X=X-1
90  LET X=X$A%3F;
    GOTO 30

```

## Hex-Listing Punktgraphik 10

```

FCA0: 70 FD 31 70 D6 FC CE 60
FCA8: E2 52 32 92 30 50 FD AF
FCB0: 70 FD 31 70 D6 FC CE 42
FCB8: 32 8B F0 70 FD 31 70 D6
FCC0: FC CE B0 52 B0 53 72 23
FCC8: EB E3 20 53 8B DF 0C FF
FCD0: 1C F8 B0 E2 38 51 56 E3
FCD8: 3F 90 E3 90 E3 10 E3 10
FCE0: E2 22 13 32 02 2C 01 38
FCE8: 4F 56 E3 3F 3E 00 E3 E0
FCF0: E2 76 E3 07 EB F7 90 E3
FCF8: F0 E3 02 13 82 30 AF 00

```

## Leservorschlag: Meßvorsatz für JU+TE-Computer

Mit Hilfe einer sehr einfachen Schaltung (Abb.) läßt sich der JU+TE-Computer zum Messen von Widerständen und Kapazitäten einsetzen. Dazu werden ein Monoflop (B 555 D) gestartet und die von einer R-C-

Kombination bestimmte Haltezeit gemessen. Zum Bestimmen von Kapazitäten braucht man einen bekannten Widerstand  $R_X$  für die Widerstandsmessung entsprechend eine bekannte Kapazität  $C_X$ .

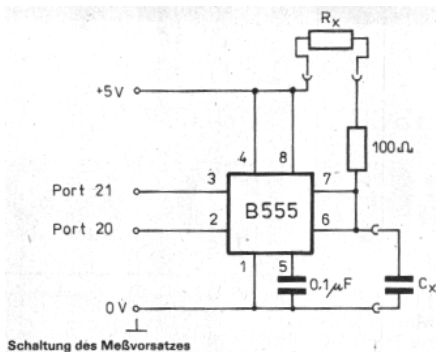
Für die Periodendauer  $t$  (in ms) gilt

$$t = 1,1 \cdot R_X \cdot C_X$$

( $C_X$  in  $\mu F$ ,  $R_X$  in  $\Omega$ ). Unser Programmbeispiel (Abb. S.943 Mitte) soll zeigen, wie einfach die Periodendauer  $t$  bestimmt werden kann. Mit  $R_X = 4,2 \text{ M}\Omega$  erfolgt die Anzeige hier in nF, mit  $R_X = 4,2 \text{ k}\Omega$  in  $\mu F$ . Zur Widerstandsmessung in  $\text{k}\Omega$  mit diesem Programm muß als  $C_X = 4,2 \mu F$  ein (oder mehrere parallel geschaltete) hochwertiger Kondensator (z. B. MKT) eingesetzt werden. Das Messen kleinerer Widerstände und Kapazitäten gelingt mit Maschinenprogrammen, die viel schnellere Zählschleifen zulassen.

Die Genauigkeit hängt wesentlich von der Güte der verwendeten Bauelemente ( $R_X$ ,  $C_X$ ) ab. Der Meßfehler kann unter 1% gedrückt werden. Bei Anwendung von Heißeleitern und Fotowiderständen lassen sich auch Temperatur und Lichtstärke mit dieser einfachen Methode bestimmen. Das eignet sich als Grundlage für Programme zur Belichtungssteuerung im Fotolabor oder zur Datenerfassung im Gewächshaus.

Michael Schmiededecke



```

10  LET Z=0
20  PROC SETR[%FB,%10]
30  PROC SETR[%F8,%92]
40  PROC SETR[%F7,9]
50  PROC SETR[%F6,2]
60  PROC SETR[2,1];
    PROC SETR[2,0];
    PROC SETR[2,1]
70  IF GETR[2]=1
    THEN GOTO 100
80  LET Z=Z+1;
    GOTO 70
100 PROC SETR[%FB,%90]
110 PRINT Z
200 END

```

## EPROM-Brennen leichtgemacht

Die Redaktion erreichen zahlreiche Anfragen zum Brennen von EPROM mit dem 4 K Betriebssystem des JU+TE-Computers.

Wir möchten allen Interessenten mitteilen, daß das Brennen der EPROM U 2716 C bzw. U 2732 C kostenlos über die Redaktion vermittelt wird. Dazu bitten wir Euch, die EPROM sicher verpackt an uns zu senden und dabei nicht das Rückporto zu vergessen (0,90 M in Briefmarken). Von der Einbis zur Rücksendung benötigen wir etwa vier Wochen. Wir weisen hier auch gleich darauf hin, daß nach unseren Erfah-



## JU+TE Computerclub

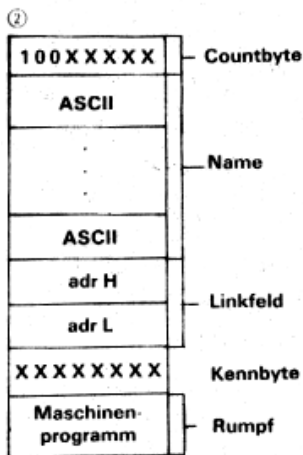
rungen die Fehlerrate beim U 2732 C sehr hoch liegt. Noch eine Bitte: Wer beim kostenlosen Brennen von EPROM mit dem Betriebssystem des JU+TE-Computers mithelfen möchte, sende uns seine Adresse. Wir geben sie an Interessenten, die in der Nähe des Wohnortes des Anbieters liegen, weiter. Auf diese Weise konnten vielleicht sogar Interessengemeinschaften „JU+TE-Computer“ entstehen. Wir würden das jedenfalls begrüßen.

*Die Redaktion*

## FORTH-83-System für JU+TE-Selbstbau- Computer

Mit der leistungsstarken Programmiersprache FORTH-83 erhält der JU+TE-Computer neue Anwendungsmöglichkeiten. Mit diesem Beitrag wird ein von Kassette ladbares FORTH-System beschrieben. Nur für Leser, die diese Sprache bereits kennen, ist es allerdings in allen Einzelheiten sofort verständlich. Diejenigen, die zum Beispiel mit unserem ABC den Einstieg beginnen (vgl. dazu S. 67 ff. in diesem Heft), sollten diese Artikel in Griffnähe behalten, um sich gleichzeitig mit FORTH-83 und der Handhabung des JU+TE-Systems vertraut zu machen. Die Redaktion vermittelt übrigens gern das kostenlose Überspielen des JU+TE-FORTH auf eingesandte Magnetbandkassetten (möglichst CrO<sub>2</sub>; Rückporto in Briefmarken bitte nicht vergessen).

```
` , -TRAILING .( >BODY ABORT
ABORT" COMPIL CONVERT CREATE
DOES> EXECUTE IMMEDIATE LITERAL
PAD SPAN [ '[' ] [COMPIL] ]
```



### f Anzeige der Wörterbuchgrenze:

HEX HERE U. DECIMAL

### „ Anzeige des Stackinhalts:

```
: S. 78 SP@ DO I
  BUSIN I 1+ BUSIN
  256 UM* DROP + CR
  2 +LOOP ;
: STACK DEPTH 0> IF
  S. ELSE ." LEER"
  THEN ;
```

### ... Eintragen von TOS ins Wörterbuch (n --):

```
: , HERE 2 ALLOT ! ;
```

### Wichtige FORTH-Worte außerhalb des Standards

VIEW ( -- )  
Anzeige einer Wortdefinition:  
VIEW (name)

LINK (adr -- )  
Fortsetzung des Wörterbuchs  
ab Adresse adr.

INPUT ( -- d)

Zahleneingabe mit doppelt  
genauem Ergebnis. Die Vari-  
able DPL wird aktualisiert.

BUSIN (adr -- n)  
Lesen des Registers mit der  
Adresse adr (untere acht Bit) in  
die unteren acht Bit von n.

BUSOUT (n adr -- )  
Laden des Registers mit der  
Adresse adr (untere acht Bit)  
mit unteren acht Bit von n.

### Konzeption

Der JUGEND+TECHNIK-Selbstbau-Computer ist durch geringen Hardwareaufwand gekennzeichnet und bietet keinen großen Spielraum für höhere Programmiersprachen. Er eignet sich eher für Steuerungsaufgaben als zur Textverarbeitung oder Datenbankverwaltung. Unter diesen Bedingungen ist FORTH die geeignetste Hochsprache, da sie sowohl maschinennahe Programmierung als auch allgemeine Formulierungen unterstützt. Sehr willkommen sind natürlich weitere Vorzüge wie die gute Ausschöpfung der Rechenleistung zur Ausführung von Anwenderprogrammen. Das JU + TE-FORTH System setzt 8 Kbyte RAM (voll bestückter Speichermodul auf Platz 1) voraus. Die ersten fünf Kilobyte benötigt der FORTH-Kern, für den Anwender bleiben die restlichen beiden Kilo bis %FC00, wo der Magnetbandpuffer beginnt. Da FORTH deutlich weniger Speicherkapazität als BASIC in Anspruch nimmt, bieten 2 Kbyte Anwender-RAM sehr viel Platz. Das FORTH-System wird von Kas-

# JU+TE Computerclub

sette geladen und läßt sich ab %E000 über PROG oder BASIC starten. Beim Sichern (SAVE) von Anwenderprogrammen muß der Kern mitgenommen werden, da hier z. B. die Systemvariable DP, die die Endadresse speichert, enthalten ist. Am einfachsten ist es, stets den RAM Bereich von %E000 bis %FBFF auf Kasette zu speichern. Das JU+TE-FORTH-System ist ein FORTH-83-Standard-Subset mit Standarderweiterungen für doppelt genaue Zahlen. Von den 132 Worten des sogenannten notwendigen Wortschatzes fehlen die für die Screenarbeit (Quelltextspeicherung) und für die Teilung des Wörterbuchs in Vokabulare nötigen sowie die zwanzig in Abb. 1 aufgeführten. Diese werden zum größten Teil implizit vom System realisiert, lassen sich aber nicht völlig standardgerecht in Anwenderprogramme einbauen.

## Besonderheiten

Auf die Screenarbeit wurde verzichtet, da mit der Bildschirmgröße und RAM-Ausstattung des JU+TE-Computers zu enge Grenzen gesetzt sind. Statt dessen können mit dem Wort VIEW alle Wörterbucheinträge rückübersetzt werden. Die Vorsätze LIT und DLIT kennzeichnen dabei mit einfachen (16 bit) und mit doppelter (32 bit) Genauigkeit gespeicherte Zahlen. Sonst erzeugt VIEW die gleiche Anzeige, die für die Eingabe von Doppelpunktdefinitionen nötig ist. Maschinenprogramme kennzeichnet VIEW mit CODE. Anschließendes Betätigen der ENTER-Taste bringt den Maschinencode im DATA-Modus,

der mit OFF wieder verlassen werden kann, zur Anzeige. Jede andere Taste unterdrückt diese Ausgabe. Die Übersicht über das Wörterbuch ist damit ganz einfach: WORDS bringt die Namen (jeweils sieben untereinander) zur Anzeige, wobei mit der ENTER-Taste fortgesetzt und mit jeder anderen abgebrochen werden kann. VIEW mit anschließender Namenseingabe zeigt die Wortdefinition auf dem Bildschirm. Damit kann der Quelltext mit Papier und Bleistift leicht verwaltet und sogar nachträglich erzeugt werden. FORTH ist flexibel genug, um auch mit minimaler Hardware auszukommen. Das JU+TE-FORTH-System speichert alles im Wörterbuch (in-line) und benötigt daher keine weiteren Dateien. Beim Übersetzen von Eingaben entsteht direkt das entsprechende Maschinenprogramm, was sogar den sonst üblichen Abstand zwischen der Spitze des Wörterbuchs und dem Pufferspeicher PAD überflüssig macht. Das erste Byte eines Wörterbucheintrags enthält nur den Count, die Kennzeichnung als normales (%00) oder als Immediate Wort (%01) folgt als zusätzliches Byte dem Linkfeld. Es wurde zum Kennzeichnen von Strukturwörtern (z. B. DO) und zur Steuerung des Rückübersetzens eingeführt. Der Rumpf ist ein ausführbares Maschinen-Unterprogramm, das sowohl Codefeld als auch Parameterfeld realisiert (Abb. 2). Das Wort > BODY ist damit ohne Sinn. Zum sofortigen Ausführen wird die Eingabe in einen solchen Rumpf an die Spitze des Wörterbuchs übersetzt und anschließend gerufen. Wenn

sich das Wörterbuch nach vielen Erweiterungen einmal der oberen Anwender-RAM-Grenze %FBFF gefährlich nähert, kann mit dem Wort LINK die Fortsetzung an beliebiger anderer Stelle (auf einem weiteren Speichermodul) ausgelöst werden. LINK nimmt die Fortsetzungsadresse vom Stack. Die aktuelle Endadresse des Wörterbuchs läßt sich mit Hilfe von HERE jederzeit anzeigen (Abb. 3).

Für die Zahleneingabe gelten allgemein übliche Regeln: Die Ziffern werden mit Hilfe der aktuellen Zahlenbasis (Variable BASE) in eine einfach genaue Dualzahl gewandelt. Enthält das Zahlwort jedoch einen Dezimalpunkt, erfolgt die Weiterverarbeitung mit doppelter Genauigkeit. Die Variable DPL erhält die Anzahl der rechts vom Dezimalpunkt stehenden Ziffern (DPL := -1, wenn kein Punkt). Ein Minuszeichen im Zahlwort bewirkt das Negieren des gewandelten Wertes.

Die in Standardprogrammen übliche Zahleneingabe mit SPAN und CONVERT läßt sich mit dem JU+TE-FORTH Kern nicht standardgerecht ausführen. Die Schwierigkeiten resultieren aus der Nutzung des BildwiederholSpeichers als Eingabepuffer. Statt dessen gibt es hier das Wort INPUT, das eine Zahleneingabe mit ENTER als Abschluß fordert und als doppelt genaue Zahl auf dem Stack hinterläßt. FORTH arbeitet mit Speicheradressen. Zum Standard gehören keine Worte, die sich auf Register beziehen. Passend zum FIG-FORTH für die Kleincomputer KC 85 enthält das JU+TE-FORTH-System die Wörter BUSIN und BUSOUT,

# JU+TE Computerclub

die wie @ und ! wirken, aber mit Register- statt Speicheradressen arbeiten (KC: E/A-Adressen). Da sich der Stack in den Registern %20 bis %4F des Einchip-Mikrorechners befindet, muß auch BUSIN statt @ für die Anzeige des Stackinhalts verwendet werden. *Abb. 4* gibt zwei dafür gut geeignete Definitionen. Das Zeichen @ wird übrigens mit SHIFT P eingegeben. Das Reservieren von Freibereich im Wörterbuch läßt sich allgemein mit n ALLOT (ENTER) ausführen. Bei n = 9 und n = 10 muß beim JU+TE-FORTH nach n die ENTER-Taste zusätzlich betätigt werden, damit sich die compilierte Eingabe und das zu verschiebende Linkfeld nicht gegenseitig stören. Das hängt damit zusammen, daß als Eintritt für die Suche nach Wortdefinitionen an der Spitze des Wörterbuchs (HERE) einfach die Adresse des zuletzt eingetragenen Wortnamens (Linkfeld) steht. FORTH lebt grundsätzlich vom Erweitern des Wörterbuchs. Fehlende Standardworte sind kein großes Problem, da sie sich grundsätzlich nachtragen lassen. (Das Überführen von TINY-BASIC in extended BASIC ist dagegen z. B. völlig unmöglich.) Die in *Abb. 1* aufgezählten Wörter werden zumindest im JU+TE-FORTH-System kaum benötigt (vgl. INPUT für SPAN und CON-  
VERT). Am ehesten wird CREATE vermißt. Es läßt sich mit VARIABLE (name) -2 ALLOT ersetzen. Das in Verbindung mit CREATE interessante Komma kann, wie in *Abb. 5* dargestellt, nachgetragen werden.

Das Wort J, das bei verschachtelten DO ... LOOP-Strukturen den Index der nächstäußeren Schleife zum Stack kopiert, setzt voraus, daß die innere Schleife als Wort definiert wurde und so nur mit dem Namen in der äußeren erscheint. Das trägt den Stillforderungen Rechnung, die nur eine Struktur wie DO ... LOOP in einer Doppelpunktdefinition zulassen.

## Maschinenprogramme

Das Wort CODE gestattet Wortdefinitionen in Maschinensprache. Nach Eingeben des Wortnamens und Betätigen der ENTER Taste ruft das JU+TE-FORTH-System das Betriebsprogramm DATA mit dem ersten Byte des Rumpfes auf der Eingabeadresse. Ab dort kann das Maschinenprogramm eingetragen werden. Der Registerpointer steht beim Rufen von Maschinenprogrammen auf %10. Die Arbeitsregister r2 und r3 speichern den TOS. Zum Rechnen stehen r4 bis r11 zur Verfügung. Auch die Register %02 bis %0F (einschließlich der Ports) dürfen benutzt werden. Beim Zugriff auf den Bereich ab %50 ist das Betriebssystem zu berücksichtigen. Die Register %20 bis %4F bleiben für den Stack reserviert und damit tabu. Stackpointer ist das Arbeitsregister r15. Diese Aufteilung gestattet das Verwenden der allgemein nutzbaren Betriebsunterprogramme (vgl. JU+TE 5/89 S. 380) auch für das FORTH-System.

Nach Eingabe des letzten Bytes des Maschinenunterprogramms (meist AF)

wird die Definition mit OFF abgeschlossen. Dabei verwendet das FORTH-System die aktuelle Eingabeadresse als HERE (Adresse des ersten freien Bytes). Es ist daher nicht gleichgültig, in welchem Zustand der DATA-Modus verlassen wird. Soll die automatische Adreßberechnung (Binden) zum Einsatz kommen, müssen die betreffenden Operandenbytes bei der Eingabe zunächst frei bleiben (z. B. %FF). Nach Eingabe dieses unfertigen Programms unter FORTH kann es mit VIEW angezeigt werden. Über RESET und PROG ist das Nachtragen der Marken und schließlich das Binden nötig. Nach dem Neustart des FORTH-Systems steht das neue Wort dann zur Verfügung. Die Adresse der Markentabelle sollte bei diesem Verfahren vor Beginn der Definition des Maschinenprogramms bereits vereinbart sein, damit PROG gleich zur richtigen Adresse führt.

## Anwendung

Das JU+TE-FORTH-System besitzt eine hohe Ausführungsgeschwindigkeit. Im Vergleich zu anderen FORTH-Systemen dauert jedoch die DO-LOOP-Schleife länger, da das erforderliche Verwalten von Index und Grenze im Return-Stack beim Einchip-Mikrorechner größere Umstände macht. Das Übersetzen von Eingaben erledigt ein Interpreter. Die hier spürbaren Laufzeiten beeinträchtigen aber nicht die Geschwindigkeit beim Ausführen auf diese Weise erzeugter Maschinenprogramme (im Rumpf).

# JU+TE Computerclub

Die Übertragbarkeit von FORTH-Programmen auf andere Rechner setzt voraus, daß in den Definitionen ausschließlich der Standardwortschatz verwendet wird. Maschinenprogramme, die mit CODE erfaßt wurden, zählen grundsätzlich nicht dazu. Übertragbar ist stets der Quelltext, nicht jedoch das auf Kassette speicherbare Wörterbuch. Das hat auf anderen Rechnertypen mit Sicherheit Abweichungen, auch wenn das gleiche FORTH-Programm kompiliert wurde. Viel Spaß beim Programmieren in FORTH!

*Dr. Helmut Hoyer*

## Software per Kasette

Zunächst noch ein Hinweis zum FORTH-System, das auf an JU+TE eingesandte Magnetbandkassetten geladen wird: Dieses Programm nutzt den RAM-Modus des 4-K-Betriebssystems zur Anzeige und zur Eingabe von Maschinenprogrammen in FORTH-Wörtern. Befindet sich das Betriebssystem in einem U 2732 C oder U 2764 C, gilt die Startadresse %1352. Ist das Betriebssystem in zwei U 2716 C geteilt, beginnt der RAM-Modus jedoch auf %2352. In diesem Fall müssen im FORTH-System die Bytes auf den Adressen %EBEB und %EDD3 von %13 auf %23 geändert werden, damit die Anwendung des Wortes CODE klappt. Für diese Änderung eignet sich der RAM-Modus, nach Laden des FORTH-Systems aus dem Resetmenü gestartet, am besten.

Auf Wunsch werden auch veröffentlichte BASIC Spiele von uns gleich mit auf Kassette geladen. Ein neues Angebot erhielten wir von H. Dömlt aus Kamenz: Das sehr unterhaltsame Spiel „Steine“, bei dem es gilt, hell dargestellte Diamanten zu sammeln, ohne von dunkel dargestellten Steinen erschlagen zu werden. Die Spielfigur bewegt sich mit der Leertaste (links), und den Tasten X (rechts), – (hoch) und ENTER (runter) durch den Sand und gibt dabei Steinen und Diamanten Gelegenheit, nach unten zu fallen. Die Spielfigur kann alles stützen, aber nichts auffangen. Für gesammelte Diamanten gibt es Punkte, die in der Kopfzeile mit B: angezeigt werden. Ab 250 lassen sie sich am rechteckig dargestellten Endfeld abliefern. Wo es sich befindet, muß beim Sammeln nebenbei erkundet werden. Die angelieferten Punkte erscheinen in der Kopfzeile hinter P: und kommen damit in die Wertung. Nach dem Abliefern geht es mit einem neuen Spielfeld weiter. In ausweglosen Situationen kann mit CLR neu begonnen werden. Nach dem fünften Unfall oder Neubeginn erfolgt die Auswertung. Steine ist ein Maschinenprogramm und kann nach dem Rücksetzen mit P FBFF G E000 L gestartet werden. Ein weiterer Eintrittspunkt ist die Adresse %F5C0. Hier bietet ein Menü das Korrigieren und völlige Neugestalten von Spielfeldern mittels der Zifferntasten an. Mit der dritten Komponente dieses Menüs (Rücken) läßt sich die neue Kreation in den aktiven Set einbeziehen. Wer also auf Modul 1 voll bestückten RAM (8

Kbyte) realisiert hat, ist in der Lage, auch dieses Spiel auszuprobieren. Dazu braucht nur eine Magnetbandkassette an uns gesandt zu werden (Rückporto in Briefmarken bitte nicht vergessen). Wenn es beim Einlesen Schwierigkeiten geben sollte, ist mit dem Anpassen der Tonkopfneigung (auf maximale Höhen) in den meisten Fällen erfolgreiches Laden erzielbar. Viel Spaß also mit den neuen Software-Angeboten!

## Höher auflösende Bildschirmsteuerung

Wie bereits angekündigt, ist eine höher auflösende Bildschirmsteuerung für den JU+TE-Selbstbau-Computer in Entwicklung. Deren Veröffentlichung ist für Mitte dieses Jahres vorgesehen. Aus Beschaffbarkeitsgründen wird kein spezieller Videocontroller eingesetzt, sondern ein 40poliger EMR-Schaltkreis. 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. Das Bitmuster des inneren ROM spielt keine Rolle. 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

# JU+TE Computerclub

volle Ausbaustufe der neuen Bildschirmsteuerung erreicht: Bei 40 Zeichen je Zeile ist jeder Bildpunkt einzeln in 16 Farben darstellbar.

Damit die Bauelementbeschaffung gut klappt, veröffentlichen wir auf Wunsch zahlreicher Leser vorab die Liste der für minimalen Ausbau nötigen Schaltkreise, die neben dem Prozessor und dem Speichermodul nötig sind:

- 1 x U 2716
- 1 x DL 000
- 1 x DL 004
- 1 x DL 010
- 1 x DL 074
- 1 x DL 257
- 1 x DL 295
- 1 x DL 299
- 4 x DL 540
- 1 x DS 8205
- 1 x DS 8212
- 1 x DS 8286

Republik wurde, nicht durch unsere Schuld, kurz vorher abgesagt. Leider konnten wir Euch wegen unseres langen technischen Vorlaufs nicht mehr selbst rechtzeitig informieren, „nur“ über DT 64. Wir bitten um Entschuldigung.

## ***Achtung! Computertreff!***

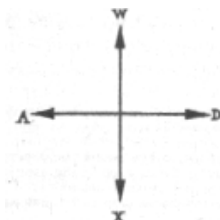
Am Sonnabend, dem 24. März 1990, lädt JU+TE alle Freunde des JU+TE-Computers sowie die Redaktion practica die Freunde des Mikrorechnerbausatzes Z1013 von 10 bis 15 Uhr ein in die Betriebsschule der Bezirksdirektion der Deutschen Post in Magdeburg, Lorenzweg (Nähe Tankstelle Albert-Vater-Str.). Unser gemeinsames Programm reicht vom kostenlosen EPROM-Brennen bis zum Softwareservice. Auch der „JU+TEComputervater“ Dr. Hoyer wird dabeisein. Bitte vergesst nicht Eure Rekorder mitzubringen. Unsere gleichartige Veranstaltung vom November 1989 im Palast der

## Solo-Halma

### Spielidee:

Auf einem Spielfeld von 7 x 7 Plätzen stehen 48 Spielsteine (symbolisiert durch Multiplikationszeichen), wobei das Mittelfeld unbesetzt ist. Nun springt man mit einem Spielstein über einen anderen und muß auf einem freien Feld landen. Der übersprungene Stein wird entfernt. Das Spiel ist beendet, wenn man keinen Spielstein mehr überspringen kann. Ziel ist es, möglichst wenig Spielsteine stehenzulassen.

### Spielablauf:



Mit Hilfe der Tasten W, X, A und D (vgl. Abb.) kann der Cursor beliebig über das Spielfeld geführt werden. Dabei ist auch Dauertastenbetätigung möglich. Ist man auf einer gewünschten Spielfeldposition angekommen und möchte springen, so betätigt man die ENTER-Taste und gibt die Sprungrichtung mittels W, X, A bzw. D an. Bei erfüllter Sprungbedingung wird der Sprung ausgeführt und der übersprungene Spielstein selbständig entfernt. Anderenfalls erfolgt eine Fehlermeldung. Das Spielende muß der Spieler selbst durch SHIFT-END anzeigen.

### Programmbeschreibung:

Mit N=8 und N=%18 wird in GOSUB 230 der Bildschirmpointer gestellt und die Überschrift außerhalb des Spielfeldes geschrieben. Anschließend erfolgt das Setzen der 7 x 7 Spielsteine (Zeile 40 und 50). Mit Zeile 60 wird der mittlere Spielstein gelöscht und der Cursor gesetzt. In Zeile 70 legt C=1 den Kursormodus fest. CALL %C1D (Zeile 300) ruft das Unterprogramm zur statischen Tastenabfrage ohne Ausführung auf dem Bildschirm auf (siehe JU+TE, Heft 5/89, S.380). Das im Register %5A stehende ASCII-Zeichen wird gelesen und dem Parameter D zur weiteren Auswertung zugewiesen. Die Zeilen 310 bis 340 bearbeiten die vier Bewegungsrichtungen und prüfen über GOTO 390 in GOSUB 500 die Kursorbewegung auf Verlassen des Spielfeldes. Mit dem Betätigen der ENTER-Taste (Zeile 370) wird der Sprung eines Spielsteines vorbereitet (C=2). Zunächst ertönt ein Beep (GOSUB 410), danach folgt der Rücksprung ins Hauptprogramm, wo in Zeile 90 wegen C=2 verzweigt wird. Die Zeilen 110 bis 130 überprüfen die drei Sprungbedingungen:

1. Ist ein Spielstein vorhanden, mit dem gesprungen werden kann (Zeile 110)?
2. Ist ein Spielstein vorhanden, der übersprungen werden kann (Zeile 120)?
3. Ist das Feld hinter dem zu überspringenden Spielstein frei (Zeile 130)?

Falls eine dieser drei Bedingungen nicht erfüllt ist, erfolgt rechts unten auf dem Bildschirm der Fehlerausdruck

"ERROR" (GOSUB 220) und der Rücksprung in den Kursormodus (C=1, Zeile 70). Bei erfüllter Sprungbedingung wird ab Zeile 140 der Sprung in drei Schritten ausgeführt:

1. Löschen des Spielsteines auf der Startposition (sie wurde in G gespeichert).
2. Löschen des übersprungenen Spielsteines (Adresse in H).
3. Setzen des Spielsteines auf das Zielfeld.

Danach erfolgt die Verringerung des Spielsteinzählers um 1 und der Rücksprung in den Kursormodus.

Durch die Tastenbetätigung SHIFT-END (Zeile 360) kann das Spiel beendet werden, indem C=3 gesetzt wird. Es ertönt eine Melodie (GOSUB 400). Über Zeile 100 folgt in Zeile 160 der Ergebnisausdruck.

Nach beliebigen Tastendruck beginnt ein neues Spiel. Das Programm benötigt knapp 1 Kbyte im BASIC-RAM.

*Bernhard Piniek*

```
10 LET B=48
20 CALL %8DD;
LET N=8;
GOSUB 230;
PRINT "SOLO-";
LET N=%18;
GOSUB 230;
PRINT "HALMA"
30 PROC SETRR[%F2,7];
PROC SETR[%F1,10];
LET A=0,N=%33,K=48
40 PROC SETR[%5B,A];
PROC PTC[%2A];
LET A=A+1;
IF A$A7=7
THEN LET A=A+9
50 IF A<%67
THEN GOTO 40
60 PROC SETR[%5B,%34];
PROC PTC[8]
70 LET C=1;
GOSUB 230
80 GOSUB 300;
IF C=1
THEN GOSUB 230;
```

# JU+TE Computerclub

```

GOTO 80
90 IF C=2
  THEN GOTO 110
100 GOTO 160
110 GOSUB 210;
  LET G=N;
  IF F<>%2A
  THEN GOSUB 220;
  GOTO 70
120 GOSUB 300;
  GOSUB 210;
  LET H=N;
  IF F<>%2A
  THEN GOSUB 220;
  GOTO 70
130 GOSUB 500;
  GOSUB 210;
  LET I=N;
  IF F<>%20
  THEN LET N=N-E;
  GOSUB 220;
  GOTO 70
140 LET N=G,J=%20;
  GOSUB 240;
  LET N=H,J=%20;
  GOSUB 240
150 LET N=I,J=%2A;
  GOSUB 240;
  LET K=K-1;
  GOTO 70
160 GOSUB 400;
  CALL %8DD;
  PRINT "ERGEBNIS:";
  PRINT K;
  PRINT "RESTSTEINE";
  PRINT " "
170 IF B>K
  THEN LET B=K;
  PRINT "*** REKORD ***"
180 PRINT "BESTWERT:";
  PRINT B;
  INPUT "1-NEUES SPIEL"U
190 IF U=1
  THEN GOTO 20
200 END
210 LET F=GETEB[%FD00+N];
  RETURN
220 PROC SETR[%5B,%68];
  PRINT "ERROR";
  WAIT 200;
  PROC SETR[%5B,%68];
  PRINT " ";
  RETURN
230 PROC SETR[%5B,N];
  RETURN
240 PROC SETR[%5B,N];
  PROC PTC[J];
  RETURN
300 PROC SETR[%6D,0];
  CALL %C1D;
  D=GETR[%5A]
310 IF D=%57
  THEN LET E=-16;
  GOTO 390
320 IF D=%58
  THEN LET E=16;
  GOTO 390
330 IF D=%41
  THEN LET E=-1;
  GOTO 390

```

```

340 IF D=%44
  THEN LET E=1;
  GOTO 390
350 IF C=2
  THEN GOTO 300
360 IF D=%55
  THEN LET C=3;
  RETURN
370 IF D=%D
  THEN LET C=2,L=10;
  GOSUB 410;
  RETURN
380 GOTO 300
390 GOSUB 500;
  RETURN
400 LET L=0
410 LET M=160
420 PROC SETR[%F1,%8A];
  PROC SETR[%F2,M];
  WAIT 10;
  IF L>10
  THEN GOTO 440
430 LET L=L+1,M=M+10;
  IF L<10
  THEN GOTO 420
440 PROC SETR[%F1,10];
  PROC SETR[%6D,0];
  RETURN
500 IF N+E$A15>6
  THEN GOTO 540
510 IF N+E<0
  THEN GOTO 540
520 IF N+E>%66
  THEN GOTO 540
530 LET N=N+E
540 RETURN

```

(Zeilen-Nr.1,2,3,4)

1	CBBAE	1	AAAAA
2	ACEBC	2	BBBBB
3	BEDDA	3	CCCCC
4	EADCE	4	DDDDD
5	DBCAD	5	EEEEE

67890 67890  
(Spalten-Nr.)

ben A bis E (Abb.) zu sortieren, so daß die Anordnung nach Abb. rechts entsteht. Dazu ist es möglich, die einzelnen Zeilen und Spalten zu verschieben. Das geschieht in zwei Schritten: zunächst wird die Zeilennummer (Zahl zwischen 1 und 5) bzw. die Spaltennummer (Zahl zwischen 6 und 0) eingegeben. Anschließend kann mit Hilfe der Bewegungstasten nach Abb. S. 142 die Zeile bzw. Spalte soweit wie gewünscht verschoben werden- Bei Spielende erscheint eine optische Anzeige. Durch beliebigen Tastendruck wird ein neues Spiel gestartet.

## Variablennutzung

A: Zähler für Spielfeldaufbau  
 B: Bestwert  
 C: Modus (Kursor, Spielzeug, Ende)  
 D: ASCII von Tastatur  
 E: Bewegungsrichtung  
 F: ASCII vom Bildspeicher  
 G: Startposition (Spielzug)  
 H: Löschesposition(Spielzug)  
 I: Zielposition (Spielzug)  
 J: ASCII für Anzeige  
 K: Spielsteinanzahl  
 L: Tonzähler  
 M: Tonhöhe  
 N: Bildposition

## Programm

Zu Beginn erfolgt die übliche Initialisierung der benötigten Parameter und das Laden des Zufallsgenerators. Die Zeilen 50 bis 140 realisieren den Spielfeldaufbau mit der Anordnung von 5 x 5 zufällig verteilten Buchstaben A bis E und der Zeilen- bzw. Spaltennumerierung. Dazu ist ein mehrmaliger Durchlauf der Zeilen 60 bis 130 erforderlich. Im ersten Durchlauf der Zeile 50 wird an die Bildschirmposition F=3 die Ziffer "1" als Zeilennummer geschrieben. Die Zeile 60 ermittelt in H Zufallszahlen zwischen 1 und 5 und weist sie den Buchstaben A bis E zu

## Schiebepuzzle

Der Speicherplatzbedarf für dieses Programm beträgt nur 1 Kbyte. Die Spielidee besteht darin, die auf dem Bildschirm zufällig angeordneten Buchsta-



# JU+TE Computerclub

(Zeilen 60 bis 110). Da jeder Buchstabe nur fünfmal vorhanden sein soll, erfolgt in jeder dieser Zeilen ein entsprechender Test. Bei negativem Testausgang wird eine neue Zufallszahl H ermittelt (Zeile 110: GOTO 60).

Zeile 120 realisiert die Bildschirmausgabe des ermittelten Buchstaben und steuert die nächste Bildschirmposition an ( $F = F+1$ ). Nachdem auf diese Weise die erste Zeile mit Buchstaben gefüllt wurde, werden durch Sprung nach Zeile 50 die nächste Zeilennummer und weitere fünf Buchstaben geschrieben. Ist das gesamte Spielfeld gefüllt, realisiert die Zeile 140 den Ausdruck der Spaltennummern durch eine einzige PRINT-Anweisung. Von Zeile 150 bis 250 erfolgt die Vorbereitung zum Verschieben von Zeilen bzw. Spalten. In den Zeilen 160 bis 250 werden in J die entsprechenden Startparameter für den Verschiebungsprozeß geladen. Außerdem erfolgt die Kennung, ob es sich um eine Zeilenverschiebung ( $K=0$ ) oder um eine Spaltenverschiebung ( $K=1$ ) handelt. Die Zeilenverschiebung kann nach rechts (Zeile 260:  $I=\%44 \wedge = 68$ ) oder links (Zeile 270) und die Spaltenverschiebung nach oben (Zeile 290) oder unten (Zeile 280) erfolgen. Entsprechend werden in den Zeilen 320 bis 350 die Zyklusparameter F und N gestellt. In Zeile 360 bis 380 gelangt der Schiebeprozess zur Ausführung. Nach jedem Schiebeprozess erfolgt das Prüfen auf Spielende (Zeilen 390 bis 420), wobei jeder einzelne Buchstabe dieser Prüfung unterliegt. Begonnen wird mit dem ersten Buchstaben in der ersten Zeile,

dessen Adresse %FD05 lautet. S speichert den dort gelesenen ASCII und vergleicht ihn mit R ( $=65 = \%41 \wedge = A$ ). Bei erfüllter Bedingung wird der nächste Buchstabe gelesen und mit R verglichen. Nach dem Prüfen der ersten Zeile erfolgen die Aktualisierung des Bildschirmpointers und des Parameters R ( $R=66 = \%42 \wedge = B$ ). Stehen alle 25 Buchstaben an der richtigen Stelle (vgl. Abb. rechts), so ist der Test  $R < 70$  in Zeile 420 nicht mehr erfüllt, und es wird das Spielende angezeigt. In jeder Spielphase kann das Spiel durch Betätigen der Taste E (wie „Ende“) abgebrochen werden (Zeile 300). Viel Spaß beim Puzzeln!

Bernhard Piniek

## Bedeutung der verwendeten Parameter

A – E	Zählparameter für die Buchstaben A bis E
F	Bildschirmposition
G	Ziffern 0 bis 9 für die Spielfeldbezeichnung
H	Zufallswert
I	Tastenbetätigung
J	Zeilen- bzw. Spaltenparameter
K	Zeilen- bzw. Spaltenkennung
L,M,N und X	Hilfsparameter zum zyklischen Zeilen- bzw. Spatendurchlauf
Q,R,S	Test auf Spielende
U,V,W	Melodie

```

10  CALL %8DD;
    PRINT "SCHIEBPUZZLE";
    WAIT 300
40  CALL %8DD;
    LET A=0,B=0,C=0,D=0,E=0;
    LET F=3,G=%31;
    PROC SETRR[%F2,%FF0F];
    PROC SETR[%F1,10]
50  PROC SETR[%5B,F];
    PROC PTC[G];
    LET F=F+2
60  LET H=GETR[%F2]%M5+1;
```

```

IF H=1
THEN IF A<5
THEN LET A=A+1;
GOTO 120
70  IF H=2
THEN IF B<5
THEN LET B=B+1;
GOTO 120
80  IF H=3
THEN IF C<5
THEN LET C=C+1;
GOTO 120
90  IF H=4
THEN IF D<5
THEN LET D=D+1;
GOTO 120
100 IF H=5
THEN IF E<5
THEN LET E=E+1;
GOTO 120
110 GOTO 60
120 PROC SETR[%5B,F];
    PROC PTC[H+64];
    LET F=F+1;
    IF F$A10<10
    THEN GOTO 60
130 LET F=F+9,G=G+1;
    IF G<%36
    THEN GOTO 50
140 PROC SETR[%5B,%65];
    PRINT "67890"
150 CALL %C56;
    LET I=GETR[%6D]$A%7F
160 IF I=%31
    THEN LET J=5,K=0
170 IF I=%32
    THEN LET J=%15,K=0
180 IF I=%33
    THEN LET J=%25,K=0
190 IF I=%34
    THEN LET J=%35,K=0
200 IF I=%35
    THEN LET J=%45,K=0
210 IF I=%36
    THEN LET J=0,K=1
220 IF I=%37
    THEN LET J=1,K=1
230 IF I=%38
    THEN LET J=2,K=1
240 IF I=%39
    THEN LET J=3,K=1
250 IF I=%30
    THEN LET J=4,K=1
260 IF I=68
    THEN IF K=0
    THEN GOTO 320
270 IF I=65
    THEN IF K=0
    THEN GOTO 330
280 IF I=88
    THEN IF K=1
    THEN GOTO 340
290 IF I=87
    THEN IF K=1
    THEN GOTO 350
300 IF I=%45
    THEN GOTO 10
310 PROC SETR[%6D,0];
    GOTO 150
320 LET F=J+4,N=-1;
```

# JU+TE Computerclub

```

      GOTO 360
330  LET F=J,N=1;
      GOTO 360
340  LET F=J+%45,N=-16;
      GOTO 360
350  LET F=J+5,N=16
360  LET L=GETEB[%FD00+F],X=1
370  LET M=GETEB[%FD00+F+N];
      PROC SETR[%5B,F];
      PROC PTC[M];
      LET F=F+N,X=X+1;
      IF X<5
      THEN GOTO 370
380  PROC SETR[%5B,F];
      PROC PTC[L]
      PROC SETR[%6D,0]
390  LET Q=5,R=65
400  LET S=GETEB[%FD00+Q];
      IF R<>S
      THEN GOTO 150
410  LET Q=Q+1;
      IF Q$A10<10
      THEN GOTO 400
420  LET R=R+1,Q=Q+11;
      IF R<70
      THEN GOTO 400
430  PROC SETR[%5B,%65];
      PRINT "ENDE!";
      LET X=GTC;
      GOTO 10
```

## Fertigtastaturen am JU+TE-Computer

*Da sich das Angebot von Rechner-Tastaturen für Amateure deutlich verbessert hat, liegt es nahe, den Bedienkomfort des JU+TE-Computers zu erhöhen. Wir stellen Euch deshalb eine Möglichkeit vor, mit der im Prinzip alle gehandelten Matrizen mit mechanischen oder Elastomerkontakten angepaßt werden können. Hall-Taster und Leiterplatten mit zusätzlichen Schaltkreisen machen in der Regel größere Schwierigkeiten, da sie mit hoher Wahrscheinlichkeit nicht dem Abtast-Prinzip des JU+TE-Computers gerecht werden.*

Unsere Abbildungen zeigen geeignete Tastaturen. Sie besitzen unterschiedliche Matrixstrukturen, passen aber durchweg in die 8 x 16-Anordnung (8 Zeilen, 16 Spalten). Diese Größe läßt sich an den JU+TE-Computer anschließen, indem wie bisher mit einem MH 74154 oder zwei DS 8205 D aus den Adreßbits A0 bis A3 die 16 Spalten-Auswahlsignale gewonnen und der U 40098 D durch einen invertierenden 8-bit-Busverstärker (DL 540 D, DS 8283 D, DS 8287 D) ersetzt werden. Der U 40098 D kann auf der Leiterplatte bleiben, wenn beide Freigabe-Eingänge (Pins 1 und 15) 5P-Pegel erhalten. Er verbraucht dann praktisch keinen Strom. Das vom DS 8205 D der Prozessorplatine kommende Auswahlsignal /CS3 wird statt dessen als Freigabe dem Busverstärker

angeboten, der die Zeilenleitungen der Tastatur auf den Datenbus (D0 bis D7) des JU+TE-Computers durchschaltet. Eine weitere Möglichkeit besteht darin, den bereits vorhandenen U 40098 D mit einem weiteren auf D0 bis D3 arbeitenden zu ergänzen. Nicht vergessen werden dürfen die insgesamt acht pull-up-Widerstände, die auf den Zeilenleitungen den Vorzugspegel 1 erzeugen.

Das im 4K-Betriebssystem enthaltene Tastatur-Abfrage-Programm und die zugehörige ASCII-Tabelle reichen für diese Matrix nicht aus. Als Raum für die Erweiterung bietet sich der Adreßbereich ab %1800 an. Das ist die erste freie Speicherzelle hinter dem 4K-System in der kompakten Variante. Als EPROM eignet sich ein 2764, der allgemein unter 10 DM, vereinzelt auch unter 6 DM gehandelt wird. Jeder andere freie Adreßbereich ist für die zusätzliche Software ebenso möglich.

Die erhältlichen Tastaturen enthalten nicht nur Tasten mit ein oder zwei Belegungen, sondern oft auch einige dreifach benutzte. Deshalb sind drei Codetabellen erforderlich, um alle Umschaltebenen zu erfassen. Neben SHIFT wird eine weitere Umschalttaste benötigt. Hierfür eignen sich ALT (alternate: umschalten) oder FCTN (function). Außerdem nutzt unser Programmbeispiel die Möglichkeit, mit einer CTRL-Taste (control: Steuerzeichen) die Codes %00 bis %1F zu erzeugen. Das Abfragen von SHIFT und ALT (oder FCTN) erfolgt außerdem unabhängig von der übrigen Abtastung, damit diese Umschaltun-

gen beliebig vor oder während einer Tasteneingabe aktiviert werden können. Von der Hardware erfordert die Erkennbarkeit zweier gleichzeitig betätigter Tasten allerdings Entkopplungsdioden in allen Spaltenleitungen (Katode zum Dekoder). Den SHIFT-Balken (Inhalt der Speicherzelle %FFFF) teilt unser Programm in zwei Hälften: links ALT und rechts SHIFT. Ein Bit steht jeweils für den Betätigungszustand und drei rechts davon für den Anmeldezustand der jeweiligen Umschaltung. Schon wenn eines dieser Bits 0-Pegel besitzt, wird die Umschaltfunktion aktiv. Das Programm speichert diese Informationen stets im Arbeitsregister r2. Die abgebildete Befehlsfolge beginnt mit dem Holen dieser Merkwerte und dem Abfragen der beiden Umschalttasten. Bei Nicht- und bei Neubetätigung wird das Merkbyte hier aktualisiert. Ab TA4 folgt die allgemeine Abfrageschleife. Falls keine Taste gefunden wird, löscht das Programm das T-Bit im Übergaberegister %6D, speichert ab END das Merkbyte wieder in den Bildwiederholtspeicher und springt zurück. Andernfalls erhält das Arbeitsregister r1 ab TA5 die Positionskennung der gefundenen Taste. Dabei steht in den Bits D3 bis D0 die Spalten-Nummer und in D6 bis D4 die Zeilen-Nummer. Eventuell erkannte Sondertasten (ALT, CTRL und SHIFT) werden ignoriert (T10). Für alle anderen Tasten bildet die in r1 stehende Positionskennung den niederen Teil und die in r0 geladene Konstante den höheren Teil der ersten ASCII-Tabellen-Adresse. Ist SHIFT aktiv, wird diese Adresse um %80, ist ALT aktiv

# JU+TE Computerclub

um %100 erhöht. Bei Erkennen beider Umschaltfunktionen entnimmt das Programm den höheren Adreßteil aus dem Register %73. Das gestattet das Anlegen und Verwenden flexibler Codetabellen. An der Marke T12 erhält r3 den Zeichencode der betätigten Taste (ASCII) aus der mit SHIFT und ALT ausgewählten Tabelle. Ist CTRL aktiv, werden daraus die Bits D5 und D6 gelöscht, so daß eine Zahl zwischen %00 und %1F entsteht.

Für die Anwendung dieses Programms muß zunächst die Zuordnung der einzelnen Tasten zu den elektrischen Kreuzungspunkten zwischen Spalten- und Zeilenleitungen analysiert werden. Empfehlenswert ist das Anlegen einer

## Dynamische Tastenabfrage

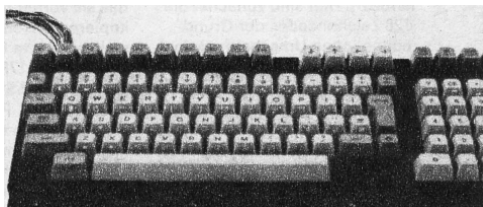
```
*TAS:70 FD      Registerpointer
E802 31 60      Registersatz 6
E804 0C FF      Adr Merkwzelle H
E806 1C FF      Adr Merkwzelle L
E808 82 20      r2 := Merkwzelle
E80A 0C 7F      Adr Tastatur H
E80C 1C 08      Adr ALT-Taste
E80E 82 30      Tasten-Eingabe
E810 76 E3 01   ALT betätigt ?
E813 6B :TA1    wenn nicht
E815 76 E2 80   neu betätigt ?
E818 6B :TA2    wenn nicht
E81A B6 E2 F0   Merkbits ändern
E81D 8B :TA2
*TA1:46 E2 80   Betätigung aus
*TA2:1C 01      Adr SHIFT-Taste
E824 82 30      Tasten-Eingabe
E826 76 E3 10   SHIFT betätigt?
E829 6B :TA3    wenn nicht
E82B 76 E2 08   neu betätigt ?
E82E 6B :TA4    wenn nicht
E830 B6 E2 0F   Merkbits ändern
E833 8B :TA4
*TA3:46 E2 08   Betätigung aus
*TA4:1C 10      Adr Tastatur L
*TA6:82 30      Tasten-Eingabe
E83C 60 E3      Bits 0-aktiv
E83E 66 E3 FF   Taste erkannt ?
E841 EB :TA5    wenn ja
*TA9:1A :TA6    nächste Spalte
E845 56 6D 7F   T-Bit := 0
*END:0C FF      Adr Merkwzelle H
E84A 1C FF      Adr Merkwzelle L
E84C 92 20      Merkbits
```



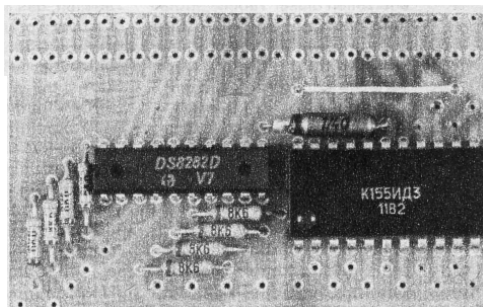
JU +7E-Computer mit 8 x 10-Tastenmatrix



6 x 8-Tastenmatrix



6 x 16-Tastenmatrix



Ansicht einer Tastatur-Interface-Platine mit Dekoder, Bus-Verstärker und Pull-up-Widerständen

# JU+TE Computerclub

```
E84E 50 FD      Registerpointer
E850 AF
*TA5:46 E1 F0   Spalte + %F0
*TA7:06 E1 10   Spalte + %10
E857 E0 E3      Zeile gefunden?
E859 7B :TA7     wenn nicht
E85B A6 E1 08   ALT-Taste ?
E85E EB :TAB     wenn nicht
*TI0:A6 E3 7F   sonst fündig ?
E863 EB :TA7     wenn ja
E865 56 E1 0F   nur Spaltenbits
E868 8B :TA9
*TAB:A6 E1 41   CTRL-Taste ?
E86D 6B :TI0     wenn ja
E86F A6 E1 01   SHIFT-Taste ?
E872 6B :TI0     wenn ja
E874 0C 18      Adr Tabelle H
E876 A6 6D 00   Übergabe frei ?
E879 EB :END     wenn nicht
E87B 66 E2 0F   SHIFT aktiv ?
E87E 6B :TI1     wenn nicht
E880 46 E2 07   quittieren
E883 06 E1 80   2. Tabelle
*T11:66 E2 F0   ALT aktiv ?
E889 6B :TI2     wenn nicht
E88B 46 E2 70   ALT quittieren
E88E 0E         3. Tabelle
E88F 42 11      SHIFT aktiv ?
E891 DB :TI2     wenn nicht
E893 08 73      Adr H aus R%73
*T12:82 30      ASCII aus Tab.
E897 46 E3 80   T-bit setzen
E89A 0C 7F      Adr Tastatur H
E89C 1C 01      Adr CTRL-Taste
E89E 82 10      Tasteneingabe
E8A0 76 E1 01   CTRL aktiv ?
E8A3 6B :TI3     wenn nicht
E8A5 56 E3 9F   Code kürzen
•T13:39 6D      Übergabe
EBAA 8B :END
```

entsprechenden Tabelle, in der z. B. anhand der Veröffentlichungen in JU+TE 2/88 oder in der TINY-Broschüre die zugeordneten ASCII erscheinen, für jede Ebene. Die Tabellen beginnen im Speicher des JU+TE-Computers auf einer beliebigen Adresse mit leerem niederen Teil (z. B. %1800). Hier sind zunächst die 128 Zeichencodes der Grundebene (ohne Umschaltfunktion) einzutragen. Das beginnt mit der mit D0 verbundenen Zeile und zwar zuerst Spalte 0, dann Spalte 15 (%F), 14 (%E), ... 2 und 1. Die Spaltennummer ergibt sich aus der Beschattung des Dekoders als mit A0 bis A3 codierter Adresse für den jeweils aktiven

Ausgang. Die nächsten 16 Speicherzellen dieser Tabelle enthalten in gleicher Folge die ASCII der mit D1 verbundenen Tastenzeile usw.

An diese 128 Codes schließen sich die mit SHIFT aktivierten Deutungen an. Sie sind in der gleichen Reihenfolge in unserem Beispiel ab %1880 einzutragen. Ein notwendiger Zusammenhang zwischen der ASCII von Grund- und Umschaltebene wie beim bisherigen Tastenanschluß fällt damit weg. Als dritte Tabelle folgen (ab %1900) die Zeichen der zweiten Umschaltfunktion (ALT bzw. FCTN). Daran kann sich dann das Unterprogramm TAS anschließen (ab %1980).

Im Programm müssen die Positionen der SHIFT-Taste (Spaltenadresse im Befehl auf TA2, Zeile auf %E826, Positionscode auf TA8) der ALT-Taste (Spalte: %E80C, Zeile: %E810, Positionscode: %E85B) und der CTRL-Taste (Spalte: %E89C, Zeile: %E8A0, Positionscode: %E86F) der tatsächlichen Anordnung in der elektrischen Matrix angepaßt werden. Die Tabellenanfangsadresse läßt sich im Befehl auf Adresse %E874 ändern. Das Programm enthält nur Relativsprünge und ist daher nach dem Binden unverändert auf beliebigen Adressen lauffähig. Nicht benötigte Tabellenplätze können jeden Inhalt haben, er bleibt gleichgültig. Beim Programmieren des neuen EPROM wird der Inhalt des alten von EMR-Adresse %0800 bis %0C55 übernommen. Dann folgen drei Byte zum Verzweigen auf das neue Programm (z. B. 8D 19 80), die sich irgendwo aus einem freien Stückchen RAM, in das sie

vorher geladen wurden, kopieren lassen. Es schließt sich weiter der alte EPROM-Inhalt von %0C59 bis %617FF an. Dann folgen die drei ASCII-Tabellen und das neue TAS-Programm wieder als Kopien aus dem RAM. Dabei kann man bei mangelnder Konzentration allerhand falsch machen, so daß wir uns hierfür Zeit nehmen und sehr besonnen handeln sollten. Also zuerst die drei Tabellen, das Programm und die drei Byte Sprungbefehle in den RAM eingeben, das Programm binden, den EPROM-Schnellbrenner laden und dann erst den EPROM brennen! Das Programm funktioniert bereits in verschiedenen Rechnern mit unterschiedlichen Tastaturen zur vollen Zufriedenheit und verbessert die Bedienfreundlichkeit deutlich. Viel Erfolg beim Anschluß und viel Freude beim Einsatz von Fertigastaturen am JU+TE-Computer!

*Dr. Helmut Hoyer*

## Höher auflösende Bildschirmsteuerung

Nach Veröffentlichung des für die neue Bildschirmsteuerung des JU+TE-Computers nötigen Bauelementeaufwands wurde noch eine kleine Veränderung der Logik unumgänglich. Daraus ergeben sich Abweichungen zu der auf Seite 73 (JU+TE 1/90) veröffentlichten Schaltkreisliste. Außer dem 40poligen Einchip-Mikrorechner (z. B. UB 8810 D) und einem Speichermodul mit 8 Kbyte RAM enthält die höher auflösende Bildschirmsteuerung bei minimalem

# JU+TE Computerclub

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

Wir bitten um Verständnis für diese kleine Veränderung, die bei frühzeitiger Veröffentlichung nun mal riskiert wird. Dafür klappt, so hoffen wir, die Bauelementbeschaffung bei Euch besser, als wenn nichts vorab gedruckt würde. Hard- und Softwarebeschreibung der höher auflösenden Bildschirmsteuerung veröffentlichen wir in unserem Juniheft 6/1990.

## Erfahrungen mit dem EPROMMER

Mit dem in JU+TE 10/88 vorgestellten EPROM-Programmierzusatz und dem im Heft 8/89 veröffentlichten Steuerprogramm (Schnellbrenner) wurden bereits mehrere hundert Schaltkreise behandelt. Dabei hat sich beides gut bewährt. Aus den gewonnenen Erfahrungen ergeben sich einige Hinweise, die wir Euch (um unnötigen Schaden zu vermeiden) nicht vorenthalten wollen. Sehr gefährlich ist falsches Stecken. Je höher der Integrationsgrad, desto sicherer führt das zum „Tod“ des EPROM. Ein defekter Schaltkreis (U2716 oder U2732) kann

zur Zerstörung des Vpp-Schalttransistors (SC 307) führen. Ein kräftigerer Typ (z. B. SF 816) übersteht derartige Belastungen besser. Beim Programmieren der Typen 2764 und 27128 darf die Programmiervoltage erst angelegt werden, wenn der Schaltkreis sicher steckt, und ist abzuschalten, bevor er wieder entnommen wird. Ein undefinierter Pegel am Steuereingang /PGM (während des Steckens sehr wahrscheinlich) löst unkontrolliertes Programmieren aus, das bei entsprechender Zeitspanne ebenfalls zum Verlust des Schaltkreises führt. Zum Lesen und Testen wird Vpp nicht benötigt. Hierfür reichen 5 V am Programmiervoltageeingang sicher aus. Eine Diode (z. B. SAY 30) von 5P (Anode) zu Vpp (Kathode) gestattet diese Betriebsarten bei abgeschalteter Programmiervoltage, ohne das Programmieren zu behindern. Sie kann daher fest eingelötet werden. Beim EPROM TMS 2764 (Texas Instruments) gab es mehrfach Verluste, da die eigentlich typische Programmiervoltage von 21 V nicht vertraglich wurde. Mit 19 V dagegen ließen sich sowohl TMS 2764 als auch TMS 27128 problemlos behandeln. Die sonst geltenden Werte enthält die folgende Tabelle.

EPROM-Typ	Vpp
2716	25,0 V
2732	25,0 V
2732A	21,0 V
2764	21,0 V
2764A	12,5 V
27128	21,0 V
27128A	12,5 V

## JU+TE-Computer als Taschenrechner

Mit einem sehr einfachen Maschinenprogramm (nur 25 byte) läßt sich der Formelinterpreter des UB 8830 D (Startadresse: %06E0) zum Lösen einfacher Rechenaufgaben nutzen. Sie werden einfach in die erste Bildschirmzeile eingegeben und mit ENTER bestätigt. Das Ergebnis erscheint in der zweiten Zeile. Eine weitere Tastenbetätigung startet das Programm von vorn.

Albrecht Klein

*CAL:31 10	Registerpointer
*CA2:D6 08 DD	Bild löschen
*CAL:D6 08 24	Tasteneingabe
E008 A6 5A 0D	ENTER ?
E00B EB :CA1	wenn nicht
E00D 0C FD	AdreSpPointer H
E00F 1C 00	AdreSpPointer L
E011 D6 06 E0	Interpreter
E014 D6 08 24	Taste abwarten
E017 8B :CA2	nächste Aufgabe

## Sieb des Eratosthenes

Nach dem Verfahren des griechischen Mathematikers ERATOSTHENES von Kyrene (ca. 275 bis 194 v. u. Z.), nachzulesen in der Kleinen Enzyklopädie Mathematik, ist dieses Programm zur Berechnung von Primzahlen gestaltet. Die Anzahl der möglichen Primzahlen richtet sich nach dem jeweils vorhandenen bzw. dafür festgesetzten Speicherbereich. Wird z. B. S = %E400 und E = %E4FF eingegeben, stehen dem Sieb des ERATOSTHENES 255 Speicherzellen zur Verfügung. In 37 Sekunden werden dafür die 54 Primzahlen

# JU+TE Computerclub

von 2 bis 251 berechnet. Für 24 Kbyte braucht der JU-GEND+TECHNIK-Computer schon eine Stunde und gibt dabei 2725 Primzahlen aus.

*Siegfried Tscheuschner*

```
10  CALL %8DD;
    PRINT "PRIMZAHLEN N"
20  PRINT "ERATOSTHENES"
30  PRINT "VON KYRENE".
40  PRINT "275-194 V.CH."
50  PRINT "RAM-ANFANG"
60  PROC SETR[%6D,%25];
    INPUT "S:"S
70  PRINT "RAM-ENDE"
80  PROC SETR[%6D,%25];
    INPUT "E:"E
90  PROC SETR[%6D,0];
    PROC SETRR[%F2,%14FF]
100 CALL %8DD
110 PRINT "NULL - SETZEN";
    PRINT " "
120 PRINT "BITTE WARTEN!"
130 LET A=S,I=0
140 PROC SETEW[A,0]
150 LET A=A+2,I=I+2
160 IF A<E
    THEN GOTO 140
200 CALL %8DD
210 PRINT "PRIMZAHLEN";
    PRINT "BIS" I
220 LET D=S+1,P=1,N=0
230 LET D=D+1,P=P+1
240 LET T=GETEB[D]
280 IF T=1
    THEN GOTO 230
300 LET N=N+1
310 PROC SETR[%F1,%8A]
320 PRINT N,"="P
330 PROC SETR[%F1,%A]
340 LET A=P*P+S
350 IF A>E
    THEN GOTO 400
360 PROC SETEB[A,1];
    LET A=A+P
370 IF A>E
    THEN GOTO 230
380 GOTO 360
400 LET D=D+1,P=P+1
410 IF D>E THEN END
420 ELSE ;
    LET T=GETEB[D]
430 IF T=1
    THEN GOTO 400
440 LET N=N+1
450 PROC SETR[%F1,%8A]
460 PRINT N,"="P
470 PROC SETR[%F1,%A]
480 GOTO 400
```

# JU+TE Computerclub

## Die Krönung:

### ***Vollgrafik und Farbe***

In die Tausende geht die Zahl der Leser, die sich seit 1987 zum Nachbau des JU+TE-Computers entschlossen haben. Und noch reißt die Kette der Leiterplattenbestellungen nicht ab. Selbstbauen macht Spaß, meinen die Freaks. Das stimmt! Auch jetzt noch ist ein Neueinstieg in das JU+TE-Computersystem möglich. Man beschaffe sich dafür nur "TINY - der kleine Selbstbaucomputer". erschienen im Verlag Junge Welt, erhältlich im Buchhandel. Oder man schlage in den „JU+TE“-Ausgaben ab Heft 7/87 (evtl. Bibliothek!) nach. Den Wünschen vieler Leser nachkommend, setzen wir im folgenden mit der Vorstellung der letzten Ausbaustufe das Pünktchen aufs i: Vollgrafik und Farbtüchtigkeit. Die Computerclubseiten der nächsten Ausgaben werden mit Hard- und Softwaretips gefüllt, von denen alle Nutzer etwas haben sollen, ganz gleich, ob 64er, Atari-Fans oder KC 85er. Die Tip-Lieferanten sollt Ihr selbst sein. Für den besten, den „Tip des Monats“ berappen wir einen kleinen Preis. Doch zurück zum JU+TE-Computer.

### **Die reifere Lösung**

Das Grundkonzept geht davon aus, die Berechnung von der ständigen Ausgabe des Bildes zu trennen. Das befreit den Hauptprozessor (UB 8830 D) von 90 Prozent Rechenleistung, die der Bildinterrupt-Service-Routine zugute kommt. Zudem führt das zu einer opti-

malen Auslegung der Videosteuerung. Ganz zu schweigen davon, daß an der modularen Grundkonzeption kein Deut geändert wird. Der stufenweise Ausbau auf Grundlage des schon Vorhandenen bleibt also möglich.

### **Vier verschiedene Ausbaustufen**

Mit einem 8-Kbyte-RAM-Modul entsteht ein monochromes Bild von 192 Zeilen á 320 Punkten. Dieses Feld gleicht dem der KC-Typen. Bei 8 x 8 Punkten pro Zeichen bereitet das Darstellen von 24 Zeilen mit je 40 Buchstaben, Ziffern oder Sonderzeichen keine Probleme. Ein zweiter 8-Kbyte-RAM-Modul und ein Multiplexer DL 253 D erhöhen die Zeilenauflösung auf 640 Punkte. So gelangen wir zum mit 80 Zeichen je Zeile bei Personal- und Bürocomputern üblichen Format. Die dritte Möglichkeit benötigt insgesamt vier 8-Kbyte-RAM-Baugruppen. Damit steht vor uns ein KC-Bildschirm (192 x 320), vollgrafisch mit 16 Farben je Bildpunkt.

Mit der vierten Variante streben wir PC-Format (192 x 640) mit acht Farben an, für je zwei benachbarte Punkte gemeinsam, ebenfalls bei voller RAM-Bestückung, versteht sich. Für die Speicher-Aufrüstung nutzen wir gleich die für die JU+TE-Prozessorplatine zugeschnittenen Modulplatinen (senkrecht aufsetzen!).

Hohe Nachbausicherheit bei geringen Kosten gewährleistet der CMOS-RAM-Schaltkreis HM 6264 (vgl. JU+TE 1/88 oder TINY-Broschüre, Modulvariante 3). Die Videoleiterplatte, genaue Bezeichnung: JU+TE-

Videomodul, hat die Größe der Prozessorplatine, enthält jedoch mehr Bauelemente.

### **Aufbau auf Vorhandenem**

Der Schaltungsentwurf baut auf dem JU+TE-Computer auf. Dort belegt der Videomodul den Platz 3 mit /CS2 für den Bildwiederhol-speicher (Adressen %4000 bis %5FFF) und Tastaturanschluß (%7C00 bis %7FFF). Neben diesen beiden Freigabesignalen, abgeleitet aus den Adreßbits A13, A14 und A15 sowie dem Steuersignal /DS (dato strobe), werden benötigt: Der Datenbus (D0 bis D7), der Adreßbus (A0 bis A12), das Rücksetzsignal /RT (als Sonderspannung U2 vom Bus übertragen) und die Schreibfreigabe R/W des Hauptprozessors. Diese Signale gewinnen wir am besten vom Platz 4 über möglichst kurze Schaltdrähte. Das Taktsignal XTAL (8 MHz) führen wir über eine verdrehte Leitung zu (Masse und Takt) oder erzeugen es mit einem 8-MHz-Quarz (zwischen den Pins 1 und 12 des DL 004 D) und einem 470-Ohm-Widerstand (zwischen 1 und 2) auf dem Videomodul separat. Daran passen auch andere Bussysteme. Allenfalls kostet es uns einen geringen Zusatzaufwand, /CS2 und /CS3 zu erzeugen. Die Last an /CS2 besteht aus sechs Low-Power-TTL-Eingängen. Der Bildwiederhol-speicher gehört zum RAM-Bereich des Hauptprozessors. Der Adreßraum gliedert sich wie folgt:  
%8000-%FFFF: RAM  
%7C00-%7FFF: Tastatur  
%6400-%78FF: frei  
%6000-%63FF: Video-Steuerung





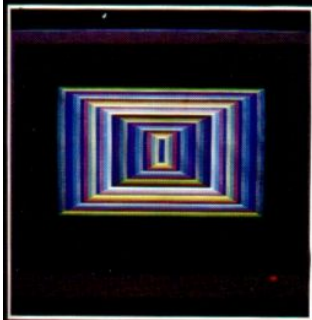
Variante mit neuer Prozessorplatine, ohne Gehäuse



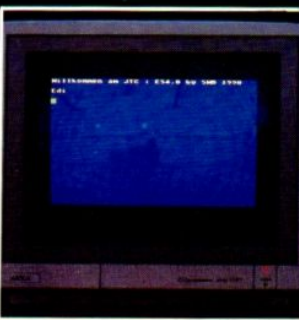
Nachgerüstete Variante



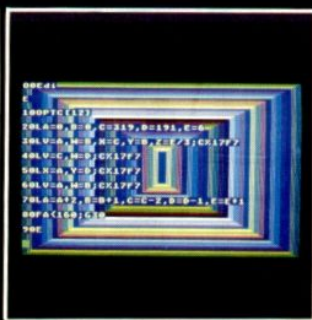
Teilbestücktes Videomodul



Farbmusterbeispiel



Begrüßungszeile des Betriebssystems



Programm mit unterlegtem Farbmuster



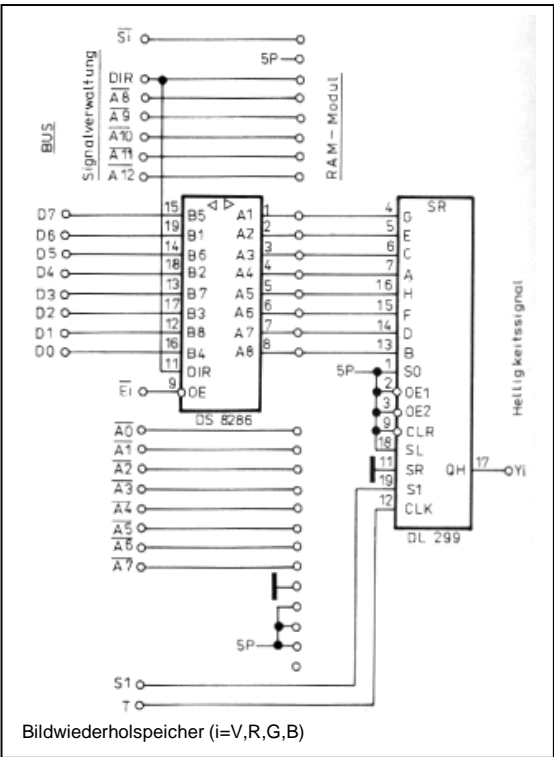
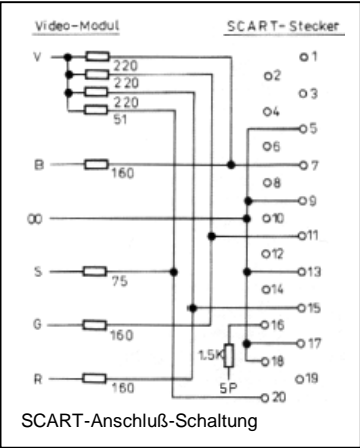
Hex-Dump in Farbe

%4000-%5FFF: Video-RAM  
 %2000-%3FFF: EPROM 2  
 %0800-%1FFF: EPROM 1  
 %0000-%07FF: interner ROM.  
 Für den freien Bereich liefert der Videomodul mit dem DS

8205 D Auswahlssignale (selects) für jeweils ein Kilobyte, die vorzugsweise für Ein-/Ausgabe-Erweiterungen zur Verfügung stehen. Der Video-RAM (Bildwiederholpeicher) besteht aus maximal vier Teilen mit je einem Busverstärker DS 8286 D, einem 8-Kbyte-RAM-Modul und einem Schieberegister DL 299 D (Abb. S. 84 oben), die den gleichen Adreßbereich belegen. Der DS 8286 D verbindet die acht Datenbits (aus topologischen Gründen etwas durcheinander) mit dem Datenbus D0 bis D7. So greift der Hauptprozessor auf den Inhalt des Video-RAMs zu. Die Daten für die Bilderzeugung übernimmt der DL 299 D vom Vi-

deo-RAM und wandelt sie in acht aufeinanderfolgende Bildpunkte (Helligkeitssignal Y) je Byte um. Dabei arbeiten die vier Teile parallel, so daß insgesamt der Bilderzeugung (Vorder-/Hintergrund, rot, grün, blau) vier Signale (YV, YR, YG und YB) zur Verfügung stehen. Die Schiebepulsfrequenz T von 8 MHz bedeutet eine Bildpunkt-länge von 125 ns. Die Ausgabe der 320 Punkte je Zeile braucht 40µ, acht Bildpunkte je µ. Die den Bildpunkten zugeordneten Bits befinden sich in Gruppen zu je drei Zeilen im Video-RAM. Ab der Anfangsadresse (%4000) füllen die obersten drei Zeilen von links nach rechts (Bit 7 jeweils zuerst) die

ersten 120 Speicherzellen. Dem folgen acht ungenutzte Zellen und darauf die 120 der vierten bis sechsten Punktzeile. Insgesamt finden im Video-RAM-Bereich 64 solcher Gruppen Platz. Das macht  $64 \times 3 = 192$  Punktzeilen und  $64 \times 128 = 8192$  Speicherzellen. Es leuchtet ein, daß für die Bildausgabe eine spezielle Adreßfolge erzeugt werden und andererseits der Hauptprozessor den Video-RAM mit seinen eigenen Adressen erreichen können muß. Dazu mußte eine Adreß- und Steuersignalverwaltung her, die je nach Bedarf zwischen diesen beiden Fällen umschaltet. Der Hauptprozessor genießt dabei Vorrecht, Solange  $/CS2 = 0$  gilt, werden seine Adressen A0 bis A12 vom Bus über zwei Tristate-Treiber DL 540 D invertiert als  $/A0$  bis  $/A12$  zum Video-RAM durchgeschaltet. Andernfalls bestimmt der Videoprozessor (UB 8810 D) über Port 2, den DS 8212 D und zwei weitere DL 540 D diese Adressen. Ein ausgefeuchstes Programm im EPROM U 2716 C sichert dabei nicht



# JU+TE Computerclub

## Video-EPROM

```
E000: FF FF FF FF FF FF FF FF
E008: FF FF FF FF FF FF FF FF
E010: FF FF 8F 31 F0 8C B2 7C
E018: 09 6C C0 1C 00 FF FF FF
E020: 31 00 FF FF FF FF FF FF
E0F8: FF FF A2 A2 3C F0 A2 A2
E100: A2 A2 A2 A2 A2 A2 A2 A2
E108: A2 A2 A2 A2 A2 A2 A2 A2
E110: A2 A2 A2 A2 A2 A2 A2 A2
E118: A2 A2 A2 A2 A2 A2 A2 A2
E120: A2 A2 A2 A2 3C 70 A2 A2
E128: A2 A2 A2 A2 3C 50 A2 A2
E130: 3C 70 A2 A2 A2 A2 80 DA
E138: 22 FF FF FF FF FF FF FF
```

```
E220: FF FF A2 A2 3C F0 A2 A2
E228: A2 A2 A2 A2 A2 A2 A2 A2
E230: A2 A2 A2 A2 A2 A2 A2 A2
E238: A2 A2 A2 A2 A2 A2 A2 A2
E240: A2 A2 A2 A2 A2 A2 A2 A2
E248: A2 A2 A2 A2 3C 70 A2 A2
E250: A2 A2 A2 A2 3C 50 A2 A2
E258: 3C 70 A2 A2 A2 A2 8D DB
E260: 4A FF FF FF FF FF FF FF
```

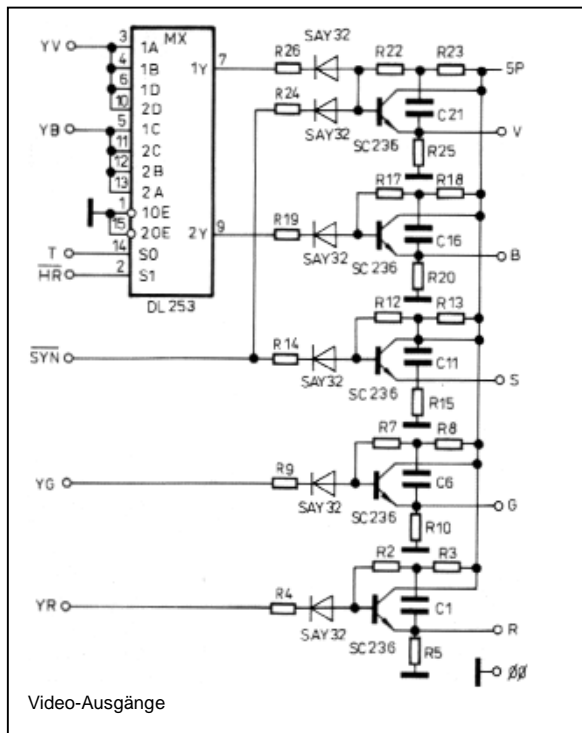
```
E348: FF FF A2 A2 3C F0 A2 A2
E350: A2 A2 A2 A2 A2 A2 A2 A2
E358: A2 A2 A2 A2 A2 A2 A2 A2
E360: A2 A2 A2 A2 A2 A2 A2 A2
E368: A2 A2 A2 A2 A2 A2 A2 A2
E370: A2 A2 A2 A2 3C 70 A2 A2
E378: A2 A2 A2 A2 3C 50 EC 37
E380: 3C 70 20 02 00 0D 42 DD
E388: ED D8 FC 8B 00 8B 00 8B
E390: 0A A2 A2 A2 A2 A2 A2 A2
E398: A2 A2 A2 3C 30 A2 A2 A2
E3A0: A2 A2 A2 A2 A2 A2 A2 A2
E3A8: A2 A2 A2 A2 A2 A2 A2 A2
E3B0: A2 A2 A2 A2 A2 A2 A2 A2
E3B8: A2 A2 A2 A2 A2 A2 A2 A2
E3C0: A2 3C 10 A2 A2 3C 30 FC
E3C8: 0A 00 0E 42 EE ED DB 91
E3D0: 8B 00 8B 00 8B 0A A2 A2
E3D8: A2 A2 A2 A2 A2 A2 A2 A2
E3E0: A2 A2 A2 A2 A2 A2 A2 A2
E3E8: A2 A2 A2 A2 A2 A2 A2 A2
E3F0: A2 A2 A2 A2 A2 A2 A2 A2
E3F8: A2 A2 A2 A2 A2 A2 A2 A2
E400: A2 A2 3C 30 A2 A2 3C 10
E408: A2 A2 3C 10 EC 30 00 0F
E410: 42 FF ED DB D6 8B 00 8B
E418: 00 8B 0A A2 A2 A2 A2 A2
E420: A2 A2 A2 A2 A2 A2 A2 A2
E428: A2 A2 A2 A2 A2 A2 A2 A2
E430: A2 A2 A2 A2 A2 A2 A2 A2
E438: A2 A2 A2 4C 04 A2 E4 4C
E440: 00 C0 04 C0 04 42 34 5C
E448: D0 6C 20 52 35 2C 00 42
```

Das eröffnet die Möglichkeit, zwischen den vier Video-RAM-Teilen auszuwählen, wobei mehrere gleichzeitig beschrieben, aber stets nur einzeln

gelesen werden dürfen. Das Steuerregister muß der Hauptprozessor auf diese Weise stets vor dem eigentlichen Zugriff auf den Video-RAM geladen haben. Die Signale /ER, /EG, /EB und /EV steuern die Busverstärker DS 8286 D (vgl. Abb. S. 84 oben). Neben dem 8-MHz-Schiebetakt T stellt die Adreß- und Steuersignalverwaltung noch das Auswahlsignal /KEY bereit, das beim JU+TE-Computer statt bisher der /CS3 für die Steuerung der Tasteneingabe zu nutzen ist. Das Rücksetzsignal RT (JU+TE-Computerbus: U2) läuft zum /RESET-Eingang

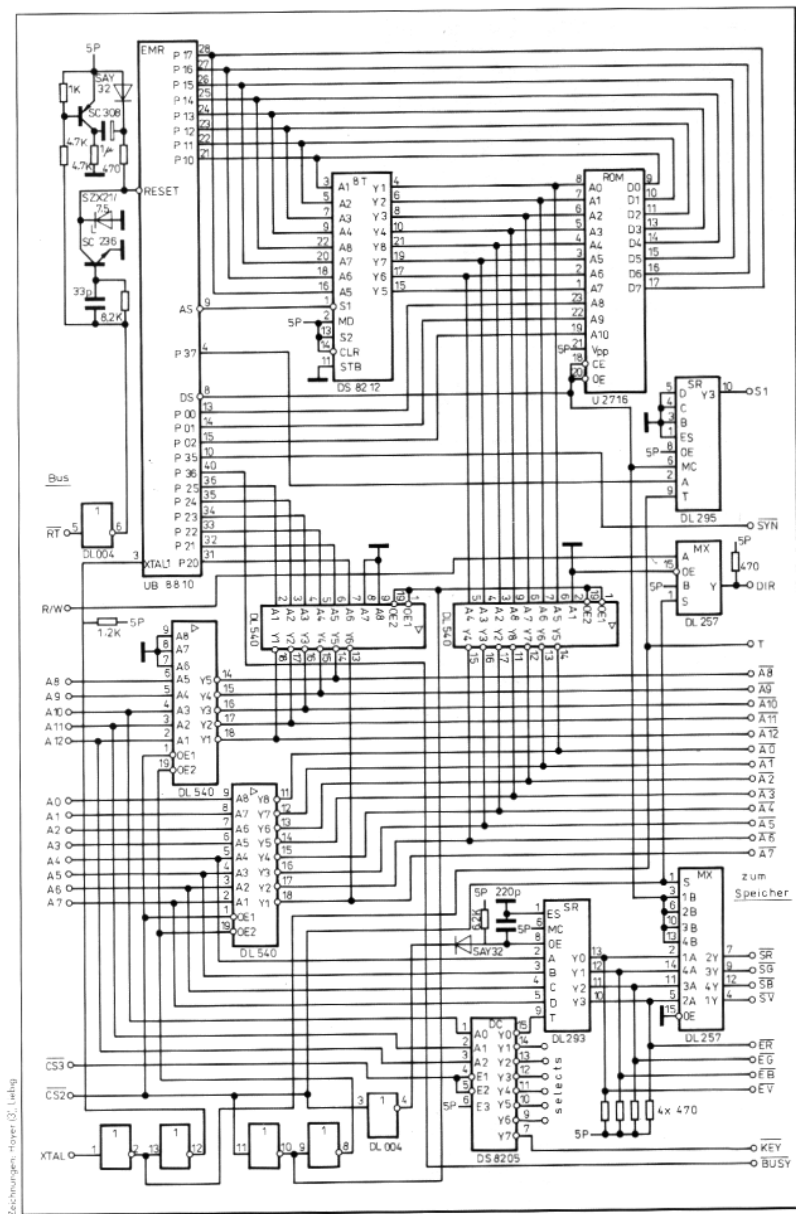
des Videoprozessors über eine Schaltung, die beim 0-1-Übergang den Pegel auf 7,5 V hebt. Deshalb können wir die Typen UB 8810 D, UB 8830 D, UB 8860 D, U 8611 DC und U 8611 DC-1 als Videoprozessor verwenden. Der Pegelhubb aktiviert den Test-ROM, so dass der Inhalt des internen ROM dieser EMR-Schaltkreise völlig gleichgültig ist. Sollen die Typen UB 8811 D, UB 8831 D, UB 8861 D, UL 8611 DC und UL 8611 DC-1 eingesetzt werden, erfordert das zusätzlich eine Lötbrücke zwischen deren Pins 1 und 2.

Das D-R-C-Glied am Pin 4 des DL 004 D verzögert die



# JU+TE Computerclub

## Adreß- und Steuersignalverwaltung



# JU+TE Computerclub

Freigabe der Video-RAM bei Zugriffen des Hauptprozessors, um dort eventuell vorhandene Adreßlatches korrekt laden zu können. So eignen sich solche CE-aktivierten RAM-Schaltkreise, wenn der Zugriff nur während BUSY = 0 erfolgt. Der allgemein mögliche Zugriff erfordert die gleiche Schaltung vor Pin 1 des DL 257 D, der /SR bis /SV bereitstellt. Deren Bemessung ist allerdings sehr problematisch. Deshalb empfehlen wir grundsätzlich adreßaktivierte RAM (HM 6264 oder TC 5565) für den Video-Modul zu verwenden. Damit erübrigen sich solche Verzögerungsglieder.

## Aufbau

Die Leiterplatte (vgl. unseren Leiterplattenservice S. 87) bohren wir nach Entfernen des Ätzschutzlacks und lackieren sie dünn mit einer Kolophoniumlösung. Nach Einlöten der freien Durchkontakte setzen wir zuerst den DL 004 D ein. Für minimalen Ausbau genügen der DS 8286 D und der DL 299 D am RAM-Modul V. Die übrigen drei DS 8286 D, drei DL 299 D und der DL 253 D brauchen wir nur für höhere Ausbaustufen. Es muß lediglich der Durchkontakt statt Pin 11 des dem Modul B zugeordneten DS 8286 D gewährleistet werden. Selbst bei Minimalausbau bestücken wir alle Stützkondensatoren Cx. Sie ermöglichen viele Durchkontakte. Zwischen Prozessor und den DS 8212 D setzen wir eine 100-nF-Scheibe. Sonst tun es für die mit "+" gekennzeichneten Stützkondensatoren stehende Elkos (ca. 10µF) und für die restlichen zwölf 22-nF-Scheiben.

Die Gestaltung der Videoausgänge (C1 bis R26, zugehörige Dioden und Transistoren, DL 253 D, Abb. S. 84 Mitte) hängt von der jeweiligen Ausbaustufe und den Eigenschaften des verwendeten Sichtgerätes ab. Bei Minimalausbau sichern C21=33p, R22=1k, R23=1k, R24 = 2,2k, R25 = 220, R26 = 6,8k, die beiden zugehörigen Dioden und der Transistor ein für Videoeingänge (FBAS) geeignetes Signal V. Den DL 253 D ersetzt die im Bestückungsplan (vgl. Sonderdruckbeilage Fa. Gerlich) mit B bezeichnete Brücke zwischen den Lötäugen für dessen Pins 7 und 10. Auf alle übrigen in Abb. S. 84 dargestellten Bauelemente können wir für die Minimalvariante verzichten. Besitzt unser Sichtgerät TTL-Eingänge, brauchen wir keins der in Abb. S. 84 Mitte enthaltenen Bauteile. Die geeigneten Signale YV und /SYN sowie YR, YG und YB (bei Farbausstattung) liegen als LS-Ausgänge (bzw. nMOS-Ausgang /SYN) über Leiterzüge auf der Bestückungsseite bereit. Die zweite Ausbaustufe bedarf eines zweiten RAM-Moduls (B), der dort zugehörigen DS 8286 D und DL 299 D sowie des Multiplexers DL 253 D. Das Signal V ist je nach Belegung des HR-Eingangs (high resolution: hohe Auflösung) bei 1-Pegel wie bisher das Bild aus dein V-RAM oder bei 0-Pegel das Gemisch der Bilder aus V-RAM und B-RAM. Am Busanschluß liegt /HR an der Stelle, wo der JU+TE-Computer P34 bereitstellt. Damit raten wir, die Umschaltung programmtechnisch zu steuern. Sonst besteht die Möglichkeit, mit der im Bestückungsplan als HR

bezeichneten Brücke entweder 0- oder 1-Pegel festzulegen. Für die dritte Ausbaustufe brauchen wir alle im Bestückungsplan aufgeführten Bauelemente außer DS 253 D, R24 und der dort zugeordneten Diode. R4, R9, R14, R19 und R26 betragen 2,2k. Ansonsten bleibt es wie bei der Minimalconfiguration. Die Brücken B und V müssen sein. Aus den Signalen R, G, S, B und V lassen sich die für das Farbsichtgerät nötigen mischen. Abb. S. 84 links zeigt eine für den genormten SCART-Anschluß bemessene Schaltung. Sie wurde für einen Computer-Monitor entwickelt. Für andere Geräte ist sie unter Umständen zu variieren. Die vierte Ausbaustufe schließlich bringt die langersehnten Farbbilder für hochauflösend darstellende Schirme (640 Bildpunkte je Zeile), wieder mit DL 253 D. Das Blausignal wird wie die Helligkeit gesteuert. Rot und grün sind für je zwei benachbarte Bildpunkte gemeinsam frei belegbar. Bei HR = 1 entsteht die gleiche Ausgabe wie mit der dritten Ausbaustufe. Die RAM-Module setzen wir grundsätzlich zuletzt auf. Zuvor müssen wir sie jedoch an der Kontaktseite stark entgraten. Das schützt vor eventuellen Schlüssen zwischen Leiterzügen, die auf der Bestückungsseite der Videoplatten unter den RAM-Modulen durchführen. Selbstverständlich achten wir beim Löten auf sparsamen Zinneinsatz und Sauberkeit.

## Inbetriebnahme

Nach dem Aufbau der Minimalvariante und gründlicher Sichtkontrolle kann der Videomodul

zunächst ungekoppelt in Betrieb gehen. Voraussetzung ist ein gesteckter EPROM U 2716 C mit dem in Abb. S. 86 dargestellten Inhalt und ansonsten leeren (%FF) Speicherzellen, die Betriebsspannung und ein Takt. Das /RT-Signal wird kurz mit Masse (00) verbunden, damit der Videoprozessor das im EPROM gespeicherte Programm ausführt und keine Ausgangskonflikte entstehen. Die Stromaufnahme beträgt etwa 500 mA. Masse und Versorgungsspannung müssen an den betreffenden Schaltkreisanschlüssen zu messen sein. Für die weitere Inbetriebnahme brauchen wir ein Oszilloskop. Nach dem Rücknetzen muß die am Pin 6 anliegende Spannungsüberhöhung des Videoprozessors deutlich zu sehen sein. Der Pegel sinkt dann auf 5 V. Als nächstes prüfen wir, ob /AS am Pin 1 des DS 8212 D ankommt (ca. 1 MHz). Danach folgt der Schwingungstest auf den Ausgängen des DS 8212 D mit Frequenzverhältnissen benachbarter Signale von 1 : 2. Die niederfrequenten lassen die mit dem Zeilenwechsel verbundenen Unregelmäßigkeiten erkennen. Die Fortsetzung (ab A7) erfolgt über P20 bis P25, auch am DL 540 D zu verfolgen. Solange /CS2 offenliegt oder mit 1-Pegel beschaltet ist, müssen diese Schwingungen an den Adreßeingängen des RAM-Schaltkreises vorliegen. Bei /CS2 = 0 dagegen weichen sie konstantem 0-Pegel. Im ausgeschalteten Zustand kann das Sichtgerät angeschossen werden. Bei eingeschaltetem Videomodul muß es synchronisieren und irgendein sinnloses Punktmuster anzeigen.

gen. Erst wenn das klappt, kann man den Hauptprozessor (JU+TE-Computer) einbeziehen.

Umschaltung auf Videoadressen erfolgt dann so langsam, daß DIR mit Sicherheit wieder

## **Bewährter Leiterplatten-Service**

Die ungebohrte Leiterplatte des JU+TE-Computer-Videomoduls könnt Ihr wie immer bestellen bei Fa. Gerlich, Markscheiderweg 08/47, Neubrandenburg, 2000 und, unter gleicher Adresse, immer noch bestellbar sind:

- Rechnerplatine (Abb. 6/7 in JU+TE 8/87, S. 635 und 9/ 87, S. 696),
- Speicherplatine U 2716 (Abb. 10/11 in 9/87, S. 699),
- Magnetbandinterface 2 K (Abb. 19 in 12/87, S. 932),
- EPROM-Programmierzusatz (in 10/88, S. 786/787),
- Magnetbandinterface 4 K (in 3/89, S. 231),
- vereinfachte Speicherplatine U 2716 für 4 K (in 3/89, S. 232),
- Speicherplatine mit 8 K x 8 Bit-EPROM oder -RAM (in 1/88, S. 72),
- Batterie-Stützung für CMOS-RAM (in 4/88, S. 287),
- Speichermodul U 6516 D (in 6/88, S. 442/443)

und außerdem:

die Leiterplatte für die im nächsten Heft (7/90) erscheinende optimierte Variante des ausgebauten JU+TE-Computers.

## **Telefonservice**

"Jugend und Technik"-Autor und Entwickler des JU+TE-Computers Dr. Helmut Hoyer stellt sich Euren Fragen. Greift zum Hörer und wählt am 20. Juni 1990 zwischen 15.30 Uhr und 17.30 Uhr die Nummer: Berlin (Vorwahl) 2233/432 oder 433! Wir warten auf Eure Fragen.

passiven 1-Pegel führt.

## **Software**

Ausschöpfen kann man die Möglichkeiten dieser Erweiterung erst mit entsprechender Software. Grundlage dazu bildet ein neuentwickeltes 6-K-Betriebssystem (OS). Es setzt einen mit 8-K-EPROM (2764) und mit acht Kbyte RAM ausgerüsteten Speichermodul auf Platz 1 der Prozessorplatine als Minimum voraus. Wie im Heft 5/90 beschrieben, eignen sie alle Tastaturen bis zu einer Matrixgröße von 8x16. Ein Full-Screen-(Ganz-Bildschirm)Editor (FSE), der Eingaben und Korrekturen in allen Bildzeilen zuläßt, wertet die neue Software gehörig auf. Neben dem im Komfort etwas eingeschränkten TINY-BASIC-Editor gehört ein als Commando-Interpreter organisiertes Monitorprogramm zum 6-K-System. Standardangebote, die das vom DATA-Modus bekannte weit übersteigen, werden durch beliebige Anwenderprogramme,

hen. Wichtig: Möglichst kurze Drahtlängen, höchstens 5 cm! Beeinträchtigen dennoch störende Schwingungen den Datenbus (falsch belegte Bildpunkte oder gar Absturz des Hauptprozessors), können acht 150-Ohm-Widerstände statt der Verbindungsdrähte für D0 bis D7 dämpfen.

Bei schnellen Schaltkreisen kann mitunter ein Wettlaufproblem zu Störungen führen. Abhilfe schafft dieses: Wir unterbrechen die Leitung von Pin 11 des DL 004 D zu den Freigabeingängen der beiden DL 540 D für die Videoadressen mit einem 1,5-k-Widerstand und parallelgeschalteter Si-Diode (Anode zum DL 004 D). Die

## JU+TE Computerclub

wie erstere per Kommando (frei wählbar) aus dem Monitor zu starten, sinnvoll ergänzt. Grafik-Unterprogramme und die Steuerung einer Centronics-Schnittstelle fehlen im EPROM nicht. Alle Ein- und Ausgaben lassen sich auf beliebige Peripherie umlenken. Das System steht für Erweiterungen offen. An dieser Stelle sei noch einmal ausdrücklich daran erinnert: Auch jetzt lohnt es sich noch, den JU+TE-Computer von der Pike aufzubauen. Die Broschüre „TINY - der kleine Selbstbaucomputer“ bietet dafür eine gute Grundlage. Als Beilage zu den bei Fa. Gerlich bestellten Selbstbaucomputer-Leiterplatten erhaltet Ihr aber auch einen Sonderdruck (Preis 4,50 M), wenn Ihr ihn nicht ausdrücklich abbestellt, per Nachnahme zugeschickt. Er enthält die Beschreibung des neuen Betriebssystems und Hinweise zu dessen Handhabung sowie den Leiterplatten-Bestückungsplan. Das Betriebssystem bekommt Ihr, wenn Ihr an "Jugend und Technik", PF 43, Berlin, 1026 eine leere Kassette zum Bespielen oder einen gelöschten EPROM schickt. Bitte dann nicht Rückporto in Briefmarken (0,90 M) vergessen!

*Dr. Helmut Hoyer*

## Selbstbau-Computer kompakt

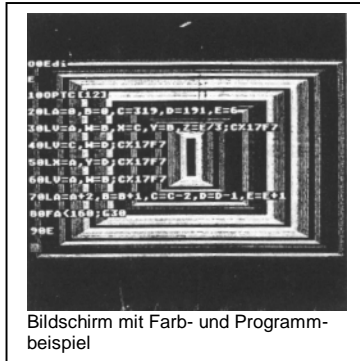
Beim Ausrüsten unseres Selbstbau-Computers mit dem in „jugend und technik“ Heft 6/90 beschriebenen höher auflösenden Videomodul wird die alte Bildschirmsteuerung entbehrlieh. Die Schaltkreise DL 030 D, DL 074 D und zweimal D 195 D auf der Prozessorplatine können entfallen. Daher wurde eine neue Prozessorplatine entwickelt, die sich sowohl dem Videomodul als auch dem aktuellen Angebot von Speicherschaltkreisen anpaßt.

### Konstruktion

Die Prozessorplatine enthält alle Baugruppen, die unser Computer benötigt, die Bildschirmsteuerung ausgenommen. Die Leiterplatte wurde so konstruiert, daß sie sich direkt an den Videomodul anpaßt. Das macht eine gemeinsame Platine (235 mm x 155 mm) für alles möglich. Damit sinkt der Verdrahtungsaufwand rapide. Bei Bedarf lassen sich der 155 mm x 135 mm große Videomodul und die 155 mm x 100 mm messende Prozessorplatine auch getrennt aufbauen. Den Bestückungsplan stellt Abb. 1 dar.

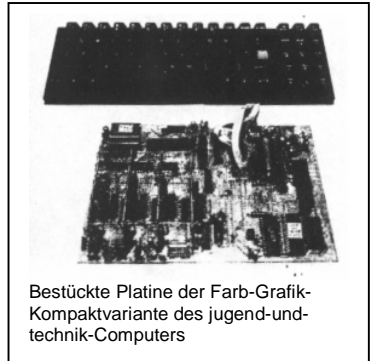
### Rechner-Kern

Die neue Prozessorplatine enthält neben dem EMR-



Bildschirm mit Farb- und Programmbeispiel

Schaltkreis UB 8830 D auch RAM (HM 62256) und EPROM (U 2764). Deshalb ist kein Speichermodul mehr nötig. Adreßdekode (DS 8205 D) und Adreßlatch (DS 8212 D) wurden wie Gleichrichter (1 PM 05) und Spannungsregler (B 3170) von der ursprünglichen Platine übernommen. Hinzu kommen die im Verlauf der drei Jahre unseres Computers ergänzten peripheren Baugruppen Kassettenschnittstelle, Batteriestützung und Tasteninterface bis zur Matrixgröße 16x8. Abb. 2 zeigt die Schaltung der neuen Prozessorplatine. Der 8-MHz-Taktgenerator (1/2 DL 000 D) versorgt den UB 8830 D und den Videomodul. Der Adreßdecode erzeugt mit der zweiten Hälfte des DL 000 D ein Auswahlsignal für die gesamte obere Adreßhälfte (%8000 bis %FFFF), die der HM 62256 belegt. Entsprechend haben darüber hinaus nur noch die Auswahlsignale /CS0 bis /CS3 Bedeutung. Der Vollständigkeit halber erhält der Modulplatz 2 auch noch wie früher /CS6. Dieses Signal wird nur dann gebraucht, wenn statt des HM 62256 ein HM 6264 bestückt ist, den ein zweiter auf Modul 2



Bestückte Platine der Farb-Grafik-Kompaktvariante des jugend-und-technik-Computers

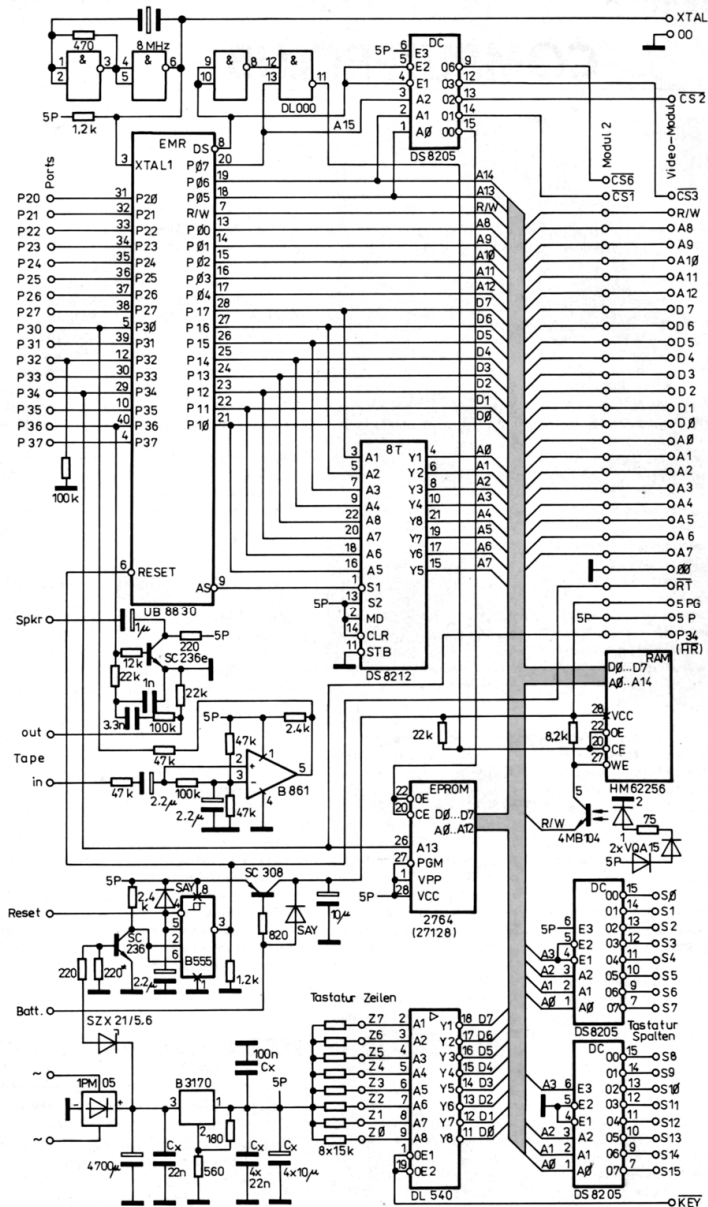
(mit /CS6 selektiert) auf insgesamt 16 Kbyte RAM ergänzt. Das Realisieren von 32 Kbyte mit dem HM 62256 hat natürlich praktische und finanzielle Vorteile.

Das Ergänzen des mit /CS0 aktivierten EPROM 2764 der Prozessorplatine mit einem zweiten auf Platz 2 (/CS1) bleibt dagegen auch weiterhin eine sinnvolle Option. Für weitere EPROM bleibt kein Platz im Adreßraum, da der Videomodul sowohl /CS2 als auch /CS3 belegt. Insgesamt wird die im Heft 6/90 dargestellte Adreßaufteilung realisiert, wobei der Modulplatz 2 aber vorerst frei bleiben kann.

### Audio-Interface

Die akustische Ausgabe über Spkr (Lautsprecher) und Tape out (Magnetbandausgang) übernimmt der Einchip-Mikrorechner UB 8830 D mit dem Anschluß P36. Das in der Schaltung dargestellte Netzwerk bezieht sich auf das KC-kompatible Kassettensystem. Für das 6-K-Betriebssystem, das bis zu 5 kHz verwendet, mußten der 1-nF-Kondensator entfernt und der 100-k-





Schaltzeichnung für die Kompaktvariante.

# JU+TE Computerclub

Widerstand sowie der 3,3-nF-Kondensator überbrückt werden. Die akustische Ausgabe über das Sichtgerät ist ebenfalls möglich. Das erfordert einen Widerstand von 22k zwischen Spkr und dem Audioeingang (rechts und links parallel) des Monitors.

Der Magnetbandeingang kommt mit einem Operationsverstärker aus, dessen Vorspannungen aus einem gemeinsamen Spannungsteiler stammen. Das sichert hohe Empfindlichkeit ohne Abgleich. Eine LED zur Ladekontrolle kann mit dieser einfachen Schaltung nicht bedient werden, für den EMR-Eingang P30 reicht der Verstärker jedoch sicher aus.

## Batteriestützung

Der Datenerhalt im RAM (HM 62256) auch bei ausgeschaltetem Rechner bleibt gewährleistet, wenn zwischen Batt. (+) und Masse zwei in Reihe geschaltete Nickel-Cadmium-Zellen an die Grundplatine angeschlossen sind. Das Umschalten erledigen der SC 308 und die Diode, wobei die Batterie bei eingeschaltetem Computer einen kleinen Nachladestrom erhält. Der Schwellwertschalter B 555 D und seine Beschaltung sichern das Rücksetzen rechtzeitig vor Abfallen der Betriebsspannung 5P und bis zum sicheren Einschwingen nach dem Einschalten. Der Optokoppler MB 104 sorgt für passive Schreibfreigabe, solange die Spannungsversorgung nicht steht. Den Anschluß Reset verbindet die Rücksetztaste zum Unterbrechen des Pro-

grammablaufs im Fehlerfall mit Masse (00). Da die Schaltkreisfreigabe des RAM mit einem DL 000 D erzeugt wird, erreicht sie trotz des 22-k-pull-up-Widerstandes im Schlafzustand nicht den passiven 1-Pegel. Daher liegt die Ruhestromaufnahme mit ca. 0,5 mA sehr hoch. Mit R6-Akkus (Mignon-Zellen) werden dennoch 1000 Stunden Datenerhalt gesichert. Bei selten eingeschaltetem Rechner sollte der den Akku-Ladestrom bestimmende Widerstand von 820 Ohm auf 180 Ohm (aber nicht darunter) verringert werden. Treten trotzdem Datenverluste beim Ein- und Ausschalten auf, liegt es an zu zeitiger Schreibfreigabe über den Optokoppler MB 104. Ein Schaltdraht von Pin 7 des B 555 D zum Pin 1 des MB 104 kann für ausreichende Verzögerung sorgen. Wird keine Stützbatterie verwendet, ist der Anschluß Batt. über einen Widerstand (ca. 2k) mit Masse zu verbinden. Das sichert die Spannungsversorgung des HM 62256 bei eingeschaltetem Rechner.

## Tastaturinterface

Die 0-aktiven Auswahlsignale für die maximal 16 Spalten der Tastatur erzeugen zwei 1-aus-8-Dekoder DS 8205 D. Das gleichzeitige Betätigen mehrerer Tasten setzt das Einfügen zusätzlicher Dioden (Kathode zum Dekoder) in alle Spaltenleitungen voraus. S0 sollte der linken Spalte, S1, S2 usw. fortlaufend nach rechts zugeordnet werden. Bei den Zeilen gehört Z7 zur obersten und Z6, Z5 usw. lückenlos angefügt zu den folgenden. Das sichert die Paßfähigkeit zum 6-K-

Betriebssystem. Das Freigabesignal /KEY erzeugt der Dekoder DS 8205 D auf dem Videomodul aus /CS3 und den Adreßbits A10, A11 und A12 (Adreßbereich %7C00 bis %7FFF).

## Stromversorgung

Die Prozessorplatine benötigt als einzige Betriebsspannung +5 V (5P), die mit ca. 500 mA belastet wird. Deren Erzeugung erledigt ein Spannungsregler B 3170 mit Standardbeschaltung. Bei entsprechend großer Kühlfläche (150 cm<sup>2</sup>) gestattet er das Versorgen von Prozessorplatine und voll bestücktem Videomodul (zusammen 1,5 A). Dafür ist der Gleichrichter 1 PM 05 allerdings zu schwach. Statt dessen sollte eine 3-A-Brücke (3 PM 05) am Trafo (8V/2,5A) angeordnet und deren Ausgang statt der Anschlüsse für + und - des 1 PM 05 mit der Platine verbunden werden. Der Ladeelko hat eine auch für diese Variante ausreichende Kapazität.

Bei Verwenden eines UB 8831 D als Prozessor erfordert die Stromversorgung des inneren RAM eine Lötbrücke zwischen den Pins 1 und 2.

Nun noch ein letzter Hinweis: Für störungsfreie Grafik-Anzeige mit dem 6-K-Betriebssystem sollte der 100-k-Widerstand an P32 nicht mit Masse, sondern mit P36 des Videoprozessors (Signal BUSY) verbunden werden. Viel Erfolg beim Aufbau und viel Spaß beim Ausprobieren!

*Dr. Helmut Hoyer*

Die Leiterplatte bekommt Ihr wie üblich bei Fa. Gerlich,

## JU+TE Computerclub

Markscheiderweg 08/17, Neu-  
brandenburg, 2000 per Nach-  
nahme. Wer dies nicht aus-  
drücklich abbestellt, erhält zu  
einem Aufpreis von 4,50 Mark  
die Beschreibung des Betriebs-  
systems in Form eines Sonder-  
drucks zur Video-Leiterplatte  
beigelegt.

Das Betriebssystem selbst  
überspielen wir auf von Euch  
eingesandte (leere) Kassetten  
oder einen gelöschten EPROM.  
Dazu aber bitte das Rückporto  
in Briefmarken für Päckchen  
plus Einschreibsendung mit-  
schicken! Unsere Anschrift:

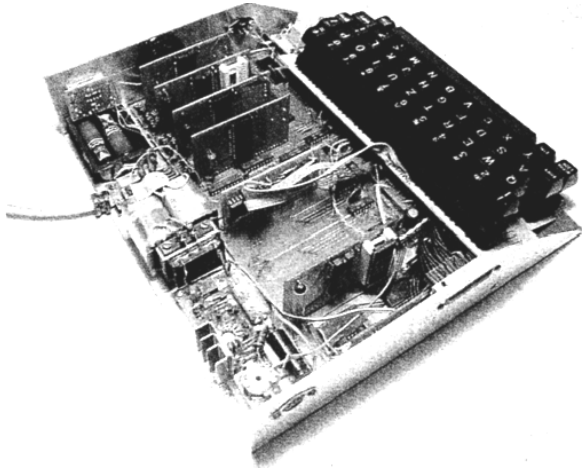
jugend und technik,  
PF 43, Berlin 1026.

## TINY-BASIC lernt Rechnen

Die vorläufig letzte Ausbaustufe des JU+TE-Computers zeichnet sich durch Farb- und Vollgrafikfähigkeit aus und arbeitet mit dem 6-K-Betriebssystem ES4.0. Unsere Abb. Zeigt die mit dem Videomodul nachgerüstete Variante (Gehäuse abgenommen).

*Als Taschenrechnerersatz und für wissenschaftliche Berechnungen taugte der JU+TE-Selbstbau-Computer bisher nichts. Mit sehr großen, kleinen und Kommazahlen wußte er nichts anzufangen. Die Verarbeitung von Gleitkommazahlen, lernt Tiny-Basic nun mit Hilfe eines Programms. Wir überspielen es kostenlos zusammen mit dem Betriebssystem ES4.0 (vgl. 6/90, S. 82ff.) auf eine eingesandte Leerkassette gegen beigelegtes Rückporto (Briefmarken im Wert von 1,60 DM).*

Die kleinste positive Zahl größer als Null ist die  $1E-98$ , die größte positive Zahl ist  $9.999999999E99$ . Für die negativen Zahlen gilt analoges. Die Zahlen werden in Variablen gespeichert. Um Platz zu sparen, erhalten sie keinen Namen, sondern eine Nummer. Die Numerierung beginnt mit null und endet mit einer durch die Speichergröße bestimmten Zahl. Die Anfangsadresse des Variablenspeichers wird auf %D6DC (Lowbyte) und %D6DF (Highbyte) eingetragen. Bei einem Eintrag auf %00 und %C0 beginnt der Variablenspeicher bei %C000. Bis zum Programmbeginn sind noch 4096 byte frei, die für 682



Variablen zu 6 byte ausreichen. Die höchste nutzbare Variablennummer lautet somit 681.

Der Maschinenprogramm-Aufruf C%DAFA am Anfang eines Basic-Programms startet eine neue Interpretationsroutine. Das Ausführen der bereits bekannten Befehle wird dadurch nicht beeinflusst. Es erscheinen aber drei neue Befehle:

**?„text“expr** druckt den Text „text“, berechnet dann den Ausdruck expr und druckt das Ergebnis aus. „text“ und expr können in beliebiger Anzahl und Reihenfolge auftreten. Steht nach dieser Folge ein Komma, dann wird nach dem Ausdrucken kein Return-Kode ausgegeben.

**!var=expr** berechnet den Ausdruck expr und übergibt den Wert an die Variable var. !var wartet auf eine Zahleneingabe und trägt die Zahl in die Variable var ein.

var steht für eine normale Basic-Variable A bis Z oder eine Gleitkommavariablen. Eine Gleitkommavariablen wird als

**/num/** geschrieben „num“ ist dabei die Nummer der Variablen und kann aus den Dezimalzahlen 0 bis 65535, den normalen Basic-Variablen A bis Z und den Operationszeichen + (Plus) und - (Minus) beliebig zusammengesetzt sein.

expr ist eine Zusammensetzung aus var, Gleitkommazahlen, den Operationszeichen + - \* / und den definierten Funktionen. Dieser Ausdruck wird von links nach rechts berechnet, ohne irgendwelche Prioritäten zu beachten. Eine andere Berechnungsreihenfolge kann nur durch das Setzen von runden Klammern erreicht werden. Die Gleitkommazahlen werden in der üblichen Computernotation eingegeben. Das „E“ als Exponentenkennzeichen muß groß geschrieben werden. Beispiele: 234, -98.07,  $5.6E-12$ , -34E45

Die Funktionen müssen aus kleinen Buchstaben bestehen. Das Argument folgt in runden Klammern. Es sind bereits folgende Funktionen definiert: **abs(expr)** liefert den absoluten Betrag von expr

# JU+TE Computerclub

**sgn(expr)** ergibt null, wenn expr null ist und -1 bzw. 1, wenn expr negativ bzw. positiv ist.

**int(expr)** setzt bei expr alle Nachkommastellen auf null. Trigonometrische und Potenzfunktionen werden später folgen. Auch in dieser Basic-Erweiterung wird keine umfangreiche Syntaxkontrolle durchgeführt. Folgende Fehler werden trotzdem erkannt und gemeldet:  
**E0:** unbekannter Befehl am Zeilenanfang oder nach einem Semikolon

**E3:** Syntaxfehler im expr (Fehlende Klammern z. B.)

**E12:** Zahlenbereichsüberschreitung oder Division durch null  
Die beiden folgenden Abschnitte enthalten Informationen über Programmdetails. Für den Gebrauch der Basic-Erweiterung ist deren Verständnis nicht erforderlich.

## Zahlenformat

Intern werden Dezimalzahlen mit einer Genauigkeit von neun bis zehn Stellen benutzt. Das erste Byte enthält im Bit 7 die Vorzeicheninformation: Ist es 1, dann ist die Zahl negativ. Bit 6 bis 0 geben den Exponenten zur Basis 100 (einhundert!) an. Dieser wurde noch zu %40 addiert. Zweites bis sechstes Byte enthalten die Ziffern im gepackten BCD-Code, wobei das zweite Byte das höchstwertige ist. Nach ihm steht der Dezimalpunkt. Nur die Zahl 0 (Null) hat eine besondere Darstellung: Alle sechs Byte sind %00. Dieses Zahlenformat wird auch im ATARI XL genutzt und vermeidet Rundungsfehler, die sonst beim Umwandeln von Dezimalzahlen in ein Binärformat auftreten.

## Rechenroutinen

Die Rechenroutinen nutzen den Registersatz %7X. Die Register %70-%75 werden als FP0 und die Register %76-%7B als FP1 bezeichnet. Ein Zahlenbereichsüberschreiten melden die Routinen FADD, FSUB, FMUL und FDIV mit gesetztem Carry-Flag. Alle Routinen verändern %7X.

%D000 FSUB FP0=FP0-FP1

%D003 FADD FP0=FP0+FP1

%D0EE FASC wandelt Zahl aus FP0 in einen ASCII-String um und legt diesen ab

(%76/77) ab, als letztes Zeichen steht ein Byte %00;

%D1F8 PASC druckt die Zahl aus FP0 aus, benutzt dazu den Puffer %F750-F76F;

%D234 ASCF wandelt den ASCII-String ab (%7C/7D) in eine Zahl in FP0, (%7C/7D) zeigt danach auf das erste nicht umwandelbare Zeichen;

%D424 FDIV FP0=FP0/FP1, wenn FP1 null ist, wird ebenfalls das C-Flag gesetzt, Registersatz %4X wird auf Stack zwischengespeichert;

%D540 FMUL FP0=FP0\*FP1, Registersätze %2X und %4X werden auf Stack zwischengespeichert;

%D578 HEXF wandelt vorzeichenbehaftete Integerzahl aus %76/77 nach FP0;

%D5E0 FHEX wandelt die Zahl aus 90 in vorzeichenbehaftete Integerzahl nach %76/77.

*Harun Scheutzwow*

# JU+TE Computerclub

## Leserbriefseiten

(Ausschnitte der Leserbriefseiten, den JU+TE-Computer betreffend.)

### JU+TE 11/89, S. 802

#### „JU+TE-Computer“ – Service

Die Nachfragen, was unsere Leiterplatten zum JU+TE-Computer angeht, reißen nicht ab Seit Anfang 1988 bis heute haben wir weit über 2000 Leiterplattensätze versandt bzw. vermittelt. Dabei hatten wir Unterstützung vom Industriepartner über Arbeitsgemeinschaften bis zu Amateuren. Seit geraumer Zeit erledigt es die Firma Gerlich aus Neubrandenburg, die zahlreichen Leiterplattenwünsche abzudecken. Das möchten wir im Einvernehmen mit ihm auch weiterhin so halten. Deshalb: Bitte alle Bestellungen für Leiterplatten zum JU+TE-Computer direkt an Gerlich  
Markseiderweg 08/417  
Neubrandenburg, 2000.

#### Achtung!

Alle Leser, die bisher Bestellungen an JU+TE geschickt haben, die aber noch keine Leiterplatten erhielten, bitten wir nunmehr erneut schriftlich um Bestellung, diesmal direkt an der Firma Gerlich. Wir bitten Euch für diese Maßnahme herzlich um Verständnis. Die Lieferungen der Firma Gerlich erfolgen in kürzeren Zeiträumen und in der Regel in höherer Qualität als die unserer bisherigen Partner. Ausgenommen davon sind alle Bestellungen, die während des Pfingsttreffens

in Berlin am JU+TE-Stand aufgegeben wurden. Dafür haben wir einen anderen Produzenten.

Die Firma Gerlich bietet folgende Platinen (ungebohrt) an:

- Prozessor. bzw. Grundplatine,
- Speicherplatine mit U 224,
- Speicherplatine mit U 6516,
- Magnetband-Interface 2K,
- Magnetband-Interface 4K und
- EPROMMER-Platine.

Die Preise für die Leiterplatten liegen in der Höhe der von JU+TE versandten.

Noch ein kleiner Hinweis:

Während des Solidaritätstages der Jugendmedien der DDR im Palast der Republik in Berlin am Sonntag, dem 26. November 1989, wird JU+TE wieder mit einem Computerstand vertreten sein. Neben dem Überspielen von Software und EPROM-Brennen für unseren Computer sowie praktischen Ratschlägen, hat sich ebenfalls die Firma Gerlich angekündigt. Er wird an unserem Stand Leiterplatten zum Verkauf anbieten. Wir erwarten Euren Besuch!

### JU+TE 01/90, S. 3

#### Nachtrag zu JU+TE 11/89

Im Computerklub dieses Heftes S. 871 ist uns ein Malheur passiert, für das wir Euch um Entschuldigung bitten möchten: Der Oszillograph mit dem JU+TE-Computer kann natürlich nur arbeiten, wenn man das zugehörige BASIC-Programm zur Hand hat. Durch ein Versehen wurde es leider nicht mit veröffentlicht. Hier nun verspätet das Programm:

```
100 PROC PTC[12];
```

```
PRINT " 4MHZ-OSZI"
110 PRINT " "
120 PRINT "0 >OSZI";
PRINT "1-8 >ZEIGEN"
130 INPUT Z; IF Z<0 THEN
GOTO 100
140 IF Z>8 THEN GOTO 100
150 IF Z>0 THEN GOTO 220
160 CALL %E180
170 LET Y=%E200,Z=%20
180 PROC SETEW[Y,GETRR[Z]]
190 LET Y=Y+2,Z=Z+2
200 IF Z<%50 THEN GOTO 180
210 GOTO 100
220 LET D=Z,C=Z-1*+%E200
230 LET E=0; PRINT "AB: "D
240 LET A=GETEB[C+E]
250 CALL %E168
290 PROC PTC[13]
300 LET E=E+1; IF E<6 THEN
GOTO 240
310 GOTO 130
```

### JU+TE 02/90, S. 82

#### JU+TE Computerclub

Ich lese regelmäßig Euren Computerclub und habe Euren Computer wegen seines extrem geringen Hardware-Aufwandes auch nachgebaut. Ich finde es gut, daß Ihr demnächst die Bildschirmsteuerung mit dem GDC 82720 vorstellt. Habt Ihr auch einen Anschluß für Diskettenlaufwerke auf der Basis des FDC 8272 vorgesehen?

Soldat A. Schedler  
7022 Leipzig

Wir haben die neue Bildschirmsteuerung mit einem 40poligen EMR-Schaltkreis entwickelt. Die Liste der nötigen Bauelemente wurde in JU+TE 1/1990 noch vor der Bauanleitung veröffentlicht. Der Anschluß von Diskettenlaufwerken entspricht nicht so ganz dem Charakter des JU+TE-Computers. Beschaffungsfragen stehen außerdem im Vordergrund. Deshalb ist das Veröffentlichen einer solchen Lösung in nächster Zeit nicht vorgesehen, was aber nicht ausschließt, daß wir dieses

# JU+TE Computerclub

Thema bei entsprechender Situation noch aufgreifen.

\*

Leider stieß ich erst jetzt auf die Zeitschrift JU +TE. Ich kaufte mir die Hefte mit dem Computer im Antiquariat. Nun baue ich mit meinem Vater gemeinsam daran. Im Radio läuft die Sendung „REM – das Computermagazin“. Dort geht es zur Zeit um die Sprache „Basicode“. Könntet Ihr nicht auch einen Bascoder für den JU+TE-Computer entwickeln?

Robert Niemann  
2793 Schwerin

Die Sprache „Basicode“ ist am ausgebauten (extended) BASIC orientiert, während der JU+TE-Computer mit einem sehr einfachen (tiny) BASIC arbeitet. Deshalb paßt beides nicht zusammen. Als weitere höhere Programmiersprache haben wir für unseren Computer wegen der guten Anpaßbarkeit an sehr viele Aufgaben keinen verbesserten BASIC-Dialekt, sondern FORTH-83 entwickelt.

\*

Auch ich baue und arbeite seit längerer Zeit am JU+TE-Computer. Dadurch konnte ich viel im Umgang mit Computern lernen. Um jedoch den sprunghaft gestiegenen Anforderungen gerecht zu werden, hat jetzt ein KC 85/4 den Platz des JU+TE-Computers eingenommen. Nun stehe ich vor dem Problem, das EPROM-Programmiergerät vom JU+TE-Computer an den KC 85/4 anzuschließen.

J. Weinhold  
9106 Niederfrohna

Der EPROM-Programmiermodul ist an alle Rechner anschließbar. Als Schnittstelle

werden ein bidirektionales 8-bit-Port und drei zusätzliche Computer-Ausgangssignale (Port-Pins) benötigt. Eine konkrete Lösung für den KC 85/4, der ja nicht ohne weiteres genügend Portkapazität bietet, ist uns leider nicht bekannt.

## JU+TE 03/90, S. 163

### Hilfe fürs EPROM-Brennen

Folgende Leser übernehmen das kostenlose Brennen der EPROM mit dem 4-K-Betriebssystem des JU+TE-Computers.

- Günter Luck, Oberhofer Str. 44/F-Nr. 12-3, Steinbach-Hallenberg, 6088. Typen. U 2716C und U 2732C
  - Detlef Rieck, H.-Mann-Allee 52, Potsdam, 1560. Typen U 2716 C bis U 27128 C
  - Herbert Jurich, CC-Bln.-Hellersdorf (KB 099), Beizenburger Str 63, Berlin, 1144. Typen. U 2716C und U 2732C
- Bitte die Schaltkreise rückversandfähig versenden (z. B. leere Kassettenschachtel) und Rückporto in Briefmarken (-,90 M) nicht vergessen!
- JU+TE dankt herzlich für die Entlastung bei EPROM-Brennen. Wir hoffen natürlich, daß sich noch weitere Leser mit solchen Angeboten an uns wenden. Die veröffentlichten Adressen könnten ja gleichzeitig auch den Informationsaustausch der „JU+TE-Computer“-Freunde untereinander befördern helfen.

### JU+TE-Computer

Übrigens, der Computer von JU+TE steuert beim Würschnitzthaler Carnevalsclub vorzüglich eine große Beleuchtungsanlage. Er erspart uns mit

wenig Mitteln doch recht viel Arbeit und vor allem Personal zum manuellen Steuern. Also, Eure Bauanleitung kam damals genau richtig, um einem jungen Faschingsklub auf die Beine zu helfen!



Hier noch ein Softwaretip: Ich habe, um mir das lästige Laden von Kassetten zu ersparen, alle oft benötigten Programme auf EPROM abgelegt. Jedoch lassen sich solche Programme aus JU +TE, die Call-Befehle im Anwenderbereich besitzen, nicht ohne Änderung vom EPROM aus mit PROC SETRR (6,...) starten (z. B. Schnellbrenner). Dazu müssen diese in den RAM auf %E000 gebracht werden. Dafür läßt sich das kleine Maschinenprogramm, das sich einfach mit 10 Call %... (20 End) starten läßt, verwenden. Es braucht nur die Anfangs- und Endadresse des umzuladenden Programms hexadezimal ohne „%“ eingegeben zu werden und schon kann man das Programm auf %E000 mit RUN von BASIC aus starten.

Klaus Klitzsch  
9123 Klaffenbach

## JU+TE 02/91, S. 88

### Unter Druck

Viele Leser fragen, ob an den JU+TE-Computer ein Drucker angeschlossen werden kann. Das ist kein Problem (siehe Broschüre „Betriebssystem

## JU+TE Computerclub

ES4.0"). Es bleibt dann nur noch, den Drucker mit dem Steuerzeichen %0D einen Wagenrücklauf und eine Zeilenschaltung ausführen zu lassen (bei den meisten Typen mit dem Einstellen von DIP-Schaltern anhand der Bedienungsanleitung möglich). Daß außerdem, z. B. in BASIC-Programmen, Ausgaben mit Umlauten vorkommen können, erklärt sich daraus: Im Computer werden für die Umlaute Codes verwendet, die im ASCII-Standard verschiedenen Steuerzeichen vorbehalten sind. Diese Vereinbarung läßt sich in EPSON- und kompatiblen Druckern mit untenstehendem Programm anpassen. Wer an der ausführlicheren Beschreibung interessiert ist, der kann sie kostenlos gegen Beilegen eines mit 0,50 DM frankierten und adressierten Rückumschlags bei uns anfordern. Viel Spaß beim Drucken!

```
,F512 70 15 A6 15 0D EB 03 D6
,F51A 17 A6 70 FD 31 60 0C F5
,F522 1C 69 2C 08 48 15 82 30
,F52A A2 34 6B 1E 1E 2A F7 49
,F532 15 50 FD D6 17 A6 E6 15
,F53A 1B D6 17 A6 E6 15 52 D6
,F542 17 A6 B0 15 D6 17 A6 50
,F54A 15 AF 06 E1 08 82 40 70
,F552 64 E6 15 1B D6 17 A6 E6
,F55A 15 52 D6 17 A6 E6 15 02
,F562 D6 17 A6 50 64 8B C8 1A
,F56A 1B 1C 1D 1E 1F 7F OD 7B
,F572 7C 7D 5B 5C 5D 7E 0A 00
```