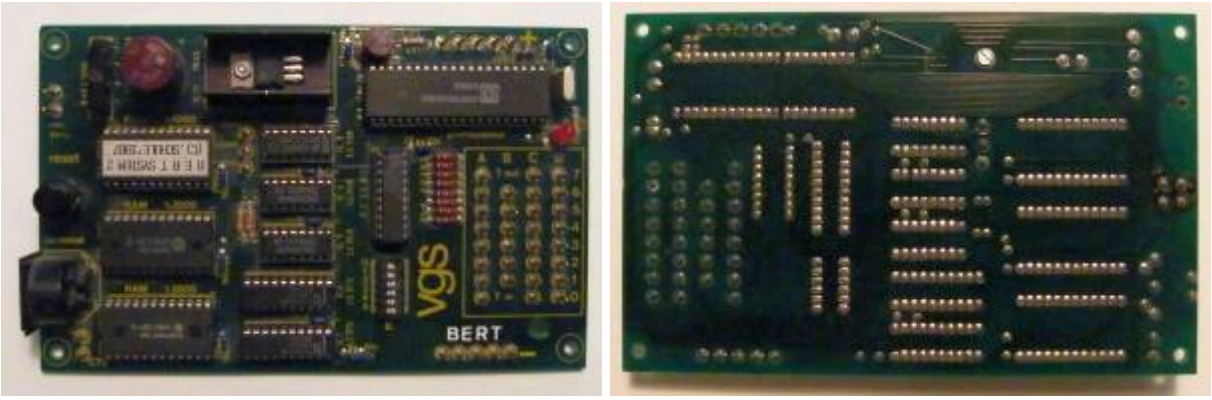
BERT

Das Mikroprozessorsystem BERT ist ein Einplatinencomputer auf Z8-Basis (Z8671), mit serieller Ansteuerung und vielfältigen Experimenten. Beschrieben wurde dieses System im Buch „Einführung in die Mikroprozessor-Anwendung“, vgs, 1987. Das Board konnte bei der vgs gekauft werden, ebenso diverse Zusatzplatinen des DIGIPROB-Systems.

Geplant war eine 10teilige TV-Serie im WDR für 1987/1988. Allerdings ist es leider nie zu einer Realisierung gekommen, nur das Begleitbuch und BERT (der eigentliche Fernseh-Star) entstanden.

Die Buchstaben von BERT stehen für „**B**asic **E**inplatinen **R**echner für **T**V-Serie“ .

Systembeschreibung



BERT Bestückungsseite, Leiterseite

BERT ist eine 10×16 cm große zweiseitige Leiterplatte, produziert von der Firma Thomsen-Elektronik, vertrieben durch den vgs-Verlag.

Zum Betrieb des BERT werden zusätzlich ein Netzteil (6-9 V, 300 mA) und ein V24-Kopplung zum PC benötigt. Im Buch sind V24-Interfaceschaltungen für Commodore V20, IBM-PC,XT,AT und APPLE-II beschrieben. Es kann aber auch jedes beliebige andere System über eine serielle Schnittstelle angeschlossen werden. Zu beachten ist, dass BERT hier mit TTL-Pegeln arbeitet und ggf. eine Pegelwandlung erforderlich ist.

Zusätzlich zu BERT mussten weitere Experimentierplatinen gebaut oder bei vgs erworben werden, z.B. eine Schalterplatine, LED-Platine, A/D-Wandler, Motor-Steuereinheit, 5×7-LED-Matrix, 4 stellige 7-Segment-Anzeige u.a. Im Buch finden sich zumindest Übersichts- und Teilstromlaufpläne, manchmal auch komplette Stromlaufpläne, aber leider keine Leiterplattenvorlagen.

BERT stellt 5 sogenannte Ports A .. E zur Verfügung, die über Kommandos abgefragt bzw. gesetzt werden können. Das sind digitale Ein-/Ausgabeleitungen, 6 bzw. 8 Bit breit.

Die Ports haben folgende Eigenschaften:

Port	**Beschreibung**
A	Port 2 des Z8 (P20..P27)

Port	Beschreibung
B	Port 3 des Z8 (P31..P36); Bit0 (P30) und Bit7 (P37) sind für die serielle Schnittstelle reserviert, deshalb nur 6 Bit
C	reiner Eingabeport mit 74LS373
D	reiner Ausgabeport mit 74LS373
E (DIP-Schalter)	kann nur gelesen werden; 6 Bit (Bit5..Bit0); Bit2..Bit0 stellen die Baudrate ein (s.u. Tabelle)

Das Buch zum BERT

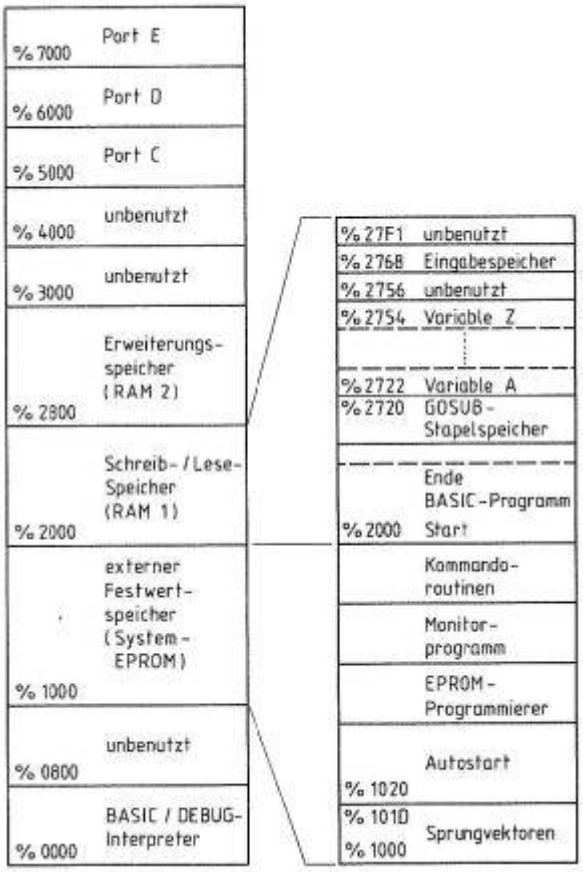


„Einführung in die Mikroprozessor-Anwendung“
Roland Schule; Axel Gruppe.
Unter Mitarb. von Michael Zillgitt. Jean Pütz (Hrsg.).
- 1. Aufl. - Köln: vgs, 1987. (Experimente)
ISBN 3-8025-1239-1

Inhalt:

1. Anwendung von Mikroprozessoren: aktuelle Technik und faszinierendes Hobby
2. Vom Schalter zum Computer
3. Impulse zählen und erzeugen
4. Codes und Zahlensysteme
5. Analog und Digital
6. Anzeige und Tastatur
7. Mechanische Antriebe und Positionierung
8. BERT ist ein eigenständiger Computer
9. Programm-Entwicklung mit BERT
10. Datenübertragung
11. Zwei Anwendungsbeispiele für BERT
12. Anhang

technische Daten



Merkmal	Beschreibung
CPU	Z8671
ROM	4K (2732)
RAM	4K
Takt	7,3728 MHz
Anzeige	-
Tastatur	-
Peripherie	Portanschlüsse herausgeführt; serieller Anschluss (5V)
Software	BASIC des Z8671, Monitoprogramm, EPROM-Programmierung, Assembler (ext. ROM)

Bereich	Verwendung
0000-07ff	Z8671 Basic/Debug
0800-0fff	-
1000-1fff	EPROM
2000-2fff	RAM1+2
3000-3fff	-
4000-4fff	-
5000-5fff	Port C
6000-6fff	Port D
7000-7fff	Port E (Schalter)

Die Adressdekodierung ist unvollständig. A15 wird nicht ausgewertet. d.h. die oberen Adressen %8000-%ffff werden auf die unteren Bereiche abgebildet. Speziell wird der DIP-Schalter auch im Adressbereich %f000-%ffff und speziell auf Adresse %FFFD gefunden. Hier liest der Z8671 auf den Bits 2..0 die Baudrate für die serielle Kommunikation aus.

DIP Schalter (Port E)	Baudrate
x x x 0 0 0	150
x x x 0 0 1	19200
x x x 0 1 0	9600
x x x 0 1 1	4800
x x x 1 0 0	2400
x x x 1 0 1	1200
x x x 1 1 0	110
x x x 1 1 1	300

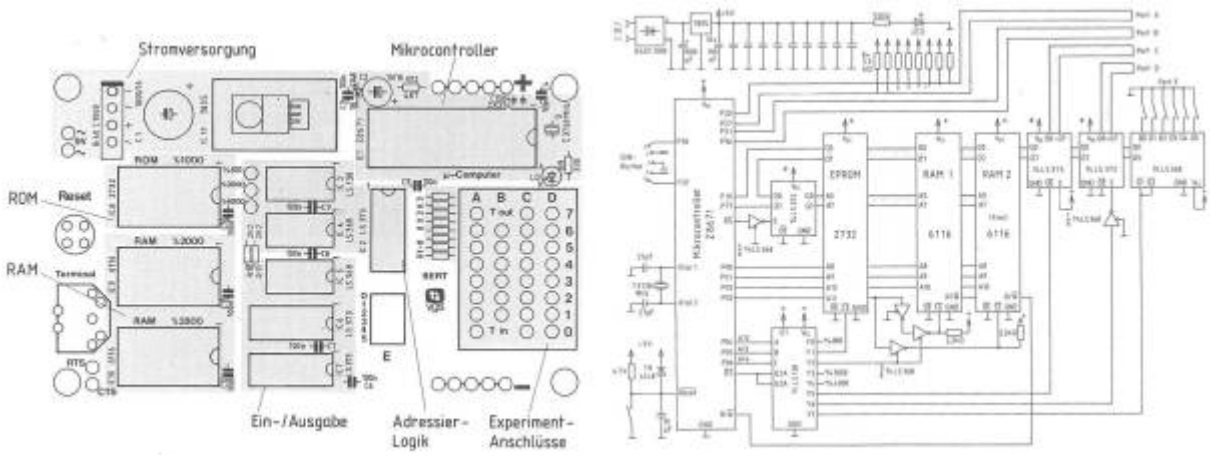
Literatur



1. „Einführung in die Mikroprozessor-Anwendung“, 1987
2. Z8_Family_Design_Handbook_Jun88.pdf (660 Seiten, 38 MB) (http://bitsavers.informatik.uni-stuttgart.de/pdf/zilog/z8/)
3. Z8671-Emulator (http://z8671sim.web-log.nl/)
4. New Life for the Z8671 BASIC Interpreter (http://www.armory.com/~rstevew/Public/Micros/Z8/Z8671-BASIC/Z8671-BASIC_ROM_Main.htm)

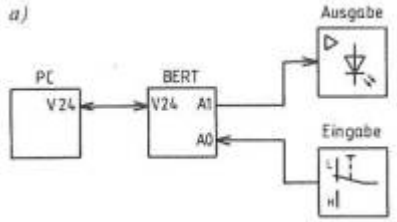
Downloads

- Listings etc. bert.zip
- Z8671-Unterlagen z8671.zip
- Assembler-ROM (leider nicht mehr auffindbar)



Bestückung der Platine, Stromlaufplan

Bedienung



BERT wird über die serielle Schnittstelle mit dem PC verbunden. Die Baudrate wird über die DIP-Schalter ausgewählt (110 bis 19200 Baud). Mit „GO@%1018“ wird der Kommandomodus aktiviert. Über Kommandos wie „A=42“ oder „A?“ werden z.B. Port A beschrieben bzw. abgefragt. Es gibt Kommandos für Porteingabe, Portausgabe, Handshake, Zeitgeber, Analog-Eingabe, Motor-Betrieb, Multiplex, Sonstiges. Das nebenstehende Bild verdeutlicht die Nutzung von BERT als (intelligentes) Interface für den PC

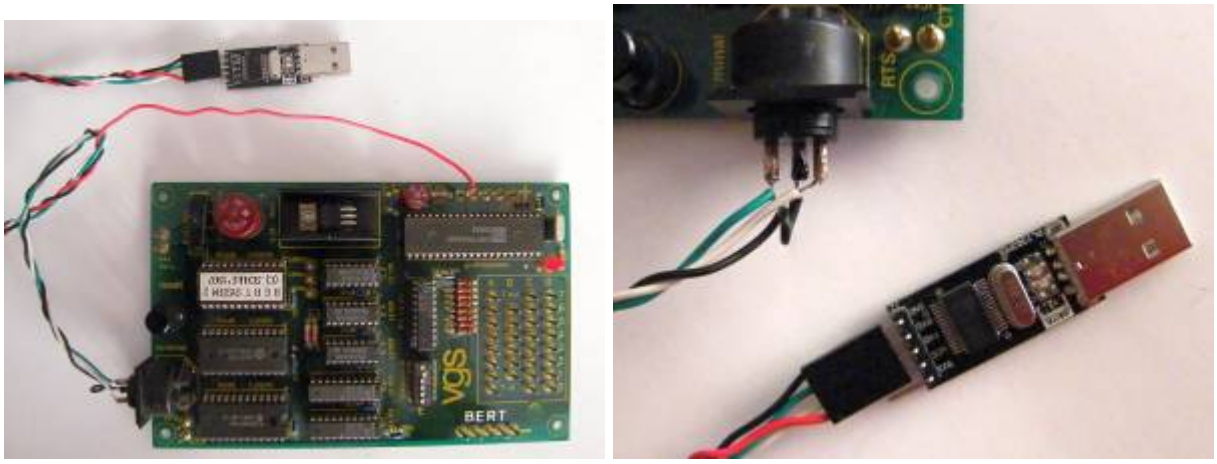
Außerdem kann BERT als eigenständiger Computer genutzt werden. In diesem Fall kommt das interne BASIC/DEBUG des Z8671 zum Einsatz. Sowohl Direktmodus als auch Programmmodus sind möglich. Im EPROM ist Platz freigehalten, um ein BASIC/DEBUG-Programm unterzubringen, was direkt nach Reset ausgeführt wird.

Dies wird genauer im Buch Kapitel 8 ff. beschrieben.

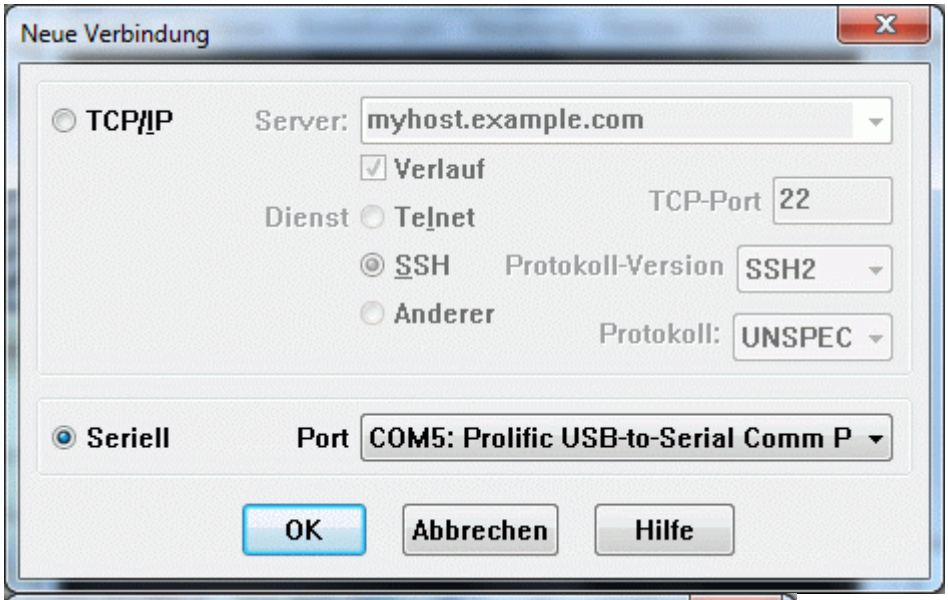
serielle Schnittstelle

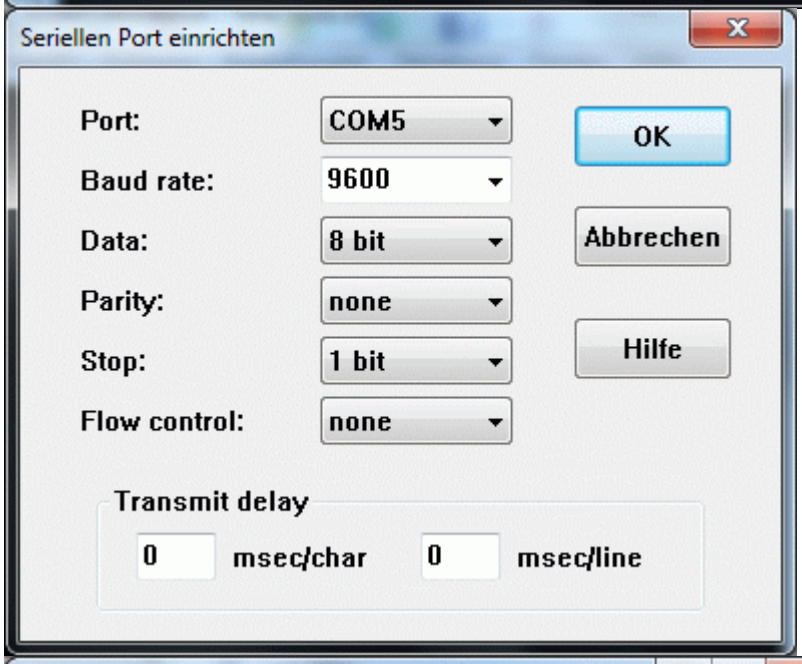
Der Z8671 kommuniziert über ein serielles Interface der Pins P30 und P37 (s. Unterlagen Z8671). Der Anschluss an einen V24-Schnittstelle am PC erfordert einen Pegelwandler, z.B. mit MAX232. Passende Schaltungen sind im Buch Anhang B zu finden.

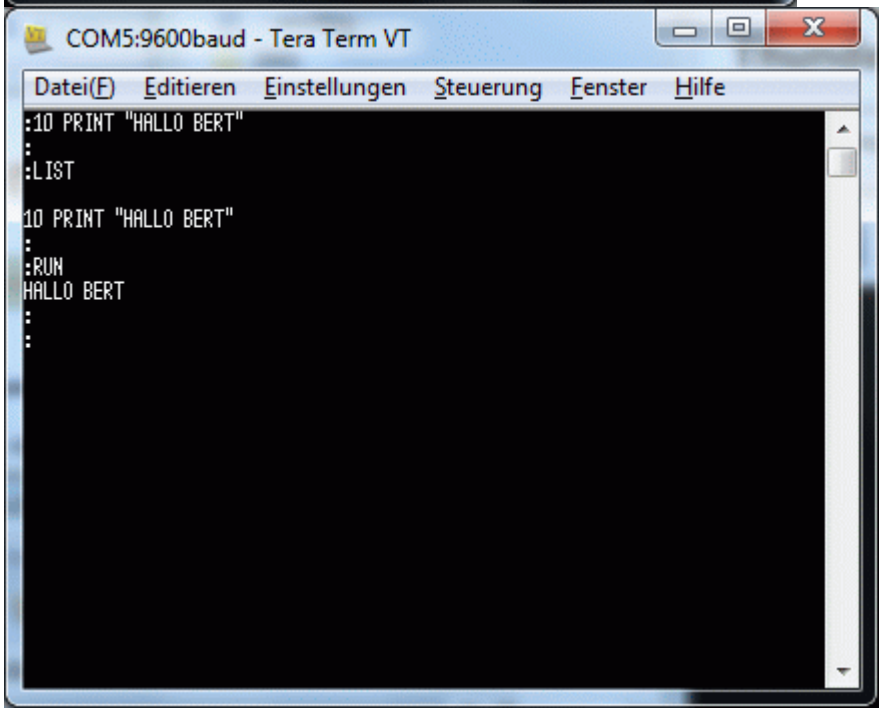
Ich habe meinen BERT über einen USB2TTL-Anschluss (1,90€ bei ebay incl. Versand!) angeschlossen. Der PC übernimmt dabei dank USB auch gleich die Stromversorgung von BERT mit.



Anschluss an BERT: Serielle Verbindung über die Diodenbuchse; das rote Kabel (+5V) dient zur Stromversorgung über USB


Neue Verbindung
TCP/IP
Server:
myhost.example.com
Verlauf
Dienst
Telnet
TCP-Port
22
SSH
Protokoll-Version
SSH2
Anderer
Protokoll:
UNSPEC
Seriell
Port
COM5: Prolific USB-to-Serial Comm P
OK
Abbrechen
Hilfe


Seriellen Port einrichten
Port:
COM5
Baud rate:
9600
Data:
8 bit
Parity:
none
Stop:
1 bit
Flow control:
none
OK
Abbrechen
Hilfe
Transmit delay
0
msec/char
0
msec/line


COM5:9600baud - Tera Term VT
Datei(F) Editieren Einstellungen Steuerung Fenster Hilfe
:10 PRINT "HALLO BERT"
:
:LIST
10 PRINT "HALLO BERT"
:
:RUN
HALLO BERT
:
:

Einstellungen beim Terminal-Programm **TeraTerm**: serieller Anschluss, 9600 Baud; Test mit BASIC/DEBUG

Initialisierung

- V24-Parameter am BERT einstellen (DIP-Schalter, z.B. x x x 1 0 1)
- V24-Parameter am PC einstellen (z.B. 1200 8,N,1)
- BERT Reset
- BERT sendet ':'
- PC sendet 'GO@%1018',ENTER (Aktivieren des Kommandomodus)

Am PC wird GWBASIC bzw. ehem. BASICA für die Ansteuerung von BERT genutzt. Die Listings im Buch beginnen alle mit Zeilennummer 100. Vor jedes dieser Programme muss ein Initialisierungsprogramm gestellt werden, dass obige Schritte abarbeitet. Das Initialisierungs-Programm INIT für den IBM-PC sieht z.B. so aus:

```
10 REM Programm INIT (IBM-PC)
12 REM Initialisierung von BERT
15 REM als Interface
17 REM
20 OPEN "COM1:1200,N,8,2" AS #2
25 WIDTH 40 : KEY OFF : CLS : COLOR 0,7
30 PRINT"         BERT-INTERFACE-BETRIEB         ";
35 PRINT"    1200 BAUD: PORT E AUF 5 = XXX101    "; : COLOR 7,0
40 PRINT"   BITTE RESET-TASTE AN BERT DRUECKEN!  ";
45 IF EOF(2) THEN 45
50 DU$=INPUT$(LOC(2),#2) : IF DU$<>":" THEN 45
60 PRINT#2,"GO@%1018"
65 FOR ZEIT=1 TO 1000 : NEXT
70 DU$=INPUT$(LOC(2),#2)
80 CLS : COLOR 0,7 : PRINT"         BERT-INTERFACE-BETRIEB        ";
90 PRINT"          START BASIC-PROGRAMM          "; : COLOR 7,0
```

Kommandos

Im Kommandomodus werden Kommandos als ASCII-Zeichenketten an BERT gesendet. Ein Kommando wird mit ENTER (0Dh) abgeschlossen. Zahlen werden als Dezimalzahlen übertragen. Die Befehle sind alle gleich aufgebaut: Buchstaben benennen den Port, eventuell gefolgt von einer Bit-Nummer (z.B. 'C3', oder aber eine Abkürzung des Kommandos ('M'). Der Befehl wird mit **=** oder **?** abgeschlossen. Es schließt sich die Ausgabe bzw. das Einlesen eines Zahlenwertes an.

= bedeutet Ausgabebefehl: Es wird ein Wert an BERT übertragen. Nach dem '=' muss eine 16-bit-Zahl (dezimal) stehen.

? bedeutet Abfrage (Eingabebefehl): BERT sendet nach Ausführung des Befehls eine mit ENTER (0Dh) abgeschlossene Zeichenkette (16-Bit-Zahlenwert, dezimal) an den PC.

```
S.36    Portein/ausgabe
    'Ax=',wert,ENTER    Port A, Bit x setzen; x = 0..7; Wert = 0..1
```

```
    'Ax?',ENTER        Port A, Bit x abfragen            BERT sendet
Zahl+ENTER
    'A=',wert,ENTER        Port A setzen; Wert = 0..255
    'A?',ENTER        Port A abfragen                BERT sendet Zahl+ENTER
    dito für die anderen Ports B,C,D,E
S.49    Ereigniszählen        Impule (1/0-Flanken) an B1 zählen
    'Z=',vorteiler,ENTER    Vorteiler 1..64; stellt Ereigniszähler auf 0
    'Z?',ENTER        aktueller Stand des Ereigniszählers    BERT sendet
Zahl+ENTER

S.52    Zeitmessen        Zähler startet mit 0/1-Flanke an B1 und stoppt bei
1/0-Flanke an B1
                Zähler läuft mit Quarz/8 = 921,6 kHz
    'T=',vorteiler,ENTER    Vorteiler 1..64; stellt Zähler auf 0
    'T?',ENTER        aktueller Stand des Zählers        BERT sendet
Zahl+ENTER

S.55    Impulserzeugung        Einzelimpuls an B6 ausgeben
    'I=',impulsdauer,ENTER    impulsdauer 1..16383; Impulslänge =
Impulsdauer * 1.085 µs

S.56    Frequenzerzeugung    symmetrische Rechteckschwingung an B6 ausgeben
    'P=',impulsdauer,ENTER    impulsdauer 1..16383; halbe Periodendauer =
Impulsdauer * 1.085 µs
    'F=',frequenz,ENTER    frequenz 29..32767 in Hz

S.71    Portein/ausgabe mit Quittierung
    'AH?',ENTER        Einlesen Port A mit Quittierung        BERT sendet
Zahl+ENTER
                'H' bedeutet Handshake
                B1 := /DAV (an Bert), B6 := RDY (von Bert)
    'AH=',wert,ENTER    Ausgabe Port A mit Quittierung
                B1 := RDY (an Bert), B6 := /DAV (von Bert)

S.98    A/D-Wandler ADC0808    Beschaltung etc. s. Buch
    'ADn?'            n 0..3; Abfrage A/D-Wandler Kanal n

S.114   Multiplexen
    'M=',x,ENTER        x > 0; Es werden die Bits A0..A(x+1) zur
Multiplexauswahl aktiviert
                M=0 stoppt Multiplexen
    'DMx=',wert,ENTER    Wert für Multiplexstelle x; Ausgabe an D-Port
S.125   'CMx?',ENTER        Abfrage Multiplexstelle x; Einlesen an C-Port
    'AMx?',ENTER        Abfrage Multiplexstelle x; Einlesen an A-Port

S.139   Gleichstrommotorsteuerung
    'DC=',tastv,ENTER    tastv 0..255; Gleichstrommotorsteuerung,
                erzeugt an B& Impulsfolge, Impulsfrequenz ca. 56Hz,
                Tastverhältnis = tastv/256

S.154   Schrittmotorbetrieb    s. Buch
```

```
    'SP=',zykl,ENTER    zykl 1..64; Schrittdauer festlegen; Schrittdauer =
zykl * 277,76 µs
    'Sn=',schrittz,ENTER    n 0..3, schrittz -255..255; Schrittzahl-Register
n setzen
    'Sn?',ENTER        Schrittzahl-Register n abfragen
```

Eingabe (vgl. Kapitel 2)

Die Ports A und C und die untere Hälfte des Ports B eignen sich zur Eingabe von Bitmustern. Aber auch Port E, der Baudraten-Einsteller, gehört zu den Eingabeports, wenn auch hier nur Schalterstellungen und keine elektrischen Signale einzulesen sind. Bei der Eingabe an Port A wird Handshake und Multiplex abgeschaltet. Das adressierte Bit bzw. der ganze Port wird auf Eingabe geschaltet. Beim Einlesen von Port B bzw. von Stift B1 wird Handshake, Zeit- und Zählereingabe abgeschaltet.

Ausgabe (vgl. Kapitel 2)

Eine Ausgabe kann am Port A, ferner an der oberen Hälfte des Ports B und am Port D erfolgen. Bei einer Ausgabe an Port A wird das adressierte Bit bzw. der ganze Port auf Ausgabe geschaltet. Außerdem wird Handshake und Multiplex abgeschaltet. Bei einer Ausgabe an Port B bzw. an Stift B6 werden Handshake-, Zeitgeber- und Motor-Betrieb (Kommando DC=expr) abgeschaltet. Bei einer Ausgabe an Port D werden alle acht Multiplexregister des Ports D gleichermaßen gesetzt, so daß sich selbst bei laufendem Multiplex statische Bitmuster an Port D ergeben.

Handshake (vgl. Kapitel 4): Beim Absetzen von Handshake-Kommandos wird nicht nur Port A benutzt, sondern auch die Stifte B1 und B6 bzw. B4. Die AH-Kommandos schalten Multiplex-, Zeitgeber- und Motorbetrieb ab. Mit dem Kommando AS=expr wird der Wert von expr an Port A ausgegeben. Anschließend folgt ein kurzer negativer Impuls an B4. Getriggert durch diesen Impuls kann ein externes Gerät die Daten von Port A übernehmen. Beim Kommando AS? erscheint am Ausgang B4 ein negativer Impuls, der ein externes Gerät veranlassen. soll, seine Daten an Port A zu legen. Nach dem Einlesen von Port A wird B4 wieder auf H geschaltet. Der USR-Aufruf am Ende des Kommandos AH=expr liest die als Quittung ausgegebene Ziffer 1 ein; vgl. Seite 72.

Zeitgeber (vgl. Kapitel 3)

Die Zeitgeber-Kommandos benutzen die Stifte B1 (Tin) und B6 (Tout) Die Zeitgeber-Kommandos schalten Handshake-, Multiplex- und Motor-Betrieb ab.

Analog-Eingabe (vgl. Kapitel 5)

Die Analog-Eingabekommandos benutzen Port C und die Stifte B3 und B4. Wird die Analogkanal-Adresse an den Stiften B5 und B6 ausgegeben, so werden Handshake-, Zeltgeber- und Motorbetrieb (Kommando DC=expr) abgeschaltet.

Motor-Betrieb (vgl. Kapitel 7)

Das Pulsbreiten-Kommando DC=expr benutzt den internen Zeitgeber mit Ausgabe an Stift B6. Handshake-, Multiplex- und Zeitgeber-Betrieb werden daher abgeschaltet. Die Schrittmotor-Kommandos benutzen ebenfalls den internen Zeltgeber; die Ausgabe erfolgt jedoch an Port D. Handshake-, Multiplex- und Zeitgeber-Betrieb werden abgeschaltet, aber die Ausgabe-Moglichkeit an Stift B6 bleibt erhalten.

Multiplex (vgl. Kapitel 6)

Die Multiplex-Kommandos verwenden den internen Zeitgeber zur Steuerung der Abläufe. Daher werden Handshake-, Zeitgeber- und Motor-Betrieb abgeschaltet. Port A wird, beginnend mit Bit 0, entsprechend der Multiplex-Stellenzahl auf Ausgabe geschaltet, um die Multiplex-Auswahlsignale zu erzeugen. Die verbleibenden Stifte von Port A sind auf Eingabe geschaltet. Bei Abfrage der Port-A-Multiplexregister wird sowohl das Ausgabe- als auch das Eingabe-Bitmuster übergeben. Die Ausgabe an Port 0 erfolgt nach dem Einschalten des Multiplex nicht mehr durch das Port-D-Register (%3E), sondern durch die Port-D-Multiplexregister (%53 - %5A). Durch einfache Ausgabe-Kommandos werden diese jedoch identisch mit dem Port-D-Register (%3E) vorbesetzt. Daher bleibt die Ausgabe an Port 0 scheinbar unverändert bestehen, wenn das Multiplex-Verfahren benutzt wird, jedoch keine DMn-Kommandos abgesetzt wurden.

Sonstiges

Mit dem Kommando G=expr kann ein BERT-Maschinenprogramm aufgerufen werden, das bei der Adresse expr beginnt. 'G=',expr

Ergänzende Hinweise

Einige BERT-Kommandos benötigen zur Durchführung wegen des Umfangs ihrer Aufgaben einige Millisekunden Ausführzeit. Während dieser Zeit kann ein kurz darauffolgendes Kommando nicht erkannt werden oder wird verstümmelt empfangen. In diesen Fällen muß im PC-BASIC-Programm eine kurze Warteschleife nach dem Absetzen des Kommandos erfolgen. Dies gilt insbesondere für die Kommandos des Multiplexverfahrens und der Motorsteuerung.

BASIC/DEBUG-Modus

Nach einem Reset befindet sich BERT bzw. der Z8671 im BASIC/DEBUG-Modus. Wird kein Programm auf Adresse 1020h gefunden, sendet das BASIC/DEBUG des Z8671 einen ':' als Bereitschaftszeichen für eine mögliche Kommunikation. Für die Umschaltung in den Kommandomodus wird nun mittels BASIC/DEBUG-Direktkommando 'GO@%1018' das Kommandomodusprogramm gestartet. Im BASIC/DEBUG-Modus können beliebige BASIC/DEBUG-Befehle an den Z8671 gesendet werden, z.B.

```
10 PRINT "HALLO BERT"
LIST
RUN
```

Mit ESC kann ein laufendes BASIC/DEBUG-Programm gestoppt und wieder in den Direktmodus gewechselt werden.

Mit der Funktion USR() können Maschinenprogramme aufgerufen werden. So können auch im BASIC/DEBUG-Modus die Funktionen des Kommandomodus genutzt werden. Die konkreten Adressen und Aufrufkonventionen stehen im Anhang D des Buchs.

Außerdem sind noch einige weitere Programme im EPROM untergebracht, die im BASIC/DEBUG-Modus aufgerufen werden können:

Hex/Ascii-Speicherdump

```
GO@%101B,anfangsadr,endadr+1
```

Gibt des Speicher zeilenweise zu je 8 Byte hexadezimal und Ascii aus. In der Ascii-Ausgabe wird Bit7 ignoriert; Steuerzeichen werden als '.' ausgegeben.

Z8-Register anzeigen

```
GO@%17C3,anfangsadr,endadr+1
```

Z8-Register ändern

Register werden direkt mit BASIC/DEBUG-Befehlen geändert. Beispiel

```
^8=%2100
NEW
```

setzt den Anfang des BASIC/DEBUG-Speichers auf %2100.

```
@%0C=@%0C-2
@%0A=@%0A-2
NEW
```

verschiebt den Bereich für Variabeln und den Stack um 512 Byte nach unten.

RAM ändern

```
GO@%1830,anfangsadr.
```

Es wird die Adresse angezeigt. Jetzt wird der Speicherinhalt als Hexzahl (ohne %) eingegeben. Nach Enter erscheint die nächste Adr. usw. Beendet wird der RAm-Editor durch sofortiges Enter ohne Eingabe einer Hexzahl.

Alternativ kann der RAM auch mit dem Indirektionsoperator ^ geändert werden.

EPROM-Programmierung

Über eine kleine Zusatzschaltung (Buch S. 196) können mit BERT anwendungsspezifische EPROMs 2732 (25V) und 2732A (21V) programmiert werden. Dabei wird eine Kopie des BERT-EPROMs erstellt und zusätzlich ein Anwender-BASIC/DEBUG-Programm aus dem RAM an die Adresse %1020 kopiert, so dass dieses bei RESET sofort ausgeführt wird.

```
GO@%17C0
```

startet den kompletten Programmierungszyklus:

- Löschtest: Es wird getestet, ob der EPROM gelöscht ist, d.h. überall mit %FF gefüllt ist. Andernfalls gibt es eine Fehlermeldung 'EPROM FEHLER'

- Formatierung: die Kommandoroutinen und die Sprungvektoren aus BERT werden in den EPROM gebrannt. Anschließend wird der EPROM kontrollgelesen. Bei Fehlern erfolgt ein Abbruch mit Fehlermeldung 'EPROM FEHLER'.
- Das BASIC/DEBUG-Programm aus dem RAM wird nach %1020 bis max. %1BA8 kopiert. Reicht der Platz nicht aus, wird auch hier mit einer Fehlermeldung abgebrochen.
- War alles erfolgreich, erscheint die Ausschrift 'EPROM OK'

Die einzelnen Programmierungsabschnitte können auch einzeln aufgerufen werden. Details siehe Monitorprogramm.

Mini-Assembler

Von B. Holzhauer stammte eine zusätzlicher EPROM, der mit BERT genutzt werden kann und einen kleinen Z8-Assembler enthält. Dieser Assembler läuft auf Adresse %0800-%0fff. um ihn im Steckplatz für RAM2 betreiben zu können, müssen am EPROM Pin 18 und 21 abgewinkelt werden, und Pin 21 mit +5V und Pin 18 mit Lötpunkt %800 verbunden werden.

Der Mini-Assembler arbeitet mit Zeilennummern. Er kennt keine Kommentare. Der Code wird als Basicprogramm erfasst. Mit GO@%800 wird der Assemblerlauf gestartet.

Beispiel: Einlesen einer zweistelligen BCD-Zahl an Port C
Binärzahl := Einerziffer + 2 * Zehnerziffer + 2*4 * Zehnerziffer

```
10 #ABS %13FD
20 LD R4,%50
30 LDC R3,@RR4
40 LD R2,R3
50 AND R3, #%0F
60 AND R2, #%F0
70 SWAP R2
80 RL R2
90 ADD R3,R2
100 RL R2
110 RL R2
120 ADD R3,R2
130 CLR R2
140 RET
```

Die Routine kann dann mit USR(%13FD) aufgerufen werden.

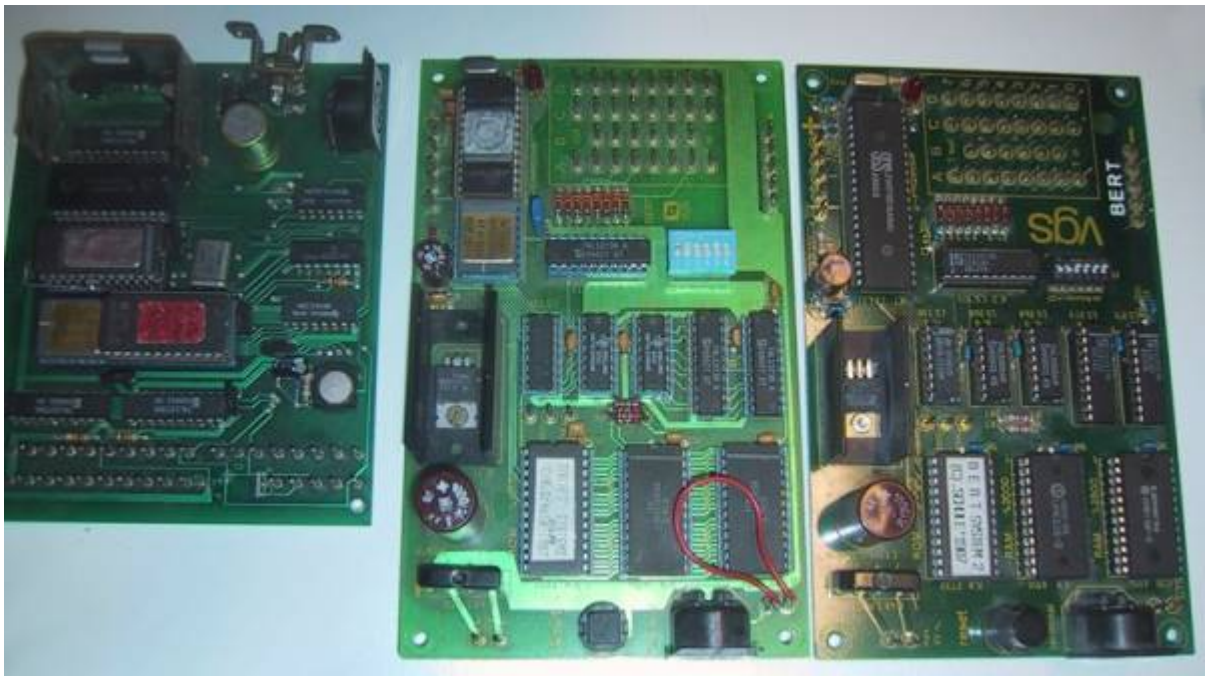
Um sie als neues Kommando 'BCD?' für den Kommandomodus bereitzustellen, muss die Kommandotabelle im System-EPROM ergänzt werden. Ab %1A87 sind 11 Byte dafür freigehalten (s. Monitorprogramm).

%1A87: 'B'+%80, 'C', 'D', '?', %13FD»7, %13FD&&%7F (die Startadr. wird in 2x7Bit aufgeteilt)

Leider gilt der Assembler-ROM als verschollen, es ist nichts mehr darüber zu finden, auch der Programmautor hat nichts mehr.

Sonstiges

Ich hatte Kontakt zum Entwickler des BERT, Herrn Dr. Roland Schulé:



Entwicklungsmuster, Vorserie, Serie

Dr. Schulé: „ich kann Ihnen aus rechtlichen Gründen weder die Software duplizieren noch andere Unterlagen zur Verfügung stellen. Ich habe meine Elektronikkiste aufgeräumt und noch vier Exemplare des BERT vorgefunden.

- Zwei frühe Entwicklungsstadien mit reichlich Handverdrahtung,
- ein Vorserienmodell
- und ein Serienmodell (siehe Foto).

Die Entwicklungs- und Vorserienmodelle tragen statt des Z8671 einen Z8613, d.h. ein Z8-Prozessor mit externem ROM.

Ich habe zwar seinerzeit den BERT entwickelt, aber die Rechte daran liegen bei der vgs (www.vgs.de) in Köln. Den Schaltplan des BERT können Sie sich sicher aus den Beschreibungen im Begleitbuch und den Unterlagen des Z8671 ableiten. Das Platinen-Layout liegt mir nicht vor; es wurde von der Fa. Thomsen-Elektronik (www.thomsen-elektronik.de) entwickelt, die auch den Computer produzierte.“

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

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/homecomputer/bert?rev=1471438503

Last update: **2016/08/17 12:55**