

Frage:
In den letzten beiden Programmen sind die unmittelbaren Befehle nachlässig geschrieben. Was wurde vergessen?

Antwort:
Hinter den Konstanten fehlt die Angabe D, H bzw. B. In allen Fällen waren dezimale Zahlen gemeint. Richtig ist: LA IR k 100D usw.

Assembler-Anweisungen

Die Programme sind in Assemblersprache formuliert. Sie müssen noch durch ein spezielles Übersetzungsprogramm in den Maschinencode „assembliert“ werden. Erst dann ist für den Mikrocomputer ein ablauffähiges Programm vorhanden. Dazu erfinden wir

noch zwei Anweisungen, die ausschließlich an das Übersetzungsprogramm gerichtet sind.

ORG n

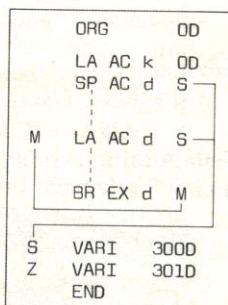
Das Programm wird ab Zelle n im Speicher abgelegt (organisiert).

VARI m

Jede symbolische Adresse bzw. Variable benötigt einen Speicherplatz, der durch die Anweisung VARI in Zelle m reserviert wird. VARI erhält als Programmmarke die symbolische Adresse. Beziehen sich symbolische Adressen nur auf die Programmmarken (z. B. M bei Befehl BR EX d), entfällt die Platzreservierung durch VARI. Das ist logisch, denn die Adresse ist schon eindeutig durch die Marke definiert.

LA AC k 0	Speicherzelle für die Summe S reservieren und 0 setzen.
SP AC d S	
LA IR k 100	Zahl 100 in das IR laden
LA AC k 201	Anzahl der erforderlichen Schritte in Z speichern.
SP AC d Z	
M LA AC d S	S in den Akku laden
AD i 0	((IR)+0) + (AC) in den Akku
SP AC d S	Summe in Zelle S speichern
IN IR	Inhalt IR inkrementieren (+1)
DE d Z	Inhalt Z dekrementieren (-1)
LA AC d Z	Solange Z ungleich 0 ist, Sprung nach M
BR EX d M	
END	

Vollständiges Programm



Das Programm steht ab Speicherzelle 0 im ROM

M bedeutet an dieser Stelle: Programmmarke (auf eng. LABEL)

M bedeutet an dieser Stelle: symbolische Adresse / Variable

Platz 300 im RAM reservieren
Platz 301 im RAM reservieren

Fortsetzung folgt

AIM-65 als Morse-Schreibmaschine

In Heft 14/1979 war beschrieben, wie sich der Mikrocomputer KIM-1 zusammen mit einem ASCII-Terminal zum Erzeugen von Morsezeichen einsetzen läßt. Der AIM-65 (und damit auch der PC-100 von Siemens) besitzt bereits eine ASCII-Tastatur, arbeitet jedoch mit anderen Timer- und Monitorprogramm-Funktionen, so daß ein Umschreiben des KIM-Programmes auf ihn kaum möglich war, obwohl in beiden Fällen die gleiche CPU benutzt wird.

Während das KIM-Programm die ASCII-Zeichen in einer Interrupt-Routine von der Tastatur holte und im „Hauptprogramm“ als Ton aussandte, arbeitet das AIM-Programm (Bild 1) genau umgekehrt: Das Senden der Morsezeichen geschieht per Interrupt-Routi-

ne; das Hauptprogramm (Startadresse 0200) holt die ASCII-Zeichen von der Tastatur und speichert sie in einem Ausgabebuffer (FIFO) ab, der im Adressbereich 0300...03FF liegt. Der Interrupt wird periodisch vom Timer 1 im 6522-VIA ausgelöst. Der Timer arbeitet im freilaufenden Modus und erzeugt am Port PB 7 gleichzeitig die Tonfrequenz.

Das Programm belegt eine Reihe von Zero-Page-Adressen, nämlich 00E2 (Save X), 00E3 (Zeichenbuffer), 00E4 (Punktzähler), 00E5 (Zeitschleife für eine Punktlänge), 00E6 (Ausgabezeiger), 00E7 (Eingabezeiger), 00E8...00FC (Anzeige) und 00FF (Save X). Alle wichtigen Betriebsparameter lassen sich gezielt ändern; dazu gehören die Morse-Geschwindigkeit (0062), die

Pause zwischen den einzelnen Zeichen (0085) und Worten (009C) sowie die erzeugte Tonfrequenz (0215, 021A). Sie beeinflusst auch die Geschwindigkeit.

Drückt man die Taste „Return“, so erfolgt ein sofortiger Sprung zum AIM-Monitorprogramm (022E), ohne die laufende Ausgabe zu unterbrechen. Die Taste F2 führt ebenfalls zu einem Sprung in das Monitorprogramm (00B3), allerdings erst, wenn das letzte Morsezeichen ausgegeben ist, d.h. sobald Eingabe- und Ausgabezeiger übereinstimmen. Dank des Ausgabebuffers darf man bis zu 255 Zeichen vorausschreiben; das Format der Morsezeichen im Buffer stimmt mit dem des erwähnten KIM-Programmes überein, d.h. die einzelnen Bits werden, beim

```

0000 80 56 80 80 80 80 80 80 80 80 CE 8C 56 94
0010 FC 7C 3C 1C 0C 04 84 C4 E4 F4 56 56 80 80 80 32
0020 4C 60 88 A8 90 40 28 D0 08 20 78 B0 48 E0 A0 F0
0030 68 D8 50 10 C0 30 18 70 98 B8 C8 80 80 00 80 80

0040 85 FC A2 00 86 FF B5 E9 95 E8 09 80 20 7B EF A6
0050 FF E8 E0 14 D0 EE A5 FB 60 00 00 00 48 C6 E5 D0
0060 42 A9 A0 85 E5 C6 E4 F0 14 10 38 06 E3 F0 15 B0
0070 04 A9 01 D0 02 A9 03 85 E4 A9 C0 D0 02 A9 40 8D

0080 0B A0 D0 1F A9 02 85 E4 86 E2 A6 E6 E4 E7 F0 11
0090 BD 00 03 F0 15 85 E3 C9 80 D0 04 A9 08 85 E4 E6
00A0 E6 A6 E2 2C 04 A0 68 40 A2 FF 9A A9 00 8D 0B A0
00B0 8D 0E A0 6C FC FF

0200 A9 5C 8D 04 A4 A9 00 8D 05 A4 A9 40 8D 0B A0 A9
0210 C0 8D 0E A0 A9 00 8D 04 A0 A9 02 8D 05 A0 A9 00
0220 A2 16 95 E6 CA 10 FB 20 40 EC C9 0D D0 03 6C FC
0230 FF 20 40 00 AA BD E0 FF A6 E7 E6 E7 9D 00 03 18
0240 90 E5
    
```

AIM-65-MORSESCHREIBMASCHINE

◀ Bild 1. Programm zum Erzeugen von Morsezeichen mit den Mikrocomputern AIM-65 von Rockwell oder PC-100 von Siemens. Startadresse ist 0200, NF-Ausgang ist der Port PB 7

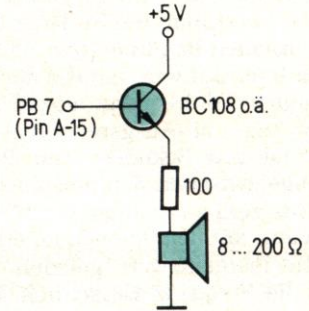


Bild 2. Anschluß eines Lautsprechers an den Mikrocomputer

höchstwertigen beginnend, als Punkte (0) und Striche (1) interpretiert, wobei die letzte 1 nicht gesendet wird, sondern das Zeichenende anzeigt. Selbstverständlich können in der Umwand-

lungstabelle (0000....003F), die die ASCII-Zeichen hex 20...5F umfaßt, auch Sonderzeichen programmiert werden. Bild 2 zeigt den Anschluß eines kleinen Lautsprechers an den AIM-65. Fe.

AIM-65 sendet RTTY

Das hier vorgestellte Programm für den Mikrocomputer AIM-65 enthält eine interessante Lösung zum Erzeugen eines Frequenz-Umtast-Signales (FSK) für die Modulation eines Funkfenschreibsenders. Es benutzt zwei programmierbare Timer im VIA 6522.

Das Programm

Funkamateure führen Fernschreib-Verbindungen (RTTY) heute meist noch im 5-bit-Baudot-Code durch. Will

```

0200 A9 A1 8D 04 A4 A9 02 8D
0208 05 A4 A9 00 85 E8 85 E9
0210 20 EF EC A9 80 8D 02 A0
0218 A9 C0 8D 0B A0 8D 0E A0
0220 A9 00 8D 05 A0 58 A2 10
0228 A9 1F 20 67 02 CA D0 F8
0230 20 F0 E9 A9 0D 20 72 02
0238 20 73 E9 C9 0D F0 F1 C9
0240 5B F0 15 AE 15 A4 E0 32
0248 30 EB C9 20 F0 E2 C9 2D
0250 D0 E3 20 72 02 4C 30 02
0258 A2 00 A9 0D 20 72 02 BD
0260 70 01 F0 CC E8 D0 F5 A4
0268 E8 E6 E8 0A 09 C0 99 00
0270 03 60 A0 00 D9 68 00 F0
0278 03 C8 D0 F8 98 48 29 20
0280 C5 FD F0 0F 85 FD C9 00
0288 F0 04 A9 1B D0 02 A9 1F
0290 20 67 02 68 C9 08 D0 05
0298 20 67 02 A9 02 20 67 02
02A0 60
    
```

Bild 1. Das Hauptprogramm weist keine Besonderheiten auf und wurde deshalb nur hexadezimal aufgelistet. Das Unterprogramm bei 0272 wandelt ein ASCII-Zeichen in den Baudot-Code um und speichert es in den FIFO

man solche Sendungen mit dem AIM-65 empfangen, so kann man sich des in Heft 15, Seite 884, veröffentlichten Programms bedienen, das die empfangenen Zeichen als durchlaufende Schrift auf dem 20stelligen alphanumerischen Display darstellt. Bei den üblichen 45 Bd (bit/s) kann man noch gut mitlesen.

Will man den AIM-65 (oder natürlich den PC-100 von Siemens) auch zum Senden benutzen, so führt dies zu folgenden Forderungen:

1. Der ASCII-Code des Tastenfeldes muß in ein bitserielles Baudot-Signal umgewandelt werden.
2. Da man u. U. manchmal schneller als 45 Bd schreibt, muß ein Bufferspeicher (FIFO) vorhanden sein.
3. Das Programm sollte auf Tastendruck einen vorher gespeicherten Standardtext, z. B. das eigene Rufzeichen nebst Standort, senden können.

Das in Bild 1 hexadezimal aufgelistete Programm sorgt außerdem dafür, daß der notwendige externe Hardware-Aufwand minimal bleibt; es erzeugt nämlich das Baudot-Ausgangssignal direkt als Tonfrequenz.

```

02A1 2C 09 A0 BIT A009
02A4 10 48 BPL 02EE
02A6 48 PHA
02A7 A9 57 LDA #57
02A9 8D 09 A0 STA A009
02AC A5 EA LDA EA
02AE F0 23 BEQ 02D3
02B0 38 SEC
02B1 66 FE ROR FE
02B3 AD 07 A0 LDA A007
02B6 90 0B BCC 02C3
02B8 F0 15 BEQ 02CF
02BA A9 E9 LDA #E9
02BC 8D 04 A0 STA A004
02BF A9 00 LDA #00
02C1 F0 09 BEQ 02CC
02C3 D0 0A BNE 02CF
02C5 A9 86 LDA #86
02C7 8D 04 A0 STA A004
02CA A9 01 LDA #01
02CC 8D 05 A0 STA A005
02CF C6 EA DEC EA
02D1 10 1A BPL 02ED
02D3 A9 08 LDA #08
02D5 85 EA STA EA
02D7 86 EB STX EB
02D9 A6 E9 LDX E9
02DB E4 E8 CPX E8
02DD F0 0C BEQ 02EB
02DF BD 00 03 LDA 0300,X
02E2 C9 C0 CMP #C0
02E4 F0 0C BEQ 02F2
02E6 85 FE STA FE
02E8 E8 INX
02E9 86 E9 STX E9
02EB A6 EB LDX EB
02ED 68 PLA
02EE 2C 04 A0 BIT A004
02F1 40 RTI
02F2 A2 FF 9A A9 00 8D
02F8 0B A0 8D 0E A0 4C
02FE 00 00
    
```

Bild 2. Interrupt-Routine zur Ausgabe der Baudot-Zeichen im FIFO als fertiges FSK-(Frequency Shift Keying-)Signal am Port PB 7. Das Programmteil bei 02F2 dient zum Rückschalten in den Empfangsmodus und schaltet den Timer-Interrupt ab

Dazu wird die Interrupt-Routine in Bild 2 verwendet. Sie bedient sich der beiden Timer im User-VIA 6522. Timer