

## KIM versteht Pseudo-Befehle

Der nachfolgende Beitrag zeigt, daß es mit recht geringem Software-Aufwand schon möglich ist, den Mikrocomputer KIM-1 dazu zu bewegen, übliche aus der Programmiersprache BASIC gewohnte Worte zu „verstehen“ und als Steuerbefehle zu interpretieren. Dies erleichtert den Umgang mit dem System ungemein; die direkte Eingabe von Steuerworten erspart oft eine Unzahl mühsamer Einzeloperationen, z. B. das Speichern von Daten auf eine bestimmte Adresse zur Definition des auf Band aufzuzeichnenden Speicherbereichs, das Verschieben von Programmteilen u. a.

Die hier gezeigte Programmversion decodiert nur Steuerbefehle, stellt also keine Programmiersprache im üblichen Sinne dar, da die Befehle nicht als Programmbeefehle gespeichert, sondern sofort ausgeführt

werden. Die verwendete Methode, wie die Worte decodiert werden, läßt sich aber selbstverständlich auch bei der Verwirklichung von Compilern z. B. zum Aufruf und Transfer ganzer Programm-Moduln einsetzen.

Bild 1 zeigt, wie die einzelnen Worte decodiert werden. Alle möglichen Steuerbefehle sind als Folge von ASCII-Zeichen in einer Tabelle gespeichert. Das Format dieser Tabelle ist:

```
FF XXXX ABCDEFG FF
```

Die Daten FF kennzeichnen dabei den Beginn und das Ende eines Tabellenteils, wobei die Enddaten FF gleichzeitig wieder den Beginn eines neuen Wortes darstellen. XXXX steht hier für eine vierstellige Adresse; zu dieser Adresse springt das Programm, wenn das nachfolgende Wort (hier ABCDEFG) decodiert wurde. Das an dieser Adresse dann stehende Programm bewirkt die Ausführung des gewünschten Steuerbefehls.

In der vorliegenden Programmversion darf jedes Wort maximal aus 10 Buchstaben bestehen; ASCII-Zeichen von 00...20 werden ignoriert. Will man z. B. START eingeben, tippt aber versehentlich STURT ein (keine Reaktion), so braucht man nur z. B. die Leertaste zu drücken (ASCII 20) um das Wort richtig einzugeben. (Das Drücken der Return-Taste ist übrigens nicht erforderlich.)

### 1 Programmieren in Maschinensprache

Nehmen wir einmal an, wir wollen ein Programm zur Anzeige von FFFFFFFF auf dem KIM-Display realisieren. Dieses Programm kann z. B. die Form haben:

```
0200 A9 FF
0202 85 F9
0204 85 FA
0206 85 FB
0208 20 1F 1F
020B 4C 00 02
```

Die Befehle sind hier bereits in der richtigen Länge pro Zeile geschrieben, und vorne steht eine vierstellige Adresse. Diese Formatierung geschieht bei Verwendung des Pseudo-Befehles „HEX“ automatisch; auch „LIST“ liefert dieses Format. Wollen wir obiges Programm eingeben, so tun wir einfach folgendes:

Zunächst einmal starten wir das Decodierprogramm an der Adresse 1200; es erfolgt sofort der Ausdruck „READY“. Dann geben wir „NEW“ ein; wir wollen ja

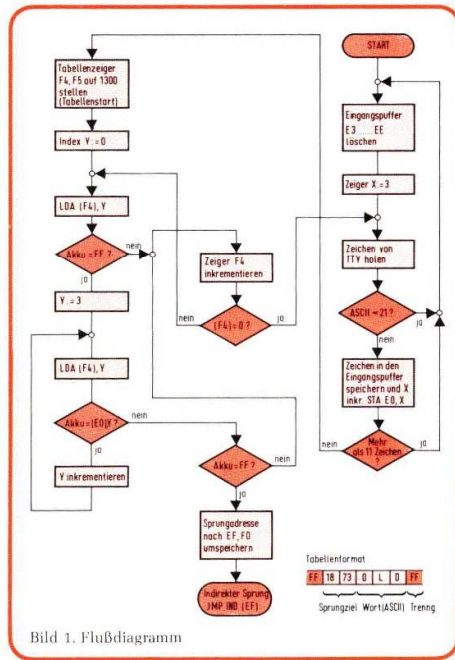


Bild 1. Flußdiagramm

ein neues Programm realisieren. Ein Leerraum (Space) wird nach NEW automatisch ausgedruckt, sobald das Wort decodiert wurde; jetzt müssen wir noch eine Kennnummer eingeben, z. B. 01, die später bei der Bandaufzeichnung als Identifikation für das Anwenderprogramm dient. Jetzt wird sofort am linken Rand der nächsten Zeile die Adresse 0200 ausgedruckt, und das System wartet nun auf eine Eingabe. (NEW geht automatisch immer zur Adresse 0200, da dies für das KIM-System eine sinnvolle Programm-Startadresse ist.)

Da wir das Programm in sedezimaler (hexadezimaler) Form eingeben müssen, beginnen wir diese Eingabe mit dem Steuerbefehl HEX. Er hat zur Folge, daß sofort in der nächsten Zeile nochmals die Adresse 0200 geschrieben wird; wenn man jetzt die Sedezimal-Daten bzw. -Befehle eingibt (A9FF85F985FA usw.), so erkennt das System selbständig die richtige Befehlslänge und liefert ein Format, wie es bei der vorher dargestellten Programm-Auflistung bereits zu sehen ist.

Haben wir alles eingegeben, so steht in der nächsten Zeile die Adresse 020E. Um den Sedezimal-Modus zu beenden, tippen wir noch FF ein; und wieder wird eine neue Zeile mit der Adresse 020E eröffnet. An diese Stelle schreiben wir einfach END. Daraufhin springt das System sofort zur Startadresse (hier 0200). Um die richtige Eingabe zu kontrollieren, kann LIST verwendet werden. Dieser Befehl listet das Programm zwischen Start- und Endadresse wiederum in richtiger Formatierung (wie oben) auf, druckt READY aus und springt wieder an die Startadresse zurück.

## 2 Einige Steuerbefehle

Mit dem Befehl SAVE kann das Programm jetzt auf Band aufgenommen werden; dabei wird das Hypertape-Programm verwendet, das auch bereits im „First Book of KIM“ veröffentlicht wurde und eine Geschwindigkeit von etwa 600 Bd ermöglicht. Der Befehl OLD, gefolgt von der gewünschten Identifikations-Kennzahl (hier 01), sorgt für das Wiedereinlesen vom Band. Danach erfolgt allerdings ein Sprung zum KIM-Monitor auf die Adresse 0000 bzw. FFFF (letzteres bedeutet eine Fehlermeldung). An der Adresse 1200 kann das Programm dann wieder gestartet werden.

Will man das zuvor eingegebene Programm starten, so braucht man nur (an der Adresse 0200) RUN einzugeben. Um dieses Programm allerdings wieder anzuhalten, muß ein RST- oder NMI-Interrupt verwendet werden.

In diesem Zusammenhang hat es sich als praktisch erwiesen, die Decodier-Startadresse als NMI-Vektor zu verwenden (17FA=00 und 17FB=12) und eine nicht belegte Taste des Terminals oder Fernschreibers mit der Stop-Taste des KIM zu verbinden. Dadurch kann man ein laufendes Anwenderprogramm jederzeit anhalten, indem man diese Taste drückt, und es

erfolgt dann der Ausdruck von READY; das System erwartet die Eingabe neuer Steuerbefehle.

Die übrigen Steuerbefehle und ihre Wirkung gehen aus der *Tabelle* hervor. Jeder richtig decodierte Steuerbefehl, dem ein Argument folgen muß, z. B. die Band-Identifikationszahl oder die gewünschte Adresse, erzeugt hinter dem letzten Buchstaben sofort einen Leerraum, dem das Argument folgen muß. Dieser Leerraum wird auch bei SAVE erzeugt, um anzuzeigen, daß der Befehl richtig decodiert wurde. Nach der Aufzeichnungszeit meldet sich das System mit READY wieder.

## 3 Mögliche Änderungen

Die in *Bild 2* aufgelistete Programmversion ist für ein KIM-System mit 4 KByte externer RAM-Erweiterung gedacht; daher liegt es im Adressenbereich

Tabelle des Befehlsvorrats

Befehl	Argument	Wirkung
BYE	–	Sprung zum KIM-Monitor auf die aktuelle Adresse
RUN	–	Ausführung des Anwenderprogramms ab der aktuellen Adresse
NEW	Progr.-Kennziffer	Speicherung der Kennziffer an der Adresse 17F9 und Sprung an die Adresse 0200
LINE	Neue Adresse	Sprung zu einer neuen Adresse
START	–	Aktuelle Adresse wird als Startadresse deklariert
END	–	Aktuelle Adresse wird als Endadresse deklariert
OLD	Progr.-Kennziffer	Laden eines Programms mit der def. Kennziffer, anschließend Sprung zum KIM-Monitor
SAVE	–	Aufzeichnen des Programms zwischen START und END auf Band (Hypertape)
LIST	–	Auflisten des Programms von der aktuellen Adresse bis zu END; Formatierung mit richtiger Befehlslänge
STRING	ASCII-Zeichen	Speichern von ASCII-Zeichen ab der aktuellen Adresse bis „Escape“ (1B); letzteres wird nicht mitgespeichert
INSERT	n	Einfügen von n Bytes an der aktuellen Adresse und entsprechende Korrektur der Endadresse ( $0 \leq n \leq FF$ )
DELETE	n	Löschen von n Bytes, sonst wie INSERT
HEX	Sedez.-Bytes	Eingabe eines Programms in Maschinsprache, beendet mit FF; automatische Formatierung wie bei LIST
?	–	Ausdruck bzw. Anzeige des Bytes an der aktuellen Adresse

1100...13FF. Wenn man bei allen 3-Byte-Befehlen, bei denen das dritte Byte 11, 12 oder 13 lautet, eine entsprechende Änderung vornimmt, läßt es sich in beliebige andere „Pages“ (Speicher-Seiten) transferieren. Will man den Befehlsvorrat erweitern, so muß man ei-

nige Programmteile aus der Page 13 in einen anderen Speicherteil legen und die Adressen entsprechend korrigieren. Damit wird in dieser Page 13 mehr Platz für die Wortetabelle frei, die nach dem oben erwähnten Format beliebig erweitert werden kann.

1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199																																																																																																																																																		
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Bild 2. Programmliste