
THOMSON

LEITFADEN ZUM

TO⁷⁰7

BDV Basilius Verlag

© 1985 Thomson Micro G.m.b.H., Deutschland

Coproduktion
BDV Basilius Verlag AG
Basel/Schweiz

ISBN-Nr. 3-85560-038-4

Alle Rechte vorbehalten. Kein Teil dieses Buches darf ohne ausdrückliche Genehmigung des Verlags in irgendeiner Form reproduziert oder übermittelt werden, weder in mechanischer noch in elektronischer Form, inkl. Fotokopie.

Satz: BDV Basler Druck- und Verlagsanstalt Basel/Schweiz
Druck: BDV Basler Druck- und Verlagsanstalt Basel/Schweiz
Einband: Grollimund, Reinach/Schweiz

Inhalt

1. Lernen Sie Ihren Mikrocomputer kennen	7
Gerätesbeschreibung	8
Aufstellung	10
Inbetriebnahme	13
Der Lightpen	16
Das Programm-Modul	19
Der Programm-Rekorder	20
Die Tastatur	21
Der Speicher	22
Erweiterungen und Peripheriegeräte	26
Die wichtigsten technischen Daten	28
Funktionsfehler und Prüfungen	30
2. Wie Ihr TO7-70 funktioniert	33
I Ihr Computer	33
II Arbeitsprinzip	35
III Der Transistor und die binäre Logik	37
IV Die binäre Codierung der Informationen	41
V Der Mikroprozessor und die Speicher	47
VI Das Programm-Modul	55
VII Die Peripheriegeräte	57
VIII Der Bildschirmspeicher	61
IX Was geschieht unter der Tastatur des TO7-70	65
X Mikro-Wörterbuch	69
3. Der Monitor des TO7-70	73

Page blanche

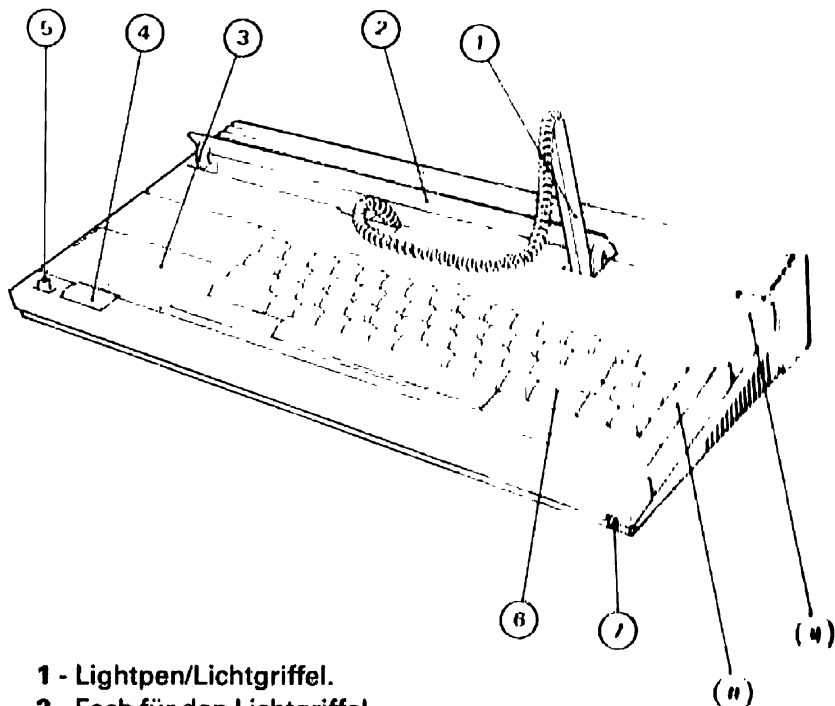
1. Lernen Sie Ihren Mikrocomputer kennen

Ihr Mikrocomputer T07-70 ist speziell auf Ihre individuelle oder familiäre Anwendung ausgerichtet, und Sie können auf ihm

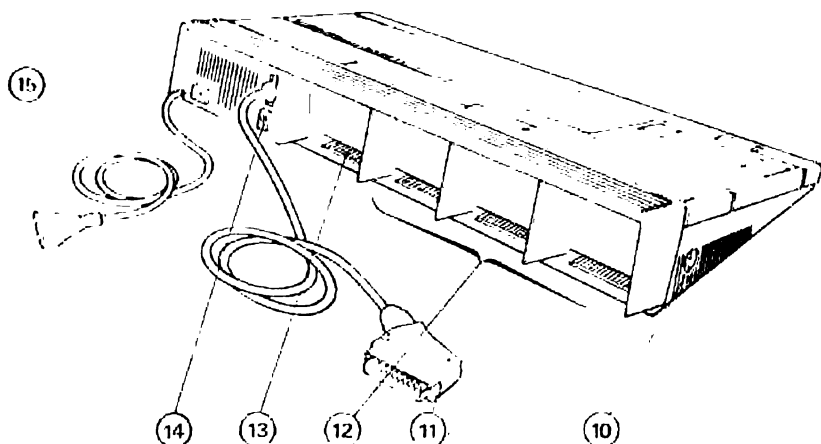
- die verschiedensten Programme einsetzen:
 - Ausbildungshilfe
 - Haushaltsführung
 - Schreiben von Musikstücken, Erzeugen von Bildern und Tönen
 - Spiele etc....
- je nach dem in den Computer eingelegten Modul in einer der Sprachen BASIC, LOGO, ASSEMBLER, FORTH etc. programmieren
- Zeichnungen oder Graphiken in 16 verschiedenen Farben herstellen
- eine Komposition über 4 Oktaven schreiben
- mit Hilfe des eingebauten Lightpens Zeichnungen erstellen
- mit Hilfe des Mischbild-Zusatzes Bilder, die Sie mit Ihrem T07-70 geschaffen haben, mit einem externen Videobild
 - * aus einer Fernsehsendung
 - * aus einem Videorekorder mit Bildsendeausgang
 - * aus einer Videokameramischen.

Gerätbeschreibung

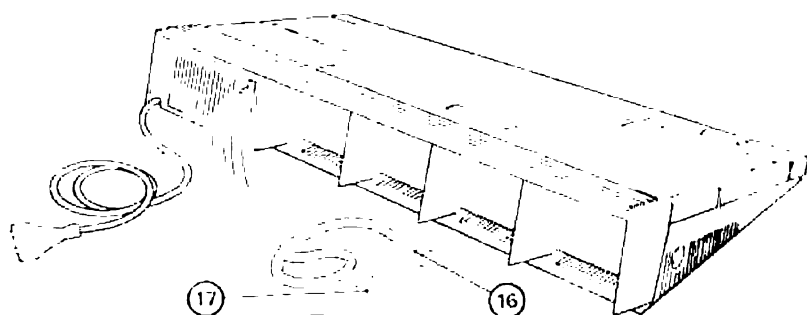
Damit Sie sich mit Ihrem Mikrocomputer vertraut machen können, folgt zunächst eine äussere Beschreibung des Gerätes



- 1 - Lightpen/Lichtgriffel.
- 2 - Fach für den Lichtgriffel.
- 3 - Fach für das Programm-Modul.
- 4 - Taste zum Öffnen der Klappe (3).
- 5 - Verriegelung der Öffnungs-Taste (4).
- 6 - Tastatur zum Eintippen von Klein- oder Großbuchstaben, Sonderzeichen und Zahlen.
- 7 - Kontrollampe: Betriebsbereitschaft des Gerätes.
- 8 - Kontrollampe: Tastatur schreibt in Kleinbuchstaben.
- 9 - Taste, die im Laufe der Anwendung die erneute Initialisierung der Zentraleinheit ermöglicht (Anzeige der Titelmelodie).



- 10 -** Buchse zum Anschluß eines Programm-Rekorders.
- 11 -** SCART-Anschluß zum direkten Anschluß an den Fernsehempfänger.
- 12 -** Steckplätze zum Anschluß von Erweiterungsmoduln.
- 13 -** Spezieller Steckplatz zum Anschluß der Speichererweiterung.
- 14 -** 12 V-Ausgang.
- 15 -** Netzschalter der Zentraleinheit.



- 16 -** Audio-Video DIN-Buchse zum Anschluß an den Videoeingang eines Fernsehgerätes.
- 17 -** Antennenstecker.

Aufstellung

Bei der Aufstellung Ihres Mikrocomputers müssen Sie einige elementare Regeln beachten:

- Einige Bestandteile im Inneren des Computers werden während des Betriebs warm und müssen belüftet werden; infolgedessen sollten Sie vermeiden, Ihr Gerät in der Nähe einer Wärmequelle aufzustellen oder die Belüftungsöffnungen zu verstopfen.

Anschluß an Ihr Fernsehgerät

- Die Verbindung zwischen Ihrem Mikrocomputer und Ihrem Fernsehgerät oder Monitor wird wie folgt vorgenommen:
- entweder über den Euro-Norm-Anschluß (SCART-Stecker) (Abb. 1);

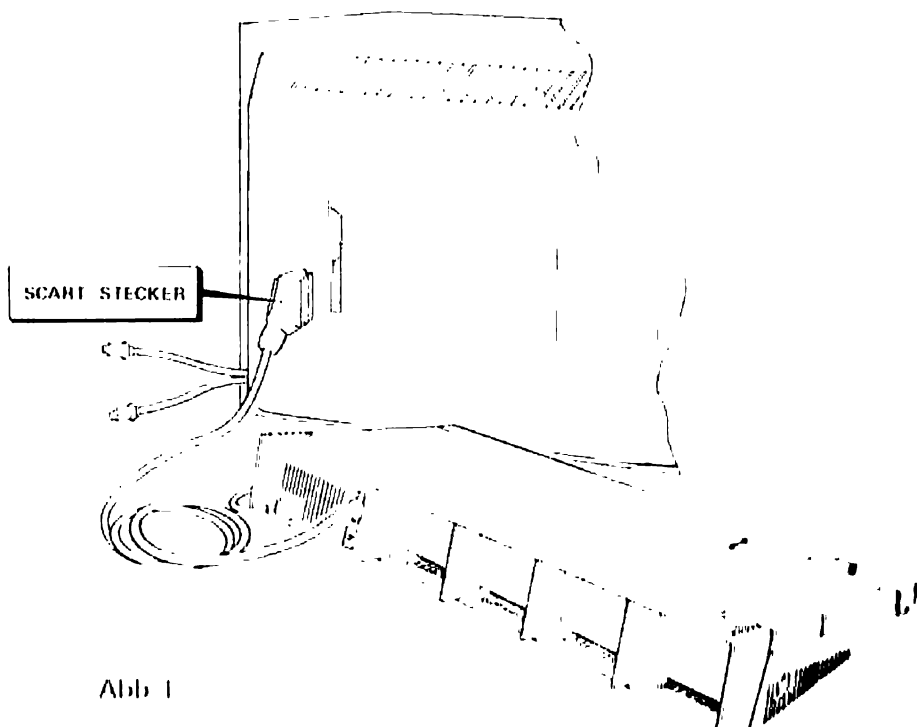
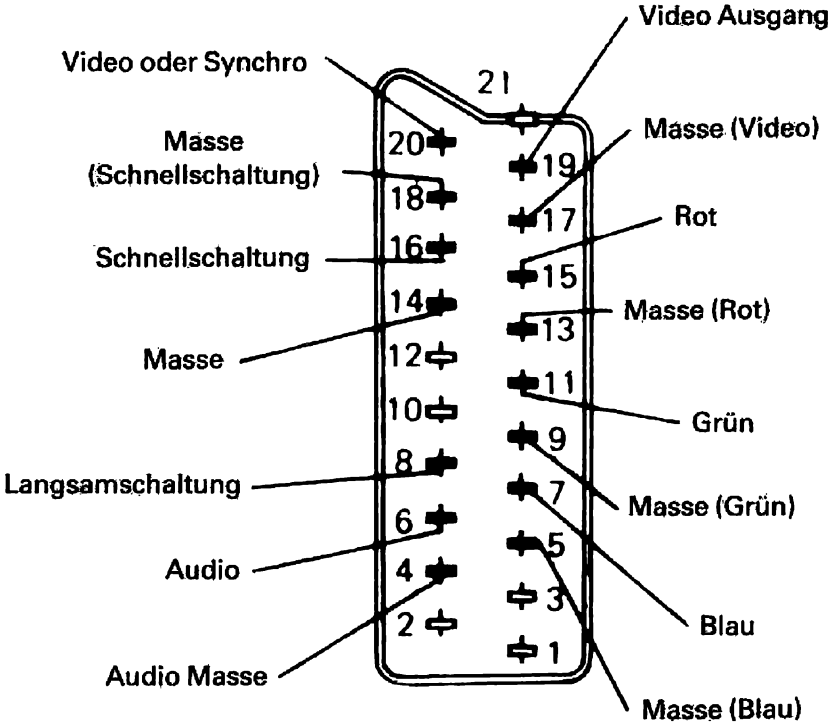


Abb. 1

Verpolung des Scart-Steckers
(Anwendungsseitig)



- über die AV-Buchse (DIN), in diesem Falle wird ein Zwischenkabel mit DIN-Steckern benötigt (Abb.3);

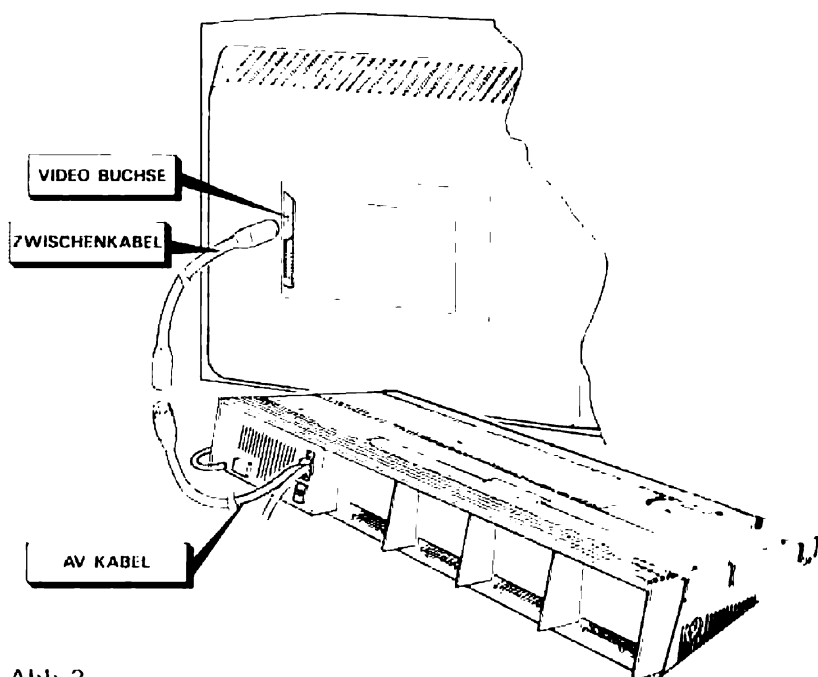
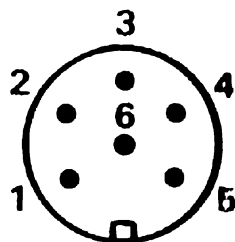


Abb.3

Kontakte der DIN-Buchse
(Von der Steckerseite aus gesehen)

- 1.5 +12 V
- 2 Video-Ausgang
- 3 Masse
- 4 Audio-Ausgang



– oder mit dem Antennenstecker (Abb.4).

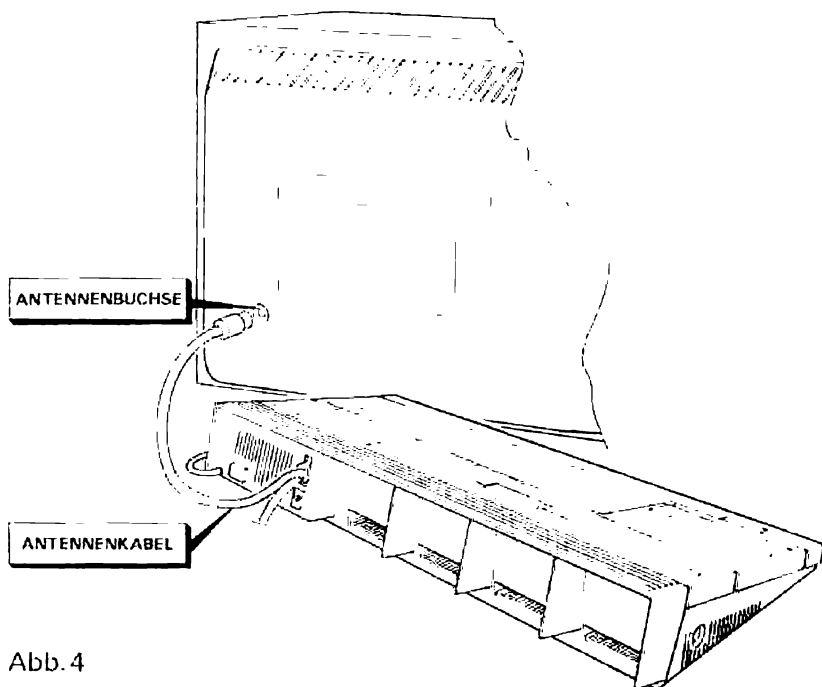


Abb.4

Inbetriebnahme

Sie sollten die einzelnen Geräte Ihrer Anlage in der folgenden Reihenfolge einschalten:

- a) Den Fernsehempfänger oder Monitor,
- b) die verschiedenen Peripheriegeräte
(Drucker, Diskettenlaufwerke etc.),
- c) die Zentraleinheit.

Für den Anschluß an Ihr Fernsehgerät oder Ihren Monitor bieten sich mehrere Möglichkeiten, da diese Geräte mit unterschiedlichen Anschlußmöglichkeiten versehen sein können. Da der TO7-70 in zwei verschiedenen Versionen lieferbar ist, ist ein Anschluß auf eine der folgenden Arten auch an Ihr Sichtgerät problemlos möglich:

*** Die SCART-Version des TO7-70.**

Diese Version sollten Sie wählen, wenn Ihr Fernsehgerät oder Ihr Monitor über eine Euronorm- oder SCART-Buchse verfügt.

Nach dem Einschalten der Zentraleinheit erscheint sofort das Menü des TO7-70. Wird ein Fernsehgerät verwendet, so schaltet dieses automatisch vom normalen Fernsehempfang auf den SCART-Eingang um. Durch die direkte RGB-Ansteuerung des Sichtgerätes ergibt diese Anschlußart die beste Bildqualität. (Abb. 1)

*** Die UHV/VIDEO-Version des TO7-70.**

Diese Ausführung ist für Sie richtig, wenn Sie den TO7-70 an einem Farbmonitor oder einem Fernsehgerät mit Video-Eingang betreiben möchten. Bei Ihrem Fachhändler erhalten Sie das richtige Verbindungskabel, um den DIN-Videoausgang des TO7-70 mit dem entsprechenden Eingang Ihres Monitors oder Fernsehgerätes zu verbinden. Manche Fernsehgeräte schalten automatisch auf den Videoeingang um, andere müssen von Hand entsprechend programmiert werden. Die Bedienungsanleitung Ihres Fernsehgerätes gibt Ihnen über die in Ihrem Fall erforderlichen Maßnahmen Auskunft. (Abb. 3)

- * Sollte Ihr Fernsehgerät weder über einen Video- noch AV-Eingang verfügen, so ist das lange HF-Kabel Ihres TO7-70 an den Antennen-Eingang des Fernsehgerätes anzuschließen. In diesem Falle empfängt der Fernseher das Bild des Computers wie einen normalen Fernsehsender. Stellen Sie das Empfangsgerät des Gerätes nach den Angaben der Bedienungsanleitung auf den Kanal 36 im UHF-Bereich ein und korrigieren Sie die Feinabstimmung, bis Sie ein scharfes, farbiges Bild und einen rauschfreien Ton erreichen. (Abb. 4)**

In allen Fällen ist zu bemerken, daß Sie Kontrast und Helligkeit Ihres Sichtgerätes auf das Computerbild einstellen müssen, da dieses allgemein wesentlich kontrastreicher als ein normales Fernsehbild ist. Auch die Lautstärke kann bei Computerbetrieb in der Regel zurückgenommen werden.

– **Anschluß an die Netzspannung**

Stecken Sie das Netzkabel Ihres Mikrocomputers in einen Netzstecker 220 V, 50 Hz.

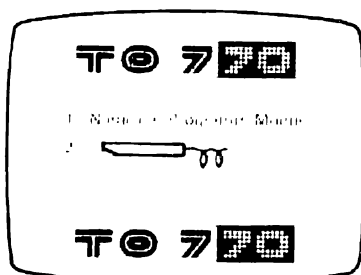
– **Anschluß des Programmrekorders**

Stecken Sie den DIN-Stecker Ihres Programmrekorders in die Buchse Nr. 10 (Abb. Seite 9) Ihres Mikrocomputers und sein Netzkabel in einen Netzstecker 220 V, 50 Hz.

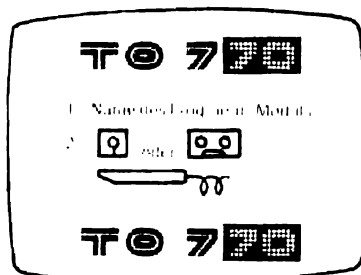
– **Anschluß der Disketteneinheit**

Schieben Sie das Interface an der Rückseite Ihres Mikrocomputers ein und stecken Sie das Netzkabel in einen Netzstecker 220 V, 50 Hz.

Auf dem Bildschirm erscheint das Menü des TO7-70.



Ohne Peripheriegerät



Mit Peripheriegerät

Es enthält eine Liste der Ihnen gebotenen Möglichkeiten:

- 1 - Name des eingelegten Programm-Moduls (siehe Seite 19).
- 2 - Piktogramm, das ein angeschlossenes Peripheriegerät darstellt:
 - entweder : Programm-Rekorder,
 - oder : Disketten-Station.

Sind beide Geräte angeschlossen, so werden die Disketten-Stationen vorrangig erkannt.

- 3 - Das Piktogramm des Lightpens stellt den in Ihrem TO7-70 eingebauten Lightpen dar.

Starten des TO7-70

Je nach der von Ihnen auf dem Titelbild getroffenen Wahl befehlen Sie die Ausführung der mit dem Symbol verbundenen Aktion.

Die Wahl kann getroffen werden:

- entweder durch Eingabe der gewählten Nummer über die Tastatur
- oder durch Antippen des neben der gewählten Nummer befindlichen Quadrats mit dem Lightpen.

Konsequenz der getroffenen Wahl:

- 1 - Sie befehlen die Ausführung des im eingelegten Programm-Modul befindlichen Programms. Für nähere Einzelheiten ist die Dokumentation zum Programm-Modul einzusehen.
- 2 - Sie wünschen die Ausführung einer auf Kassette oder Diskette gespeicherten Software:
 - auf Kassette: Es wird das erste Programm ausgeführt, das gelesen wird,
 - auf Diskette: Es wird das «AUTO.BAT»-Programm ausgeführt.

VORSICHT: Die Ausführung des Programms findet nur statt, wenn es in einer dem Programm-Modul entsprechenden Sprache geschrieben ist.

- 3 - Sie verlangen die Einstellung des Lichtgriffels (Lightpen) (siehe nächstes Kapitel).

Der Lightpen

Der Lightpen und sein Anschlußkabel befinden sich in einem Fach oberhalb der Tastatur. Wird der Lightpen benutzt, ist die Klappe offen. Rechts kann man den Lightpen in einem Halter absetzen.

Der Lightpen erhöht die Möglichkeiten Ihres Computers; er ermöglicht Ihnen zu zeichnen, Informationen auf dem Bildschirm Ihres Fernsehgerätes anzuwählen und einen richtigen Dialog durchzuführen.

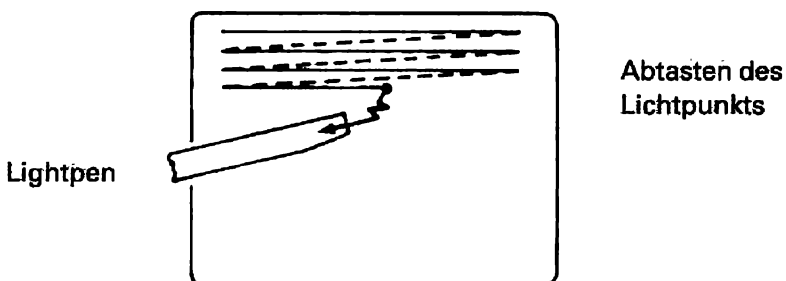
Beim Umgang mit dem Lightpen muß man zwei Funktionen unterscheiden:

1. - das Lokalisieren.
2. - das Gültigmachen (Quittieren).

Lokalisieren

Das vom Computer erzeugte Fernsehbild entsteht technisch dadurch, daß ein Elektronenstrahl die Leuchtschicht der Bildröhre zeilenweise von oben nach unten beschreibt.

Beim Lokalisieren meldet der lichtempfindliche Sensor im Lightpen (Phototransistor) dem Computer den Lichtpunktdurchlauf des Elektronenstrahls. Aus dieser Angabe berechnet der TO7-70 die relative Bildschirmposition, auf die der Lightpen gerade gerichtet ist.



Eine zu geringe Helligkeit des Fernsehgerätes kann ein unsicheres Arbeiten des Lightpens zur Folge haben. Außerdem werden Werte schwerer erfaßt, wenn ein schwarzer Bereich angesprochen wird.

Gültigmachen

Das Gültigmachen wird durch einen in der Spitze des Griffels angebrachten Schalter gewährleistet. Der Kontakt schließt sich, wenn der Griffel auf eine ebene Fläche, wie die des Fernsehbildschirms, gedrückt wird.

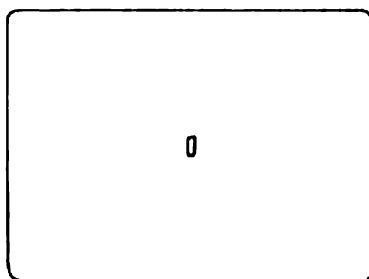
Unter Programmkontrolle ist es möglich, die Koordination des auf dem Bildschirm angezielten Punktes zu lesen und andererseits zu wissen, ob der Kontakt geschlossen ist oder nicht.

Der Lichtgriffel kann infolgedessen bei dialogausgerichteten Anwendungen äußerst nützlich sein: Entwerfen, Zeichnungen, Spiele und computergestützter Unterricht.

Einstellen des Lightpens

Anhand eines Einstellbildes kann der Computer den Lightpen an alle Fernsehgeräte anpassen. Vor allem gewährleistet das Einstellbild eine korrekte Synchronisierung zwischen der durch den Lightpen gelieferten Lichtpunktinformation und dem Abtastsignal des Fernsehgerätes.

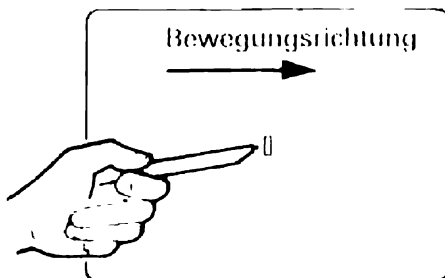
Die Funktion zum Einstellen des Lightpens wird beim Einschalten des Computers oder im Anschluß an eine Initialisierung über das Titelbild des **T07-70** angewählt.



Einstellbild
des Lightpens

Durchführung:

- Den Lightpen mit leichtem Druck an dem weißen Strich vorbeibewegen; ein «bip»-Signal ertönt, und das Titelbild erscheint von neuem: der Lightpen ist eingestellt.



ANMERKUNGEN:

- Der Lightpen muß erneut eingestellt werden, wenn der Computer ausgeschaltet war.
- Es wird empfohlen, den Lightpen senkrecht zum Bildschirm zu halten, um Lesefehler zu vermeiden.

Ablage des Lightpens

Besondere Aufmerksamkeit muß der Ablage des Lightpens an der für ihn vorgesehenen Stelle gewidmet werden.

Es muß darauf geachtet werden, daß der Schalter frei bleibt.

Das Programm-Modul

Beim Einlegen oder Herausnehmen eines Programm-Moduls muß der TO7-70 immer ausgeschaltet sein. Das Programm-Modul wird mit dem Etikett nach oben in das Fach gelegt.

Ihr Mikrocomputer funktioniert nur, wenn ein Programm-Modul eingelegt ist. In diesen einlegbaren Modulen sind nämlich die verschiedenen Spiel-, Haushalts- und Ausbildungshilfsprogramme sowie die Sprachen, mit deren Hilfe Sie Ihre Programme selbst schreiben können, untergebracht.

Sie müssen in folgender Weise vorgehen (Abb.5):

- Vergewissern Sie sich, daß Ihr Mikrocomputer ausgeschaltet ist,
- heben Sie die Blockierung auf, indem Sie Schalter (4) nach links schieben,
- drücken Sie Taste (3) in Richtung des Pfeils (A), die Klappe geht auf,
- legen Sie das Modul mit dem Etikett nach oben ein, so wie es in der Abbildung gezeigt wird,
- schließen Sie Klappe (2) wieder und drücken Sie, bis sie einrastet,
- schieben Sie Schalter (4) nach rechts, um die Taste zu blockieren.

Möchten Sie das Modul herausnehmen, gehen Sie in der umgekehrten Reihenfolge vor, nachdem Sie den TO7-70 ausgeschaltet haben.

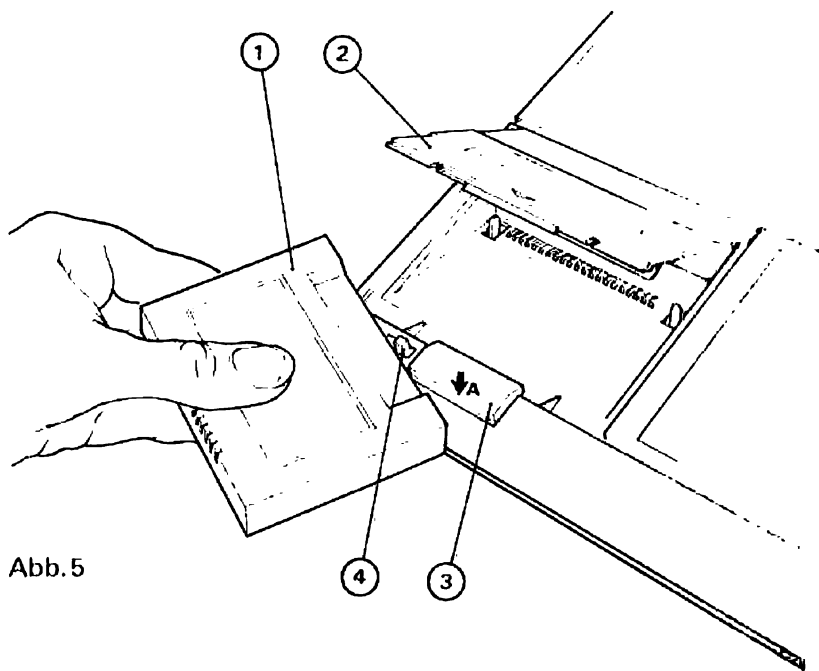


Abb.5

Der Programm-Rekorder

Um den Programmrekorder einsetzen zu können, muß im TO7-70 ein Programm-Modul liegen.

- Um ein auf Kassette gespeichertes Programm in Ihren TO7-70 zu laden, gehen Sie folgendermaßen vor:
 - Kassette in den Programmrekorder einlegen
 - auf die Wiedergabe-Taste am Programm-Rekorder drücken
 - Taste 2 drücken oder das weiße Quadrat vor der 2 mit dem Lightpen antippen
 - warten Sie, bis das Programm in den Speicher des TO7-70 geladen ist, und befolgen Sie dann die der Kassette beigefügten Anweisungen.
 - Sie werden feststellen, daß Sie während des Ladevorgangs bei manchen Kassetten gleichzeitig Musik oder Kommentare hören können.
- Um ein Programm Ihres TO7-70 auf Kassette zu speichern,
 - Befolgen Sie bitte die entsprechenden Anweisungen Ihres

Die Tastatur

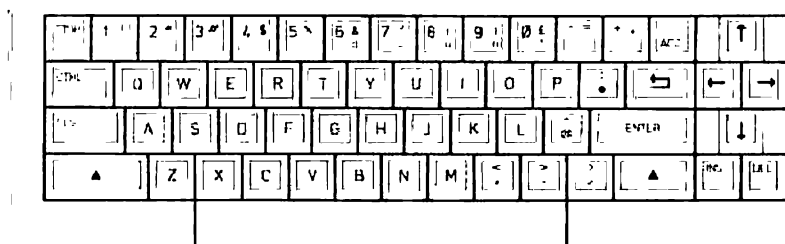
Die Tastatur dient vor allem zum Eingeben von Daten und Befehlen.

Sie enthält:

- 14 Funktionstasten;
- 43 Tasten für alphanumerische Zeichen und Sonderzeichen;
- eine Leertaste.

Bei jedem Druck auf eine Taste wird ein «bip»-Signal an den Tonkanal des Fernsehgerätes übertragen.

Längeres Drücken einer Taste bewirkt nach einer Sekunde das automatische Umschalten auf Dauerfunktion mit 10 Wiederholungen pro Sekunde.



Funktion der verschiedenen Tasten:

CTRL

: Diese «**CONTROL**»-Taste wird gleichzeitig mit einer anderen Taste gedrückt und ändert damit deren Funktion. (Sie findet ihre besondere Anwendung bei **BASIC**-Programmen.)

▲

GROSSBUCHSTABEN

Diese Taste wird gleichzeitig mit einer anderen Taste gedrückt und ändert damit deren Funktion.

▲	und Leertaste : Schaltet von der Großbuchstaben-schreibweise in die Kleinbuchstaben-schreibweise und umgekehrt.
▲	und Taste mit 2 Zeichen : Gibt die grünen Symbole ein (unabhängig von der Groß- oder Kleinbuchstaben-schreibweise).
▲	und alpha-numerische Taste : Erzeugt einen Großbuchstaben bei Kleinbuchstaben-schreibweise.

HINWEIS: Die beiden Tasten | ▲ | (rechts und links auf der Tastatur) haben dieselbe Funktion.

| ENTER | : Das Betätigen dieser Taste ermöglicht die Gültigmachung eines Befehls oder einer Programmzeile und anschließende Anwendungskontrolle.

Leertaste : Diese Taste bewirkt, daß eine Zeichenstelle leer bleibt. Sie wird benutzt, um Wörter oder Ausdrücke zu trennen.

| ACC | : **AKZENT**
Mit dem Drücken dieser Taste leitet man die Wahl besonderer Zeichen oder Symbole, akzentuierter Kleinbuchstaben und der Umlaute ein. Anschließend muß die für das gewünschte Zeichen vorhandene Taste gedrückt werden. Bei Spalten hat diese Taste Sonderfunktionen, die in den Anleitungen der betreffenden Programme erläutert sind.

A - Tastenbetätigung in 2 Schritten:

- Drücken der Taste
- Drücken der Taste, die dem einzugebenden Zeichen entspricht.

- Tastenbetätigung in 2 Schritten. Diese Liste beschränkt sich auf:

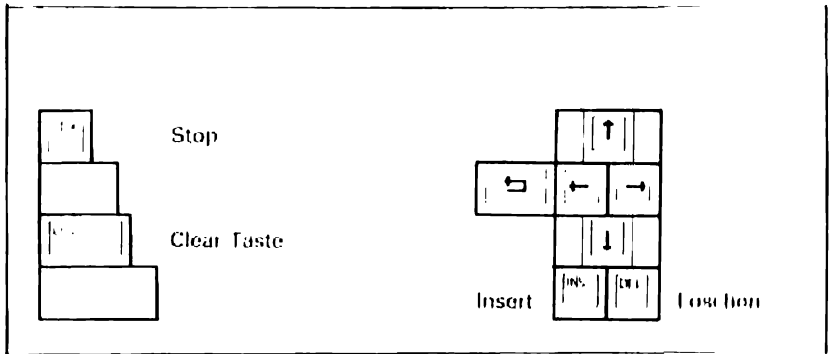
Einzugeb. Zeichen	1 Drücken der Taste	2 Drücken der Taste
a		[6]
o	[ACC]	[7]
u		[8]
n		[9]

B - Tastenbetätigung in 3 Schritten:

- Auf Kleinbuchstabenschreibweise übergehen, indem gleichzeitig die **Leertaste** und  gedrückt werden.
- Die Taste **[ACC]** drücken.
- Die dem einzugebenden Zeichen entsprechende Taste drücken.
- Die dem einzugebenden Buchstaben entsprechende Taste drücken.

- Tastenbetätigung in 3 Schritten. Siehe nachstehendes Beispiel:

Einzugeb. Zeichen	Schreib- weise	Drücken der Taste	2 Simult. Drücken der Tasten	3 Drücken der Taste
a			[] [2]	[A]
o			[] [2]	[O]
u	Klein- buch- staben	[ACC]	[] [2]	[U]
n			[] [2]	[N]
Buchstabe mit Akzent			Akzent	Klein- buchstabe

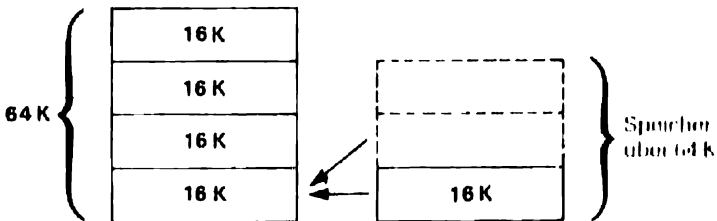


Der Speicher

Organisationsprinzip

Beim **T07-70** wird durch einen technischen Trick ein Speicher größer als 64 K adressiert. Das Prinzip: der Speicher, der den 64 K Adreßraum überschreitet, überlagert bestimmte Teile des Basis-speichers. Für den Computer entspricht der verwaltete Speicher stets 64 K, jedoch handelt es sich physisch nicht immer um denselben Speicherteil.

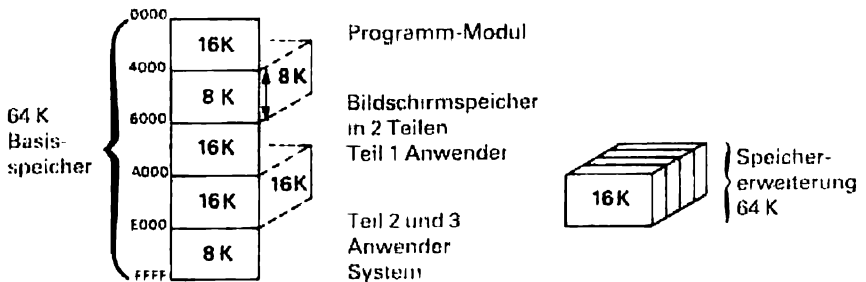
Diese Überlagerung wird beim **T07-70** pro Speicherblock vorge-nommen (wobei ein Speicherblock 16 K entspricht).



Betrieb

- Für die Benutzung des Speichers über 64 K genügt folglich die Angabe des Speicherblocks, der fiktiv anstelle eines anderen Basisspeicherblocks gesetzt wird.
- Der Anwender-Speicher des **T07-70** enthält 48 K in drei Teilen:
 - Teil 1** → 16 K, die in den 64 K des Basisspeichers fest enthalten sind,
 - Teil 2** → 16 K, die in den 64 K des Basisspeichers fest enthalten sind. Dieser Teil wird fiktiv von den anderen überlagert.
 - Teil 3** → Es handelt sich um einen lediglich bei Überlagerung nutzbaren Speicherblock von 16 K Byte.

Speicher-Schema des T07-70



Benutzung

Der Teil des Anwenderspeichers (Teil 3) zwischen 32 und 48 K ist bei Überlagerung ansprechbar, unter der Voraussetzung, daß diese Überlagerung durch Programmierung gesteuert wird.

Die Benutzung dieses Speichers ist demzufolge mit dem auf dem Programm-Modul enthaltenen Programm verbunden.

Speichererweiterung

Die Speichererweiterung (unter den Erweiterungen dargestellt) besteht aus 4 Speicherblöcken von je 16 K. Für ihre Benutzung wird der Teil 2 des Anwenderspeichers überlagert.

Erweiterungen und Peripheriegeräte

Wie beim Einlegen eines Programm-Moduls ist es **UNBEDINGT** erforderlich, den Computer am Netzschalter auszuschalten, bevor ein E/A-Erweiterungs-Modul oder ein Peripheriegerät angeschlossen oder herausgenommen wird.

Jedes von unserem Händler verkaufte Programm enthält auf seiner Verpackung die Liste der Erweiterungen, die erforderlich sind oder als Option angeboten werden, um die von diesem Programm gebotenen Möglichkeiten voll zu nutzen.

Die für Ihren Computer vorgesehenen Erweiterungen, die bei Ihrem Fachhändler erhältlich sind, erhöhen seine Leistungsfähigkeit.

Schnittstellenadapter

Dieser enthält einen seriellen sowie einen parallelen Anschluß.

Er ist für den Anschluß der Thomson-Drucker notwendig.

Speichererweiterung

Zur Erweiterung der Speicherkapazität Ihres Computers existiert ein Einsteckmodul, das an die 26polige Kontaktleiste, die sich an der Rückseite der Zentraleinheit befindet, angeschlossen werden kann und die Kapazität des Anwenderspeichers von 48 000 Byte auf 112 000 Byte (112 K RAM-Speicher) erhöht.

HINWEIS: Diese Erweiterung macht besondere Benutzeraktionen erforderlich. Das Adressierungsprinzip der Speichererweiterung ist identisch mit der Adressierung des Teils 2 und 3 des Anwenderspeichers. Die dementsprechende Beschreibung ist im Kapitel **SPEICHER** enthalten.

Die Benutzung dieser Erweiterung ist mit besonderen Aktivitäten der Programme verbunden.

Drucker:

Mit Druckern können Sie die von Ihnen geschriebenen Programme oder die Ergebnisse Ihrer Programme auf dem Papier «aufbewahren».

Bei unserem Vertragshändler sind verschiedene Druckermodelle mit unterschiedlichen Leistungsmerkmalen erhältlich, sodaß Sie eine Ihren Bedürfnissen entsprechende Wahl treffen können:

- grafikfähiger Matrixdrucker, 80 Spalten;
- Matrixdrucker, 40 Spalten.

ANMERKUNG: Die sich auf die Steuerung eines Druckers beziehenden Hinweise, die jedem Programm-Modul eigen sind, sind in den beigelegten Anweisungen enthalten.

Programm-Rekorder

Die leichte Anwendung, das schnelle Arbeiten und die Zuverlässigkeit des Programm-Rekorders machen ihn unentbehrlich, um Ihre Programme zu sichern oder um die ggf. von Ihnen gekauften Programme* in den Computer zu laden oder um Daten zu speichern.

Eine zweite Spur ist für Tonwiedergabe reserviert.

ANMERKUNG: Nur die Programm-Rekorder mit der entsprechenden Kennzeichnung können mit Ihrem Computer **TO7-70** benutzt werden.

* Nur die Programme mit dem Vermerk **TO7** oder **TO7-70** können von Ihrem Computer gelesen werden.

Disketten-Laufwerk mit Controller

Das Disketten-Laufwerk mit Controller ermöglicht die Aufzeichnung von Programmen, wie beim Programm-Rekorder, jedoch auf viel raschere und anpassungsfähigere Weise. Ebenso ist es möglich, Daten (Zahlen-, Wortliste...) zu speichern und zu verwalten, die mengenmäßig sehr viel umfangreicher sind, als jene des Zentralspeichers, und dies flexibler als auf dem Programm-Rekorder.

Es ermöglicht Ihnen auch, auf Disketten aufgezeichnete Programme zu benutzen, vor allem die Diskette **DOS BASIC des T07-70**, die das **BASIC-Programm-Modul** durch Hinzufügung neuer Anweisungen und Funktionen (ebenso wie bei **FORTH, LOGO...**) leistungsfähiger macht.

Musik- und Spielerweiterung

Diese besteht aus einem Modul und zwei Steuerknüppeln. Ein Frequenzgeber ermöglicht, unter spezieller Programmsteuerung, über das Fernsehgerät einen Ton-Synthesizer (ein- bis vierstimmig) zu betreiben.

Je nach Neigung des Steuerknüppels werden verschiedene Befehle generiert und an den Computer übermittelt, der sie interpretiert und ausführt. Ein Action-Knopf ermöglicht, je nach Art und Beschaffenheit des Spielprogramms, die Auslösung eines Schusses, den Wurf eines Balls, usw...

Mischbild-Erweiterung

Anhand dieses Moduls kann das Bild von **T07-70** mit jedem Video-Bild (Videorecorder, Video-Kamera, Fersehempfang) gemischt werden.

Die wichtigsten technischen Daten

Aufmachung

GEHÄUSE	: formgespritzter Kunststoff.
TASTATUR	: mechanische Tastatur
ABMESSUNGEN	: L. 455 - H. 75 - T. 260 mm.
GEWICHT	: 2,6 kg

Elektrische Merkmale

NETZANSCHLUSS	: Netzspannung 220 V-50 Hz.
STROMVERBRAUCH	: 20 W (ohne Erweiterung) 35 W (mit sämtlichen Erweiterungen).
SICHERUNG	: T 800 mA.

Ein unter dem Gerät angebrachtes Schild enthält die oben erwähnten elektrischen Merkmale.

Technische Merkmale

MIKROPROZESSOR	: 6809E-1 MHz.
SPEICHER 70 K	: Anwender: 48 K RAM erweiterbar auf 112 K RAM Graphik-Bildschirm: 16 K RAM Monitor: 6 K ROM
STROMVERSORGUNG	: + 5 V = 1,8 A + 12 V = 300 mA – 5 V = 100 mA.
BILDSCHIRM	: 320×200 Punkte - 16 Farben 25 Zeilen×40 Zeichen Groß- und Kleinbuchstaben.
TASTATUR	: 57 QWERTY-Tasten.
MUSIK	: Tongenerator/5 Oktaven.
LIGHTPEN (LICHTGRIFFEL)	: Auflösung 320×200 Punkte.
PROGRAMM-REKORDER-SCHNITTSTELLE	: 900 Baud, automatische Motorsteuerung Tonkanal auf der 2. Spur.
ANTENNENSTECKER (CU77-PBG)	: – UHF-Ausgang, Kanal 36
DIN-BUCHSE (CU77-PBG)	: – Video-Signal 1 V _{ss} 75 Ω – Audio-Signal 1 V _{ss} -Impedanz > 10.
EURO-NORM-ANSCHLUSS (CU 77-PER)	: RGB-Ausgänge + Ton.

Funktionsfehler und Prüfungen

FUNKTIONSFEHLER	PRÜFUNGEN
Die Kontrolllampe «EIN/AUS» leuchtet nicht beim Einschalten	– Ist die Netzspannung (220 V) vorhanden und sitzt der Stecker fest in der Steckdose? – Netzkabel und Netzanschluß prüfen.
Das Titelbild erscheint nicht auf dem Bildschirm des Fernsehgeräts	– Die Einstellungen des Fernsehgeräts überprüfen – Die Verbindungen zu den Euro-Norm-Anschlüssen des Fernsehempfängers prüfen.
Der Name des Programm-Moduls erscheint nicht auf dem Bildschirm	– Die Klappe des Programm-Moduls ist nicht richtig eingerastet. – Eine Initialisierung des Programms vornehmen
Schwarze Balken oder schlechte Bildsynchronisierung. Die Tastatur ist gesperrt. Die Zentraleinheit spricht auf keine Steuerung an	– Netzstörungen oder benachbarte Magnetstrahlung. – Sicherstellen, daß das Fernsehgerät oder ggf. die Peripheriegeräte vor der Inbetriebnahme der Zentraleinheit eingeschaltet wurden. – Eine Initialisierung des Programms vornehmen.
Kein Ton	– Das «BIP»-Tonzeichen der Tastatur muß zu hören sein; den Lautstärkeregler des Fernsehgerätes prüfen. – Die Verbindungen zu den Euro-Norm-Anschlüssen des Fernsehempfängers prüfen.
Schlechter Betrieb des Programm-Rekorders nach dem Anschluß an die Zentraleinheit	– Prüfen ob der Programm-Rekorder funktioniert, ohne an den Computer angeschlossen zu sein. – Seine Anschluß-Schnur oder den Anschluß an die Buchse «PROGRAMM-REKORDER» prüfen. – Die Beschaffenheit der in das Programm Modul Fach eingeschobenen Software kontrollieren.
Der Programm-Rekorder lädt kein Programm in den Hauptspeicher oder zeichnet weder Daten noch Programme auf	– Die Steuerungen des Programm-Recorders prüfen. – Die Bedienungs- und Softwareanleitungen einsehen, aus denen die Verwaltung dieses Peripheriegerätes hervorgeht.
Ungenügende Empfindlichkeit des Lightpens	– Die Helligkeitsregelung des Fernsehgeräts prüfen und die Einstellung des Lichtgriffels vornehmen – Sicherstellen, daß der mit dem Lightpen angesprochene Bereich nicht schwarz ist
Blockieren der Tastatur	– Prüfen, ob der Lichtgriffel korrekt an seinem Platz liegt und ob der Taster nicht gedrückt ist

Page blanche

2. Wie Ihr T07-70 funktioniert

I Ihr Computer

Schalten Sie ihn ein, und er steht zu Ihrer Verfügung!

Die Tastatur ermöglicht Ihnen, Befehle zu erteilen, und der Bildschirm zeigt, daß der Computer Sie verstanden hat und Ihnen gehorcht.

Ihr Computer ist eine Maschine.

Zum einen eine Schreibmaschine, bei der das Papier durch den Bildschirm ersetzt wurde, zum anderen eine superschnelle Rechenmaschine, mit der Sie aber nicht nur rechnen, sondern auch zeichnen, komponieren, und spielen können.

Unter der Tastatur und hinter dem Bildschirm beginnt der Bereich der Elektronik, der Informatik und der Technik.

Ihr Microcomputer ist ein elektrisches Gerät. Es hat die Aufgabe, Informationen in Form von elektrischen Impulsen zu verarbeiten.

Der interne Aufbau, der die Verarbeitung und Logik seiner Organisation ermöglicht, ist sehr komplex. Wir wollen hier keine Bestandsaufnahme davon machen, vielmehr wollen wir zusammenfassend die großen Prinzipien darstellen, die das wichtigste Rüstzeug für die automatische Verarbeitung der Information sind.

Üblicherweise teilt man das Problem in zwei Teile auf! In den anfaßbaren und den logischen Teil, oder, wie es im Englischen heißt: in «hard» und «soft». In die elektronischen Bestandteile einerseits und die Programme, die die Maschine in den Dienst des Menschen stellen, andererseits. Es braucht nicht besonders betont zu werden, daß «Hardware» und «Software» nicht ohne einander auskommen, und daß Fortschritte des einen auch Fort-

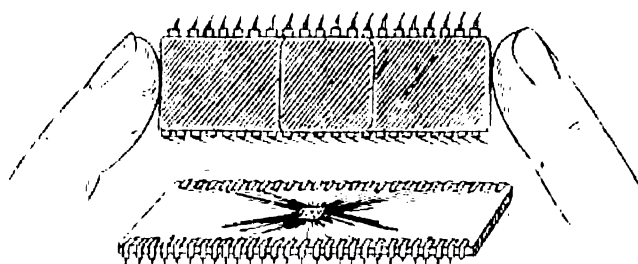
Page blanche

II Arbeitsprinzip

Der Mikroprozessor des T07-70

Jede Tätigkeit des Mikrocomputers wird durch einen auf diese Aufgabe orientierten integrierten Schaltkreis, dem Mikroprozessor, kontrolliert. Die Möglichkeiten der Maschine sind häufig identisch mit den Möglichkeiten des Mikroprozessors.

Der Mikroprozessor unterscheidet sich durch seine Größe; mit Hilfe von 40 «Beinchen» kann er sich mit den anderen, auf bestimmte Aufgaben spezialisierten Schaltkreisen in Verbindung setzen.



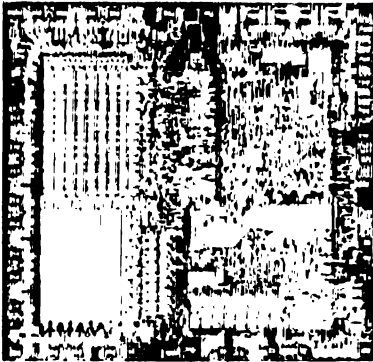
Der Mikroprozessor

Die Entwicklung des Mikroprozessors gehört zu den bedeutendsten Errungenschaften der modernen Elektronik. Der technische Fortschritt, den er ermöglicht, beruht auf seiner Schnelligkeit und

Integrationsdichte. Eine unglaubliche Anzahl (fast eine Million) elementarer elektronischer Schalter, die auf der winzigen Fläche von 25 Quadratmillimetern untergebracht sind, ermöglichen (dank kurzer Übertragungswege) eine enorme Verarbeitungsgeschwindigkeit.

Ein Mikroprozessor wird aus einer Siliziumscheibe hergestellt, wobei die Schaltkreise auf fotomechanischem Wege aufgedruckt werden. Diese erfordert eine hochspezialisierte Fabrikationstechnologie, denn ein herkömmlicher Transistor unterscheidet sich beträchtlich von einem gedruckten Transistor.

Nur eine Vergrößerung kann einen Eindruck von der Komplexität der Schaltkreise eines Mikroprozessors vermitteln:



Wie kann dieses kleine Plättchen Informationen verarbeiten?

Um diese Frage zu beantworten, müssen wir uns zunächst auf den folgenden Seiten etwas mit den Grundlagen der Informatik befassen:

III Der Transistor und die binäre Logik

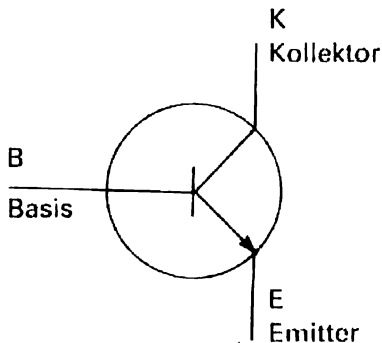
1) Was ist eigentlich ein Transistor?

Wie Sie später noch sehen werden, besteht der Mikroprozessor aus einer großen Anzahl elektronischer Schalteinheiten, die man «Transistoren» nennt. Um die Funktion eines Transistors einfach zu erläutern, bietet sich der Vergleich mit einem Wasserhahn an:

öffnet man den Hahn so fließt das Wasser,
schließt man den Hahn so fließt kein Wasser.

Der Transistor ähnelt dem Wasserhahn in Aufbau und Wirkungsweise sehr:

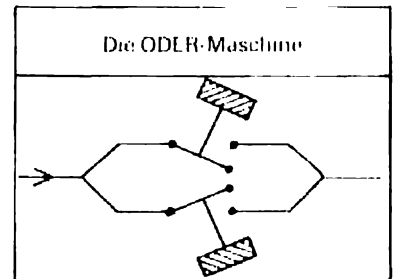
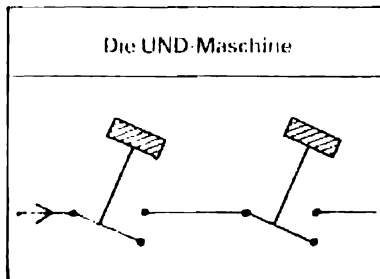
Der Regelknopf entspricht dem Basis-Anschluß,
das Wasserzulaufrohr entspricht dem Emitter-Anschluß und
das Wasserablaufrohr entspricht dem Kollektor-Anschluß.



Stellen Sie sich vor, daß ein Bit, zu verstehen als ein Elektronenbündel, am Punkt E erscheint. Werden an Punkt B keine besonderen Maßnahmen getroffen, der Hahn also nicht aufgedreht, so ist der Weg zum Punkt K gesperrt. Wird jedoch über Punkt B der Weg geöffnet, so gelangt Strom ungehindert durch den Transistor zum Anschluß K. Wir erkennen hieraus klar das Schalterprinzip, nach dem ein Transistor arbeitet.

2) Die binäre Logik

Durch geschicktes Zusammenschalten zweier Transistoren lassen sich zwei interessante «Maschinen» herstellen:



Die UND-Maschine läßt den Strom passieren, wenn der Schalter A **und** der Schalter B gleichzeitig geschlossen sind. Die ODER-Maschine läßt den Strom passieren, wenn der Schalter A **oder** der Schalter B geschlossen ist, außerdem dann, wenn die beiden Schalter A und B gleichzeitig geschlossen sind. In diesem Fall spricht man von einer nicht ausschließenden ODER-Maschine.

Nun kommt die große Überraschung:

Alle zur Transformation von Bitfolgen erforderlichen Operationen lassen sich mit Hilfe der ODER- und der UND-Maschine sowie dem «Operator» NICHT, der eine 0 in eine 1 umwandelt und umgekehrt, realisieren. Ein Mikroprozessor ist somit nichts weiter als ein System, das aus vielen Kombinationen dieser drei Elemente aufgebaut und an bestimmte Leitungspfade angeschlossen ist.

Tatsächlich genügt es, die genannten Elemente zu Schaltkreisen zusammenzusetzen, die nichts weiter zu tun haben, als zwei Binärwerte zu addieren. Das ist nicht sehr schwierig, reicht aber aus, um alle erforderlichen Rechnungen auszuführen:

- Jede komplizierte Formel läßt sich auf die vier Grundrechenarten zurückführen.
- Die Multiplikation ist eine Folge von Additionen, die Division eine Folge von Subtraktionen.
- Die Subtraktion ist eine «umgekehrte» Addition.
- Jede Addition besteht aus der mehrfachen Addition zweier Einzelwerte.

Jede Rechnung besteht im Grunde also nur aus der Addition einzelner Ziffern, und zu beachten ist nur die richtige Addition der «Zahl im Sinn», des sogenannten Überlaufs.

Es genügt somit, wenn der Computer im Binärsystem addieren kann. Er ist dann in der Lage, alle Rechnungen auszuführen!

Page blanche

IV Die binäre Codierung der Informationen

Unsere Erklärung basiert in Wirklichkeit auf dem Verständnis von zwei wesentlichen Punkten:

1. Jede Information kann durch eine (eventuell sehr lange) Folge der Werte 0 und 1 ausgedrückt (man sagt auch «codiert») werden.
2. Der Mikroprozessor hat die Aufgabe, diesen Folgen der Werte 0 und 1 in andere aus 0 und 1 bestehende Folgen umzuwandeln, d.h., er wandelt eine Eingangsinformation in eine Ausgangsinformation um.

1) Die Informationseinheit: das Bit

Am Anfang ist das Rohmaterial, der elektrische Strom. Die erste Aufgabe des TO7-70 bzw. des eingebauten Transformators, besteht nun darin, aus den angelieferten 220 V Wechselspannung 12 V Gleichspannung zu machen. Anschließend muß dieser gleichmäßige Stromfluß in einem genau festgelegten Raster durch ein «Uhr»-Element «zerstückelt» (getaktet) werden. Die Uhr des TO7-70 schlägt als «MegaHertz» d.h. 1 Million mal pro Sekunde (1 MHz).

Der elektrische Strom kann also als eine Folge von Impulsen angesehen werden, die durch die Magie der Wörter und die Macht der Einbildungskraft zu elementaren Informationseinheiten, englisch «bit» (binary digit), werden.

Ein Bit kann zwei Zustände annehmen: Es kann vorhanden sein oder nicht vorhanden sein, je nachdem, ob Strom fließt oder nicht. In einem ersten Schritt definieren wir das Vorhandensein eines elektrischen Impulses als 1 und dessen Nichtvorhandensein als 0. Die Herausforderung, die es anzunehmen gilt, lautet dann: Lässt sich jede Information durch eine Folge von 0 und 1 ausdrücken?

In unserer Welt und bei dem Gebrauch, den wir von ihr machen, stellt sich die Information meistens in drei Formen dar: als Ziffern, als Bilder und als Texte. Wir werden sehr bald aufzeigen, wie diese drei Formen der Information durch eine Reihe von 0 und 1, d.h. also in binärer Form, dargestellt werden können.

2) Die Zahlen

Es gibt in der Informatik ein System, das sogenannte Binärsystem, in dem Zahlen nur durch die Zeichen 0 und 1 ausgedrückt werden.

Es beruht auf dem gleichen Prinzip wie das Dezimalsystem, das wir gewöhnlich benutzen (da wir zehn Finger besitzen). Im Binärsystem spielt die Ziffer 2 die Rolle der Zahl 10 aus dem Dezimalsystem. Die Ziffer 2 wird binär durch die Zeichenfolge 1 und 0 dargestellt.

Um die Handhabung der Binärwerte zu erleichtern, werden die Bits jeweils in einer Gruppe von 8 zusammengefasst, die man ein Byte nennt.

Hier sehen Sie ein solches Byte:



In einem Byte hängt der Wert eines Bits von seiner jeweiligen Position ab:

- das erste Bit von rechts hat den Stellenwert 1
- das zweite Bit von rechts hat den Stellenwert 2
- das dritte Bit von rechts hat den Stellenwert 4
- das vierte Bit von rechts hat den Stellenwert 8
- usw.

Das dargestellte Byte repräsentiert damit also die Dezimalzahl:
 $32 + 8 + 4 + 1 = 45$

Die dezimalen Ziffern können auf diese Weise mit Hilfe einiger Byte dargestellt werden. Beim TO7-70 werden die ganzen Zahlen (von -32768 bis $+32767$) in 2 Bytes dargestellt, die reellen Zahlen in einfacher Genauigkeit (8 Ziffern) in 4 Byte, und die reellen Zahlen in doppelter Genauigkeit (16 Ziffern) in 8 Byte.

3) Die Texte

Da sich ein Text aus darstellbaren Zeichen zusammensetzt, genügt es, jedes dieser Zeichen durch eine Zahl zu codieren, was keine besonderen Probleme aufwirft, es sei denn das einer notwendigen internationalen Standardisierung. Es gibt einen solchen Standard, den ASCII Code (American Standard Code for Information Interchange). Er wird von fast allen Konstrukteuren und Planern von Computern respektiert. Jedes Zeichen wird in einem Byte dargestellt, welches dafür insgesamt 256 Möglichkeiten bietet.

Der TO7-70 verwendet die ersten 128 Möglichkeiten für seinen eigenen Gebrauch und reserviert die übrigen 128 für Benutzer, die ihre eigenen Zeichen entwerfen wollen. Der ASCII Code ist in Abschnitte geordnet:

- die alphabetischen Großbuchstaben von 65 bis 90
- die alphabetischen Kleinbuchstaben von 97 bis 122
- die Ziffern von 48 bis 57.

Die Werte von 1 bis 32 sind die sogenannten Kontrollzeichen: mit ihnen ist es unter anderem möglich, die Druckanordnung der Texte darzustellen: Übergang zur neuen Linie, Vorschub, Verstellung des Cursors, Blinken, etc.

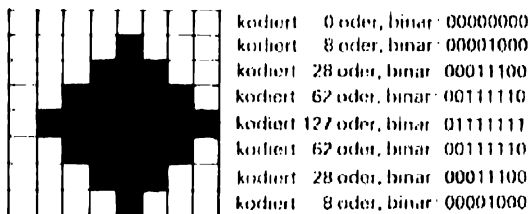
Die übrigen Werte betreffen die Interpunktionszeichen, mathematische Zeichen und Sonderzeichen (\$, £, à etc.)

4) Die Bilder

Um ein Bild auf eine Folge von 0 und 1 zu reduzieren, oder genauer gesagt, auf eine Folge von Bytes, muß man zunächst wissen, daß das Bild auf eine Anordnung von Punkten reduziert

wird. Es sei übrigens in diesem Zusammenhang darauf hingewiesen, daß die Informatik nichts neues erfunden hat: man denke nur an die Mosaik, an das Fernsehen, um nur einige Beispiele von Vorläufern zu nennen, die sich der Anordnung von Punkten bedienen.

Beispiel für ein Zeichen in einer 8×8 -Matrix:



Das Bild ist die meiste Zeit hindurch auf dem Fernsehapparat sichtbar. Jeder Punkt des Rasters wird Bildelement (picture element) genannt. Je nachdem, ob er leuchtet oder gelöscht ist, nimmt das entsprechende Bit den Wert 1 oder 0 an. Dieser Beschreibung der Form kann man eine Beschreibung der Farben hinzufügen, indem man das Bit eines jeden Bildelementes um Bits erweitert, die einer Farbe entsprechen. Handelt es sich um 8 Farben, so werden die Farben in 3 Bits dargestellt; handelt es sich um 16 Farben, so werden sie in 4 Bits dargestellt.

Das BASIC-Modul beispielsweise erlaubt 16 Farben: zu den 8 Primärfarben durch COLOR und SCREEN, 8 Pastellfarben durch

blau	grün	rot		Code
0	0	1	→ Rot	1
0	1	0	→ Grün	2
0	1	1	→ Gelb	3
1	0	0	→ Blau	4
1	0	1	→ Violett	5
1	1	0	→ Hellblau	6
1	1	1	→ Weiss	7

einen anderen Befehl. Diese Farbe erhält man durch das Mischen der drei Grundfarben: blau, grün und rot.

Wichtig ist letzten Endes, sich zu merken, daß ein Farbbild auf eine lange Folge von 0 und 1 zurückgeführt werden kann. Natürlich unter der Bedingung, daß diese Folge in gleicher Weise vom Mikrocomputer und vom Menschen, der sich seiner bedient, verstanden wird. Man kann sich leicht vorstellen, daß je größer die Zahl der Bildelemente auf dem Bildschirm ist und je mehr Farben es sind, die Bitfolge immer länger, das Bild selbst aber auch immer schöner wird.

Page blanche

V Der Mikroprozessor und die Speicher

Nachdem wir uns mit den Prinzipien des Codierens auseinandergesetzt haben, müssen wir noch verstehen lernen, wie das Werkzeug, d.h. der Mikrocomputer, es fertig bringt, die Pflichten zu erfüllen, die der Benutzer von ihm erwartet. Konkret heißt das, daß das Werkzeug in der Lage sein muß, eine bestimmte Anzahl von Operationen, mit diesem Rohmaterial, wie es die Information darstellt, durchzuführen, nämlich:

- erfassen
- verarbeiten
- speichern
- wiedergeben

In diesem Abschnitt untersuchen wir die Funktionen der Verarbeitung und Speicherung.

1. Der Mikroprozessor

Der Mikroprozessor des T07-70 ist der 6809E. Es ist ein 8-Bit-Prozessor, d.h. es handhabt ein Byte nach dem anderen. Seine Rolle besteht darin, den Informationsaustausch in der Zentraleinheit zu überwachen, bestimmte Aufgaben durchzuführen, wie etwa Operationen zwischen numerischen Daten oder die Verbindung mit der Außenwelt über die speziellen Schaltkreise zu lenken. Es ist bei weitem der aktivste Baustein der gesamten Anlage.

Im Innern des Mikroprozessors werden immer 2 Bytes verarbeitet. Der Mikroprozessor verfügt über 6 Register als interne Speicher, die vorübergehende Speicherung von Daten ermöglichen, bevor sie eventuell zu externen Speichern gesandt werden (siehe weiter unten). Er kann mit Hilfe von 59 Anweisungen gesteuert werden.

Die Speicher

Beim Speicher handelt es sich um integrierte Schaltkreise, sogenannte IC's (integrated circuits). Auf der Platine des Computers unterscheiden sie sich schon äußerlich vom Mikroprozessor; ihr Gehäuse ist viel kleiner. Aber auch ihr interner Aufbau ist völlig unterschiedlich.

Man kann sich einen Speicherbaustein wie einen Schrank mit sehr vielen Schubladen vorstellen. Jede Schublade hat eine Nummer, die sogenannte Adresse, und enthält ein Byte, dessen Inhalt durch den Mikroprozessor verändert werden kann.

Die Speicheradresse 58235
enthält das Byte 11010110

Da nicht alle Bytes in einen Schrank passen, sind sie auf mehrere verteilt, d.h., eine bestimmte Anzahl von Speicherbausteinen bilden den Speicher des Computers.

Die einzelnen Schubladen (Byte) des Speichers sind von 0 bis 65535 nummeriert.

Der Adressenbereich ($65536 = 2 \text{ hoch } 16$) wurde nicht willkürlich von einem Ingenieur festgelegt, sondern beruht auf den Möglichkeiten des Mikroprozessors.

Der 6809-E merkt sich die Adresse, mit der er arbeiten will, in Form von zwei Byte. Die höchste Adresse kann somit den Binärwert 1111111111111111 besitzen, und er entspricht genau dem Dezimalwert 65535. Man sagt auch, daß der 6809-E insgesamt 64 Kilobyte adressieren kann.

Betrachtet man die Angelegenheit etwas genauer, werden die Speicher dazu veranlasst, so unterschiedliche Dienstleistungen zu erbringen, daß man sich genötigt fühlt, eine Klassifizierung vorzunehmen. Glücklicherweise deckt sich diese Klassifizierung mit einigen technischen Gegebenheiten.

Die verschiedenen Speichertypen lassen sich durch drei Kriterien beschreiben:

- Speicherkapazität
- Möglichkeiten, sie zu lesen und zu beschreiben
- Zugriffszeit

Die Speicherkapazität wird in Kilo-Byte gemessen, die 1024 Byte entsprechen. ($2^{10} = 1024$, Kilo = 1000)

Die Zugriffszeit entspricht der Zeit, die der Speicher benötigt, um die Information, die er besitzt, dem Mikroprozessor wiederzugeben, und zwar von dem Augenblick an, wo er seine Speicherstelle erkennt.

Geht man von diesen technischen Gegebenheiten aus, so versteht man die Klassifizierung der Speicher in drei Kategorien besser, nämlich den programmierbaren Festwertspeicher (ROM), den Speicher mit direktem Zugriff (RAM) und den Massenspeicher.

Der ROM-Speicher

Abkürzung ROM (Ready Only Memory).

Der ROM des TO7-70 umfasst 6 Kilo-Byte bzw. 6144 Speicherzellen. Sein Inhalt ist unveränderbar, auch ohne Stromversorgung. Die Zugriffszeit ist sehr kurz.

Der ROM enthält den Monitor, das Betriebssystem. Dieses Programm wird immer aufgerufen, sobald der TO7-70 eingeschaltet wird.

Dieses von den Ingenieuren bei THOMSON entwickelte Programm ermöglicht es Ihrem Mikrocomputer außerdem, einfache Aufgaben durchzuführen:

- das Lesen der Tastatur
- die Anzeige auf dem Bildschirm
- die Kontrolle des Lightpens
- die Kontrolle des Programm-Rekorders
- das Erzeugen von Tönen etc.

Für jede dieser Aufgaben ist eine spezielle Routine im Monitor vorhanden. Sie können diese Routinen für Ihren eigenen Bedarf verwenden, unter der Voraussetzung, daß Sie wissen, wo sich die ROM-Adressen (Einsprungadressen) befinden und wie diese Routinen aufgerufen werden müssen. Eine ausführliche Beschreibung finden Sie am Ende dieses Handbuches.

In Anbetracht der wichtigen Rolle des Monitors für das Funktionieren des Mikrocomputers versteht man die Notwendigkeit, ihn in einem unveränderbaren Speicher unterzubringen, der nur gelesen werden kann, nur besser.

Ein weiterer ROM kann diesem in Form eines Programm-Moduls hinzugefügt werden. Davon wird noch die Rede sein.

Der RAM-Speicher

Abkürzung RAM (Random Access Memory).

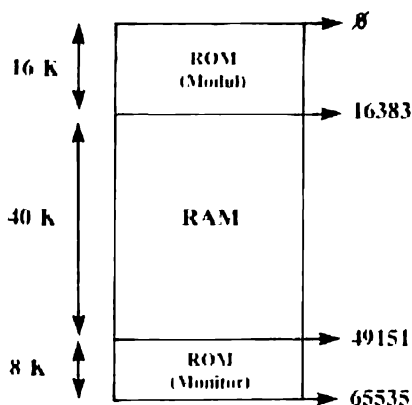
Ganz anders verhalten sich die Speicher mit direktem Zugriff, die aber wie die programmierbaren Festspeicher eine sehr kurze Zugriffszeit haben.

Der RAM steht dem Benutzer zur Verfügung, damit er Daten darin lagern kann. Ohne Stromversorgung ist der RAM leer. Schreibt man beispielsweise ein BASIC-Programm, so füllt man den RAM nach und nach mit Daten, dem Programm. Beim Ausschalten wird alles gelöscht.

In einem Speicher mit direktem Zugriff kann der Mikroprozessor nicht nur lesen, sondern auch schreiben. Damit der Benutzer bequem arbeiten kann, muß der RAM-Bereich sehr groß sein.

Der gesamte Speicher des TO7-70 unterteilt sich ganz natürlich in zwei Abschnitte: in ROM-Abschnitte und einen RAM-Abschnitt. Der Umfang des ROM gewährleistet das Arbeitsvermögen des Systems, und der Umfang des RAM gewährleistet das bequeme Arbeiten des Benutzers.

Das folgende Schema die Speicheraufteilung (Memory Map), des TO7-70. Eine ausführlichere Version findet sich auf Seite 108.



Die ersten 8 Kilo-Byte des RAM sind für den Bildschirmspeicher reserviert (siehe Kapitel Bildschirmspeicher).

Der Massenspeicher

Die Flüchtigkeit des RAM ist offensichtlich eine seiner Schwachstellen. Wird der Mikrocomputer nicht mehr mit Elektrizität versorgt, so verschwindet sein Inhalt vollständig und unverzüglich. Wenn es sich um Ergebnisse einer langen Arbeit handelt, ist dies sehr ärgerlich.

Es ist daher unbedingt erforderlich, über diesen RAM hinaus noch über eine zusätzliche Speicherkapazität zu verfügen, der die folgenden Vorteile bieten würde:

- großes Speichervermögen
- nicht flüchtig
- Möglichkeit des Speicherns und des Bewegens von Daten

Es geht also darum, für einen geringen Preis einen unterstützenden ständigen Speicher von hoher Leistungsfähigkeit zu haben, der sowohl zum Beschreiben als auch zum Lesen zur Verfügung steht. Diese Ziele können durch zwei gebräuchliche Geräte verwirklicht werden:

- die Kassette
- die Diskette

Man kann mühelos Dutzende von Kilo-Byte in eine gewöhnliche Audio-Kassette und Hunderte von Kilo-Byte auf einer Diskette lagern. Man kann sich also eine persönliche Programmbibliothek anlegen, die man entweder im Handel erwirbt oder die das Ergebnis der eigenen schöpferischen Arbeit ist.

Diese Vorteile kosten etwas in bezug auf die Zugriffszeit. Der Nachteil ist besonders schwerwiegend für die Kassette, deren sequentieller Ablauf Zugriffszeiten von mehreren Minuten zur Folge haben kann.

Das Diskettensystem bietet, verglichen mit dem Kassettensystem, mehrere Vorteile: Der Zugriff ist direkt und die Zugriffszeit wird auf einige Zehntelsekunden gesenkt.

Füllen des Speichers

Um den Speicher des TO7-70 zu füllen, müssen Bytes in die Speicherzellen geschrieben werden. Dafür sind zwei Personenkreise zuständig:

- die Konstrukteure des Computers
- die Benutzer des Computers

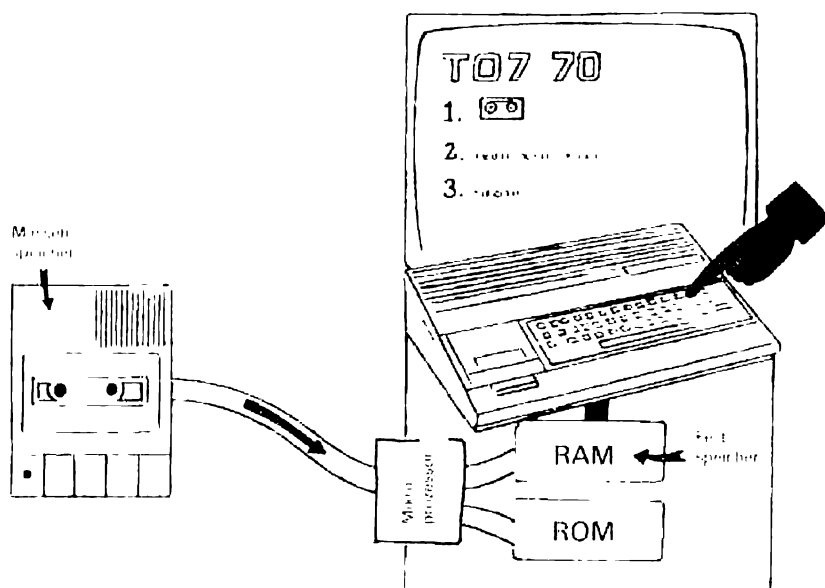
Oder genauer gesagt:

- die Ingenieure bei Thomson
- Sie

Der ROM ist ein für allemal von den Entwicklern des TO7-70 gefüllt worden. Der RAM hingegen steht Ihnen zur Verfügung und wird meistens teilweise von Ihnen gefüllt werden.

Die im RAM enthaltenen Byte stellen im allgemeinen ein Programm und seine Daten dar. Der einfachste Weg den Speicher zu

füllen, ist, den Inhalt eines Massenspeicher-Systems zu übertragen. Im Falle des T07-70, der mit einem Programm-Rekorder verbunden ist, genügt es, eine 2 zu tippen, nachdem Sie das System angeschaltet haben.



Übertragung eines Programms des Massenspeichers in den Speicher mit direktem Zugriff (RAM).

Der Monitor (im ROM) ist für die Tastatur-Erfassung (Taste 1) verantwortlich. Er ist jedoch nicht verantwortlich für die Übertragung als solche, die von einem anderen Programm, z.B. einem Sprachmodul, das als Programm-Modul (ROM) einliegt, übernommen wird.

Page blanche

VI Das Programm-Modul

Es enthält im wesentlichen ROM (programmierbarer Festwertspeicher) Speicherbausteine, dessen Speichervermögen zwischen 16 bis 64 Kilo-Byte variieren kann. Die Bedeutung des Moduls liegt darin, daß das in ihm enthaltene Programm sofort verfügbar ist und im RAM-Speicher keinen Platz beansprucht; er bleibt ganz und gar für den Benutzer verfügbar.

Das Basic-Programm-Modul

Es enthält je nach der Version 16 oder 32 Kilo-Byte ROM-Speicher, der den BASIC-Interpreter enthält.

BASIC ist eine höhere Programmiersprache, deren Befehle im Programm-Modul gespeichert sind und die es dem Benutzer ermöglichen, dem Mikroprozessor Anweisungen zu geben.

Die Anweisungen werden in einer einfachen standardisierten Sprache ausgedrückt. Der BASIC-Interpreter übersetzt (interpretiert) sie in eine Form, die der Mikroprozessor verarbeiten kann.

Um zum Beispiel das Produkt $48 * 1024$ zu berechnen, ist es nicht erforderlich, dem Mikroprozessor mitzuteilen, in welchen Adressen sich die Zahlen 48 und 1024 befinden, wie sie zu verarbeiten sind und wo das Ergebnis abzulegen ist, sondern es genügt, über die Tastatur einzugeben:

PRINT 48 * 1024

Sobald Sie dann die ENTER-Taste betätigen, erscheint das Resultat auf dem Bildschirm:

49252

BASIC arbeitet zwar mit englischen Befehlen, ist aber leicht zu erlernen.

Um in ein BASIC geschriebenes Spiel, das sich auf einer Kassette oder Diskette befindet, zu spielen, müssen Sie das BASIC-Programm-Modul einlegen (um dem Mikroprozessor die BASIC-Befehle verständlich zu machen), und das Spielprogramm dann in den RAM-Speicher laden.

Neben BASIC stehen weitere Programmiersprachen als Module zur Verfügung: LOGO, FORTH, ASSEMBLER...

Andere Module

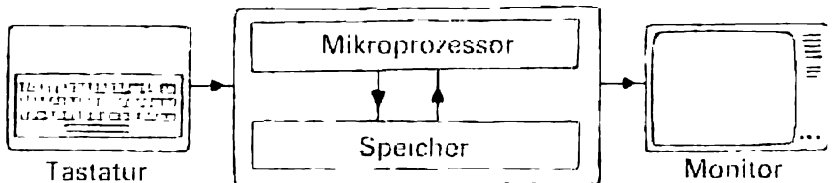
Ein Spiel-Modul wird genauso eingelegt wie das BASIC-Programm-Modul. Natürlich können Sie mit einem Spiel-Modul nicht das Ergebnis einer Multiplikation abfragen oder Programme schreiben, denn es enthält bereits ein anderes, fertiges Programm.

Die Befehle in einem Spiel-Modul müssen für den Mikroprozessor nicht erst interpretiert werden. Sie sind in der Assembler-Sprache des Mikroprozessors 6809-E geschrieben und daher direkt ausführbar (sogenannte «Maschinenbefehle»).

VII Die Peripheriegeräte

Nachdem wir nun die Rolle des Programm-Moduls geklärt haben, können wir uns mit den beiden Geräten befassen, die es dem Benutzer ermöglichen, mit dem Mikroprozessor zu kommunizieren. Diese Geräte sind die Tastatur und der Bildschirm. Beide sind (für den Mikroprozessor) Peripheriegeräte, auch wenn die Tastatur scheinbar zum Computer gehört.

Den Bildschirm nennt man Ausgabe-Peripheriegerät wegen seiner Sichtbarmachung und Eingabe-Peripheriegerät dank des Lightpen.



1) Die Tastatur und der ASCII-Code*

Über die Tastatur können Sie dem Mikrocomputer Befehle erteilen. Dabei wird die entsprechende Information in binäre Zeichen umgewandelt und an den Mikroprozessor übergeben.

Diese Umwandlung ist erforderlich, da der 6809-E und alle anderen Komponenten des Computers nur die Werte 0 und 1 verarbeiten können.

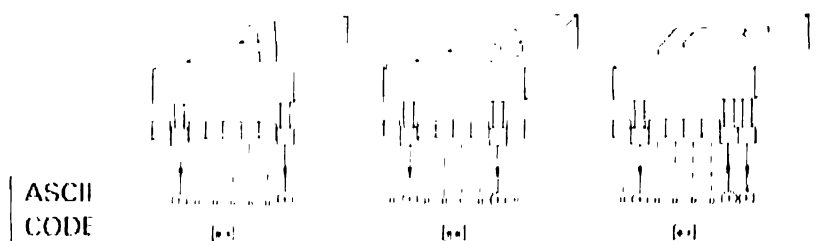
* American Standard Code for Information Interchange.

Wie alle anderen Tastaturen enthält auch die Tastatur des TO7-70 neben den 26 Buchstaben des Alphabets noch weitere Zeichen. Wenn Sie zum Beispiel den Buchstaben A eingeben, muß dieser erst in eine Folge der Werte 0 und 1 umgewandelt werden, damit ihn der Mikroprozessor erkennen kann. Die Umwandlung aller Zeichen der Tastatur erfolgt mit Hilfe des ASCII-Codes.

Jedes Zeichen wird durch ein Byte dargestellt, in einem Byte können 256 verschiedene Werte dargestellt werden.

Ab.	Zeichen	36	4	107	1
1	Taste []	47	54	108	2
2	Phed	48	55	109	3
3	Phed	49	56	110	4
4	Phed	50	57	111	5
5	Phed	51	58	112	6
6	Phed	52	59	113	7
7	Taste []	53	60	114	8
8	Taste []	54	61	115	9
9	Taste []	55	62	116	10
10	Taste []	56	63	117	11
11	Taste []	57	64	118	12
12	Taste []	58	65	119	13
13	Taste []	59	66	120	14
14	Taste []	60	67	121	15
15	Taste []	61	68	122	16
16	Taste []	62	69	123	17
17	Taste []	63	70	124	18
18	Taste []	64	71	125	19
19	Taste []	65	72	126	20
20	Taste []	66	73	127	21
21	Taste []	67	74	128	22
22	Taste []	68	75	129	23
23	Taste []	69	76	130	24
24	Taste []	70	77	131	25
25	Taste []	71	78	132	26
26	Taste []	72	79	133	27
27	Taste []	73	80	134	28
28	Taste []	74	81	135	29
29	Taste []	75	82	136	30
30	Taste []	76	83	137	31
31	Taste []	77	84	138	32
32	Taste []	78	85	139	33
33	Taste []	79	86	140	34
34	Taste []	80	87	141	35
35	Taste []	81	88	142	36
36	Taste []	82	89	143	37
37	Taste []	83	90	144	38
38	Taste []	84	91	145	39
39	Taste []	85	92	146	40
40	Taste []	86	93	147	41
41	Taste []	87	94	148	42
42	Taste []	88	95	149	43
43	Taste []	89	96	150	44
44	Taste []	90	97	151	45
45	Taste []	91	98	152	46
46	Taste []	92	99	153	47
47	Taste []	93	100	154	48
48	Taste []	94	101	155	49
49	Taste []	95		156	50
50	Taste []	96		157	51
51	Taste []	97		158	52
52	Taste []	98		159	53
53	Taste []	99		160	54
54	Taste []	100		161	55
55	Taste []	101		162	56

Die Umwandlung in den ASCII-Code erfolgt beim Betätigen der Tasten, denn unter jeder Taste befinden sich – bildlich gesprochen – acht elektrische Kontakte. Jede Taste schließt beim Niederdrücken diejenigen Kontakte, die den entsprechenden ASCII-Code erzeugen:



Jede Taste, die Sie betätigen, ist also unabhängig von den anderen codiert. Der BASIC-Befehl PRINT ergibt somit die Zahlenfolge 80, 82, 73, 78, 84.

Diese Zahlenfolge wird dann vom BASIC-Interpreter als ein Wort der Sprache erkannt, die einer genauen Funktion (der Ausgabe) entspricht und sofort durch eine einzige Zahl neu verschlüsselt wird.

2) Der Bildschirm

Wenn das System fehlerfrei arbeitet, erscheint der Buchstabe A auf dem Bildschirm, sobald Sie die entsprechende Taste drücken. Das scheint nicht weiter aufregend zu sein, doch betrachten wir einmal alle Vorgänge, die erforderlich sind, um diesen Schritt auszuführen.

Durch die Leitungen des Computers bewegt sich nichts, das Ähnlichkeit mit einem A hätte.

Dieses A ist sehr schnell zur Binärzahl 01000001 geworden (65 im ASCII-Code).

Um die Fortsetzung des Prozesses zu verstehen, an dessen Ende ein A auf dem Bildschirm erscheint, müssen Sie sich Ihren Fernsehapparat genauer ansehen.

Die Oberfläche des Bildschirms ist von innen mit einer dünnen Schicht bedampft, die zu leuchten beginnt, sobald sie von einem Elektronenstrahl getroffen wird.

Der aus einer «Elektronenkanone» abgegebene Elektronenstrahl überstreicht den Bildschirm zeilenweise mit großer Geschwindigkeit, und zwar 50mal pro Sekunde.

Durch das Auftreffen des Strahls wird ein Mosaik aus leuchtenden oder dunklen Punkten geschaffen; jeder Punkt wird ein Bildelement genannt.

Der Bildschirm des TO7-70 besteht aus 64000 Pixel, die in einer 320*200-Matrix angeordnet sind.

3) Die Darstellung eines Zeichens

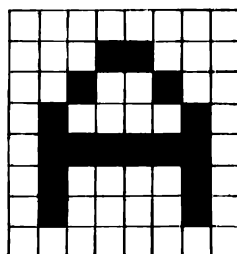
Kehren wir zurück zu unserem Tastendruck, den der Computer in den Binärwert 01000001 (für A) umgewandelt hat. Aufgrund

dieser Information liest der Mikroprozessor im ROM-Speicher eine Folge von 8 Bytes, die es ihm ermöglichen, den Buchstaben A auf dem Bildschirm darzustellen.

Die 8 Byte aus
dem ROM-Speicher

N° 1	0	0	0	0	0	0	0
N° 2	0	0	0	1	1	0	0
N° 3	0	0	1	0	0	1	0
N° 4	0	1	0	0	0	0	1
N° 5	0	1	1	1	1	1	0
N° 6	0	1	0	0	0	0	1
N° 7	0	1	0	0	0	0	1
N° 8	0	0	0	0	0	0	0

Ihre graphische
Darstellung



Um diese Informationen an die Elektronenkanone weiterzugeben, bedient sich der Mikroprozessor eines speziellen Schaltkreises, «Videointerface» genannt. Dieser Baustein steuert die Elektronenkanone entsprechend den vom Mikroprozessor empfangenen Informationen und «schießt» auf alle Punkte, die erleuchtet werden müssen, um den Buchstaben A darzustellen.

Sie werden jetzt verstehen, daß die Anzeige der über die Tastatur eingegebenen Zeichen nur ein Service für den Benutzer ist und für den Mikroprozessor eine reine Zeitverschwendung darstellt, denn er kennt ja unmittelbar nach jedem Tastendruck den Code des eingegebenen Zeichens. Der TO7-70 kann also ihre Befehle auch empfangen, wenn kein Bildschirm angeschlossen ist.

4) Der Lightpen

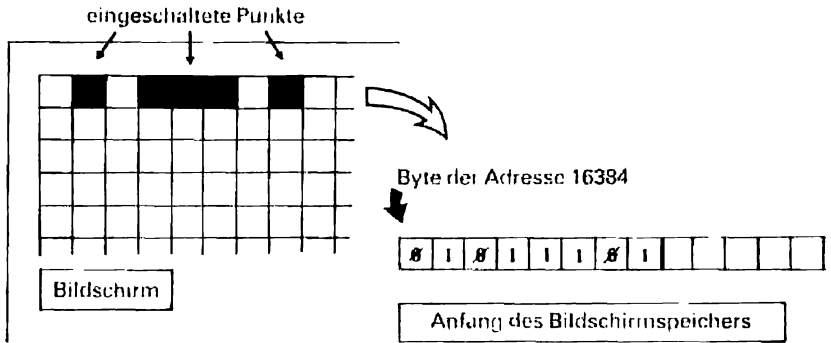
Der Lightpen reagiert empfindlich auf das vom Bildschirm erzeugte Licht. Dieses Licht, das Ihnen so beständig vorkommt, blinkt in Wirklichkeit 50 mal pro Sekunde, das heißt so schnell, daß Ihr Auge dies nicht erkennen kann. Der Lightpen hingegen nimmt dieses Blinken sehr wohl wahr und kann erkennen, auf welche Stelle des Bildschirms Sie ihn richten. Aus der Zeit zwischen Start eines Lichtpunktdurchlaufs und dem Erkennen durch den Lichtgriffel können Zeile und Spalte genau berechnet werden. Wird z.B. nach der Hälfte der Zeit der Lichtpunkt abgetastet, folgt der Lightpen auf die Bildmitte.

VIII Der Bildschirm- speicher

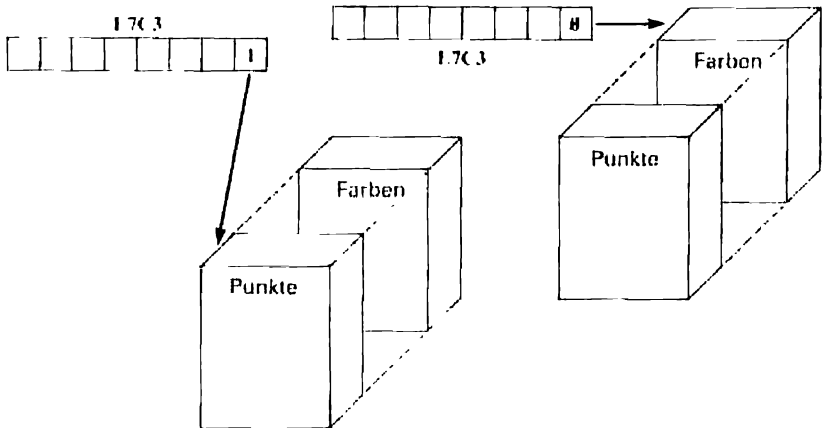
Der Bildschirm besteht aus 64000 Bildpunkten (Bildelement, pixel) und sein Zustand kann als eine Folge von 0 und 1 angesehen werden, je nachdem, ob ein Bildpunkt angeschaltet oder ausgeschaltet ist. Der Bildschirmspeicher ist also eine dem Bildschirm entsprechende Kopie. Die Speicherung erfolgt von links nach rechts und von oben nach unten. Den 64000 Bildpunkten des Bildschirms werden 8000 Byte RAM-Speicher zugewiesen.

Was den TO7-70 betrifft, mußte, da es sich um einen Farbmikrocomputer handelt, auch ein Speicherbereich für die Farben vorgesehen werden. Diese beiden Abschnitte müssen natürlich eng miteinander verbunden sein, da jedes Bildelement automatisch mit einer Farbe verbunden ist. Statt nun diese beiden Abschnitte zu trennen, haben die Konstrukteure es vorgezogen, sie durch eine Methode zu verbinden, die man Speicherbankumschaltung oder «bank-switching» nennt. Sie soll im folgenden beschrieben werden.

Der zwischen den Adressen 16384 und 24575 (d.h. 8192 Byte, wobei die letzten 192 Byte nicht verwendet werden) liegende Speicherbereich ist eine Kopie des Bildschirms. Eine erste Bank entspricht dem Punkt-Speicher. Jedes Bit das 0 ist, entspricht einem ausgeschalteten Punkt (Hintergrundpunkt), jedes Bit das 1 ist, entspricht einem eingeschalteten Punkt (Vordergrundpunkt). Die «Abbildung» des Bildschirms in den Speicher ist sehr einfach: die Bildschirmlinien werden von links nach rechts durchlaufen.



Das Erstaunlichste ist aber der Farbenspeicher, der im selben Speicherbereich untergebracht ist, jedoch an anderer Stelle. Der Punkt-Speicher und der Farb-Speicher können als zwei Banken angesehen werden, die die gleichen Adressen haben und hintereinander angeordnet sind. Erhält der Mikroprozessor den Befehl, eine dieser Adressen abzufragen, so wird er in die eine oder die andere Bank gehen, je nachdem, welchen Wert das niedrigstwertige Bit einer bestimmten Speicherzelle (Adresse E7C3) hat.

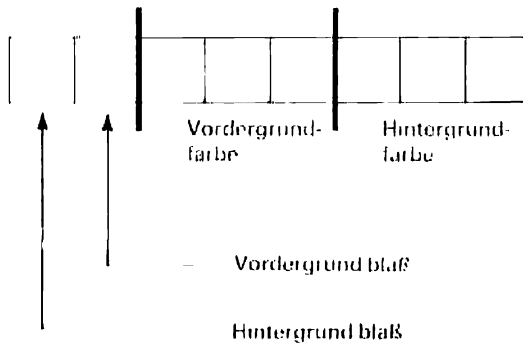


Das letzte Bit dieses Speichers dient als «Flag» zum Lesen und Schreiben in eine der beiden Banken.

Jedem Byte des Punkt-Speichers entspricht ein Byte im Farb-Speicher, doch dieser Schluß gilt nicht in bezug auf den Bildpunkt; es ist nämlich nicht möglich, mehr als zwei Farben einem einzigen Bit zu codieren. Man geht nach folgendem Prinzip vor:

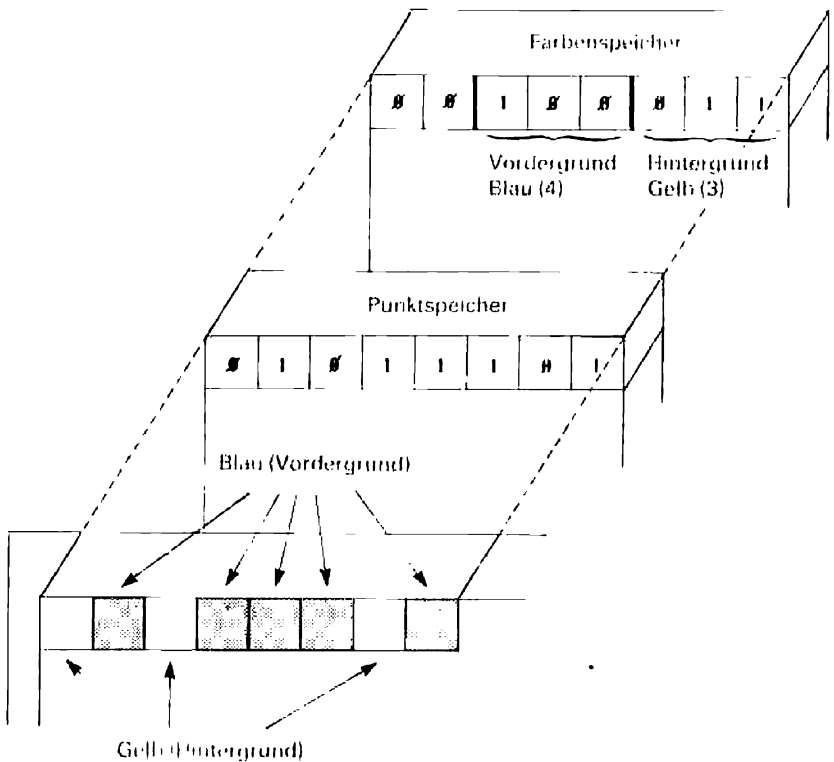
jede horizontale Gruppe von 8 Bildpunkten kann zwei Farben aufnehmen, eine für die angeschalteten Punkte (Vordergrundfarbe) und eine für die ausgeschalteten Punkte (Hintergrundfarbe). Die 8 Bits eines jeden Byte genügen für diese Codierung.

Sie sind wie folgt angeordnet:



Die wichtigste Auswirkung dieses Aufbaus ist die «Starrheit» der Farben.

Hier als Beispiel die Anordnung des ersten Byte des Bildschirms und seine Abbildung in die beiden Banken des Bildschirmspeichers.



Wir möchten noch zum Schluß darauf hinweisen, daß bei der Speichererweiterung des TO7-70 von 64 auf 128 K dieselbe Methode des «bank switching» angewendet wird.

IX Was geschieht unter der Tastatur des T07-70

Sobald Sie das BASIC-Programm-Modul eingelegt haben, können Sie mit dem Programmieren beginnen.

Auf den folgenden Seiten haben wir den Zusammenhang zwischen Bildschirmdarstellung und Speicherinhalt anhand eines kurzen Programms schematisch dargestellt.

Das Programm erfordert nur folgende einfache Befehle:

A = 7	(schreibt die Ziffer 7 in die Speicherzelle A)
PRINT C	(schreibt den Inhalt der Zeile C auf den Bildschirm)
CLS	(löscht den Bildschirm)
LIST	(zeigt das Programm auf dem Bildschirm an)
RUN	(startet das Programm)
NEW	(löscht den RAM-Speicher und damit auch das Programm)

Eingabe	1 A = 7	ENTER	2 B = 13 ENTER																														
Anzeige auf dem Bildschirm	1 A = 7 _	1 A = 7 _	1 A = 7 2 B = 13 _																														
Der jeweilige Speicherinhalt (unsichtbar!)	<table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>											<table><tr><td>1 A = 7</td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>	1 A = 7										<table><tr><td>1 A = 7</td></tr><tr><td>2 B = 13</td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>	1 A = 7	2 B = 13								
1 A = 7																																	
1 A = 7																																	
2 B = 13																																	

Drücken der
ENTER -Taste
bewirkt die Speicherung
der Programmzeile.

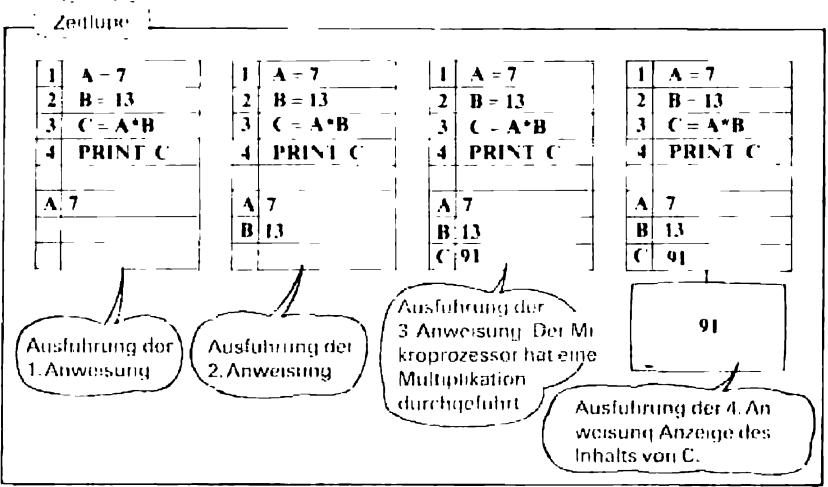
Eingabe	1 C = A * B ENTER 4 PRINT C ENTER	CLS	LIST ENTER
	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C	-	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C
Der jeweilige Speicherinhalt	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C

Der Bildschirm ist leer, aber das Programm ist weiterhin im Speicher vorhanden.

Nach dem LIST-Befehl wird das Programm angezeigt.

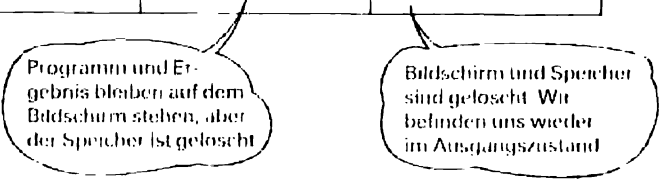
Eingabe	RUN ENTER
	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C RUN 91
Der jeweilige Speicherinhalt	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C
	A = 7 B = 13 C = 91

Nach Drucken von **ENTER** erfolgt sofort die Anzeige der 91. Da zwischen spielt sich jedoch einiges ab.



Fortsetzung

Eingabe	NEW ENTER	CLS																																																																																																																																																																																																								
Anzeige auf dem Bildschirm	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C NEW	-																																																																																																																																																																																																								
Der jeweilige Speichereinhalt	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																																																					<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																																																				



X Mikro-Wörterbuch

A

Adresse (Seite 48): Nummer einer der 65536 Speicherzellen.

ASCII (Seite 57): American Standard Code for Information Interchange. Codetabelle mit 128 Zeichencodes von 0 bis 127 für ein Byte.

Assembler (Seite 56): Programmiersprache, als Programm-Modul für den TO7-70 erhältlich. Die Assemblersprache ermöglicht eine besonders schnelle Programmausführung unter Verwendung direkt ausführbarer Maschinenbefehle.

B

BASIC (Seite 55): Programmiersprache, als Programm-Modul für den TO7-70 erhältlich. Eine der einfachsten und bekanntesten Sprachen für Mikrocomputer.

Binärsystem (Seite 41): Zahlensystem auf der Basis 2. Es werden nur die Ziffern 0 und 1 benutzt. Mit diesem System arbeitet der Mikroprozessor.

Bit (Seite 41): Kleinste Informationseinheit, die dem Vorhandensein oder Nichtvorhandensein eines Elektronenbündels entspricht. Ein Bit kann den Wert 0 oder 1 haben.

Bus: Elektrische Verbindung für die Kommunikation zwischen den verschiedenen Komponenten eines Mikrocomputers. Man unterscheidet:

- den **Befehlsbus** zur Übertragung der Befehle,
- den **Adreßbus** zur Übertragung der Speicheradressen,
- den **Datenbus** zur Übertragung der Speicherinhalte.

Byte (Seite 42): Eine Gruppe von 8 Bit.

E

Elektron (Seite 59, 37): Elementarteilchen mit negativer elektrischer Ladung. Durch die Bewegung von Elektronen entsteht ein Strom.

Elektronenkanone (Seite 59): Gerät im Inneren der Bildröhre des Fernsehgerätes, das einen Elektronenstrahl auf die Leuchtschicht schießt, um bestimmte Zeichen darzustellen.

F

Forth (Seite 56): Programmiersprache, als Programm-Modul für den T07-70 erhältlich. Forth (Fourth = vierte Generation) ist eine moderne stapelorientierte Computersprache.

H

Hertz (Seite 41): Einheit der Frequenz (Schwingungen pro Sekunde). 1 Megahertz sind 1 000 000 Hertz.

I

IC (Seite 35): Siliziumplättchen, auf dem elektronische Schalter (Transistoren) und Leiterbahnen aufgedruckt sind. Die Anzahl der Transistorfunktionen wird durch die «Integrationsdichte» bestimmt. Ein IC wird auch «Chip» genannt.

Interface (Seite 60): Elektronischer Schaltkreis, der die Kommunikation zwischen der Zentraleinheit und einem Peripheriegerät ermöglicht. Das Videointerface z.B. übermittelt die Informationen vom Mikroprozessor an den Bildschirm. Es gibt auch extern (außen) aufsteckbare Interfaces, z.B. für Drucker, Joysticks usw.

K

L

Lightpen (Seite 16): Gerät, das auf das Licht des Bildschirms empfindlich reagiert; dank des Lightpens kann man direkt auf dem Bildschirm zeichnen oder auswählen.

Logo (Seite 56): Programmiersprache, als Programm-Modul für den TO7-70 erhältlich. Logo ist besonders leicht zu erlernen und damit auch für Kinder gut geeignet.

M

Mikroprozessor (Seite 30, 40): Integrierte Schaltung zur Steuerung und Verarbeitung von Informationen. Der Mikroprozessor des TO7-70 ist der 6809-E.

Modul (Seite 47): Kleines Gehäuse, das einen ROM-Speicher enthält und die Kommunikation des Benutzers mit der Zentraleinheit ermöglicht. Es sind Spiel-, Sprach- und Programm-Module von 4 bis 64 Kilobyte erhältlich.

Monitor (Seite 49): Zwei Bedeutungen:

- Ein Programm im ROM-Speicher, das dem Mikroprozessor nach dem Einschalten die Kommunikation mit den Peripheriegeräten ermöglicht.
- Ein speziell für die Datenverarbeitung geeigneter Bildschirm.

P

Peripherie (Seite 57): Externe Geräte, die mit der Zentraleinheit kommunizieren.

Peripheriegerät für die Eingabe: Tastatur.

Peripheriegerät für die Ausgabe: Bildschirm.

Weitere Peripheriegeräte: Programmrecorder, Diskettenstation, Lightpen, Drucker, Joystick, usw.

Pixel (Seite 59): Bildelement. Der Bildschirm des TO7-70 ist unterteilt in 64 000 Bildelemente, die in einer 320*200-Matrix angeordnet sind.

S

Silizium (Seite 36): Für die Herstellung von integrierten Schaltkreisen benötigtes chemisches Element.

Speicher (Seite 24, 48, 50, 51, 61): Integrierte Schaltung zur Speicherung von Informationen.

ROM-Speicher: Festwertspeicher, aus dem nur gelesen werden kann. Sein Inhalt bleibt auch in stromlosem Zustand erhalten.

RAM-Speicher: Schreib/Lese-Speicher, der sowohl gelesen als auch beschrieben werden kann. Sein Inhalt geht nach dem Ausschalten verloren. Der RAM-Speicher ist der Speicher für den Benutzer.

Bildschirmspeicher (Seite 61): Teil des RAM-Speichers zur Speicherung der Punkte und Farben der anzuzeigenden Zeichen.

Massenspeicher: Externer Speicher zur magnetischen Aufzeichnung großer Datenmengen (Programmrekorder, Diskettenstation).

Speicherbankumschaltung (Seite 61) (oder bank switching): Mit Hilfe dieser Organisationstechnik kann man die gleiche Adresse zwei physisch verschiedenen Speicherzellen zuordnen.

Speicherkapazität (Seite 49): Anzahl der vom Mikroprozessor adressierbaren Speicherplätze. Die Kapazität des TO7-70 beträgt 70 Kilobyte. Mit Speichererweiterung: 134 Kilobyte.

Speicherzelle (Seite 48): Element des Speichers, in dem ein Byte abgelegt werden kann.

T

Takt (Seite 41): Steuersignal für den Mikroprozessor. Der Takt wird von einem quarzgesteuerten Taktgeber erzeugt. Die Taktfrequenz im TO7-70 beträgt 1 MHz (1 Megahertz).

Transformator (Seite 41): Ein elektrisches Gerät zur Umwandlung von 220 Volt Wechselstrom in eine niedrigere Spannung. Der Transformator befindet sich im Gehäuse des TO7-70.

Transistor (Seite 37): Elektronischer Schalter.

Z

Zeichen (Seite 60): In einer 8*8-Matrix codiertes Muster. Es können alle Buchstaben und Zahlen dargestellt werden, die auf der Tastatur des TO7-70 vorhanden sind. Nicht darstellbar sind die sogenannten Steuerzeichen (ASCII 0 bis 31), die zur Positionierung des Cursors und zur Bildaufbereitung dienen.

Zentraleinheit: Sammelname für die durch den Bus verbundenen Einheiten Mikroprozessor, Speicher und Interface.

3. Der Monitor des TO7-70

Inhaltsverzeichnis

1.0	Einleitung	76
1.1	Der Monitor des TO7-70	76
1.2	Zugriff auf die Monitor-Routinen	77
2.0	Interrupt-Steuerung	78
3.0	Initialisierung	79
3.1	Kalt/Warmstart-Test	79
3.2	Initialisierung der System-PIA-Bausteine	79
4.0	Zeichengeneratoren	80
4.1	G0-Standardalphabet	80
4.2	G2-Alphabet	81
4.3	Vom Benutzer definierte Zeichen	81
5.0	Routinen zur Steuerung des Bildschirms	81
5.1	Zugang zum Farbspeicher	83
5.2	Zugang zum Zeichenspeicher	83
5.3	Signalton	83
5.4	Alphanumerische Zeichendarstellung	83
5.4.1	Normalsequenz	84
5.4.1.1	Darstellbare Codes	84
5.4.1.2	Interpretierbare Codes	84
5.4.1.3	Vom Benutzer definierbare Zeichen	85
5.4.2	Definition des Bildschirmfensters	85
5.4.2.1	Definieren des unteren Randes	85
5.4.2.2	Definieren des oberen Randes	85
5.4.2.3	Cursor-Positionierung	86
5.4.3	SS2-Sequenz	86
5.4.4	TELETEL-Zeichen (Semigrafik)	87
5.4.5	Escape-Sequenz	88
5.4.5.1	Farbattribute	88
5.4.5.2	Sonstige Attribute	88

6.0	Tastatur	91
6.1	Schnelle Tastaturabfrage	91
6.2	Tastaturdecodierung	91
7.0	Hochauflösende Grafik	92
7.1	Bildpunktansteuerung	92
7.1.1	Grafikmodus	93
7.1.2	Zeichenmouds.	94
7.2	Darstellung einer Linie	94
7.2.1	– im Grafikmodus	94
7.2.2	– im Zeichenmodus	95
7.3	Horizontalvektoren	95
7.4	Auslesen der Farbe eines Bildpunktes	95
8.0	Auslesen von Zeichen aus dem Bildschirm	96
8.1	Normale Zeichen.	96
8.2	Umlaute und akzentuierte Zeichen.	96
9.0	Tonerzeugung	96
10.0	Abfrage der Joysticks	98
11.0	Lightpen	99
11.1	Test des Lightpen-Schalters	99
11.2	Auswertung der Lightpen-Signale	99
12.0	Schnittstellen-Adapter	100
13.0	Steuerung des Programmrekorders	101
14.0	Steuerung der Diskettenlaufwerke	102
14.1	Physischer Diskettenzugriff	102
14.2	MICROSOFT(R) BASIC-Format	105
14.2.1	Dateibelegungs-Tabelle (FAT)	105
14.2.2	Inhaltsverzeichnis (Directory)	106
15.0	Allgemeine Informationen	108
15.1	Speicheraufteilung (Memory Map)	108
15.2	Einsprungadressen der Standardroutinen	109

15.3	Vom Monitor belegte RAM-Register (Zeropage)	109
15.4	E/A-Adressen	113
15.4.1	System-PIA	113
15.4.2	PIA im Musik-/Spiele-Interface	114
15.4.3	PIA im Kommunikations-Interface	114

1.0 Einleitung

1.1 Der Monitor des TO7-70

Der Monitor des TO7-70 setzt sich aus verschiedenen «Routinen» zusammen. Jede Routine ist ein spezielles Programm zur Lösung einer ganz bestimmten Aufgabe. Im Monitor sind Routinen zur Bildschirmsteuerung, für die Abfrage des Lightpens, der Joysticks und der Tastatur, sowie Steuerrouninen für Programmrekorder, Diskettenlaufwerke usw. enthalten.

Wie das BASIC oder die Anwenderprogramme in ROMs können auch Sie auf diese Routinen zugreifen, wenn Sie eigene Assemblerprogramme schreiben. Der Zugriff erfolgt nach den folgenden Prinzipien:

1. Laden bestimmter Prozessor- und /oder RAM-Register mit den Eingabeparametern.
2. Aufrufen der gewünschten Standardroutine (siehe Abschnitt 1.2).
3. Wenn nötig, die erhaltenen Ausgabeparameter nach Abschluß der Routine aus den Prozessor- und/oder RAM-Registern auslesen.

In den folgenden Abschnitten werden manche Werte oder Adressen in hexadezimalen Zahlen angegeben, diese sind durch den Zusatz «H» gekennzeichnet. Fehlt dieser Zusatz, so ist die Zahl als Dezimalzahl zu verstehen, sofern sie nicht ausdrücklich anders bezeichnet wurde.

Prozessor- und RAM-Register sind als XXXXH oder XXXXH-XXXXH bezeichnet, wobei X ein Hexadezimalzeichen darstellt. Zum besseren Verständnis werden den RAM-Registern Namen gegeben, die auf ihre Bedeutung hinweisen (z.B. STADR für Startadresse).

Beispiel: Soll das Register STADR (6060H-6061H) mit dem Wert 4B0H geladen werden, so lädt man den Wert 04H in das Register STADR (6060H) und den Wert B0H in das Register STADR + 1 (6061H).

Den Binärzahlen geht das Zeichen «%» voran. Die Bits werden von rechts nach links numeriert, wobei b0 (rechts) die geringste, b7 (links) die höchste Wertigkeit hat.

Beispiel: %11001100

b7 enthält eine 1, b0 enthält eine 0. In den weiteren Beschreibungen bezeichnen wir diese beiden Zustände als gesetzt (1) bzw. rückgesetzt oder gelöscht (0).

1.2 Zugriff auf die Monitor-Routinen

Der Standard-Zugriff auf die Monitor-Routinen erfolgt entweder über einen «JSR»- oder einen «JMP»-Befehl auf die Einsprungsadresse der Routine. Beim TO7-70 gibt es 17 solcher Adressen.

Beispiel 1

PROG	EQU	★
	LDB	#07
	JSR	\$E803
		nächster Befehl

Der Summer ertönt und das Programm wird bei «nächster Befehl» fortgesetzt.

Beispiel 2

PROG1	EQU	★
	JSR	PROG2
		nächster Befehl 1
PROG2	EQU	★
	LDB	#07
	JMP	\$E803
		nächster Befehl 2

Der Summer ertönt, doch das Programm wird nun bei «nächster Befehl 1» fortgesetzt, statt – wie man vielleicht erwartet hätte – bei «nächster Befehl 2».

Bei der Verwendung einer Monitor-Routine werden alle Register gespeichert und nach Beendigung des Programmes wiederhergestellt. Eine Ausnahme bilden hier das CC-Register, das den Wert des momentanen CC enthält, und natürlich die Register, in denen die Rückkehrparameter übergeben werden (meistens B, X und Y).

Es muß dringend vor dem Versuch gewarnt werden, mit einer anderen als der hier beschriebenen Standardmethode auf die Monitor-Routinen zuzugreifen. Diese Routinen setzen nämlich voraus, daß alle Register auf dem Stackspeicher abgelegt und bestimmte Prozessorregister mit den richtigen Werten geladen wurden. Eine Liste der Standard-Einsprungadressen finden Sie in Abschnitt 15.2.

2.0 Interrupt-Steuerung

Während eines Programmablaufes müssen bestimmte Routinen, wie z.B. die Tastaturabfrage, in regelmäßigen Abständen immer wieder angesprungen werden. Dieses Problem wird durch Programmunterbrechungen (Interrupts) gelöst: Tritt ein Interrupt auf, so unterbricht der Prozessor die Ausführung des Hauptprogrammes und springt zu der Routine, deren Einsprungadresse in einem Interrupt-Pointer (Zeiger) steht.

Um zum Beispiel den 100 Millisekunden TIMER-Interrupt, der die automatische Tastaturwiederholung und das Blinken des Cursors steuert, für Ihre eigenen Programme zu nutzen, brauchen Sie nur Bit 5 des RAM-Registers STATUS (6019H) auf 1 zu setzen und die Startadresse Ihres Unterprogramms in TIMEPT (6027H-6028H) abzulegen. Ihre Routine muß in diesem Fall aber unbedingt mit der Anweisung JMP KBIN\$ (E830H) enden.

Nun eine Liste der beim TO7-70 möglichen Interrupts und ihrer Pointer:

IRQ	IRQPT	(6021H-6022H)
FIRQ	FIRQPT	(6023H-6024H)
NMI	NMIPT	(6025H-6026H)
SWI	SWI1	(602FH-6030H)
SWI2	springt direkt nach 6800H	
SWI3	springt direkt nach 7000H	

Achten Sie bitte darauf, daß auch manche Monitor-Routinen unterbrochen werden können. Ihre Programme dürfen deren Parameter (Registerinhalte) nicht verändern!
Die Rückkehr zum Hauptmenü erreichen Sie durch JMP MENU\$ (E82DH).

3.0 Initialisierung

Nach dem Einschalten des Gerätes oder Drücken der RESET-Taste führen spezielle Monitor-Routinen verschiedene Initialisierungen und Tests durch:

3.1 Kalt/Warmstart-Test

Je nach aufgetretenem Startvorgang (Kalt/Warmstart) werden bestimmte Daten beibehalten oder neu initialisiert:

- Lightpen-Einstellung
- Grafik-Kopiercodes (in Abhängigkeit vom verwendeten Drucker)
- Standardwert des Tastatur-Autorepeats

In beiden Fällen werden:

- die Interrupt-Pointer auf die Startadressen der Monitor-Routinen gesetzt,
- die Tabellenzeiger mit den Standardwerten initialisiert (Tastaturbelegung, Standard- und Benutzer-Zeichengenerator),
- alle anderen Register mit Standardwerten belegt.

3.2 Initialisierung der System-PIA-Bausteine

Der Baustein 6846 ist wie folgt initialisiert:

★ Kontrollregister:

- Ausgang: Tonerzeugung, Datenausgang zum Programm-Rekorder

★ Datenregister:

- Eingang: Datenspeicher vom Programm-Rekorder, Lightpenswitcher

Ausgang: Farbspeicherumschaltung, Tastatur-LED, Rahmenfarbe

★ Timer:

Initialisiert mit 100 Millisekunden

Der Baustein 6821 ist wie folgt initialisiert:

★ Port A:

- Eingang: Abfrage der Tastatur-Matrix

★ Port B:

- Eingang: Tastatur-Ausgang
- Ausgang: Versorgung der Tastatur-Matrix, Bank-Switching

★ Kontrollregister:

- Ausgang: Rekorder-Motorsteuerung, Mischbild-Einblendung

Die Datenleitungen im Spiele-Interface (6821) sind als Eingänge programmiert.

4.0 Zeichengeneratoren

4.1 G0-Standardalphabet

Der G0-Zeichensatz umfaßt die darstellbaren ASCII-Zeichen (Codes 20H bis FFH).

Jedes Zeichen ist aus 8 Bytes zusammengesetzt, das erste Byte entspricht der untersten Zeile einer 8×8-Punktmatrix, die ein Zeichen darstellt.

Beispiel: Matrix Buchstabe A

```

0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 1 0 0 0 0 1 0
0 1 1 1 1 1 1 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0

```

Im Generator ist das Zeichen mit der Bytefolge 00H, 42H, 42H, 7EH, 42H, 24H, 18H, 00H gespeichert.

Die Anfangsadresse der G0-Tabelle ist im Register PTGENE (60CFH-60D0H) abgelegt. Sie können das Standardalphabet

umdefinieren, indem Sie zunächst Ihr eigenes Alphabet in der oben beschriebenen Weise erstellen und anschließend das Register PTGENE mit der Startadresse Ihrer eigenen Tabelle laden.

4.2 G2-Alphabet

Das G2-Alphabet enthält 5 akzentuierte Zeichen, darunter die Umlaute und das Zeichen ç. Die Definition erfolgte auf die zuvor dargelegte Art; die Darstellung sollte nur in Verbindung mit eingeschalteter Kleinschreibung erfolgen.

4.3 Vom Benutzer definierte Zeichen

Der Benutzer kann nach dem vorherigen Prinzip bis zu 128 Zeichen selbst definieren. Das Register USERAF (602DH-602EH) ist mit der Adresse der Benutzer-Zeichentabelle zu laden. Die Zeichencodes beginnen bei 80H und sind aufsteigend durchnummeriert. Die ersten 8 Bytes entsprechen also dem Code 80H, die zweiten 8 Bytes dem Code 81H usw. Standardmäßig ist USERAF mit der Startadresse der G0-Tabelle geladen.

5.0 Routinen zur Steuerung des Bildschirms

Der TO7-70 hat eine grafische Auflösung von 320×200 Bildpunkten, im Zeichenmodus sind 25 Zeilen zu je 40 Zeichen darstellbar. Jedes Zeichen besteht aus einer 8×8-Punkt-Matrix. Für die Zeichen- (Bildpunkt-), Hintergrund- und Rahmengestaltung stehen 16 Farben zur Verfügung.

Im Grafikmodus verwaltet die CPU 8000 Blöcke zu je 8 (horizontal angeordneten) Bildpunkten. Jeder Bildpunkt entspricht einem Bit, jeder Block einem Byte. Zur Speicherung der Zeichen- und der Farbinformationen eines Blocks werden zwei Bytes benutzt, die beide unter der gleichen logischen Adresse erreichbar sind. Die Umschaltung zwischen dem Zeichen- (Bildpunkte-) Speicher und dem Farbspeicher geschieht über eine Funktion des System-PIA 8846.

Innerhalb eines Farb-Bytes bestimmen die drei niederwertigen Bits die Farbe des Hintergrundes (im Zeichenspeicher nicht gesetzte Bits), die drei folgenden Bits die Farbe der Bildpunkte

(Zeichen-Bits gesetzt). Die zwei höchstwertigen Bits dienen dazu, die Auswahl zwischen normaler und blasser Farbe zu treffen. Eine 0 in Bit 6 wählt blasse Farben für die Bildpunkte, eine 0 in Bit 7 wählt eine blasse Farbe für den Hintergrund. Die 3-Bit-Farbcodes RGB beziehen sich auf die drei Grundfarben des Fernsehens: Rot (R), Grün (G) und Blau (B).

Es ergeben sich folgende Mischfarben:

Rot + Grün = Gelb

Rot + Blau = Violett

Blau + Grün = Hellblau

Rot + Grün + Blau = Weiß

Schwarz entsteht selbstverständlich dadurch, daß weder Rot noch Grün noch Blau eingeschaltet wird.

Beispiel: Farbcode = 1xxxx000

R = 0: kein Rot

G = 0: kein Grün

B = 0: kein Blau

f = 1: normale Farbe

Das Ergebnis ist ein schwarzer Hintergrund.

Beispiel: Farbcode = x0111xxx

R = 1: Rot vorhanden

G = 1: Grün vorhanden

B = 1: Blau vorhanden

f = 0: blasse Farbe

Das Ergebnis wäre «Blaßweiß», dies wäre jedoch von dem normalen Weiß kaum zu unterscheiden. Hier wurde deshalb eine Ausnahme gemacht und dieser Kombination die Farbe Orange zugeordnet.

Der TO7-70 bietet mit einem Zusatzmodul auch die Möglichkeit, ein Fernsehbild in das Computerbild einzublenden. Ist diese Funktion eingeschaltet (siehe Abschnitt 5.4.5.2), so erscheint das Fernsehbild an Stelle der schwarzen Bildteile so, als würde die

5.1 Zugang zum Farbspeicher

Entscheidend für die Anwahl des Farbspeichers ist Bit 0 (und nur dieses!) der Adresse E7C3H. Dies ist das Datenregister des System-PIA 6846.

BIT 0 = 0: Farbspeicher angewählt

Wenn Sie nun den Inhalt einer Adresse im Bereich zwischen 4000H und 5FFFH verändern, ändert sich die Farbe des entsprechenden Bildpunktes.

5.2 Zugang zum Zeichenspeicher

Entscheidend für die Anwahl des Zeichenspeichers ist Bit 0 (und nur dieses!) der Adresse E7C3H. Dies ist das Datenregister des System-PIA 6846.

BIT 0 = 1: Zeichenspeicher angewählt

Wenn Sie nun den Inhalt einer Adresse im Bereich zwischen 4000H und 5FFFH verändern, ändert sich der Status der entsprechenden Bildpunkte (Vorder-/Hintergrund).

5.3 Signalton

Um den Summer anzusprechen und einen Signalton zu erzeugen, muß

- Routine PUTC\$ (E803H) mit dem Code 7 aufgerufen und
- Inhalt des Registers BUZZ (6073H) gleich 0

sein. Beim TO7-70 testet die Routine das Register BUZZ auf seinen Inhalt. Ist dieser 0, so wird der Ton erzeugt. Bei allen anderen Werten ertönt kein Signalton.

5.4 Alphanumerische Zeichendarstellung

Einsprungsadresse: PUTC\$ (E803H)

Eingangsparameter: in 6809-Register B

Es gibt mehrere verschiedene Bildschirm-Steuersequenzen, die über die Routine PUTC\$ – zum Teil durch mehrfachen Aufruf – ausgeführt werden können. Alle diese Sequenzen setzen voraus, daß in Register B der Code für das darzustellende Zeichen oder die auszuführende Operation übergeben wird.

5.4.1 Normalsequenz

5.4.1.1 Darstellbare Codes

Alle Codes zwischen 20H und 7FH sind darstellbar. Das jeweilige Zeichen wird an der aktuellen Cursorposition dargestellt und der Cursor um eine Schreibstelle nach rechts bewegt. Falls Sie eigene Zeichen definiert (siehe 4.1) und diese ebenfalls unter den Codes 20H bis 7FH abgelegt haben, so sind auch diese darstellbar.

5.4.1.2 Interpretierbare Codes

Die Codes 07H bis 1FH werden von PUTCS\$ interpretiert. Die folgende Liste zeigt, welche Funktionen die Codes im Einzelnen bewirken:

- 07H : BEL → Signalton
- 08H : BS → Rückschritt (Backspace) oder kopiert das abgebildete Zeichen nach links, wenn Register COPCHR (6043H) den Wert FFH enthält.
- 09H : HT → Horizontaltabulator oder kopiert das abgebildete Zeichen nach rechts, wenn Register COPCHR (6043H) den Wert FFH enthält.
- 0AH : LF → Zeilenvorschub (Line Feed)
- 0BH : VT → Vertikaltabulator
- 0CH : FF → Bildschirmfenster löschen
- 0DH : CR → Wagenrücklauf (Carriage Return)
- 0EH : SO → TELETEL-Modus ein (G1-Satz)
- 0FH : SI → Rückkehr zum Normalmodus (TELETEL aus)
- 10H : DLE → keine Funktion
- 11H : DC1 → Cursor an
- 12H : DC2 → Repeat-Funktion
- 13H : DC3 → keine Funktion
- 14H : DC4 → Cursor aus
- 15H : NAK → keine Funktion
- 16H : SS2 → Akzent-Sequenz (G2-Satz)
- 17H : ETB → keine Funktion
- 18H : CAN → Abbruch (Cancel), löscht die Zeile ab Cursorstelle
- 19H : EM → keine Funktion
- 1AH : SUB → keine Funktion
- 1BH : ESC → Escape Sequenz

- 1CH : FS → keine Funktion
- 1DH : GS → keine Funktion
- 1EH : RS → setzt den Cursor in die linke obere Ecke
- 1FH : US → Sequenz zur Cursorpositionierung und Fensterdefinition

5.4.1.3 Vom Benutzer definierbare Zeichen

Codes, die größer als 7FH sind, entsprechen den vom Benutzer definierten Zeichen. Bevor PUTC\$ mit einem dieser Codes aufgerufen wird um ein solches Zeichen darzustellen, muß die Startadresse der Benutzer-Zeichentabelle in USERAF (602DH-602EH) geschrieben werden.

5.4.2 Definition des Bildschirmfensters

Durch den US-Code (1FH) wird eine Sequenz eingeleitet, mit der die Cursorposition verändert und die Definition des Bildschirmfensters vorgenommen werden kann. Es sind drei PUTC\$-Aufrufe erforderlich, und zwar mit der Codefolge US/n1/n2. Das Ergebnis hängt von den n1 und n2 zugeordneten Werten ab.

5.4.2.1 Definieren des unteren Randes (N1, N2 (10H, 19H))

Die Zeilennummer ergibt sich aus $10 \times n1 + n2$, wobei n1 und n2 den vier niederwertigen Bits (LSB) von N1 und N2 entsprechen.

Beispiel:

Um Zeile 21 als unteren Bildschirmrand zu definieren, müssen Sie folgende PUTC\$-Aufrufe vornehmen:

1. Aufruf: 1FH (US-Code)
2. Aufruf: 14H (Zeile 2×10)
3. Aufruf: 11H (+ 1)

5.4.2.2 Definieren des oberen Randes (N1, N2 (20H, 29H))

Die Zeilennummer ergibt sich aus $10 \times n1 + n2$, wobei n1 und n2 die vier niederwertigen Bits (LSB) von N1 und N2 entsprechen.

5.4.2.3 Cursor-Positionierung (N1, N2 «40H, 7H»)

Die Cursorposition ergibt sich aus den Werten für n1 (0–24) und n2 (1–40), wobei n1 und n2 den sechs niederwertigen Bits (LSB) von N1 und N2 entsprechen.

Beispiel:

Um den Cursor in die 3. Zeile, Spalte 20 zu stellen, müssen Sie folgende PUTCS\$-Aufrufe vornehmen:

1. Aufruf: 1FH (US-Sequenz)
2. Aufruf: 43H (Zeile 3)
3. Aufruf: 54H (Spalte 20 = 14H)

Eine Positionierung des Cursors an den Anfang einer bestimmten Zeile können Sie auf der Basis der Codes (N1, N2 «30H, 39H») erreichen. Die Zeilennummer ergibt sich aus $10 \times n1 + n2$, wobei n1 und n2 den vier niederwertigen Bits (LSB) von N1 und N2 entsprechen.

5.4.3 SS2-Sequenz

Mit der SS2-Sequenz lassen sich akzentuierte Sonderzeichen anzeigen. Diese Sonderzeichen sind zusammengesetzt aus einem Zusatzzeichen und einem Kleinbuchstaben. Die Sequenz der PUTCS\$-Aufrufe lautet:

1. Aufruf: 16H (SS2-Code)
2. Aufruf: 41H für ´
- 42H für `
- 43H für ^
- 48H für ¨
- 4BH für ç
3. Aufruf: XXH (Kleinbuchstaben-Code)

Ist XXH kein gültiger Code eines Kleinbuchstaben, so wird der entsprechende Buchstabe dargestellt und der Akzent überschrieben.

Zusätzlich enthält der G2-Zeichensatz, der mit Hilfe der SS2-Sequenz dargestellt wird, noch Sonderzeichen, die als Erweiterung des G0-Satzes anzusehen sind. Diese Zeichen sind:

- 23H : Pfund Sterling
- 24H : Dollar
- 26H : #
- 2CH : Pfeil links
- 2DH : Pfeil aufwärts
- 2EH : Pfeil rechts
- 2FH : Pfeil abwärts
- 30H : Grad-Zeichen
- 31H : Plus/Minus-Zeichen
- 38H : Divisions-Zeichen
- 3CH : ¼-Zeichen
- 3DH : ½-Zeichen
- 3EH : ¾-Zeichen
- 6AH : Œ-Zeichen
- 7AH : œ-Zeichen

5.4.4 TELETEL-Zeichen (Semigrafik)

Wird der PUTCS-Routine der SO-Code 0EH übergeben, so schaltet diese in den TELETEL-Modus um. In diesem Modus wird ein Blockgrafik-Zeichensatz (G1-Satz) dargestellt. Die 8×8-Punkt-matrix der normalen Zeichendarstellung wird hierzu in 6 Blöcke aufgeteilt:

	0	1	2	3	4	5	6	7
0								
1	Block 0				Block 1			
2								
3	Block 2				Block 3			
4								
5								
6	Block 4				Block 5			
7								

Zu jedem Block gehört ein bestimmtes Bit innerhalb des Code-Bytes. Ist das Bit gesetzt, so erhält der Block die Zeichenfarbe (Vordergrund), ist es rückgesetzt, so erhält der Block die Hintergrundfarbe.

	0	x	1	x	x	x	x	x
Bits	6	4	3	2	1	0		

Die Bits 7 und 5 beeinflussen das Zeichen – die Blockkombination – nicht, müssen sich aber in dem gezeigten Zustand befinden (Bit 7 rückgesetzt, Bit 5 gesetzt).

5.4.5 Escape-Sequenz

Mit dieser Sequenz werden Farbattribute, diverse Darstellungsattribute wie Scroll-Modus, Zeichengröße usw. und Mischbild definiert. Zwei Arten gilt es hier zu unterscheiden: Definition für Einzelzeichen und Definition für den gesamten Bildschirm (full screen). Zeichenweise Steuerung erreicht man durch die Sequenz ESC/ATT, für den Fullscreen-Modus benötigt man die Folge ESC/#!/Space/ATT; dabei leitet ESC (1BH) die Escape-Sequenz ein, # entspricht dem Code 23H, Space (20H) ist der Code der Leertaste und ATT ist der Code des gewählten Attributes. Die ATT-Codes für Zeichen- und Fullscreen-Modus sind identisch, doch haben verschiedene Attribute wie Rahmenfarbe usw. (ausgenommen Inversdarstellung) in einer Fullscreen-Sequenz keine Bedeutung.

5.4.5.1 Farbattribute

Bei den Farbattributen (ATT = «40H, 47H» oder «50H, 57H» oder «60H, 67H») bestimmen die vier höherwertigen Bits (MSB) den Teil des Bildschirms, der beeinflußt werden soll, die vier niederwertigen Bits (LSB) wählen eine der 8 Grundfarben:

- ★ MSB: 4 → Vordergrundfarbe
- 5 → Hintergrundfarbe
- 6 → Rahmenfarbe

★ LSB: 00 → schwarz
 01 → rot
 02 → grün
 03 → gelb
 04 → blau
 05 → violett
 06 → hellblau
 07 → weiß

Die 8 blassen Farben erreichen Sie, indem Sie zu den Codes «70H» (Vordergrund), «78H» (Hintergrund) und «80H» (Rahmen) folgende Werte addieren:

00 → grau
 01 → blaßrot
 02 → blaßgrün
 03 → blaßgelb
 04 → blaßblau
 05 → blaßviolett
 06 → blaßhellblau
 07 → orange

Beispiel:

Durch die folgende Sequenz erhalten Sie einen grünen Hintergrund:

PUTC\$	EQU	\$E803	
	LDB	#\$1B	Escape-Code
	JSR	PUTC\$	
	LDB	#\$52	5 = Hintergrund, 2 = grün
	JSR	PUTC\$	

Einen orangefarbenen Rahmen erhalten Sie mit:

	LDB	#\$1B	Escape-Code
	JSR	PUTC\$	
	LDB	#\$87	8 = blasser Rahmen, 7 = orange
	JSR	PUTC\$	

5.4.5.2 Sonstige Attribute

ATT-Codes <4CH, 4FH>, 58H, 5FH> und <68H,6EH>

- 4CH → Zeichengröße normal
- 4DH → Zeichen doppelt hoch, normal breit
- 4EH → Zeichen doppelt breit, normal hoch
- 4FH → Zeichen doppelt breit, doppelt hoch
- 58H → Zeichen maskieren
- 5FH → Zeichen demaskieren
- 5CH → inverse Darstellung
- 68H → Schreiben eines Zeichens ohne Farbänderung
- 69H → Schreiben eines Zeichens mit Farbänderung
- 6AH → normaler Scroll-Modus
- 6BH → Page-Modus (Scroll-Modus abgeschaltet)
- 6CH → Einschalten des Mischbildes
- 6DH → Ausschalten des Mischbildes
- 6EH → langsamer Scroll-Modus

Codes außerhalb des Bereiches von 40H bis 6EH haben in einer Escape-Sequenz keine Wirkung. Ignoriert werden auch Zeichen-Codes, wenn sie in einer Fullscreen-Sequenz angewandt werden.

Im Maskierungs-Modus werden alle Zeichen in schwarz/schwarz dargestellt. Wird die Demaskierung in einer Fullscreen-Sequenz eingeschaltet, so erscheinen alle maskiert dargestellten Zeichen in der Farbe, die vor dem Einschalten der Maskierung gewählt war.

Im Page-Modus initialisieren die Codes 0CH, 1EH und 1FH erneut die Attribute für Farben und Zeichengröße, sofern diese nicht durch eine Fullscreen-Sequenz festgelegt wurden. Farbe, Maskierung und Zeichengröße können auch zeilenweise umgeschaltet werden, wenn keine Definition über eine Fullscreen-Sequenz vorgenommen wurde.

Beispiel:

Durch die folgende Sequenz werden die Farben innerhalb des definierten Bildschirmfensters umgekehrt.

PUTC\$	EQU	\$E803	
	LDB	#\$1B	Aufruf der Escape-Sequenz
	JSR	PUTC\$	
	LDB	#\$23	erster Code für «Fullscreen»
	JSR	PUTC\$	
	LDB	#\$20	zweiter Code für «Fullscreen»
	JSR	PUTC\$	
	LDB	#\$5C	Code für inverse Darstellung
	JSR	PUTC\$	

6.0 Tastatur

6.1 Schnelle Tastaturabfrage

Einsprungsadresse: KTST\$ (E809H)

Ausgangsparameter: in 6809-Register CC

Diese Routine führt eine schnelle Tastaturabfrage durch. Bei der Rückkehr aus der Routine zeigt das Carry-Flag im CC-Register an, ob eine Taste betätigt wurde. Carry = 1: Taste gedrückt; Carry = 0: keine Taste gedrückt.

6.2 Tastaturdecodierung

Einsprungsadresse: GETC\$ (E806H)

Eingangsparameter: in Register PTCLAV (60CDH-60CEH), BUZZ (6073H)

Ausgangsparameter: in 6809-Register B und CC

Diese Routine führt die Tastaturabfrage und die Decodierung der Eingaben durch. Zusätzlich wird die Autorepeat-Funktion unterstützt. Der Signalton, der bei der Betätigung einer Taste erzeugt wird, kann durch einen Wert > 0 im Register BUZZ unterdrückt werden.

Das Prozessor-Register B enthält bei der Rückkehr aus der Routine den ASCII-Code der betätigten Taste. B hat den Wert 0, wenn keine Taste, SHIFT oder CONTROL gedrückt wurde. Diese beiden Tasten wirken nur in Verbindung mit einer Zeichentaste, wobei über SHIFT die Großbuchstaben bzw. Sonderzeichen angesprochen werden. Bei Anwendung von CONTROL wird Bit 6 des erzeugten ASCII-Codes auf 0 gesetzt.

Die Erzeugung von Akzent-Zeichen (Umlaute) erfolgt im Normalfall durch die Betätigung von drei Tasten:

1. Taste ACC,
2. SHIFT-Taste in Verbindung mit der betreffenden Akzent-Taste,
3. den gewünschten Kleinbuchstaben.

Für die am häufigsten benötigten deutschen Umlaute steht jedoch auch eine einfachere Methode zur Verfügung:

1. Taste ACC, danach
2. – Taste 6 für ä
– Taste 7 für ö
– Taste 8 für ü

Obwohl die Eingabe der Zeichen hierdurch auf die Betätigung zweier Tasten reduziert wurde, müssen zum Auslesen der Codes trotzdem drei Tastaturabfragen über GETC\$ erfolgen:

1. Aufruf: liefert den ACC-Code (16H),
2. Aufruf: liefert den Code des Akzentes,
3. Aufruf: liefert den Code des Kleinbuchstaben.

Die in B ausgegebenen ASCII-Codes werden einer Tabelle entnommen, deren Anfangsadresse im RAM-Register PTCLAV (60CDH-60CEH) abgelegt ist. Sollten Sie die Tastaturbelegung geändert haben (siehe auch 4.1), so kann die ACC-Sequenz zu unvorhersehbaren Ergebnissen führen.

7.0 Hochauflösende Grafik

7.1 Bildpunktsteuerung

Einsprungsadresse: PLOT\$ (E80FH)

Eingangsparameter: in 6809-Register X und Y, sowie Register FORME (6038H), CHDRAW (6041H), COLOUR (603BH), STATUS (6019H)

Ausgangsparameter: in PLOTX (603DH-603EH) und PLOTY (603FH-6040H)

7.1.1 Grafikmodus

Mit Hilfe dieser Routine kann ein einzelner Bildpunkt angesprochen werden. Die Lage des Punktes wird durch die Koordinaten in den Registern X (0,319) und Y (0,199) festgelegt. Auch die gewünschte Farbe läßt sich hierbei über das Register FORME (6038H) wählen, indem der Farbcode (-8 bis +15) hier abgelegt wird. Ist dieser Code negativ, so handelt es sich um eine Hintergrundfarbe und das entsprechende Bit wird im Zeichenspeicher rückgesetzt. Bei einem positiven Code ist die Vordergrundfarbe angesprochen und das Bit im Zeichenspeicher wird gesetzt. Ist Bit 4 des STATUS-Registers gesetzt, so erfolgt in jedem Falle nur eine Aktualisierung des Zeichenspeichers, der Farbspeicher bleibt unverändert.

Die Codes der Vorder- und Hintergrundfarben sind:

Farbe	Vordergrund	Hintergrund
Schwarz	0	- 1
Rot	1	- 2
Grün	2	- 3
Gelb	3	- 4
Blau	4	- 5
Violett	5	- 6
Hellblau	6	- 7
Weiß	7	- 8
Grau	8	9
Blaßrot	9	- 10
Blaßgrün	10	- 11
Blaßgelb	11	- 12
Blaßblau	12	- 13
Blaßviolett	13	- 14
Blaßhellblau	14	- 15
Orange	15	- 16

Der Code einer Hintergrundfarbe läßt sich errechnen, indem zu dem Vordergrund-Code die Zahl 1 addiert, und das Ergebnis negiert wird.

Im Register CHDRAW muß bei der Einzelpunktansteuerung der Wert 0 stehen, denn sonst arbeitet die Routine im Zeichenmodus (siehe auch 7.1.2 und 7.2.2).

7.1.2 Zeichenmodus

Ist der Inhalt des Registers CHDRAW größer als 0, so nimmt PLOT\$ an, daß es sich um einen ASCII-Code handelt und verzweigt automatisch zu der Routine CHPL\$ (E833H); diese zeigt das entsprechende Zeichen (8×8-Matrix) an. Bei der Angabe der Koordinatenpunkte müssen die Registerwerte in diesem Falle zwischen «1,40» für X und «0,24» für Y liegen.

Im Zeichenmodus wird der Farbcode nicht aus dem Register FORME, sondern aus dem Register COLOUR entnommen, das die aktuelle Zeichen- und Hintergrundfarbe enthält.

In beiden Modi werden die Inhalte der Register X und Y in die RAM-Register PLOTX (603DH-603EH) und PLOTY (603FH-6040H) kopiert, die die jeweils letzte angesprochene Koordinate enthalten.

7.2 Darstellung einer Linie

Einsprungsadresse: DRAW\$ (E80CH)

Eingangsparameter: 6809-Register X und Y,
die RAM-Register, PLOTX (603DH-603EH),
PLOTY (603FH-6040H), CHDRAW (6041H),
FORME (6038H), COLOUR (603BH),
STATUS (6019H)

Ausgangsparameter: RAM-Register PLOTX and PLOTY

7.2.1 – im Grafikmodus

Diese Routine gleicht dem LINE-Befehl des BASIC. Sie erzeugt eine Linie zwischen der Anfangskoordinate, die in den RAM-Registern PLOTX und PLOTY steht, und der Endkoordinate, die in den Prozessor-Registern X und Y übergeben werden muß. Die Farbe

der Linie wird durch den Inhalt des Registers **FORME** (siehe 7.1.1) bestimmt.

Ist Bit 4 des **STATUS**-Registers (6019H) gesetzt, so wird der Inhalt des Farbspeichers nicht verändert. Um im Grafikmodus zu arbeiten, muß **CHDRAW** den Wert 0 enthalten.

7.2.2 – im Zeichenmodus

Ist der Inhalt des **RAM**-Registers **CHDRAW** nicht gleich 0, so nimmt die Routine **DRAW\$** an, daß es sich um den ASCII-Code eines Zeichens handelt. Die Farbe wird in diesem Falle vom Inhalt des Registers **COLOUR** bestimmt und die in **X** und **Y** übergebenen Koordinaten müssen zwischen (1,40) und (0,24) liegen.

In beiden Modi werden die Inhalte der Register **X** und **Y** in die **RAM**-Register **PLOTX** (603DH-603EH) und **PLOTY** (603FH-6040H) kopiert, die die jeweils letzte angesprochene Koordinate enthalten.

7.3 Horizontalvektoren

Wird eine waagrechte Linie gezeichnet, so wird ein besonderer, schneller Algorithmus angewandt, bei dem nicht jeder Einzelpunkt berechnet, sondern jeweils acht Punkte (1 Block) gleichzeitig geschrieben werden. Dies ist sehr hilfreich bei grafischen Füllfunktionen wie sie z.B. in den **BASIC**-Befehlen **BOXF** und **PAINT** angewandt werden.

7.4 Auslesen der Farbe eines Bildpunktes

Einsprungsadresse: **GETP\$** (E821H)

Eingangsparameter: 6809-Register **X** und **X**

Ausgangsparameter: 6809-Register **B**

Diese Routine ermittelt die Farbe eines Bildpunktes. Die Farbcodes sind 8 bis 1, wenn der Punkt eine Hintergrundfarbe hat, bzw. 0 bis 15, wenn er eine Vordergrundfarbe hat. Die Ausgabe des Farbcodes erfolgt über das Prozessor-Register **B**.

Die Koordinaten des Bildpunktes werden über die Register **X** (0,319) und **Y** (0,199) festgelegt. Die den Farbcodes entsprechenden Farben sind in Abschnitt 7.1.1 beschrieben.

8.0 Auslesen von Zeichen aus dem Bildschirm

Einsprungsadresse: GETS\$ (E824H)

Eingangsparameter: 6809-Register A und X

Ausgangsparameter: 6809-Register B

Diese Routine liefert in Prozessor-Register B den ASCII-Code des Zeichens, dessen Koordinaten durch die Register X (1,40) und A (0,24) bestimmt worden sind. Findet die Routine, an der durch X und A angegebenen Stelle, kein gültiges ASCII-Zeichen, sondern z.B. ein Grafiksymbol, so erhält B den Wert 0.

Im anderen Falle ergeben sich folgende zwei Möglichkeiten:

8.1 Normale Zeichen

Ist das erkannte Zeichen ein Element des GO-Zeichensatzes, steht der entsprechende ASCII-Code im Register B.

8.2 Umlaute und akzentuierte Zeichen

In diesem Fall sind drei Aufrufe der GETS\$-Routine notwendig:

1. Aufruf: Register B liefert den Wert 16H (ACC-Code)
2. Aufruf: B enthält den Code des Akzents
3. Aufruf: B enthält den ASCII-Code des Kleinbuchstabens

9.0 Tonerzeugung

Einsprungsadresse: NOTE\$ (E81EH)

Eingangsparameter: 6809-Register B
RAM-Register OCTAVE (66036H-6037H),
DUREE (6033H-6034H), TEMPO (6031H-
6032H), TIMBRE (6035H)

Der Parameter für die zu spielende Note wird über das Prozessor-Register B eingegeben. 13 Grundnoten von DO bis UT und eine Pause können erzeugt werden. Die Codes hierfür entnehmen Sie bitte der nachfolgenden Tabelle:

Note	Code
Pause	30H
DO	31H
DO #	32H
RE	33H
RE #	34H
MI	35H
FA	36H
FA #	37H
SO	38H
SO #	39H
LA	3AH
LA #	3BH
SI	3CH
UT	3DH

Folgende Parameter sind vorzugeben: Tempo, Oktave, Klangfarbe und Länge.

Oktave:

5 Oktaven sind möglich, wobei Oktave 1 die tiefste, Oktave 5 die höchste ist. Dem Kammerton A (440 Hz) entspricht das LA in der 4. Oktave. Die im RAM-Register OCTAVE abzulegenden Werte sind:

Oktave	Wert
1	16
2	08
3	04
4	02
5	01

Dauer:

Die Dauer jeder einzelnen Note kann relativ zur generellen Abspielgeschwindigkeit zwischen $\frac{1}{4}$ und $\frac{1}{32}$ festgelegt werden. Der in DUREE abzulegende Wert beträgt 96 für eine ganze Note. Kürzere Noten erhält man durch Teilung dieses Wertes durch 2er oder 3er Potenzen für normale und punktierte Noten oder Triolen.

Note	Wert
$\frac{1}{1}$	96
$\frac{1}{2}$ punkt.	72
$\frac{1}{2}$	48
$\frac{1}{4}$ punkt.	36
$\frac{1}{4}$	24
$\frac{1}{8}$ punkt.	18
$\frac{1}{8}$	12
$\frac{1}{16}$ punkt.	09
$\frac{1}{16}$	06
$\frac{1}{32}$ punkt.	05
$\frac{1}{32}$	03

Bei einer Triole	Wert
$\frac{1}{4}$	16 (3×16 = 48)
$\frac{1}{8}$	08 (3× 8 = 24)
$\frac{1}{16}$	04 (3× 4 = 12)
$\frac{1}{32}$	02 (3× 2 = 06)

Tempo:

Der Wert im RAM-Register TEMPO «1,255» bestimmt die Gesamtgeschwindigkeit, sodaß sich die wirkliche Spieldauer einer Note aus der Kombination der Werte in «DUREE» und «TEMPO» ergibt.

Klang:

Die Schwingungsform und dadurch der Klang des erzeugten Tones läßt sich über einen im RAM-Register abzulegenden Wert zwischen 0 und 5 beeinflussen. Einen ungedämpften Dauerton ergibt den Wert 0.

10.0 Abfrage der Joysticks

Einsprungsadresse: JOYS\$ (E827H)

Eingangsparameter: 6809-Register A

Ausgangsparameter: 6809-Register B und CC

Zur Auswahl des abzufragenden Joysticks wird der Wert 0 oder 1 im 6809-Akkumulator A übergeben. Das B-Register liefert bei

Rückkehr aus der Routine einen der Stellung des Joysticks entsprechenden Code laut folgender Tabelle:

- 0 → Mitte
- 1 → Nord
- 2 → Nordost
- 3 → Ost
- 4 → Südost
- 5 → Süd
- 6 → Südwest
- 7 → West
- 8 → Nordwest

War der «Action»-Knopf des Joysticks gedrückt, so ist das Carry-Flag im CC-Register gesetzt, ansonsten ist das Bit rückgesetzt.

11.0 Lightpen

11.1 Test des Lightpen-Schalters

Einsprungsadresse: LPIN\$ (E81BH)

Ausgangsparameter: 6809-Register CC

Diese Routine überprüft, ob der Schalter an der Spitze des Lightpens geschlossen ist. Das Ergebnis liefert das Carry-Flag im 6809-Register CC. Ist das Flag gesetzt, so ist der Schalter betätigt, im anderen Fall ist das Bit rückgesetzt.

11.2 Auswertung der Lightpen-Signale

Einsprungsadresse: GETL\$ (E818H)

Ausgangsparameter: 6809-Register X, Y und CC

Die Routine liest die Koordinaten des Punktes ein, auf den der Lightpen gerichtet ist, wobei die Daten der Abszisse «0,319» im X-Register, die Daten der Ordinate «0,199» im Y-Register übergeben werden. Sind die Koordinaten durch zu geringe Bildschirmhelligkeit nicht lesbar oder ist der Lightpen nicht auf den Bildschirm gerichtet, so ist das Carry-Flag im CC-Register gesetzt; nach einem gültigen Lesevorgang ist es rückgesetzt.

12.0 Schnittstellen-Adapter

Einsprungsadresse: RSCO\$ (E812H)

Eingangsparameter: 6809-Register B, RAM-Register RS.OPC (602BH), BAUDS (6044H-6045H), NOMBRE (6046H) und GRCODE (6047H)

Ausgangsparameter: 6809-Register CC, RAM-Register RS.STA (602CH)

Diese Routine dient zur Verwaltung des Schnittstellenadapters. Die auszuführende Funktion wird ausgewählt, indem der Funktions- (OP-) Code in dem RAM-Register RS.OPC abgelegt wird. Folgende Funktionen sind ausführbar:

Inhalt in RS.OPC	Funktion
% 0 0 0 0 0 0 0 1	Öffnen im Schreib/Lese-Modus (RS-232)
% 0 0 0 0 0 0 1 0	Zeichen einlesen (RS-232)
% 0 0 0 0 0 1 0 0	Öffnen im Schreib-Modus (RS-232)
% 0 0 0 0 1 0 0 0	Zeichen schreiben
% 0 0 0 1 0 0 0 0	Schließen
% 0 0 1 0 0 0 0 0	Screenprint-Funktion
% 0 1 0 0 0 0 0 0	Öffnen im Parallel-(Centronics-) Modus

Die Übertragung der Schreib- und Lesedaten erfolgt über Register B.

Wählt man die Screenprint-Funktion an, die einen Ausdruck des Bildschirminhaltes veranlaßt, so muß im RAM-Register GRCODE der jeweils druckerabhängige Umschaltcode für den Grafikmodus stehen. Der Standardwert 7 wird hier von System vorgegeben.

Den Status, in dem sich der Schnittstellen-Adapter befindet, erhält man durch die Abfrage des Bytes in dem RAM-Register RS.STA:

Inhalt in RS.STA	Zustand
% 0 0 0 0 0 0 0 1	Geöffnet im Schreib/Lese-Modus (RS-232)
% 0 0 0 0 0 1 0 0	Geöffnet im Schreib-Modus (RS-232)
% 0 0 0 1 0 0 0 0	Geschlossen
% 0 1 0 0 0 0 0 0	Geöffnet im Parallel-(Centronics-) Modus
% 1 0 0 0 0 0 0 0	Peripheriegerät nicht bereit

Das Carry-Flag im CC-Register gibt an, ob ein Fehler bei der Ausführung der Funktion aufgetreten ist. Carry = 1: Fehler, Carry = 0: kein Fehler.

Auch die anderen, bei einer seriellen Übertragung wichtigen Parameter werden über RAM-Register angegeben. So enthält NOMBRE einen Code, der die Anzahl der Datenbits angibt: 80H für 8 Bit-, 40H für 7 Bit-Datenworte. Der Parameter für die Übertragungsrate muß in BAUDS abgelegt werden. Die Parameter aller beim TO7-70 möglichen Geschwindigkeiten stehen mit einer Länge von jeweils zwei Bytes in der Tabelle BDTAB (E836H) und können dort entnommen werden. Die Reihenfolge innerhalb der Tabelle entspricht 110, 300, 600, 1200, 2400 und 4800 Baud.

13.0 Steuerung des Programmrekorders

Einsprungsadresse: K7C0\$ (E815H)

Eingangsparameter: 6809-Register B und RAM-Register K7.OPC (6029H)

Ausgangsparameter: 6809-Register B und CC, RAM-Register K7.STA (602AH)

Mit Hilfe dieser Routine ist es möglich, Daten auf den Programmrekorder zu schreiben oder von dort zu lesen. Die auszuführende Funktion wird durch den im RAM-Register K7.OPC abgelegten OP-Code bestimmt:

Inhalt in K7.OPC	Funktion
% 0 0 0 0 0 0 1	Öffnen im Lese-Modus
% 0 0 0 0 0 1 0	Zeichen einlesen
% 0 0 0 0 1 0 0	Öffnen im Schreib-Modus
% 0 0 0 1 0 0 0	Zeichen schreiben
% 0 0 1 0 0 0 0	Schließen

Die Datenbytes bei Schreib- und Lesevorgängen werden durch den Prozessor-Register B übergeben.

Wurde eine Funktion durchgeführt, so enthält das RAM-Register K7.STA einen Code, der den Status des Programmrekorders angibt:

Inhalt K7.STA	Zustand
% 0 0 0 0 0 0 0 1	Geöffnet im Lese-Modus
% 0 0 0 0 0 1 0 0	Geöffnet im Schreib-Modus
% 0 0 0 1 0 0 0 0	Geschlossen
% 1 0 0 0 0 0 0 0	Recorder nicht bereit

Trat ein Fehler bei der Ausführung einer Funktion auf oder ist der Programmrekorder nicht angeschlossen, so wird zur Fehleranzeige das Carry-Bit im CC-Register gesetzt; trat kein Fehler auf, so ist es rückgesetzt.

14.0 Steuerung der Diskettenlaufwerke

Prinzipiell gibt es zwei verschiedene Möglichkeiten, um Schreib- und Lesevorgänge auf der Diskette durchzuführen:

- logische Dateiverwaltung über das Microsoft (R) – BASIC;
- physischer Zugriff über die Monitor-Routinen.

Um nochmals daran zu erinnern: Jede Diskette ist in 40 Spuren zu je 16 Sektoren unterteilt. Abhängig von der verwendeten Schreibdicke enthält ein Sektor 128 (single density) oder 256 Bytes (double density). Die Schreibdichten werden durch Programmierung des Controllers umgeschaltet, wobei er nach dem Einschalten vom System auf double density (doppelte Schreibdichte) eingestellt wird.

14.1 Physischer Diskettenzugriff

Einsprungadresse: DKCO\$ (E82AH)

Eingangsparameter: DK.OPC (6048H), DK.DRV (6049H), DK.SEC (604CH) DK.TRK (604AH-604BH), DK.BUF (604FH-6050H)

Ausgangsparameter: 6809-Register CC,
RAM-Register DK.STA (604EH)

Das Betriebssystem prüft nach dem Einschalten des Gerätes, ob ein Disk-Controller angeschlossen ist. Hierzu dient das Register DK.FLG (6080H), das den Wert FFH enthält, wenn der Controller vorhanden ist, anderenfalls enthält es den Wert 0.

Das Auftreten eines Fehlers während einer Diskettenoperation wird über das CC-Register angezeigt. Das Carry-Flag ist gesetzt, wenn ein Fehler auftrat, anderenfalls ist es rückgesetzt.

Durch Übergabe eines bestimmten Funktionscodes an das Register DK.OPC kann eine der folgenden Funktionen ausgewählt werden:

Code 01: Initialisierung des Controllers.

Trat bei der Initialisierung ein Fehler auf, wird das Carry-Flag des CC-Registers gesetzt. Bei Rückkehr aus der Routine enthält DK.STA einen Hinweis auf den Typ des angeschlossenen Controllers: «C» bei single density, «D» bei double density. Im Fehlerfall (Carry-Bit gesetzt) enthält DK.STA den Fehlercode 40H (siehe Fehlertabelle).

Code 02: Lesen eines Sektors.

Die möglichen Fehlercodes sind: 02, 04, 08, 10H, 80H. Tritt ein Fehler auf, so wird der entsprechende Code im Register DK.STA abgelegt.

Code 04: Umschalten auf einfache Schreibdichte.

Mit dieser Funktion wird der double density Controller auf einfache Schreibdichte umgeschaltet. Fehler können hierbei nicht auftreten.

- **Code 08: Schreiben eines Sektors.**

Die möglichen Fehlercodes sind: 01, 02, 04, 08, 10H, 20H, 80H. Tritt ein Fehler auf, so wird der entsprechende Fehlercode im Register DK.STA abgelegt.

- **Code 10H: Umschalten auf doppelte Schreibdichte.**

Mit dieser Funktion wird der double density Controller auf doppelte Schreibdichte umgeschaltet. Fehler können hierbei nicht auftreten.

- **Code 20H: Positionieren auf Spur 0.**

Der Schreib-/Lesekopf wird auf Spur 0 positioniert. Die möglichen Fehlercodes sind: 10H und 80H.

- **Code 40H: Positionieren auf eine bestimmte Spur.**

Diese Funktion positioniert den Schreib-/Lesekopf auf die Spur, deren Nummer im Register DK.TRK angegeben ist. Die möglichen Fehlercodes sind 10H und 80H.

- **Code 80H: Verify-Option.**

Die Verify-Funktion überprüft nach jedem Schreibvorgang durch eine logische «ODER»-Verknüpfung den Inhalt des geschriebenen Sektors mit dem Inhalt des Datenpuffers. Alle bisher genannten Fehlercodes und der Code 20H sind möglich.

Übergabe der Eingabeparameter:

★ Register DK.DRV (6049H): Dieses Register enthält die Nummer des Laufwerkes, auf dem die Funktion ausgeführt werden soll. Mögliche Werte sind 0 bis 3.

★ Register DK.TRK (604AH-604BH): Dieses Register enthält die Nummer der Spur, auf die zugegriffen werden soll. Möglich sind die Werte 0 bis 39.

★ Register DK.SEC (604CH): Dieses Register enthält bei Schreib- und Lesevorgängen die Nummer des jeweils angesprochenen Sektors. Mögliche Werte sind 1 bis 16.

★ Register DK.BUF (604FH-6050H): Dieses Register muß die Anfangsadresse des Disk-Buffers enthalten. Dieser Schreib-/Lese-Puffer muß 128 Bytes bei einfacher, 256 Bytes bei doppelter Schreibdichte umfassen.

★ Register DK.STA (604EH): Dieses Register muß nicht vom Anwender geladen werden. Bei Rückkehr aus der gewählten Funktion enthält es einen etwaigen Fehlercode oder – nach der Initialisierung – die Kennung des angeschlossenen Controller-Typs.

Hier nun die verschiedenen möglichen Fehlercodes:

– Code 01: Diskette schreibgeschützt.

Dieser Fehler kann nur auftreten, wenn versucht wurde, eine Schreiboperation auf einer schreibgeschützten Diskette durchzuführen.

– Code 02: Spur-Lesefehler.

Die Spur wurde zwar gefunden, die dort angegebene Spurnummer entspricht jedoch nicht der genannten Spur.

– Code 04: Sektor-Lesefehler.

Die Sektor-Kennung ist unkorrekt oder nicht lesbar, die Spur wurde jedoch richtig gefunden.

– Code 08: Datenfehler.

Es sind zuviel oder zuwenig Daten vorhanden oder es liegt ein Paritätsfehler vor. Spur und Sektor wurden richtig gefunden.

– Code 10H: Fehler im Disk-Laufwerk.

Das angegebene Laufwerk ist nicht vorhanden, der Motor dreht sich nicht oder der Index-Impuls wird nicht erkannt (keine Diskette im Laufwerk oder Verriegelung nicht geschlossen).

– Code 20H: Verify-Fehler.

Der Inhalt des gerade geschriebenen Sektors ist nicht identisch mit dem Inhalt des Pufferspeichers.

– Code 40H: Controller nicht bereit.

Der Disketten-Controller ist defekt oder nicht betriebsbereit.

– Code 80H: Diskette nicht formatiert.

Die Spurkennung ist nicht lesbar, die Diskette nicht formatiert.

14.2 Microsoft(R) BASIC-Format

Um eine durchgehende Kompatibilität zu gewährleisten, entsprechen alle vom T07-70 erzeugten Dateien dem Standardformat des Microsoft(R)-BASIC.

Die Spur 20 ist reserviert und dient dazu, ein Inhaltsverzeichnis (Directory) der Diskette mit folgender Aufteilung aufzunehmen:

★ Sektor 1: reserviert

★ Sektor 2: Spurbelegungstabelle der Dateien («FAT»)

★ Sektor 3–16: Inhaltsverzeichnis (Directory)

14.2.1 Dateibelegungs-Tabelle (FAT)

Die Diskette ist in Blöcke aufgeteilt, von denen jeder bei einfacher Schreibdichte 1KByte, bei doppelter Dichte 2KByte umfaßt. In beiden Fällen hat man somit je 2 Blöcke pro Spur, die von 0 an durchnummeriert sind. Jedes Byte der Dateibelegungstabelle, beginnend mit Byte 1, stellt also einen Datenblock dar.

Die Organisation des «FAT»:

- ★ Byte 0: 0
- ★ Byte 1: Block 0, Spur 0, Sektoren 1 bis 8
- ★ Byte 2: Block 1, Spur 0, Sektoren 9 bis 16
- ★ Byte 3: Block 2, Spur 1, Sektoren 1 bis 8
- ★ Byte 4: Block 3, Spur 1, Sektoren 9 bis 16
- ★ - - - - -
- ★ Byte $2i-1$: Block $2i-2$, Spur $i-1$, Sektoren 1 bis 8
- ★ Byte $2i$: Block $2i-1$, Spur $i-1$, Sektoren 9 bis 16
- ★ - - - - -
- ★ Byte 80: Block 79, Spur 39, Sektoren 9 bis 16

Die innerhalb dieser Bytes – von denen jedes einen physischen Datenblock darstellt – abgelegten Werte geben Auskunft darüber, ob und in welcher Form dieser Datenblock einer bestehenden Datei zugeordnet ist. Die Werte und ihre Bedeutung sind:

- ★ FFH – Dieser Block ist frei (nicht zugeordnet)
- ★ FEH – Dieser Block ist reserviert. Er enthält das Boot-Programm, das Directory oder – bei System-Disketten – das Disk-BASIC.
- ★ Werte von 0 bis BFH zeigen an, daß dieser Block durch eine Datei belegt ist. Der Wert gibt die Nummer des nächsten, zu dieser Datei gehörenden Blocks an.
- ★ Werte von C1H bis C8H zeigen an, daß dieses der letzte Block einer Datei ist. Die Werte geben ferner einen Hinweis darauf, wieviele Sektoren innerhalb dieses letzten Blockes von der Datei belegt sind.

14.2.2 Inhaltsverzeichnis (Directory)

Das Inhaltsverzeichnis ist 14 Sektoren lang und jeder Dateieintrag belegt 32 Bytes. Bei einfacher Schreibdichte nimmt jeder Sektor 4, bei doppelter Schreibdichte 8 Einträge auf. Daraus ergibt sich ein Fassungsvermögen von 56 Dateieinträgen bei einfacher, 112 Einträgen bei doppelter Schreibdichte.

Jeder Eintrag ist nach folgendem Schema aufgebaut:

Bytes 00H-07H:	Dateiname, linksbündig geschrieben, mit Leerzeichen aufgefüllt.
Bytes 08H-0AH:	Zusatzname (.BAS, .BIN, etc...), linksbündig geschrieben, mit Leerzeichen aufgefüllt.
Byte 0BH:	Dateityp: 0 = BASIC-Programm (ASCII oder Binär) 1 = BASIC-Datei (Daten in ASCII-Form) 2 = Programm in Maschinencode (Binär) 3 = Assembler-Textdatei (ASCII-Datei)
Byte 0CH:	ASCII-Flag: FFH = ASCII, 00 = Binär
Byte 0DH:	Nummer des ersten logischen Blocks der Datei.
Bytes 0EH-0FH:	Anzahl der belegten Bytes im letzten Sektor der Datei
Bytes 10H-1FH:	reserviert

Das erste Byte eines jeden Eintrages hat noch eine weitere Funktion: es gibt den Status der Datei an. 00 bedeutet, daß dieser Eintrag frei ist, während die Werte 20H bis 7FH den ASCII-Code des ersten Buchstaben des Dateinamens repräsentieren. Ein FFH an dieser Stelle zeigt das logische Ende des Directory an. Beim Aufbau des Inhaltsverzeichnisses, also beim Formatieren der Diskette, werden alle Einträge mit FFH gefüllt. Wird nun ein Eintrag belegt, so wird das logische Ende des Directory um jeweils einen Eintrag weiter nach hinten geschoben, bis das Inhaltsverzeichnis voll ist. Ein freier Eintrag (00) entsteht, wenn eine bestehende Datei gelöscht wird. Wird nun eine neue Datei eingerichtet, so wird zunächst dieser freie Eintrag benutzt, statt das Directory unnötig zu vergrößern.

15.0 Allgemeine Informationen

15.1 Speicheraufteilung (Memory Map)

Adressen (Hexadezimal)

0000-3FFF	ROM-Modul
4000-5FFF	2×8 KByte des Bildschirmspeichers
6000-60FF	RAM-Register des Monitors
6100-9FFF	fester Benutzerspeicher, 16K
A000-DFFF	2 16K-Bereiche (RAM-Banks), umschaltbar
A000-DFFF	4 16K-RAM-Banks in der Speichererweiterung
E000-E7BF	1.9 K ROM für Disketten-Routinen
E7C0-E7C7	Adressen des System-PIA 6846
E7C8-E7CB	Adressen des System PIA-6821
E7CC-E7CF	Adressen des PIA 6821 im Musik-Spieleinterface
E7D0-E7DF	Adressen des Disk-Controllers
E7E0-E7E3	Adressen des PIA 6821 im Kommun.-Interface
E7E4-E7E7	Lightpen-Zähler
E7E8-E7FF	frei für Erweiterungen
E800-FFFF	6KByte Monitor

In dem 16K-Bereich von A000H bis DFFFFH liegen mehrere RAM-Banks bildlich gesprochen «hintereinander». Das Grundgerät des TO7-70 enthält neben dem festen 16K-Bereich von 6100H-9FFFFH zwei solcher Banks, also insgesamt 48KByte. Hiervon sind jeweils 32K zugänglich. Die Umschaltung der RAM-Banks erlaubt z.B. folgendes Unterprogramm:

Eingangsparameter: 6809-Register A (= Bank-Nummer 0 bis 5)

BNKSW	EQU	★
	PSHS	D, X, U
	LDU	#\$E7C0
	LDB	11, U
	ANDB	#\$FB0
	STB	11, U
	LDX	# TAB
	LDA	A, X
	STA	9, U
	ORB	#\$04
	STB	11, U
	PULS	D, X, U, PC
TAB	EQU	★

15.2 Einsprungadressen der Standardroutinen

PUTC\$	(E803H)	– Darstellen eines Zeichens
GETC\$	(E806H)	– Lesen der Tastatur
KTST\$	(E809H)	– Schnelle Tastaturabfrage
DRAW\$	(E80CH)	– Darstellung einer Linie
PLOT\$	(E80FH)	– Anzeigen oder Löschen eines Grafikpunktes
RSCO\$	(E812H)	– Steuerung des Kommunikations-Interface
K7C0\$	(E815H)	– Schreiben/Lesen des Programmrekorders
GETL\$	(E818H)	– Einlesen der Koordinaten des Lightpens
LPIN\$	(E81BH)	– Testen des Lightpen-Schalters
NOTE\$	(E81EH)	– Erzeugen eines Tones
GETP\$	(E821H)	– Einlesen der Farbe eines Grafikpunktes
GETS\$	(E824H)	– Lesen eines Zeichens vom Bildschirm
JOYS\$	(E827H)	– Abfragen der Joysticks
DKC0\$	(E82AH)	– Steuerung des Disketten-Controllers
MENU\$	(E82DH)	– Rückkehr zum Hauptmenü
KBIN\$	(E830H)	– Abschluß eines Interrupt-Programmes
CHPL\$	(E833H)	– Punkt setzen im Zeichenmodus

15.3 Vom Monitor belegte RAM-Register (Zeropage)

Die nun folgenden Adressen sind im Hexadezimalsystem angegeben:

- ★ 6000-6018 (TERMIN): Zeilenende-Tabelle
- ★ 6019 (STATUS): diverse Flags:
 - b7: Semigrafik
 - b6: Fast-Scroll-Modus
 - b5: User-Interruptprogramm
 - b4: Punktgrafik ohne Farbe
 - b3: Flag der Tastaturabfrage
 - b2: Cursor ein/aus
 - b1: reserviert
 - b0: Flag für Autorepeat der Tastatur
- ★ 601A-601B (TABPT): Zeiger der Zeilenende-Tabelle
- ★ 601B (RANG): momentane logische Zeilennummer
- ★ 601C-601D (TOPTAB): Zeiger auf den Anfang der Zeilenende-Tabelle
- ★ 601D (TOPRAN): Erste logische Zeile des Bildschirmfensters

- ★ 601E-601F (BOTTAB): Zeiger auf das Ende der Zeilenende-Tabelle
- ★ 601F (BOTRAN): Letzte logische Zeile des Bildschirmfensters
- ★ 6020 (COLN): momentane logische Spalte
- ★ 6021-6022 (IRQPT): Adresse der IRQ-Interruptroutine
- ★ 6023-6024 (FIRQPT): Adresse der FIRQ-Interruptroutine
- ★ 6025-6026 (NMIPT): Adresse der NMI-Interruptroutine
- ★ 6027-6028 (TIMEPT): Adresse der Timer-Interruptroutine
- ★ 6029 (K7.OPC): OP-Code für Rekorderfunktionen
- ★ 602A (K7.STA): Statusbyte des Rekorders
- ★ 602B (RS.OPC): OP-Code für Kommunikations-Interface
- ★ 602C (RS.STA): Statusbyte des Kommunikations-Interface
- ★ 602D-602E (USERAF): Anfang der User-Zeichentabelle
- ★ 602F-6030 (SWI1): Adresse der SWI-Interruptroutine
- ★ 6031-6032 (TEMPO): Abspielgeschwindigkeit einer Melodie
- ★ 6033-6034 (DUREE): relative Länge einer Note (1–96)
- ★ 6035 (TIMBRE): Klangfaktor der Note (0 = Dauerton)
- ★ 6036-6037 (OKTAVE): Oktave (1, 2, 4, 8, oder 16)
- ★ 6038 (FORME): Farbcode für Grafikmodus (–8 bis +15)
- ★ 6039 (ATRANG): Flags zur Bildschirmsteuerung
 - b7: Scroll-Flag
 - b6: reserviert
 - b5: reserviert
 - b4: reserviert
 - b3: reserviert
 - b2: reserviert
 - b1: einfache oder doppelte Zeichenbreite
 - b0: einfache oder doppelte Zeichenhöhe

- ★ 603A (ATRSCR): Flags für den Fullscreen-Modus
b7: Fullscreen-Hintergrundflag
b6: Fullscreen-Vordergrundflag
b5: reserviert
b4: reserviert
b3: reserviert
b2: reserviert
b1: einfache oder doppelte Zeichenbreite
b0: einfache oder doppelte Zeichenhöhe
- ★ 603B (COLOUR): Aktuelle Vorder-/Hintergrundfarbe (Zeichen)
- ★ 603C (TELETL): FFH = Page-Modus (kein Scrollen)
- ★ 603D-603E (PLOTX): horiz. Position des letzten Grafikpunktes
- ★ 603F-6040 (PLOTY): vert. Position des letzten Grafikpunktes
- ★ 6041 (CHDRAW): Parameter für DRAW-Funktion:
ASCII-Code = Zeichenmodus,
0 = Grafikmodus
- ★ 6042 (CURSFL): Flag, verhindert unzulässige Zeilenverbindung
- ★ 6043 (COPCHR): FFH = BS und HT kopieren das momentane Zeichen
- ★ 6044-6045 (BAUDS): Übertragungsrate der seriellen Schnittstelle
- ★ 6046 (NOMBRE): Wortlänge der seriellen Schnittstelle
- ★ 6047 (GRCODE): Druckercode für Umschaltung in Grafikmodus
- ★ 6048 (DK.OPC): OP-Code für die Disketten-Funktionen
- ★ 6049 (DK.DRV): Nummer des angesprochenen Laufwerkes
- ★ 604A-604B (DK.TRK): Nummer der angesprochenen Spur
- ★ 604C (DK.SEC): Nummer des angesprochenen Sektors
- ★ 604D (DK.NUM): Faktor der Sektorenverflechtung
- ★ 604E (DK.STA): Statusbyte des Disk-Controllers
- ★ 604F-6050 (DK.BUF): Anfangsadresse des Disk-Buffers
- ★ 6051-6052 (TRACK0): Position des Schreib-/Lesekopfes in Drive 0
- ★ 6053-6054 (TRACK1): Position des Schreib-/Lesekopfes in Drive 1

- ★ 6055-6056 (TRACK2): Position des Schreib-/Lesekopfes in Drive 2
- ★ 6057-6058 (TRACK3): Position des Schreib-/Lesekopfes in Drive 3
- ★ 6059 (SEQUECE): Flag zur Kennung der Bildschirm-Steuersequenz
- ★ 605A-605B (SCRPT): aktuelle Cursorposition
- ★ 605C (SAVCOL): Sicherung des aktuellen Farbcodes
- ★ 605D (ASCII): Code des letzten dargestellten Zeichens
- ★ 605E (KEY): Code der letzten betätigten Taste
- ★ 605F (CMPTKB): Zähler des Tastatur-Autorepeats
- ★ 6060-6061 (STADR): Startadresse des aktuellen Bildschirm-speichers
- ★ 6062-6063 (ENDDR): Endadresse des aktuellen Bildschirm-speichers
- ★ 6064 (TCRSAV): Sicherung aktueller Timer-Zustand
- ★ 6065-6066 (TCTSAV): Sicherung aktueller Zählerstand des Timers
- ★ 6067 (LATCLV): Verzögerungsfaktor des Autorepeat
- ★ 6068-6069 (SAVATR): Sicherung der aktuellen Bildschirm-attribute
- ★ 606A (US1): Flag für US-Sequenz
- ★ 606B (COMPT): Zähler für Zeichenwiederholung
- ★ 606C-606D (TEMP): temporäres Register für Datenübertragung
- ★ 606E-606F (SAVEST): Sicherung des Stack-Pointers
- ★ 6070 (ACCENT): Flag für Akzent-Sequenz
- ★ 6071 (SS2GET): Flag zur Darstellung akzentuierter Zeichen
- ★ 6072 (SS3GET): Flag zur Darstellung akzentuierter Zeichen
- ★ 6073 (BUZZ): Flag für den Signalton-Summer
- ★ 6074 reserviert
- ★ 6075 (EFCMPT): Zähler für gelöschte Zeichen (DEL-Taste)
- ★ 6076-6077 (BLOCZ): ständig 0 für Initialisierungen
- ★ 6078 (SCROLS): Flag für Soft-Scroll
- ★ 6079-607E (TABCHX): Pointer der Menü-Auswahltablette
- ★ 607F (RUNFLG): Flag für Menüposition 2 (Auto-Start)

- ★ 0080 (DKFLG): Flag für Disk-Controller
- ★ 0081-00CC: Stack-System
- ★ 00CD-00CE (PTCLAV): Pointer der Tastaturtabelle
- ★ 00CF-00D0 (PTGENE): Pointer der Standardzeichentabelle
- ★ 00D1 (APPLIC): Prüfsumme des laufenden Anwendungsprogrammes
- ★ 00D2 (DECALG): Einstellung des Lightpen
- ★ 00D3-00EA (LPBUFF): Puffer für Lightpen-Lesedaten
- ★ 00EB-00FD reserviert
- ★ 00FE-00FF (TSTRST): Flag für Kalt- oder Warmstart

1b.4 E/A-Adressen

Nachfolgend sind die Adressen der Ein- und Ausgabebausteine angegeben. Um die logische Richtung der programmierbaren Datenleitungen zu kennzeichnen, wurden die Zeichen »E« für »Eingang« und »A« für »Ausgang« verwandt. Die Adressangaben erfolgen in Hexadezimalzahlen.

1b.4.1 System-PIA

★★★★★ SYSTEM-PIA 6846 ★★★★★

- ★ E7C0 (CSR): Zustandsregister
- ★ E7C1 (CRC): Kontrollregister
- CP2: Ausgabe Ton
- CT0: Schreibdaten (zum Rekorder)
- ★ E7C2 (DDRC): Datenrichtungs-Register
- ★ E7C3 (PRC): Datenregister
- b0 »A« Umschaltung Farb-/Bildpunktspeicher
- b1 »E« Lightpen-Schalter
- b2 »A« Flag für Rahmenfarbe »blau«
- b3 »A« LED der Tastatur
- b4 »A« Rahmenfarbe rot
- b5 »A« Rahmenfarbe grün
- b6 »A« Rahmenfarbe blau
- b7 »E« Lesen vom Programmrekorder
- ★ E7C5 (TCR): Kontrollregister Timer
- ★ E7C6-E7C7 (TMSB-TLSB): Zählerstand des Timers

★★★★★ SYSTEM-PIA 6821 ★★★★★

★ E7C8	(PRA):	Datenregister Port A
	b0-b7	⌋E⌋ Auslesen der Tastatur-Matrix
★ E7C9	(PRB):	Datenregister Port B
	b0-b2	⌋A⌋ Multiplexing der Tastatur
	b3-b7	⌋A⌋ Auswahl der RAM-Banks
★ E7CA	(CRA):	Kontrollregister Port A
	CA1	⌋E⌋ Mischbild angeschlossen
	CA2	⌋A⌋ Motorsteuerung des Rekorders
★ E7CB	(CRB):	Kontrollregister Port B
	CB1	⌋E⌋ Eingang Lightpen-Impulse
	CB2	⌋A⌋ Mischbild einschalten

15.4.2 PIA im Musik-/Spiele-Interface

★★★★★ PIA 6821 ★★★★★

★ E7CC	(PRA1):	Datenregister Port A
	b0-b7	⌋E⌋ Abfrage der Joystick-Zustände
★ E7CD	(PRB1):	Datenregister Port B
	b0-b5	⌋A⌋ D/A-Konverter für Ton
	b6	⌋E⌋ Action-Knopf Joystick 0
	b7	⌋E⌋ Action-Knopf Joystick 1
★ E7CE	(CRA1):	Kontrollregister Port A
	CA1	⌋E⌋ Action-Knopf Joystick 0
	CA2	⌋E⌋ Action-Knopf Joystick 1
★ E7CF	(CRB1)	Kontrollregister Port B

15.4.3 PIA im Kommunikations-Interface

★★★★★ PIA 6821 ★★★★★

★ E7E0	(PRA2):	Datenregister Port A
	b0	⌋A⌋ receive data (RD)
	b1	⌋A⌋ clear to send (CTS)
	b5	⌋E⌋ request to send (RTS)
	b6	⌋E⌋ data terminal ready (DTR)
	b7	⌋E⌋ transmit data (TD)
★ E7E1	(PRB2):	Datenregister Port B
	b0-b7	⌋A⌋ Ausgang Druckerdaten (parallel)

- ★ E7E2 (CRA2): Kontrollregister Port A
 CA1 »E« request to send (RTS)
- ★ E7E3 (CRB2): Kontrollregister Port B
 CB1 »E« acknowledge (ACK)
 CB2 »A« strobe (STR)