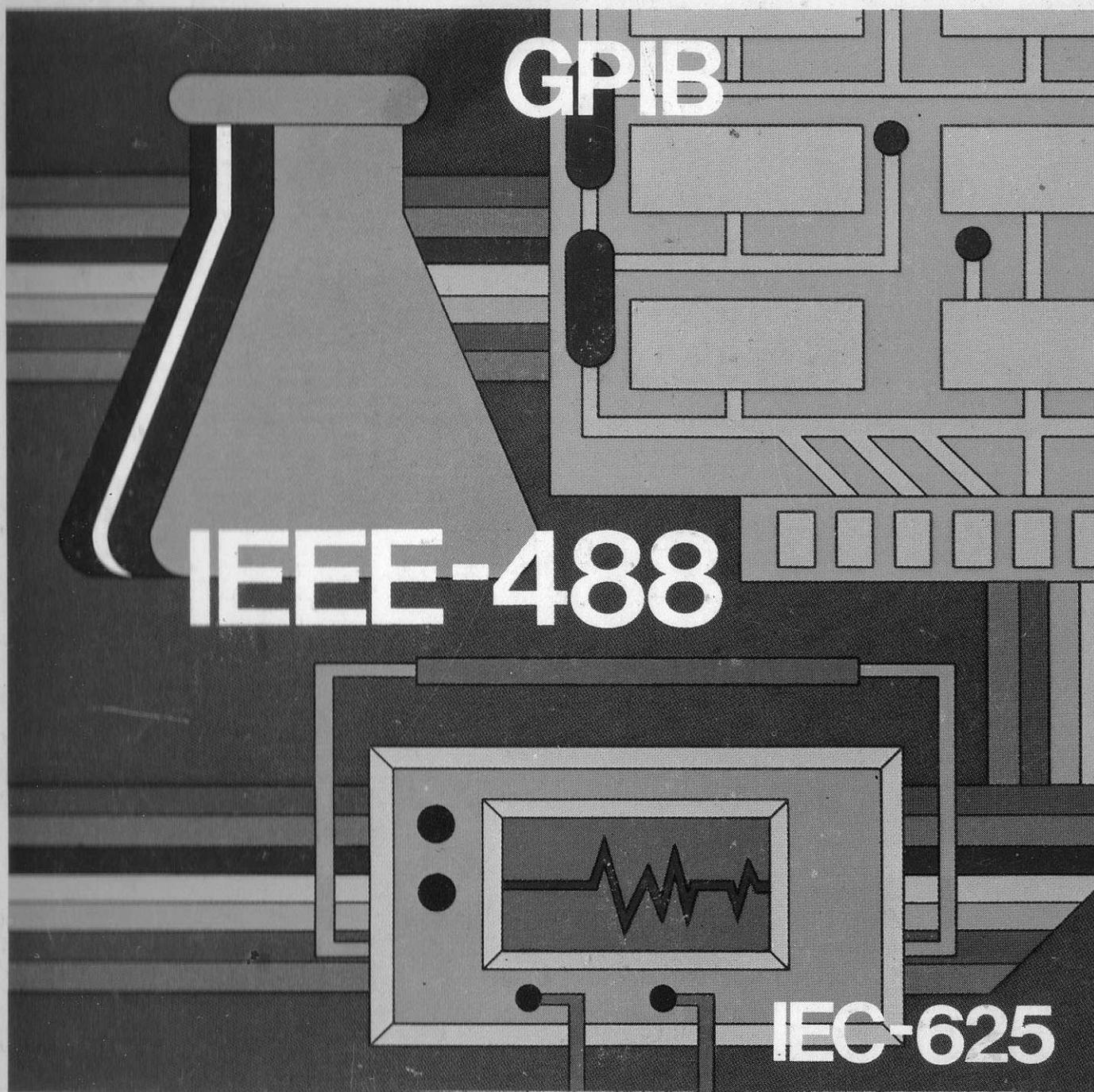


Apple II



IEEE-488 Interface

Bedienungsanleitung



Zur Beachtung

Apple Computer Inc. behält sich das Recht von Verbesserungen des in diesem Handbuch beschriebenen Produktes zu jeder Zeit und ohne Ankündigung vor.

AUSSCHLUß JEDER GARANTIE UND HAFTUNG

APPLE COMPUTER INC. GIBT KEINE GARANTIE, WEDER AUSDRÜCKLICH ODER UNAUSGESPROCHEN, IN BEZUG AUF DIESES HANDBUCH NOCH IN BEZUG AUF DIE IN DIESEM BUCH BESCHRIEBENE SOFTWARE, IHRE QUALITÄT, DURCHFÜHRBARKEIT, VERKÄUFLICHKEIT ODER VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. APPLE COMPUTER INC. SOFTWARE IST GEKAUFT ODER LIZENSIERT „WIE BESEHEN“. DAS RISIKO FÜR DIE QUALITÄT UND DURCHFÜHRBARKEIT LIEGEN GANZ BEIM KÄUFER. SOLLTEN SICH DIE PROGRAMME NACH IHREM KAUF ALS FEHLERHAFT HERAUSSTELLEN, TRÄGT DER KÄUFER (UND NICHT APPLE COMPUTER INC., DEREN DISTRIBUTOR ODER EINZELHÄNDLER) DIE GESAMTEN KOSTEN FÜR ALLE NOTWENDIGEN ARBEITEN, REPARATUREN ODER KORREKTUREN UND JEDWEDE DAVON GEFOLGTEN ODER DARAUS RESULTIERENDEN SCHÄDEN. IN KEINEM FALL IST APPLE COMPUTER INC. HAFTBAR FÜR DIREKTE, INDIREKTE, VERURSACHTE ODER GEFOLGTE SCHÄDEN DIE AUS IRGENDWELCHEN SCHÄDEN AUS DER SOFTWARE RESULTIEREN, SELBST WENN APPLE COMPUTER INC. AUF DIE MÖGLICHKEIT SOLCHER SCHÄDEN HINGEWIESEN WURDE.

Dieses Handbuch ist urheberrechtlich geschützt. Alle Rechte sind vorbehalten. Dieses Dokument darf nicht in Teilen oder im Ganzen kopiert, fotokopiert, reproduziert, übersetzt oder verkleinert werden auf irgendein elektronisches Medium oder in maschinenlesbare Form ohne vorherige schriftliche Zustimmung durch Apple Computer Inc.

© 1980 durch **Apple Computer Inc.**

Apple Computer International
7, rue de Chartres
92200 Neuilly-sur-Seine
FRANCE

Das Wort APPLE und das Apple-Zeichen sind registrierte Warenzeichen der Apple Computer Inc.

Apple Produkt Nr. A2LD037 (030-D197-A)

Apple II

IEEE-488 Interface

Bedienungsanleitung

Lesen Sie dieses Handbuch genau durch, bevor Sie mit dem Einbau der IEEE-488 Interface-Karte beginnen. Eine fehlerhafte Installation der Karte kann Interface-Karte und Apple Computer ernsthaft beschädigen.

INHALTSVERZEICHNIS

ABSCHNITT 1

Einführung

Was ist der Apple II GPIB	2	Was Sie benötigen (Hard- und Software)	4
Einige Hinweise	2	Einbau der Interface-Karte	4
Spezielle Symbole	3		

ABSCHNITT 2

Die IEEE-488-Interface-Karte

Controller, Talker und Listener	10	Datenleitungen	14
Physikalische Anforderungen	11	Management-Steuerleitungen	14
Datenraten	13	Transfer-Steuerleitungen	15
Elektrische Anforderungen	13	GPIB-Funktionen	17
GPIB-Anschlüsse	14		

ABSCHNITT 3

Programmierung -es GPIB

Einführung	20	WRITECNT: WC	29
Notation der Kommandos	21	READ: RD	30
Adressen	24	READCNT: RC	30
Empfängeradressen	24	XFER: XF	31
Senderadressen	24	TRIGR: TG	31
Sekundäradressen	24	Clear-Kommando: CA	31
Zur Benutzung von Adressen	25	Selektives Rücksetzen: CL	32
Beschreibung		Betriebsbereitschaft: RA	32
der GPIB-Kommandos	25	Gezieltes Setzen	
Senden eines Kommandos	25	in den Betriebszustand: RM	32
Kommando-Protokoll	26	Lokalsperrung: LL	33
Übertragung auf dem Bus	27	Lokale Betriebsart: LA	33
GPIB-Kommandos	27	Gezieltes Einschalten der lokalen	
Zeilenvorschub: LF	27	Betriebsart: LO	33
Stringende: ES	28	Bedienungsanforderung: SR	33
Bildschirm-Kommando: SC	28	Seriell Polling: SP	34
Geräte-Kommando: DV	28	Paralleles Polling: PP	34
WRITE: WT	29	Pollingbereitschaft: PE	35

Abschaltung des Polling: PD	35	Sender abschalten: UT	36
PU	36	GPIB-Fehlermeldungen	36
AB	36	Die Anweisung ONERR ... GOTO	37

ANHANG A:

ASCII-Umwandlungs-Tabelle

Tabelle A-1 – GPIB-Empfängeradressen	40	Tabelle A-3 – GPIB-Sekundär - A	42
Tabelle A-2 – GPIB-Senderadressen	41	Tabelle A-4 – ASCII-Steuer - Codes	43

ANHANG B

Bus-Ein - /Ausgabefolgen

ANHANG C

9914-Register

ANHANG D

Programmierhilfen

Programm 1:		Programm 3:	
CHR\$-Funktion für Integer-Basic	54	Funktion zum Verhindern	
Programm 2:		der Reinitialisierung	56
Fehlerbehandlungsroutine	55	Programm 4:	
		Eingabe mit Kommas	57

ANHANG E

Routinen für beschleunigte Übertragung

GPIB-Subroutine DEMO	60	Warteroutine (WAIT)	66
Subroutinen	61	Datenpuffer	66
Allgemeine Steuer-Werte	61	Unterprogramm TEST	67
Initialisierung des 9914	62	Symbol-Tabelle	68
Senderroutine	63	Datendurchsatz	68
Empfangsroutine	64	Ausführung und Test	69

ANHANG F

Programmieren in Pascal

ANHANG G

Programmbeispiele

Leseroutine für den HP-3438A	74
Allgemeine HP 3586C Routine	75

ANHANG H

Schematisches Diagramm

Glossar

Stichwortverzeichnis

ABSCHNITT 1

Einführung

- 2 Was ist der Apple II GPIB
- 2 Einige Hinweise
- 3 Spezielle Symbole
- 4 Was Sie benötigen (Hard- und Software)
- 4 Einbau der Interfacekarte

Mit der IEEE-488-Interface-Karte für den Apple II steht ein steckkompatibler, standardmäßiger Ein- / Ausgabe-Bus für Mikrocomputer zur Verfügung. Er ermöglicht es Ihrem Apple II, mit bis zu 14 externen Geräten Daten auszutauschen oder diese Geräte zu steuern. Bei den externen System-Komponenten kann es sich beispielsweise um Meßgeräte, Daten-Logger oder ein bzw. mehrere Peripheriegeräte aus dem schon jetzt umfangreichen Angebot verschiedener Hersteller handeln. Voraussetzung dafür ist, daß alle Geräte über Bus-Interfaces verfügen, die mit den technischen, elektrischen und mechanischen Vorschriften des IEEE-Standards 488 übereinstimmen. Zur Zeit gibt es ca. 600 verschiedene Geräte, die in irgendeiner Form diesen als Standard vorgeschlagenen Ein - / Ausgabe-Bus verwenden.

Was ist der Apple-II-GPIB?

Die offizielle Bezeichnung für diesen Bus lautet: 'IEEE Standard Digital Interface for Programmable Instrumentation 1978' (Digitales IEEE-Standard-Interface für programmierbare Instrumente, 1978). Der Instrumenten-Bus ist auch unter den Bezeichnungen:

- IEEE-488-Interface-Bus
- General Purpose Interface Bus (GPIB)
- Hewlett-Packard-Interface Bus (HPIB)

bekannt. Die Original-Dokumentation des IEEE Standards 488 müssen Sie nur dann kennen, wenn Sie ausführliche Unterlagen für die Entwicklung einer Interface-Schaltung benötigen.

In diesem Handbuch wird das Apple-II-IEEE-488-Interface künftig nur noch als 'Apple-II-GPIB-Interface', oder kurz lediglich als 'GPIB-Interface' bezeichnet. Das Apple-II-GPIB-Interface besteht aus der Interface-Karte und dem Interface-Kabel. Bitte denken Sie bei der Erklärung des GPIB daran, daß die Ausdrücke 'Apple-II-IEEE-488-Interface-Karte', 'IEEE-488-Bus' und 'HP-IB' sich jeweils auf das gleiche universelle Ein - / Ausgabe-Bus-Interface und den gleichen Bus-Betrieb beziehen. Das Board mit der eigentlichen Interface-Schaltung wird als 'Apple-II-Interface-Karte' oder kurz als 'IEEE-488-Karte' bezeichnet.

Die Apple-II-GPIB-Interface-Karte ist das Herz des Ein - / Ausgabe-Betriebs. Sie kann in eine Apple-II-Karten-Steckfassung eingesetzt werden und steuert alle Übertragungen über den Bus. Die Interface-Karte arbeitet positionsunabhängig, d.h. sie kann in jeden beliebigen freien Slot (Steckfassung) im Apple (außer Platz 0) eingesetzt werden. Die Interface-Karte ermöglicht es, mit einfachen BASIC-Befehlen in Ihrem Programm externe Geräte fernzusteuern.

Die Interface-Karte übernimmt die Übersetzung der in Integer BASIC oder Applesoft BASIC geschriebenen Instruktionen in Kommandos, die von den angeschlossenen Geräten verstanden werden. Die Interface-Schaltung enthält dazu u.a. einen Controller, der das gesamte Timing, die Steuerung und die Daten-Formatierung für den Instrumenten-Bus ausführt. Auf der Interface-Karte ist auch ein Nur-Lese-Speicher untergebracht (ROM), der die vom Interface-Controller benötigten Informationen zur Interpretation der BASIC-Kommandos liefert.

Einige Hinweise

Obwohl die elektrischen Bus-Eigenschaften eindeutig durch den IEEE-Standard 488 spezifiziert sind, können trotzdem die Adressierungs-Konventionen, das Daten-Eingabe-Format und das Format der Ausgangsdaten von einem Gerät zum anderen variieren. Es ist daher notwendig, daß Sie zunächst das Betriebs-Handbuch der Hersteller der einzelnen Geräte sorgfältig durchlesen, bevor Sie darangehen, ein Gerät am Bus in Betrieb zu nehmen. Beispielsweise können die Daten im ASCII-Code oder in binärem Format ausgegeben werden. Bei binärem Format wiederum kann beispielsweise das niederwertigste Bit eines jeden Bytes zuerst oder zuletzt übertragen werden usw.

Bis jetzt gibt es leider keine guten und leicht verständlichen Handbücher mit Anleitungen für den Aufbau und Betrieb eines System gemäß dem IEEE-Standard 488. Sie müssen jedoch bereits über gründliche Kenntnisse der Programmierung in BASIC oder APPLESOFT verfügen sowie praktische Erfahrungen mit dem Disketten-Betriebssystem (DOS) besitzen. Sollte das noch nicht der Fall sein, nehmen Sie bitte zunächst die BASIC- und DOS-Manuals zur Hand und üben Sie die darin angegebenen Beispiele. Darüber hinaus müssen Sie auch den Zusammenhang zwischen der Daten-Darstellung im ASCII-Code und in hexadezimaler, oktaler und binärer Form kennen. Umwandlungs-Tabellen enthält der Anhang.

Sehr vorteilhaft wäre es, wenn Sie über ein Prüfgerät verfügen, das am Bus angeschlossen werden kann und mit dem sich die Kommandos und Daten überwachen lassen, die von Ihrem Programm gesendet werden. Ein Gerät, mit dem das Bus-Programm schrittweise überprüft werden kann, ist für ein komplexes System praktisch unumgänglich nötig. Ein derartiges Test-Gerät ist beispielsweise der GPIB Analyser Modell ZT488 von Ziotech Corporation. Die Bus-Arbeitsweise kann auch mit einem Logik Analysator von Tektronix oder Hewlett-Packard überwacht werden. Ein weiteres, sehr nützliches Gerät ist auch der GPIB Analyser Modell K100D/408 von Gould.

Spezielle Symbole

In diesem Manual werden einige spezielle Symbole verwendet, um besonders wichtige Passagen des Textes zu kennzeichnen. Wenn Sie zum Beispiel das Symbol



sehen, soll es darauf hinweisen, daß der folgende Abschnitt ein möglicherweise unerwartetes Verhalten des Apple-Computers erläutert. Das Symbol



bedeutet, daß der folgende Abschnitt spezielle Informationen enthält, die für Sie sehr nützlich sein können. Lesen Sie die folgenden Abschnitte daher besonders sorgfältig.

Was Sie benötigen (Hard- und Software)

Die IEEE-488-Interface-Karte für den Apple II wurde für erfahrene Anwender entwickelt, die mit APPLE-SOFT oder Integer BASIC sowie den Betriebs-Systemen DOS 3.2 oder DOS 3.3 vertraut sind. Wer die Karte in Zusammenhang mit dem Apple-PASCAL-Betriebs-System einsetzen möchte, sollte über ausreichende Kenntnisse und Erfahrungen mit APPLE-PASCAL, seinem Betriebs-System und der 6502-Assembler-Sprache verfügen.

Die in diesem Manual beschriebenen Programme sowie die von der Karte verwendete Firmware sind auf einem Apple II oder einem Apple II Plus mit mindestens 32 K Speicher und wenigstens einem

Disk-II-Laufwerk mit Controller lauffähig.

Das IEEE-488-Interface-Card-Package für den Apple II besteht aus folgenden Teilen:

1. Apple-II IEEE-488-Interface-Karte
Part-No. 030-0670-0016.
2. Verbindungskabel mit darauf befindlicher Metall-Anschluß-Halterung
Part-No. 030-0590-0030
3. zwei IEEE-Standard-Schrauben
4. zwei nicht dem Standard entsprechenden Schrauben
5. Garantiekarte
6. diesem Manual

Packen Sie alles sorgfältig aus. Füllen Sie nach Überprüfung die Garantiekarte aus und schicken Sie sie an Ihren Apple-Händler.

Einbau der Interface-Karte

Lesen Sie diesen Abschnitt sorgfältig durch, auch wenn Sie bereits andere Apple-Peripherie-Interface-Karten installiert haben. Das Einsetzen der Apple-II-GPIB-Interface-Karte ist recht einfach, aber es ist wichtig, daß es in der vorgeschriebenen Weise erfolgt:



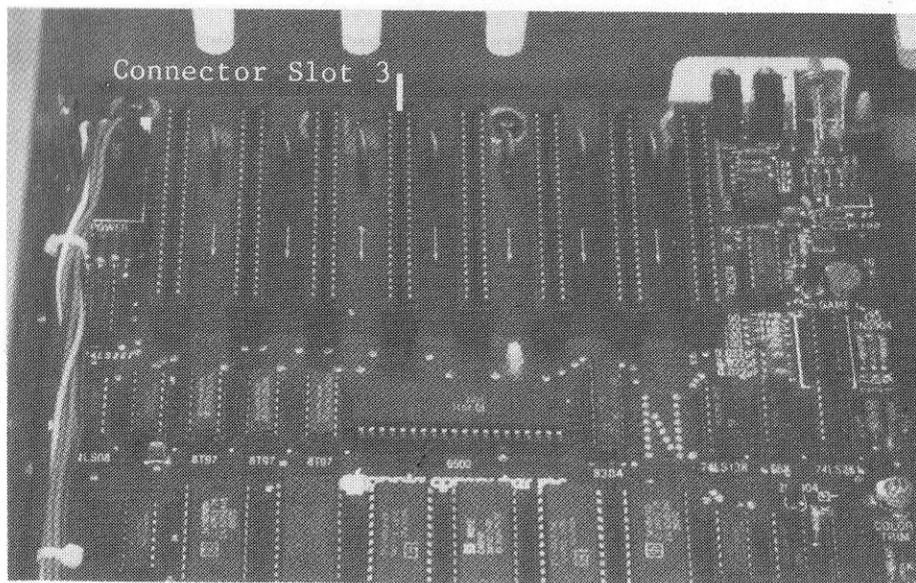
Bevor Sie irgend etwas am Apple anschließen oder eine Verbindung lösen, schalten Sie unbedingt mit dem links an der Rückseite befindlichen Schalter die Netzspannung ab. Das ist notwendig! Andernfalls besteht die Gefahr, daß die Schaltung beschädigt wird und außerdemdermomentane Inhalt des Schreib - /Lesespeichers (RAM) verlorengeht, wenn derartige Arbeiten bei einge schaltetem Gerät erfolgen.

Ziehen Sie jedoch nicht den Netzstecker des Apple II heraus, sondern schalten Sie das Gerät lediglich aus. Sobald Sie nämlich den Netzstecker des Apple herausziehen, wird er von der Erdung abgetrennt und verfügt dann über keinerlei Schutz mehr gegen statische Entladungen.

Entfernen Sie zunächst den Deckel des Apple-Gehäuses, indem Sie an den beiden hinteren Ecken des Oberteils ziehen, bis die dort befindlichen Verschlüsse aufspringen. Schieben Sie dann den Deckel zurück, bis er frei vom Gehäuse ist, und heben Sie ihn ab.

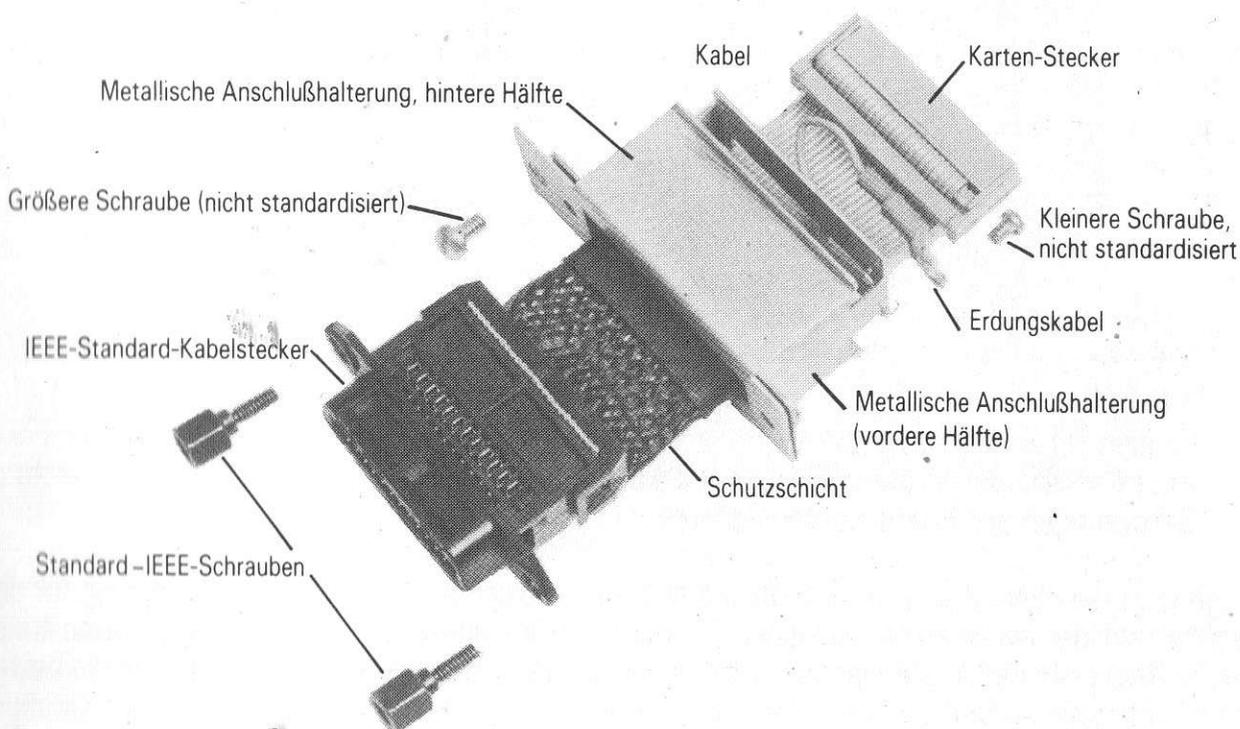
Lokalisieren Sie jetzt im Apple das Gehäuse des Netzteils. Es handelt sich dabei um das rechteckige Metall-Kästchen in der linken hinteren Ecke. Berühren Sie das Metallgehäuse mit einer Hand, um alle statischen Ladungen abzuleiten, die sich auf Ihrer Bekleidung oder Ihrem Körper befinden könnten.

Längs der hinteren Kante des Apple-Computer-Boards sehen Sie acht lange, schmale Karten-Steckfassungen, genannt 'Connector Slots'. Sie sind von 0 in der linken Ecke bis 7 in der rechten Ecke durchnummeriert. Die jeweilige Positions-Nummer ist hinter den einzelnen Steck-Fassungen aufgedruckt. Die Lage der Slots zeigt das folgende Bild.



Die Karte Ihres Disk-Controllers steckt in Slot 6. Falls Sie mehr als zwei Laufwerke angeschlossen haben, befindet sich der zweite Controller in Slot 4 oder 5. Sofern Sie auch noch über eine Apple-Language-Card verfügen, dann ist sie in Slot 0 eingesetzt. Haben Sie einen Drucker, so steckt die Printer-Interface-Karte vermutlich in Slot 1. Bei den folgenden Erläuterungen in diesem Manual wird angenommen, daß Sie die Apple-II-GPIB-Interface-Karte in Slot 3 stecken. Wenn Sie einen anderen Steckplatz verwenden wollen, so setzen Sie einfach jedesmal die Nummer dieses Slots ein, wenn im Manual Slot 3 angesprochen wird.

Das IEEE-Verbindungskabel des Apple zeigt das folgende Bild.

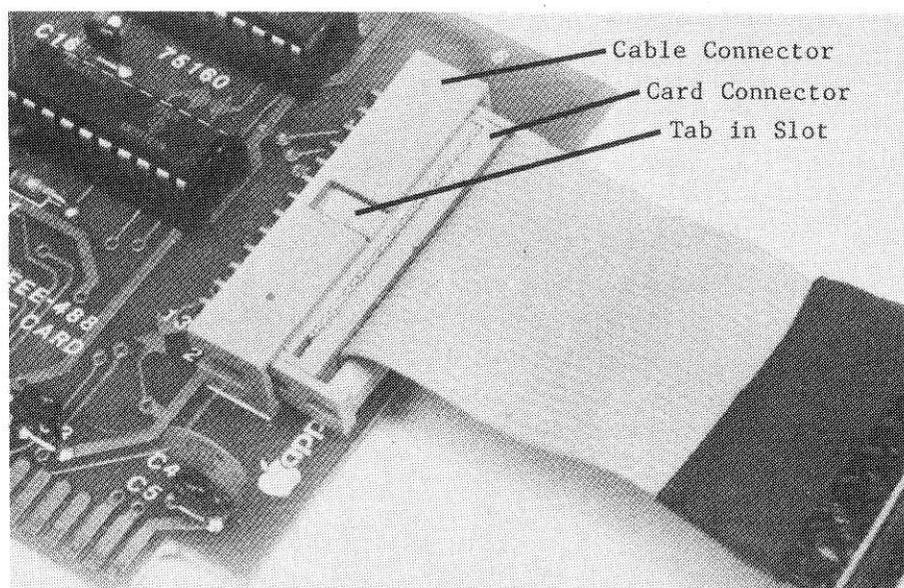


Der IEEE-Standard-Kabelanschluß befindet sich links. An ihm werden die anderen Geräte angeschlossen. Rechts davon sehen Sie eine zweiteilige Metall-Halterung. Der dem IEEE-Anschluß am nächsten liegende Teil der Halterung soll künftig als die hintere Hälfte der Halterung bezeichnet werden. Die andere Hälfte der Anschluß-Halterung, also die vordere, hat einen schmalen Kabel-Durchlaß, durch den das Kabel hindurchführt. Am rechten Kabelende befindet sich der Anschluß für die Interface-Karte.

Wenn die beiden Hälften der Anschluß-Halterung fest zusammengedrückt werden, so wie es das obige Foto zeigt, bilden sie ein kleines, rechteckiges Metall-Gehäuse mit einer Öffnung für den IEEE-Anschluß an der linken Seite, einem Durchlaß für das Kabel an der rechten Seite sowie zwei anderen großen Öffnungen. Diese beiden Öffnungen ermöglichen es, die Halterung in einen der Schlitze an der Rückseite des Gehäuses einzuschieben. Vergewissern Sie sich, daß die beiden Hälften zusammengedrückt sind, aber nicht zu sehr, da die Gehäusewand in die beiden Öffnungen passen muß.

Das linke Ende des Flachkabels, also am IEEE-Anschluß, ist mit einer Schutzschicht bedeckt, unter der einer der Drähte, der für die Erdung der Abschirmung vorgesehen ist, abgetrennt ist. Ein neuer Draht, der am IEEE-Steckerende des abgetrennten Drahtes angebracht ist, wird aus der Abschirmung herausgeführt. Vergewissern Sie sich, daß dieser Draht parallel zum IEEE-Kabel und durch den Kabeldurchlaß in der vorderen Hälfte der Anschluß-Halterung verläuft. Dieser Draht wird mit der Gehäuse-Erde verbunden.

Der Karten-Stecker am Bus-Kabel paßt in ein entsprechendes Gegenstück auf der Interface-Karte. Halten Sie zunächst das Bus-Kabel so, daß die Nase auf dem Karten-Stecker des Kabels dem Schlitz in der Steckfassung auf der Karte gegenüberliegt. Drücken Sie dann den Kabelstecker in die Fassung, bis er sicher festsetzt, so wie es die folgende Abbildung zeigt.

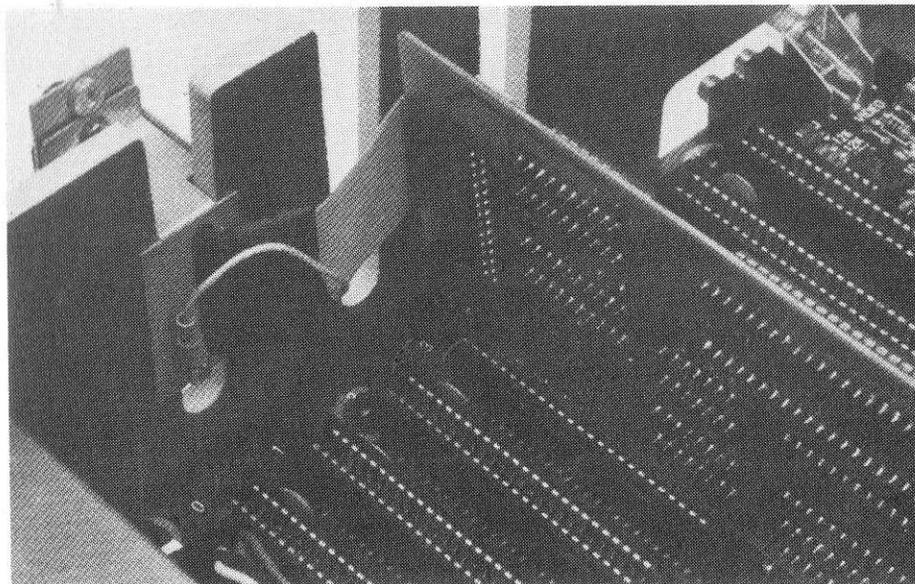


Fassen Sie die Interface-Karte bitte so wie eine teure Schallplatte an, also nur an den Ecken oder Kanten, und berühren Sie weder die Bauteile noch deren Anschlüsse, vor allem nicht die Steckungen des Board-Kantensteckers.

Erfassen Sie jetzt die obere Kante der Interface-Karte mit der rechten und die Metall-Halterung für den IEEE-Anschluß mit der linken Hand. Schieben Sie dann die Metall-Anschluß-Halterung in einen freien Schlitz an der Rückseite des Apple-Gehäuses; ergeben sich dabei Schwierigkeiten, ziehen Sie die beiden Hälften der Halterung wieder ein wenig weiter auseinander, wodurch die Öffnung für das Gehäuse größer größer wird. Führen Sie gleichzeitig die Karte mit der rechten Hand in Richtung zum Karten-Slot Nr. 3.

Setzen Sie die vergoldeten Steckzungen des Kantensteckers der Interface-Karte in Slot 3 ein, und zwar zuerst die hintere Kante. Bitte drücken Sie anschließend auch die vordere Kante der Karte nach unten, bis sie fest sitzt, so wie es das folgende Bild zeigt.

Befestigen Sie jetzt den Erdungsdraht für die Abschirmung des Bus-Kabels, indem Sie ihn unter der kleineren der beiden nicht standardmäßigen Schrauben in dem im unteren Teil der vorderen Hälfte der Anschluß-Halterung befindlichen Gewindeloch festklemmen.



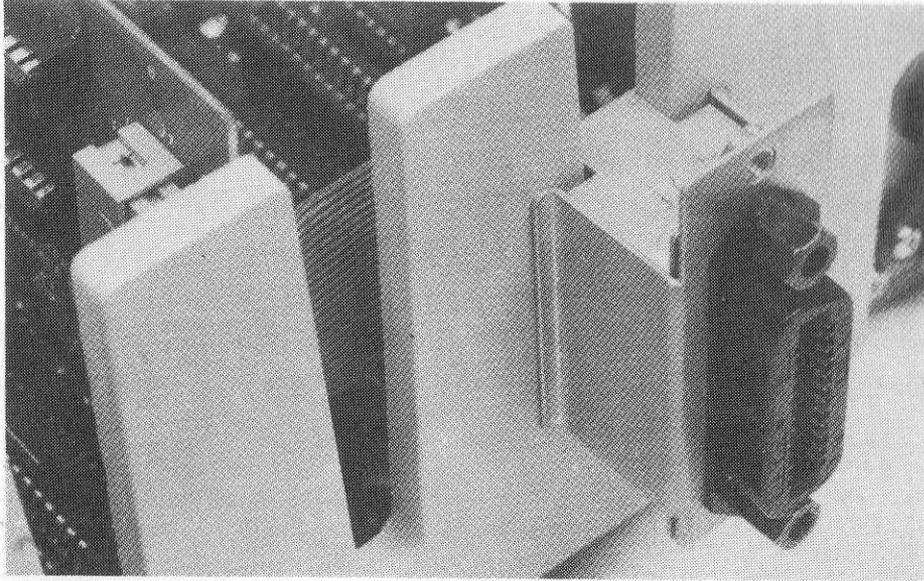
Dadurch verbinden Sie den Erdungsdraht der Bus-Abschirmung mit der Gehäuseerde des Apple II. Der Draht sollte so angeschlossen werden, wie es das obige Bild zeigt.



Sofern Ihr Apple-II-Gehäuse allerdings innen nicht metallisiert ist, hat Ihr Erdungsdraht keine Masseverbindung!

Für den letzten Schritt des Installationsvorgangs drehen Sie den Apple herum, so daß Ihnen die Rückseite des Gehäuses zugewandt ist. Vergewissern Sie sich, daß die metallische Anschluß-Halterung sicher im Gehäuseschlitz fest sitzt und so weit wie möglich nach unten gedrückt ist. Nehmen Sie dann eine der beiden IEEE-Standard-Schrauben, stecken Sie sie durch eins der beiden Löcher im IEEE-Anschluß und schrauben Sie sie im entsprechenden Gewindeloch in der Anschluß-Halterung fest. Wiederholen Sie diesen Vorgang mit der anderen IEEE-Schraube und dem zweiten Loch im IEEE-Anschluß.

Sie haben damit gleichzeitig zwei Dinge erreicht: Einerseits wurde die Anschluß-Halterung fest mit der Gehäuse-Rückseite verbunden, so daß sich der Anschluß nicht mehr im Schlitz bewegen kann und eine gute Verbindung zwischen Halterung und Gehäuse-Erde hergestellt ist. Darüber hinaus wurde der IEEE-Anschluß so befestigt, daß keine Gefahr besteht, daß auf das Kabel eine unzulässige Zugbelastung ausgeübt wird. Wie der Anschluß an der Rückseite des Apple endgültig aussieht, zeigt das folgende Bild.



Die noch übriggebliebene Schraube kann zur Unterdrückung hochfrequenter Störsignale verwendet werden, die das Bus-Kabel ausstrahlt. Dazu muß die Abschirmung des Kabels, die am IEEE-Anschluß an der Rückseite des Apple angebracht ist, mit der Schraube am oberen Gewindeloch in der vorderen Hälfte der Anschluß-Halterung befestigt werden.

Bringen Sie nun den Gehäuse-Deckel des Apple wieder in seine vorgesehene Lage und drücken Sie ihn an den hinteren Ecken herunter, bis die Halterungen einrasten. Das Apple-II-IEEE-488-Bus-Interface ist jetzt installiert und bereit für den Einsatz.

ABSCHNITT 2

Die IEEE-488-Interface-Karte

- 10 Controller, Talker und Listener
- 11 Physikalische Anforderungen
- 13 Datenraten
- 13 Elektrische Anforderungen
- 14 GPIB-Anschlüsse
- 14 Datenleitungen
- 14 Management-Steuerleitungen
- 15 Transfer-Steuerleitungen
- 17 GPIB-Funktionen

Die Apple-II-IEEE-488-Interface-Karte ist eins von mehreren 'standardisierten' Ein-/Ausgabe-Interfaces, die seit 1972 mit der Absicht entwickelt wurden, einen universellen I/O-Bus zur Verfügung zu haben.

Der Vorteil des Apple-II-GPIB besteht darin, daß er mit dem IEEE-Standard 488 übereinstimmt und eine einzige Interface-Karte Ihrem Apple II ermöglicht, bis zu 14 andere Datenerfassungs- oder Verarbeitungseinheiten zu steuern, bzw. mit ihnen in Verbindung zu treten. Das Interface benutzt ein kommerziell erhältliches Interface-Kabel mit 'Huckepack-Steckverbindern', das sowohl eine ketten- (Daisy-Chain) als auch sternförmige Verbindung der Geräte am Bus ermöglicht, wie es die Bilder 2-2 und 2-3 zeigen. Das Standard-Kabel läßt sich an alle Geräte anschließen, die für den Betrieb gemäß dem vorgeschlagenen IEEE-488-Standard entwickelt wurden.

Controller, Talker und Listener

Der Apple-II-GPIB verwendet acht Daten- und ebenso viele Steuerleitungen zur Übertragung von Befehlen sowie Daten zwischen den einzelnen Ein-/Ausgabegeräten, die am Bus angeschlossen sind. Bild 2-1 zeigt ein Blockdiagramm der Bus-Struktur. Die Geräte am Bus können, entsprechend ihrer jeweiligen Grund-Funktion, in Controller, Sender (Talker) und Empfänger (Listener) eingeteilt werden.

Ein am Bus angeschlossenes Gerät wird als 'Sprecher' bezeichnet, wenn seine primäre Funktion in der Übertragung von Daten besteht. Ein Beispiel für einen Sprecher ist ein Digital-Voltmeter. Wenn ein Sender Daten kontinuierlich über den Bus überträgt, können die anderen Geräte den Bus nicht benutzen. Daher wird die Sender-Funktion vom Controller, wenn erforderlich, eingeschaltet.

Jedes Gerät am Bus, dessen primäre Funktion in der Ausführung von Befehlen besteht, wird als Empfänger (Listener) bezeichnet. Ein Drucker beispielsweise ist ein derartiges Gerät. Würde ein Empfänger sämtliche Kommandos und Daten aufnehmen, die über den Bus übertragen werden, wäre ein einziges Chaos die Folge. Daher muß auch die Listener-Funktion vom Controller aktiviert werden. Einige Empfänger verfügen zusätzlich über eine Talker-Funktion.

Der 'Controller' ist das Gerät, welches alle Daten-Transfers auf dem Bus regelt. Er kann einen Sender für die Ausgabe von Daten einschalten, ebenso aber auch einen Empfänger für den Empfang von Daten aktivieren.



Bei der im Apple vorliegenden Bus-Implementation ist nur ein Controller erlaubt. Dieser Controller muß Ihr Apple sein.

Ein Controller ist in der Lage, sowohl mit Sendern als auch Empfängern von verschiedener Art zusammenzuarbeiten. Die Fähigkeiten des Apple II als Controller werden ausführlich in dem Kapitel über die Programmierung des GPIB beschrieben.

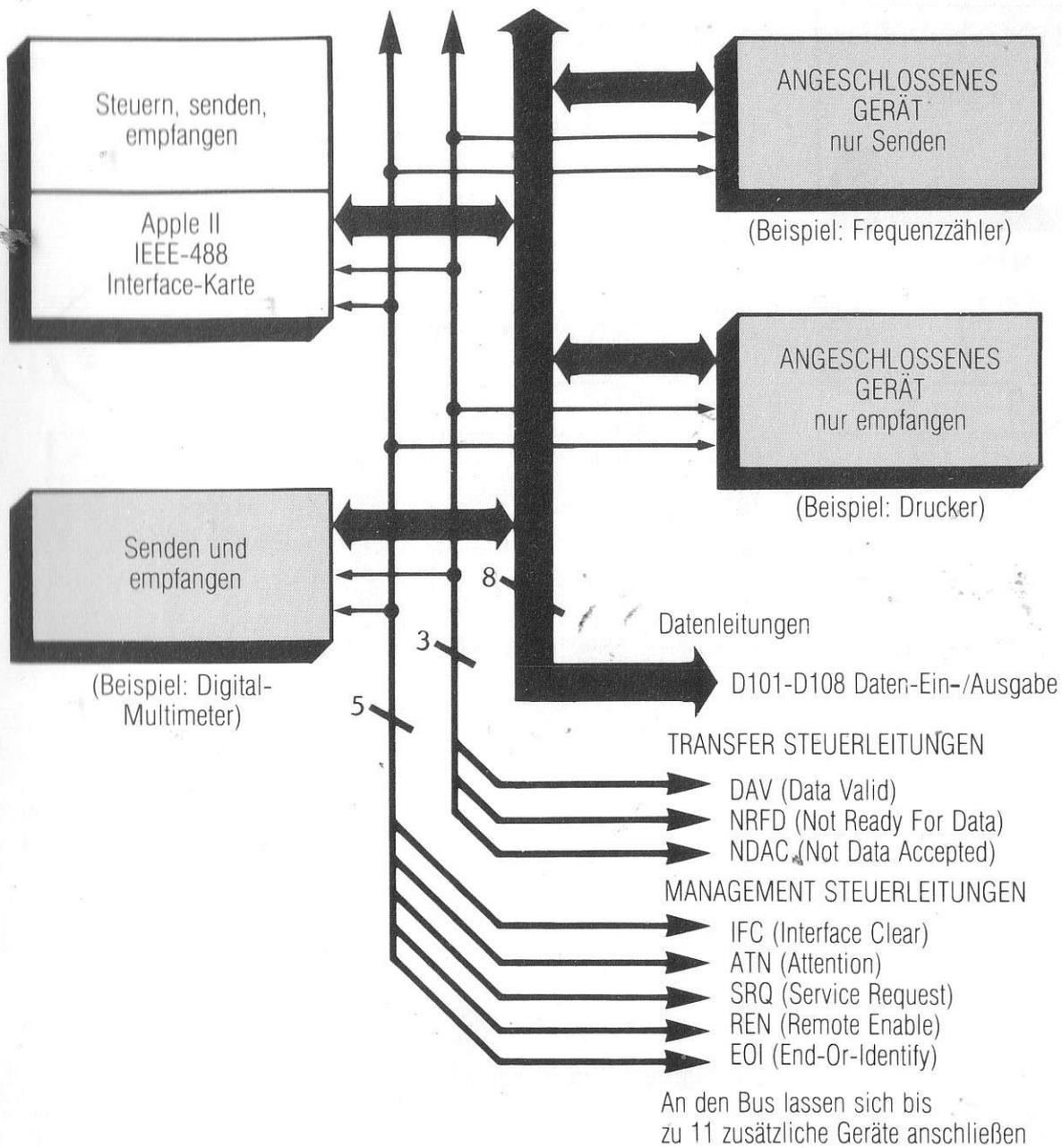


Bild 2-1. Apple II GPIB-Funktions-Blockdiagramm

Physikalische Anforderungen

Es wurde bereits darauf hingewiesen, daß der Apple-II-GPIB bis zu 14 externe Geräte steuern kann. Hinsichtlich der Bus-Gesamtlänge und der Distanz zwischen den einzelnen Geräte besteht eine Reihe von Vorschriften. Der Standard schreibt vor, daß die Bus-Gesamtlänge nicht größer als eine Strecke sein darf, die sich aus der Länge 2m, multipliziert mit der Anzahl der am Bus angeschlossenen Geräte, ergibt. Die maximale Länge des gesamten Bus darf 20 m nicht überschreiten.

Die Geräte am Bus können entweder in Daisy-Chain- oder in Stern-Anordnung miteinander verbunden werden. Dies wird durch die stapelbaren Steckverbinder der erhältlichen Standard-Bus-Kabel ermöglicht. Daisy-Chain-Verbindungen werden von den meisten System-Entwicklern bevorzugt, da der in Stern-Anordnung entstehende Stecker-Stapel bei größeren Systemen sehr unhandlich werden kann. Die Zusammenschaltung Bus-kompatibler Geräte in Stern- und Daisy-Chain-Anordnung zeigen die Abbildungen 2-2 und 2-3.

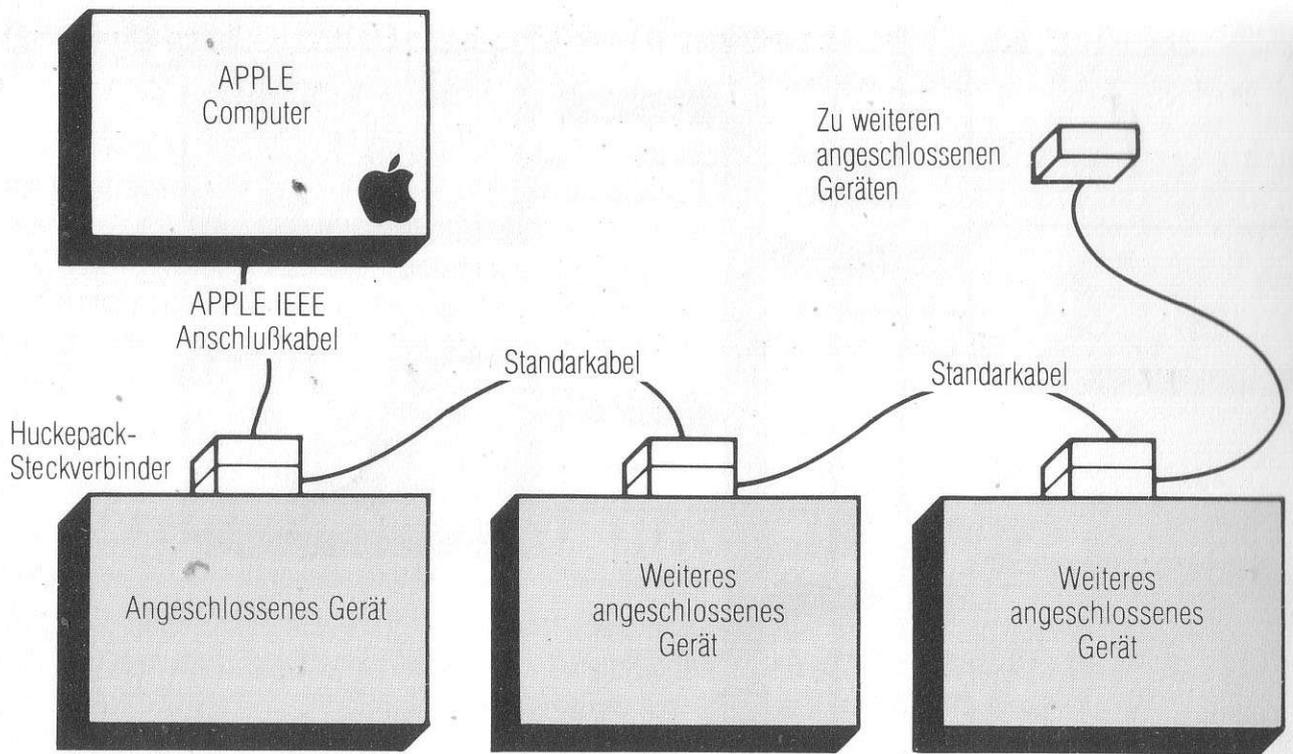


Figure 2-2. Daisy-chain-Anordnung

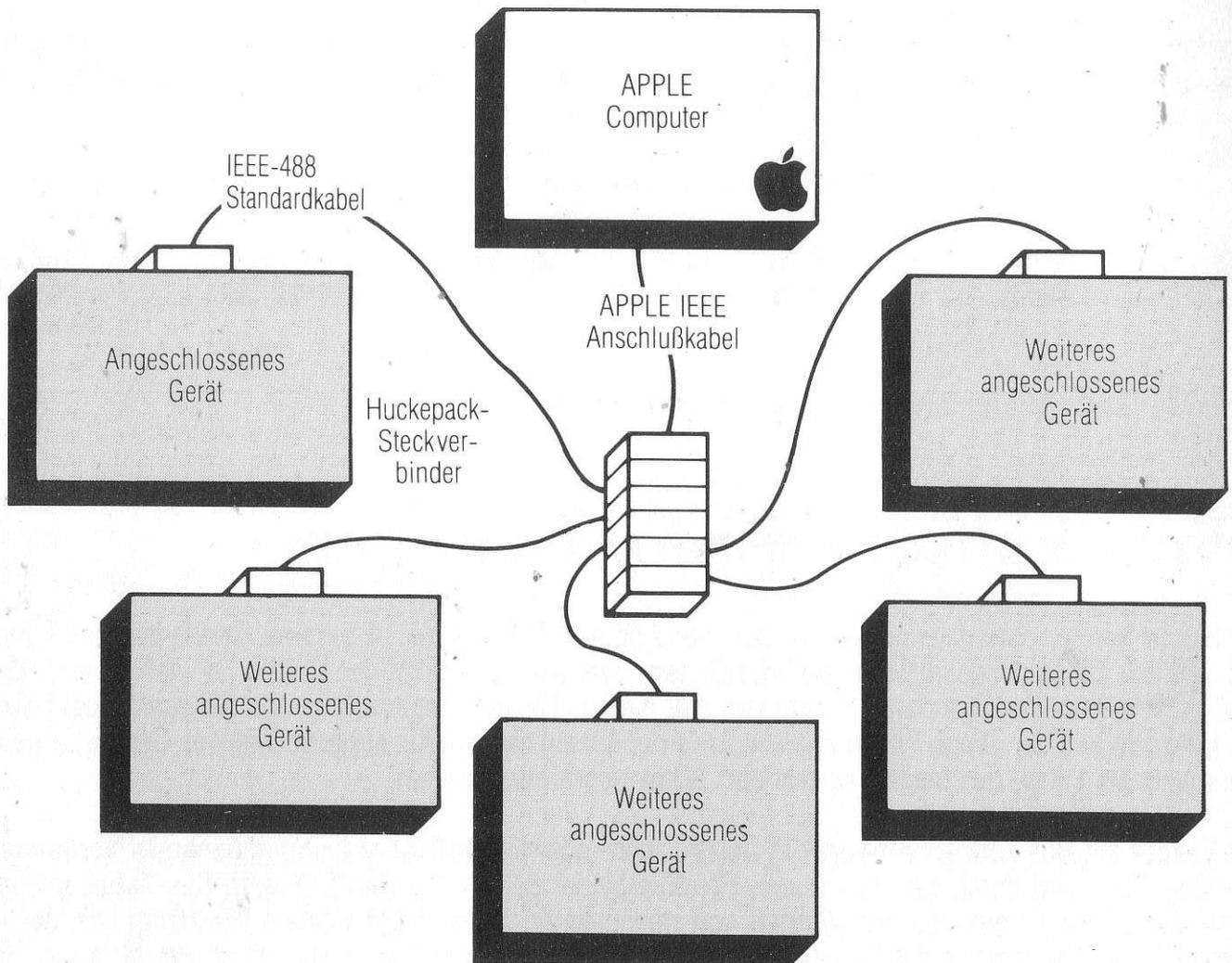
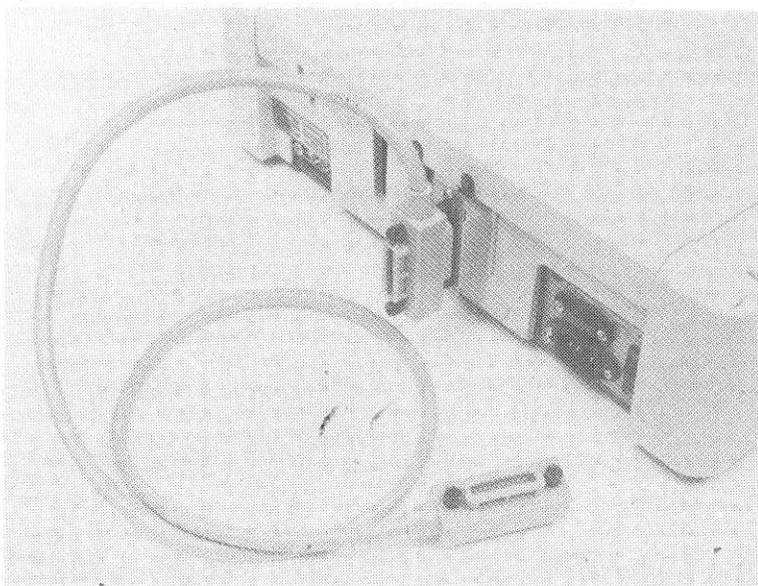


Bild 2-3: Sternförmige Anordnung

Stapelbare oder 'Huckepack'-Kabel sind in verschiedenen Längen von Elektronik-Distributoren und teilweise auch über Apple-Händler zu beziehen.

Ein typisches, handelsübliches Kabel zeigt das folgende Bild.



Datenraten

Beim GPIB handelt es sich um einen parallelen Bus, der jeweils gleichzeitig 8-Bit-Daten-Bytes mit hoher Geschwindigkeit überträgt. Der GPIB kann aber auch Daten- oder Befehlsblöcke Byte für Byte übertragen.

Block-Übertragung ermöglicht einen rascheren Daten-Transfer, mit einem Minimum an zeitraubenden Handshake- und Steuer-Funktionen. Mit anderen Worten: Der Steuerungs-Overhead wird dadurch auf ein Minimum reduziert. Die von einem typischen Gerät ausgehenden Blöcke haben eine Länge von 13 Bytes. Ein Drucker oder ein Disketten-Speicher können eine ganze Zeile oder einen vollständigen Sektor in einem einzigen Block übertragen.

Die theoretisch höchste Daten-Transfer-Geschwindigkeit über den Bus liegt bei 1 MByte/s. Praktisch erreicht werden maximal ungefähr 250 KByte/s, und zwar während eines direkten Speicher-Zugriffs (DMA). Gewöhnlich erfordert ein DMA-Transfer aber soviel Zeit für die Vorbereitungen, daß der Geschwindigkeitsvorteil bei den kurzen Daten-Blöcken, die gewöhnlich während GPIB-Operationen übertragen werden, jedoch wieder verlorenght.

Die typische Datenrate der am Bus angeschlossenen Geräte liegt bei ca. 1 oder 2 KByte/s. Der Betrieb des GPIB läuft am effektivsten ab – wobei gleichzeitig der erforderliche Organisationsaufwand am geringsten wird –, wenn alle Geräte am Bus über gepufferte Eingänge verfügen; das bedeutet, daß sie einen Daten-Block so schnell aufnehmen können, wie ihn der als Controller arbeitende Apple ausgibt.

Elektrische Anforderungen

Idealerweise sollte die Stromversorgung der am Bus angeschlossenen, aber nicht in Betrieb befindlichen Geräte abgeschaltet werden, um Energie zu sparen und die Lebensdauer der Geräte zu erhöhen. Der beste Kompromiß zwischen optimaler Energie-Einsparung und einwandfreiem Bus-Betrieb erfordert aber nur die Abschaltung einiger Geräte. Als Faustregel kann man sich merken, daß der Apple selbst und wenigstens die Hälfte der angeschlossenen Geräte eingeschaltet sein sollten.

GPIB-Anschlüsse

Der Interface-Bus besteht aus 8 Daten- und 8 Steuerleitungen. Die Organisation dieser Leitungen im Standard-Bus-Kabel zeigt Tabelle 2-1. Die mnemonischen Symbole der einzelnen Anschlüsse werden in den folgenden Abschnitten erklärt.

<u>PIN</u>	<u>SIGNAL</u>
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	EOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	shield (earth ground)
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	logic ground
19	logic ground
20	logic ground
21	logic ground
22	logic ground
23	logic ground
24	logic ground

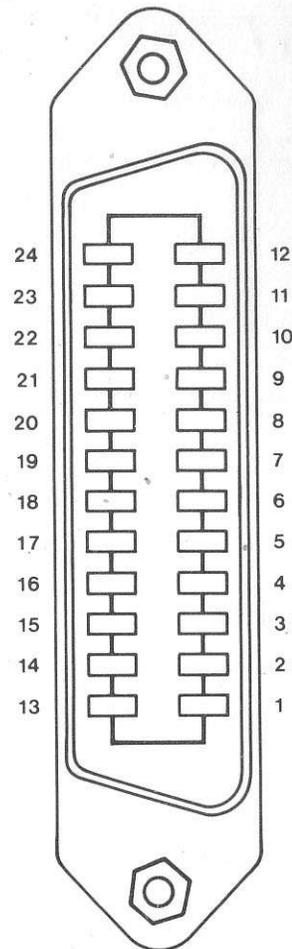


Tabelle 2-1: Bus-Anschluß-Belegung (standard)

Datenleitungen

Im GPIB sind 8 Datenleitungen vorgesehen. Die Datenleitungen werden mit DIO1 bis DIO8 (Data Input/Output 1 bis 8) durchnummeriert und entsprechen den vom Apple verwendeten Datenbits 0 bis 7. Die von den Datenleitungen übertragenen Daten werden durch negative Logik dargestellt (aktiv LOW). In negativer Logik entspricht ein niedriger Pegel auf der Leitung (0 Volt) einer logischen '1' ('wahr'), eine hohe Spannung dagegen (5 Volt) einer logischen '0' ('falsch'). Die Interface-Karte invertiert die Daten, bevor sie sie an den Apple weitergibt bzw. nachdem sie sie vom Apple erhalten hat. Somit werden alle Daten, die der Apple sendet und empfängt, in positiver Logik dargestellt.

Management-Steuerleitungen

Der GPIB verwendet 8 Leitungen, um die Geräte am Bus zu steuern und den Datenfluß auf dem Bus zu lenken. Fünf Steuerleitungen werden für das Management der Daten-Transfers verwendet. Die restlichen drei Steuerleitungen werden für den Handshake-Betrieb oder die Koordination des Transfers einzelner Daten-Bytes benötigt. Eine kurze Beschreibung der fünf Management-Steuerleitungen wird nachfolgend gegeben. Die mnemonische Bezeichnung ergibt sich aus der jeweiligen Bedeutung der Leitung.

Die Signale aller Steuerleitungen sind aktiv LOW; das heißt, daß eine Leitung als aktiv angesehen wird, wenn die Spannung auf dieser Leitung 0 V beträgt.

Bei den meisten Anwendungen muß man nicht unbedingt wissen, welche Steuerleitungen für die einzelnen Befehle verwendet werden. Der Anhang über Bus-Ein- / Ausgabe-Sequenzen gibt Beispiele für das von jedem Befehl verwendete Protokoll.

ATN (ATenNtion) = Achtung

Um anzuzeigen, daß eine Adresse oder ein Steuer-Byte auf den Bus ausgegeben wird, setzt der Controller die ATN-Leitung auf 'wahr', also LOW. Die ATN-Leitung wird auf HIGH gesetzt, um einem Gerät die Ausgabe von Daten auf den Bus zu ermöglichen. Die ATN-Leitung wird auch zusammen mit der EOI-Leitung verwendet, um einen 'Parallel Poll' auszuführen. Sie wird synchron mit den Handshake-Steuerleitungen gesetzt, um eine Verwechslung von Steuer- und Daten-Bytes zu vermeiden.

EOI (End Or Identify) = Ende oder Identifizieren

Das EOI-Signal hat zwei Bedeutungen:

1. Der Controller (Apple) setzt EOI und ATN auf LOW (aktiv), um einen Parallel Poll auszuführen.
2. Ein Sprecher setzt EOI mit dem letzten Daten-Byte auf 'wahr', um das Ende der Übertragung anzuzeigen. Leider verwenden nicht alle verfügbaren Geräte das EOI-Signal, um das Ende einer Übertragung anzuzeigen.

SRQ (Service ReQuest) = Service-Anforderung

Das SRQ-Signal wird von jedem Gerät am Bus auf 'wahr' gesetzt, das ein Eingreifen des Controllers verlangt. Nach Empfang des SRQ-Signals gibt die Controller-Software nacheinander die Adresse jedes einzelnen Gerätes auf den Bus-Datenleitungen aus (Serial Poll). Jedes Gerät gibt daraufhin ein Status-Byte ab. Sobald dasjenige Gerät, das die SRQ-Leitung aktiviert hat, seine Adresse auf dem Bus erkennt, identifiziert es sich selbst, indem es ein Statusbyte ausgibt, bei dem DIO7 wahr ist, und gibt anschließend die SRQ-Leitung frei.

IFC (InterFace Clear) = Interface Löschen

Das IFC-Signal wird nur vom Controller verwendet, um alle Geräte-Interfaces in einen bekannten Anfangs-Zustand zu versetzen. Dieses Signal wird nur durch das Abort-Kommando benutzt.

REN (Remote ENable) = Fernsteuerungs-Freigabe

Das REN-Signal wird nur vom Controller verwendet, um die Fernsteuerung eines Gerätes zu ermöglichen. Das Gerät verbleibt solange in diesem Zustand, wie das REN-Signal wahr ist. Das Gerät antwortet nicht eher auf Daten, die über den Bus geschickt werden, bis das REN-Signal wahr ist.

Transfer-Steuerleitungen

Der Daten-Transfer auf dem Bus wird durch die drei Transfer-Steuerleitungen geregelt. Diese Leitungen arbeiten mit einem Drei-Leitungs-Handshake-System, das von Hewlett-Packard patentiert ist.

Das Handshake-System arbeitet asynchron und ermöglicht den Anschluß von Geräten mit hohen und niedrigen Datenraten am gleichen Bus. Der Drei-Draht-Handshake wird zur Übertragung jedes Informationsbytes verwendet, das über die Datenleitungen übermittelt werden soll. Die drei Transfer-Steuersignale werden nachfolgend beschrieben. Ein Diagramm des zeitlichen Ablaufs des Handshake-Protokolls zeigt Bild 2-4.

NRFD (Not Ready For Data) = Nicht bereit für Daten

Das NRFD-Signal wird durch einen Empfänger auf LOW (wahr) gesetzt, um anzuzeigen, daß er für den Empfang des nächsten Daten- oder Steuerbytes noch nicht bereit ist. Das NRFD-Signal wird vom Controller erst dann als 'falsch' angesehen (d.h. 'bereit für Daten'), bis alle Empfänger am Bus bereit sind.

NDAC (Not Data Accepted) = Daten nicht aufgenommen

Das NDAC-Signal wird durch einen Empfänger auf 'wahr' (LOW) gesetzt, um anzuzeigen, daß er das Daten- oder Steuer-Byte auf dem Daten-Bus noch nicht aufgenommen hat. Das NDAC-Signal am Controller wird nicht eher als 'falsch' angesehen (d.h. 'Daten empfangen'), bis alle Empfänger am Bus die Daten aufgenommen haben.

DAV (Data Valid) = Daten gültig

Das DAV-Signal wird von einem Sender auf 'wahr' gesetzt, um anzuzeigen, daß ein Daten- oder Steuer-Byte auf dem Bus liegt und, allen Empfängern zur Verfügung steht.

Den Ablauf des Drei-Draht-Handshake-Protokolls, das in folgenden Schritten abläuft, zeigt das Diagramm in Bild 2-6:

- 1. Daten nicht mehr gültig:**
Der Talker setzt DAV auf HIGH, sobald er sieht, daß alle Empfänger die Daten empfangen haben (3).
Der Sprecher kann dann die Daten ändern.
- 2. Daten gültig:**
Der Sender sieht, daß alle Empfänger für Daten bereit sind (5). Er legt die Leitung DAV auf LOW, sobald die neuen Daten bereit sind.
- 3. Daten empfangen:**
Der letzte Empfänger hat die Daten empfangen und daraufhin die NDAC-Leitung freigegeben.
- 4. Daten-Bestätigung nicht gültig:**
Der erste Empfänger, der sieht, daß die Daten gültig sind (2), zieht NDAC auf LOW.
- 5. Bereit für Daten:**
Der letzte Empfänger hat erkannt, daß die Daten nicht mehr gültig sind, und daraufhin die Leitung NRFD freigegeben, um anzuzeigen, daß er bereit für den Empfang von Daten ist.
- 6. Nicht bereit für Daten:**
Der erste Empfänger, der Daten empfängt, nachdem sie gültig wurden (2), zieht NRFD auf LOW.
- 7.** Sobald er feststellt, daß die Daten nicht mehr gültig sind (1), zieht jeder Empfänger NDAC auf LOW (4) und gibt gleichzeitig NRFD (5) frei.
- 8.** Wenn er erkennt, daß die Daten gültig sind (2), zieht jeder Empfänger NRFD auf LOW (6) und gibt gleichzeitig NDAC frei.

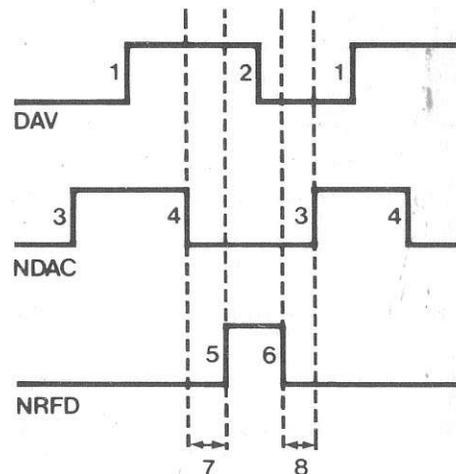


Bild 2-4:
Handshake-Protokoll

GPIB-Funktionen

Der Apple-II-GPIB verfügt über elf allgemeine Funktionen, die Geräte am Bus steuern und Daten übermitteln können. Jedes IEEE-488-kompatible Gerät ist in der Lage, zumindest eine dieser Funktionen auszuführen. Bitte beachten Sie, daß nicht alle Geräte, die Sie am Bus verwenden möchten, sämtliche aufgeführten Funktionen verwenden. Die allgemeinen Funktionen werden in den nachfolgenden Abschnitten kurz erläutert.

SHI = Source Handshake

Die Source-Handshake-Funktion verwendet die drei Transfer-Leitungen, um es einem Sender (außer dem Controller) zu ermöglichen, Daten zu einem oder mehreren Empfängern oder dem Apple über den Bus zu übertragen.

AHI = Acceptor Handshake

Die Acceptor-Handshake-Funktion verwendet die drei Transfer-Leitungen, um es einem Empfänger zu ermöglichen, Daten von einem Talker zu empfangen, der gewöhnlich der Apple ist.

T3 = BASIC Talker

Der Controller benutzt die Steuer- und Datenleitungen, um ein Gerät am Bus zu adressieren und es als Talker zu aktivieren. Sobald ein Gerät als Sender aktiviert ist, kann es Status- und Daten-Bytes über den Bus übertragen. Der Name dieser Funktion hat nichts mit der Programmiersprache BASIC zu tun.

L1 = BASIC Listener

Der Controller benutzt die Steuer- und Datenleitungen, um ein Gerät am Bus zu adressieren und es als Listener zu aktivieren. Sobald ein Gerät als Empfänger aktiviert ist, kann es Befehle und Daten vom Bus empfangen. Der Name dieser Funktion hat keinen Zusammenhang mit der Programmiersprache BASIC.

C4 = Respond to Service Request

Die Service-Request-Funktion verwendet die SRQ-Steuer-Leitung, um dem Controller anzuzeigen, daß sein Eingreifen erforderlich ist.

C3 = Remote Enable

Die Remote-Enable-Funktion ermöglicht die Fernsteuerung eines Gerätes durch den Apple, statt über die Frontplatten-Bedienungselemente des jeweiligen Gerätes.

C25 = Parallel Poll

Die Parallel-Poll-Funktion verwendet die 8 Daten-Bus-Leitungen, um bis zu 8 Geräte am Bus simultan abzufragen. Ein Parallel Poll erfolgt, um zu ermitteln, ob ein Gerät einen Service durch den Apple benötigt und um welches Gerät es sich dabei handelt. Der Parallel Poll ist wesentlich schneller als die Service-Request-Funktion, erfordert aber, daß der Apple häufig parallele Abfrage-Operationen durchführt. Viele der zur Zeit erhältlichen Geräte verfügen über keine Möglichkeiten für Parallel Poll.

DC = Device Clear

Die Device-Clear-Funktion ermöglicht es dem Controller, ein Gerät zu 'löschen' oder zu initialisieren (d.h. in einen bekannten Zustand zu versetzen). Die Device-Clear-Funktion ist unabhängig von der Interface-Clear-Funktion, die durch die IFC-Leitung gesteuert wird.

C2 Interface Clear

Die Interface Clear-Funktion setzt GPIB Interface-Karten-Controller und alle Geräte in einen definierten Zustand vor Einsatz dieser Einheiten.

DT Device Trigger

Diese Funktion wird vom Controller zur Synchronisierung der Operationen von einem oder mehreren am Bus angeschlossenen Geräten verwendet. Der Apple kann einen GET (group execute trigger) initialisieren, um eine Anzahl von Geräten am Bus zu synchronisieren. Der group execute trigger darf nicht mit der GET-Instruktion in BASIC verwechselt werden.

C1 Controller

Die Controller-Funktion erlaubt dem steuernden Gerät (Apple) die Übertragung von Adressen, universellen und adressierten Kommandos an die angeschlossenen Geräte. Es darf jedoch nur ein System-Controller am Bus angeschlossen werden. Der Apple fungiert als System-Controller.

ABSCHNITT 3

Programmierung -es GPIB

- 20 Einführung
- 21 Notation der Kommandos
- 24 Adressen
- 24 Empfängeradressen
- 24 Senderadressen
- 24 Sekundäradressen
- 25 Zur Benutzung von Adressen
- 25 Beschreibung der GPIB-Kommandos
- 25 Senden eines Kommandos
- 26 Kommando-Protokoll
- 27 Übertragung auf dem Bus
- 27 GPIB-Kommandos
- 27 Zeilenvorschub: LF
- 28 Stringende: ES
- 28 Bildschirm-Kommando: SC
- 28 Geräte-Kommando: DV
- 29 WRITE: WT
- 29 WRITECNT: WC
- 30 READ: RD
- 30 READCNT: RC
- 31 XFER: XF
- 31 TRIGR: TG
- 31 Clear-Kommando: CA
- 32 Selektives Rücksetzen: CL
- 32 Betriebsbereitschaft: RA
- 32 Gezieltes Setzen in den Betriebszustand: RM
- 33 Lokalsperrung: LL
- 33 Lokale Betriebsart: LA
- 33 Gezieltes Einschalten der lokalen Betriebsart: LO
- 33 Bedienungsanforderung: SR
- 34 Seriellles Polling: SP
- 34 Paralleles Polling: PP
- 35 Pollingbereitschaft: PE
- 35 Abschaltung des Polling: PD
- 36 PU
- 36 AB
- 36 Sender abschalten: UT
- 36 GPIB-Fehlermeldungen
- 37 Die Anweisung ONERR . . . GOTO

Dieses Kapitel enthält die programmiertechnischen Voraussetzungen zum Schreiben von BASIC-Programmen für den GPIB. Der erste Abschnitt vermittelt einen Überblick über die Arbeitsweise des Busses und der Zusammenarbeit mit den Geräten. Es folgt eine Zusammenfassung des Adressierungsschemas, das der Apple-II-GPIB benutzt. Daran anschließend finden Sie eine ausführliche Beschreibung der einzelnen Kommandos des GPIB. Der letzte Abschnitt des Kapitels behandelt Fehlermeldungen und ihre Ursachen.

Einführung

Dieser Abschnitt erklärt kurz die Programmierung der IEEE-488-Karte und einige Gemeinsamkeiten der verschiedenen Geräte. Ferner werden Sie darauf hingewiesen, was bei der Beschreibung der Geräte, die an den Bus angeschlossen werden sollen, zu beachten ist. Sie finden hier eine Übersicht über die Bus-Kommandos. All diese Kommandos werden später erklärt.

Vor der Benutzung des Busses muß ein PR#n oder IN#n-Kommando gegeben werden (für die IEEE-488-Karte im Einschubfach Nummer n). Bei der Verwendung eines dieser Kommandos werden in der Karte Standardwerte gesetzt:

Bildschirm aus (SCØ),

Zeilenvorschub aus (LFØ),

Gerätenummer Ø (DVØ) und

Zeichenbegrenzungssymbol auf Wagenrücklauf (<CR>, Carriage return).

Wenn Sie zum Beispiel PR#Ø und IN#Ø geben, um eine Anfrage des Bedieners zu beantworten, und anschließend die Ein-/Ausgabe zurück auf die Karte lenken wollen – ohne Setzen von Standardwerten –, können Sie das Unterprogramm von Programm 3, Anhang D verwenden.

Das Format der Ausgabedaten hängt vom jeweiligen Gerät ab. Manche Geräte verwenden <CR><LF> als Endezeichen, andere wiederum <CR> allein. Ein geeignetes Zeilenvorschubkommando LF1 oder LFØ sorgt für den korrekten Empfang der Daten. Einige Geräte senden Kommas oder Strichpunkte als Begrenzungszeichen zwischen einzelnen Daten oder zwischen Funktionscode und Daten. Applesoft beendet das Lesen, wenn eines dieser Zeichen gefunden wird. Deshalb müssen in der Input-Anweisung ebensoviele Variable verwendet werden, wie Kommas bzw. Strichpunkte im Eingabestring vorkommen. Wenn die Anzahl dieser Begrenzungszeichen variabel ist, muß das Unterprogramm 'Eingabe mit Kommas' in Programm 4, Anhang D verwandt werden.

Manche Geräte müssen vor der Eingabe durch Kommandos initialisiert werden: Die einen werden mit <CR><LF>, andere wiederum mit <CR> startklar gemacht. Manche Geräte müssen in den Zustand 'systemgesteuert' versetzt werden, bevor sie auf vom Bus übertragene Kommandos reagieren, wieder andere benötigen eine 'Lokalsperrung'. Mit den Kommandos RA, RM, LA, LL und LO können die Geräte in die verschiedenen Zustände versetzt werden.

Bestimmte Geräte müssen ein Trigger-Kommando (TG) empfangen, bevor sie eine Leseanforderung an den Bus senden. Die Daten werden gewöhnlich zum Triggerzeitpunkt in das Instrument gespeichert, bis sie von einem Gerät gelesen werden. Die Leseoperation löscht den Zwischenspeicher, bis ein neues Triggerkommando für dieses Gerät empfangen wird. Manche Geräte lesen die Daten fehlerhaft, wenn sie vorher nicht getriggert werden.

Andere Kommandos werden benutzt, um den Bus und die angeschlossenen Instrumente in einen definierten Zustand zu versetzen. Dies sind die Kommandos CA, CL, UT und AB. Sie setzen gewisse Zustände zurück, gewöhnlich die Sende- oder Empfangsbereitschaft.

Dies sind typische Bedingungen, die zu beachten sind, wenn ein Gerät über den Bus betrieben wird. Die Bedienungshandbücher der einzelnen Geräte geben exakt darüber Auskunft, welche Kommandos man geben muß, damit das Gerät korrekt arbeitet.



Dieser Abschnitt handelt über Kommandos, über die Sie jetzt noch wenig wissen. Deshalb schlagen wir vor, den Abschnitt zu wiederholen, wenn Sie die Kommandobeschreibungen weiter hinten durchgearbeitet haben.

Notation der Kommandos

Die Standardform eines GPIB-Kommandos ist eine Zeichenreihe. Jedes Zeichen ist ein 7-Bit-ASCII-Code, wie er von der Tastatur des Apple gesendet wird. Es wird in die sieben niederwertigen Bits eines Bytes übertragen. Das allgemeine Format einer Kommandozeichenreihe lautet:

<Kommando> <Steuerzeichen> <Daten>

<Kommando> ist zwei Zeichen lang und stellt einen mnemotechnischen Namen für ein GPIB-Kommando dar. <Steuerzeichen>- und <Daten>-Parameter werden nur bei einigen Kommandos verwendet.

Tabelle 3-1 enthält eine Zusammenstellung aller GPIB-Kommandos. Die linke Spalte listet die Kommandonamen auf. Unter der Überschrift 'Format' finden Sie die Form der Kommandozeichenreihe in Backus-Naur-Notation (BNF), und unter 'BASIC' steht, mit welcher BASIC-Anweisung das Kommando zu senden ist (PRINT oder INPUT).

Die GPIB-Kommandos werden von BASIC zur IEEE-488-Karte gesandt. Ob Daten auf dem Bus übertragen werden, hängt vom jeweiligen Kommando ab. In der rechten Spalte der Tabelle 3-1 – unter der Überschrift 'BUS?' – sind die Kommandos, die eine Datenübertragung bewirken, mit YES gekennzeichnet.

In Tabelle 3-2 finden Sie eine BNF-Beschreibung der Bestandteile <Steuerzeichen> und <Daten> der Kommandozeichenreihe. Eine in Klammern eingeschlossene Komponente heißt Deskriptor und ist in der Tabelle 3-2 durch ASCII-Zeichen und weitere Deskriptoren beschrieben.

Ein Symbol in BNF-Notation, das nicht in Klammern eingeschlossen ist, muß buchstabengetreu übernommen werden. Ein in Klammern eingeschlossener Deskriptor stellt ein Element dar oder eine Liste von Elementen, die an dieser Stelle in der Kommandozeichenreihe stehen müssen. Deskriptoren, die durch senkrechte Striche getrennt sind, sind untereinander austauschbar. Ein senkrechter Strich kann als das Wort 'oder' gelesen werden.



Zur besseren Lesbarkeit sind die Elemente der Kommandozeichenreihe durch Zwischenräume getrennt – sowohl in den Tabellen als auch in den späteren Kommandobeschreibungen. Gültige Kommandos dürfen jedoch keine solchen Zwischenräume enthalten, da diese als Empfängeradresse interpretiert werden könnten.

Beachten Sie besonders in Tabelle 3-2, daß die Parameter (listen list) und (count) durch <CTRL-Z>-Zeichen abgeschlossen werden. Dies ist nicht aus Tabelle 3-1 ersichtlich. Richten Sie Ihr Augenmerk auch auf die Verwendung der Symbole <CTRL-Z>, <CR> und <SP>. Sie werden erzeugt, indem Sie Z drücken, während Sie gleichzeitig die CTRL-Taste gedrückt halten und die RETURN- bzw. Leertaste drücken.

Die Tabellen 3-1 und 3-2 sind auch in der Übersichtskarte am Ende dieses Handbuchs enthalten.

Tabelle 3-1: GPIB-Kommandos (Überblick).

<u>COMMAND</u>	<u>FORMAT</u>	<u>BASIC</u>	<u>BUS?</u>
WRITE	WT <listen list> <data> <EOS>	PRINT	YES
WRITECNT	WC <listen list> <count> <data>	PRINT	YES
READ	RD <talker> <CTRL-Z> <data> <term>	INPUT	YES
READCNT	RC <talker> <CTRL-Z> <count> <data> <term>	INPUT	YES
XFER	XF <talker> <CTRL-Z> <listen list> <CR>	PRINT	YES
TRIGR	TG <listen list> <CR>	PRINT	YES
CLRAL	CA <CR>	PRINT	YES
CLEAR	CL <listen list> <CR>	PRINT	YES
REMAL	RA <CR>	PRINT	YES
REMDV	RM <listen list> <CR>	PRINT	YES
LLKAL	LL <CR>	PRINT	YES
LOCAL	LA <CR>	PRINT	YES
LOCDV	LO <listen list> <CR>	PRINT	YES
SRQD	SR <srq> <CR>	INPUT	YES
SROLL	SP <talker> <CTRL-Z> <status word> <CR>	INPUT	YES
PPOLL	PP <status word> <CR>	INPUT	YES
PPENB	PE <enable list> <CR>	PRINT	YES
PPDIS	PD <listen list> <CR>	PRINT	YES
PPUAL	PU <CR>	PRINT	YES
ABORT	AB <CR>	PRINT	YES
LINEFEED	LF <off/on> <CR>	PRINT	NO
EOS	ES <EOS> <CR>	PRINT	NO
SCREEN	SC <off/on> <CR>	PRINT	NO
DEVICE	DV <device number>	PRINT	NO
UNTALK	UT <CR>	PRINT	YES

Tabelle 3-2: Definitionen der GPIB-Kommando-Strings.

<listen list> ::= <CTRL-Z> | <listener> <CTRL-Z> | <listener> <listen list>
 <listener> ::= <primary listener> | <primary listener> <secondary address>
 * <primary listener> ::= <SP> | ! | " | # ... > | ?
 * <secondary address> ::= ' | a | b | c ... } | ~
 <count> ::= <decimal digit> <CTRL-Z> | <decimal digit> <count>
 NOTE: The maximum total value for count is 255.
 <EOS> ::= any ASCII character
 <data> ::= empty | any ASCII character | any ASCII character <data>
 <talker> ::= empty | <primary talker> | <primary talker> <secondary address>
 * <primary talker> ::= @ | A | B | C ... ^ | _
 * <hex digit> ::= 0 | 1 | 2 ... 9 | A | B ... F
 * <decimal digit> ::= 0 | 1 | 2 ... 9
 <status word> ::= <hex digit> <hex digit>
 <srq> ::= ASCII "T" | ASCII "F"
 <term> ::= empty | <CR> | <EOS> | EOI
 NOTE: empty if last char = <CR> or <CR><LF>
 <device number> ::= <decimal digit> <decimal digit>
 NOTE: Maximum device number is 30
 <off/on> ::= ASCII "0" | ASCII "1"
 <enable list> ::= <listener> <enable> | <listener> <enable> <enable list>
 * <enable> ::= @ | A | B ... N | O
 <CR> ::= ASCII carriage return (13 decimal)
 <LF> ::= ASCII linefeed (10 decimal)
 <SP> ::= ASCII space (32 decimal)
 <CTRL-Z> ::= ASCII SUB (26 decimal)
 <CTRL-D> ::= ASCII ET (4 decimal)

* NOTE: Marked elements are ASCII characters.

Adressen

Drei verschiedene Arten von Adressen können im Parameter <Steuerzeichen> vorkommen. Dies sind:

Empfängeradresse, Senderadresse und Sekundäradresse.

Die Adressen werden durch verschiedene Zeichen des ASCII-Codes dargestellt. Im folgenden Text sind die Zeichen für Adressen in Anführungszeichen eingeschlossen.

Empfängeradressen

Die Empfängeradressen im GPIB liegen zwischen den Hexadezimalwerten 20 und 3E. Dieser Bereich entspricht der Folge von ASCII-Zeichen von <SP> bis ">" (die auch auf der Tastatur vorhanden sind). Die dezimalen, oktalen und hexadezimalen Werte jeder gültigen Empfängeradresse finden Sie in Tabelle A-1 im Anhang A. Der Hexadezimalwert 3F (ASCII-Zeichen "?") ist reserviert für ein universelles Kommando "Empfänger abschalten" (UNL, UNListen). Dieses wird bei Bedarf vom Apple-II-GPIB automatisch gesendet.

Senderadressen

Es kann zu jedem Zeitpunkt nur einen Sender geben. Die Senderadresse ist im GPIB auf die Hexadezimalwerte 40 bis 5E beschränkt. Dieser Adreßbereich entspricht den ASCII-Zeichen (die von der Tastatur gesendet werden können) "@" bis "^". Die dezimalen, oktalen und hexadezimalen Werte aller Senderadressen finden Sie in Tabelle A-2 des Anhangs A. Der Hexadezimalwert 5F (ASCII-Zeichen "-") ist für ein universelles Kommando 'Sender abschalten' (UnTalk) reserviert, das mit dem UT-Kommando gesendet wird.

Sekundäradressen

Mit Sekundäradressen kann man die Arbeitsweise von Geräten bestimmen. Sie können auch zur Adressierung von Teileinheiten komplexer Geräte benutzt werden. So kann zum Beispiel der Arbeitsbereich und die Meßart eines digitalen Voltmeters vom Apple mit Hilfe von Sekundäradressen gesteuert werden.

Beim GPIB kann eine Sekundäradresse sowohl zum Sender als auch zum Empfänger geschickt werden. Wenn Sekundäradressen benutzt werden, sind sie Bestandteil der Parameter <Sender> oder <Empfängerliste>. Eine Sekundäradresse muß der zugehörigen Primäradresse unmittelbar folgen.

Beim GPIB sind die Sekundäradressen auf die Hexadezimalwerte 60 bis 7E beschränkt. Dieser Adreßbereich entspricht den ASCII-Zeichen "" bis "~" (wie sie von der Tastatur geschickt werden). Die dezimalen, oktalen und hexadezimalen Werte der Sekundäradressen stehen in Tabelle A-3 des Anhangs A.

Beim Applesoft sind die Zeichen für die Sekundäradressen jedoch nicht über die Tastatur zugänglich. Sie können statt dessen mit der CHR\$-Funktion erzeugt werden. Die CHR\$-Aufrufe sind in Tabelle A-3 des Anhangs A zusammengestellt. Beispiele zur Benutzung dieser Funktion finden Sie bei der Beschreibung der einzelnen Kommandos.

Bei Integer-Basic müssen die Sekundäradressen durch ein spezielles Unterprogramm erzeugt werden. Dieses ist in Programm 1, Anhang D dargestellt.

Zur Benutzung von Adressen

Sie müssen das Bedienungshandbuch des Geräts, das sie unter GPIB betreiben wollen, sorgfältig studieren, bis Sie sicher sind, daß Sie die Adressierungsmethode des Geräts verstanden haben. Die Geräteadressen können durch eingebaute Jumper voreingestellt sein, oder sie müssen durch Adreßschalter eingestellt werden. Es ist wichtig, herauszufinden, welche Adresse der Hersteller dem Gerät zugedacht hat.

Die Geräteadresse kann in binärer, hexadezimaler, oktaler oder ASCII-Form vorgegeben sein. Sie müssen feststellen, welches Zeichen der Apple-Tastatur dieser Adresse entspricht. Ist die Adresse nicht fest verdrahtet, sollten Sie die Adreßschalter oder Jumper auf eine passende Adresse einstellen. Adreßtabellen sind im Anhang A abgedruckt.

Jedem Gerät ist eine Gerätenummer zugeordnet. Diese ist vom Hersteller zugewiesen oder wie oben beschrieben eingestellt. Die Gerätenummer ist nur für die IEEE-488-Karte von Bedeutung.



Versichern Sie sich, daß kein Gerät die Nummer des Apple hat. Die Standardgerätenummer des Apple ist Ø (nicht ASCII-Zeichen Ø). Gerätenummer Ø entspricht der Empfängeradresse <SP> und der Senderadresse "@".

Beschreibung der GPIB-Kommandos

Tabelle 3-1 enthält eine Zusammenstellung aller Kommandos, die an den Apple-GPIB übertragen werden können. Es gibt drei Kommandotypen: Ausgabekommandos, Eingabekommandos und Kommandos zur Festlegung der Funktionsweise der IEEE-488-Karte. Der letzte Typ bewirkt keine Datenübertragung auf dem Bus. Die zwei rechten Spalten in Tabelle 3-1 zeigen den Typ jedes Kommandos an. Aus der Spalte mit der Überschrift 'BASIC' ersehen Sie, welche Basis-Anweisung – PRINT oder INPUT – für das in der Zeile angegebene Kommando zu verwenden ist.

Senden eines Kommandos

Kommandos, die Daten übertragen, haben die Form

```
PRINT "<Kommando> <Steuerzeichen> <Daten>"
```

Kommandos, die Daten von einem Gerät lesen, sehen so aus:

```
INPUT "<Kommando> <Steuerzeichen>"; S$  
oder
```

```
PRINT "<Kommando> <Steuerzeichen>";  
INPUT S$
```

S\$ ist ein dem Kommando angepaßter String.

Wenn in einem Programm Kommandos zum GPIB geschickt werden, muß zuvor die Slot-Nr. der Interface-Karte angezeigt werden. In den folgenden Beispielen wird die Nummer 3 angenommen. Das Kommando

```
PR#3
```

muß der ersten PRINT-Anweisung vorausgehen. Ebenso muß vor der ersten INPUT-Anweisung das Kommando

IN#3

eingegeben werden.

Auf diese einfache Weise können Sie den Apple anweisen, Daten an das Apple-II-GPIB zu senden oder von dort zu empfangen. Der GPIB befindet sich in unserem Beispiel im Einschubfach 3.



Gibt ein Programm ein DOS-Kommando, so muß dieses in einem PRINT-Statement mit <CTRL-D> als erstem Zeichen enthalten sein. Von den nachfolgenden Ausgabegrößen muß es durch <CR> getrennt werden.

Wenn Sie also DOS verwenden, müssen Sie die Anweisungsfolge

```
1Ø D$="": REM ZWISCHEN DEN ANFÜHRUNGSZEICHEN STEHT <CTRL-D>
2Ø PRINT D$;"PR#3"
3Ø PRINT D$;"IN#3"
```

der ersten PRINT- bzw. INPUT-Anweisung vorausschicken.

In den folgenden Beispielen verwenden wir die DOS-Form der Kommandos.



Die Kommandos PR#n und IN#n stellen auf der Karte die Standardwerte ein. Will man das verhindern, muß das Unterprogramm von Programm 3 in Anhang D verwandt werden.

Kommando-Protokoll

Wie bereits erwähnt, müssen <Empfängerliste> und <Zähler> mit <CTRL-Z> abgeschlossen werden. Für die folgenden Beispiele nehmen wir an, daß der Variablen Z\$ der Wert <CTRL-Z> zugewiesen wurde. Dies kann mit einer der Anweisungen

```
5 Z$="" : REM ZWISCHEN DEN ANFÜHRUNGSZEICHEN STEHT <CTRL-Z>
5 Z$=CHR$(26) : REM CHR$(26) ENTSpricht <CTRL-Z> BEI APPLESOFT
```

geschehen.

D\$ habe den Wert <CTRL-D> wie oben.

Wenn beim GPIB die Bildschirm-Option eingeschaltet ist, wird <CTRL-Z> als Z in invertierter Darstellung (dunkel/hell) angezeigt. Einzelheiten erfahren Sie bei der Beschreibung des Bildschirm-Kommandos in diesem Kapitel.

Eine Kommandozeilenreihe wird abgeschlossen mit <CR> oder einem Stringendezeichen <EOS> (end of string).

Die PRINT-Anweisung sendet als letztes Zeichen normalerweise <CR>. Wenn das <EOS>-Zeichen nicht das <CR> ist, oder bei den PRINT-Parametern sowohl Zeichenkonstanten als auch Variablen vorkommen, kann man das Zeichen ";" verwenden. Dieses bewirkt, daß die PRINT-Anweisung kein <CR> sendet.

Beispiel:

Es soll die Zeichenfolge "WT1", gefolgt von <CTRL-Z> und der Variablen A\$ ohne dazwischenliegende <CR> geschickt werden.

Dies geschieht mit der Anweisung

```
1Ø PRINT "WT1"; Z$; A$ : REM DER WERT VON Z$ IST <CTRL-Z>
```

Diese ist gleichwertig zu

```
10 PRINT "WT1";
11 PRINT Z$;
12 PRINT A$
```

Man kann natürlich Steuerzeichen in einen beliebigen String einbauen, da aber solche Zeichen nicht sichtbar sind, wurde obige Lösung gewählt.

Übertragung auf dem Bus

Bei der Ausführung eines Kommandos durch GPIB wird eine Folge von Daten und Steuersignalen auf den 16 Daten- und Steuerleitungen übertragen. Beispiele für die Ein-/Ausgabefolgen aller GPIB-Kommandos sind im Anhang B enthalten. Die Bytes, die über die DIO-Leitungen geschickt werden, sind Daten und IEEE-488-Kommandos. Wegen Einzelheiten vergleichen Sie den IEEE-Standard.

GPIB-Kommandos

Zuerst werden vier Kommandos besprochen - Zeilenvorschub-, Stringende-, Bildschirm- und Geräte-Kommando -, die die Funktionsweise der Karte beeinflussen. Diese Kommandos veranlassen zwar selbst keine Datenübertragung auf dem Bus, sie haben jedoch Auswirkung auf die Daten, die von anderen Kommandos gesendet werden.

Die übrigen Kommandos veranlassen eine Datenübertragung auf dem Bus. Dies kann ein Informationsfluß von einem Gerät zu einem anderen sein, außer auch Kommandos, die die Arbeitsweise der am Bus angeschlossenen Geräte steuern.

Zeilenvorschub: LF (linefeed)

Zeilenvorschubsteuerung. Dieses Kommando veranlaßt die Baugruppe, nach jedem Wagenrücklauf (<CR>) ein Zeilenvorschubzeichen (<LF>) zu senden.

Es hat die Form:

```
LF <aus = 0/ein = 1> <CR>
```

Wird bei eingeschalteter Zeilenvorschubsteuerung ein WRITE-Kommando gegeben, so sendet die IEEE-488-Karte nach jedem <CR>-Zeichen ein <LF>, auch dann, wenn dem <CR> bereits ein <LF>-Zeichen folgt. Das Zeilenvorschub-Kommando beeinflusst jedoch nicht das WRITECNT-Kommando. Ist die Zeilenvorschubsteuerung eingeschaltet und es wird ein READ-Kommando abgesetzt, so erwartet die IEEE-488-Karte, daß jedem abschließenden <CR> ein <LF> folgt.

Beispiel:

```
10 PRINT D$;"PR#3" : REM D$ HAT ALS WERT <CTRL-D>
20 PRINT "LF1" (ein <LF>-Zeichen wird nach jedem <CR> erzeugt)

20 PRINT "LF0" (nach einem <CR> wird kein <LF>-Zeichen erzeugt)
```

Die Zeilenvorschubsteuerung wird ausgeschaltet, wenn das <EOS>-Zeichen nicht mehr das <CR> ist.

Stringende: ES

Stringenzeichen. Mit dem EOS-Kommando wird die Definition des Zeichens geändert, das einen String abschließt. Standardmäßig ist <CR> das Stringenzeichen. Eine Rücksetzanweisung oder PR#3 setzt das Stringendesymbol wieder auf <CR>.

Hat das Symbol einen von <CR> verschiedenen Wert, wird die Zeilenvorschubsteuerung ausgeschaltet.

Form des EOS-Kommandos:

```
ES <EOS> <CR>
```

Beispiel:

```
1Ø PRINT D$;"PR#3" : REM DER WERT VON D$ IST <CTRL-D>
2Ø PRINT "ES!"
```

Damit ist '!' das neue Stringenzeichen.

Bildschirm-Kommando: SC (SCREEN)

Bildschirmsteuerung. Das Bildschirm-Kommando steuert den Bildschirm während der Tätigkeit des GPIB. Bei eingeschalteter Bildschirmsteuerung werden alle ein- und ausgegebenen Zeichen gezeigt. Dies ist beim Testen eines Programms sehr nützlich. Standardmäßig ist die Bildschirmsteuerung ausgeschaltet. Das Kommando hat die Form:

```
SC <aus = 0/ein = 1> <CR>
```

Beispiel:

```
1Ø PRINT D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>
2Ø PRINT "SC1" (die Daten werden am Bildschirm gezeigt)
2Ø PRINT "SC0" (die Daten werden am Bildschirm nicht angezeigt)
```

Geräte-Kommando: DV (DEVICE)

Gerätenummer des Steuerteils. Mit diesem Kommando wird die Gerätenummer des Apple-Steuerteils festgelegt. Normalerweise ist die Gerätenummer 0, und auf diesen Wert wird sie auch bei einer PR#n-Anweisung oder durch Drücken der RESET-Taste zurückgesetzt.

Die Nummer wird modulo 32 verwandt. Sie kann aus einer oder zwei Dezimalziffern im ASCII-Format bestehen. Bei einer Zahl bis zu 09 kann die führende Null weggelassen werden. Die Gerätenummer 31 (und damit 63, 95, . . .) darf nicht verwendet werden, da sie für die Kommandos Abschalten Empfänger (UNLISTEN) und Abschalten Sender (UNTALK) reserviert ist.

Das Geräte-Kommando hat die Form:

```
DV <Gerätenummer> <CR>
```

Beispiel:

```
1Ø PRINT D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>
2Ø PRINT "DV01"
```

WRITE: WT

Ausgeben von Daten. Das WRITE-Kommando schickt Daten zu allen in <Empfängerliste> aufgeführten Geräten. Dieses ist das übliche Kommando zur Datenübertragung vom Apple zu einem peripheren Gerät. Das Kommando hat die Form:

```
WT <Empfängerliste> <Daten> <EOS>
```

Mit dem letzten Zeichen (das ist das <EOS>-Zeichen) wird automatisch ein <EOI> (End-Or-Identify)-Bussignal gesendet. Normalerweise ist <CR> das <EOS>-Zeichen. Ist die Zeilenvorschubsteuerung eingeschaltet (siehe LF-Kommando) und <CR> ist das <EOS>-Zeichen, dann wird <EOI> zusammen mit dem <LF>-Zeichen gesendet.

Es folgt ein Beispiel:

```
10 PRINT D$;"PR#3" : REM D$ HAT ALS WERT <CTRL-D>  
30 PRINT "WT1"; Z$; A$ : REM Z$ HAT ALS WERT <CTRL-Z>
```

Das Kommando sendet den Datenstring A\$ an dem Empfänger mit Adresse 1 (dezimal 49, oktal 61 bzw. hexadezimal 31).

```
40 PRINT "WT8"; CHR$(97); Z$; A$ : REM Z$ ENTHÄLT <CTRL-Z>
```

Dieses Kommando sendet den Datenstring A\$ zum Empfänger mit Primäradresse "8" und Sekundäradresse "a". Um Kleinbuchstaben zu erzeugen, ist die CHR\$-Funktion zu verwenden. Eine Übersicht über die Werte der CHR\$-Funktion finden Sie in Tabelle A-3 in Anhang A. Der ASCII-Code für "a" ist 97, folglich erzeugt CHR\$(97) ein "a".

Mit Programm 1 in Anhang D können Sie auch mit Integer-Basic-Kleinbuchstaben senden.

WRITECNT: WC

Senden von Daten mit einem Zähler. Das WRITECNT-Kommando arbeitet ähnlich wie das WRITE-Kommando. Es wird jedoch ein Zähler für die Anzahl der Datenbytes im String mitgeführt. Das Kommando hat die Form:

```
WC <Empfängerliste> <Zähler> <Daten>
```

Das <EOS>-Zeichen (gewöhnlich <CR>) wird im Normalfall nicht am Ende des Datenstrings gesendet, dafür aber das Bussignal EOI. Der Zähler muß größer 0 und kleiner als 256 sein.

Es folgt ein Beispiel:

```
10 D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>  
20 PRINT "WC1"; Z$; LEN(B$); Z$; B$ : REM Z$ ENTHÄLT <CTRL-Z>
```

Mit diesem Kommando wird der String B\$ in der Länge LEN(B\$) zum Empfänger mit der Adresse '1' geschickt.



Ein wichtiger Einsatzfall des WRITECNT-Kommandos ist, Daten ohne abschließendes <EOS>-Zeichen zu senden. Das ist mit dem WRITE-Kommando nicht möglich.



Ist die Länge des Strings größer als der Wert des Zählers, dann werden die überschüssigen Zeichen als neues Kommando betrachtet und verursachen einen Fehler. Ist die Länge des Strings zu klein, wird das nächste Kommando als Teil des Datenstrings mitgeschickt. Dies verursacht wiederum einen Fehler. Ist die PRINT-Anweisung nicht mit einem ";" abgeschlossen,

sen, wird automatisch ein <CR> mitgeschickt. Das muß beim Zähler berücksichtigt werden. Ebenso wird bei eingeschalteter Zeilenvorschubsteuerung <CR><LF> mitgeschickt.

READ: RD

Lesen von Daten. Mit dem READ-Kommando empfängt man Daten vom angegebenen Sender. Dies ist das übliche Kommando zum Lesen von Daten von einem Gerät. Das Kommando hat die Form:

```
RD <Sender> Z$ <Daten> <term>
```

Die Eingabe wird beendet, wenn ein <CR> oder <EOS>-Zeichen empfangen wird oder ein EOI-Signal.

Die IEEE-488-Karte gibt immer ein <CR> an das Programm zurück, um die Dateneingabe zu beenden. Ist das letzte gelesene Zeichen bereits ein <CR>, wird kein zusätzliches <CR>-Zeichen gesendet. Ein Zeilenvorschubzeichen vom GPIB wird unterdrückt und gelangt nicht zum Programm.

Es folgt ein Beispiel:

```
5   D$;"PR#3" : REM  D$ ENTHÄLT ALS WERT <CTRL-D>
10  D$;"IN#3" : REM  D$ ENTHÄLT ALS WERT <CTRL-D>
20  PRINT "RDA"; Z$; : REM  Z$ ENTHÄLT <CTRL-Z>
25  INPUT  C$
```

Mit diesem Kommando wird der Datenstring C\$ vom Sender mit der Adresse "A" empfangen.

```
5   D$;"PR#3" : REM  D$ HAT DEN WERT <CTRL-D>
10  D$;"IN#3" : REM  D$ HAT DEN WERT <CTRL-D>
30  PRINT "RDA"; CHR$(98); Z$; : REM  Z$ ENTHÄLT <CTRL-Z>
35  INPUT  C$
```

Mit diesem Kommando wird der Datenstring C\$ vom Sender mit der Adresse "A" und der Sekundäradresse "b" empfangen. Den Kleinbuchstaben "b" kann man mit CHR\$(98) erzeugen.



Beachten Sie den Strichpunkt, der die letzten zwei PRINT-Anweisungen abschließt. Wenn mit einer PRINT-Anweisung ein Kommando geschickt wird und mit INPUT die daraufhin auf den Bus gesendeten Daten gelesen werden, muß das normalerweise mit dem PRINT geschickte <CR> unterdrückt werden.

READCNT: RC

Lesen von Daten mit Zählen. Das READCNT-Kommando arbeitet ähnlich wie das READ-Kommando. Es wird ein Zähler mitgeführt, der die Länge des Eingabestrings in Byte angibt. Das Kommando hat die Form:

```
RC <Sender> Z$ <Zähler> <term>
```

Der Sender kann EOI mit dem letzten Zeichen des Strings schicken. Der Datenstring darf weniger Zeichen enthalten als der Zähler angibt, wenn vom Sender EOI oder <EOS> benutzt wird.

Die IEEE-488-Karte schickt immer ein <CR> als Abschluß der Eingabedaten an das Programm. Wenn das letzte übertragene Zeichen bereits ein <CR> ist, wird kein zusätzliches übertragen.

Es folgt ein Beispiel:

```

9   D$;"PR#3" : REM  D$ HAT DEN INHALT <CTRL-D>
10  D$;"IN#3" : REM  D$ HAT DEN INHALT <CTRL-D>
20  PRINT "RCB"; Z$; "26"; Z$; : REM  Z$ ENTHÄLT <CTRL-Z>
25  INPUT C$

```

Mit diesem Kommando erhält man einen 26 Byte langen Datenstring vom Sender mit der Adresse "B".



Bei Verwendung des RC- oder RD-Kommandos bei eingeschalteter Zeilenvorschubsteuerung versichern Sie sich, daß das Gerät nicht eine Null (ASCII-Code 00) schickt, während GPIB <LF> erwartet.

XFER: XF

Datenübertragung. Das XFER-Kommando erlaubt direkte Übertragung von Daten zwischen externen Geräten. Zuerst wird der Sender adressiert, dann der Empfänger. Die Geräteadresse des Apple (standardmäßig 0) darf weder als Sender noch als Empfänger vorkommen. Das Kommando hat die Form.

```
XF <Sender> Z$ <Empfängerliste> <CR>
```

Der Sender muß so programmiert sein, daß er EOI als letztes Zeichen schickt. Daran erkennen die Empfänger das Ende der Übertragung. Dies erlaubt auch dem Apple, die Steuerung über den GPIB zurückzugewinnen.

Beispiel:

```

10  PRINT D$;"PR#3" : REM  DER WERT VON D$ IST <CTRL-D>
20  PRINT "XFA"; Z$; "1"; Z$

```

Das Kommando überträgt direkt Daten vom Sender mit der Adresse "A" zum Empfänger mit der Adresse "1".

TRIGR: TG

Triggern einer Gruppe von Geräten. Das Kommando macht eine Gruppe von Geräten betriebsklar und veranlaßt das gleichzeitige Starten dieser Geräte. Das Kommando hat folgende Form:

```
TG <Empfängerliste> <CR>
```

Beispiel:

```

10  PRINT D$;"PR#3" : REM  D$ HAT ALS WERT <CTRL-D>
20  PRINT "TC"; "19"; Z$

```

Mit dem Kommando werden die Empfänger mit den Adressen "1" und "9" simultan getriggert.

Clear-Kommando: CA

Rücksetzen aller Geräte. Das Clear-Kommando gibt eine universelle Rücksetzmeldung an alle Geräte aus.

Das Kommandoformat lautet:

```
CA <CR>
```

Ein Beispiel zu diesem Kommando:

```
1Ø PRINT D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>
2Ø PRINT "CA"
```

Dieses Kommando setzt sämtliche Geräte am GPIB zurück.

Selektives Rücksetzen: CL

Gezieltes Rücksetzen von Geräten. Das Kommando setzt alle Geräte zurück, die in der Adreßliste angegeben sind.

Die Form des Kommandos wird beschrieben durch

```
CL <Empfängerliste> <CR>
```

Wir geben ein Beispiel:

```
1Ø PRINT D$;"PR#3" : REM WERT VON D$ IST <CTRL-D>
2Ø PRINT "CL"; "19"; Z$
```

Das Kommando schickt eine Rücksetzmeldung an die Geräte mit den Adressen "1" und "9".

Betriebsbereitschaft: RA

Versetze alle Geräte in Betriebsbereitschaft. Mit dem Kommando wird die REN-Steuerleitung (Anzeige für Betriebsbereitschaft) auf true gesetzt. Erst wenn das Gerät als Empfänger adressiert wird, arbeitet es unter der Steuerung des Systems.

Die Syntax ist einfach

```
RA <CR>
```

Ein Beispiel:

```
1Ø PRINT D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>
2Ø PRINT "RA"
```



Alle folgenden Empfängeradressen werden mit dem REN-Signal true gesendet. Es wird erst wieder mit dem LA-Kommando auf den Wert false gesetzt.

Gezieltes Setzen in den Betriebszustand: RM

Ausgewählte Geräte in den Betriebszustand versetzen. Das Kommando setzt die REN-Steuerleitung auf true und sendet die Adressen der angegebenen Empfänger zum GPIB.

Kommandoformat:

```
RM <Empfängerliste> <CR>
```

Es folgt ein Beispiel:

```
1Ø PRINT D$;"PR#3" : REM D$ HAT ALS WERT <CTRL-D>
15 E$ = "12349" + Z$
2Ø PRINT "RM"; E$
```

Damit wird für die Geräte mit den Adressen "1", "2", "3", "4" und "9" der abhängige Betrieb eingeschaltet.

Die Form des Kommandos:

• SR <srq> <CR>

Anwendungsbeispiel:

```
5 PRINT D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>
10 PRINT D$;"IN#3" : REM D$ HAT DEN WERT <CTRL-D>
20 INPUT "SR"; G$
```

Der Wert, der in G\$ zurückgegeben wird, ist entweder "T" oder "F".

Serielles Polling: SP

Das Kommando sendet ein serielles Polling an ein adressiertes Gerät des GPIB. Es wird üblicherweise benutzt, wenn "T" das Ergebnis eines vorangegangenen SR-Kommandos war. Das angesprochene Gerät übergibt ein Zustandsbyte, das anzeigt, ob ein Bedienungswunsch für das Gerät ansteht. Die IEEE-488-Karte wiederum übersetzt das Zustandsbyte in ein zwei Byte langes Zustandswort.

Das Kommandoformat lautet:

SP <Sender> Z\$ <Zustandswort> <CR>

Beispiel:

```
5 PRINT D$;"PR#3" : REM DER WERT VON D$ IST <CTRL-D>
10 PRINT D$;"IN#3" : REM DER WERT VON D$ IST <CTRL-D>
20 PRINT "SPA"; Z$;
30 INPUT H$
```

Nach H\$ wurde ein Zustandswort von einem Gerät mit Adresse "A" übertragen. Die Länge von H\$ beträgt zwei Bytes. Das erste Byte im ASCII-Format enthält die vier führenden Bits des binären Zustandsworts und stellt eine Hexadezimalzahl zwischen 0 und F dar. Das zweite Byte ist die ASCII-Form der vier letzten Bits des binären Zustandsworts. Während der Ausführung des SP-Kommandos wird <CTRL-Z> nicht am Bildschirm angezeigt. Welches Zustandsbyte ein Gerät überträgt, wird vom Hersteller bestimmt.

Wenn das angesprochene Gerät SRQ gesandt hatte, setzt es nun statt dessen das DIO7-Bit im Statusbyte. Das bedeutet, daß das erste Zeichen von H\$ 4,5,6,7,C,D,E oder F ist. Die Bedeutung der anderen DIO-Bits ist geräteabhängig.

Paralleles Polling: PP

Mit diesem Kommando veranlassen Sie ein paralleles Polling. Als Ergebnis wird ein Statusbyte zur IEEE-Karte geschickt. Jedes der acht Bits kann einem Gerät zugeordnet werden. Dazu kann man das PE-Kommando mit Angabe der entsprechenden Geräte benutzen oder auch einen DIP-Schalter bei Geräten, die die PP2-Untermenge des IEEE-488-Standard realisieren. Die IEEE-488-Karte übersetzt das Zustandsbyte in ein zwei Bytes langes Zustandswort.

Das Kommandoformat lautet:

PP <Zustandswort> <CR>

Es folgt ein Beispiel.

PU

Auflösen der Konfiguration des parallelen Pollings für alle Geräte. Die Konfiguration wird so geändert, daß die Geräte nicht mehr auf paralleles Polling antworten.

Ein Beispiel:

```
10 PRINT D$;"PR#3" : REM DER WERT VON D$ IST <CTRL-D>
20 PRINT "PU"
```

AB

Rücksetzen aller Schnittstellen. Das Kommando wird nur in Notfällen zum Rücksetzen aller Schnittstellen benutzt. Es setzt die IFC-Steuerleitung auf true und setzt damit alle Geräte in einen definierten Zustand. Der IFC-Impuls dauert ungefähr 5 Millisekunden.

Beispiel:

```
10 PRINT D$;"PR#3" : REM D$ HAT DEN WERT <CTRL-D>
20 PRINT "AB"
```

Sender abschalten: UT

Das Kommando sendet ein universelles Abschalt-(untalk-)Signal. Das ist für einige Geräte erforderlich. Lesen Sie hierzu sorgfältig die Bedienungshandbücher der einzelnen Geräte.

Es folgt ein letztes Beispiel:

```
10 PRINT D$;"PR#3" : REM DER WERT VON D$ IST <CTRL-D>
20 PRINT "UT"
```

GPIB-Fehlermeldungen

Trifft der GPIB eine nicht interpretierbare Zeichenfolge an, gibt er am Bedienplatz eine Fehlermeldung aus. Das Programm wird beendet, es sei denn, ein Bus-Fehler tritt bei gesetzter ONERR-Anzeige auf. Es gibt vier verschiedene Fehlermeldungen. Diese sind anschließend zusammen mit den typischen Ursachen aufgeführt.

Bei der Anzeige am Bildschirm werden die drei letzteren Fehlertypen mit der Buchstabenfolge "ERR" versehen.

CMD — Die ersten beiden Zeichen stellen keinen gültigen Namen für ein Kommando dar. Das kann zum Beispiel vorkommen, wenn <CR> nicht mehr das <EOS>-Zeichen ist und bei einem vorangegangenen WRITE-Kommando der abschließende Strichpunkt zur Unterdrückung des <CR> fehlte. Eine denkbare Ursache ist auch ein unerwartetes Komma in einer Eingabezeichenfolge, die das Beenden der INPUT-Anweisung bewirkt. Auch ein Systemfehler während einer Eingabeanweisung kann einen CMD-Fehler hervorrufen. Eine Systemfehlermeldung beginnt nämlich mit einem Fragezeichen, was als ungültiges Kommandozeichen interpretiert wird.

LST — Die Steuerzeichen und/oder Daten halten nicht das vom Kommando erwartete Protokoll ein. Dieser Fehler wird normalerweise von einem fehlenden <CTRL-Z> oder aufeinanderfolgenden Sekundäradressen verursacht.

CNT — Der Zähler liegt außerhalb des gültigen Bereichs (gewöhnlich ≥ 256). Dieser Fehler tritt nur bei den Kommandos WC, RC und DV auf.

```

5   PRINT D$;"PR#3"   : REM  DER WERT VON D$ IST <CTRL-D>
1Ø  PRINT D$;"IN#3"   : REM  DER WERT VON D$ IST <CTRL-D>
2Ø  INPUT "PP"; I$

```

In I\$ wird ein zwei Byte langes Zustandswort übergeben. Das erste Byte ist die ASCII-Form der vier führenden Bits des binären Zustandsworts und stellt eine Hexadezimalzahl zwischen 0 und F dar. Das zweite Byte entspricht den vier letzten Bits des binären Zustandsworts.

Pollingbereitschaft: PE

Freigabe des parallelen Polling. Das Kommando definiert die Konfiguration der Geräte, die die PP1-Untermenge des IEEE-488-Standard realisieren. Es weist jedem Gerät, das bereit zum Polling werden soll, die Nummer eines Bits und dessen Bedeutung zu.

```
PE <Pollingliste> <CR>
```

Ein Element der Pollingliste besteht aus Empfängeradresse und Bereitschaftsbyte. Letzteres hat die Form 0100SPQR.

Das Bit S legt die Bedeutung des Pollingbits im Statusbyte des Empfängers fest (siehe oben), d. h. ob das Gerät eine Bedienungsanforderung mit 0 oder mit 1 anzeigt. Die Bits PQR, als Zahl zwischen 0 und 7 aufgefaßt, ordnen dem Gerät das entsprechende Bit des Zustandsbytes zu. Das Bereitschaftsbyte entspricht einem ASCII-Zeichen zwischen "@" und "Ø". Zur Anpassung an den IEEE-Standard führt die IEEE-Karte eine logische Oder-Verbindung des Statusbytes mit dem Hexadezimalwert 20 aus. Erst dann wird das Byte auf dem Bus gesendet.

Beispiel:

```

1Ø  PRINT D$;"PR#3"   : REM  DER WERT VON D$ IST <CTRL-D>
2Ø  PRINT "PE"; "1A2B3K4L"

```

Das Kommando im Beispiel erzeugt folgende Konfiguration:

Empfänger	verwendetes Bit im Zustandsbyte	Anzeige für Bedie- nungsanforderung
"1"	1	Ø
"2"	2	Ø
"3"	3	1
"4"	4	1

Abschaltung des Polling: PD

Abschalten des parallelen Pollings. Nachdem dieses Kommando gegeben wurde, antworten die angesprochenen Geräte nicht mehr auf PP-Kommandos.

Das Kommando hat folgendes Format:

```
PD <Empfängerliste> <CR>
```

Wir bringen dazu ein Beispiel:

```

1Ø  PRINT D$;"PR#3"   : REM  DER WERT VON D$ IST <CTRL-D>
2Ø  PRINT "PD"; "1234"; Z$

```

Das Kommando verhindert, daß die Empfänger mit den Adressen "1", "2", "3" und "4" auf paralleles Polling antworten.

BUS — Fehler durch unvollständige Senderquittung: Er tritt auf, wenn ein Byte auf den GPIB gelangt und gleichzeitig NDAC und NRFD gesetzt sind. Das kommt vor, wenn der Bus nicht richtig angeschlossen oder das adressierte Gerät ausgeschaltet ist. Ein Bus-Fehler kann mit Hilfe der Anweisung ONERR . . . GOTO abgefangen werden.

Die Anweisung ONERR . . . GOTO

Von den vier Fehlerzuständen, die das GPIB behandelt, kann nur der Bus-Fehler abgefangen werden. Benutzen Sie unter APPLESOFT in Ihrem Programm die Anweisung

ONERR GOTO zeilennummer ,

findet die GPIB-Firmware die ONERR-Anzeige und überträgt die Steuerung an eine geeignete Stelle in der APPLESOFT-Firmware. Von dort gelangt die Steuerung weiter zur Anweisung mit der Nummer zeilennummer wie bei einem normalen APPLESOFT-Fehler. An diese Stelle können Sie eine beliebige Fehlerbehandlungsroutine einfügen. Ein Beispiel für eine solche Routine finden Sie im Programm 2 in Anhang D.

Im APPLESOFT-II-Programmierhandbuch ist die Anweisung ONERR . . . GOTO genauer erklärt. Tabelle Anhang A.

ANHANG A

ASCII-Umwandlungs-Tabelle

- 40 Tabelle A-1 GPIB-LISTNER-ADRESSEN
- 41 Tabelle A-2 GPIB-TALKER-ADRESSEN
- 42 Tabelle A-3 GPIB-SEKUNDÄR-ADRESSEN
- 43 Tabelle A-4 ASCII-STEUER-CODES

ANHANG A

ASCII-Umwandlungs-Tabelle

- 40 Tabelle A-1 GPIB-LISTNER-ADRESSEN
- 41 Tabelle A-2 GPIB-TALKER-ADRESSEN
- 42 Tabelle A-3 GPIB-SEKUNDÄR-ADRESSEN
- 43 Tabelle A-4 ASCII-STEUER-CODES

Tabelle A-1 GPIB-Listener-Adressen

<u>DEC</u>	<u>OCT</u>	<u>HEX</u>	<u>CHAR</u>	<u>WHAT TO TYPE</u>
32	Ø4Ø	2Ø	SPACE	space
33	Ø41	21	!	!
34	Ø42	22	"	"
35	Ø43	23	#	#
36	Ø44	24	\$	\$
37	Ø45	25	%	%
38	Ø46	26	&	&
39	Ø47	27	'	'
4Ø	Ø5Ø	28	((
41	Ø51	29))
42	Ø52	2A	*	*
43	Ø53	2B	+	+
44	Ø54	2C	,	,
45	Ø55	2D	-	-
46	Ø56	2E	.	.
47	Ø57	2F	/	/
48	Ø6Ø	3Ø	Ø	Ø
49	Ø61	31	1	1
5Ø	Ø62	32	2	2
51	Ø63	33	3	3
52	Ø64	34	4	4
53	Ø65	35	5	5
54	Ø66	36	6	6
55	Ø67	37	7	7
56	Ø7Ø	38	8	8
57	Ø71	39	9	9
58	Ø72	3A	:	:
59	Ø73	3B	;	;
6Ø	Ø74	3C	<	<
61	Ø75	3D	=	=
62	Ø76	3E	>	>
63	Ø77	3F	?	Reserved for UNLISTEN

DEC = ASCII decimal code
 OCT = ASCII octal code
 HEX = ASCII hexadecimal code
 CHR\$ = Applesoft function

Tabelle A-2 GPIB-Sender-Adressen

<u>DEC</u>	<u>OCT</u>	<u>HEX</u>	<u>CHAR</u>	<u>WHAT TO TYPE</u>
64	100	40	@	@
65	101	41	A	A
66	102	42	B	B
67	103	43	C	C
68	104	44	D	D
69	105	45	E	E
70	106	46	F	F
71	107	47	G	G
72	110	48	H	H
73	111	49	I	I
74	112	4A	J	J
75	113	4B	K	K
76	114	4C	L	L
77	115	4D	M	M
78	116	4E	N	N
79	117	4F	O	O
80	120	50	P	P
81	121	51	Q	Q
82	122	52	R	R
83	123	53	S	S
84	124	54	T	T
85	125	55	U	U
86	126	56	V	V
87	127	57	W	W
88	130	58	X	X
89	131	59	Y	Y
90	132	5A	Z	Z
91	133	5B	[CHR\$(91)
92	134	5C	\	CHR\$(92)
93	135	5D]] (shift-M)
94	136	5E	^	^
95	137	5F	_	Reserved for UNTALK

Tabelle A-3 GPIB-Sekundär-Adressen

<u>DEC</u>	<u>OCT</u>	<u>HEX</u>	<u>CHAR</u>	<u>WHAT TO TYPE</u>
96	140	60	'	CHR\$(96)
97	141	61	a	CHR\$(97)
98	142	62	b	CHR\$(98)
99	143	63	c	CHR\$(99)
100	144	64	d	CHR\$(100)
101	145	65	e	CHR\$(101)
102	146	66	f	CHR\$(102)
103	147	67	g	CHR\$(103)
104	150	68	h	CHR\$(104)
105	151	69	i	CHR\$(105)
106	152	6A	j	CHR\$(106)
107	153	6B	k	CHR\$(107)
108	154	6C	l	CHR\$(108)
109	155	6D	m	CHR\$(109)
110	156	6E	n	CHR\$(110)
111	157	6F	o	CHR\$(111)
112	160	70	p	CHR\$(112)
113	161	71	q	CHR\$(113)
114	162	72	r	CHR\$(114)
115	163	73	s	CHR\$(115)
116	164	74	t	CHR\$(116)
117	165	75	u	CHR\$(117)
118	166	76	v	CHR\$(118)
119	167	77	w	CHR\$(119)
120	170	78	x	CHR\$(120)
121	171	79	y	CHR\$(121)
122	172	7A	z	CHR\$(122)
123	173	7B	{	CHR\$(123)
124	174	7C		CHR\$(124)
125	175	7D	}	CHR\$(125)
126	176	7E	~	CHR\$(126)

Tabelle A-4 ASCII-Steuer-Codes

<u>DEC</u>	<u>OCT</u>	<u>HEX</u>	<u>CHAR</u>	<u>WHAT TO TYPE</u>
0	000	00	NULL	CTRL-@
1	001	01	SOH	CTRL-A
2	002	02	STX	CTRL-B
3	003	03	ETX	CTRL-C
4	004	04	ET	CTRL-D
5	005	05	ENQ	CTRL-E
6	006	06	ACK	CTRL-F
7	007	07	BEL	CTRL-G
8	010	08	BS	CHR\$(8)
9	011	09	HT	CTRL-I
10	012	0A	LF	CTRL-J
11	013	0B	VT	CTRL-K
12	014	0C	FF	CTRL-L
13	015	0D	CR	CTRL-M or RETURN
14	016	0E	SO	CTRL-N
15	017	0F	SI	CTRL-O
16	020	10	DLE	CTRL-P
17	021	11	DC1	CTRL-Q
18	022	12	DC2	CTRL-R
19	023	13	DC3	CTRL-S
20	024	14	DC4	CTRL-T
21	025	15	NAK	CHR\$(21)
22	026	16	SYN	CTRL-V
23	027	17	ETB	CTRL-W
24	030	18	CAN	CTRL-X
25	031	19	EM	CTRL-Y
26	032	1A	SUB	CTRL-Z
27	033	1B	ESCAPE	ESC
28	034	1C	FS	CHR\$(28)
29	035	1D	GS	CTRL-SHIFT-M
30	036	1E	RS	CTRL-SHIFT-N
31	037	1F	US	CHR\$(31)

ANHANG B

Bus-Ein-/Ausgabefolgen

Dieser Anhang beschreibt die Ein-/Ausgabefolgen des GPIB und gibt ein Beispiel für jedes Kommando. Die DIO-Mnemotechnik sind im IEEE-488-Standard definierte Meldungen. Hier finden Sie die vom Apple verwandten Meldungen. Weitere Informationen stehen im IEEE-488-Standard.

<u>DIO Command</u>	<u>Hex</u>	<u>Binary</u>	<u>Translation</u>
DCL	14	X0010100	Device CLear
GET	08	X0001000	Group Execute Trigger
GTL	01	X0000001	Go To Local
LLO	11	X0010001	Local LockOut
MLA		X01 Dev.Adr.	My (the Apple) Listen Address
MTA		X10 Dev.Adr.	My (the Apple) Talk Address
PPC	05	X0000101	Parallel Poll Configure
PPD	70	X1110000	Parallel Poll Disable
PPE	60	X110SPPP	Parallel Poll Enable
PPU	15	X0010101	Parallel Poll Unconfigure
SDC	04	X0000100	Select Device Clear
SPD	19	X0011001	Serial Poll Disable
SPE	18	X0011000	Serial Poll Enable
UNL	3F	X0111111	UNListen
UNT	5F	X1011111	UNTalk

MLA und TLA bestehen aus den 5 niederwertigen Bits der Geräteadresse des Apple. Drei mnemotechnische Codes, LSTN, TALK und XFER, sind nicht vom IEEE-488-Standard abgeleitet.

LSTN – Der Steuerteil wartet darauf, eine Transaktion mit einem Empfänger durchzuführen. Der Apple wartet mit dem Senden neuer Daten (durch Herabsetzen der Spannung auf der DAV-Leitung angezeigt), bis alle Empfänger aufnahmebereit sind (NRFD = 1) und ein Empfänger erkannt hat, daß augenblicklich keine gültigen Daten vorhanden sind (durch Herabsetzen der Spannung auf der NDAC-Leitung).

TALK – Der Steuerteil wartet darauf, eine Transaktion mit einem Sender durchzuführen. Der Apple nimmt keine Daten an (indem er NDAC auf 1 setzt), bis die Daten gültig sind (DAV = 0) und ein Empfänger erkannt hat, daß auf dem Bus gültige Daten vorhanden sind (durch Erniedrigung der Spannung auf der NRFD-Leitung).

XFER – Der adressierte Sender und die Empfänger tauschen eine Quittung über die Datenübertragung auf dem Bus aus. Der Apple ist daran nicht beteiligt.

Das Zeichen <CR> in den Kommandos auf der nächsten Seite wird durch die mit RETURN beschriftete Taste erzeugt (ASCII-Wert 13). Es ist nicht als Buchstabenfolge zu interpretieren. Wie zuvor ist Z\$ eine Variable, der als Inhalt <CTRL-Z> (ASCII-Wert 26) zugewiesen worden ist.

<u>COMMAND</u>	<u>DIO MNEMONIC</u>	<u>DIO (HEX)</u>	<u>MGMNT</u>	<u>HNDSHK</u>
PRINT "WT1";Z\$;"DATA"				
or				
PRINT "WC1";Z\$;"5";Z\$;"DATA<CR>";				
	MTA	40	ATN	LSTN
	UNL	3F	ATN	LSTN
	primary listener	31	ATN	LSTN
	D	44		LSTN
	A	41		LSTN
	T	54		LSTN
	A	41		LSTN
	<CR> [EOS]	0D	EOI	LSTN
	idle	XX	ATN	
PRINT "RDAA";Z\$;				
INPUT C\$				
or				
PRINT "RCAa";Z\$;"5";Z\$;				
INPUT C\$				
	primary talker	41	ATN	LSTN
	secondary talker	61	ATN	LSTN
	UNL	3F	ATN	LSTN
	MLA	20	ATN	LSTN
	D	44		TALK
	A	41		TALK
	T	54		TALK
	A	41		TALK
	<CR>	0D	[EOI]	TALK
	idle	XX	ATN	
PRINT "XFA";Z\$;"19";Z\$				
	primary talker	41	ATN	LSTN
	UNL	3F	ATN	LSTN
	primary listener	31	ATN	LSTN
	primary listener	39	ATN	LSTN
	D	44		XFER
	A	41		XFER
	T	54		XFER
	A	41	EOI	XFER
	idle	XX	ATN	
PRINT "TG1";Z\$				
	UNL	3F	ATN	LSTN
	primary listener	31	ATN	LSTN
	GET	08	ATN	LSTN
	idle	XX	ATN	

<u>COMMAND</u>	<u>DIO MNEMONIC</u>	<u>DIO (HEX)</u>	<u>MGMNT</u>	<u>HNDSHK</u>
PRINT "CA"	DCL idle	14 XX	ATN ATN	LSTN
PRINT "CL1";Z\$	UNL primary listener SDC idle	3F 31 04 XX	ATN ATN ATN ATN	LSTN LSTN LSTN
PRINT "RA"	idle	XX	ATN,REN	
PRINT "RM1";Z\$	UNL primary listener idle	3F 31 XX	ATN ATN,REN ATN,REN	LSTN LSTN
PRINT "LL"	LLO idle	11 XX	ATN ATN	LSTN
PRINT "LA"	idle	XX	ATN,REN	
PRINT "LO1";Z\$	UNL primary listener GTL idle	3F 31 01 XX	ATN ATN ATN ATN	LSTN LSTN LSTN
INPUT "SR";G\$	idle	XX	[SRQ]	
PRINT "SPA";Z\$; INPUT H\$	SPE primary talker UNL MLA status byte SPD idle	18 41 3F 20 (40) 19 XX	SRQ,ATN SRQ,ATN SRQ,ATN SRQ,ATN [SRQ] [SRQ],ATN [SRQ],ATN	LSTN LSTN LSTN LSTN TALK LSTN

<u>COMMAND</u>	<u>DIO MNEMONIC</u>	<u>DIO (HEX)</u>	<u>MGMNT</u>	<u>HNDSHK</u>
INPUT "PP";I\$	pp byte idle	(40) XX	ATN,EOI ATN	
PRINT "PE1A2B"	UNL primary listener PPC PPE + SPPP * primary listener PPC PPE + SPPP * idle	3F 31 05 61 32 05 62 XX	ATN ATN ATN ATN ATN ATN ATN ATN	LSTN LSTN LSTN LSTN LSTN LSTN LSTN
PRINT "PD1";Z\$	UNL primary listener PPC PPD idle	3F 31 05 70 XX	ATN ATN ATN ATN ATN	LSTN LSTN LSTN LSTN
PRINT "PU"	PPU idle	15 XX	ATN ATN	LSTN
PRINT "AB"	idle	XX	IFC	
PRINT "LF1" PRINT "WT1";Z\$;"DATA"	MTA UNL primary listener D A T A <CR> <LF> idle	40 3F 31 44 41 54 41 0D 0A XX	ATN ATN ATN EOI ATN	LSTN LSTN LSTN LSTN LSTN LSTN LSTN LSTN LSTN

* SPPP refers to the Sense bit and the 3 bit assignment bits. The code for PPE is logically ORed with these bits. Refer to the section on the PE command for more details.

ANHANG C

9914-Register

<u>COMMAND</u>	<u>DIO MNEMONIC</u>	<u>DIO (HEX)</u>	<u>MGMNT</u>	<u>HNDSHK</u>
PRINT "ES!"				
PRINT "WT1";Z\$;"DATA!";	MTA	40	ATN	LSTN
	UNL	3F	ATN	LSTN
	primary listener	31	ATN	LSTN
	D	44		LSTN
	A	41		LSTN
	T	54		LSTN
	A	41		LSTN
	!	21	EOI	LSTN
	idle	XX	ATN	
PRINT "SC1"	idle	XX	ATN	
PRINT "DV16"				
PRINT "WT1";Z\$;"DATA"	MTA	50	ATN	LSTN
	UNL	3F	ATN	LSTN
	primary listener	31	ATN	LSTN
	D	44		LSTN
	A	41		LSTN
	T	54		LSTN
	A	41		LSTN
	<CR>	0D	EOI	LSTN
	idle	XX	ATN	
PRINT "UT"	UNT	5F	ATN	LSTN
	idle	XX	ATN	

ANHANG C

9914-Register

Im TMS-9914-Chip der IEEE-488-Karte sind Register enthalten. Wenn sie Informationen direkt zu diesen Registern senden wollen, brauchen Sie die Angaben dieses Anhangs. Einzelheiten über die Handhabung des 9914 gehen über diese Dokumentation hinaus. In der 9914-Beschreibung finden Sie weitere Information zu diesem Thema.

RS2, RS1 und RS0 sind die Register-Adreßleitungen auf dem 9914. Sie sind mit den drei unteren Adreßleitungen auf dem Apple-Bus verbunden.

In den Apple-Adressen bedeutet N die Summe aus Nummer des Einschubfachs mit dem GPIB und der Zahl 8. Nehmen wir an, der GPIB ist im Slot 3 installiert, dann ist die Adresse des Hilfskommandoregisters C0B3 oder C0BB. Weil das Adreßbit 3 nicht entschlüsselt wird, kann man jedes 9914-Register mit zwei verschiedenen Adressen ansprechen.

<u>9914 REGISTER</u>				<u>APPLE ADDRESS</u>	
<u>RS2</u>	<u>RS1</u>	<u>RS0</u>	<u>READ</u>		
0	0	0	INT STATUS 0	C0N0	C0N8
0	0	1	INT STATUS 1	C0N1	C0N9
0	1	0	ADDRESS STATUS	C0N2	C0NA
0	1	1	BUS STATUS	C0N3	C0NB
1	0	0	ADDRESS SWITCH 1	C0N4	C0NC
1	1	0	CMD PASS THROUGH	C0N6	C0NE
1	1	1	DATA IN	C0N7	C0NF
			<u>WRITE</u>		
<u>RS2</u>	<u>RS1</u>	<u>RS0</u>			
0	0	0	INT MASK 0	C0N0	C0N8
0	0	1	INT MASK 1	C0N1	C0N9
0	1	1	AUXILIARY CMD	C0N3	C0NB
1	0	0	ADDRESS REG	C0N4	C0NC
1	0	1	SERIAL POLL	C0N5	C0ND
1	1	0	PARALLEL POLL	C0N6	C0NE
1	1	1	DATA OUT	C0N7	C0NF

Bemerkung: Der 6502 führt vor jeder Schreiboperation einen Lesebefehl aus. Da einige Register auf dem 9914 beim Lesen zurückgesetzt werden, müssen Sie aufpassen, daß auch wirklich die gewünschte Aktion ausgeführt wird. Lesen Sie wegen Details im 6502-Programmierhandbuch und in der 9914-Beschreibung nach.

Im Anhang E finden Sie ein Beispiel zur Verwendung von Registern des 9914.

ANHANG D

Programmierhilfen

- 54 Programm 1: CHR\$-Funktion für Integer-Basic
- 55 Programm 2: Fehlerbehandlungsroutine
- 56 Programm 3: Funktion zum Verhindern der Reinitialisierung
- 57 Programm 4: Eingabe mit Kommas

Dieser Anhang enthält eine Reihe lehrreicher Subroutinen und Programme zum Umgang mit der IEEE-488-Interface-Karte.

Programm 1: *CHR\$-Funktion für Integer-Basic*

Zweck: Mit dieser Funktion können Sie jene Zeichen erzeugen, für die keine Taste auf der Tastatur des Apple vorgesehen sind (siehe die Tabellen A-3 und A-4 im Anhang A). Sie ersetzt in Integer-Basic die in APPLESOFT anwendbare CHR\$-Funktion.

Aufbau: Fügen Sie nachfolgende Zeilen an beliebiger Stelle im Programm ein. Natürlich können Sie die Zeilennummern ändern, aber bitte, ändern Sie sonst nichts. Zeile 120 fügt das gewünschte ASCII-Zeichen zwischen die Anführungszeichen " in der Anweisung CHR\$="A" ein.

Code:

```
110 CHS = CHR +128* (CHR<128)

120 LC1= PEEK (224): LC2= PEEK (
    225)-(LC1>243): POKE 79+LC1-
    256*(LC2>127)+(LC2-255*(LC1>
    127))*256, CHS:CHR$="A":RETURN
```

Anwendung:

- 1) Setzen Sie den ASCII-Code des gewünschten Zeichens in die Variable CHR ein.
- 2) GOSUB 110 (oder die von Ihnen gewählte Zeilennummer)
- 3) Nach CHR\$ wird der gewünschte Wert übertragen.

Fehler: Wenn man Zeile 120 in einem langen Programm auf einem Apple mit mehr als 32-Kilobyte-Speicher benutzt, besteht eventuell die Möglichkeit, daß die Fehlermeldung*** >32767 ERR auftritt. Fügen Sie dann eine REM-Anweisung mit etwa 80 Zeichen an einer höheren Zeilennummer ein. Dadurch wird die CHR\$-Funktion im Speicher weiter nach unten geschoben.

Quelle: Contributed Programs Volumes 3-5 (enthält eine Erklärung)
Apple-Produktnummer #A2L0014

Programm 2: Fehlerbehandlungsroutine

Zweck: Mit der Anweisung ONERR . . . GOTO kann man Bus-Fehler abfangen. Da beim normalen Bus-protokoll kein Busfehler auftreten kann, zeigt sie eine abnormale Situation an. Eine solche kann zum Beispiel vorliegen, wenn der Bus nicht richtig angeschlossen ist.

Code:

```
5 DIM N$(5)
10 D$ = CHR$(4) :REM CTRL-D = DOS COMMAND
20 Z$ = CHR$(26) :REM CTRL-Z = BUS COMMAND
25 PRINT
30 ONERR GOTO 500 :REM ONERR ROUTINE AT 500
40 PRINT D$;"PR#3" :REM ENABLE OUTPUT TO BUS
50 PRINT D$;"IN#3" :REM ENABLE INPUT FROM BUS
55 PRINT "SC1" :REM SET SCREEN TO ON
60 PRINT "WT1"; CHR$(243); Z$;"ONERR GOTO TEST"
61 REM DO A NORMAL WRITE TO LISTENER 1
62 REM ERROR IF BUS NOT CONNECTED
70 PRINT D$;"PR#0" :REM ENABLE OUTPUT TO SCREEN
80 PRINT D$;"IN#0" :REM ENABLE INPUT FROM KEYBOARD
90 END
500 PRINT : PRINT :REM ONERR ROUTINE
510 PRINT "'ONERR GOTO' GOT US HERE"
511 PRINT : PRINT "ERROR FLAG = "; :REM SHOULD BE 49
512 PRINT PEEK(222) : PRINT :REM PRINT ERROR NUMBER
513 PRINT "ERROR OCCURRED AT LINE ";
514 PRINT PEEK(218) + 256 * PEEK(219)
515 REM USE DOS ONERR LOCATIONS
516 REM TO PRINT LINE NUMBER
520 POKE 216,0 :REM RESET ONERR FLAG
530 PRINT : PRINT
540 INPUT "FIX BUS THEN HIT RETURN TO CONTINUE";N$
550 GOTO 25
```

Anwendung: Testen der Fehlerbehandlungsroutine:

- 1) Eingabe des Programms
- 2) Trennen des Busanschlusses
- 3) Starten des Programms
- 4) Den Bus wieder korrekt anschließen und fortfahren

Bemerkung: Eine ausführliche Beschreibung der ONERR-Anweisung finden Sie im DOS-3.2-Handbuch, Apple-Produktnummer #A2L0012.

Programm 3: Funktion zum Verhindern der Reinitialisierung

Zweck: Wenn an die IEEE-488-Karte ein Befehl PR#n bzw. IN#n gegeben wird, erhalten alle Parameter der Karte wieder ihre Standardwerte. Das Unterprogramm, das an Zeile 2000 beginnt, führt PR#3 und IN#3 ohne Neuinitialisierung aus. Es kann an jeden beliebigen Slot angepaßt werden.

Code:

```
100 DIM A$(10) :REM DELETE LINES 100-120
110 DIM C$(10) :REM IF YOU ARE USING
120 DIM G$(10) :REM APPLESOFT.
130 Z$ = "" :REM Z$ CONTAINS CTRL-Z
140 D$ = "" :REM D$ CONTAINS CTRL-D
1000 PRINT D$;"IN#3" :REM ENABLE INPUT FROM SLOT 3
1010 PRINT D$;"PR#3" :REM ENABLE OUTPUT TO SLOT 3
1020 PRINT "SC1" :REM TURN ON THE SCREEN
1030 PRINT "WT123";Z$;"HI" :REM SEND "HI" TO THE GPIB & SCREEN
1050 INPUT "SR";G$ :REM SERVICE REQUESTED?
1060 PRINT D$;"IN#0" :REM ENABLE INPUT FROM KEYBOARD
1070 PRINT D$;"PR#0" :REM ENABLE OUTPUT TO SCREEN ONLY
1080 INPUT "THIS LINE DOES NOT APPEAR ON THE BUS -- PRESS RETURN TO
CONTINUE.";A$
1090 GOSUB 2000 :REM RESTORE INPUTS AND OUTPUTS
1100 PRINT "WC123";Z$;"5";Z$;"-DATA"; :REM SEND "-DATA" TO
:REM GPIB AND SCREEN
1110 PRINT "RDA";Z$; :REM SEND COMMAND, ADRS, CTRL-Z
1120 INPUT C$ :REM AND GET INPUT STRING
1130 PRINT D$;"IN#0" :REM ENABLE INPUT FROM KEYBOARD
1140 PRINT D$;"PR#0" :REM ENABLE OUTPUT TO SCREEN
1200 END
1998 REM ---- THIS SUBROUTINE DOES A PR#3/IN#3 WITHOUT RESETTING
1999 REM ---- THE GPIB DEFAULT VALUES.
2000 POKE 54,6 :REM POKE ADDRESS OF CHARACTER
2010 POKE 55,195 :REM OUTPUT SUBROUTINE.
:REM 195 = 192 + SLOT NUMBER
2020 POKE 56,3 :REM POKE ADDRESS OF CHARACTER
2030 POKE 57,195 :REM INPUT SUBROUTINE.
:REM 195 = 192 + SLOT NUMBER
2040 CALL 1002 :REM CALL DOS INITIALIZATION
2050 RETURN
```

Verwendung:

- 1) Eintippen des Codes
- 2) Löschen Sie Zeile 2040 und die Variable D\$, falls Sie nicht DOS benutzen
- 3) Starten des Programms
- 4) Sie können die Zeilen 2000-2050 als allgemeines Unterprogramm verwenden

Fehler: Das Programm läuft nicht unter DOS 3.1.

Bemerkungen:

- 1) Ein-/Ausgabeadressen im DOS werden im Kapitel über Ein-/Ausgabe im DOS-3.2-(bzw.3.3-)Manual beschrieben.
- 2) Die Zeilen 100-120 sollten nur bei Integer-Basic benutzt werden.

Programm 4: Eingabe mit Kommas

Zweck: APPLESOFT interpretiert ein Komma im Datenstring als Ende der Eingabe. Benutzen Sie diese Routine, wenn Sie mit einem Gerät arbeiten, das Kommas sendet. Die Eingabe wird beim Lesen eines CR beendet. (Beachten Sie, daß mit kleinen Änderungen erreicht werden kann, daß EOS als Endesymbol interpretiert wird.)

Aufbau: Geben Sie die Codezeilen ein.

Code:

```
5 REM    CAN ONLY BE USED IN APPLESOFT
6 REM    INPUT STRING IS STORED IN B$
7 REM    INPUT IS TERMINATED WITH CR
25 A$ = "" : B$ = ""           :REM STRINGS INITIALLY EMPTY
30 GET A$                       :REM READ A CHARACTER AND
40 PRINT A$;                   :REM ECHO IT. SEE NOTE.
50 IF ASC (A$) = 13 THEN RETURN:REM RETURN ON CR
60 B$ = B$ + A$                :REM ADD CHARACTER TO STRING
70 GOTO 30
```

Verwendung:

- 1) Aufruf mit GOSUB 25 (oder einer anderen Nummer, die Sie dieser Zeile gegeben haben)
- 2) Der Eingabestring wird in B\$ übergeben.

Bemerkung:

Die Firmware erwartet, daß Zeichenreihen mit der INPUT-Anweisung vom Bus gelesen werden. Die INPUT-Anweisung zeigt die gelesenen Daten am Bildschirm an. Um dies zu erreichen, muß dem GET-Statement ein PRINT-Statement folgen. So können Sie der Firmware vortäuschen, daß normale Eingabe erfolgt.

Beispiele: Die Routine wird im zweiten Programm im Anhang G benutzt. Dieses Beispiel erklärt ihre Anwendung.

ANHANG E

Routinen für beschleunigte Übertragung

- 60 GPIB-Subroutine DEMO
- 61 Subroutinen
- 61 Allgemeine Steuer-Werte
- 62 Initialisierung des 9914
- 63 Senderoutine
- 64 Empfangsroutine
- 66 Warteroutine (WAIT)
- 66 Datenpuffer
- 67 Unterprogramm TEST
- 68 Symbol-Tabelle
- 68 Datendurchsatz
- 69 Ausführung und Test

Zweck : Diese Routinen zeigen, wie man die beschleunigte Blockübertragung auf dem GPIB bewerkstelligen kann. Sie stehen im Gegensatz zur normalen GPIB-Übertragung direkt mit dem 9914 in Verbindung. Die GPIB-Übertragung sendet einen Datenstring zum DOS, das die Ein-/Ausgaberegister des Apple-Monitors benutzt. Die Daten werden zeichenweise zur IEEE-488-Karte übertragen und, von dort schließlich zum 9914.

Diese Routinen zeigen auch, wie man Unterprogramme in Maschinsprache benutzt, um einen GPIB-Befehl zu implementieren. Dies ist nützlich bei der Verwendung des GPIB im Pascal-System. Eine ausführliche Beschreibung für Pascal-Anwendung folgt im Anhang F.

Hintergrund : Die Routinen wurden mit dem Apple-6502-Assembler/Editor erzeugt. Der Code ist an \$5000 und die Daten sind an \$5200 abgelegt. Sie werden wahrscheinlich diese Adressen an Ihren Anwendungsfall anpassen müssen. Bei der Programmierung wurde davon ausgegangen, daß die Karte im Slot 3 ist und <CR> das EOS-Zeichen ist. (Beachten Sie die Deklarationen des EOS-Zeichens und des Slots an den Stellen \$5200 und \$5202.) Am Ende dieses Anhangs erfahren Sie, wie die Routinen gestartet und getestet werden.

GPIB-Subroutine DEMO

Mit diesem Basic-Testprogramm kann man die Unterprogramme in Maschinsprache aufrufen. Es ersetzt zuerst das BRK an Adresse 21569 (\$5441) durch RTS. BRK wird für Testzwecke benutzt. Dann ruft es die Testroutine an der Adresse \$5400 auf, welche wiederum die anderen Routinen aufruft. Schließlich führt sie einige GPIB-Befehle aus, um zu zeigen, daß die Routinen mit normalen Operationen des GPIB verträglich sind.

```
10 DIM S$(30)           : REM DELETE THIS LINE FOR APPLESOFT
15 POKE 21569,96       : REM PUT AN RTS IN PLACE OF BRK
20 CALL 21504          : REM $5400 IS THE TEST PROGRAM
30 Z$ = ""             : REM Z$ IS CTRL-Z, GPIB COMMAND
40 D$ = ""             : REM D$ IS CTRL-D, DOS COMMAND
50 PRINT D$;"PR#3"     : REM ENABLE OUTPUT TO CARD
60 PRINT D$;"IN#3"    : REM ENABLE INPUT FROM CARD
65 PRINT "SC1"        : REM TURN SCREEN ON
70 PRINT "RDB";Z$;    : REM READ FROM TALKER "B"
75 INPUT S$
80 PRINT D$;"PR#0"    : REM ENABLE OUTPUT TO SCREEN
90 PRINT D$;"IN#0"    : REM ENABLE INPUT FROM KEYBOARD
100 PRINT "S$ = ";S$  : REM ECHO INPUT STRING
110 PRINT D$;"PR#3"   : REM ENABLE OUTPUT TO CARD
120 PRINT D$;"IN#3"   : REM ENABLE INPUT FROM CARD
125 PRINT "SC1"       : REM TURN SCREEN ON
130 PRINT "WT1";Z$;"FIRMWARE" : REM SEND "FIRMWARE" TO "1"
140 PRINT D$;"PR#0"   : REM ENABLE OUTPUT TO SCREEN
150 PRINT D$;"IN#0"   : REM ENABLE INPUT FROM KEYBOARD
200 END
```

Subroutinen

Die folgenden Routinen haben willkürliche Startadressen. Passen Sie diese der Konfiguration Ihres Systems an, ebenso die Sieben - Bytes - Speicher. Die Vergabe der Adressen für die Datenpuffer und Gerätelisten liegt bei Ihnen.

Allgemeine Steuer-Werte

The general control values assume that the Apple is device #0.

```
C080: 19 GPIB EQU $C080 ;SET BASE ADDRESS OF ROM
0000: 20 ; OFFSET FOR SLOT 3 ASSUMED
C080: 21 INT0 EQU GPIB+0 ;INTERRUPT REG. 0
C080: 22 INTM0 EQU GPIB+0 ;INTRP MASK REG 0
0020: 23 BIM EQU $20 ;BYTE IN MASK
0010: 24 BOM EQU $10 ;BYTE OUT MASK
0008: 25 EOIMK EQU $08 ;EOI MASK
0000: 26 ;
C081: 27 INTM1 EQU GPIB+1 ; INTRP MASK REG 1
0000: 28 ;
C083: 29 AUXCMD EQU GPIB+3 ;AUXILIARY COMMAND REGISTER
0080: 30 RESET EQU $80 ;SOFTWARE CHIP RESET
0000: 31 RSTCLR EQU $00 ;STOP SOFTWARE RESET
0083: 32 HDFA EQU $83 ;HOLDOFF ON ALL DATA
0003: 33 HDACLR EQU $03 ;CLEAR HOLDOFF ON ALL
0089: 34 LON EQU $89 ;LISTEN ONLY
008A: 35 TON EQU $8A ;TALK ONLY
008F: 36 SIC EQU $8F ;SEND INTERFACE CLEAR
000F: 37 SICLR EQU $0F ;CLEAR SIC
0002: 38 RHDF EQU $02 ;RELEASE RFD HOLDOFF
0008: 39 FEOI EQU $08 ;SEND EOI WITH NEXT BYTE
000B: 40 GTS EQU $0B ;GO TO STANDBY
000C: 41 TCA EQU $0C ;TAKE CONTROL ASYNCHRONOUSLY
000D: 42 TCS EQU $0D ;TAKE CONTROL SYNCHRONOUSLY
0000: 43 ;
C087: 44 DIN EQU GPIB+7 ;DATA IN REGISTER
C087: 45 DOUT EQU GPIB+7 ;DATA OUT REGISTER
0000: 46 ;
0000: 47 ; GPIB COMMANDS
0000: 48 ;
0020: 49 MLA EQU $20 ;MY LISTEN ADDRESS
0040: 50 MTA EQU $40 ;MY TALK ADDRESS
003F: 51 UNL EQU $3F ;UNIVERSAL UNLISTEN
```

Initialisierung des 9914

Alle Routinen setzen beim Eingang denselben Zustand des 9914-Chips voraus. Ebenso erzeugen sie alle denselben Zustand des 9914 beim Ausgang. Die INIT-Routine versetzt den 9914 in den gewünschten Anfangszustand. Dieser ist durch folgende Annahmen charakterisiert:

- BO gesetzt
- Alle Unterbrechungen ausgeblendet
- TON (talk only) eingeschaltet
- kein Empfänger aktiv (LA)
- kein Haltezustand angemeldet oder aktiv
- kein 6502-Register zerstört
- Polling des 9914 durchgeführt

SUBROUTINES CALLED: WAIT

```

5000:          101          ORG   $5000
5000:          102  ;
5000:08        103  INIT:   PHP           ;SAVE REGISTERS
5001:48        104          PHA
5002:98        105          TYA
5003:48        106          PHA
5004:8A        107          TXA
5005:48        108          PHA
5006:AC 02 52  109          LDY   SLOT       ;GPIB SLOT INDICATOR
5009:A9 80     110          LDA   #RESET     ;RESET 9914
500B:99 83 C0  111          STA   AUXCMD,Y
500E:A9 00     112          LDA   #RSTCLR   ;TURN OFF RESET
5010:99 80 C0  113          STA   INTM0,Y    ;CLEAR INTERRUPTS
5013:99 81 C0  114          STA   INTM1,Y    ;CLEAR INTERRUPTS
5016:99 83 C0  115          STA   AUXCMD,Y
5019:A9 8F     116          LDA   #SIC      ;SEND INTERFACE CLEAR
501B:99 83 C0  117          STA   AUXCMD,Y  ;AND TAKE CONTROL
501E:A2 04     118          LDX   #4        ;5.12 MSEC DELAY
5020:A0 00     119          LDY   #0        ;INITIALIZE INNER LOOP
5022:88        120  INIT1:  DEY           ;5 USEC INNER LOOP
5023:D0 FD     121          BNE   INIT1     ;1280 USEC TOTAL
5025:CA        122          DEX
5026:D0 FA     123          BNE   INIT1
5028:AC 02 52  124          LDY   SLOT
502B:A9 0F     125          LDA   #SICLR   ;CLEAR INTERFACE CLR
502D:99 83 C0  126          STA   AUXCMD,Y
5030:A9 8A     127          LDA   #TON      ;SET TALK ONLY MODE
5032:99 83 C0  128          STA   AUXCMD,Y
5035:20 32 51  129          JSR   WAIT      ;WAIT FOR BYTE OUT
5038:          130  ;          TO INDICATE CONTROL TAKEN
5038:68        131          PLA           ;RESTORE REGISTERS
5039:AA        132          TAX
503A:68        133          PLA
503B:A8        134          TAY
503C:68        135          PLA
503D:28        136          PLP
503E:60        137          RTS           ;RETURN TO CALLING POINT

```

Senderoutine

Die Senderoutine kann das WT- oder WC-Kommando ersetzen. Sie beendet sich, wenn so viele Bytes gesendet wurden, wie die Zählvariable COUNT angibt, oder wenn EOS übertragen wurde. Sie fügt in beiden Fällen ein EOI-Zeichen an.

Eingabe: ABUF = Zeiger zur Empfängerliste
 DBUF = Zeiger zum Datenpuffer
 COUNT = Anzahl zu sendender Bytes; Ø bedeutet keine Daten
 EOS = Endezeichen

Ausgabe: keine Ausgabe an die Software
 aufgerufene Unterprogramme: WAIT
 gelöschte Register: keine

503F:08	157	SEND:	PHP	SAVE REGISTERS
5040:48	158		PHA	
5041:98	159		TYA	
5042:48	160		PHA	
5043:8A	161		TXA	
5044:48	162		PHA	
5045:AC 02 52	163		LDY SLOT	;GPIB SLOT INDICATOR
5048:A9 40	164		LDA #MTA	;MY TALK ADDRESS
504A:99 87 C0	165		STA DOUT,Y	
504D:20 32 51	166		JSR WAIT	;WAIT FOR BYTE OUT
5050:A9 3F	167		LDA #UNL	;UNIVERSAL UNLISTEN
5052:99 87 C0	168		STA DOUT,Y	
5055:A2 00	169		LDX #0	;BUFFER INDEX
5057:BD 03 52	170	SEND1:	LDA ABUF,X	;GET LISTENER ADDRESS
505A:C9 20	171		CMP #\$20	;LOWER LIMIT OF RANGE
505C:30 10	172		BMI SEND2	
505E:C9 3F	173		CMP #\$3F	;UPPER LIMIT
5060:10 0C	174		BPL SEND2	
5062:20 32 51	175		JSR WAIT	;WAIT FOR UNL BYTE OUT
5065:BD 03 52	176		LDA ABUF,X	;GET LISTENER AGAIN
5068:99 87 C0	177		STA DOUT,Y	;OUTPUT IT TO GPIB
506B:E8	178		INX	
506C:10 E9	179		BPL SEND1	;ALWAYS TAKEN
506E:	180			(MAX OF 31 LISTENERS!)
506E:A9 0B	181	SEND2:	LDA #GTS	;GO TO STANDBY
5070:99 83 C0	182		STA AUXCMD,Y	
5073:20 32 51	183		JSR WAIT	;WAIT FOR ATN TO BE
5076:	184			DE-ASSERTED
5076:A2 00	185		LDX #0	;INITIALIZE DATA BUF INDEX
5078:EC 01 52	186		CPX COUNT	;IS INITIAL COUNT =0?
507B:F0 28	187		BEQ SEND6	;YES--GO QUIT
507D:CE 01 52	188	SEND3:	DEC COUNT	
5080:F0 11	189		BEQ SEND5	;IF COUNT FINISHED GO SEND EOI
5082:	190			WITH LAST CHARACTER
5082:BD 05 52	191		LDA DBUF,X	;GET DATA BYTE
5085:CD 00 52	192		CMP EOS	;IS IT LAST CHARACTER?
5088:F0 09	193		BEQ SEND5	;YES--GO HANDLE IT
508A:99 87 C0	194		STA DOUT,Y	;OUTPUT DATA TO GPIB
508D:20 32 51	195	SEND4:	JSR WAIT	;WAIT FOR DATA BYTE OUT

5090:E8	196	INX		
5091:D0 EA	197	BNE SEND3		;ALWAYS TAKEN
5093:A9 08	198 SEND5:	LDA #FE01		;FORCE EOI W/ EOS
5095:99 83 C0	199	STA AUXCMD,Y		
5098:BD 05 52	200	LDA DBUF,X		;GET BYTE AGAIN
509B:99 87 C0	201	STA DOUT,Y		;OUTPUT TO GPIB
509E:E8	202	INX		
509F:CE 01 52	203	DEC COUNT		;FOR CONSISTENCY
50A2:20 32 51	204	JSR WAIT		;WAIT FOR LAST BYTE OUT
50A5:A9 0C	205 SEND6:	LDA #TCA		;TAKE CONTROL
50A7:99 83 C0	206	STA AUXCMD,Y		
50AA:20 32 51	207	JSR WAIT		;WAIT FOR CONTROL
50AD:68	208	PLA		;RESTORE REGISTERS
50AE:AA	209	TAX		
50AF:68	210	PLA		
50B0:A8	211	TAY		
50B1:68	212	PLA		
50B2:28	213	PLP		
50B3:60	214	RTS		;RETURN TO CALL POINT

Empfangsroutine

Die Empfangsroutine kann das RD- bzw. RC-Kommando ersetzen. Sie beendet sich, wenn entweder so viele Bytes, wie COUNT angibt, empfangen wurden oder EOS gelesen wurde oder EOI gesetzt wird.

Eingaben: ABUF = Zeiger zur Senderliste
 DBUF = Zeiger zum Datenpuffer
 COUNT = Anzahl der zu empfangenden Bytes; 0 bedeutet keine Daten
 EOS = Endezeichen

Ausgaben: der Datenpuffer ist gefüllt
 aufgerufene Unterprogramme: WAIT
 gelöschte Register: keine

50B4:08	235 RECV:	PHP		;SAVE REGISTERS
50B5:48	236	PHA		
50B6:98	237	TYA		
50B7:48	238	PHA		
50B8:8A	239	TXA		
50B9:48	240	PHA		
50BA:AC 02 52	241	LDY* SLOT		;GPIB SLOT IDENTIFIER
50BD:A2 00	242	LDX #0		
50BF:AD 03 52	243	LDA ABUF		;GET TALKER DEVICE NO.
50C2:99 87 C0	244	STA DOUT,Y		;OUTPUT IT TO GPIB
50C5:20 32 51	245	JSR WAIT		;WAIT FOR BYTE OUT
50C8:A9 3F	246	LDA #UNL		;UNIVERSAL UNLISTEN
50CA:99 87 C0	247	STA DOUT,Y		;OUTPUT IT TO GPIB
50CD:20 32 51	248	JSR WAIT		;WAIT FOR BYTE OUT
50D0:A9 20	249	LDA #MLA		;MY LISTEN ADDRESS
50D2:99 87 C0	250	STA DOUT,Y		;OUTPUT IT TO GPIB
50D5:A9 83	251	LDA #H DFA		;ENABLE HOLDOFF ON ALL
50D7:99 83 C0	252	STA AUXCMD,Y		
50DA:A9 89	253	LDA #LON		;LISTEN ONLY-CLEAR TON
50DC:99 83 C0	254	STA AUXCMD,Y		

50DF1A9 0B	255	LDA #GTS	;GO TO STANDBY
50E1199 83 C0	256	STA AUXCMD,Y	
50E4120 32 51	257	JSR WAIT	;WAIT FOR MLA BYTE OUT
50E71B9 80 C0	258	RECV1: LDA INT0,Y	;WAIT FOR BYTE IN
50EA129 28	259	AND #EOIMK+BIM	;TEST FOR EOI OR BI
50EC1F0 F9	260	BEQ RECV1	
50EE129 08	261	AND #EOIMK	;IS IT EOI?
50F01D0 18	262	BNE RECV2	;YES
50F21B9 87 C0	263	LDA DIN,Y	;GET DATA BYTE FROM GPIB
50F519D 05 52	264	STA DBUF,X	;STORE IT IN BUFFER
50FB1CD 00 52	265	CMP EOS	;IS IT THE EOS CHAR?
50FB1F0 14	266	BEQ RECV3	;YES
50FD1EB	267	INX	
50FE1A9 02	268	LDA #RHDF	;RELEASE HOLDOFF
5100199 83 C0	269	STA AUXCMD,Y	
51031CE 01 52	270	DEC COUNT	
51061F0 0C	271	BEO RECV4	;GO IF COUNT =0
51081D0 DD	272	BNE RECV1	;ALWAYS TAKEN--GO FOR MORE
510A1B9 87 C0	273	RECV2: LDA DIN,Y	;GET LAST BYTE
510D19D 05 52	274	STA DBUF,X	
51101EB	275	INX	
51111CE 01 52	276	RECV3: DEC COUNT	
51141A9 0D	277	RECV4: LDA #TCS	;TAKE CONTROL SYNC
5116199 83 C0	278	STA AUXCMD,Y	
5119120 32 51	279	JSR WAIT	;WAIT FOR CONTROL
511C1A9 02	280	LDA #RHDF	;RELEASE HOLDOFF
511E199 83 C0	281	STA AUXCMD,Y	
51211A9 8A	282	LDA #TON	;SET TALK ONLY, CLR LON
5123199 83 C0	283	STA AUXCMD,Y	
51261A9 03	284	LDA #HDACLR	;CLEAR HOLDOFF MODE
5128199 83 C0	285	STA AUXCMD,Y	
512B168	286	PLA	;RESTORE REGISTERS
512C1AA	287	TAX	
512D168	288	PLA	
512E1A8	289	TAY	
512F168	290	PLA	
5130128	291	PLP	
5131160	292	RTS	;RETURN TO CALL POINT

Warteroutine (WAIT)

Die Warteroutine liest das INTO-Register, bis das Byteausgabe- oder das Byteeingabe-Bit gesetzt ist.

Eingaben: Register Y enthält die Slot-Nr.

Ausgaben: keine

aufgerufene Unterprogramme: keine

zerstörte Register: keine

```
5132:B9 80 C0 310 WAIT: LDA INT0,Y ;GET BYTE OUT FLAG
5135:29 10 311 AND #BOM
5137:F0 F9 312 BEQ WAIT ;LOOP UNTIL SET
5139:60 313 RTS
```

Datenpuffer

Die folgenden Datenpuffer arbeiten mit der angegebenen Testroutine. Sie müssen sie an Ihre jeweilige Aufgabe anpassen. Beachten Sie, daß folgende Standardwerke angenommen wurden: <CR> als EOS-Zeichen, Wert des Bytezählers 255 und 3 als Nummer des Slots ABUF, die Empfänger- bzw. Senderliste besteht nur aus einem Element. Ihr Inhalt ist entsprechend der jeweiligen Aufgabe festzulegen. Diese Liste kann mit einem beliebigen Nicht-Empfänger bzw. Nicht-Sender je nach verwendeter Routine enden.

```
5200: 325 ORG $5200
5200: 326 ;
5200:0D 327 EOS DFB $0D ;EOS CHARACTER--DEFAULT IS CR
5201:FF 328 COUNT DFB $FF ;BYTE COUNT FOR DATA BUFFER
5202: 329 ; --DEFAULT IS 255
5202:30 330 SLOT DFB $30 ;OFFSET TO SLOT 3 ADDRESSES
5203: 331 ; --DEFAULT IS SLOT #3
5203: 332 ABUF DS 2 ;DEVICE LIST
5205: 333 ; --MUST END WITH $FF BYTE
5205: 334 DBUF DS 256 ;DATA BUFFER
```

Unterprogramm TEST

Das Unterprogramm testet die bisher besprochenen Subroutinen. Es wird von dem am Anfang des Abschnitts vorgestellten GPIB-Unterprogramm DEMO aufgerufen. Es sendet Daten zum Empfänger \$21 (ASCII-Zeichen "!") und empfängt Daten vom Sender \$41 (ASCII-Zeichen "A"). Der Datenpuffer (DBUF) wird gefüllt mit den Zahlen 1 bis 5, gefolgt von CR. Da beim ersten Senden COUNT den Anfangswert hat (255), beendet CR die Ausgabe. Beim nächsten Senden hat COUNT den Wert 5. Deshalb wird CR nicht mitgeschickt. Das BRK in Zeile 376 wurde durch einen RTS-Befehl ersetzt. Dadurch wird die RECV-Routine nur einmal aufgerufen und die Schleife nicht durchlaufen.

Bemerkung: Beim Code wird angenommen, daß <CR> das <EOS>-Zeichen ist. Falls Sie EOS ändern, schickt der erste SEND-Aufruf 255 Bytes.

```

5400:          347      ORG   $5400
5400:          348 ;
5400:20 00 50 349      JSR   INIT       ;SET UP GPIB INITIALLY
5403:EA          350      NOP
5404:A9 21      351      LDA   #$21       ;LISTENER ADDRESS
5406:8D 03 52 352      STA   ABUF
5409:A9 FF      353      LDA   #$FF       ;LIST END
540B:8D 04 52 354      STA   ABUF+1
540E:A9 01      355      LDA   #1
5410:8D 05 52 356      STA   DBUF       ;DATA TO BE SENT
5413:A9 02      357      LDA   #2
5415:8D 06 52 358      STA   DBUF+1
5418:A9 03      359      LDA   #3
541A:8D 07 52 360      STA   DBUF+2
541D:A9 04      361      LDA   #4
541F:8D 08 52 362      STA   DBUF+3
5422:A9 05      363      LDA   #5
5424:8D 09 52 364      STA   DBUF+4
5427:A9 0D      365      LDA   #$0D       ;6 BYTES OF DATA INCL CR
5429:8D 0A 52 366      STA   DBUF+5
542C:20 3F 50 367      JSR   SEND
542F:EA          368      NOP
5430:A9 05      369      LDA   #5       ;5 BYTES OF DATA, NO CR
5432:8D 01 52 370      STA   COUNT
5435:20 3F 50 371      JSR   SEND
5438:EA          372      NOP
5439:A9 41      373      LDA   #$41       ;TALKER ADDRESS
543B:8D 03 52 374      STA   ABUF
543E:20 B4 50 375 LOOP: JSR   RECV
5441:00          376      BRK
5442:4C 3E 54 377      JMP   LOOP
5445:          378 ;
5445:          379 ;END

```

Symbol-Tabelle

5203	ABUF	C083	AUXCMD	20	BIM	10	BOM
5201	COUNT	5205	DBUF	C087	DIN	C087	DOUT
08	EOIMK	5200	EOS	08	FEOI	C080	GPIB
0B	GTS	03	HDACLR	83	HDF A	5000	INIT
5022	INIT1	C080	INT0	C080	INTM0	C081	INTM1
89	LON	543E	LOOP	20	MLA	40	MTA
510A	RECV2	50B4	RECV	50E7	RECV1	5111	RECV3
5114	RECV4	80	RESET	02	RHDF	00	RSTCLR
5057	SEND1	506E	SEND2	503F	SEND	507D	SEND3
?508D	SEND4	5093	SEND5	50A5	SEND6	0F	SICLR
8F	SIC	5202	SLOT	0C	TCA	0D	TCS
8A	TON	3F	UNL	5132	WAIT		

Datendurchsatz

Beim Schreiben überträgt der GPIB 1,5 Kilobytes pro Sekunde. Das schließt nicht die ungefähr 3,5 Millisekunden Befehlsausführungszeit ein. Bei einer Leseoperation beträgt die Datenrate etwa 1 Kilobyte pro Sekunde.

Die Leistungsfähigkeit des IEEE-488-Busses kann mit den vorher beschriebenen Routinen gesteigert werden. Sie senden und empfangen Daten mit einer Rate von 20 Kilobytes in der Sekunde. Auch das ist nicht optimal. Wenn man die obigen Routinen so modifiziert, daß sie Daten nur mit dem Zählerfeld übertragen, gelangt man zu einer Rate von 50 Kilobytes pro Sekunde.

Ausführung und Test

Unsere Version des APPLESOFT-Programms zum Ausführen der Subroutinen heißt GPIB SUBR DEMO. Der Quellcode des Assemblerprogramms wurde in der Datei GPIB SUBR gespeichert. Der 6502-Assembler bringt den übersetzten Code in die Dateien:

GPIB SUBR.OBJØ (Routinen)
GPIB SUBR.OBJ1 (Daten)
GPIB SUBR.OBJ2 (Testcode)

Die Routinen können mit der Kommandofolge

```
| BLOAD GPIB SUBR.OBJØ  
| BLOAD GPIB SUBR.OBJ1  
| BLOAD GPIB SUBR.OBJ2  
| RUN GPIB SUBR DEMO
```

getestet werden.

Wir empfehlen, die Funktion der Programme mit dem ZT-488-GPIB-Analysator auf Korrektheit zu prüfen. Die Folge der Bussignale sollte lauten:

<u>DIO</u>	<u>MGMNT</u>								
4Ø	ATN	4Ø	ATN	41	ATN	42	ATN	4Ø	ATN
3F	ATN								
21	ATN	21	ATN	2Ø	ATN	2Ø	ATN	31	ATN
1		1		INPUT		INPUT		46	
2		2		any		any		49	
3		3		sequence		sequence		52	
4		4		terminated		terminated		4D	
5		5	EOI	with		with		57	
ØD	EOI			ØD or EOI		ØD or EOI		41	
								52	
								45	
								ØD	EOI

ANHANG F

Programmieren in Pascal

Da die GPIB-Firmware für die Interaktion mit Basic, dem Monitor und DOS entwickelt wurde, arbeitet es bei den Pascalanweisungen WRITE und WRITELN nicht richtig. Das heißt jedoch nicht, daß der Apple-GPIB im Pascalsystem nicht verwendet werden kann.

Sie können die meisten GPIB-Kommandos im Pascalsystem anwenden. Dazu müssen Sie allerdings mit dem 6502-Assembler und dem Pascalbetriebssystem von Apple vertraut sein.

Was zu tun ist

Sie müssen für jeden zu implementierenden GPIB-Befehl eine Pascalprozedur schreiben. Einige davon – zum Beispiel SC – können vollständig in Pascal geschrieben werden. Für andere wiederum ist zumindest teilweise der 6502-Assembler zu verwenden. Im 6502-Code gehen die Kommandos direkt an den 9914-Chip.

Vor dem ersten Versuch, Assemblerrouinen zu schreiben, müssen Sie die Kommandos des 9914 kennen. Aus der TMS-9914-Spezifikation von Texas Instrument ersehen Sie, welche Werte zu welchen Registern des 9914 zu senden sind, um die gewünschten GPIB-Befehle zu implementieren.

Die Apple-Adressen der Register des 9914 finden Sie im Anhang C. Mit diesen Adressen müssen Ihre Assembler-Routinen arbeiten, und zwar genau so, wie die Sende-, Empfangs- und Initialisierungsroutine aus Anhang E. (Achten Sie auf die Adressen INT0, INTM0, INTM1, DIN und DOUT.) Sie können auch leicht abgeänderte Versionen dieser drei Routinen als Anfang Ihres Pascal-GPIB-Pakets benutzen.

Es fehlt noch eine 6502-Prozedur zum Ausführen der implementierten Funktionen. Diese Prozedur ist ins Pascalsystem einzubinden.

Angaben zu den 6502-Assembleranweisungen finden Sie in der Apple-Pascalbetriebssystem-Beschreibung im Kapitel über den 6502-Assembler (Appleprodukt #A2L0028). Das Kapitel erklärt den Aufbau einer Prozedur, die ins Pascalsystem gebunden wird.

Das Kapitel über den Binder im selben Handbuch beschreibt den Ablauf des Einbindens einer Prozedur in das Pascalsystem. Schließlich erfahren Sie aus der Apple-Pascal-Sprachbeschreibung (Apple-Produkt #A2L0027), wie externe Prozeduren (alle Assemblerunterprogramme sind extern zum umgebenden Programm) zu deklarieren und vom umgebenden Programm aufzurufen sind.



Da die Assemblerrouinen mit Adressen in Verbindung stehen, die außerhalb des Apple liegen, ist es ratsam, einen Logikprüfer wie den ZT488-GPIB-A-ANALYZER zum Überwachen der Bussignale zu benutzen. Das ist die einzige Möglichkeit zur Überprüfung der Busoperationen.

ANHANG G

Programmbeispiele

- 74 Leseroutine für den HP-3438A
- 75 Allgemeine HP 3586C Routine

Dieser Anhang enthält zwei Programmbeispiele für die IEEE-488-Karte. Sie können diese Programme nur verwenden, wenn Sie ein HP-3438A-Multimeter oder ein HP-3586C-Selective-Level-Meter besitzen. Trotzdem veranschaulichen Sie hervorragend die Funktion der Programme für den GPIB.

Leseroutine für den HP-3438A

Das HP-3438A-Multimeter hat die Gerätenummer 3. Seine Empfängeradresse ist "#" und seine Sendeadresse ist "C". Beim vorliegenden Programm ist angenommen, daß die IEEE-488-Karte sich im Slot 3 befindet. Der 3438A sendet Daten der Form SD.DDDESDF<CR><LF>.
Es bedeuten

S das Vorzeichen, entweder "+" oder "-"
D eine Dezimalziffer zwischen 0 und 9
E ein "E" als Zeichen für den Exponenten
, das Trennzeichen zwischen dem Wert und dem Funktionscode
F den Funktionscode, eine Ziffer zwischen "1" und "5"
<CR><LF> Wagenrücklauf, gefolgt von Zeilenvorschub

```

10 DIM S$(30): REM ----- DELETE FOR APPLESOFT
20 DIM C$(30): REM ----- DELETE FOR APPLESOFT
30 Z$ = "": REM CTL-Z
40 D$ = "": REM CTL-D
50 PRINT D$;"PR#3"
60 PRINT D$;"IN#3"
70 PRINT "SC1": REM ----- TURN THE SCREEN ON
80 PRINT "TG#";Z$: REM --- TRIGGER A READING
90 PRINT "LF1": REM ----- 3438A SENDS A <CR><LF>
100 PRINT "RDC";Z$;: REM - READ FROM TALKER "C"
110 INPUT S$,C$: REM ---- S$ GETS CHARACTERS BEFORE
120 REM ----- THE ", " SENT BY THE 3438A
130 REM ----- C$ GETS CHARACTERS AFTER THE ", "
140 PRINT D$;"PR#0": REM - SEND OUTPUT TO SCREEN
150 PRINT D$;"IN#0": REM - GET INPUT FROM KEYBOARD
160 PRINT "READING = ";S$;
170 REM ----- NEXT LINE COMPUTES VALUE OF FUNCTION
180 REM ----- CODE RETURNED BY THE 3438A
190 ON ASC ( LEFT$( C$,1)) - 48 GOTO 200,210,220,230,240
200 PRINT " DCV": GOTO 250
210 PRINT " ACV": GOTO 250
220 PRINT " DCI": GOTO 250
230 PRINT " ACI": GOTO 250
240 PRINT " OHMS": GOTO 250
250 INPUT "WOULD YOU LIKE ANOTHER READING(Y OR N) ?";A$
260 IF A$ = "Y" THEN GOTO 50
270 END

```

Allgemeine HP 3586C Routine

```
10 REM ROUTINE FOR HP3586C SELECTIVE LEVEL METER
20 REM METER IS DEVICE 16
30 REM LISTEN ADDRESS IS ASCII 30 = "0"
40 REM TALK ADDRESS IS ASCII 50 = "P"
50 REM ASSUMES CARD IN SLOT #3
60 D0 = "": REM CONTROL-D
70 Z0 = "": REM CONTROL-Z
80 DIM C$(100): REM DELETE FOR APPLESOFT
90 GOSUB 590: REM INITIALIZE CARD FOR I/O
100 GOSUB 660: REM INITIALIZE APPLE FOR I/O
110 HOME
120 VTAB 5: HTAB 18: PRINT "MENU": PRINT : PRINT
130 HTAB 8: PRINT "1 : PRINT TO THE DEVICE"
140 HTAB 8: PRINT "2 : LISTEN TO DEVICE"
150 HTAB 8: PRINT "3 : CLEAR DEVICE"
160 HTAB 8: PRINT "4 : REMOTE ENABLE DEVICE"
170 HTAB 8: PRINT "5 : LOCAL LOCKOUT DEVICE"
180 HTAB 8: PRINT "6 : SET DEVICE TO LOCAL MODE"
190 HTAB 8: PRINT "7 : ABORT SEQUENCE"
200 HTAB 8: PRINT "8 : SEND LITERAL STRING"
210 HTAB 8: PRINT "9 : QUIT": PRINT : PRINT
220 HTAB 8: PRINT "WHAT IS YOUR COMMAND?"
230 GET A
240 ON A GOTO 300,350
250 GOSUB 690
260 ON A - 2 GOTO 480,490,500,510,520
270 GOSUB 660
280 ON A - 7 GOTO 530,580
290 GOTO 110: REM NO SUCH COMMAND
300 PRINT "ENTER COMMAND AND DATA": REM PRINT TO THE DEVICE
310 INPUT C$
320 GOSUB 690: REM INITIALIZE CARD FOR I/O
330 PRINT "WT0";Z$;C$: REM SEND COMMAND TO METER
340 GOTO 100: REM GET NEXT COMMAND
350 PRINT "INCOMING DATA IS..."
360 GOSUB 690: REM INITIALIZE CARD FOR I/O
370 PRINT "TG0";Z$: REM DEVICE MUST BE TRIGGERED BEFORE READING
380 PRINT "RD0";Z$;: REM READ FROM DEVICE
390 C0 = "": REM SET C$ TO EMPTY
400 GET A$: PRINT A$;: REM GET CHARACTER.
410 REM --- SEE NOTE FOR REINITIALIZE FUNCTION.
420 IF ASC (A$) = 13 THEN 440: REM REPEAT UNTIL <CR>
430 C0 = C0 + A$: GOTO 400: REM SAVE CHARACTERS IN C$
440 GOSUB 660: REM INITIALIZE APPLE FOR I/O
```

```

450 PRINT C$: REM PRINT INPUT STRING
460 PRINT "HIT ANY KEY TO CONTINUE..."
470 GET A$: GOTO 110: REM GET NEW COMMAND
480 PRINT "CA": GOTO 100: REM CLEAR ALL REMOTE DEVICES
490 PRINT "RA": GOTO 100: REM ENABLE DEVICES FOR REMOTE OPERATION
500 PRINT "LL": GOTO 100: REM LOCAL LOCKOUT DEVICE
510 PRINT "LA": GOTO 100: REM LOCKOUT DEVICE'S FRONT PANEL
520 PRINT "AB": GOTO 100: REM SET ALL DEVICES TO LOCAL MODE
530 PRINT "ENTER THE LINE TO BE TRANSMITTED:"
540 INPUT C$: REM READ LINE
550 GOSUB 690: REM INITIALIZE CARD FOR I/O
560 PRINT C$: REM TRANSMIT LINE
570 GOTO 100
580 END
590 PRINT D$;"PR#3": REM SEND OUTPUT TO IEEE-488 CARD
600 PRINT D$;"IN#3": REM GET INPUT FROM IEEE-488 CARD
610 PRINT "SC1": REM SET THE SCREEN TO ON
620 PRINT "RA": REM ENABLE ALL DEVICES FOR REMOTE OPERATION
630 PRINT "DV1": REM SET THE APPLE TO DEVICE 1
640 PRINT "LF1": REM SEND AND RECEIVE <LF> AFTER <CR>
650 RETURN
660 PRINT D$;"PR#0": REM SEND OUTPUT TO THE SCREEN
670 PRINT D$;"IN#0": REM GET INPUT FROM THE KEYBOARD
680 RETURN
690 PRINT
700 PRINT D$;"PR#3": REM SEND OUTPUT TO IEEE-488 CARD
710 PRINT D$;"IN#3": REM GET INPUT FROM IEEE-488 CARD
720 REM -AT THIS POINT ALL DEFAULT COMMANDS MUST BE RESET
730 REM -FOR EXAMPLE: LF1,SC1,DV1,ETC
740 RETURN

```

Bemerkung: Die Zeilen 410-450 enthalten die Routine "Eingabe mit Kommas". Im Ahang D finden Sie nähere Erläuterungen zu diesem Code.

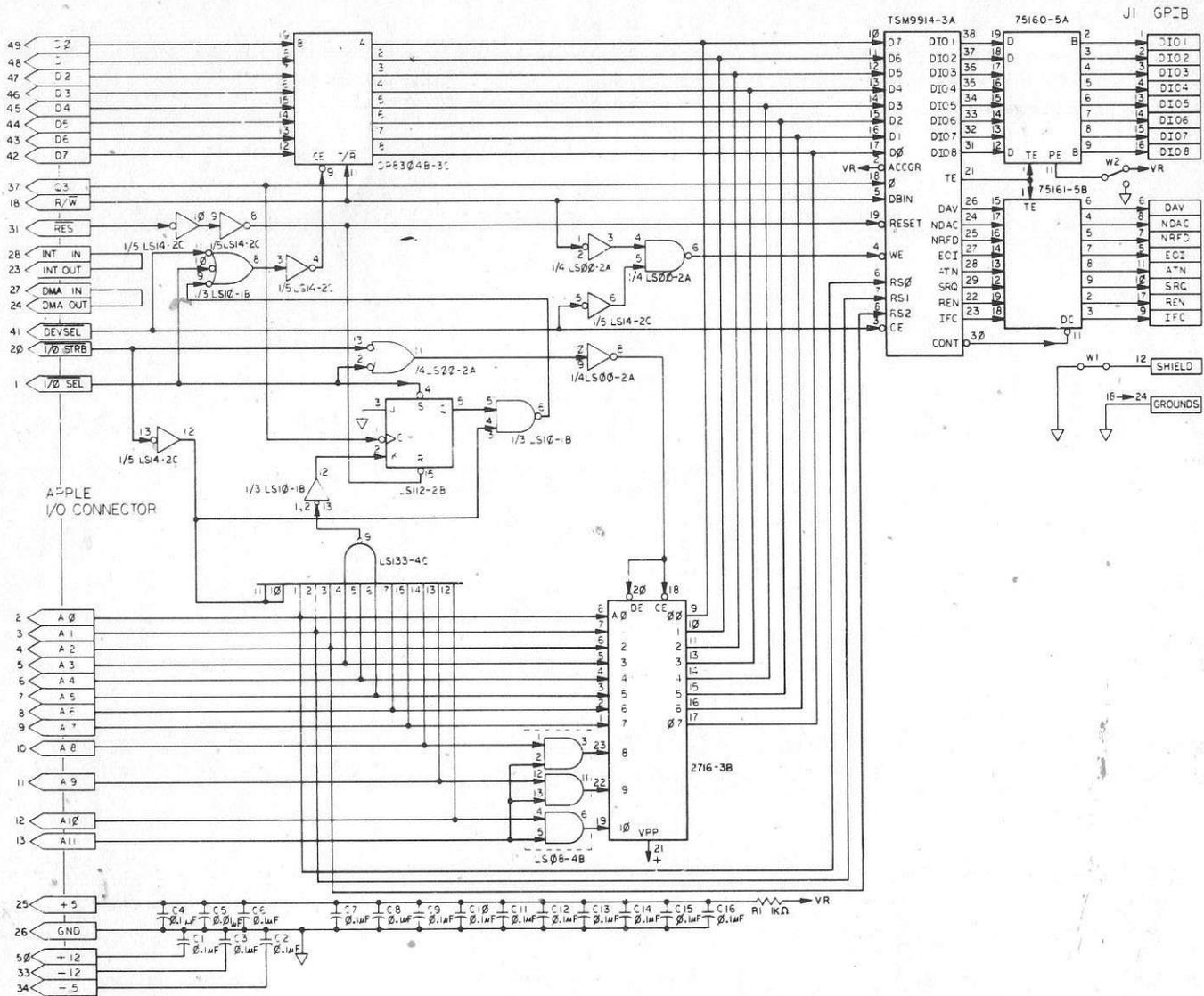
ANHANG H

Schematisches Diagramm

Glossar

Stichwortverzeichnis

Schematisches Diagramm



Physikalische und elektrische Spezifikationen

Die Schaltungsanordnung des Apple-II-IEEE-488 ist auf einer vollständigen Apple-Peripherie-Karte mit dem Maßen 2.75" auf 7" montiert. Die Steckerleiste hat 50 Kontakte, 25 auf jeder Seite des Bausteins. Die Kontakte haben Mittenabstand von 0.1 Zoll.

Auf der Karte fließt während normalem Betrieb ein Strom von 400 Milliampere. Als Höchstwert kann 800 Milliampere erreicht werden.

Alle Ausgaben der Karte sind auf drei Zuständen aufgebaut. Die Endwiderstände sind im IEEE-488-Standard spezifiziert.

Glossar

APPLESOFT: Die Gleitpunkt-Version von BASIC, die vom Apple-Computer verwendet wird. Sie besitzt die CHR\$-Funktion. Diese wird zur Erzeugung von Sekundäradressen benötigt.

Asynchron: Gegenteil von synchron. Synchrone Ereignisse werden im allgemeinen von einem gemeinsamen Zeittakt gesteuert. Bei asynchronen Ereignissen ist dies nicht der Fall.

Gepufferte Eingabe: Wenn Daten schneller zu einem Gerät gesandt werden, als sie dort verarbeitet werden können, so werden sie zwischenzeitlich in einem speziellen Speicherbereich gehalten (einem Eingabepuffer). Das Gerät bearbeitet sie dann zeilich versetzt. Eingabepuffer befinden sich im allgemeinen innerhalb der Geräte.

Steuerfunktion: Eine Funktion, die die Tätigkeit eines Geräts überwacht.

CTRL: Beschriftung einer Taste links auf der Tastatur des Apple-II-Computers. Diese Taste muß gleichzeitig mit einer anderen gedrückt werden, um die Kontrollcodes der Tabelle A-4, Anhang A zu erzeugen.

Steuerteil: Gerät, das die gesamte Übertragung von Daten und Steuerbytes auf dem Bus überwacht. Den Steuerteil für den Bus bildet der Apple zusammen mit der IEEE-488-Karte.

Daisy Chained: Geräte, die hintereinanderliegend mit dem Bus verbunden sind, werden 'daisy chained' genannt. Eine weitere Art der Verbindung mit dem Bus ist die sternförmige Anordnung.

Direkter Speicherzugriff (DMA, Direct Memory Access): Speicherzugriff mit hoher Geschwindigkeit unter Umgehung der Zentraleinheit (in unserem Fall der 6502). Der Apple-Prozessor kann vom Speicherzugriff befreit werden. Dazu ist Pin 22 im E/A-Anschluß zu verwenden.

Quittungsaustausch: Eine Folge von Signalen auf dem Bus, mit dem die Geräte ihren Zustand anzeigen: empfangsbereit, sendebereit oder keines von beiden. Die drei Quittungsleitungen eines IEEE-488-Busses sind NREFD, DAV und NDAC.

Integer-Basic: Die Apple-Version von Basic, bei der die Gleitkomma-Arithmetik fehlt. Die CHR\$-Funktion kann mit Programm 1 aus Anhang D simuliert werden.

Interface: Der Begriff hat in diesem Handbuch folgende Bedeutung: Hardware und Software, die die Kommunikation verschiedener Geräte ermöglichen.

Jumper: Ein oder mehrere Drähte (manchmal zu einem Stecker geführt), die auf verschiedene Weise verbunden werden können. Sie können mit Jumpers eine bestimmte Speicherkonfiguration des Apple auswählen. Andere Hersteller benutzen Jumper zum Festlegen der Geräteadressen.

Empfänger: Gerät, das Daten vom Bus empfangen kann.

Lokale Operation: Tätigkeit des Geräts, die mit den Knöpfen und Schaltern der Bedienfront ausgelöst wird. Im Gegensatz dazu steht die gesteuerte Operation, die unter Kontrolle des GPIB abläuft.

Paralleles Polling: Eine Methode der gleichzeitigen Abfrage vieler Geräte (beim GPIB bis zu acht) nach einem Bedienungswunsch. Ein Gerät, das auf Bedienung wartet, antwortet dem Polling, indem es ein Bit im Statusbyte setzt. Dieses Byte wird mit dem Kommando PP gelesen.

Huckepack-Verbindungsstecker: Verbindung, die Stecker über ihren Pins haben. Damit können mehrere Kabel an einem Punkt verbunden werden.

Gesteuerte Operation: Tätigkeit eines Geräts unter Kontrolle eines Steuergeräts (zum Beispiel der Steuerteil des GPIB).

Seriellles Polling: Eine Art des Polling, bei dem die Geräte nacheinander abgefragt werden, ob sie bedient werden müssen. Diese Art ist zwar langsamer als paralleles Polling, kann jedoch auf mehr Geräte angewendet werden. Seriellles Polling wird angestoßen, wenn die Spannung in der SRQ-Leitung herabgesetzt und ein SRQ-Kommando gegeben wurde.

Standard: Einhaltung der Vorschläge eines Normungsgremiums. Produkte, die demselben Standard angepaßt sind, sind zueinander kompatibel. So berücksichtigt zum Beispiel das Apple-GPIB den IEEE-488-Standard.

Sternförmige Anordnung: Buskonfiguration, bei der ein Gerät mit der Nabe eines Rades verglichen werden kann. Die anderen Geräte befinden sich auf der Felge, und die Speichen stellen die Verbindungen zwischen ihnen dar. Diese Konfiguration ist nicht zu empfehlen.

Sender: Ein Gerät, das Daten über den Bus senden kann. Der IEEE-488-Standard erlaubt zu jedem Zeitpunkt nur einen Sender.

Term: Abkürzung für Terminator.

Terminator: Ein spezielles Symbol oder Zeichen, das zur Endmarkierung benutzt wird (zum Beispiel für eine Zeichenfolge). <CR>, <EOS> und EOI werden im GPIB als Terminatoren benutzt.

TMS 9914 - GPIB-Adapter-Chip, an das der Apple-GPIB angepaßt ist. Dieses wird von Texas Instruments hergestellt.

Trigger: Sofortiges Veranlassen eines Ereignisses. Triggers werden zur Synchronisation oder zum Einleiten von Ereignissen benutzt.

E

- Eigene Senderadresse 46
- Ein-/Ausgaberegister 58
- Ein-/Ausgeschaltet 23
- Eingabe mit Kommas 57
- Elektrische Anforderungen 13
- Empfängerliste 2 1-26, 29-34, 35
- Empfänger abschalten 24
- Empfängeradressen 24, 40
- Eigene Empfängeradresse 46
- Empfänger 10, 23, 80
 - Primär- 23
- Empfangsquittungsfunktion 17
- Empfangsroutine 62
- Enable 22-23
- Enable-Byte 35
- Enable-Liste 23, 35
- EOI-Leitung 14-15, 28-31, 61-62
- EOS-Zeichen 23, 26, 27, 29-31, 61-62
- Erdung 6, 14
- ES-Kommando 22, 26-28, 50

F

- Fehlermeldungen 36-37
- Fernbetrieb
- Freigabe des parallelen Polling 34, 46
- Freigabe des seriellen Polling 46
- Funktion des Steuerteils 18
- Funktionen
 - Steuer- 79
 - IEEE- 16-18

G

- Gepufferte Eingabe 13, 79
- Geräte
 - Maximalzahl 10, 11
 - Test 3
- Gerätenummer 23
- Gerät rücksetzen (?) 46
- Geräterücksetzfunktion 17
- Gerätetriggerfunktion 18
- GET-Anweisung 46, 57
- Gezieltes Rücksetzen eines Geräts 46
- Go to local 46
- GPIB
- GTL 46

H

- Hexadezimalziffer 23
- HP-IB 1
- HP-3438A-Multimeter 74
- HP-3586C-Selective-Level-Meter 74-75
- Huckepack-Verbindungsstecker 10, 80

I

- IFC-Leitung 14, 15
- IEEE-Standard 1-3
- IEEE-Standard-Schraube 4, 6-7
- IEEE-Verbindungsstecker 6
- IEEE-488-Bus 1
- IEEE-488-Karte 1, 78
- INPUT-Anweisung 25, 57
- IN n 19, 26, 55-56
- Installation 4-8
- Integer-Basic 79
- Interface 79
- Interface-Clear-Funktion 18
- Interface-Clear-Leitung 15
- Interfacekarte
 - Apple-Language 5

J

- Jumper 25, 79

K

- Kabel
 - Apple-IEEE-Verbindung 5
 - IEEE-Standard 13-17
- Kartenstecker 6
- Kettenförmig (daisy-chained) 10-12, 79
- Kommandos 21-36
 - Datenübertragung 27
 - Beschreibungen 25
 - Protokoll 26
 - Festlegen der Arbeitsweise 27
 - String 21
- Kommas, Eingabe mit 57
- Konfiguration des parallelen 36, 46
 - Polling auflösen
- Konfiguration für paralleles 46
 - Polling

L

Länge des Busses 11
LA-Kommando 22, 34, 48
Leitungen
 ATN 15
 DAV 16
 EOI 15
 IFC 15
 NDAC 16
 NRFD 16
 REN 15
 SRQ 15
LF-Kommando 19, 22, 27, 29-31
LL-Kommando 22, 33, 48, 76
LLO 46
Lokale Operation 80
Lokalsperre 46
LO-Kommando 22, 33, 48
LST-Fehler 36
LSTN-Busfolge (?) 46
L1-Funktion 17

M

Maximale
 Länge des Busses 11
 Anzahl von Geräten 10
 Stromstärke 78
Maximalzahl von Geräten 10-11 metallische
 Anschlußhalterung 6
Metallisiert, innen (?) 7
Metallklemmverbinder (?) 5
MLA 46
MTA 46

N

NDAC-Leitung 14, 16
Negativlogik (?) 14
Nicht bereit für Daten 16
NRFD-Leitung 14, 16
Nur-Lesespeicher (ROM 1)

O

Öffnen des Apple 4
ONERR GOTO-Anweisung 37
ONERR-Routine 55
Operation
 lokale 80
 gesteuerte 80

P

Paralleles Polling 35, 80
 Festlegen der Konfiguration 46
 abschalten 35, 46
 freigeben 35, 46
 Funktion 17
 Auflösen der Konfiguration 36, 46
Pascal 58, 72
PD-Kommando 22, 35, 49
PE-Kommando 22, 35, 49
PP-Kommando 22, 34, 49
PPC 46
PPD 46
PPE 46
PPU 46
Primär- 23
Primärempfänger 23
Primärsender 23
PRINT-Anweisung 25, 27, 57
PR n 19, 26, 28, 29, 55-56
Programmbeispiele 74
Programmieren des GPIB 19
Prüfgerät 3*
PU-Kommando 22, 49

Q

Quittungsaustausch (handshake) 15, 79
Quittungs-Austauschfunktion (handshake) 17

R

RA-Kommando 22, 32, 48, 76
RC-Kommando 22, 31, 47
RD-Kommando 22, 30, 47, 56, 58, 74, 75
Register
 Ein-/Ausgabe- 58
 9914- 52
Remote-Enable (Fernsteuerungsfreigabe) 15
Remote-Enable-Funktion 17
REN-Leitung 14, 15, 33-34
RESET-Taste 28, 33
RM-Kommando 22, 32, 48
ROM (Nur-Lesespeicher) 1
Routinen für schnelle Übertragung 58-69
Rücksetzen aller Schnittstellen 36
Rücksetzen der instrumenten-
 betriebszustände 20
Rücksetzen der Karten-Standardwerte 19, 26
Rücksetzen eines Geräts 46

S

SC-Kommando 19, 22, 26-28, 50, 55, 56
Schaltungsanordnung 78
Schließen des Apple 7
Schraube 4, 6, 8
SDC 46
Sender
 Adressen 10, 22-24, 29-32, 80, 41
Sekundäradresse 23-24, 44
Sender abschalten 24
Senderroutine 61
Sender-Busfolge 46
Seriellles Polling 34, 80
 abschalten 46
 freigeben 46
SH1-Funktion 16
Slot. (Steckfassung) 1, 5
SPD 46
SPE 46
SP-Kommando 22, 34, 48
Speicher-Direktzugriff (DMA) 13, 79
Spezifikationen 78
Sonderzeichen 3
SR-Kommandos 22, 33, 48, 56
SRQ 23
SRQ-Leitung 14, 15, 33-34
Standard 80
Standardwerte 19
 nicht zurücksetzen 26
 rücksetzen 26
Statische Entladung 4
Steckfassung (Slot) 1, 5
Steckerleiste 78
Steckplatz (slot) 1, 5
Sternförmig verbunden 10-12, 80
Steuerfunktion 79
Steuerleitungen 10, 14, 15
Steuerteil 10, 79
Steuerzeichen 45
Steuerzeichenparameter 21
Strichpunkt 26, 31
Strom 78

T

Terminator 22-23, 29-31, 80
TG-Kommando 20, 22, 32, 47, 74-75
TMS-9914 52, 58, 60, 72, 80
Tiggern 80
Tiggern einer Gerätegruppe 18, 32, 46
T3-Funktion 17

U

Übergehen der Reinitialisierungs-
 56 funktion
Übertragungssteuerleitungen 15
UNL-Kommando 24, 46
UNT-Kommando 46
UT-Kommando 22, 36, 56

V

Verbindungskabel des
 Apple IEEE 5
Verbindungsstecker
 Karte 6
 IEEE 6
 Huckepack 10
Verwendung von Adressen 25

W

Warteroutine 66
WC-Kommando 22, 29, 47, 56
WT-Kommando 22, 29, 47, 49-50, 55-56, 58, 75

X

XFER-Busfolge 46
XF-Kommando 22, 31, 47

Z

Zählparameter 21-26, 29-31
Zustandswort 23, 34-35
auf drei Zuständen aufgebaute
 Operation 78

Apple GPIB-Kommandos (Überblick)

<u>COMMAND</u>	<u>FORMAT</u>	<u>BASIC</u>	<u>BUS?</u>	<u>PAGE</u>
WRITE	WT <listen list> <data> <EOS>	PRINT	YES	29
WRITECNT	WC <listen list> <count> <data>	PRINT	YES	30
READ	RD <talker> <CTRL-Z> <data> <term>	INPUT	YES	30
READCNT	RC <talker> <CTRL-Z> <count> <data> <term>	INPUT	YES	31
XFER	XF <talker> <CTRL-Z> <listen list> <CR>	PRINT	YES	32
TRIGR	TG <listen list> <CR>	PRINT	YES	32
CLRAL	CA <CR>	PRINT	YES	32
CLEAR	CL <listen list> <CR>	PRINT	YES	33
REHAL	RA <CR>	PRINT	YES	33
REHDV	RM <listen list> <CR>	PRINT	YES	33
LLKAL	LL <CR>	PRINT	YES	34
LOCAL	LA <CR>	PRINT	YES	34
LOC DV	LO <listen list> <CR>	PRINT	YES	34
SRQD	SR <srq> <CR>	INPUT	YES	35
SPOLL	SP <talker> <CTRL-Z> <status word> <CR>	INPUT	YES	35
PPOLL	PP <status word> <CR>	INPUT	YES	36
PPENB	PE <enable list> <CR>	PRINT	YES	36
PPDIS	PD <listen list> <CR>	PRINT	YES	37
PPUAL	PU <CR>	PRINT	YES	37
ABORT	AB <CR>	PRINT	YES	37
LINEFEED	LF <off/on> <CR>	PRINT	NO	27
EOS	ES <EOS> <CR>	PRINT	NO	28
SCREEN	SC <off/on> <CR>	PRINT	NO	28
DEVICE	DV <device number>	PRINT	NO	29
UNTALE	UT <CR>	PRINT	YES	37

Apple GPIB-Kommando-Strings

<listen list> ::= <CTRL-Z> | <listener> <CTRL-Z> | <listener> <listen list>
<listener> ::= <primary listener> | <primary listener> <secondary address>
* <primary listener> ::= <SP> | ! | " | # ... > | ?
* <secondary address> ::= ' | a | b | c ... } | ~
<count> ::= <decimal digit> <CTRL-Z> | <decimal digit> <count>
NOTE: The maximum total value for count is 255.
<EOS> ::= any ASCII character
<data> ::= empty | any ASCII character | any ASCII character <data>
<talker> ::= empty | <primary talker> | <primary talker> <secondary address>
* <primary talker> ::= @ | A | B | C ... ^ | _
* <hex digit> ::= 0 | 1 | 2 ... 9 | A | B ... F
* <decimal digit> ::= 0 | 1 | 2 ... 9
<status word> ::= <hex digit> <hex digit>
<srq> ::= ASCII "T" | ASCII "F"
<term> ::= empty | <CR> | <EOS> | EOI
NOTE: empty if last char = <CR> or <CR><LF>
<device number> ::= <decimal digit> <decimal digit>
NOTE: Maximum device number is 30
<off/on> ::= ASCII "0" | ASCII "1"
<enable list> ::= <listener> <enable> | <listener> <enable> <enable list>
* <enable> ::= @ | A | B ... N | O
<CR> ::= ASCII carriage return (13 decimal)
<LF> ::= ASCII linefeed (10 decimal)
<SP> ::= ASCII space (32 decimal)
<CTRL-Z> ::= ASCII SUB (26 decimal)
<CTRL-D> ::= ASCII ET (4 decimal)

* NOTE: Marked elements are ASCII characters.

 apple computer