

**Gerry Kane
Steve Harper
David Ushijima**

Das HP-IL-System

**Einführung in die
Hewlett-Packard
Interface-Schleife**



Gerry Kane, Steve Harper, David Ushijima · Das HP-IL-System

Gerry Kane
Steve Harper
David Ushijima

Das HP-IL-System Einführung in die Hewlett-Packard Interface-Schleife

Übersetzt von Ursula Wilk und Otmar Frey

McGraw-Hill Book Company GmbH

Hamburg · New York · St. Louis · San Francisco · Auckland · Bogotá · Guatemala
Johannesburg · Lissabon · London · Madrid · Mexiko · Montreal · New Delhi
Panama · Paris · San Juan · São Paulo · Singapur · Tokio · Toronto

Titel der Originalausgabe:
»The HP-IL System:
An Introductory Guide to the Hewlett-Packard
Interface Loop«
by Gerry Kane, Steve Harper, David Ushijima
© Copyright 1982 by McGraw-Hill, Inc.

Gerry Kane, Steve Harper, David Ushijima
Das HP-IL-System
Einführung in die Hewlett-Packard
Interface-Schleife
Hamburg: McGraw-Hill Book Company GmbH, 1984
ISBN 3-89028-009-9

Herausgeber und Verlag übernehmen für die Fehlerfreiheit der Programme keine Gewährleistung oder Haftung, da es nach dem Stand der Technik nicht möglich ist, Datenverarbeitungsprogramme so zu entwickeln, daß sie fehlerfrei arbeiten.
Der Verlag übernimmt keine Gewähr dafür, daß die beschriebenen Verfahren, Programme usw. frei von Schutzrechten Dritter sind.

© Copyright 1984 by McGraw-Hill Book Company GmbH, Hamburg
Alle Rechte vorbehalten. Ohne ausdrückliche, schriftliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus in irgendeiner Form durch Fotokopie, Mikrofilm oder ein anderes Verfahren zu vervielfältigen oder zu verbreiten. Dasselbe gilt für das Recht der öffentlichen Wiedergabe.

Umschlaggestaltung: Jan Enns, Wentorf
Übersetzung: Ursula Wilk, Otmar Frey
Textverarbeitung: CORVALLIS TEAM GmbH, Friedrichsdorf
Druck und Bindung: Druckerei Bitsch KG

Vorwort des Übersetzers

Die Nomenklatur des HP-IL Interface-Systems beruht auf der von Hewlett-Packard entwickelten "HP-IL reference specification", einer grundlegenden Definition aller mit diesem System zusammenhängenden Begriffe, Konventionen und Verfahrensvorschriften. Dieses in englischer Sprache verfasste Werk bildet die Grundlage des vorliegenden Buches von Kane, Harper und Ushijima: "THE HP-IL SYSTEM: An Introductory Guide to the Hewlett-Packard Interface Loop". In diesem Buch wird durchgehend die von HP vorgegebene Nomenklatur verwendet. Somit ist nach dem Durcharbeiten dieser Einführung ein reibungsloser Anschluß zur "HP-IL reference specification" gewährleistet. Die Zielsetzung der Übersetzer ist es, diesen Anschluß auch in der vorgelegten deutschen Übersetzung beizubehalten. Viele der englischen Fachausdrücke wurden aus diesem Grunde nicht übersetzt, sondern in englischer Originalversion belassen. Deutsche Übersetzungen wurden nur dann gegeben, wenn diese fällig sind. Von einer zwanghaften Eindeutschung eines jeden Begriffes wurde abgesehen. Dem Leser steht somit ein Wortschatz zur Verfügung, der an die weitergehende Literatur heranführt.

Vorbehalt

Die Zustandsdiagramme, die in diesem Buch abgebildet sind, können von Hewlett-Packard ohne besondere Mitteilung geändert werden. Um größtmögliche Genauigkeit zu erreichen, sollten Sie sich beim Aufbau eines HP-IL Systems immer auf die "HP-IL reference specification" beziehen.

INHALT

1	Eine Einführung in das HP-IL	1
2	Auf der Schleife	17
3	Anwendung eines Allzweck-Interfaces zum HP-IL	29
4	Ein HP-IL Interface auf der Ebene der elektronischen Bausteine	55
5	Einige typische Meldungsabfolgen	87
6	Die Interface-Funktionen	99
7	Was Sie außerdem wissen sollten	119
	Anhang	131
	Glossar	141
	Liste der Fußnoten	157
	Index	161

Kapitel 1

Eine Einführung in das HP-IL

HP-IL (Abkürzung für Hewlett-Packard Interface(1) Loop(2)) ist ein Interface-System. Es definiert die Art und Weise, wie eine große Bandbreite von festinstallierten Geräten, Tischcomputern und tragbaren Geräten in einem einheitlichen System zusammenarbeiten.

Die Eigenschaften des HP-IL sind in einem Dokument niedergelegt, das bekannt ist unter dem Namen "HP-IL reference specification". Die Spezifikation umfaßt sowohl die physikalische Verbindung als auch ein Meldungsprotokoll zwischen einer Anzahl von Geräten und Controllern(3). Bevor wir nun im Text fortfahren, wollen wir einige dieser Begriffe näher erläutern.

Die Notwendigkeit der Kommunikation

Der Begriff der Kommunikation ist bei Rechnersystemen sehr entscheidend. Die meisten Systeme bestehen aus einer Anzahl unterschiedlicher Geräte. Jedes einzelne Gerät ist dabei von seiner Konstruktion her dafür vorgesehen, eine spezielle Aufgabe auszuführen. Wenn diese Geräte in einer planmäßigen Weise zusammenarbeiten sollen, müssen sie miteinander kommunizieren können. Zu diesem Zweck müssen sie physikalisch verbunden werden und, was ebenso wichtig ist, sie müssen eine gemeinsame Sprache sprechen.

Die physikalische Verbindung

Ein Hardware-Interface erlaubt die Verbindung eines Gerätes mit einem anderen durch ein physikalisches Bindeglied. Die Beschaffenheit dieser Verbindung unterscheidet sich von Fall zu Fall (siehe Abb. 1.1)

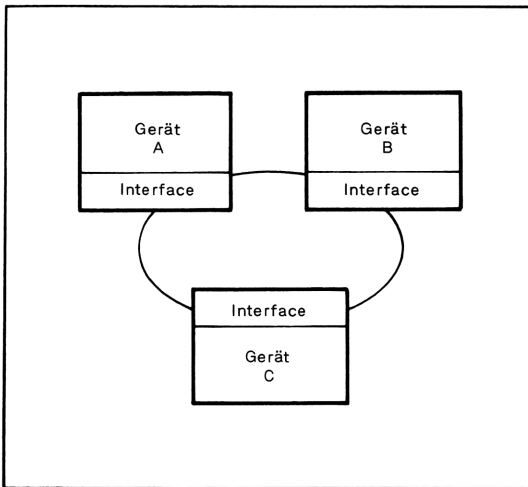


Abb. 1.1 Drei Geräte, die durch ein gemeinsames Interface-System verbunden sind

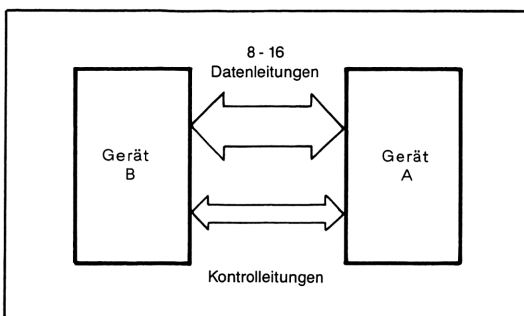


Abb. 1.2 Ein paralleles Interface

Abb. 1.2 zeigt ein paralleles Interface. In diesem Fall besteht es aus acht Datenleitungen und einem Satz Kontroll-

leitungen. Die Information wird von Gerät A nach B in acht parallelen Bits gleichzeitig über die acht Datenleitungen übertragen. Die Kontrollleitungen werden benutzt, um die Datenübertragung zwischen den Geräten zu synchronisieren, ein Prozeß, den man als "Handshake"(4) bezeichnet. Typischerweise wird Gerät A Gerät B signalisieren, daß es gültige Daten auf die Datenleitungen geschickt hat. Wenn die Daten einmal angenommen worden sind, zeigt B den Erhalt des Datums an, indem es die dafür vorgesehene Kontrollleitung benutzt.

In Abb. 1.3 ist ein serielles Interface dargestellt. Es besteht aus einer einzigen Datenleitung und einer Referenzleitung. Zu einem bestimmten Zeitpunkt wird nur jeweils ein Bit über die einzelne Datenleitung von Gerät A nach Gerät B übertragen. Man kann hier leicht feststellen, daß ein serielles Interface die Datenübertragungsgeschwindigkeit einschränkt. Es ist jedoch einfacher im Aufbau und für die meisten Anwendungen im langsamen bis mittleren Geschwindigkeitsbereich ist die Datenübertragungsrate eines seriellen Interfaces mehr als ausreichend.

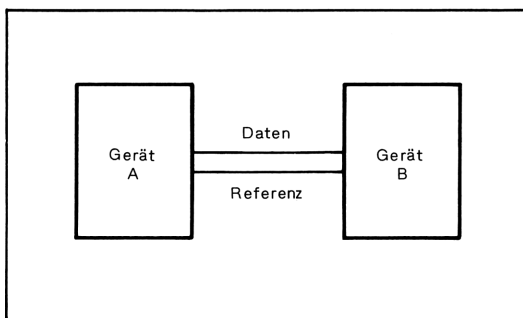


Abb. 1.3 Ein serielles Interface

Aufbau

Wenn das allgemeine Wesen der physikalischen Verbindung einmal als parallel oder seriell festgelegt ist, gibt es eine Vielzahl von Möglichkeiten, ein Interface-System aufzubauen. Eine der gebräuchlichsten Hardwarekonfigurationen ist das Bus-System(5) in Abb. 1.4. Parallele Bus-Inter-

faces werden normalerweise in Fällen gebraucht, bei denen hohe Geschwindigkeiten wichtig sind. Die Information wird über den Bus übertragen, und jedes Gerät, das an den Bus

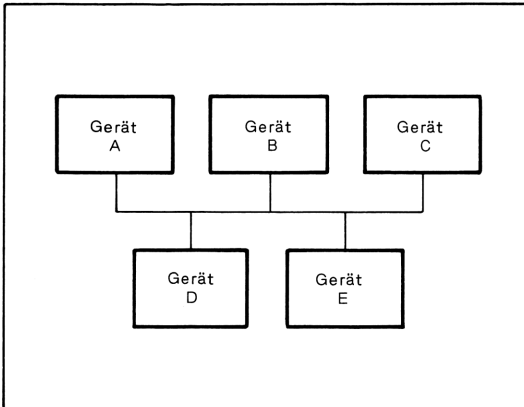


Abb. 1.4 Ein Bus Interface-System

angeschlossen ist, hat zur gleichen Zeit (oder so gleichzeitig wie physikalisch möglich) die gesamte Bus-Information. Normalerweise ist der einschränkende Faktor des Bus-Systems die Länge des Übertragungsmediums. Größere Entfernungen verschlechtern die Qualität der Signale und führen zu Fehlern.

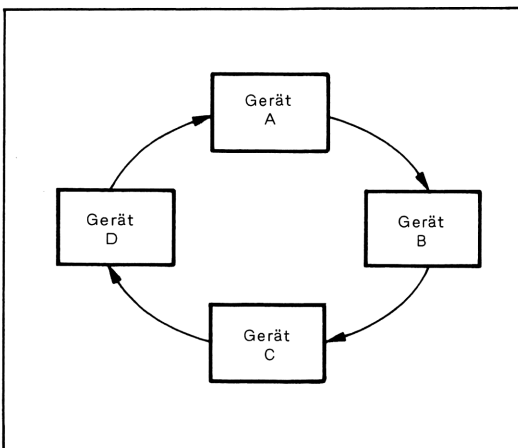


Abb. 1.5 Ein Ring- oder Schleifeninterface

Eine andere Interface-Struktur, gewöhnlich Ring- oder Schleifeninterface genannt, sehen Sie in Abb. 1.5. Die Information wird über die Schleife gesendet und geht von Gerät zu Gerät, bis sie zu ihrem Ausgangspunkt zurückkehrt. Als Beispiel für diese Funktionsweise wollen wir annehmen, daß Gerät A in Abb. 1.5 eine Information sendet, die für Gerät D bestimmt ist. Die Information geht über die Schleife über die Geräte B und C, bis sie bei Gerät D ankommt, wo sie weiterverarbeitet wird. Währenddessen läuft die Information weiter rund um die Schleife, bis sie zum Ausgangspunkt zurückkehrt, in diesem Falle Gerät A. Jetzt vergleicht Gerät A seine neu erhaltene Information mit der, die es abgeschickt hat. Dadurch wird eine Fehlerüberprüfung vorgenommen.

Die Geschwindigkeit einer Ring- oder Schleifenstruktur hängt von der Zeitspanne ab, die die Information braucht, um einmal vollständig über die Schleife übertragen zu werden. Beim gegenwärtigen Stand der Nachrichtentechnik können jedoch ausreichende Übertragungsgeschwindigkeiten erreicht werden.

Die zwei wichtigsten Vorteile einer seriellen Schleifenstruktur sind die niedrigen Kosten und der geringe Energieverbrauch. Da nur zwei Adern für die Verbindung notwendig sind, ist auch die Möglichkeit eines Hardware-Versagens im Vergleich zum Bus-System (vgl. Abb. 1.4.) sehr viel geringer.

Protokoll und Kompatibilität

Um die Kommunikation zwischen Geräten zu ermöglichen, ist mehr nötig als nur eine gemeinsame physikalische Verbindung. Damit sich Geräte verständigen können, müssen sie die Information in einer Form senden, die von allen Geräten verstanden wird. Die Geräte müssen weiterhin einer Anzahl von gemeinsamen Regeln folgen, die die Art der Kommunikation über die physikalische Verbindung bestimmen. Dies ist bei einem seriellen Interface besonders wichtig, da es hier keine Kontrolleitungen gibt, die die Handshake-Funktion übernehmen können. Das gemeinsame Regelsystem und das Format der einzelnen Meldungen bezeichnet man als Meldungsprotokoll.

Verschiedene Hersteller konstruieren ihre Protokolle aus-

schließlich für ihre eigenen Computersysteme. In den meisten Fällen sind deshalb die Protokolle verschiedener Hersteller nicht kompatibel (verträglich). Zum Beispiel ist eine Information, die von einem IBM-Protokoll wie dem SDLC gesendet wird, nicht kompatibel mit der Information eines DEC-Computers und umgekehrt.

Um dieses Problem zu lösen, hat man über Jahre hinweg Interface-Standards weiterentwickelt. Dies erlaubt es heute den Herstellern von Datenkommunikations- und Datenverarbeitungsgeräten, ihre eigenen Geräte unabhängig voneinander herzustellen und trotzdem einen gewissen Grad an Kompatibilität sicherzustellen. Ein solcher Standard ist der IEEE-488, auch bekannt als GPIB (kurz für: General-Purpose Interface-Bus(6)).

Vom HP-IB zum HP-IL

Der IEEE-488 wurde ursprünglich von Hewlett-Packard für die Computer und Meßgeräte unter dem Namen HP-IB entwickelt. Es ist ein paralleles Bus-Interface, das allen Geräten auf dem Bus erlaubt, unter der Aufsicht eines oder mehrerer Controller miteinander zu kommunizieren. 1975 beschloß das Standards Board of the Institute for Electrical and Electronic Engineers (IEEE)(7), das HP-IB als Standard einzuführen. Über den IEEE-Standard können gegenwärtig mehr als 1000 verschiedene Geräte verbunden werden, die von Firmen rund um die Erde hergestellt werden.

Der IEEE-488 war ursprünglich dafür vorgesehen, eine große Anzahl von Geräten zu verbinden, von den schnellsten bis zu den sehr langsamen. Mittlerweile ist durch den Fortschritt der Technik die Bedeutung der tragbaren Geräte mittlerer Geschwindigkeit stark erweitert worden. Geräte wie der Taschencomputer und seine Peripheriegeräte stellen heutzutage den Trend dar. Durch die Tendenz zu Geräten, die mittlere Leistung erbringen, dazu leichtgewichtig und preisgünstig sind, wurden neue Standards entwickelt, die der neuen Linie der Technologie entsprechen und sie weiterentwickeln. Eine dieser Entwicklungen ist die Interface-Schleife der Hewlett-Packard Corporation.

Obwohl es nicht das IEEE-488 Interface System ersetzen sollte, ist das HP-IL das Ergebnis hin zur kleineren, tragbaren und preisgünstigen Gerätekonfiguration. Das HP-IL

ist hauptsächlich bestimmt für den Benutzer eines kleinen Systems, ob im Labor, im Büro oder zuhause.

Ein Überblick über das HP-IL

Am einfachsten erläutert man das HP-IL, indem man eine typische Konfiguration untersucht. Abb. 1.6 zeigt ein HP-IL

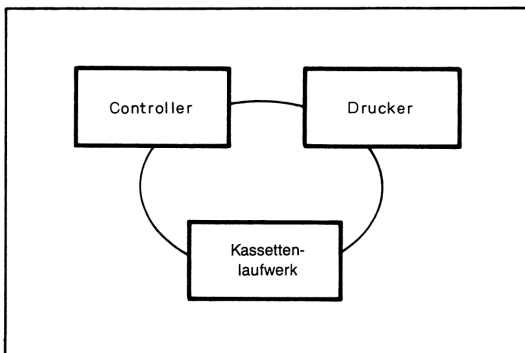


Abb. 1.6 Ein einfaches HP-IL System

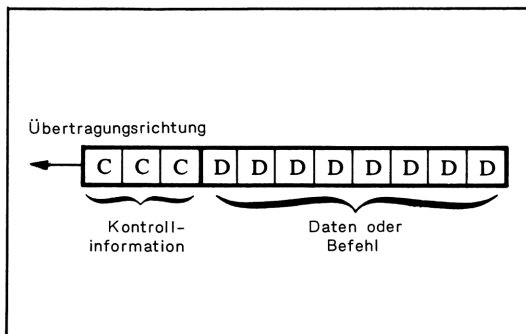


Abb. 1.7 Die Struktur einer HP-IL Meldung

System, bestehend aus einem Controller und zwei anderen Geräten: einem Drucker und einem Magnetbandkassettenlaufwerk. Da das HP-IL ein serielles Kommunikationsinterface ist, wird zu einem bestimmten Zeitpunkt jeweils nur ein Bit über die zweiadrige Leitung übertragen, die Geräte und Controller verbindet. Alle Geräte kommunizieren

(senden und empfangen), indem sie Meldungen über diese zweiadrige Schleife senden. Jede Meldung besteht aus elf Bits, wie in Abb. 1.7 gezeigt wird.

Um sicherzustellen, daß die Geräte und der Controller die gleiche Sprache sprechen, hat Hewlett-Packard eine Meldungsstruktur festgelegt. Diese umfaßt eine gemeinsame Menge von Meldungen, die alle HP-IL Geräte verstehen müssen. Diese Meldungen legen die Art und Weise fest, wie ein Gerät vorgehen muß, wenn es eine Information über die Schleife senden will.

Meldungen, die vom Controller oder einem Gerät gesendet werden, gehen über die gesamte Schleife nacheinander zu jedem Gerät, bis sie wieder beim sendenden Gerät ankommen. Dies ist in Abb. 1.8 dargestellt. HP-IL Systeme können aus

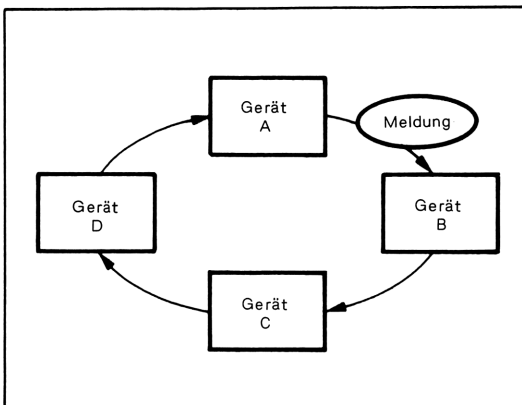


Abb. 1.8 Eine HP-IL Meldung auf der Interface-Schleife

drei Arten von Geräten bestehen: Talker(8), Listener(9) und Controller. Talker sind Geräte, die Daten über die Interface-Schleife senden. Listener sind Geräte, die Daten von einem Talker oder Befehle von einem Controller empfangen. Die Rollen von Talker und Listener werden den Geräten durch den Controller zugewiesen.

Der Controller ist für die gesamte Schleifenoperation verantwortlich. Er ist insbesondere zuständig für die Adressenzuweisung an die Geräte, er bestimmt die Rollen der Geräte, er beantwortet die Serviceanforderungen und ver-

anlasst die Datenübertragung vom Talker zum Listener (bzw. zu den Listnern).

Grundlegende Eigenschaften des Systems

Sie wissen nun darüber Bescheid, was das HP-IL ist. Nachfolgend wollen wir hier eine Beschreibung einiger seiner Fähigkeiten geben.

Maximale Geräteanzahl

Das HP-IL verbindet maximal 31 Geräte auf einer einzigen Schleife, wenn die normale Adressierungsmethode benutzt wird (ein Thema, das später in diesem Buch noch behandelt werden wird). Eine erweiterte Adressierungsmethode (die die Verwendung einer zwei-Byte Adresse einschließt) erlaubt es dem HP-IL Interface, bis zu 960 Geräte miteinander zu verbinden.

Datenübertragungsrage

Die Geschwindigkeit, in der Daten über das Schleifeninterface gesendet werden können, ist theoretisch auf 20 k (20.000) Bytes/Sekunde begrenzt. Bei Benutzung der gegenwärtig (zur Zeit der Drucklegung der deutschen Übersetzung) erhältlichen Geräte sind Geschwindigkeiten von ungefähr 5k Bytes/Sekunde erreichbar. Dies entspricht etwa 1 1/4 DIN A4 Textseite/Sekunde.

Art der Übertragung

Die HP-IL Übertragung geschieht durch die Benutzung eines Drei-Pegel-Codes, wie in Abb. 1.9 dargestellt. Eine logische Eins wird durch einen positiven Impuls von 1,5 Volt dargestellt, dem ein negativer Impuls von -1,5 Volt folgt.

Eine logische Null wird dargestellt von einem Impuls von -1,5 Volt, dem ein positiver Impuls von +1,5 Volt folgt. Ein Pegel von 0 Volt wird benutzt, um den inaktiven Zustand darzustellen.

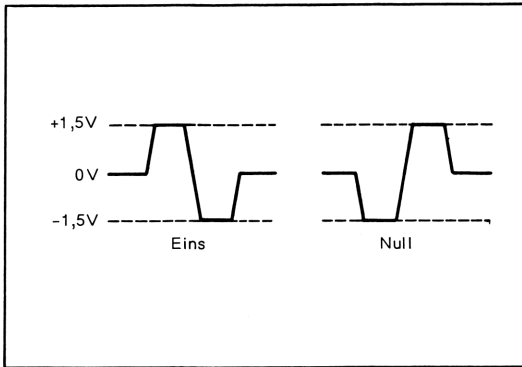


Abb. 1.9 Der HP-IL Übertragungscode

Die Übertragungsleitung ist durch einen Impulstransformator von den Geräten elektrisch isoliert. Dieser wirkt sowohl als Pegel Transformator als auch als Isolator zwischen der Übertragungsleitung und den logischen Pegeln des Geräts. Driver(10) und Receiver(11) sind definiert als Geräte mit zweiadrigen potentialfreien Leitungen an ihren Ein- bzw. Ausgängen. Ein elektrisches Diagramm einer HP-IL Übertragungsleitung ist in Abb. 1.10 dargestellt.

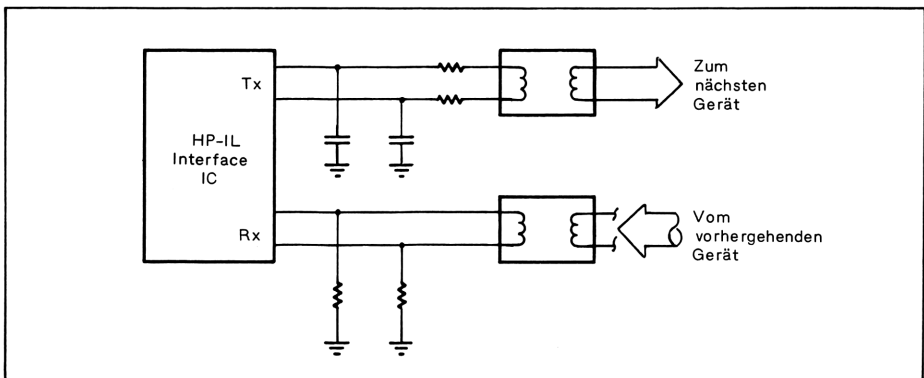


Abb. 1.10 Diagramm einer HP-IL Übertragungsleitung

Übertragungsmedium

Die HP-IL Spezifikation gestattet die Benutzung eines einfachen zweiadrigen Leitung für die Überbrückung einer Distanz von bis zu 10 Metern. Entfernungen von bis zu 100 Meter sind möglich, wenn man eine abgeschirmte verdrillte Leitung benutzt.

Zusätzliche Möglichkeiten

Neben den elementaren Eigenschaften des HP-IL sind eine Anzahl zusätzlicher Funktionen definiert. Eine dieser Funktionen erlaubt es z.B. den Geräten, in einen Zustand mit niedrigem Energieverbrauch zu gehen und auf ein Kommando des Controllers hin wieder voll funktionsfähig zu werden.

Dieser "power-down(12)"-Zustand erlaubt es dem Schleifencontroller, alle Geräte in einen Zustand mit minimalem Leistungsbedarf zu versetzen (eine Eigenschaft, die bei batteriebetriebenen Systemen besonders wichtig ist). Wenn der Controller die Fähigkeit hat, Echtzeit-Aufrufe durchzuführen, kann er alle Geräte aktivieren, damit sie periodisch Messungen oder Kontrollfunktionen ausführen. Wenn diese Funktionen beendet sind, kann der Controller das System wieder in den "Ruhezustand" versetzen.

Triggern(13)

Triggern ist eine Funktion, die es einem Controller erlaubt, eine Handlung an einem ferngesteuerten Gerät auszulösen. So kann ein Signal des Controllers die Aufnahme eines Meßpunktes an einem Meßgerät zu einem bestimmten Zeitpunkt auslösen. Dadurch können z.B. in Laborsystemen ganze Serien von Messungen in bestimmten Zeitabständen aufgenommen werden.

Adressieren

HP-IL Controller haben die Fähigkeit, Geräte auf der Schleife zu initialisieren und den Geräten Adressen zuzuweisen (eine Eigenschaft, die als automatische Adressierung bezeichnet wird). Auf diese Art können Geräte der Schleife hinzugefügt werden, ohne daß man sich um Adres-

sierungsdetails kümmern muß, wie z.B. der richtigen Einstellung von Adress-Schaltern.

Anwendungen

Durch seine leichtgewichtige Konstruktion und seinen minimalen Leistungsverbrauch ist das HP-IL besonders gut für ein weites Anwendungsfeld geeignet, in dem die Mobilität der Geräte ein wichtiger Faktor ist. Einige Beispiele sollen diese Möglichkeiten illustrieren.

Tragbare Datenerfassung

Abb. 1.11 zeigt ein tragbares Datenerfassungssystem, das aus HP-IL Geräten besteht. Solch ein System kann z.B. im

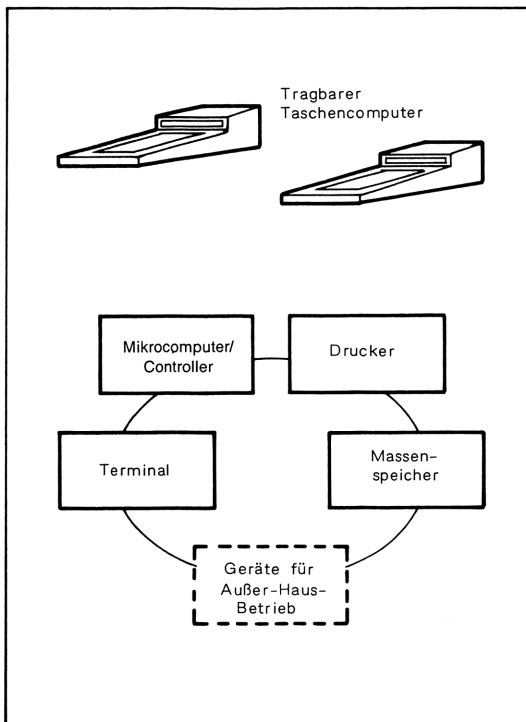


Abb. 1.11 Ein tragbares Datenerfassungs- und Analysesystem

Gelände benutzt werden. Ein Taschencomputer wie der HP-41 kann programmiert werden, um Informationen anzunehmen, die von Außendienstmitarbeitern eingegeben werden. Informationen wie Lagerbestände, Zählerablesungen oder individuelle Antworten auf Marktforschungsuntersuchungen werden direkt in den Taschencomputer eingegeben.

Sobald dieser Vorgang abgeschlossen ist, erfolgt die Datenübertragung zur Zentrale. Jetzt wird das Datenanalyse-system aus Abb. 1.11 benutzt, um die Daten, die außer Haus erfaßt wurden, zu speichern und zu analysieren. Die Daten können auch (durch das HP-IL Interface) auf ein Magnetband oder auf eine Diskette übertragen werden. Wenn sie einmal gespeichert sind, kann man die Daten in jeder gewünschten Art analysieren, indem man ein entsprechendes Programm auf dem Mikrocomputer (Controller) ablaufen läßt. Eine Kopie der Resultate (Lagerbestandslisten oder die Ergebnisse der Marktforschung) können dann auf einem Drucker ausgedruckt werden.

Computergesteuerte Messungen

Das System, das in Abb. 1.12 gezeigt wird, ist für die Datenerfassung mit oder ohne menschliche Hilfe geeignet. Dies kann sowohl außer Haus im Freien sein, wo Umweltdaten gesammelt werden, als auch in einem Labor, das für ein spezielles Experiment eingerichtet wurde. In jedem dieser Fälle kann der tragbare Controller (z.B. ein HP-41) programmiert werden, Messungen in vorherbestimmten Intervallen vorzunehmen und sie für eine spätere Analyse zu speichern.

Weil der Energieverbrauch extrem niedrig ist, kann dieses automatische Meßsystem wochenlang ohne jegliche menschliche Aufsicht arbeiten.

Wenn die notwendige Anzahl von Informationen einmal erfaßt worden ist, läßt sich der tragbare Computer einfach aus dem System entfernen und zum Labor zurückbringen, um die Daten zu speichern und zu analysieren. Nachdem die jeweiligen Datenanalyseprogramme abgelaufen sind, können die Resultate ausgedruckt werden. Als Alternative können die Daten zum Zweck der Analyse auch per Modem über eine Telefonleitung ins Labor übertragen werden.

Selbstverständlich sind dies nur ein paar Beispiele

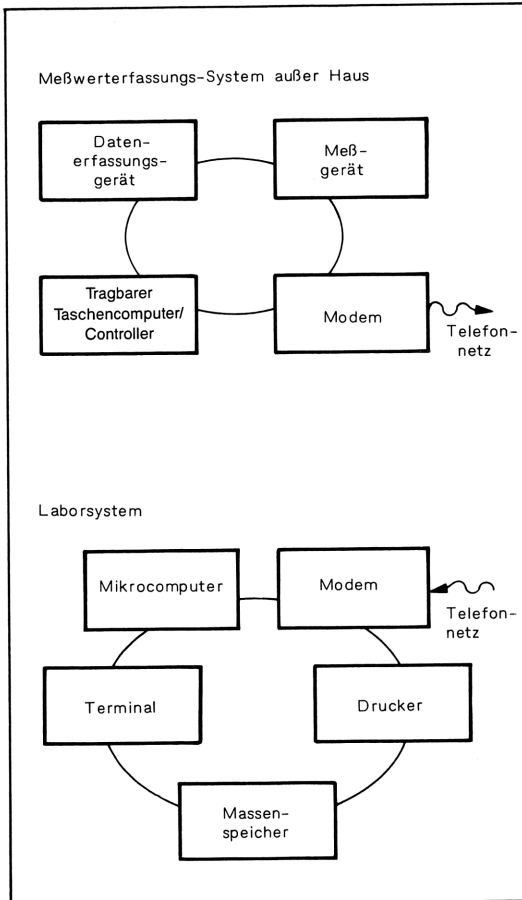


Abb. 1.12 Meßwerterfassung außer Haus und im Labor

dafür, was man mit dem HP-IL System machen kann. Natürlich ermöglicht das HP-IL nicht nur Anwendungen, bei denen es darauf ankommt, daß das System tragbar ist. Für viele andere persönliche oder geschäftliche Anwendungsweisen ist das HP-IL System wegen seiner niedrigen Kosten und seiner Unkompliziertheit hervorragend geeignet.

Ein System wie in Abb. 1.13 kann z.B. benutzt werden, um ein Finanzierungsmodell vorzubereiten oder einen Haushaltsplan aufzustellen. Wenn die Information einmal vollständig ist, kann sie zu einem Zentralrechner übertragen werden, der Tausende von Kilometern entfernt liegt. Durch

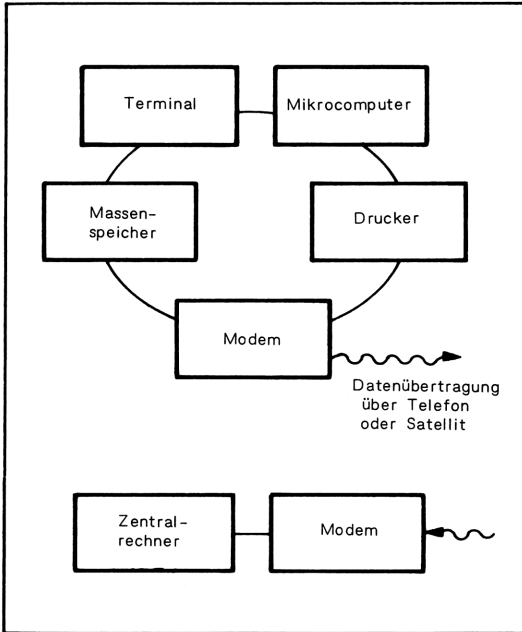


Abb. 1.13 Eine Möglichkeit der geschäftlichen Anwendung unter Einbeziehung der Datenfernübertragung

die Benutzung eines Modems, das an die HP-IL Interface-Schleife angeschlossen wird, ist die Übertragung der Daten per Telefon oder Satellitenverbindung möglich.

Hinweise zu diesem Buch

Da es viele Herangehensweisen an das HP-IL Interface gibt, haben wir dieses Buch so gestaltet, daß es für eine große Anzahl von Benutzern hilfreich ist. Ihr Interesse wird von der jeweiligen Anwendung abhängig sein und kann von einem einfachen Überblick bis zu einem umfassenderen Wissen über die innere Arbeitsweise des HP-IL Meldungsprotokolls reichen.

Wir wollten Ihnen im ersten Kapitel einen Überblick darüber geben, was das HP-IL ist und was es für Möglichkeiten bietet. Diejenigen unter Ihnen, die über die Benutzung eines Systems von HP-IL Produkten nachdenken, werden

in den Kapiteln 1 und 2 genügend Informationen finden, die es Ihnen erlauben, selbständig ein kleines System zusammenzustellen.

Während viele von Ihnen mit dem Wissen darüber auskommen werden, was auf der Schleife jeweils passiert, ist oftmals ein tieferes Verständnis der inneren Arbeitsweise des HP-IL notwendig. Wenn Sie Ihrem System z.B. ein Gerät hinzufügen wollen, das nicht speziell für die Benutzung mit dem HP-IL konstruiert wurde, oder wenn Sie ein HP-IL Interface in Ihr selbstentwickeltes Gerät integrieren wollen, ist es oft notwendig, die Arbeitsweise der HP-IL Meldungen zu verstehen.

In den Kapiteln 3, 4 und 5 stellen wir Ihnen die HP-IL Meldungen und ihre jeweilige Benutzung im typischen Schleifensystem vor. Diese Kapitel sollen Sie mit der HP-IL Meldungsstruktur vertraut machen. Zusätzlich dazu sind die Meldungen am Ende dieses Buches noch einmal in einem Glossar aufgeführt. Eine kurze Zusammenfassung darüber, wie die Meldungen organisiert und klassifiziert werden, erscheint in Anhang A.

Wenn Sie an der Konstruktion eines HP-IL kompatiblen Gerätes interessiert sind, finden Sie im Kapitel 4 die Erläuterung der Bauweise eines HP-IL Interfaces. Die Einzelheiten der funktionalen Spezifikation des HP-IL, die nötig sind, um ein Interface zu konstruieren, werden in den Kapiteln 6 und 7 erläutert. Unsere Absicht ist es, Ihnen genügend Wissen zu vermitteln, so daß Sie die "HP-IL reference specification" besser verstehen; aber unser Buch soll dieses Dokument keineswegs ersetzen.

Kapitel 2

Auf der Schleife

Das HP-IL System umfaßt eine Anzahl von Geräten, die durch eine gemeinsame Interface-Schleife verbunden sind. Es gibt die verschiedensten Möglichkeiten, ein solches System zu gestalten. Welche Geräte Sie brauchen und wie sie zusammenarbeiten sollen, wird bestimmt durch die jeweilige Anwendung und ist nur durch Ihre Vorstellungskraft begrenzt. Der Schwierigkeitsgrad wächst, wenn Sie sich neuen und umfassenderen Anwendungen zuwenden.

Als Ausgangspunkt können Sie ein HP-IL System aus gekauften Geräten zusammenstellen. Systeme, die so aufgebaut werden, können mit einem minimalen Aufwand erweitert werden. Sie müssen die Geräte nur mit den richtigen Kabeln verbinden und das System einschalten.

Wenn Sie sich für die Erweiterungsmöglichkeiten Ihres Systems interessieren, können Sie der Schleife ein nicht standardmäßig verfügbares Gerät hinzufügen. Ein Adapter wie der HP82166A-IL-Konverter(14) erlaubt Ihnen, eine große Anzahl nicht standardgemäßer Geräte mit der Schleife zu verbinden.

Wenn Sie ein HP-IL Interface in ein Gerät, das Sie selbst entwickelt haben, einbauen wollen, gibt es viele unterschiedliche Vorgehensweisen.

Aber zunächst wollen wir ein einfaches Schleifensystem mit fertiggekauften Geräten zusammenstellen.

Ein einfaches HP-IL System

Die einfachste Art, Ihnen eine Einführung in das HP-IL

System zu geben, besteht darin, einige der HP-IL Geräte zusammenzustellen, die von Hewlett-Packard angeboten werden.

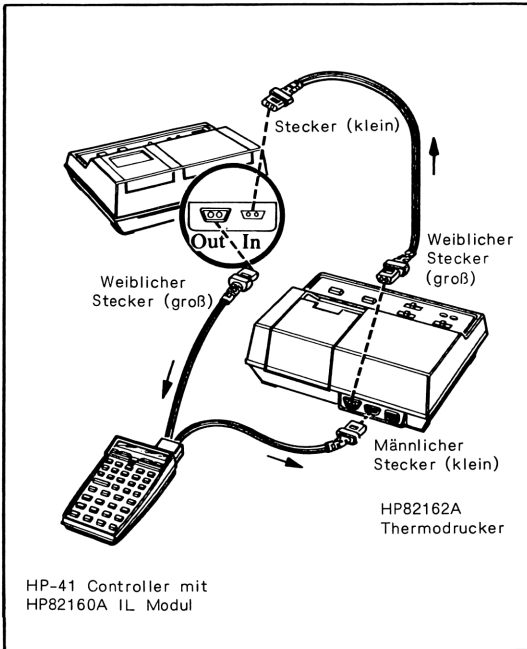


Abb. 2.1 Ein HP-IL System aus fertiggekauften HP-IL Geräten

Abb. 2.1 zeigt ein System, das aus einem HP-41C Taschencomputer, einem HP82160A HP-IL Modul, einem HP82162A Thermo-Drucker und einem HP82161A Digital-Kassettenlaufwerk zusammengesetzt ist. Indem man die Geräte zusammensteckt, wie in Abb. 2.1 gezeigt, ist man mit einem leichten und tragbaren Rechnersystem ausgestattet. Es ist für jede Anwendung geeignet, die einen direkt angeschlossenen Massenspeicher und Druckfähigkeiten benötigt, wie das tragbare Datenerfassungssystem, das in Kapitel 1 beschrieben wurde.

Zur Einführung wollen wir Ihnen hier eine Beschreibung von jedem der im Handel erhältlichen Geräte geben, die in Abb. 2.1 dargestellt sind.

Der HP-41 ist ein äußerst vielseitiger Taschencomputer. Ein näherer Blick auf seine Eigenschaften soll Ihnen einen

besseren Eindruck seiner Fähigkeiten vermitteln.

Die 12stellige LCD-Anzeige des HP-41 ist in der Lage alphanumerische Zeichenketten bis zu der Länge von 24 Zeichen darzustellen (Zeichenketten, die länger als 12 Zeichen sind, werden von rechts nach links durch die Anzeige gerollt). Das Tastenfeld erlaubt die Eingabe von Texten und Zahlen. Die Standardrechnerfunktionen werden ebenfalls über das Tastenfeld ausgeführt. Zusätzlich können Sie jeder Taste des HP-41 die Ausführung von Funktionen oder Programmen im Programmspeicher zuweisen. Den Speicher kann man mittels Einsteckmodulen zu einem Maximum von 320 Registern (1000-2000 Programmzeilen) erweitern. Mit dem HP82160A HP-IL Interface Modul zusammen kann der HP-41 als HP-IL System Controller fungieren.

Das HP-IL Interface-Modul, das in Abb. 2.1 gezeigt wird, wird einfach in einen der vier Erweiterungsschächte des HP-41 eingesteckt. Das Interface Modul besteht aus einem Satz von Routinen im ROM(15), um die HP-IL-Basisfunktionen zu unterstützen. Ebenso enthält es die Interface-Elektronik, die notwendig ist, um das physikalische Interface zum HP-IL herzustellen. Wenn das Interface einmal amgeschlossen ist, ist der HP-41 in der Lage, mit anderen HP-IL Geräten zu kommunizieren und sie zu kontrollieren.

In Abb. 2.2 wird das HP82161A Digital-Kassettenlaufwerk gezeigt. Hier werden Daten und Programme auf HP-Minidatenkassetten gespeichert. Jede Kassette hat eine Speicherkapazität von bis zu 131k Bytes. Die Daten werden in Datensätzen (records) von je 256 Bytes gespeichert (bis zu 512 Datensätze können pro Kassette gespeichert werden). Die Daten werden in der Geschwindigkeit von 23 cm/Sekunde gelesen und geschrieben. Die Such- und Spulgeschwindigkeit der Kassette beträgt 76 cm/Sekunde. In Verbindung mit dem HP-41 können alle Kassettenfunktionen entweder durch direkte Eingabe vom Tastenfeld oder unter Programmkontrolle ausgeführt werden.

Der HP82162A Thermodrucker, dargestellt in Abb. 2.3, ist ein tragbares Gerät, das in der Lage ist, bis zu 24 Standardzeichen oder 12 Zeichen bei doppelter Schriftbreite auf 5,7 cm breites Thermopapier zu drucken. Die Zeichensätze, die der 82162A drucken kann, umschließen sowohl die 96 ASCII-Standardzeichen als auch einen zusätzlichen alternativen Zeichensatz von 127 Zeichen. Der 82162A kann



Abb. 2.2 HP82161 Digital-Kassettenlaufwerk

zusätzlich Bar Codes, Sonderzeichen und Graphiken ausdrucken.

Da die gesamte Kommunikation zwischen den Geräten über die Interface-Schleife abläuft, müssen Sie die Einzelheiten der Meldungen, die über die Schleife gesendet werden, nicht kennen. Das Aussenden und die Reaktion auf die HP-IL Meldungen werden vom Controller und den Geräten selbst übernommen. Im Falle eines Systems wie in Abb. 2.1 wird die gesamte HP-IL Kommunikation durch die Funktionen des HP-IL Interfaces bewerkstelligt. Ein Programm, das die Aufgabe hat, das Zusammenspiel zwischen den verschiedenen HP-IL Geräten zu kontrollieren, muß dementsprechend aus einer Anzahl von Aufrufen dieser Funktionen bestehen. Eine andere Art der Kontrolle besteht in der direkten Eingabe einer Folge von Tastenbefehlen.

Um ein Beispiel für diese Arbeitsweise zu geben, stellen Sie sich vor, Sie wollten per Tastenfeld des HP-41 den Befehl geben, die Kassette abzusuchen und eine Liste der darauf gespeicherten Daten oder Programme auszudrucken.

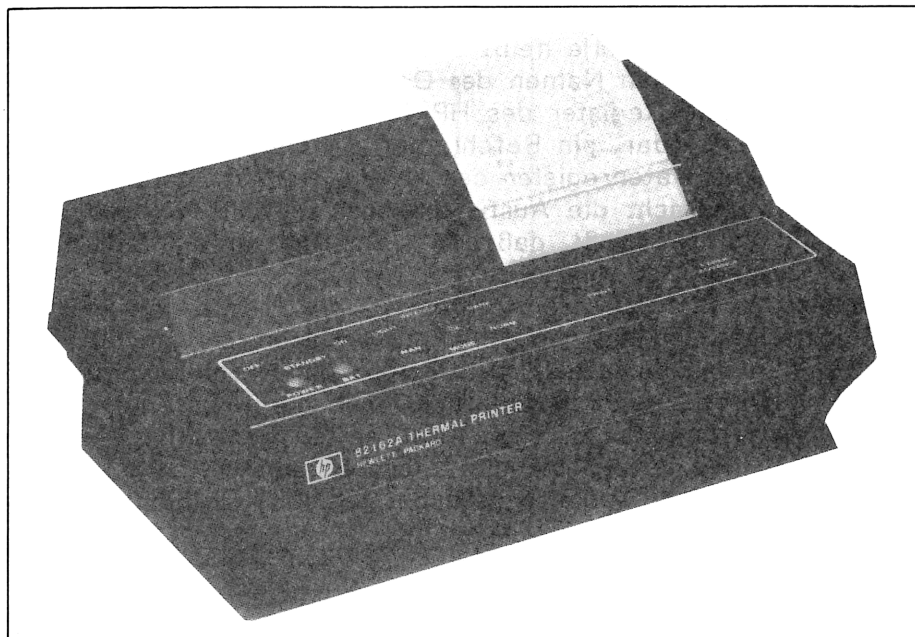


Abb. 2.3 HP82162A Thermodrucker

Geben Sie dazu einfach den "DIR"-Befehl in den HP-41 ein. Die gesamte Kommunikation über die Interface-Schleife bleibt unsichtbar für Sie. Der Controller übersetzt Ihre Befehle in die passende Anzahl von Meldungen und führt alle Einzelheiten aus, die notwendig sind, um das Inhaltsverzeichnis vom Band zu lesen und diese Information an den Drucker weiterzugeben.

Um die Benutzung eines gespeicherten Programmes zum Zweck der Kontrolle des Systems von Abb. 2.1 zu erläutern, folgt nun ein kurzes Programm, das die Befehle enthält, die nötig sind, um ein Datenfeld vom Kassettenlaufwerk zu lesen und vom Drucker drucken zu lassen.

```
01 LBL "SYS1" (Name des Programmes)
02 "DAT" (Name des gewünschten Datenfeldes)
03 READR (Einlesen des Feldes in Datenregister)
04 PRREG (Ausdruck der Datenregister)
05 END
```

Zeile 01 ist eine Programmarke, die dieses Programm kennzeichnet. In diesem Falle heißt unser Programm "SYS1". Zeile 02 schreibt den Namen des Datenfeldes, das wir lesen wollen, ins Alpha Register des HP-41. Zeile 03 stellt den Befehl "READR" dar, ein Befehl, der die Daten von der Kassette in die Datenregister des HP-41 einliest. Zeile 04 schließlich verursacht die Ausführung des Befehls PRREG, ein Befehl, der veranlaßt, daß alle Datenregister auf dem Drucker ausgedruckt werden.

Ogleich es nicht unsere Absicht ist, die Programmierung des Controllers und die des HP-41 zu erklären, sollten Sie (wenn Sie im Handel erhältliche Geräte benutzen) beachten, daß die Kontrolle der Schleifenfunktionen durch Befehle ausgeführt werden, die in einen bestimmten Controller eingebaut sind. Das Wissen um die tatsächlichen HP-IL Meldungen, die auf der Schleife gesendet werden, ist nicht unbedingt notwendig.

Anschluß von Geräten

Vielleicht müssen Sie zu einer bestimmten Zeit einmal ein Gerät anschließen, das kein HP-IL Interface hat. Dies erlaubt Ihnen ein HP-IL Konverter, ohne daß sie zu tief in die HP-IL Meldungsstruktur eindringen müssen.

In Abb. 2.4 sehen Sie einen HP-IL Konverter. Er fügt sich in die Schleife ein, als wäre er ein Standard HP-IL Gerät. Die Meldung, die für ein Fremdgerät bestimmt ist, wird vom Konverter übersetzt und in einer Form an es weitergegeben, die dem Fremdgerät entspricht. Ebenso werden Meldungen, die von dem Fremdgerät ausgehen in das Standard HP-IL Format übersetzt, bevor sie über die Schleife geschickt werden.

Ein Fremdgerät, das in der Lage ist, 8 bis 16 Bits paralleler Daten zu senden bzw. zu empfangen, kann der Interface-Schleife (mit Hilfe eines HP-IL Konverters) angeschlossen werden. Ein solcher Konverter ist der HP82166A, der in Abb. 2.5 dargestellt ist. Meldungen, die für das Fremdgerät bestimmt sind, werden über die Schleife an den Konverter geschickt, der sie für das Gerät in ein Parallel-Datenformat umwandelt. Ebenso werden Parallel-daten des Fremdgerätes vom Konverter in eine HP-IL Standard-Meldung umgewandelt. Handshake-Leitungen sorgen

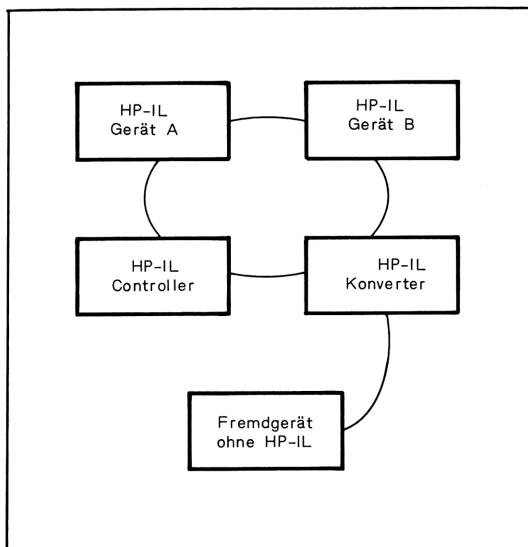


Abb. 2.4 Anschluß eines Fremdgerätes an die Interface-Schleife

sorgen für die ordnungsgemäße Datenübertragung zwischen Konverter und Fremdgerät.

Wenn das Gerät, das Sie an die Interface-Schleife anschließen wollen, ein RS-232 serielles Interface hat,

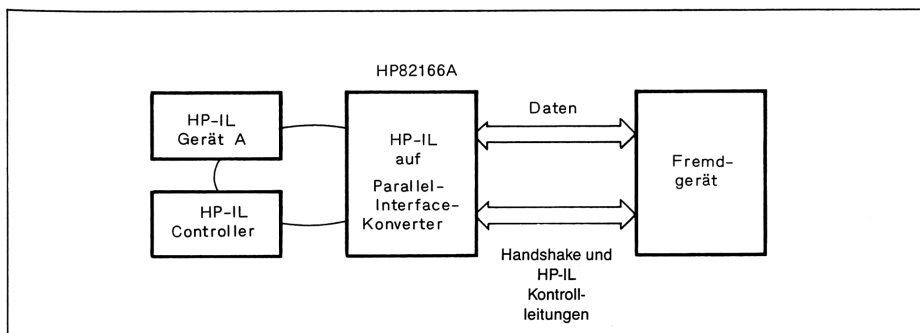


Abb. 2.5 Anschluß eines Fremdgerätes mit parallelem Interface an die Schleife

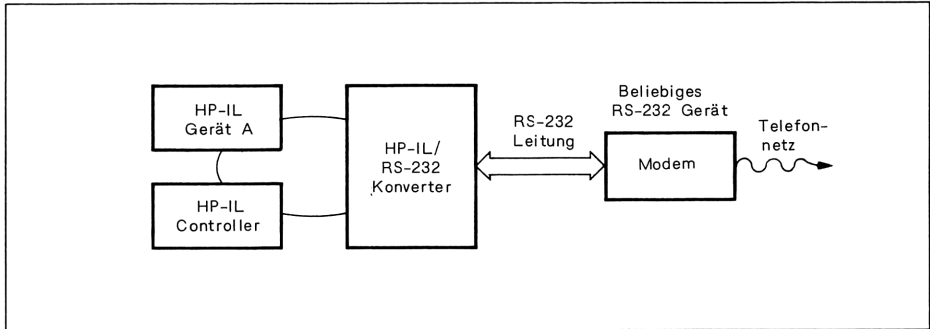


Abb. 2.6 Anschluß eines Fremdgerätes mit seriellm Interface an die Schleife

können Sie ein HP-IL/RS-232 Interface benutzen. Abb. 2.6 zeigt einen solchen Aufbau. Sie sehen ein Interface, das an ein Modem angeschlossen ist. Sie können aber auch jedes andere RS-232 Gerät anschließen.

Ein Fremdgerät durch ein Interface an die Interfaceschleife anzuschließen, erfordert einiges Wissen über die Interfacesignale, die notwendig sind, um Daten auf ein Gerät zu übertragen oder abzurufen. Um Ihnen eine bessere Vorstellung davon zu geben was im Aufbau eines Interfacesystems passiert, zeigt Abb. 2.7 die Verbindung zwischen einem HP82166A Konverter und einem Drucker mit einer Centronics Parallel-Schnittstelle.

Konverter wie der HP82166A erfordern auch ein gewisses Maß an Voreinstellung, um die richtigen Arbeitsparameter für das benutzte Gerät herzustellen. Dies umfaßt auch die Festlegung der Anzahl der Bits (8 oder 16) auf den Datenleitungen, die Anzahl und die Polarität (positiv oder negativ) der Handshake-Leitungen und die Definition des Statuswortes. Diese Voreinstellung hat normalerweise die Form von HP-IL Meldungen, die vom Controller zum Konverter geschickt werden. Im Falle des Interfaces in Abb. 2.7 muß eine "device dependent Listener 0"-Meldung (siehe Kapitel 6) an den Konverter geschickt werden, damit ein Kontrollregister mit dem Wert 100 110 10 beschrieben wird. Die Programmierung eines Konverters wie in Abb. 2.7 wird im Kapitel 3 eingehend erläutert.

Um eine größtmögliche Flexibilität zu erreichen, benöti-

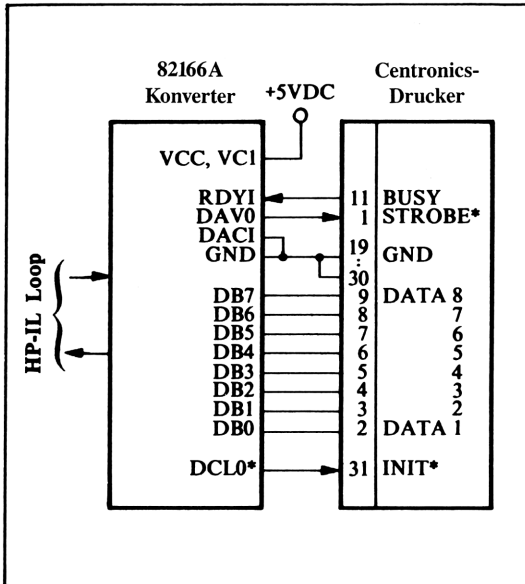


Abb. 2.7 Verbindung zwischen dem HP82166A Konverter und einem Drucker mit Centronics Parallel-Interface

gen Sie ein umfassenderes Verständnis der HP-IL Meldungen; nur damit können Sie einen Controller so programmieren, daß er mit einem Konverter zusammenarbeiten kann. Der Wissensstand, den Sie in den Kapiteln 3 bis 5 erhalten werden, ist sicherlich ausreichend, um Ihnen die Benutzung und Programmierung eines solchen Konverters zu erlauben. Wie auch immer, in vielen Fällen genügen Ihnen sehr wahrscheinlich die vorprogrammierten Funktionen eines Controllers, um Ihre Geräte ohne weiteres Wissen über die HP-IL Meldungen anzusteuern.

Bau eines Geräte-Interfaces

Wenn Sie selbst ein HP-IL Interface aufbauen wollen, entweder, um ein Fremdgerät an das Interface anschließen zu können, oder um ein Gerät nach Ihren eigenen Vorstellungen zu bauen, gibt es eine ganze Anzahl von Vorgehensweisen. Das Wissen, das Ihnen dieses Buch vermittelt, reicht dazu nicht aus. Die Informationen, die wir Ihnen hier geben,

sollen Ihnen zwar als Einführung und Überblick über die Struktur des HP-IL dienen, aber Sie werden zusätzlich ein Exemplar der "HP-IL reference specification" benötigen.

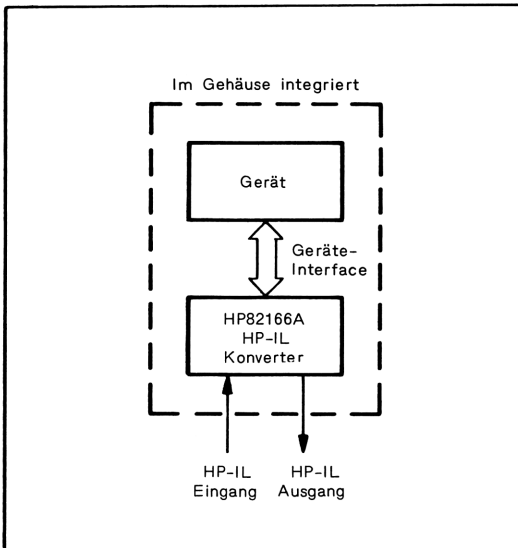


Abb. 2.8 HP-IL Konverter, der im Gerät eingebaut ist

Zunächst können Sie einen HP-IL Konverter in ein von Ihnen entwickeltes Gerät einbauen. Konverter wie der HP82166A können in ein Gehäuse integriert werden. Dies ist in Abb. 2.8 dargestellt. In diesem Fall muß Ihr Gerät nur (durch ein entsprechendes paralleles oder seriell Interface) mit dem Konverter kommunizieren. Der Konverter sorgt für alles, was notwendig ist, um Ihr Gerät an das HP-IL System anzuschließen.

Wenn Sie Ihr eigenes HP-IL Interface bauen wollen, müssen Sie eine Anzahl von Basisfunktionen in Ihre Konstruktion einfügen.

In Abb. 2.9 sind die Funktionen, die zum Aufbau eines HP-IL Interfaces notwendig sind, in zwei Gruppen eingeteilt: die Schleifen-Funktionen und die Geräte-Interface-Funktionen.

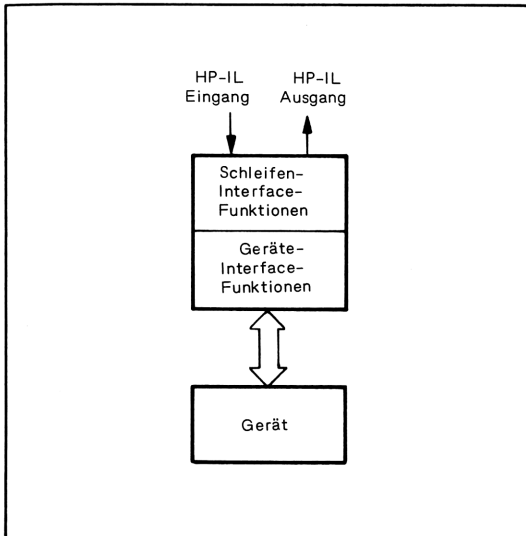


Abb. 2.9 Die Basisfunktionen eines HP-IL Interfaces

Die Schleifen-Interface-Funktionen sind die komplexesten. Wenn Sie eine Anzahl von Basisfunktionen in Ihre Konstruktion einfügen, müssen Sie für die korrekte Verar-

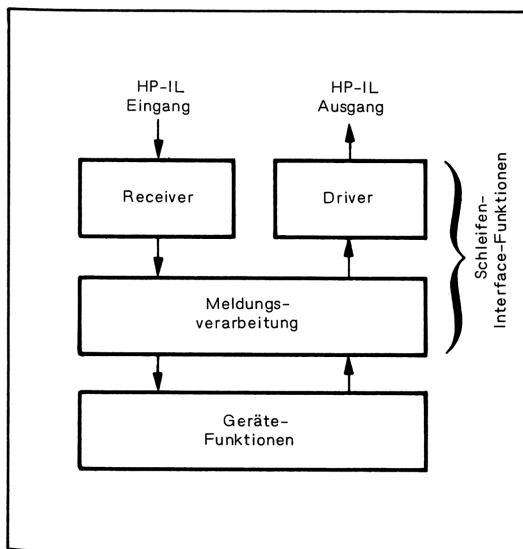


Abb. 2.10 HP-IL Schleifen-Interface-Funktionen

beitung aller Typen von HP-IL Meldungen Sorge tragen.

Abb. 2.10 ist eine einfache Darstellungsweise der Schleifen-Interface-Funktionen. HP-IL Meldungen werden von einem Baustein, der Receiver(16) heißt, empfangen und durch einen anderen Baustein, Driver(17) genannt, gesendet. Die Eigenschaften dieser Bausteine sind genauestens in den funktionalen und elektrischen Spezifikationen von HP dargestellt. Die mechanischen Spezifikationen von HP enthalten die Details und physikalischen Abmessungen der Stecker, die benutzt werden, um ein Gerät an die Interface-Schleife anzuschließen.

Der Baustein mit der Bezeichnung "Meldungsverarbeitung" in Abb. 2.10 ist der eigentliche Kern des Schleifen-Interfaces. Er ist verantwortlich für die Entscheidung, ob eine Meldung an ein bestimmtes Gerät adressiert ist, und wenn das der Fall ist, welche Signale an das Gerät gesendet werden müssen. Wenn das Gerät eine Meldung senden will, muß dieser Meldungsverarbeitungsblock die Meldung in das richtige Format umwandeln, sie abschicken und überprüfen, ob sie korrekt angekommen ist.

Ein Interface wie dieses kann aus einzelnen elektronischen Bauteilen, oder bequemer, aus einem einzigen LSI(18) Interface Controller-Baustein aufgebaut werden.

Die Erfordernisse eines solchen Interfaces werden detaillierter in den Kapiteln 3-7 und noch detaillierter in der "HP-IL reference specification" behandelt.

Kapitel 3

Anwendung eines Allzweck-Interfaces

zum HP-IL

In diesem Kapitel wollen wir besprechen, wie Sie mit Hilfe eines Allzweckadapters ein Gerät an ein bereits vorhandenes HP-IL "System" anschließen können. Dazu brauchen Sie ein größeres Wissen über die Schleife und die Schleifenoperationen, als wenn Sie einfach ein Gerät von Hewlett-Packard kaufen. Bevor wir an dieser Stelle einige Einzelheiten über die Schleifenoperation anführen wollen, lassen Sie uns auf die Eigenschaften eines Allzweckadapters für HP-IL zu sprechen kommen.

Der Adapter, von dem hier die Rede ist, erlaubt den Anschluß von Geräten mit einem 8-Bit parallelen Interface an das HP-IL. Andere Adapter ermöglichen dies für Geräte mit anderen Interfaces (16-Bit parallel, IEEE-488, RS-232 usw.).

Ein Allzweckadapter

Abb. 3.1 zeigt das Blockdiagramm eines Adapters, der benutzt werden kann, um Fremdgeräte an das HP-IL anzuschließen. Indem wir jeden Block erläutern, wollen wir erklären, wie der Adapter arbeitet.

Seriell/parallele Umwandlung

Informationen werden in einer bit-seriellen Art über die

Schleife geschickt. Weil viele Geräte für die Datenübertragung eine bit-parallele Methode anwenden, muß das Interface die Informationen, die es von der Schleife erhält, von seriellen in parallele umsetzen, und umgekehrt müssen die Informationen, die es an die Schleife abgibt, von parallelen in serielle umgesetzt werden.

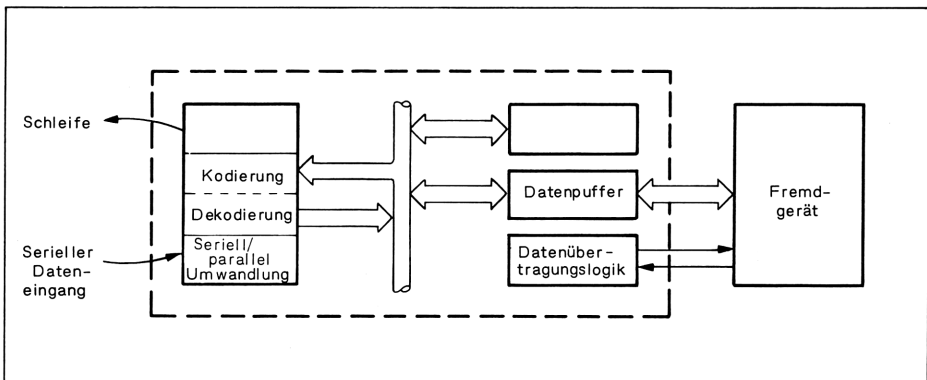


Abb. 3.1 Blockdiagramm eines HP-IL Allzweckadapters

Kodieren und Dekodieren von Meldungen

Wenn der Adapter eine Meldung erhält, die aus einer seriellen Bit-Folge besteht, muß er einige der Meldungs-Bitmuster dekodieren, um z.B. festzustellen:

- ist diese Meldung für mich bestimmt? (Adressendekodierung)
- soll die Meldung einfach weitergegeben werden?
- welcher Kategorie gehört die Meldung an?
- muß ich auf die Meldung antworten?

Wenn das Gerät eine Meldung auf die Schleife senden will, muß der Adapter in der Lage sein, die Information in das korrekte Meldungsformat zu kodieren, damit die Übertragung durch die parallel/serielle Umwandlungslogik stattfinden kann.

Datenpuffer

Weil die Datenübertragungsraten je nach Typ des Geräts stark variieren, und weil es unvernünftig wäre, wenn die einzelnen Geräte völlig von der Datenübertragungsrate auf der Schleife abhängig wären, beinhaltet ein guter Allzweck-adapter Datenpufferfähigkeiten. Dies verleiht den Geräten eine größtmögliche Flexibilität und vermeidet zugleich die Verringerung der Datenübertragungsraten auf der Schleife, wenn ein bestimmtes Gerät besonders langsam beim Empfangen oder Senden von Daten ist.

Geräte-Interface-Logik

Die Informationsübertragung zwischen dem Fremdgerät und dem Adapter muß relativ unabhängig von den Datenaustauschfunktionen zwischen Adapter und Schleife vonstatten gehen. Unterstellt, dieser Adapter habe die Aufgabe, Geräte zu verbinden, die ein Parallel-Interface besitzen, so zeigt Abb. 3.1 einen Adapter, der dem Fremdgerät ein 8-Bit Parallel-Interface zur Verfügung stellt. Ein Teil der Datenübertragungslogik ist darin eingeschlossen, um den Informationsaustausch zwischen dem Adapter und dem Gerät zu kontrollieren.

Kontroll-Logik

Dies ist eine allumfassende funktionelle Kategorie, die die Logik einschließt, die notwendig ist, um den zeitlichen Ablauf zu kontrollieren, die angemessene Antwort auf verschiedene Meldungskategorien festzulegen usw. Die Logik kann aus PLA-(Programmable Logic Array(19), ROM-(Read Only Memory) und ähnlichen Bausteinen bestehen. Dies sind die Logikbausteine, die dem Adapter "Intelligenz" verleihen. Da es für uns von geringerem Interesse ist, wie diese Intelligenz aufgebaut ist, wollen wir es dabei belassen, daß eine angemessene Logik zur Verfügung steht, um die erwünschten Funktionen zu erfüllen. Kapitel 4 wird tiefer in dieses Thema eindringen, um Ihnen zu zeigen, wie diese Logik aufgebaut werden kann.

Das Gerät als Listener

Stellen Sie sich vor, daß das Gerät, das Sie per Adapter an das HP-IL anschließen wollen, ein Thermodrucker ist. Bevor wir näher darauf eingehen, wie der Drucker mit der Schleife zusammenarbeitet, wollen wir die Eigenschaften dieses Gerätes und seine Rolle als Teil der Schleife bestimmen.

Noch bevor wir etwas näheres über den Drucker wissen, wollen wir festhalten, daß er ein reines Ausgabegerät ist. Das System (d.h. die anderen Geräte auf der Schleife) senden Informationen an den Drucker - der Drucker sendet an die anderen Geräte auf der Schleife keinerlei Informationen. Deshalb weiß man, daß der Drucker primär die Rolle des Listeners übernimmt. Als Listener hat er eine bestimmte Verantwortung für das System, festgelegte Antworten auf bestimmte Meldungen und bestimmte Meldungsfolgen, an die er sich halten muß. Eine Darstellung aller Meldungen, die zwischen dem Drucker und dem HP-IL System (per Adapter) ausgetauscht werden, erklärt die Regeln, die die Schleifenoperation ausmachen. Sie wird auch einen Einblick in die Logik geben, die Ihr Drucker haben muß, um in der Lage zu sein, mit dem System zu arbeiten.

Aber lassen Sie uns zunächst einige Eigenschaften des Druckers nennen, denn diese bestimmen in gewissem Umfang, wie er in seiner Rolle als Listener agiert.

Der Drucker soll in einen Standby-Zustand mit niedriger Leistungsaufnahme versetzt werden können, indem der System-Controller dazu einen Befehl erteilt. Dies ermöglicht es Ihnen, den Drucker bei unbeaufsichtigten Anwendungen zu benutzen, bei denen der Drucker nur zeitweise für die Aufnahme von Meßwerten aktiviert wird. Natürlich muß der Drucker, bevor er druckt, auf einen Power-Up(20)-Befehl des Systems antworten. Der Drucker muß in seinen Fähigkeiten flexibel sein. Zum Beispiel sollte er mehr als einen Zeichensatz zur Verfügung haben; er sollte in der Lage sein, verschiedene Zeilenabstände zu benutzen usw. Dies bedeutet, daß er in einem bestimmten Ausmaß programmierbar ist. Hierzu muß er zusätzlich zu den Datenzeichen, die er drucken soll, auch andere Informationen vom System erhalten.

Der Drucker muß in der Lage sein, das System über seinen Status zu informieren. Er muß z.B. melden, wenn ihm das Pa-

pier ausgeht, während er gerade druckt, er muß mitteilen, wie sein gegenwärtiger Druckmodus ist usw.. In den meisten Systemen sollten alle Geräte in der Lage sein, zumindest die minimalsten Statusinformationen zu liefern. Diese Fähigkeit beinhaltet, daß ein Gerät ebenfalls in der Lage ist, die Rolle des Talkers zu übernehmen, denn ein Listener kann nur Daten aufnehmen, aber keine senden. So muß auch ein reines Ausgabegerät wie ein Drucker in der Lage sein, gelegentlich die Rolle des Talkers zu übernehmen.

Nun gehen wir über zu den Meldungen, mit denen ein Drucker während der Schleifenoperationen konfrontiert ist. Nicht alle HP-IL Meldungen werden in den folgenden Abschnitten geschildert; eingehen wollen wir nur auf diejenigen, auf die der Drucker antwortet. Alle anderen Meldungen, die der Adapter/Drucker erhält, werden einfach an das nächste Gerät auf der Schleife weitergegeben.

Meldungen der Command(21)-Klasse

Die Meldungen aus der Command-Klasse werden dazu benutzt, die Ausgangsbedingungen für die Operationen auf der Schleife herzustellen, die Arbeitsmethode und die Rollen der Geräte auf der Schleife zu verändern, und ganz allgemein, um die geregelte Arbeitsweise und Kontrolle über das Schleifensystem aufrecht zu erhalten. Nachfolgend finden Sie die Meldungen der Command-Klasse, die ein Drucker wie z.B. der oben erwähnte von einem Interface-Adapter erhalten kann.

- Interface Clear (IFC)
- Device Clear (DCL)
- Loop Power-Down (LPD)
- Unlisten (UNL)
- Talk Address (TAD)
- Untalk (UNT)

Bevor wir diese Meldungen jedoch eingehender besprechen, wollen wir eine einheitliche Darstellungsweise für die Erläuterung der gesamten Schleifenoperationen vereinbaren.

Schleifen-Nomenklatur

In den folgenden Beispielen werden die Geräte auf der Schleife, denen aktive Rollen wie Talker oder Listener zugewiesen wurden oder die eine Schleifenoperation ausführen, durch Quadrate mit durchgezogenen Kanten dargestellt. Quadrate mit gestrichelten Kanten repräsentieren andere Geräte auf der Schleife, denen entweder bis jetzt noch keine Rolle zugewiesen wurde, oder deren Anwesenheit auf der Schleife für unser Thema unerheblich ist.

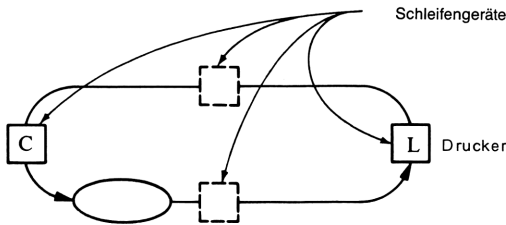


Abb. 3.2

In Abb 3.2 ist das Gerät, das durch das linke Quadrat symbolisiert wird, der System- oder Schleifen-Controller(C) und das Gerät im rechten Quadrat ist ein aktiver Listener(L)

Meldungen, die über die Schleife gehen, werden durch Ovale dargestellt. Die Meldung, die in Abb. 3.3 gerade über die

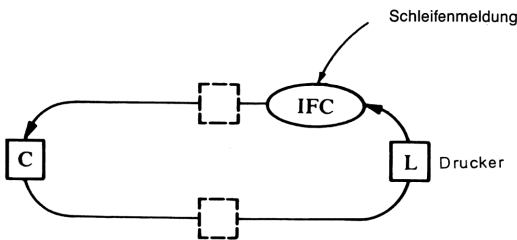


Abb. 3.3

Schleife geht, ist die Interface Clear (IFC)-Meldung. Die Bedeutung der Symbole unserer Abbildungen kann mit dem tieferen Eindringen in die Materie erweitert werden, aber die Erweiterung der Nomenklatur ist entweder direkt ersichtlich oder wird an gegebener Stelle erklärt.

Interface Clear (IFC)

Die IFC-Meldung ist gleichbedeutend mit einer Meldung, die das System in den Ausgangszustand versetzt. Sie kann nur vom System-Controller ausgesendet werden, der gleichzeitig auch der aktive Controller der Schleife sein kann, aber nicht sein muß. Die Meldung erlaubt es dem System-Controller die Kontrolle der Schleife zu jeder Zeit zu übernehmen. Normalerweise sendet der System-Controller eine Serie von IFC-Meldungen über die Schleife, wenn das System eingeschaltet wird, um alle Geräte auf der Schleife in einen inaktiven oder "Ruhe"zustand zu versetzen.

Wenn der Interface-Adapter des Druckers die IFC-Meldung erhält und dekodiert, löscht er seinen aktiven Talker- oder Listener-Status, sofern er sich in einem dieser Zustände befunden hat, als die Meldung empfangen wurde. Dadurch geht er in einen Ruhezustand und wartet darauf, daß der Schleifen-Controller ihm eine neue Rolle zuweist.

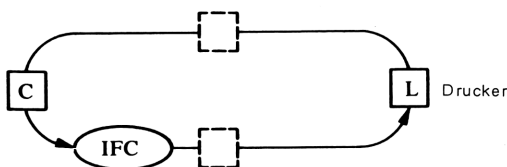


Abb. 3.4

Beachten Sie, daß die Interface Clear-Meldung keine direkte Auswirkung auf den Drucker selbst hat. Nur der Interface-Adapter ist betroffen.

Weil die IFC-Meldung an alle Geräte auf der Schleife gerichtet ist, kann es eine längere Zeitspanne in Anspruch nehmen, bis die Meldung den System-Controller wiedererreicht, denn jedes Interface wartet, bis es sich selbst gelöscht hat, bevor es die Meldung weitergibt. Um diesen Vorgang zu beschleunigen, muß jedes Interface die Meldung unmittelbar weitergeben und sich dann in den Ruhezustand versetzen. Wenn die IFC-Meldung wieder beim System-Controller ankommt, bedeutet dies dann nicht notwendigerweise, daß schon alle Interfaces auf der Schleife gelöscht sind.

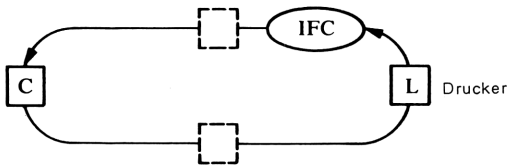


Abb. 3.5

In dieser Situation sendet der System-Controller die Ready For Command (RFC)-Meldung, die wir nun kurz erklären wollen. Die RFC-Meldung wird von jedem Interface auf der Schleife festgehalten, bis sein eigener Löschvorgang beendet ist.

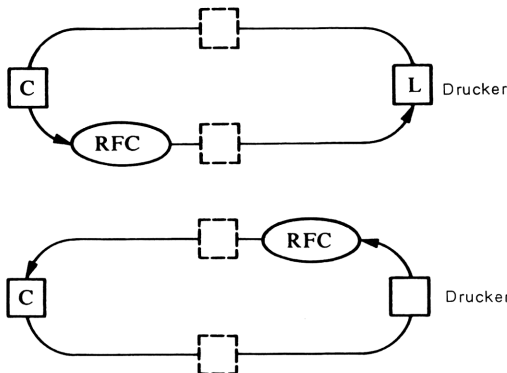


Abb. 3.6

Die gleiche Abfolge (d.h. den Befehl weitergeben und die RFC-Meldung festhalten, bis das jeweilige Gerät bereit ist, den folgenden Befehl bzw. eine andere Meldung zu erhalten) wird für alle Meldungen der Command-Klasse benutzt, um die Bearbeitung von Befehlen auf der Schleife für alle Geräte zu beschleunigen. Andere Meldungen werden von den Geräten einfach festgehalten, bis sie bereit sind, eine neue Meldung zu empfangen. Der Controller gehorcht der einfachen Regel, immer dann eine RFC-Meldung auszusenden, wenn eine Befehlsmeldung zu ihm zurückkehrt. Dies wird im Abschnitt über die Meldungen der Ready(22)-Klasse näher beschrieben.

Device Clear (DCL)

Die DCL-Meldung kann vom Schleifen-Controller dazu benutzt werden, ein Gerät (im Gegensatz zu dem Interface des Gerätes) zurückzusetzen. Beachten Sie, daß alle Interfaces der Geräte auf der Schleife in der Lage sein müssen, auf die IFC-Meldung zu antworten, denn dadurch übernimmt der System-Controller die Kontrolle über die Schleife. Nicht alle Geräte müssen jedoch fähig sein, auf die DCL-Meldung zu reagieren. Weiterhin ist die Handlung, die ein bestimmtes Gerät als Antwort auf die DCL-Meldung ausführt, gänzlich dem Gerät selbst überlassen. Die einzige Aufgabe, die das Gerät hat, ist es, die DCL-Meldung an das nächste Gerät auf der Schleife weiterzugeben. Unser Drucker soll auf die DCL-Meldung antworten. Wenn der Interface-Adapter des Druckers diese Meldung empfangen und dekodiert hat, muß er eine Information an die Elektronik des Druckers senden, um das Gerät zu löschen oder zurückzusetzen. Dies ist in Abb.3.7 dargestellt.

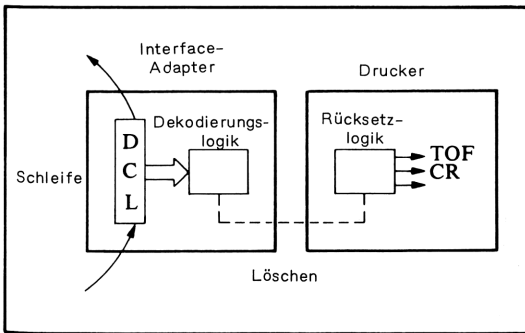


Abb. 3.7 Die Wirkung der DCL-Meldung

Wenn der Interface Adapter die DCL-Meldung dekodiert hat, schickt er das Löschesignal an den Drucker. Innerhalb des Druckers löst dieses Signal Dinge aus wie z.B.: Papiertransport zum Blattanfang, der Druckmechanismus bewegt sich in seine Ausgangsposition (Wagenrücklauf), alle vorher erhaltenen Daten des Datenpuffers werden gelöscht usw. Alle diese Vorgänge sind vollständig von den Geräten selbst abhängig und haben keinerlei Bedeutung für die Schleife oder für den Interface-Adapter des Druckers.

Genau wie bei der Interface Clear-Meldung dürfen die Geräte auf der Schleife nicht darauf warten, daß die vorhergehenden Geräte ihre Löschoperationen beendet haben, bevor sie die DCL-Meldung weitergeben. Stattdessen soll jedes Gerät die DCL-Meldung weitergeben und dann damit fortfahren, die Löschofunktion zu vollziehen. Der Controller läßt der DCL-Meldung die Ready For Command (RFC)-Meldung folgen, um festzustellen, wann alle Geräte die DCL-Meldung ausgeführt haben.

Listen Address (LAD)

Die LAD-Meldung wird vom Schleifen-Controller gesendet, um ein bestimmtes Gerät zum aktiven Listener zu machen. Damit ein Gerät von anderen Geräten auf der Schleife Daten erhalten kann, muß es zuerst als Listener adressiert werden. Da es vorkommen kann, daß der Controller dieselben Informationen an mehr als ein Gerät senden will, kann die Schleife zur gleichen Zeit mehr als einen aktiven Listener haben. Aber da die LAD-Meldung eine Adresse enthält, ist sie nur an jeweils ein Gerät gerichtet. Wenn es gleichzeitig mehr als einen Listener geben soll, muß der Controller eine eigene LAD-Meldung für jedes Gerät senden. Jeder LAD Meldung folgt natürlich eine RFC-Meldung.

Es gibt noch eine andere Listen Address-Meldung, die in der HP-IL Spezifikation My Listen Address (MLA) genannt wird. Diese Meldung ist die LAD-Meldung, die eine Adresse enthält, die mit der Adresse des Geräts übereinstimmt, das die LAD-Meldung empfängt. Stellen Sie sich zum Beispiel vor, daß dem Drucker-Interface eine Adresse von 2 zugewiesen wurde (dies kann durch eingebaute Schalter oder durch eine programmierbare Methode geschehen).

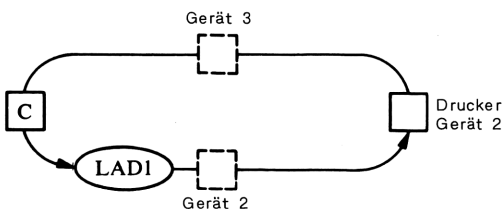


Abb. 3.8.1

Wenn die LAD-Meldung empfangen wird und eine Adresse von 1 enthält, dann gibt das Interface die LAD-Meldung einfach an das nächste Gerät weiter. In diesem Fall war die LAD-Meldung nicht an das Drucker-Interface gerichtet, sondern an das Gerät, das die Rolle des Listeners übernimmt.

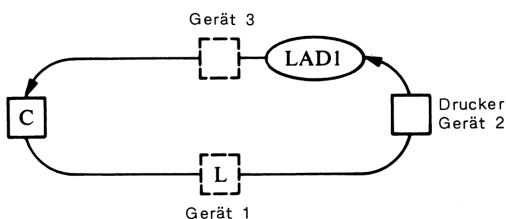


Abb. 3.8.2

Wenn eine LAD-Meldung empfangen wird, die die Adresse 2 enthält, so dekodiert das Drucker-Interface sie als My Listen Address (MLA) und unternimmt die Schritte, die notwendig sind, um sich selbst auf die Listener-Rolle vorzubereiten. (Beachten Sie, daß Gerät 1 ebenfalls seine Listener-Rolle beibehält, wie in Abb. 3.9 dargestellt.)

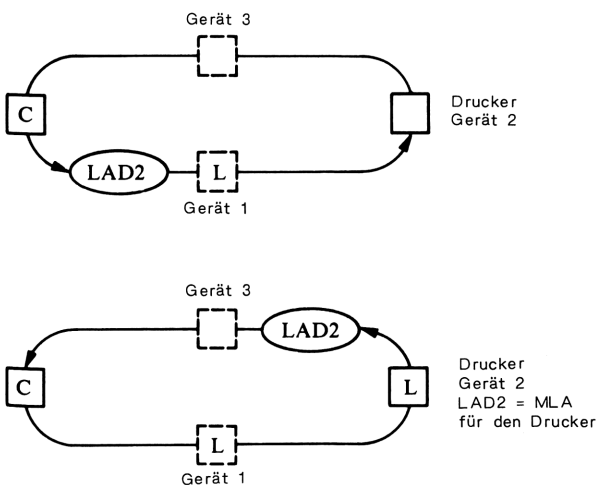


Abb. 3.9

Die Unterscheidung zwischen LAD und MLA wird einfach deshalb gemacht, um klarzustellen, daß die verschiedenen Handlungen, die die Interface-Logik auszuführen hat, davon abhängen, ob die Adresse übereinstimmt oder nicht.

Wenn das Drucker-Interface einmal ein aktiver Listener ist, kann es die Daten, die von der Schleife kommen, annehmen. Diese Daten können aus den Zeichen bestehen, die gedruckt werden sollen, aus Kontrollinformationen für den Drucker, oder Programminformationen für den Interface-Adapter. Die Interpretation des Inhalts der Datenmeldungen ist die Aufgabe des Gerätekonstruktors (in diesem Fall des Adapter- oder Druckerkonstruktors).

Unlisten (UNL)

Der UNL-Befehl dient dazu, alle adressierten Listener in einen Ruhezustand zurückzusetzen. So wird jedes Gerät herausgesucht, das vorher durch die LAD-Meldung als Listener adressiert wurde.

Beachten Sie, daß mit der LAD-Meldung nur jeweils ein Listener adressiert wird, während mit der UNL-Meldung alle Listener gleichzeitig angesprochen werden. Der Schleifen-Controller benutzt deshalb diesen Befehl jedesmal, wenn er einen neuen Listener (oder mehrere neue Listener) für eine Datenübertragung adressiert.

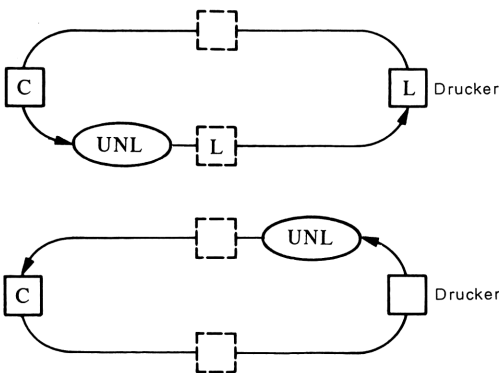


Abb. 3.10

Wenn das Drucker-Interface die UNL-Meldung erhält, versetzt es sich in den Ruhezustand, wobei es zwar auf der Schleife verbleibt, aber keine aktive Rolle mehr spielt.

Loop Power Down (LPD)

Die LPD-Meldung wird vom Schleifen-Controller benutzt, um die Geräte auf der Schleife in einen Zustand mit niedriger Leistungsaufnahme (Standby) zu versetzen, um Leistung einzusparen. Diese Eigenschaft kann in Systemen, in denen Leistungseinsparung wichtig ist, sehr sinnvoll sein, so z.B. in unbeaufsichtigten Meßstationen, bei denen das System nur während vorherbestimmter kurzer Zeitintervalle voll in Funktion sein muß. Wenn z.B. der Drucker dazu benutzt wird, Messungen zu protokollieren, die nur alle paar Stunden gemacht werden sollen, ist es günstig, ihn in einen Ruhezustand zu versetzen, wenn er gerade nicht gebraucht wird.

Der LPD-Befehl ist ein universeller Befehl. Wenn der Schleifen-Controller ihn daher sendet, ist er an alle Geräte auf der Schleife gerichtet. Trotzdem müssen nicht alle Geräte die Fähigkeit haben, sich in einen Standby-Zustand versetzen zu können. Nur diejenigen Geräte, die einen Zustand mit niedriger Leistungsaufnahme eingebaut haben, werden dem Befehl antworten; andere Geräte geben die LPD-Meldung einfach auf der Schleife weiter.

Unser Drucker hat die Standby-Fähigkeit und ist mit einem Schalter mit drei Schaltstellungen ausgerüstet: AUS - STANDBY - EIN. Wenn der Schalter auf AUS steht, werden der Drucker und der Interface-Adapter vollständig von der Stromversorgung getrennt. Wenn er zu diesem Zeitpunkt an die Schleife angeschlossen wäre, würde die Schleife aktionsunfähig, denn das Interface des Druckers wäre nicht in der Lage, Meldungen auf der Schleife weiterzugeben. Wenn der Schalter auf EIN steht, ist der Drucker voll funktionsfähig, aber er wird auf die LPD-Meldung nicht antworten; er gibt die Meldung einfach weiter. Steht der Schalter auf STANDBY so ist der Drucker voll funktionsfähig und kann auf die LPD-Meldung antworten. Sobald eine LPD-Meldung empfangen wird, wird der Drucker in den Standby-Zustand mit niedriger Leistungsaufnahme versetzt. Nur die Stromkreise auf dem Interface-Adapter, die notwendig sind, um auf eine

nachfolgende Meldung auf der Schleife zu antworten, werden noch mit Leistung versorgt.

Wenn der Schalter des Druckers auf STANDBY steht und der Interface Adapter eine LPD-Meldung erhält, bereitet sich das Interface darauf vor, in den Ruhezustand überzugehen. Die LPD-Funktion wird jedoch nicht ausgelöst, bevor eine Ready For Command (RFC)-Meldung empfangen wird. Sobald das Interface die nachfolgende RFC-Meldung erhält, sorgt es dafür, daß jede unnötige Stromversorgung unterbleibt.

Beachten Sie, daß die Art, wie das Gerät auf die Meldungsabfolge Befehl - RFC reagiert, sich von anderen Abfolgen, unterscheidet, die wir bereits besprochen haben. In den zuvor erläuterten Situationen würde das Gerät die Befehlsmeldung sofort an das nächste Gerät weitergeben und dann mit der Ausführung des Befehls beginnen. Bei der LPD-Meldung jedoch wartet das Gerät, bis es die nachfolgende RFC-Meldung erhält, bevor es den LPD-Befehl ausführt.

Für diesen Unterschied gibt es zwei Gründe. Wahrscheinlich werden Sie die Schleife für eine längere Periode in den Ruhezustand versetzen wollen. Deswegen gibt es keinen Grund für den Versuch, die Operation für Bruchteile von Sekunden zu beschleunigen. Zum Zweiten gibt es keinen speziellen Befehl oder eine Meldung, die bewirkt, daß die Geräte wieder voll mit Leistung versorgt werden. Stattdessen müssen sich Geräte im Ruhezustand selbst wieder aktivieren, sobald sie eine Meldung von der Schleife erhalten. Wenn das Gerät in dem Moment, in dem die RFC-Meldung ankommt, den LPD-Befehl schon vollständig ausgeführt hätte, würde es sich selbst sofort wieder voll einschalten. Die Geräte müssen also warten, bis sie die nachfolgende RFC-Meldung empfangen, bevor sie LPD ausführen.

Dieses EIN/AUS-System ist einfach und effektiv, aber es bedeutet, daß sich alle Geräte auf der Schleife, die in der Lage sind, der LPD-Meldung zu antworten, im gleichen Zustand befinden; man kann nicht getrennt von den anderen Geräten eine einzelne Einheit aus- oder einschalten.

Ready(23)-Meldungen

Die Befehlsmeldungen, die eben beschrieben wurden, können nur vom Controller gesendet werden. Die Ready-Meldungen, die jetzt besprochen werden, können sowohl vom Schleifen-

Controller als auch von einem Gerät, das aktiver Talker ist, gesendet werden. Diese Meldungen werden allgemein gebraucht, um die Übertragung von Befehlen und Daten zwischen den Geräten auf der Schleife zu koordinieren und zu synchronisieren. Im folgenden sind die Ready-Meldungen, auf die ein Drucker reagieren kann, aufgelistet.

- Ready For Command (RFC)
- Send Status (SST)
- End of Transmission OK (ETO)
- End of Transmission Error (ETE)

Ready For Command

Die RFC-Meldung wird vom Schleifen-Controller benutzt, um festzustellen, ob eine direkt vorhergegangene Befehlsmeldung ausgeführt wurde. Um die Notwendigkeit dieser Meldung zu erklären, muß das Spektrum unserer Darstellung erweitert werden. Normalerweise geht nur jeweils eine Meldung über die Schleife. Der Controller oder Talker, der die Meldung gesendet hat, wartet, bis er die Meldung erfolgreich zurück-erhält, bevor er eine neue Meldung sendet. Normalerweise gibt ein Gerät eine Meldung solange nicht weiter, bis es in der Lage ist, selbst eine neue Meldung zu empfangen. Diese Technik stellt sicher, daß Talker und Controller eine Meldung nicht schneller senden können, als sie andere Geräte auf der Schleife verarbeiten können.

Wenn es auf der Schleife eine Anzahl von langsamen Geräten gibt, und da jedes Gerät die Meldung zurückhält, bis es eine neue empfangen kann, kann die Schleifengeschwindigkeit sehr niedrig werden. Datenmeldungen sind normalerweise für nur ein Gerät bestimmt, und andere Geräte geben sie sofort weiter. Dadurch kann ein zufriedenstellender Durchsatz erreicht werden.

Befehlsmeldungen sind jedoch oft zur gleichen Zeit an alle Geräte auf der Schleife gerichtet. Um die Ausführung dieser Befehlsmeldungen zu beschleunigen müssen, die Geräte die Meldung sofort weitergeben und erst dann mit der Ausführung des Befehls beginnen. So können alle Geräte auf der Schleife Befehle mehr oder weniger parallel zueinander ausführen.

Wenn der Controller die Befehlsmeldung nach ihrem Durch-

lauf auf der Schleife wiedererhält, zeigt die Meldung nicht an, daß der Befehl von allen Geräten ausgeführt wurde. Es wird damit nur angezeigt, daß er von allen Geräten empfangen wurde. Jetzt schickt der Controller die RFC-Meldung. Diese Meldung wird von jedem Gerät zurückgehalten und erst dann auf der Schleife weiterübertragen, wenn der vorher eingegebene Befehl vollständig erfüllt wurde. So hat der Controller erst die Sicherheit, daß der Befehl vollständig ausgeführt wurde, wenn die RFC-Meldung zurückgekommen ist.

Send Status (SST)

Die SST-Meldung wird vom Controller benutzt, um den adressierten Talker zu veranlassen, Statusinformationen an den Controller zurückzusenden. Wie Sie sich erinnern, begannen wir zu Anfang der Beschreibung der Eigenschaften des Druckers mit der Aussage, daß er gelegentlich in der Lage sein muß, die Rolle des Talkers zu übernehmen, um Statusinformationen auf die Schleife zu senden. Das Schleifenprotokoll besagt, daß man, um Informationen zu erzeugen und auszusenden, die Rolle des Talkers annehmen muß. Es tut hierbei nichts zur Sache, daß der Drucker in 99% der Fälle nur ein Listener ist. Die Aussage, daß ein Gerät meist Listener, gelegentlich Talker ist, wird nach der Erläuterung der Send Status-Meldung näher erklärt.

Die SST-Meldung wird benutzt, um Informationen von einem Gerät (unserer Drucker/Interfacekombination) anzufordern. Die Information, die der Controller anfordert, ist normalerweise eine gerätespezifische Information. Der Controller kann z.B. Informationen anfordern wie:

- Druckst du gerade?
- Bist du gerade beim Papiervorschub?
- Ist dein Puffer leer?
- Bist du in irgendeinem spezifischen- z.B. Escape(24)-
Modus?

Die SST-Meldung beinhaltet keine Geräteadreibinformation. Die Meldung ist immer an den aktiven Talker gerichtet. Weil es auf der Schleife zur gleichen Zeit immer nur einen aktiven Talker geben kann, gibt es keine Probleme festzustellen, für wen die SST-Meldung bestimmt ist. Wenn Geräte,

die nicht der aktive Talker sind, die SST-Meldung erhalten, geben sie sie einfach auf der Schleife weiter. Wenn es auf der Schleife keinen aktiven Talker gibt, erhält der Controller die unveränderte SST-Meldung einfach zurück. Dies sollte natürlich nie passieren, denn der Controller muß immer wissen, welche Rollen die verschiedenen Geräte auf der Schleife spielen, und schickt deshalb auch keine SST-Meldung aus, wenn es keinen aktiven Talker auf der Schleife gibt.

Nehmen Sie an, dem Drucker-Interface sei die Rolle des Talkers zugewiesen worden. Wenn er die SST-Meldung erhält, gibt er sie nicht, wie die meisten Meldungen, weiter an das nächste Gerät auf der Schleife. Stattdessen ersetzt er diese Meldung durch seine eigene Information - ein oder mehrere Bytes Statusinformation - und schickt diese Information auf die Schleife.

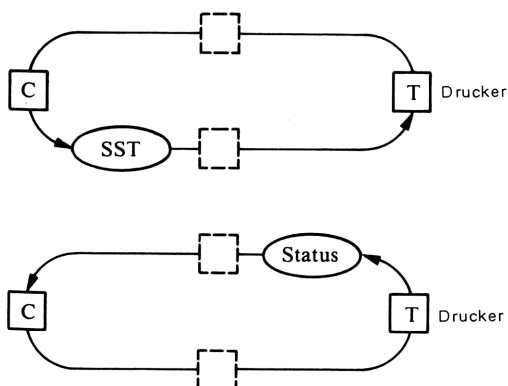


Abb. 3.11

Wenn die anderen Geräte auf der Schleife die Statusbytes des Druckers erhalten, geben sie sie einfach weiter, bis sie wieder beim Controller ankommen. Der Controller erwartet nun natürlich nicht, daß die SST-Meldung zurückkommt, sondern er erwartet das Statusbyte oder die Statusbytes des Druckers.

In der HP-IL Spezifikation gibt es einige allgemeine Richtlinien und Vorschläge für das Format der Statusbytes, die die Geräte an den Controller zurückschicken. Ein tieferes Eindringen in die Materie ist an dieser Stelle nicht notwendig, sondern wird in Kapitel 4 vorgenommen.

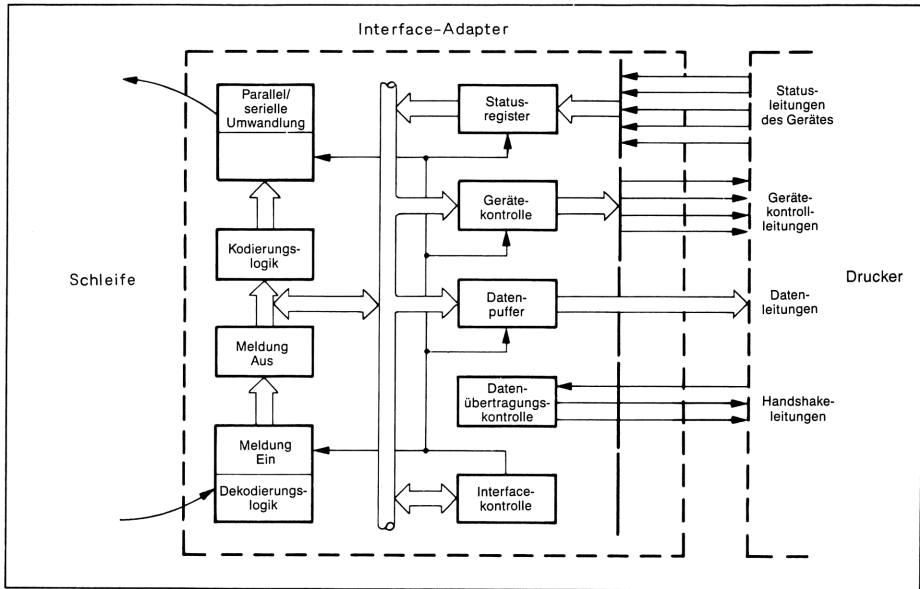


Abb. 3.12 Ausführliches Blockdiagramm eines Drucker-Interface-Adapters

Abb. 3.12 zeigt ein genaueres Blockdiagramm eines Interfaces, das einen Drucker ansteuern kann. Sie werden feststellen, daß wir einige Blöcke hinzugefügt haben, die in Abb. 3.1 noch nicht aufgeführt waren. Weil das Interface auf die SST-Meldung antworten soll, ist ein Statusregister hinzugefügt worden. Dieses Statusregister speichert die Statusinformationen des Druckers und kann auch Informationen über den Status des Interfaces selbst beinhalten. Zwei Pufferregister sind hinzugefügt worden: eines für hereinkommende Meldungen, eines für abgehende.

Abb. 3.13 stellt ein Blockdiagramm dar, das zeigen soll, was passiert, wenn das Interface die SST-Meldung erhält.

Die SST-Meldung wird nach der Dekodierung einfach festgehalten. Dann schickt das Interface ein Datenbyte (DAB), das die Statusinformation enthält, über die Schleife an den Controller. (Wir wollen die DAB's einige Abschnitte später behandeln)

Das Interface kann beliebig viele Bytes an Statusdaten

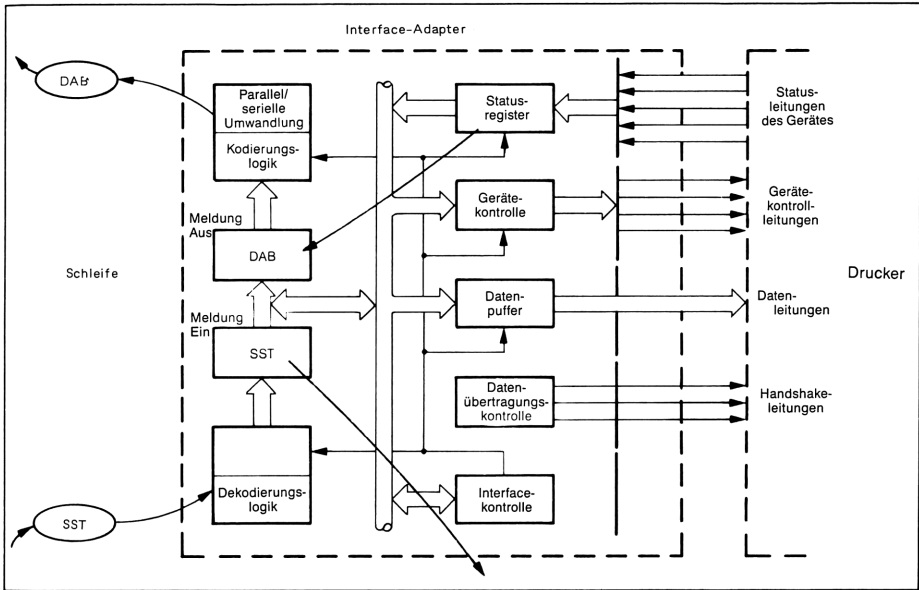


Abb. 3.13 Wirkung der SST-Meldung

schicken. Sie werden sich erinnern, daß es jetzt der aktive Talker auf der Schleife ist.

Sobald der Controller das Statusbyte oder mehrere Statusbytes vom Drucker erhält, verarbeitet er sie in einer dem System angemessenen Art und Weise, und gibt dann das Statusbyte auf der Schleife zum Drucker zurück. Dies ist das Signal für den Drucker, daß er das nächste Byte senden kann.

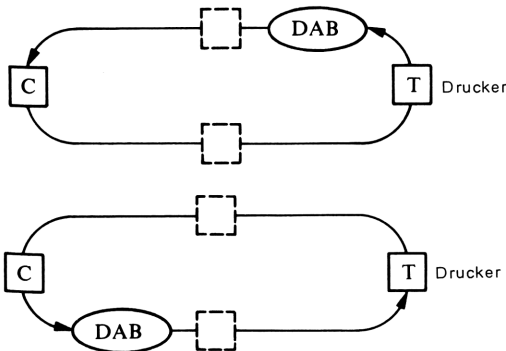


Abb. 3.14

Lassen Sie uns einen letzten Punkt bezüglich der SST-Meldung anführen. Die gerade behandelte Befehlsfolge ist bekannt als "serielle Abfrage". Dies ist eine Technik, die vom Controller benutzt werden kann, um sukzessive den Status eines jeden Gerätes auf der Schleife zu erhalten. Er adressiert hierzu jedes Gerät einzeln als Talker, schickt ihm die SST-Meldung, schaltet es als Talker wieder ab und fährt bei dem nächsten Gerät auf der Schleife damit fort. Eine andere Art der Abfrage, die "parallele Abfrage", wird in späteren Kapiteln besprochen.

End Of Transmission (EOT)

Nachdem die Drucker/Interfacekombination die Statusbytes geschickt und wiederempfangen hat, muß sie eine spezielle Meldung an den Controller schicken, um diesen darüber zu informieren, daß der Drucker seine Rolle als aktiver Talker wieder beendet hat. Die Meldung, die hierzu benutzt wird, ist die End Of Transmission (EOT)-Meldung.

Sie erinnern sich daran, daß der Controller die Statusbytes, die er vom Drucker erhalten hat, weiter auf der Schleife an diesen zurücksendet.

Wenn er zuletzt die EOT-Meldung erhält, gibt er diese Meldung jedoch nicht weiter. Da die EOT-Meldung die Beendigung der Rolle des Druckers als aktiven Talker signalisiert, gibt der Controller stattdessen nun einen neuen Befehl heraus.

Er sendet dann beispielsweise einen Untalk (UNT)-Befehl,

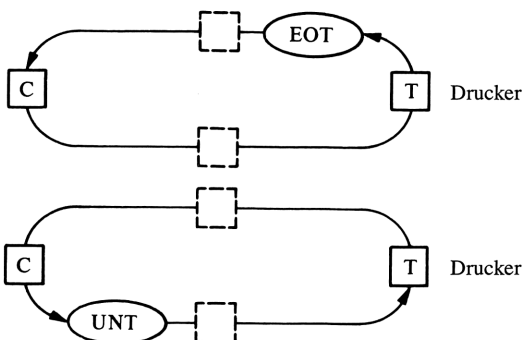


Abb. 3.15

um die aktive Talkerrolle des Druckers aufzuheben (der UNT-Befehl wird später erläutert).

Es gibt zwei Formen der End Of Transmission-Meldung.: End Of Transmission OK (ETO) und End Of Transmission Error (ETE). Diese Meldungen werden immer vom aktiven Talker abgeschickt und sind an den Schleifen-Controller gerichtet. Wie aufgezeigt wurde, gehen diese Meldungen nicht um die ganze Schleife, sondern werden festgehalten, wenn sie den Controller erreichen. Wenn die Datenbytes zum Controller zurückkommen, kann er sie daraufhin untersuchen, ob sie mit den ausgesendeten Bytes identisch sind. Wenn dies zutrifft, sendet der Talker die ETO-Meldung als Abschluß. Wenn die Bytes falsch zurückkommen, muß der Talker die Übertragung mit der Fehlermeldung beenden: ETE. Der Controller reagiert dann entsprechend auf die Fehleranzeige, indem er z.B. den Prozeß von neuem beginnt.

Beachten Sie, daß es nicht erforderlich ist, daß Geräte auf der Schleife die zurückkommenden Bytes auf Fehler überprüfen. Wenn sie keine Fehlerüberprüfung vornehmen, dürfen sie auch keine ETE-Meldung senden. Gut konstruierte Systeme sollten die Fehlerüberprüfung selbstverständlich mit einschließen. Beachten Sie auch, daß die Antwort eines Controllers auf die Fehleranzeige nicht weiter spezifiziert ist. Die genauen Korrekturprozeduren sind gänzlich dem Konstrukteur des Systems überlassen.

Die Talker-Befehle

Wie schon früher dargestellt, muß der Drucker die Rolle des Talkers übernehmen, wenn er Statusinformationen auf die Schleife senden will. Es gibt zwei Befehle, die mit der Rolle des Talkers verbunden sind, und sie entsprechen den beiden Listener-Befehlen, die in den vorherigen Abschnitten behandelt wurden - die Listen Address (LAD)- und die Unlisten (UNL)-Meldung. Die beiden Talkerbefehle lauten: Talk Address (TAD) und Untalk (UNT).

Talk Address (TAD)

Die Talk Address-Meldung wird vom Schleifen-Controller gesendet, um ein bestimmtes Gerät zum aktiven Talker zu

machen. Damit ein Gerät in der Lage ist, Informationen auf die Schleife zu schicken, muß es zunächst als Talker adressiert werden. Während es zur gleichen Zeit mehrere aktive Listener auf der Schleife geben kann, gibt es immer nur einen aktiven Talker.

Wie der Listen Address-Befehl (LAD) beinhaltet auch der TAD-Befehl eine Adresse und ist an ein einzelnes spezifisches Gerät auf der Schleife gerichtet. Wie bei der LAD-Meldung gibt es auch beim TAD-Befehl eine bestimmte Form, die als My Talk Address (MTA) definiert ist. Dies ist einfach die TAD-Meldung, die die gleiche Adresse enthält, wie das Gerät, das sie empfängt. Stellen Sie sich vor, daß unserem Drucker-Interface eine Adresse von 2 zugewiesen worden ist.

Wenn eine TAD-Meldung mit der Adresse 1 empfangen wird, gibt das Drucker-Interface sie einfach an das nächste Gerät weiter.

In diesem Fall war die TAD-Meldung nicht an das Drucker-Interface gerichtet, sondern an Gerät 1, das die Rolle des Talkers übernimmt. In Abb. 3.16 dekodiert Gerät 1 die TAD-

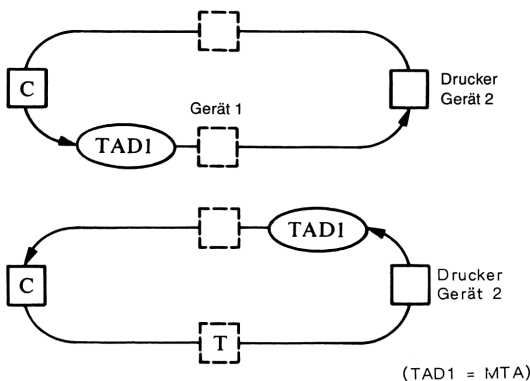


Abb. 3.16

Meldung als My Talk Address (MTA). Wenn eine TAD-Meldung, die Adresse 2 enthält, dekodiert das Drucker-Interface dies als My Talk Address (MTA) und unternimmt die notwendigen Schritte, um die Rolle des Talkers zu übernehmen.

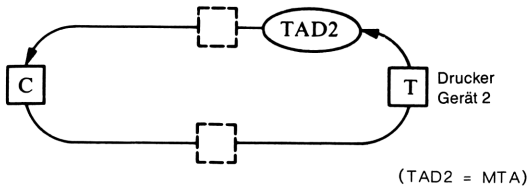


Abb. 3.17

Beachten Sie, daß Gerät 1 nicht mehr als Talker (T) gezeichnet ist, denn es kann nur jeweils einen aktiven Talker auf der Schleife geben. Wenn irgendein Gerät eine TAD-Adressenmeldung erhält, die eine andere Adresse beinhaltet, als die ihm zugewiesene (nicht MTA), muß das Gerät sich selbst automatisch als Talker abschalten. In diesem Falle dekodiert es diese Meldung als OTA (Other Talker Address).

Natürlich muß auch auf die TAD-Meldung (genau wie bei der LAD-Meldung) eine RFC-Meldung (Ready For Command) folgen, um sicherzustellen, daß die Geräte ihre neuen Rollen angenommen haben.

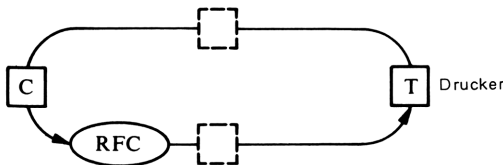


Abb. 3.18

Untalk (UNT)

 Der UNT-Befehl wird gebraucht, um den aktiven Talker in einen Ruhezustand zu versetzen. Wie in den vorherigen Abschnitten beschrieben, kann der aktive Talker auch dadurch in einen inaktiven Zustand versetzt werden, daß man einfach mittels einer TAD-Meldung einen anderen Talker adressiert. Deshalb werden Sie diese Meldung selten benutzen. Sie kann aber unter bestimmten Umständen trotzdem sehr nützlich sein, wenn Sie keinen aktiven Talker auf der Schleife mehr brauchen. Die Spezifikation beinhaltet deshalb diesen Befehl, damit abgesichert ist, daß es keine

logischen oder funktionellen Lücken gibt, die unter bestimmten Umständen unvorhergesehene Schwierigkeiten verursachen. Weil dieser Befehl selten gebraucht wird, wollen wir ihn an dieser Stelle nicht weiter erläutern.

Meldungen der Daten-Klasse (Data or End)

Alle bisher besprochenen Meldungen wurden benutzt, um die koordinierte Arbeitsweise des Systems sicherzustellen. Sie können deshalb als dem System übergeordnet betrachtet werden, denn der Zweck eines Systems ist selbstverständlich nicht, einfach nur funktionsfähig zu bleiben, sondern eine bestimmte Aufgabe auszuführen. Die Aufgabe, die ein HP-IL System ausführen soll, ist die Übertragung von Informationen und Daten zwischen den Geräten, die das System bilden. Die Daten, die über die Schleife übertragen werden, sind in Datenbyte-Meldungen (DAB) enthalten, und diese bilden den größten Teil der auf der Schleife übertragenen Information.

Datenbyte-Meldungen werden vom aktiven Talker auf die Schleife geschickt und sind an den aktiven Listener (an die aktiven Listener) auf der Schleife gerichtet. Datenbyte Meldungen enthalten keine Adressen. Wenn ein Gerät ein Listener ist, der eine Datenbyte-Meldung erhält, muß es annehmen, daß die Meldung für es bestimmt ist, d.h. die primäre Funktion eines Listener ist die, Datenbytes zu empfangen.

Die Information, die in einer Datenmeldung enthalten ist, ist ganz und gar dem Systemkonstrukteur überlassen. Die Information und ihre Interpretation ist völlig abhängig vom jeweiligen Gerät. Die Kodierung der Information kann in jeder gewünschten Art und Weise geschehen, obwohl es soweit als irgend möglich empfohlen wird, den ASCII-Standard zu benutzen, um eine möglichst weitreichende Kompatibilität mit anderen Geräten sicherzustellen.

Wenn Geräte, die nicht Listener sind, die Datenbyte-Meldung erhalten, müssen sie sie einfach auf der Schleife weitergeben. Listener, die die Datenbyte-Meldung erhalten, geben diese ebenfalls auf der Schleife weiter, während sie die Information für ihr betreffendes Gerät zurückhalten. Wenn nun unser Drucker-Interface als Listener adressiert ist, gibt es alle empfangenen Daten erstens auf der Schleife weiter und schickt sie zweitens zum Drucker.

Es gibt einen speziellen Typ von Datenmeldungen, die End Data Byte (END)-Meldungen genannt werden. Diese Meldung ist identisch mit der oben beschriebenen Datenmeldung und enthält ebenfalls ein Datenbyte. Sie kann aber zusätzlich dazu benutzt werden, um dem Listener das Ende eines logischen Datensatzes anzuzeigen. Die Wirkung einer END-Meldung auf das Gerät ist dem Konstrukteur überlassen und sie wird von den Geräten genauso gehandhabt wie jedes andere Datenbyte. Sie kann z.B. für einige Geräte das Ende einer Zeile signalisieren.

Es ist wichtig, daß Sie die END-Meldung nicht mit der vorher beschriebenen EOT-Meldung verwechseln. Die EOT-Meldung beendet keine Übertragung. Der Talker muß immer eine EOT-Meldung senden, wenn er seine Aufgabe beendet hat. Beachten Sie auch, daß Ihnen der Gebrauch der END-Meldung freigestellt ist. Wenn ein Gerät sie nicht benötigt, brauchen Sie sie auch nicht zu benutzen.

Kapitel 4

Ein HP-IL Interface auf der Ebene der elektronischen Bausteine

Dieses Kapitel soll erläutern, wie man ein Interface zum HP-IL aufbauen kann, indem man einen beliebigen LSI (Large Scale Integrated)-Baustein oder "Chip" benutzt. Um sich mit diesem Thema zu befassen, muß man sich noch intensiver mit den Details der HP-IL Spezifikation auseinandersetzen. Wieder wollen wir unsere Darstellung mit der Beschreibung der Eigenschaften, die ein LSI HP-IL Interface-Baustein hat beginnen.

Ein Überblick über den Interface-Chip

Abb. 4.1 zeigt das Blockdiagramm eines elektronischen Bausteins, der benutzt werden kann, um Fremdgeräte mit dem HP-IL zu verbinden. Wenn Sie diese Abbildung mit der Abb. 3.1 des vorhergehenden Kapitels vergleichen, werden Sie viele Ähnlichkeiten entdecken. Dies ist nicht verwunderlich, denn die Funktionen, die der Chip ausführt, müssen denen, die ein Adapter wie in Kapitel 3 beschrieben ausführt, recht ähnlich sein.

Abb. 4.1 ist jedoch detaillierter als Abb. 3.1, denn die HP-IL Operationen werden in diesem Kapitel ausführlicher besprochen. Aber zunächst wollen wir die Funktionen beschreiben, die unser Interface-Chip erfüllt.

Receiver-Kontroll-Logik

Diese Logik wandelt die Eingangssignale der Schleife auf eine logische Ebene um und kontrolliert das Laden des seriellen Bit-Stromes in den Eingabepuffer. Jede Meldung auf der Schleife wird als Abfolge von 11 Bits gesendet. Diese 11 Bits enthalten das, was man als Meldungsrahmen (frame) bezeichnet. Das erste Bit, das "Synchronisationsbit", ist auf eine besondere Art und Weise kodiert, so daß die Geräte der Schleife den Beginn eines HP-IL Rahmens erkennen können. Deshalb muß die Eingabe-Detektor-Logik die Signale von der Schleife überwachen, um das Synchronisationsbit zu ermitteln, das den Beginn eines Meldungsrahmens signalisiert.

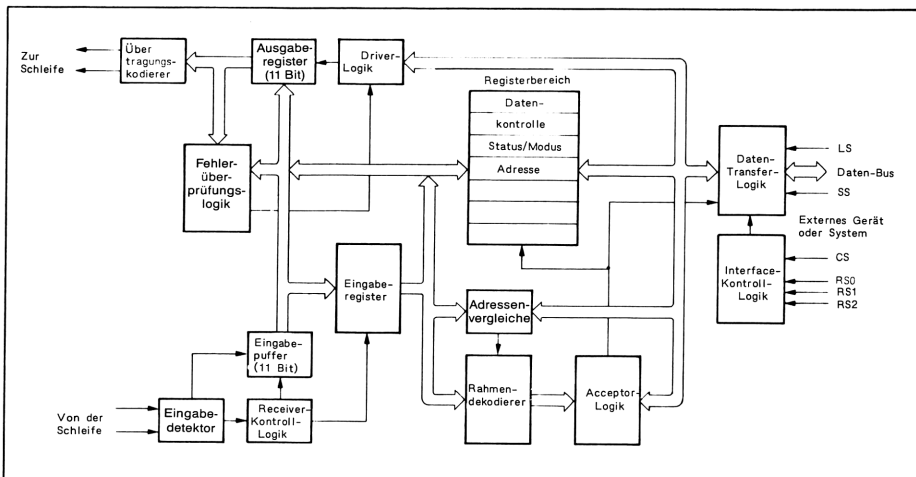


Abb. 4.1 Blockdiagramm eines Interfaces zum HP-IL auf der Ebene der elektronischen Bausteine

Wenn das Synchronisationsbit ermittelt worden ist, wird es zusammen mit den folgenden Bits unter Kontrolle der Receiver-Logik in den 11-Bit Eingabepuffer geladen. Daß wir an dieser Stelle das Wort Receiver benutzen, um diese logische Funktion zu beschreiben, ist Absicht. In früheren Kapiteln benannten wir Talker, Listener und Controller als Rollen, die die Geräte haben können. Dies sind jedoch nur einige der Rollen, die die HP-IL Spezifikation vorsieht.

Es sind 17 verschiedene Interfacefunktionen definiert, und wir werden sie alle ausführlicher im Kapitel 6 behandeln. An dieser Stelle wollen wir sie einfach in unser Konzept einfügen, denn diese Funktionen bilden die Eigenschaften von Interfaces für die Schleife. Die Receiver-Interfacefunktion ist dafür verantwortlich, Meldungen von der Schleife zu empfangen. Selbstverständlich müssen alle Geräte auf der Schleife diese Fähigkeit haben, gleichgültig, welche primäre Rolle sie spielen, ob Talker, Listener oder Controller. Wie Sie später sehen werden, sind viele der Interface-Funktionen wahlfrei.

Nachdem alle 11 Bits in den Eingabepuffer geladen wurden, werden sie durch die Receiver-Kontroll-Logik in das Eingaberegister übertragen. Von dort aus kann die Eingabemeldung dekodiert und an andere logische Einheiten des HP-IL weitergegeben werden.

Rahmen-Dekodierungs-Logik

Wie wir in den vorangegangenen Abschnitten gesehen haben, besteht jeder Meldungsrahmen aus 11 Bits. Die 3 höchstwertigen Bits bestimmen die spezielle Kategorie der Meldung, z.B. Befehl, Daten usw. Die verbleibenden 8 Bits sind Datenbits, die den Meldungstyp innerhalb einer der Kategorien weiter spezifizieren, oder die einfach ein Byte ASCII-Daten enthalten.

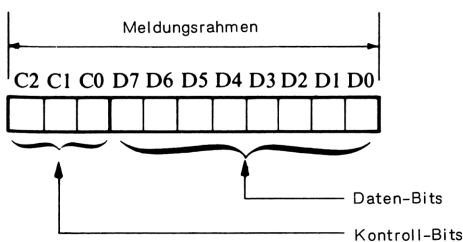


Abb. 4.2

Innerhalb des Meldungsrahmens gehen die höchstwertigen Bits als erste über die Schleife. So ist es möglich, die Kategorie der empfangenen Meldung schon festzustellen, wenn die ersten 3 Bits (C2, C1, C0) empfangen worden sind. Weil eine Meldung so nicht immer vollständig dekodiert werden muß,

bevor man feststellen kann, ob sie zum nächsten Gerät auf der Schleife weitergegeben werden soll, kann man dies in manchen Fällen dazu benutzen die Schleifenoperation zu beschleunigen. Wenn Sie von dieser Tatsache Gebrauch machen wollen, können Sie eine zusätzliche Logik benutzen, um die ersten 3 Bits der Meldung zwischen dem Eingabepuffer und dem Eingaberegister schon vorab zu dekodieren, wie in Abb. 4.1 gezeigt wird.

Adressen-Vergleicher-Logik

Kapitel 3 beschrieb zwei Befehle: Listen Address (LAD) und Talk Address (TAD), die als Teil des Befehls eine Adresse enthielten. Wie Sie später in diesem Kapitel sehen werden, gibt es auch andere Befehle, die Adreßinformationen enthalten. Wenn eine Meldung, die eine Adresse enthält, von einem Gerät empfangen wird, dem eine andere Adresse zugewiesen wurde, wird dieses Gerät die Meldung üblicherweise einfach weitergeben.

Abb. 4.1 zeigt eine Adressen-Vergleicher-Logik. Diese Logik vergleicht die Adresse, die in der Meldung enthalten ist, mit der eigenen Adresse, die in einem Adreßregister auf dem Chip enthalten ist. Wenn die zwei Adressen übereinstimmen, ist die Rahmen-Dekodierungslogik aktiviert. Im anderen Fall wird die Meldung einfach auf die Schleife weitergegeben.

Wenn eine Meldung eine Adresse enthält, dann ist die Adresse in den 5 niedrigstwertigen Bits (die letzten 5 Bits der empfangenen Meldung) des Meldungsrahmens enthalten.

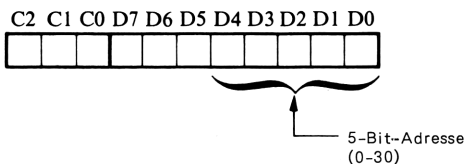


Abb. 4.3

Beachten Sie, daß gültige Adressen im Bereich von 0 bis 30 liegen können und nicht im Bereich von 0 bis 31. Eine Adresse von "1111" (dezimal 31) ist nicht zulässig, denn

diese Adresse ist speziellen Befehlen (Unlisten, Untalk) vorbehalten. Dieser spezielle Gebrauch wird später zusammen mit der erweiterten Adressierungsfähigkeit besprochen, die benutzt werden kann, um bis zu 961 Geräte auf der Schleife miteinander zu verbinden.

Acceptor(25)-Logik

Die Acceptor-Logik in Abb. 4.1 ist die primäre Kontroll-Logik unseres Chips. Sie enthält die Informationen des dekodierten Meldungsrahmens und bestimmt, welche Handlungen vom Chip unternommen werden sollen (z.B. Weitergabe der Meldungen auf die Schleife, Datenübertragung an das externe Gerät usw.). Diese Logik kontrolliert gleichfalls das Auslesen und Schreiben in den Registerbereich des Chip. Die Handlungen, die die Acceptor-Logik dem Chip zuweist, hängt natürlich nicht nur von der Rolle ab, die der Chip und das an ihn angeschlossene Gerät zu diesem Zeitpunkt spielen. Die ausgeführten Handlungen können unterschiedlich sein, je nachdem, ob der Chip Talker oder Listener ist.

Bis zu dieser Stelle haben wir das Konzept der Interface-Funktionen mit der Erklärung der Receiver-Logik in Abb. 4.1 eingeführt. Der Ausdruck Acceptor ist ein anderer wichtiger Begriff, der als Interfacefunktion in der HP-IL Spezifikation definiert ist. Auch hier wollen wir die Erläuterung über das Acceptor-Interface bis zum Kapitel 7 zurückstellen, wo wir es ausführlich behandeln.

Registerbereich

Der Registerbereich in Abb. 4.1 speichert spezifische Informationen des Chip, wie die Adresse des Gerätes, Status- oder Kontrollinformationen und Daten. Zusätzliche "Schmierblattregister" können für andere spezielle Funktionen oder Datenmanipulationen eingesetzt werden. Das Datenregister enthält Daten, die zwischen der Schleife und dem externen Gerät übertragen werden.

Das Kontrollregister enthält Informationen, die vom Chip gebraucht werden, um seine verschiedenen internen Funktionen zu erfüllen, wie z.B. Informationen über die Meldung, die gerade verarbeitet wird, oder über den Interface-Chip des externen Gerätes.

Das Statusregister enthält Informationen über die Rolle des Chip und des daran angeschlossenen Gerätes. So wie z.B. "ich bin ein aktiver Talker, der Systemcontroller" usw. Das Adreßregister enthält die Adresse, die dem Chip und dem daran angeschlossenen Gerät zugewiesen wurde. Diese Adresse wird gewöhnlich durch das externe Gerät in das Register geladen und wird von der Adressen-Vergleicher-Logik benutzt, um festzustellen, ob die Meldungen für dieses Gerät bestimmt sind.

Datenübertragung und Interface

Diese logischen Bausteine beinhalten das Interface zwischen dem Chip und dem daran angeschlossenen Gerät. Datenübertragungen zwischen Chip und Gerät gehen normalerweise ziemlich unabhängig oder asynchron von der Übertragung zwischen dem Chip und der Schleife vonstatten. Wenn die Schleife z.B. Informationen des Gerätes benötigt, signalisiert der Chip dies dem Gerät, erhält von ihm die Daten und leitet sie auf die Schleife weiter.

Stellen Sie sich vor, das Gerät, das mit dem Chip verbunden ist, sei eine Art Mikrocomputer. Man kann dann zugleich annehmen, daß das Interface, das der Chip zur Verfügung stellt, ganz allgemein ein Mikrocomputer-Interface ist. Obwohl wir im Rahmen dieses Buches keine genaue Darstellung eines solchen Interfaces geben wollen, liefern wir Ihnen hier einige allgemeine Eigenschaften, die Sie bedenken sollten.

Zunächst enthält der Chip einen bidirektional 8-Bit-Datenbus für die Informationsübertragung zwischen dem Mikrocomputer und dem Chip. Diese Informationsübertragung wird durch die Lese(LL)- und Schreib(SS)- Signale, die der Mikrocomputer erzeugt, synchronisiert.

Zum zweiten behandelt ein Mikrocomputer den HP-IL Interface-Chip einfach wie irgendein anderes Eingabe/Ausgabegerät. Zu diesem Zweck ist auf dem Chip eine Anwahlleitung (Chip-Select CS) vorgesehen, um den Mikrocomputer zu befähigen, den Chip von anderen Geräten zu unterscheiden, die ebenfalls mit dem Mikrocomputer verbunden sind. Damit der Mikrocomputer Daten zwischen sich und einem bestimmten Register innerhalb des Registerbereichs auf dem Chip übertra-

gen kann, haben wir drei Registerauswahlleitungen (RS0-RS) vorgesehen.

Driver-Logik

Diese Logik übermittelt Meldungen auf die Schleife. Driver ist eine in der HP-IL Spezifikation festgelegte Interface-Funktion, deshalb ist auch hier die Nomenklatur wieder vorab definiert. Obwohl die meisten dieser und anderer Interface-Funktionen im Kapitel 7 behandelt werden, wollen wir schon an dieser Stelle kurz darauf eingehen, wie sich die Driver-Interface-Funktion von der wichtigeren primären Funktion des Talkers unterscheidet.

Ein Driver muß Informationen erzeugen und sie auf die Schleife senden können. Deshalb muß ein Gerät, das Talker ist, zugleich auch Driver-Interface-Funktionen beinhalten. Selbst wenn ein Gerät kein Talker ist, muß es in der Lage sein, empfangene Meldungen wenigstens zum nächsten Gerät auf der Schleife weiterzugeben. Deshalb müssen alle Geräte mit der Driver-Interface-Funktion ausgestattet sein.

An dieser Stelle beginnt das Konzept der "Interface-Funktionen", die von den primären "Rollen" der Schleifengeräte unterschieden sind, Gestalt anzunehmen. Interface-Funktionen bezeichnen die Art und Weise der Aufgabenteilung in verschiedene Kategorien und Funktionen, die die Geräte auf der Schleife vollziehen müssen. Damit ein Gerät überhaupt an der Schleifenoperation teilnehmen kann, muß es außer dem Vollzug der Funktionen, die auf seinen eigenen Eigenschaften basieren, auch noch Interface-Funktionen übernehmen.

Ausgaberegister

Dieses Register beinhaltet den 11-Bit-Meldungsrahmen, der zum nächsten Gerät auf der Schleife geschickt werden soll. Diese Meldung kann im einfachsten Falle gerade erst vom vorhergehenden Gerät empfangen worden sein und direkt vom Eingaberegister des Chip eingelesen werden. Die Meldung kann alternativ dazu über den Registerbereich des Mikrocomputers oder gesteuert von der Kontroll-Logik des Chip aus dem Registerbereich kommen. Wer die Quelle der abgehenden Meldung ist, hängt davon ab, auf welche Art das

Gerät die vorhergehende Meldung verarbeitet, die empfangen wurde.

Fehlerüberprüfung

Die Fehlerüberprüfungslogik in Abb. 4.1 ist ein unverzichtbarer und wichtiger Teil des HP-IL Schemas. Da alle Geräte, die eine Meldung senden können (d.h. eine Meldung erzeugen und auf die Schleife schicken), normalerweise warten, bis diese Meldung zurückkehrt, bevor sie eine weitere Meldung herausgeben, kann die Fehlerüberprüfung leicht durchgeführt werden. Die HP-IL Spezifikation verlangt nicht von jedem Gerät die Durchführung der Fehlerüberprüfung, rät jedoch dazu, daß alle Geräte diese Fähigkeit besitzen. Diesem Rat sollte man Folge leisten. Da Sie normalerweise immer auf den Rücklauf einer Meldung warten, die Sie ausgeschickt haben, und so lange nichts weiteres unternehmen können, warum sollten Sie dann während dieser Zeit nicht untersuchen, ob sie genauso zurückkommt, wie Sie sie geschickt haben?

Kodierung und Übertragung

Diese logische Abteilung verwandelt einfach den parallelen Meldungsrahmen, der im Ausgaberegister enthalten ist, in einen seriellen Bit-Strom mit den Spannungspegeln, die die Schleife benötigt.

Ein Mikrocomputer als Controller

Nehmen Sie an, daß das Gerät, das durch den Chip an die Schleife angeschlossen ist, ein Mikrocomputer ist. Dieser Mikrocomputer soll die Funktion des Schleifencontrollers erfüllen. Da HP-IL für Anwendungen vorgesehen ist, die mit niedriger Geschwindigkeit und mit geringem Leistungsbedarf arbeiten, kann man annehmen, daß ein relativ leistungsstarkes Gerät wie ein Mikrocomputer der Schleifen-Controller ist.

Als Controller hat der Mikro (der Kürze halber wollen wir diese Abkürzung für den Rest des Kapitels benutzen) viel mehr Verantwortung für die Schleife wie der Thermodrucker in Kapitel 3. Der Drucker war primär Listener. Der Mikro

muß in der Lage sein, alle Rollen (Talker, Listener und Controller) zu übernehmen. Lassen Sie uns nun die verschiedenen Meldungen erläutern, mit denen der Mikro umgehen muß.

Meldungen der Command-Klasse

Wie wir in Kapitel 3 festgestellt haben, werden die Meldungen der Command-Klasse dazu benutzt, die Ausgangs-Arbeitsbedingungen auf der Schleife herzustellen, den Arbeitsmodus und die Rollen, die die Geräte auf der Schleife spielen, zu verändern und ganz allgemein die geordnete Arbeitsweise und Kontrolle des Schleifensystems aufrecht zu erhalten. Nachfolgend finden Sie die Meldungen aus der Command-Klasse, die ein Mikro in der Rolle des Controllers senden kann.

- Interface Clear (IFC)
- Device Clear (DCL)
- Loop Power-Down (LPD)
- Listen Address (LAD)
- Unlisten (UNL)
- Talk Address (TAD)
- Untalk (UNT)

Beachten Sie: wir haben zwar alle diese Meldungen bereits in Kapitel 3 besprochen; wir wollen sie jedoch an dieser Stelle, wo sie den Mikro als Controller betreffen, nochmals anführen.

- Selected Device Clear (SDC)
- Go To Local (GTL)
- Remote Enable (REN)
- Not Remote Enable (NRE)
- Parallel Poll Enable (PPE)
- Parallel Poll Disable (PPD)
- Parallel Poll Unconfigure (PPU)

Interface Clear-Meldung

Die IFC-Meldung kann nur vom System-Controller gesendet werden. Unser Mikro soll nun nicht nur die Rolle des Schleifen-Controllers, sondern auch die des System-Controllers haben. Wie in Kapitel 3 ausgeführt, wird die IFC-

Meldung normalerweise um die Schleife geschickt, wenn das System zum ersten Mal eingeschaltet wird, um alle anderen Geräte in den Ausgangszustand zu versetzen.

Lassen Sie uns hier einen Moment lang innehalten und betrachten, wie der Mikro und das Interface-Chip mit der IFC-Meldung arbeiten.

Nehmen Sie an, daß sich der Chip selbst beim Einschalten des Systems in einer Art neutralem oder Ruhezustand befindet: er spielt keine aktive Rolle auf der Schleife. Der Mikro schreibt zunächst Informationen in das Statusregister des Chip, die dem Chip mitteilen, daß er (und der mit ihm verbundene Mikro) als System-Controller der Schleife fungieren soll.

Dann sendet der Mikro Daten an den Chip, die den IFC-Befehl repräsentieren, und dieser sendet die IFC-Meldung auf die Schleife. Wenn die IFC-Meldung von ihrer Reise um die Schleife zurückkehrt, informiert der Chip den Mikro über diese Tatsache. Jetzt sendet der Mikro den RFC-Befehl auf die Schleife, um sicherzustellen, daß alle Geräte das Interface Clear-Kommando ausgeführt haben (ausführliche Erläuterung der IFC- und RFC-Befehle siehe Kapitel 3).

Selected Device Clear (SDC)

Die Device Clear-Meldung und ihre Wirkung auf die der Schleife angeschlossenen Geräte ist im Kapitel 3 beschrieben. Jetzt kann die SDC-Meldung eingeführt werden. Die DCL-Meldung bewirkt, daß alle Geräte (sofern sie in der Lage sind, auf die Meldung zu antworten) ihre jeweiligen Löschoptionen durchführen, nachdem sie die Meldung erhalten haben. Die SDC-Meldung weicht davon insofern ab, als daß sie nur diejenigen Geräte auf der Schleife, die gerade die Rolle des Listeners spielen, dazu veranlaßt, sich selbst zu löschen.

Dieser Unterschied ist sehr wichtig, denn es kann auf der Schleife zwar zur gleichen Zeit eine Vielzahl von Listenern geben, aber nur jeweils einen Talker oder Controller. Wenn der Mikro also eines oder mehrere Geräte löschen will (aber nicht unterschiedslos alle Geräte), schickt er den SDC-Befehl und führt somit das Löschen aller adressierten Listener durch. Beachten Sie, daß es noch einen Befehl gibt, der benutzt wird, um den Listenerstatus aller Geräte

zu löschen - den UNL-Befehl, den wir in Kapitel 3 erläutert haben. Der SDC-Befehl bewirkt, daß sich alle Listener löschen, aber nicht notwendig ihre Listener-Rolle ablegen. Der SDC-Befehl kann z.B. einen Drucker dazu veranlassen, das Papier zum Blattanfang zu transportieren, oder ein Kassettenlaufwerk dazu, die Kassette bis zum Bandanfang zurückzuspulen usw.

Remote/Local(26)-Kommandos

Manche Geräte haben an ihrer Frontseite Knöpfe und Schalter, die dazu benutzt werden, ihre Arbeitsweise einzustellen. Ein Voltmeter ist dafür ein gutes Beispiel. Manchmal ist es angebracht, die Geräte über ihre manuellen Kontrollknöpfe zu bedienen (Local-Modus), und manchmal ist es günstiger, sie automatisch über die Schleife arbeiten zu lassen (Remote-Modus, Fernsteuermodus). Drei Befehle erlauben es dem Controller, zwischen diesen beiden Arbeitsweisen umzuschalten: Remote Enable (REN), Not Remote Enable (NRE) und Go To Local (GTL). Die meisten Geräte haben keine eigenen Kontrollknöpfe, oder ihre Kontrollknöpfe sind nicht abschaltbar (wie beim Drucker oder beim Kassettenlaufwerk). Solche Geräte ignorieren diese Befehle einfach und geben sie an das nächste Gerät auf der Schleife weiter.

Wenn der Voltmeter zum ersten Mal eingeschaltet wird, befindet er sich im Local-Modus. Er kommuniziert auf der Schleife ganz normal, aber die Instrumente-Einstellung (z.B. der Spannungsbereich) wird durch die Einstellknöpfe kontrolliert und kann durch Steuerdaten von der Schleife nicht verändert werden.

Eine der ersten Handlungen, die der Schleifen-Controller normalerweise ausführt, ist das Senden des Remote Enable (REN)-Befehls. Allein bewirkt der REN-Befehl noch nicht viel; die Geräte reagieren weiterhin nur auf ihre Einstellknöpfe. Aber jetzt ist die ganze Schleife in einem Zustand, der das Umschalten vom Local zum Remote-Modus und umgekehrt gestattet, sobald dies notwendig ist.

Nachdem das Voltmeter den REN- und den LAD-Befehl erhalten hat, wird es jetzt sofort auf Remote-Modus umschalten. Dies ist einleuchtend, weil das Gerät auf jeden Fall Listener werden muß, um die richtigen Einstellwerte von der Schleife zu erhalten. Dadurch ist es dem Controller auch

möglich, das Gerät oder die Geräte herauszusuchen, die er fernsteuern möchte, währenddessen er andere Geräte im Local Modus beläßt. Wenn es einmal auf Remote-Modus umgeschaltet ist, wird das Gerät in diesem Zustand verbleiben, auch wenn der Controller später den Unlisten (UNL)-Befehl schickt und ein anderes Gerät zum Listener macht.

Wenn der Controller das Voltmeter auf manuelle Bedienung umstellen will, schickt er den Go To Local (GTL)-Befehl. Dieser Befehl ist dem SDC-Befehl sehr ähnlich, von dem wir bereits wissen, daß ihm nur diejenigen Geräte Folge leisten, die als Listener adressiert sind. Wieder kann der Schleifen-Controller sich aussuchen, welches Gerät (oder welche Geräte) er in den Local-Modus versetzen will.

Wenn der Controller alle Geräte in den Local-Modus zurückversetzen will, kann er den Not Remote Enable (NRE)-Befehl schicken. Dieser versetzt alle Geräte zur gleichen Zeit in den Local-Modus und löscht gleichzeitig den Zustand, der es der Schleife erlaubt, in den Remote-Modus umzuschalten. Nun sind alle Geräte im Local-Modus (auch wenn sie ein LAD-Kommando erhalten) bis sie erneut die REN-LAD Abfolge erhalten.

Parallele Abfrage(27)

In dieser Gruppe gibt es drei Befehlsmeldungen, und da sie alle miteinander verknüpft sind, wollen wir sie auch zusammen besprechen.

In Kapitel 3 erläuterten wir die Send Status (SST)-Meldung und erwähnten, daß man sie auch als serielle Abfrage bezeichnen kann, denn der Controller kann mit dieser Meldung sukzessive den Status und die speziellen Anforderungen jedes beliebigen Gerätes auf der Schleife feststellen. Natürlich nimmt die Prozedur der seriellen Abfrage einige Zeit in Anspruch, wenn viele Geräte auf der Schleife sind. Deshalb haben die Konstrukteure von HP-IL eine zeitsparendere Methode entwickelt, um sich über den Status und die speziellen Anforderungen der Geräte zu informieren: die parallele Abfrage.

Die SST-Meldung ist auf ein einzelnes adressiertes Gerät auf der Schleife beschränkt (serielle Abfrage) und kann weiterhin nur von Geräten beantwortet werden, die gerade als Talker adressiert sind. Die parallele Abfragefähigkeit,

die von den Geräten auf der Schleife beinhaltet werden kann, erlaubt dem Controller, schnell die Anforderungen aller Geräte auf der Schleife festzustellen (dies gilt natürlich nur für die Geräte, die diese Fähigkeit haben). So müssen Sie das Gerät nicht zuerst als Talker adressieren und dann seinen Status abfragen.

Parallel Poll Enable (PPE)

Diese Meldung sendet der Controller an ein bestimmtes Gerät auf der Schleife, um die parallele Abfrage-Fähigkeit dieses Gerätes zu aktivieren. Selbst wenn ein Gerät so konstruiert ist, daß die parallele Abfrage-Fähigkeit vorgesehen ist, muß sie erst vom Controller aktiviert werden. Dies versieht den Controller mit größtmöglicher Flexibilität bei der Kontrolle und Gestaltung der Schleife. Bestimmte Abfrageoperationen können z.B. nur von bestimmten Geräten ausgeführt werden. In diesem Fall will der Controller auch nur von diesen bestimmten Geräten Antwort. Die Antwort von anderen Geräten würde die Effektivität der Abfrageoperation einschränken.

Wie Sie sich erinnern, ist die PPE-Meldung an ein bestimmtes Gerät auf der Schleife gerichtet. Die Meldung enthält nicht die Adresse dieses Gerätes. Stattdessen ist die PPE-Meldung immer an den aktiven Listener auf der Schleife gerichtet. In Kapitel 3 sagten wir, daß es mehr als einen aktiven Listener auf der Schleife geben kann. Deshalb empfiehlt es sich, zuerst mit der Unlisten (UNL)-Meldung alle Listener außer Aktion zu setzen. Dann sendet man die Listen Address (LAD)-Meldung an dasjenige Gerät, das zur parallelen Abfrage befähigt werden soll. Schließlich sendet man die PPE-Meldung um die Schleife, um die Abfrage-Antwortfähigkeit des Gerätes zu aktivieren. Im folgenden sehen Sie diese Meldungsabfolge noch einmal:

- UNL (Unlisten-Meldung, um alle Listener außer Aktion zu setzen)
- RFC (Ready For Command stellt sicher, daß die UNL-Meldung ausgeführt wurde)
- LAD2 (Gerät 2 wird als Listener adressiert)
- RFC (stellt sicher, daß die RFC-Meldung durchgeführt wurde)

- PPE (befähigt Gerät 2 zur parallelen Abfrage)
- RFC (stellt sicher, daß die PPE-Meldung durchgeführt wurde)

Für jedes Gerät auf der Schleife, das die parallele Abfrage durchführen soll, muß diese vollständige Meldungsabfolge wiederholt werden. Das mag umständlich erscheinen, und Sie könnten sich fragen, ob es nicht ratsamer ist, einfach eine serielle Abfrage für alle Geräte auf der Schleife durchzuführen. Denken Sie jedoch daran, daß diese Befehlsfolge noch zum Konfigurationsteil der gesamten Schleifenoperation gehört, der dazu benötigt wird, um die Geräte auf die parallele Abfrage vorzubereiten. Deshalb müssen Sie diese Abfolge entweder nur einmal durchlaufen, nämlich wenn sie das System einschalten, oder nur in sehr großen Zeitabständen während der Systemoperation. Die Abfrage selbst wird, wie Sie gleich sehen werden, leicht und schnell vorstatten gehen.

Obwohl die PPE-Meldung nicht die Adresse des Geräts enthält, das aktiviert werden soll, so enthält sie doch einige Konfigurationsinformationen, die festlegen, wie das nachfolgende Gerät auf eine parallele Abfrage antworten soll. Abb. 4.4 zeigt das Format einer PPE-Meldung.

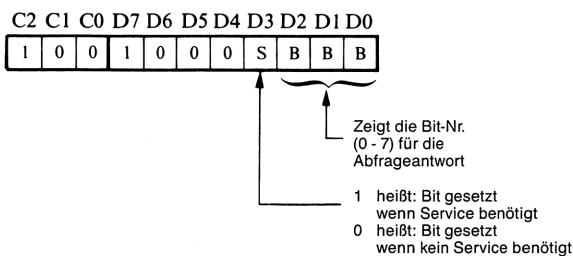


Abb. 4.4

Die sieben höchstwertigen Bits enthalten das Bitmuster für die PPE-Meldung. Die vier niedrigstwertigen Bits teilen dem Gerät, das für die parallele Abfrage aktiviert werden soll, mit, wie es auf die folgenden Abfrageoperationen antworten soll. Bit D3 (das "S-Bit") definiert den "Sinn" oder die Polarität der Antwort des Geräts auf eine Abfrage. Wenn das S-Bit "1" ist, muß das Gerät auf die Abfrage antworten,

indem es das entsprechende Bit im Abfragewort setzt, wenn es Service benötigt. Ist das S-Bit "0", antwortet das Gerät auf die Abfrage, indem es das entsprechende Bit im Abfragewort setzt, wenn es keinen Service braucht (d.h. das entsprechende Bit wird gelöscht, wenn das Gerät Service braucht).

Die drei niedrigstwertigen Bits in der PPE-Meldung legen fest, welches Bit das Gerät im Abfragewort für seine Antwort setzen soll. Wenn z.B. alle drei Bits Nullen (000) sind, wird das Gerät das niedrigstwertige Bit im Abfragewort setzen, und wenn alle drei Bits eins (111) sind, wird das Gerät mit dem höchstwertigen Bit (Bit 07) im Abfragewort antworten.

Die letzten Abschnitte bezogen sich auf das Abfragewort, das das Gerät während der Abfrage verändert. In Wirklichkeit ist dieses Abfragewort Teil der Identify (IDY)-Meldung.

Identify (IDY)

Diese Meldung wird vom Controller gesendet, um festzustellen, ob die Geräte auf der Schleife Service benötigen. Wenn

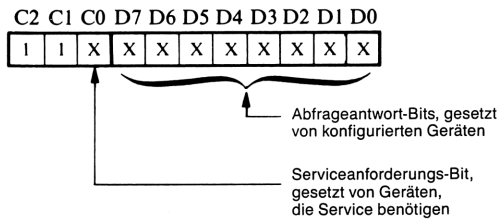


Abb. 4.5

ein Gerät den Eingriff des Controllers braucht, muß es das Serviceanforderungs-Bit setzen (C0). Beachten Sie, daß dieses Bit auch von Geräten gesetzt werden kann, die nicht die Fähigkeit zur parallelen Abfrage haben. Die IDY-Meldung wird auch bei der seriellen Abfrage benutzt. Wenn die IDY-Meldung um die Schleife gelaufen ist und zum Controller zurückkehrt, kann das C0-Bit untersucht werden, um zu sehen, ob irgendein Gerät Service anfordert. Wenn C0 gesetzt ist,

zeigt es an, daß ein Gerät auf der Schleife Service will. Wenn Geräte zur parallelen Abfrage befähigt sind, setzen sie nicht nur einfach das C0-Bit, sondern auch das ihnen zugewiesene Bit im Datenteil (D7-D0) der IDY-Meldung. So kann der Inhalt des IDY-Datenbytes untersucht werden, wenn die Meldung wieder beim Controller angelangt ist, um festzustellen, welches der Geräte, die die parallele Abfrage durchführen können, Service angefordert hat.

Beachten Sie, daß es möglich ist, daß mehr als ein Gerät auf die parallele Abfrage antwortet. Jedes Gerät, das Service braucht, benutzt sein eigenes spezifisches Bit, um zu antworten. Ebenso sollten Sie beachten, daß die Dekodierung der zurückkehrenden IDY-Meldung, zur Feststellung, welches Gerät (oder welche Geräte) Service brauchen, keine dem Controller immanente Funktion ist. Die Dekodierung der IDY-Meldung und die nachfolgende Reaktion des Controllers um den Serviceanforderungen genüge zu tun, hängt vollständig vom jeweils aufgebauten System ab. Der Interface Chip kann beispielsweise die Tatsache dekodieren, daß das Serviceanforderungs-Bit in der zurückkehrenden IDY-Meldung gesetzt ist. Der Chip gibt die Information, zusammen mit dem IDY-Datenbyte an den Mikro weiter. Jetzt liegt die Dekodierung des Datenbytes beim Mikro, der dann feststellt, wer Service benötigt, um dann die angemessenen Schleifenoperationen einzuleiten, die den Service sicherstellen.

Parallel Poll Disable (PPD) und Unconfigure (PPU)

Diese beiden Meldungen beenden die parallele Abfragemöglichkeit, die durch den Parallel Poll Enable (PPE)-Befehl initiiert wurde. Durch die PPD-Meldung werden Geräte, die gerade als Listener adressiert sind, veranlaßt, nicht mehr auf die parallele Abfrage zu antworten. Wenn Sie eines dieser Geräte später wieder dazu befähigen wollen, müssen Sie die PPE-Meldung senden. Die Geräte, die nicht als Listener adressiert sind, aber zur parallelen Abfrage konfiguriert wurden, werden von der PPD-Meldung nicht betroffen und behalten ihre Fähigkeit, auf die parallele Abfrage zu antworten.

Die Parallel Poll Unconfigure (PPU)-Meldung wird vom Controller benutzt, um alle Geräte außer Stand zu setzen, auf eine parallele Abfrage zu antworten. Wenn ein Gerät diesen

Befehl erhält, verliert es die Fähigkeit, auf die parallele Abfrage zu antworten, gleichgültig, ob es gerade adressiert ist oder nicht. So kann man die PPU-Meldung als eine Systemlöscher- oder Zurücksetzmeldung für die parallele Abfrage betrachten, während die PPD-Meldung eher eine gerätespezifische Operation darstellt.

Die Meldungen der Ready(28)-Klasse

Die Meldungen der Ready-Klasse koordinieren und synchronisieren die Übertragung von Meldungen und Daten zwischen den einzelnen Geräten auf der Schleife. Wie in Kapitel 3 ausgeführt, können die Ready-Meldungen nicht nur vom Controller, sondern auch von anderen Geräten ausgesendet werden. In der folgenden Darstellung soll jedoch angenommen werden, daß unsere Mikro/Interface-Kombination die Meldungen erzeugt. Dies sind die Ready-Meldungen, die unser Mikro aussenden kann:

- Ready For Command (RFC)
- Send Status (SST)
- End of Transmission, OK (ETO)
- End of Transmission, Error (ETE)

Beachten Sie: Zwar haben wir die o.g. Meldungen schon in Kapitel 3 besprochen; wir wollen sie aber an dieser Stelle insoweit nochmals aufführen, als sie den Mikro in seiner Rolle als Controller betreffen.

- Send Data (SDA)
- Not Ready for Data (NRD)
- Send Device ID (SDI)
- Take Control (TCT)

Ready For Command (RFC)

Wie in Kapitel 3 beschrieben, sendet der Controller den RFC-Befehl, um festzustellen, ob ein unmittelbar vorausgegangener Befehl ausgeführt wurde. Die Abfolge der beiden Meldungen Befehl - RFC wird noch häufig auftauchen, denn das HP-IL Protokoll fordert, daß jeder Befehlsmeldung eine RFC-Meldung folgt - dies ist Teil der Schleifen-Handshake Pro-

zedur. Weil diese Reihenfolge immer notwendig ist, muß sich unser Mikro nicht damit befassen. Stattdessen muß der Interface-Chip automatisch dann, wenn er einen Befehl, den er vorher auf die Schleife geschickt hat, zurückerhält, die RFC-Meldung schicken. Die Zusammenarbeit zwischen dem Chip und dem Mikro sieht dann folgendermaßen aus: der Mikro versetzt den Chip (wenn er sich nicht sowieso schon darin befindet) in den Controller Modus und sendet dem Chip anschließend den Befehl (in Abb. 4.6 CMD), der auf die Schleife geschickt werden soll.

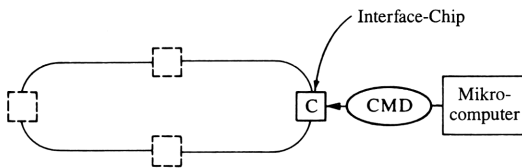


Abb. 4.6

Der Chip sendet die Befehlsmeldung nun auf die Schleife, ohne daß ein weiterer Eingriff oder eine weitere Überwachung des Mikros notwendig ist.

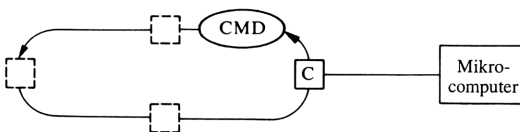
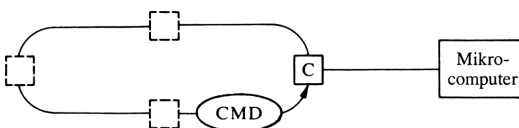


Abb. 4.7

Wenn der Befehl (CMD) zum Chip zurückkehrt, erzeugt er automatisch die RFC-Meldung und schickt sie auf die Schleife. Dies geschieht wiederum völlig unabhängig vom Mikro.



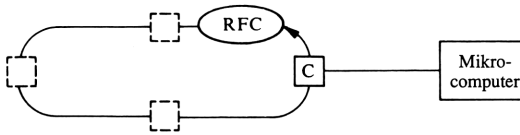


Abb. 4.8

Kehrt die RFC-Meldung schließlich zum Chip zurück, so informiert der Chip den Mikro, daß der ausgegebene Befehl erfolgreich erfüllt wurde. Dies geschieht, indem er entweder beim Mikro eine Unterbrechung herbeiführt, oder eine andere vorher festgelegte Prozedur benutzt. Der Mikro fährt dann mit der nächsten erwünschten Schleifenoperation fort.

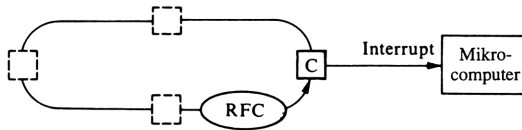


Abb. 4.9

Send Status (SST)

Wie Sie sich erinnern werden, dient diese Meldung dem Controller dazu, von dem Gerät, das der aktive Talker auf der Schleife ist, geräteorientierte Statusinformation von einem oder mehreren Bytes zu erlangen. Wenn der aktive Talker die SST-Meldung empfängt, überträgt er diese Meldung nicht weiter, sondern sendet stattdessen eines oder mehrere Datenbytes (DAB's) zum Controller.

Beachten Sie, daß die Übersetzung der Datenbytes, die als Antwort auf die SST-Meldung zurückgeschickt werden, ganz im Belieben des Systemkonstruktors liegt. Sie können deshalb nicht vom Interface-Chip die Dekodierung oder Analyse der Datenbytes erwarten. Stattdessen gibt er die Datenbytes zum Mikro weiter, damit sie dort weiterverarbeitet werden.

End of Transmission (ETO, ETE)

In Kapitel 3 wurde ausgeführt, daß diese Meldungen immer vom aktiven Talker gesendet werden und an den Controller

gerichtet sind. Die ETO-Meldung zeigt an, daß der Talker alle Datenbytes, die er geschickt hat, korrekt zurückerhalten hat. Wenn der Chip diese Meldung vom Talker erhält, bedeutet das, daß die Übertragung vollständig erfolgt ist und daß die nächste Schleifenoperation erfolgen kann.

Wenn der Chip die ETE-Meldung erhält, die besagt, daß der Talker in den zurückgekehrten Datenbytes Fehler entdeckt hat, so teilt der Chip dem Mikro diesen Fehlerstatus mit. Welche Schritte nun unternommen werden, obliegt ganz dem Mikro. Der Chip unternimmt nichts weiter, als den Fehler mitzuteilen. Der Mikro setzt das System dann normalerweise wieder instand, z.B. durch einen zweiten Übertragungsversuch. Und wenn dieser fehlschlägt, fordert er den Eingriff des Bedienungspersonals an.

Send Data (SDA)

Diese Meldung schickt der Controller an den aktiven Talker und veranlaßt diesen damit, die Datenbytes (DAB's) auf die Schleife zu übertragen. Da es zu einem bestimmten Zeitpunkt immer nur einen aktiven Talker auf der Schleife gibt, enthält die SDA-Meldung keine Adresse. Sie ist immer an den gegenwärtigen Talker gerichtet.

Wenn der Talker die SDA-Meldung erhält, gibt er diese nicht weiter zum Controller. Stattdessen ersetzt er die SDA-Meldung durch sein erstes Datenbyte und beginnt so die Übertragung. Nachdem der Talker das letzte Datenbyte geschickt hat, sendet er die entsprechende End of Transmission (ETO oder ETE)-Meldung, um dem Schleifencontroller mitzuteilen, daß er eine neue Schleifenoperation beginnen kann.

Ein Datenbyte, das vom Talker um die Schleife geschickt wird, wird von jedem Gerät weitergegeben, bis es wieder beim Talker angelangt, der es nun durch das nächste ersetzt. Die Daten sind nicht an ein bestimmtes Gerät adressiert; sie sind an alle Geräte auf der Schleife gerichtet, die gerade als Listener adressiert sind. Deshalb muß der Controller sicherstellen, daß nur diejenigen Geräte, die die Daten empfangen sollen, die aktiven Listener sind, wenn die Meldung gesendet wird.

Hier nun ein Beispiel für eine mögliche Abfolge mit der SDA-Meldung:

Der Mikro schickt die SDA-Meldung an den Chip, wobei er ihm zuvor die Controllerfunktion zugeteilt und den aktiven Talker und aktiven Listener auf der Schleife bestimmt hat.

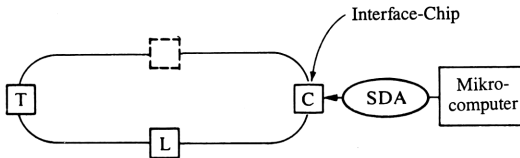


Abb. 4.10

Jetzt überträgt der Chip die SDA-Meldung, die an den Talker gerichtet ist, auf die Schleife.

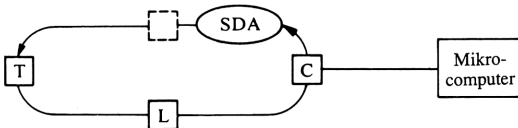


Abb. 4.11

Wenn der Talker die SDA-Meldung erhält, ersetzt er sie durch sein erstes Datenbyte (DAB1), statt sie weiterzugeben.

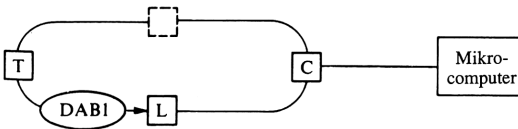


Abb. 4.12

Der aktive Listener macht eine Kopie des Datenbytes, um damit, je nach Art und Aufgabe des jeweiligen Gerätes, weiterzufahren. Wenn der Drucker z.B. der aktive Talker ist, gibt er die Meldung in den Druckpuffer. Dann wird das Datenbyte zurück zum Talker geschickt.

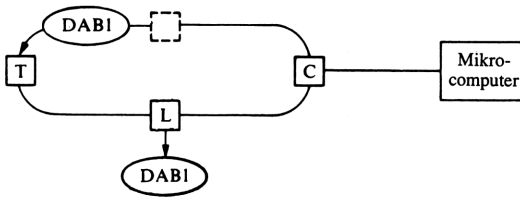


Abb. 4.13

Wenn der Talker das Datenbyte (DAB1) von der Schleife zurückerhält, kann er das nächste Datenbyte (DAB2) senden. Beachten Sie, daß der Talker auch jedes Datenbyte auf Fehler untersucht, wenn es zurückkommt.

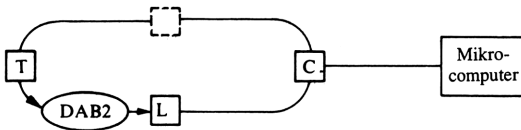


Abb. 4.14

Nachdem der Talker das letzte Datenbyte zurückerhalten hat, sendet er die ETO- (oder, wenn es Fehler gab, die ETE)-Meldung, um die Schleife zum Controller.

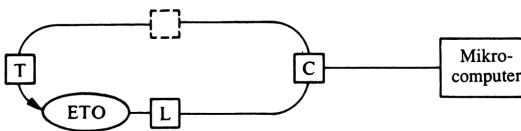


Abb. 4.15

Wenn der Controller (unser Interface-Chip) die ETO-Meldung vom Talker erhält, benachrichtigt er den Mikro, daß die Übertragung vollständig ist, und daß eine neue Schleifenoperation begonnen werden kann. Beachten Sie, daß die ETO-Meldung nicht die Rollen der Geräte auf der Schleife verändert. Sowohl der Talker als auch der Listener und Controller behalten ihre Rollen, bis sie eine andere zugewiesen bekommen.

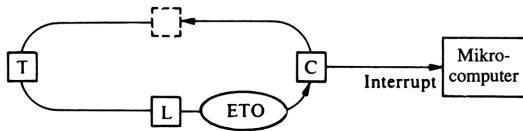


Abb. 4.16

Abschließend soll noch ein weiterer Aspekt erläutert werden. Der Interface-Chip sollte in der Lage sein, mehr als eine Rolle zur gleichen Zeit einnehmen zu können. Er kann z.B. Controller und gleichzeitig Listener sein. Wenn dies in der vorhergehenden Abb. 4.16 der Fall gewesen wäre, hätte der Interface-Chip, außer daß er die vom Talker empfangenen Datenbytes auf der Schleife weitergibt, diese zusätzlich an den Mikro übertragen müssen. Wenn der Talker z.B. ein Kassettenlaufwerk ist und Daten an den Listener schickt, der ein Drucker ist, kann es sein, daß auch der Mikro die Daten untersuchen will, während sie sich noch auf der Schleife befinden. In diesem Fall adressiert der Mikro, bevor er die SDA-Meldung sendet, einfach alle beide als Listener: sowohl den Drucker als auch den Interface-Chip.

Not Ready For Data (NRD)

Die vorhergehende Beschreibung der Send Data (SDA)-Meldung hat gezeigt, daß der Talker, wenn er diese Meldung erhalten hat, damit beginnt, Daten zu senden und damit bis zur vollständigen Ausführung fortfährt. Dies wird dadurch angezeigt, daß er die ETO- oder die ETA-Meldung schickt. Es gibt jedoch in jedem System, das aus mehreren Geräten besteht, oftmals Situationen, in denen es nötig ist, Prozesse, die gerade ablaufen, zu unterbrechen, um andere wichtigere Aktivitäten durchführen zu lassen. Dies ist genau der Fall, für die die Not Ready For Data (NRD)-Meldung vorgesehen ist.

Die NRD-Meldung gestattet dem Controller, eine Talker/Listener-Übertragung zu unterbrechen, damit die Schleife für eine andere Aufgabe zur Verfügung steht. Wie die Send Data-Meldung enthält auch die NRD-Meldung keine Adressinformation, denn sie ist immer an den jeweiligen aktiven Talker gerichtet (und es kann jeweils nur einen aktiven Talker auf der Schleife geben).

Wir schildern nun eine typische Befehlsfolge, die die NRD-Meldung einschließt, um zu zeigen, wie sie gebraucht wird.

Der Talker schickt gerade sein sechstes Datenbit (DAB6) auf die Schleife. Der Mikro benötigt die Schleife jedoch für eine andere Operation und schickt die NRD-Meldung an den Interface-Chip.

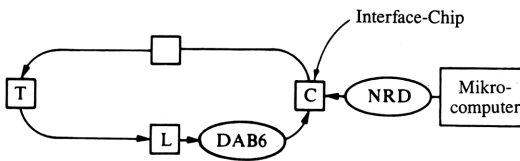


Abb. 4.17

Der Chip erhält nun das Datenbyte, hält das Datum (in einem internen Register) zurück und schickt stattdessen die NRD-Meldung auf die Schleife.

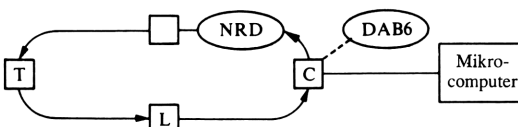


Abb. 4.18

Wenn der Talker statt des erwarteten Datenbytes die NRD-Meldung vom Controller erhält, muß er diese Tatsache registrieren und die NRD-Meldung auf der Schleife weitergeben.

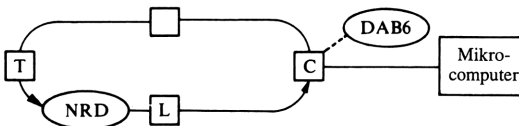


Abb. 4.19

Wenn der Chip die NRD-Meldung zurückerhält, schickt er das Datenbyte (DAB6), das er gespeichert hatte, weiter an den Talker.

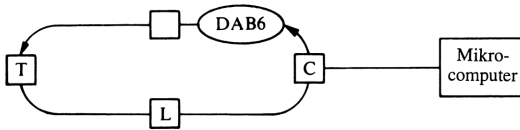


Abb. 4.20

Der Talker, der ja zuvor die NRD-Meldung erhalten hat, weiß nun, daß er sein vorhergehendes Datum zurückerhalten hat und muß nun die Datenübertragung abbrechen. Er schickt jetzt die ETO- (oder die ETE)-Meldung. Wenn noch ein Datum (z.B. DAB7,8 usw) gesendet werden müßte, sollte der Talker es für eine spätere Wiederaufnahme der Übertragung zurückhalten.

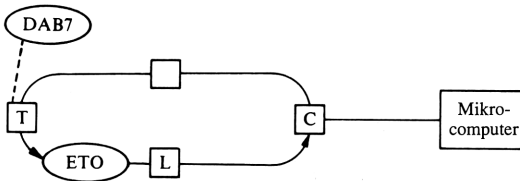


Abb. 4.21

Wenn der Mikro die Schleifenoperation, die wichtiger war, abgeschlossen hat, kann er im folgenden die unterbrochene Übertragung dort, wo sie unterbrochen wurde, wiederaufnehmen. Der Mikro gibt dann eine neue SDA (Send Data)-Meldung an den Interface-Chip, der die Meldung dann auf die Schleife weiterleitet.

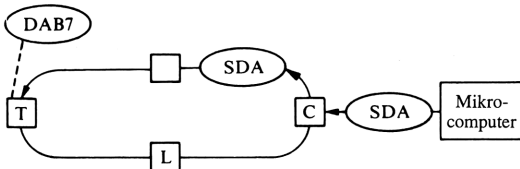


Abb. 4.22

Wenn der Talker diese neue SDA-Meldung erhält, sollte er die Datenübertragung an der Stelle, an der er sie unterbrochen hat, wieder aufnehmen, indem er das nächste Datenbyte (DAB7) auf die Schleife sendet. Wenn der Controller die

Rolle des Talkers für die andere Schleifenoperation verändern mußte, sollte der erste Talker seine vorherige Aktivität einschließlich seiner vorherigen Rolle an dem Punkt wieder aufnehmen, an dem sie unterbrochen wurde. Die Verantwortung für die Rollenzuweisung und die Aufrechterhaltung der Schleifenfunktionen während der Unterbrechung liegt jedoch allein beim Controller.

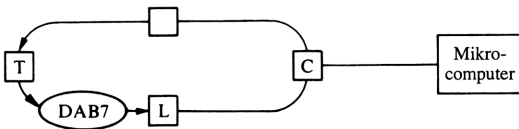


Abb. 4.23

Send Device ID (SDI)

Diesen Befehl benutzt der Controller, um über die Identität und die Eigenschaften des gegenwärtigen Talkers detaillierte Information zu erlangen. Der Befehl ist dem Send Status-Befehl (SST) in Kapitel 3 sehr ähnlich.

Wieder enthält die SDI-Meldung keine Adresse, weil sie an den aktiven Talker der Schleife gerichtet ist.

Der Talker geht mit dieser Meldung genauso um wie mit der SST-Meldung. Er gibt sie nicht auf der Schleife weiter, sondern er ersetzt sie durch Datenbytes und leitet diese weiter zum Controller. Wenn der Talker alle Identifikations-Datenbytes geschickt hat (und sie von ihrer Runde auf der Schleife zurückerhalten hat), sendet er die ETO- (oder, wenn es Fehler gab die ETE)-Meldung, um den Controller über die Beendigung der Übertragung zu informieren.

Genau wie bei der Send Status-Meldung ist die Interpretation der Identifikationsinformation Sache des Konstrukteurs; sie beinhaltet jedoch normalerweise die Initialen des Herstellers und die Modellnummer des Gerätes (z.B. HP3468A)

Take Control (TCT)

Diese Meldung erlaubt es dem aktiven Controller, ein anderes Gerät zum Controller zu machen und ihm die Kontrolle über die Schleife zu übertragen. Das Gerät, auf das

die Kontrolle übertragen werden soll, muß zunächst als aktiver Talker adressiert werden. Wenn der Talker dann im folgenden die TCT-Meldung erhält, nimmt er die Rolle des Controllers an. Die Befehlsfolge, die die Kontrollübertragung ermöglicht, soll hier dargestellt werden.

Zunächst schickt der Mikro die Talk Address (TAD)-Meldung an den Interface-Chip, der die Meldung dann auf die Schleife schickt. In Abb. 4.24 wird die TAD-Meldung auf der Schleife an Gerät 2 geschickt. Wenn Gerät 2 schon der aktive Talker ist, ist dieser Schritt natürlich überflüssig.

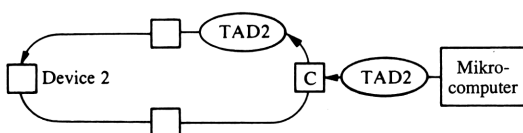


Abb. 4.24

Der Controller muß wie immer nach der TAD-Meldung eine RFC-Meldung schicken. Aber da dieser Schritt immer notwendig ist, wollen wir ihn nicht gesondert abbilden.

Nachdem die RFC-Meldung zum Controller zurückgekehrt ist, kann die TCT-Meldung gesendet werden. Da die TCT-Meldung immer an den aktiven Talker gerichtet ist und es immer nur einen aktiven Talker auf der Schleife gibt, enthält sie keine Adresse.

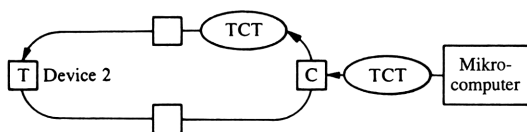


Abb. 4.25

Sobald der Talker die TCT-Meldung erhält, nimmt er die Rolle des Controllers an. Er gibt die TCT-Meldung nicht weiter auf der Schleife, sondern ersetzt sie durch seinen ersten Befehl (in Abb. 4.26 CMD).

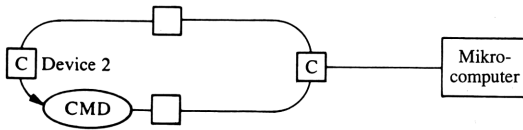


Abb. 4.26

Der ursprüngliche Controller, unser Interface-Chip, behält seine Rolle als Controller, bis er eine Meldung erhält. Wenn die erste Meldung, die er erhält, eine andere als die TCT-Meldung ist, weiß er, daß es einen neuen Controller gibt, und legt diese Rolle ab.

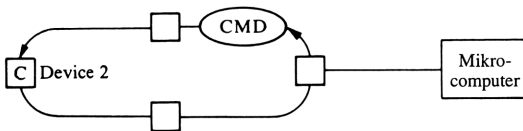


Abb. 4.27

Wenn die TCT-Meldung zum ursprünglichen Controller zurückkehrt, zeigt sie an, daß der neue Controller diese Rolle aus irgendeinem Grund nicht annehmen kann. In diesem Fall behält der ursprüngliche Controller seine Rolle bei.

Service-Anforderungs- und

Daten-Meldungen

In Kapitel 3 haben wir zwei Arten von Datenmeldungen beschrieben: DAB und END. Es gibt allerdings einen Aspekt bei diesen Meldungen, den wir noch nicht besprochen haben: das ist die Fähigkeit, eine Service-Anforderung an den Controller zu schicken, indem man Datenmeldungen benutzt. Der Mechanismus, der dazu benötigt wird, ist genau derselbe, der zu Anfang in diesem Kapitel in der Abteilung über die parallele Abfrage und den damit zusammenhängenden Meldungen erklärt wurde.

Um zu verstehen, wie dieser Mechanismus mit Datenmeldungen arbeitet, untersuchen wir das Bitmuster für diese Meldungen.

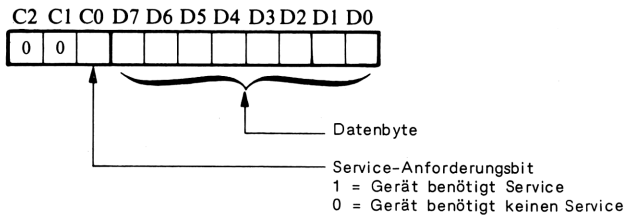


Abb. 4.28

Die acht niedrigstwertigen Bits enthalten das jeweilige Datum, das übertragen werden soll. Die beiden höchstwertigen Kontrollbits (C2, C1) zeigen an, daß es sich um eine Datenbyte-Meldung handelt. Bit C0 ist das Service-Anforderungsbit. Wenn dieses Bit auf 1 gesetzt ist, zeigt dies an, daß irgendein Gerät auf der Schleife den Eingriff des Controllers benötigt.

Der Controller überwacht jede Datenbyte-Meldung, die über die Schleife geht. Wenn er eine Meldung mit gesetztem Service-Anforderungsbit entdeckt, weiß er, daß ein Gerät seinen Eingriff verlangt, aber er weiß nicht, um welches Gerät es sich dabei handelt. Obwohl die Datenbyte-Meldungen vom aktiven Talker der Schleife erzeugt werden, kann jedes Gerät auf der Schleife das Service-Anforderungsbit in die Datenbytemeldung einfügen, wenn die Meldung das Gerät auf der Schleife passiert.

Diese Tatsache ist für die Geräte auf der Schleife eine effiziente Methode, laufende Operationen zu unterbrechen, wenn das System dies verlangt. Geräte müssen so nicht darauf warten, bis auf der Schleife Ruhe einkehrt, bevor sie die Aufmerksamkeit des Controllers auf sich lenken können. Sie müssen nur beim nächsten Mal, wenn eine Datenbyte-Meldung vorbeikommt, das Service-Anforderungsbit setzen.

Es ist wichtig festzuhalten, daß die Handlungen des Controllers, der entdeckt, daß ein Gerät Service anfordert, nicht festgelegt sind. Seine Reaktion hängt davon ab, wie das System konstruiert ist. Normalerweise wird man die gerade ablaufende Datenübertragung unterbrechen (außer sie ist sehr wichtig oder äußerst kritisch) und eine Prüfung der Geräte auf der Schleife veranlassen. Diese Prüfung kann je nach der Konstruktion des Systems aus einer seriellen oder parallelen Abfrage bestehen.

Auto-Address-Meldungen(29)

Jedes Gerät hat eine bestimmte Schleifenadresse, und bestimmte Meldungen werden an die Geräte gerichtet, indem man eine Adresse benutzt. In Kapitel 3 haben wir vorgeschlagen, den Geräten über Schalter oder durch vorherige Programmierung Adressen zuzuweisen.

HP-IL besitzt einen Mechanismus, der es dem Controller erlaubt, den Geräten auf der Schleife Adressen zuzuweisen. Bevor wir näher darauf eingehen, muß gesagt werden, daß die Geräte die Fähigkeit, Adressen unter Schleifenkontrolle zugewiesen zu bekommen, nicht unbedingt haben müssen. Sie können ein Gerät konstruieren, das eine vorher festgelegte festverdrahtete Adresse hat, und es kann ebenso innerhalb der Schleife betrieben werden. Die Schleife ist jedoch viel flexibler, wenn die Adressen dynamisch zugewiesen werden können.

Es gibt zwei automatische Adressierungsweisen, die auf der Schleife benutzt werden können: die erste automatische Adressierung erlaubt die Verbindung von bis zu 31 Geräten auf der Schleife, während die zweite oder erweiterte Methode den Betrieb von 961 Geräten auf der Schleife ermöglicht.

Primäre Adressierungsweise

Bei dieser Adressierungsart, die dazu dient, den Geräten auf der Schleife automatische Adressen zuzuweisen und die Schleife so zusammzusetzen, wie man sie benötigt, werden zwei Meldungen benutzt: die Auto-Address (AAD)-Meldung veranlaßt Geräte, die diese Fähigkeit haben, eine neue Geräteadresse anzunehmen und dann eine neue Adresse an das nächste Gerät auf der Schleife weiterzugeben. Dieser Ansatz ist sehr einfach, aber wirkungsvoll.

Das erste Gerät, das die AAD-Meldung erhält, nimmt die Adresse "1" an, erhöht sie um 1 und gibt diese Meldung dann an das nächste Gerät weiter. So werden die Geräte in ihrer Abfolge von 1 bis zum Maximum von 30 nummeriert, indem das erste Gerät (vom Controller aus gesehen) die Adresse 1 annimmt und das letzte Gerät vor dem Controller die höchste Adresse.

Wenn der Controller neue Adressen ausgeben muß, schickt

er die Auto-Address Unconfigure (AAU)-Meldung, um alle Geräte auf eine neue Adressenzuweisung vorzubereiten.

Sekundäre (erweiterte) Adressierungsweise

Mit dieser Adressierungsweise kann man bis zu 961 Geräten auf der Schleife Adressen zuweisen, indem man eine Zwei-Byte-Adresse benutzt. Um diesen Vorgang einzuleiten, sendet der Controller eine Meldung, die der AAD-Meldung ähnlich ist. Jedes Gerät erhält nicht nur eine Primäradresse, die von 0 bis 30 reichen kann, sondern auch eine Sekundäradresse, die ebenfalls von 0 bis 30 reichen kann. Um eines der Geräte zum aktiven Listener oder Talker zu machen, sendet der Controller die Primäradresse (TAD oder LAD) und läßt ihr die richtige Sekundäradresse folgen, die SAD genannt wird. Weil nicht viele Systeme diese Fähigkeit brauchen, wollen wir an dieser Stelle nicht näher darauf eingehen.

Beachten Sie, daß es auf der Schleife neben solchen Geräten, die Sekundäradressen annehmen können, auch solche geben kann, die nur die Primäradresse umfassen, oder vielleicht überhaupt keine automatische Adressierung durchführen können. Geräte, für die letzteres zutrifft, geben die Meldungen, die mit der automatischen Adressierung verbunden sind, einfach zum nächsten Gerät auf der Schleife weiter, ohne daß sie die Meldung in irgendeiner Art verändern.

Asynchrone Operationen

Alle Operationen, die hier und in den vorhergehenden Kapiteln erklärt wurden, sind insofern synchron, als daß jede Operation einzeln ausgeführt wird - es gibt nie zwei konkurrierende oder konfligierende Operationen auf der Schleife. Sie haben gesehen, daß es ordnungsgemäße Unterbrechungen einer laufenden Operation, wie z.B. einer Datenübertragung, geben kann, um den schnellen Eingriff des Controllers auf die speziellen Anforderungen eines bestimmten Gerätes zu gewährleisten. Diese Unterbrechungen waren aber eingebettet in den synchronen Fluß der gerade ablaufenden Operation, indem man Techniken wie das Setzen von Service-

Anforderungsbits oder das Einfügen der Not Ready for Data (NRD)-Meldung benutzt.

Diese Techniken, die dazu dienen, in weiterlaufende Operationen einzugreifen, folgen alle einer Regel: es gibt nie mehr als eine laufende Meldung, die zu einer bestimmten Zeit auf der Schleife ist. Es mag jedoch Fälle geben, in denen diese Techniken nicht ausreichend sind. Wenn z.B. das Gerät, das Daten über die Schleife überträgt, extrem langsam ist, und ein anderes Gerät auf der Schleife dringend Antwort auf seine Service-Anforderung benötigt, müssen andere Techniken angewendet werden.

Für diese möglicherweise problematische Situation haben die Konstrukteure des HP-IL eine Lösung vorgesehen. Unter besonderen Umständen ist es möglich, daß mehrere Meldungen gleichzeitig über die Schleife gehen. Die Arbeitsweise, die hier angewendet wird, nennt man "asynchrone Operationen".

Beachten Sie hierbei jedoch, daß diese asynchrone Operationsweise nur dann angewendet werden soll, wenn es für Ihr System absolut notwendig ist und auch dann nur mit größter Vorsicht, denn die Logik in den System-Geräten kann dadurch stark verkompliziert werden. Zusätzlich kann diese Art der Operation die ansonsten extrem sichere Arbeitsweise des Systems gefährden. Während einer asynchronen Operation ist es viel schwieriger, die Vollständigkeit der Daten sicherzustellen. Doch wir wollen dieses Thema hier nicht weiter vertiefen, da es ebenso wie die automatische Adressierungsweise nur in ganz wenigen Systemen benötigt wird.

Kapitel 5

Einige typische Meldungsabfolgen

Kapitel 3 und 4 behandelten verschiedene Meldungen, die über die Schleife gehen. Dieses Kapitel soll die Illustration und Darstellung einer Anzahl von Befehlsabfolgen geben, die die bis jetzt vermittelte Information über das HP-IL zusammenfaßt. Weil wir bis zu dieser Stelle schon fast alle Meldungen, die über die Schleife gehen können, behandelt haben, ist Ihnen auch die Nomenklatur bekannt. Wenn Sie über irgendeine Funktion oder über den Gebrauch einer bestimmten Meldung im Unklaren sind, schlagen Sie einfach zum Kapitel 3 oder 4 zurück, wo sie zuerst gebraucht wurde, oder benutzen Sie das Glossar, das alle HP-IL Meldungen noch einmal aufführt.

Selbstverständlich können wir nicht alle Meldungsabfolgen wiedergeben, die über die Schleife gehen können. Außerdem sind die dargestellten Abfolgen weder der einzig mögliche noch der notwendig beste Weg, eine bestimmte Aufgabe auszuführen. Stattdessen wollen wir Ihnen die Abfolgen vorstellen, damit Sie besser verstehen, wie die Meldungen benutzt werden und wie die Geräte auf der Schleife zusammenarbeiten. Ohne Zweifel werden Sie, wenn Sie mit dem HP-IL vertrauter sind, in der Lage sein, selbst effizientere und kreativere Lösungen für diese Aufgaben zu finden.

 Die Befehlsabfolge beim Einschalten

In fast jedem System, das man konstruiert, muß man Ausgangs-Arbeitsbedingungen herstellen, wenn es eingeschaltet wird. Die Aufgaben, die dazu normalerweise im HP-IL System zuerst erfüllt werden müssen, umfassen die Festlegung, welches Gerät der Controller ist, die Feststellung, wann alle Geräte auf der Schleife bereit sind, Operationen auszuführen, und die Initialisierung dieser Geräte.

Hier ist ein Beispiel einer solchen Meldungsabfolge:

- IFC Als Ausgangspunkt sendet der Systemcontroller in gleichmäßigen, langsamen Zeitabständen die Interface Clear-Meldung über die Schleife. Er fährt damit solange fort, bis er die IFC-Meldung von der Schleife zurückerhält. Wenn die IFC-Meldung wieder beim Controller angelangt ist, zeigt dies an, daß alle Geräte auf der Schleife eingeschaltet sind, Meldungen erhalten können und sich gerade in ihre Ausgangsstellung versetzen.
- IFC
- RFC Wenn die IFC-Meldung zurückgekommen ist, sendet der Controller die Ready For Command-Meldung. Die Geräte auf der Schleife geben diese Meldung erst dann weiter, wenn sie sich in ihrer Ausgangsstellung befinden und in der Lage sind, den nächsten Befehl des Controllers zu empfangen. Beachten Sie, daß die RFC-Meldung nur einmal gesendet wird, während die IFC-Meldung solange kontinuierlich gesendet wird, bis sie wieder beim Controller angelangt ist.

Wenn die RFC-Meldung wieder beim Controller ist, können andere (auch gerätespezifische) Initialisierungsprozeduren eingeleitet werden, indem das Standard-Schleifenprotokoll benutzt wird.

 Initialisierung eines Systems ohne Controller

HP-IL gestattet das Betreiben von Systemen ohne Controller. Dies ist die elementare "freundliche" Methode, bei der

jedes Gerät auf der Schleife vom anderen genau weiß, was es zu erwarten hat und wofür es Sorge tragen muß. Bei bestimmten Anwendungen ist die Konstruktion dieser "freundlichen" Geräte sinnvoll; so für einen Drucker, der z.B. Anzeigen von einem Voltmeter protokolliert.

Wir wollen nun betrachten, wie sich die Initialisierung eines solchen Systems von unserem vorherigen unterscheidet.

Wenn es auf der Schleife keinen Controller gibt, muß sich ein Gerät zu Anfang in die Rolle des Talkers versetzen. In diesem Fall muß der Talker, obwohl er nicht die Aufgabe eines Schleifen- bzw. System-Controllers erfüllen muß, die Aktivität auf der Schleife einleiten.

DAB1 Zunächst schickt der Talker sein erstes Datenbyte auf die Schleife. Das Gerät muß in langsamen, gleichmäßigen Zeitabständen damit fortfahren, dasselbe Datenbyte auf die Schleife zu senden, bis der Talker es wieder empfängt. Wenn das erste Datenbyte zurückkommt, zeigt es an, daß die Geräte auf der Schleife eingeschaltet sind, Meldungen erhalten und weiterübertragen können.

So ist diese Datenbyte-Meldung gleichwertig mit der IFC- und der RFC-Meldung in den gewöhnlichen Systemen.

DAB2 Wenn der Talker der Schleife seine erste Datenbyte-Meldung zurückerhalten hat, beginnt er, den
DAB3 Rest der Datenkette in normaler Geschwindigkeit zu senden. Wenn das letzte Datenbyte gesendet
DABn wurde und die Übertragung vollständig ist, kann der Talker damit beginnen, die nächste Datenkette zu senden.

In diesem Fall ist auch keine EOT-Meldung notwendig, denn die Geräte auf einer solchen "freundlichen" Schleife wissen, daß es nur Talker und Listener auf der Schleife gibt. Deshalb ist sowohl das Schleifenprotokoll als auch die Ordnung auf der Schleife stark vereinfacht. Die Übertragung dieser Meldungen kann auf diese relativ ungeordnete Weise unbegrenzt weiterlaufen, bis das System erfordert, daß die Geräte auf Meldungen antworten müssen, die über diejenigen des einfachen Talker und Listener hinausgehen.

Die Abfolge einer Datenübertragung

Die allgemeinste Aufgabe eines HP-IL Systems ist die der Datenübertragung um die Schleife. Wenn die Rollen festgelegt sind und die Ordnung des Systems hergestellt ist, beginnt man mit der Übertragung der Informationen. Hier ist eine der möglichen Abfolgen, die zu diesem Zweck benutzt werden kann:

- UNL Wenn notwendig, wird der Controller zunächst die Unlisten-Meldung senden, um alle anderen Geräte, die als Listener adressiert wurden (und diese Datenübertragung erhalten sollen), davon auszuschließen.
- RFC Wie allen Befehls-Meldungen folgt auch der UNL-Meldung eine Ready For Command-Meldung.
- LAD1 Dann sendet der Controller die angemessene Listen Address-Meldung, um ein Gerät zum Listener der Schleife zu machen. Der Controller kann mehr als ein Gerät als Listener bestimmen. Um die einzelnen Geräte zu adressieren, muß er pro Gerät jeweils eine LAD-Meldung senden, und auf jede LAD-Meldung muß eine RFC-Meldung folgen.

Nachdem die jeweiligen Geräte auf der Schleife zu Listnern bestimmt worden sind, schickt der Controller eine Meldung an den Talker, der dann Daten an die Schleife abgeben soll.

- TAD Der Controller sendet die Talk Address-Meldung, um ein Gerät auf der Schleife zu befähigen, Daten an die Listener auf der Schleife zu senden.
- RFC Wenn der Controller die RFC-Meldung von der Schleife zurückerhält, schickt er die Send Data (SDA)-Meldung. Wenn das Gerät, das als Talker adressiert wurde, die SDA-Meldung erhält, ersetzt
- DAB1 er sie durch sein eigenes erstes Datenbyte (DAB1).
- DAB2 Wenn der Talker sein letztes Datenbyte zurücker-
- DAB3 halten hat, muß er die End of Transmission (ETO)-
- DABn Meldung senden, um anzuzeigen, daß alle Daten,
- ETO die er geschickt hat, zurückgekehrt und korrekt auf Fehler untersucht worden sind.

Wenn der Controller am Ende der Datenübertragung die ETO-Meldung vom Talker erhält, beginnt er mit der nächsten Schleifenoperation.

Beachten Sie, daß durch die Beendigung der Datenübertragung die Rollen der Geräte auf der Schleife nicht verändert wurden. Das Gerät, das Daten an die Listener auf der Schleife gesendet hat, ist noch immer der Talker, und die Empfänger der Daten sind noch immer als Listener adressiert.

Wenn ein Schleifengerät seine Rolle ändern soll, muß der Controller Meldungen senden, die die Rollen nach Erhalt der ETO-Meldung neu zuweisen. (Sie erinnern sich, daß der Controller die End of Transmission-Meldung nicht weiterüberträgt, gleichgültig, ob es eine ETO- oder eine ETE-Meldung war. Er beginnt nach Erhalt der Meldung mit der nächsten Schleifenoperation.)

Unterbrechung einer Datenübertragung

Diesen Punkt haben wir bereits in Kapitel 4 erwähnt. Dort wurde klargestellt, daß der Schleifen-Controller in der Lage sein muß, einzuspringen, wenn ein langsam arbeitendes Gerät die Schleife für sich allein in Anspruch nimmt, während ein anderes Gerät einen dringenderen Bedarf nach einem Zugang zur Schleife hat. Obwohl das HP-IL eigentlich speziell für relativ langsame Systeme konstruiert wurde, gibt es keinen Grund, nicht auch Vorsorge zur Erzielung eines maximalen Ablaufs innerhalb dieser Einschränkungen zu treffen.

Die Unterbrechung einer Datenübertragung ist eine der Möglichkeiten, so viel wie möglich aus der Schleife herauszuholen (soweit es die einfache Datenübertragung betrifft). Die Unterbrechung einer Datenübertragung, die zwischen dem Talker der Schleife und einem oder mehreren Listnern im Gange ist, macht den Eingriff des Schleifen-Controllers nötig. Nachfolgend finden Sie eine Befehlsfolge, um eine Unterbrechung durchzuführen:

DAB5 Nehmen wir an, der Talker habe gerade sein fünftes Datenbyte auf die Schleife geschickt. Jetzt beschließt der Controller, daß wichtigere Aufgaben verfolgt werden müssen, und hält das fünfte Datenbyte (DAB5) des Talkers zurück.

- NRD Der Controller ersetzt das fünfte Datenbyte (DAB5) durch die Not Ready for Data (NRD)-Meldung. Sobald der aktive Talker auf der Schleife statt des Datenbytes, das er gesendet hat, diese Meldung erhält, beendet er die Datenübertragung und gibt die NRD-Meldung an den Controller weiter. Diese Antwort unterstellt auf der Seite des Talkers ein hohes Maß an Verantwortung. Er muß erkennen, daß die NRD-Meldung anzeigt, daß er seine Datenübertragung unterbrechen muß. Wenn der Controller die NRD-Meldung zurückerhält, gibt er das festgehaltene Datenbyte wieder frei.
- DAB5 Wenn der Talker zuletzt sein "verlorenes" Datenbyte zurückerhält, muß er nun eine ETO-Meldung
ETO auf die Schleife schicken.
- CMD Wenn der Controller die ETO-Meldung des Talkers erhalten hat, kann er die Operation, wegen der er die Datenübertragung unterbrochen hat, beginnen, indem er seinen nächsten Befehl (CMD) sendet.

Wenn der Controller nach der erneuten Adressierung des Talkers, der während der zuvor unterbrochenen Datenübertragung aktiv war, die Send Data (SDA)-Meldung sendet, wird die Datenübertragung an der Stelle, an der sie unterbrochen wurde, wieder aufgenommen. So wird in unserem Beispiel der Talker, wenn er beim Erhalt der SDA-Meldung wieder der aktive Talker ist, auf die Meldung antworten, indem er sein nächstes Datenbyte schickt.

Eine serielle Abfrageoperation

Serielle Abfragen wird vom Schleifen-Controller angewandt, um die Identität desjenigen Gerätes (oder derjenigen Geräte) festzustellen, die zuvor durch das Setzen des Service-Anforderungsbits signalisierten, daß sie Unterstützung benötigen. Wie schon vorher erwähnt, kann ein Gerät auf der Schleife dadurch, daß es das Service-Anforderungsbit in einer beliebigen Datenbyte-Meldung setzt, anzeigen, daß es Hilfe benötigt - gleichgültig, wer die Übertragung des Datenbytes veranlaßt hat.

Wenn der Controller sieht, daß eine Meldung auf der

Schleife das Service-Anforderungsbit gesetzt hat, kann er entweder sofort antworten (indem er z.B. die vorher beschriebene NRD-Abfolge benutzt, um die gerade ablaufende Übertragung zu unterbrechen, oder er wartet geduldig, bis die gegenwärtige Schleifenoperation normal beendet ist. In beiden Fällen kann der Controller eine Abfolge wie die folgende benutzen, um herauszufinden, welches Gerät seinen Eingriff angefordert hat.

- UNL Zunächst sendet der Controller den Unlisten Befehl, um zu verhindern, daß alle Listener auf
- RFC der Schleife automatisch die Statusbytes erhalten, die von den abgefragten Geräten auf die Schleife gesendet werden sollen.(Hierauf folgt wie immer die RFC-Meldung).
- TAD1 Der Controller adressiert das erste Gerät auf der Schleife als Talker.
- RFC (Natürlich kann der Controller wissen, daß nur bestimmte Geräte auf der Schleife in der Lage sind, Service anzufordern.)
- SST Der Controller sendet die Send Status (SST)-Meldung, die der adressierte Talker auf der Schleife
- DAB durch sein Statusbyte oder mehrere Bytes (DAB's) ersetzt.
- ETO Nachdem der Talker sein letztes Statusbyte zurückgehalten hat, muß er die ETO-Meldung senden.
- TAD2 Dies zeigt dem Controller die Beendigung der Statusorientierten Übertragung an, und er ersetzt die ETO-Meldung durch eine Talk Address-Meldung
- RFC (in unserem Beispiel TAD2), die an das nächste Gerät an der Schleife gerichtet ist, das in der Lage ist, Service anzufordern.
- SST Dieselbe Abfolge, die Geräte als Talker zu adressieren und sie dann dazu zu veranlassen, ihren
- DAB Status zu senden, wird für alle einzelnen Geräte auf der Schleife wiederholt. Sobald die Status-
- ETO bytes von jedem der abgefragten Geräte wieder zum Controller zurückkehren, muß dieser sie untersuchen, um festzustellen, welches Gerät gerade Service benötigt.
- Wenn ein bestimmtes Gerät statt eines Datenbytes

einfach die SST-Meldung zurücksendet, weiß der Controller, daß dieses Gerät die serielle Abfrage nicht beherrscht (und somit auch nicht das Gerät sein kann, daß seinen Eingriff angefordert hat). Es ist einleuchtend, daß die serielle Abfrage der Datenübertragung sehr ähnlich ist. Der Hauptunterschied ist, daß der Controller die "Daten" (Statusbytes) erhält, ohne als Listener adressiert worden zu sein.

Parallele Abfrage und Konfiguration

Wie in den vorhergehenden Kapiteln erklärt wurde, kann das serielle Abfragen aller Geräte auf der Schleife oder auch nur derjenigen Geräte, die dazu in der Lage sind, eine geraume Zeit in Anspruch nehmen. Wenn Ihr System eine zeitsparendere Methode benötigt, können Sie die in der HP-IL Spezifikation definierte parallele Abfragefunktion in Ihr System einfügen.

Natürlich sind die Vorteile, die Sie durch die Implementierung der parallelen Abfrage-Fähigkeit in Ihre Schleifen-geräte erhalten, nicht kostenlos. Dazu ist es notwendig, daß die Geräte eine zusätzliche Intelligenz oder Logik besitzen.

Lassen Sie uns nun zeigen, wie die parallele Abfrage funktioniert, damit sie ein Gefühl dafür bekommen, um wieviel die Logik komplizierter wird, die für die Schleifenoperation notwendig ist.

UNL	Zuerst sendet der Controller den Unlisten-Befehl (UNL). Dieser verhindert, daß unerwünschte Geräte
RFC	auf die parallele Abfrage antworten. Alternativ
PPU	oder zusätzlich dazu kann der Controller die Parallel Poll Unconfigure-Meldung (PPU) senden. Dadurch wird jede parallele Abfragekonfiguration, die vorher auf der Schleife etabliert war, zurückgesetzt.

Jetzt kann der Controller damit beginnen, die Schleife für die parallele Abfrage zu konfigurieren. Dies tut er, indem er jedes einzelne Gerät adressiert, von dem der Controller wünscht, daß es die parallele Abfrage durchführt.

- LAD1 Zunächst adressiert der Controller das gewünschte Gerät (in unserem Beispiel Gerät 1) als Listener, RFC
so daß es im folgenden den Parallel Poll Enable (PPE) Befehl erhalten kann.
- PPE13 Dann sendet der Controller den Parallel Poll Enable-Befehl an den adressierten Listener. Die PPE-Meldung enthält eine Information, die das Bit bezeichnet, das das Gerät in der nachfolgenden RFC
Abfrage/Antwort benutzen soll, sowie die Polarität dieser Antwort (1 oder 0). In diesem Beispiel soll der adressierte Listener (Gerät 1) Bit 3 für die Abfrage/Antwort benutzen und dieses Bit auf "1" setzen, wenn es Service benötigt.
- UNL Jetzt muß der Controller die Adressierung von Gerät 1 zurücknehmen, so daß er die Abfrage-Information an ein anderes Gerät auf der Schleife senden kann.

Diese Abfolge (LAD, PPE, UNL) muß für jedes einzelne Gerät auf der Schleife, das die parallele Abfrage durchführen soll, wiederholt werden. Nachdem alle erwünschten Geräte auf der Schleife aktiviert worden sind, kann man die Schleife als "konfiguriert" betrachten, und die normalen Schleifenoperationen können fortgesetzt werden. Obwohl diese Prozedur langwierig erscheinen mag, sollten Sie bedenken, daß sie normalerweise nicht wiederholt angewendet werden muß. Außerdem ist die tatsächliche Abfrageoperation bedeutend vereinfacht, wenn diese Konfiguration einmal vorgenommen wurde.

- IDY Der Controller führt die parallele Abfrage durch, indem er die Identify-Meldung (IDY) sendet. Wenn diejenigen Geräte, deren parallele Abfrage-Fähigkeit zuvor aktiviert wurde, diese Meldung erhalten, setzen sie das entsprechende Bit in der IDY-Meldung in der Polarität, die während der Konfiguration festgelegt wurde. Dann geben sie die IDY-Meldung weiter um die Schleife. Geräte, die die Fähigkeit zur parallelen Abfrage nicht besitzen oder deren Fähigkeit nicht aktiviert wurde, geben die IDY-Meldung weiter, ohne sie zu verändern.

Wenn der Controller die IDY-Meldung von der Schleife zurückerhält, muß er nur den Zustand der Datenbits der Meldung untersuchen, um festzustellen, welches Gerät (oder welche Geräte) auf der Schleife Service benötigen. Wenn nur ein Gerät einem Datenbit zugewiesen ist, können schnell bis zu acht Geräte vom Controller abgefragt werden.

 Automatische Adresszuweisung

Soweit es den Controller betrifft, ist die automatische Adresszuweisung eine sehr einfache Aufgabe. Allerdings ist dazu eine zusätzliche Logik für jedes der Geräte notwendig, das auf die automatische Adreßzuweisung antworten soll.

- AAU Zunächst muß der Controller die Auto Address Un-configure-Meldung auf die Schleife schicken, um jede vorherige Adresse zu löschen. Dies bereitet
- RFC die Geräte auf der Schleife außerdem auf den Empfang ihrer neuen Adresse vor. Wenn der Controller die RFC-Meldung von der Schleife zurückerhält, kann er mit der Adressenzuweisung beginnen.
- AAD1 Zunächst sendet der Controller die erste Auto-Adresse auf die Schleife, und das erste Gerät, das
- AAD2 die Meldung erhält, nimmt die Adresse 1 als seine Adresse an, erhöht die Adresse der Meldung und
- AAD3 schickt sie an das nächste Gerät auf der Schleife. Das nächste Gerät wiederholt diesen Vorgang. Es nimmt die Adresse 2 an, erhöht die Adresse und überträgt die Adreßmeldung weiter.
- AADn Dieser Prozeß geht solange weiter, bis die Auto Address-Meldung zum Controller zurückkehrt. Wenn der Controller die AAD-Meldung zurückerhält, kann er sie untersuchen, um festzustellen, wievielen Geräten Adressen zugewiesen worden sind. Die AAD-Meldung enthält dann die höchste zugewiesene Adresse plus 1.

Wenn die AAD-Meldung, die zum Controller zurückkehrt, die Adresse 31 enthält, kann dies bedeuten, daß zu viele Geräte auf der Schleife sind. Um dies herauszufinden, kann der Controller die AAD1-Meldung nochmals senden. Weil Geräte erst dann auf eine andere Adressenzuweisung antworten, wenn

zuvor die AAU-Meldung gesendet wurde, muß dieser zweite Versuch der Adressenzuweisung ignoriert werden und die AAD1-Meldung kehrt unverändert zurück. Wenn die AAD-Meldung jedoch erhöht worden ist, wenn sie zum Controller zurückkehrt, wird damit angezeigt, daß zu viele Geräte an die Schleife angeschlossen sind. Deshalb muß eine korrigierende Handlung von außen eingeleitet werden, denn wenn zu viele Geräte an die Schleife angeschlossen sind, kann sie nicht korrekt arbeiten.

Übertragung der Kontrolle auf einen anderen Controller

Wie bereits erwähnt, kann es zu einem bestimmten Zeitpunkt nur einen aktiven Controller auf der Schleife geben. Außerdem muß ein Gerät System-Controller sein, der z.B. für die Initialisierung der Schleife beim Einschalten verantwortlich ist. Mehr als ein Gerät kann in der Lage sein, die Rolle des Schleifen-Controllers zu übernehmen; deshalb kann die Kontrolle der Schleife von Gerät zu Gerät weitergegeben werden. Dieser Vorgang ist ziemlich einfach.

- TADn Der gegenwärtig aktive Talker adressiert zunächst das Gerät, das später Controller werden soll, als Talker, indem er die Talk Address-Meldung (TAD) sendet. Ist das gewünschte Gerät bereits der aktive Talker, so erübrigt sich dieser Schritt.
- TCT Jetzt sendet der gegenwärtige Controller die Take Control (TCT)-Meldung. Das Gerät, das als Talker adressiert ist, gibt diese Meldung nicht über die Schleife an den Controller weiter, sondern übernimmt sofort die Rolle des Controllers und ersetzt die Meldung durch seinen ersten Befehl.

Wenn der vorherige Controller eine andere Meldung als die TCT-Meldung zurückerhält, weiß er, daß ein anderes Gerät nun der aktive Schleifen-Controller ist. Sollte der vorherige Controller die TCT-Meldung unverändert zurückerhalten, wird damit angezeigt, daß der adressierte Talker die Rolle des Controllers nicht übernehmen kann. Der vorherige Controller muß daher die Kontrolle der Schleife weiterbehalten und die notwendigen Schritte übernehmen.

Kapitel 6

Die Interface-Funktionen

Wir haben uns in den letzten Kapiteln schon einige Male auf die Interface-Funktionen bezogen, die in der HP-IL Spezifikation bestimmt werden. Bis jetzt haben wir diese Funktionen jedoch noch nicht näher definiert oder erläutert. Nun wollen wir uns eingehender mit der HP-IL Spezifikation beschäftigen, um zu erklären, in welcher Weise die Interface-Funktionen alle schleifenbezogenen Eigenschaften der Geräte bestimmen.

Wir wollen damit beginnen, alle Interface-Funktionen aufzuzählen. Sie gliedern sich in drei Kategorien: primäre Funktionen, Geräte-Kontroll-Funktionen und Adressen/Status/Service-Anforderungs-Funktionen

Primäre Interface-Funktionen:

- Receiver
- Acceptor Handshake(30)
- Driver
- Listener
- Source Handshake(31)
- Talker
- Controller

Geräte-Kontroll-Interface-Funktionen:

- Power-down
- Device Clear
- Device Trigger(32)
- Remote Local

Adressen/Status/Service-Anforderungs-Funktionen

- Parallele Poll (Parallele Abfrage)
- Service-Request (Service-Anforderung)
- Automatic Address (Automatische Adressierung)
- Auto-Extended Address (Automatisch erweiterte Adresse)
- Auto-Multiple Address (Automatische Mehrfach-Adresse)
- Device Dependent(33)

Die Namen einiger dieser Interface-Funktionen sind Ihnen zweifellos bekannt. Talker, Listener und Controller wurden zusammen mit den Service-Anforderungen und der automatischen Adressierung in den vorhergehenden Kapiteln besprochen. Meist haben wir diese Bezeichnungen im Zusammenhang mit den Rollen der Geräte oder den Fähigkeiten der Geräte gebraucht. Die vollständige Bestimmung der Rolle oder der Fähigkeit eines Gerätes ist geleistet, wenn man festlegt, welche der 17 Interface-Funktionen im Gerät implementiert sind.

Einige der Funktionen müssen in jedem Gerät auf der Schleife enthalten sein. Andere Funktionen, wie die des Talkers oder Listeners, müssen nur dann vorhanden sein, wenn das Gerät eine dieser primären Rollen übernehmen soll. Zusätzlich dazu gibt es bei den meisten Interface-Funktionen Untergruppen, die dann die Charakteristika dieses Gerätes noch näher bestimmen.

Grundsätzlich müssen Sie entscheiden, welche Fähigkeiten ein Gerät auf der Schleife haben soll, und es dann so konstruieren, daß die Interface-Funktionen implementiert werden, die die gewünschte Fähigkeit beinhalten. Man kann diesen Vorgang mit der Entscheidung vergleichen, die man treffen muß, wenn man eine bestimmte Ausstattung bei einem neu zu erwerbenden Auto auswählt. Teile dieser Ausstattung wie Räder, Motor und Bremsen sind notwendig, damit der Wagen überhaupt fahrtüchtig ist. Jedoch gibt es auch inner-

halb dieser notwendigen Fähigkeiten die Möglichkeit der Auswahl, wie z.B. den Motortyp (4-Zylinder, 6-Zylinder), die Bereifung (Diagonal- oder Gürtelreifen) usw. Ein Teil der Ausstattung, wie die Musikanlage, liegt völlig in Ihrem Belieben; auch hier können Sie eine unterschiedliche Wahl treffen (einfaches Radio oder Stereo-Kassetten Kombination)

Die Entscheidung darüber, welche Funktionen und Möglichkeiten Sie von Ihrem Gerät erwarten, ist auch ein Kostenvergleich, ähnlich dem, den Sie machen, wenn Sie ein neues Auto kaufen wollen: Leistung und Komfort im Vergleich zu den entstehenden Kosten. Für jede Interface-Funktion, die Ihre Geräte beinhalten sollen, indem sie ihm mehr Intelligenz und Logik verleihen, müssen Sie mehr bezahlen.

Im Nachfolgenden wollen wir Ihnen eine Beschreibung der Interface-Funktionen geben, indem wir mit den primären Funktionen beginnen, die am meisten benötigt werden.

Denken Sie daran, daß sich die folgende Darstellung oft auf eine Interface-Funktion bezieht, die mit einer anderen zusammenhängt. Diese Art der Darstellung mag zuerst verwirrend erscheinen, denn die gesamte bisher erläuterte Kommunikation bezog sich jeweils auf ein einzelnes Gerät auf der Schleife. Diese Darstellungsweise ist jedoch notwendig, weil die Fähigkeiten und Möglichkeiten eines Gerätes stets nur in der Zusammenarbeit mit einem anderen Gerät sinnvoll sind.

In Bezug auf unsere Analogie zum Automobil kann man sagen, daß die Wahl eines automatischen Getriebes mit der Funktion des Motors, die Geschwindigkeit des Wagens zu bestimmen, zusammenhängt. Das Getriebe arbeitet nicht in einem Vakuum. Wenn man sich ein Getriebe ohne Wagen und Motor, als selbständige Einheit vorstellt, die sich über die Landstraße bewegt, ergibt dies keinen Sinn. Ebenso sind viele Interface-Funktionen wechselseitig voneinander abhängig. Jede führt eine genau festgelegte Funktion innerhalb eines Gerätes/Systems aus, so daß man sich dann nach wirtschaftlichen Überlegungen nur die Möglichkeit aussuchen kann, die man wirklich benötigt. Damit das Gerät auf der Schleife arbeiten kann, müssen alle diese Möglichkeiten eng zusammenarbeiten.

Die primären Interface-Funktionen

Es sind diejenigen Interface-Funktionen, die entweder für alle Geräte auf der Schleife notwendig sind, oder die eng mit den drei primären Rollen: Talker, Listener und Controller auf der Schleife verbunden sind.

Receiver (R)

Alle Geräte müssen die Receiver-Interface-Funktion beinhalten, denn jedes Gerät muß in der Lage sein, Meldungen von der Schleife zu empfangen. Die Receiver-Interface-Funktion hat die Aufgabe, hereinkommende Meldungen zu dekodieren und festzustellen, ob dieses Gerät in irgendeiner Weise mit der Meldung zu tun hat, oder ob sie einfach sofort weiterübertragen werden soll.

Die weitere Übertragung der Meldung wird jedoch nicht von der Receiver-Funktion selbst ausgeführt. Stattdessen informiert sie die Driver-Interface-Funktion (die später beschrieben wird) darüber, daß die Meldung rückübertragen werden soll.

Die Receiver-Funktion arbeitet eng mit der Driver-Funktion zusammen, denn bevor sie dem Driver meldet, daß eine weitere Meldung gesendet werden soll, muß sie warten, bis sie vom Driver informiert wird, daß alle vorhergehenden Meldungen abgeschickt wurden.

Da die Mehrzahl der Meldungen, die ein Gerät auf der Schleife erhält, für ein anderes Gerät bestimmt sind, hat die Receiver-Funktion zumeist die Aufgabe, die hereinkommenden Meldungen hinsichtlich des Gerätes, für das sie bestimmt sind, zu untersuchen und die Driver-Funktion darüber zu informieren, daß sie eine neue Meldung auf die Schleife schicken soll.

Da die Meldungen auf der Schleife bit-seriell empfangen werden, und da die drei höchstwertigen Bits (die ersten, die empfangen werden) den Typ der empfangenen Meldung benennen, kann man normalerweise schon bevor die restlichen Bits empfangen wurden, feststellen, ob die Meldung für dieses Gerät bestimmt ist. Wenn Ihr Gerät z.B. eine Befehlsmeldung erhält und diese Meldung nicht gerade selbst ausgesendet hat, weiß es, daß diese Meldung sofort weiter-

übertragen werden muß. (Wie Sie sich erinnern, muß auf jede Befehlsmeldung eine RFC-Meldung folgen, um festzustellen, ob der vorher gesendete Befehl erfüllt wurde).

Beachten Sie, daß es bei der Receiver-Funktion keine Untergruppen gibt. Die Geräte müssen alle diese primäre Funktion beinhalten.

Acceptor Handshake (AH)

Alle Geräte auf der Schleife müssen diese Funktion beinhalten. Die AH-Interface-Funktion ermöglicht es einem Gerät, von der Receiver-Funktion(einer anderen notwendigen Funktion) Meldungen zu erhalten, die an dieses bestimmte Gerät gerichtet sind. Die AH-Funktion muß ebenso festlegen, ob die empfangene Meldung nach der Dekodierung weiterübertragen werden muß. Beachten Sie, daß diese Dekodierung über die hinausgeht, die von der Receiver-Funktion ausgeführt wird. Der Receiver legt fest, ob das Gerät mit dieser empfangenen Meldung überhaupt etwas zu tun hat, und wenn nicht, informiert er die anderen Funktionen, daß die Meldung unmittelbar weiterübertragen werden muß.

Die Acceptor Handshake-Funktion muß die Meldung weiterdekodieren, um festzustellen, wann die Meldung vollständig (über die drei ersten Bits hinaus) dekodiert wurde, und ob der Typ der Meldung ihre Weiterübertragung erfordert. Wenn dieses Gerät z.B. die Meldung gesendet hat und es eine Datenmeldung war, muß sie nicht weiterübertragen werden.

Die Acceptor Handshake-Funktion hat keine Untergruppen. Alle Geräte müssen diese Fähigkeit vollständig beinhalten.

Driver (D)

Weil der Driver dem Gerät die Fähigkeit verleiht, Meldungen auf die Schleife zu übertragen, müssen alle Geräte die Driver-Interface-Funktion besitzen. Gleichgültig, welche primäre Rolle ein Gerät auf der Schleife spielt, muß es in der Lage sein, Meldungen auf der Schleife weiterzugeben. Dies ist die Leistung der Driver-Funktion.

Meldungen, die die Driver-Funktion betreffen, können von drei verschiedenen Stellen kommen: empfangene Meldungen, die sofort weiterübertragen werden müssen, Meldungen, die weiterübertragen werden müssen, nachdem das Gerät sie deko-

diert hat und Meldungen, die das Gerät selbst erzeugt hat.

Weil die Driver-Funktion eine solch elementare Funktion ist, hat sie keine Untergruppen.

Listener (L)

Diese Interface-Funktion müssen nur Geräte beinhalten, die die Funktion des Listeners auf der Schleife übernehmen. Beachten Sie an dieser Stelle den Unterschied zwischen der Acceptor Handshake-Logik, die nötig ist, damit ein Gerät eine Meldung empfangen kann (und, wenn nötig, ohne eine andere Handlung weiterübertragen), und der Listener Logik, die eine größere und anders geartete Aufgabe besitzt. Die genaue Arbeitsteilung zwischen den Funktionen wurde vorgesehen, um Sie mit einer maximalen Flexibilität und (hoffentlich) Wirtschaftlichkeit auszustatten. Sie kompliziert jedoch die Beschreibung der Eigenschaften und Fähigkeiten der Schleifengeräte.

Die Listener-Funktion ist nur dann aktiv, wenn das Gerät zuvor als Listener adressiert wurde (d.h. die LAD Meldung erhalten hat). Wenn ein Gerät auf der Schleife einmal der Listener ist, muß diese Logik Meldungen der Acceptor Handshake-Logik erhalten, die für das Gerät selbst bestimmt sind. So kann man die Acceptor Handshake-Logik als eine Schnittstelle zwischen der Schleife und der Listener-Funktion betrachten und die Listener-Funktion als Schnittstelle zu den gerätespezifischen Funktionen.

Diese Funktion gliedert sich in zwei Teile: der eine ermöglicht den Gebrauch einer ein-Byte Adresse, und der andere erlaubt die Benutzung einer erweiterten zwei-Byte Adresse. Wir werden auf die automatische erweiterte Adressierungsweise später in diesem Kapitel noch näher eingehen; auch in Kapitel 4 finden Sie eine ausführliche Erörterung.

Source Handshake (SH)

Diese Funktion muß in allen Geräten implementiert sein, die Meldungen selbst erzeugen müssen, also Geräte, die Talker oder Controller auf der Schleife sind. Weil die meisten Geräte zu einer bestimmten Zeit Talker sein müssen (und wenn nur, um Statusinformationen zu senden), ist die Source

Handshake-Funktion normalerweise notwendig.

Die Source Handshake-Funktion koordiniert die Übertragung von Meldungen vom Gerät selbst zur Driver Interface-Funktion. Diese Funktion erzeugt selbst keine Meldungen, sondern empfängt nur Meldungen vom Gerät und gibt sie dann weiter an die Driver-Funktion.

Es gibt keine Untergruppen bei der Source Handshake-Funktion. Wenn ein Gerät diese Funktion benötigt, muß es sie vollständig beinhalten.

Talker (T)

Dies ist natürlich eine der primären Funktionen. Wenn ein Gerät diese Funktion beinhaltet, hat es auf der Schleife eine Stimme und eine nicht unwesentliche Bedeutung. Die Funktion gliedert sich in zwei Hauptteile: die Basisfunktion, die sich auf eine ein-Byte Adresse bezieht, und die erweiterte Talker-Funktion, die sich auf eine zwei-Byte Adresse bezieht.

Wenn ein Gerät die Talkerfähigkeit besitzt, ist sie erst dann aktiviert, wenn es die Talk Address-Meldung (TAD) erhält. Wenn es einmal als Talker adressiert wurde, kann die Talker-Interface-Funktion Meldungen, die das Gerät erzeugt, auf die Schleife senden. Bevor die Talker-Funktion jedoch Meldungen (durch die Driver-Funktion) aussenden kann, muß sie warten, bis eine Send Data (SDA)-Meldung empfangen wird.

Die Talker Interface-Funktion ist auch für die Antwort bei einer seriellen Abfrage verantwortlich, indem sie eines oder mehrere Bytes Statusinformation des Gerätes sendet.

Auch die End of Transmission (EOT)-Meldung, die nach Daten- und Statusmeldungen gesendet werden muß, muß von der Talker-Funktion erzeugt werden.

Controller (C)

Dies ist, wie bekannt, eine Hauptfunktion, die nicht von allen Geräten auf der Schleife durchgeführt werden muß. Wie Sie sich erinnern werden, gibt es jeweils nur einen Controller auf der Schleife.

Wenn ein Gerät die Controller-Interface-Funktion beinhaltet, kann es die Command (CMD)-, Ready (RFC)- und Identify

(IDY)-Meldungen an andere Geräte auf der Schleife schicken. Es muß gleichfalls in der Lage sein, die parallele Abfrage durchzuführen, zu untersuchen, ob Service-Anforderungen vorliegen und Übertragungsfehler in geräteabhängigen Übertragungen herausfinden.

Wie in Kapitel 5 dargestellt, kann mehr als ein Gerät auf der Schleife in der Lage sein, die Schleifenkontrolle auszuüben. Wenn man die Rolle des Schleifen-Controllers auf ein anderes Gerät übertragen will, benutzt man die Take Control-Meldung (TCT). Es kann jedoch immer nur ein Gerät als System-Controller definiert werden. Der System-Controller verrichtet seine spezifischen Aufgaben gewöhnlich beim Einschalten des Systems und ist das einzige Gerät auf der Schleife, das in der Lage ist, die Interface Clear (IFC)-Meldung zu senden, damit die Geräte auf der Schleife in ihre Ausgangsbedingung gehen.

Man kann sich die meisten Eigenschaften der Controller-Interface-Funktion ableiten, wenn man sich die Schleifenabfolgen, die in den vorhergehenden Kapiteln abgeleitet wurden, näher betrachtet. Wann immer eine bestimmte Abfolge von Meldungen stattfinden soll, ist die Controller-Funktion für die Erzeugung und Durchführung dieser Abfolge verantwortlich.

Die Controller-Interface-Funktion erhält Meldungen von der Receiver- und Acceptor Handshake-Funktion und sendet Meldungen über die Driver- und Source Handshake-Funktion. Insofern kann man sich die Controller-Funktion als Arbeit in einer Art Elfenbeinturm vorstellen, völlig abgeschirmt von den niederen Aktivitäten, die auf der Schleife oder innerhalb des Gerätes vonstatten gehen. Die Controller-Funktion erhält Informationen vom Gerät und der Schleife und erzeugt Befehle, die auf dieser Information und auf systemspezifischen Kriterien beruhen, die Sie zuvor in der Controller-Funktionslogik festgelegt haben.

Es gibt mehrere Untergruppen von Fähigkeiten, die ein Gerät, das die Controller-Funktion inne hat, je nach Bedarf beinhalten kann:

- C0 (das Gerät hat keine Controllerfähigkeit)
- C1 (das Gerät hat die Basis-Controllerfähigkeit, kann Meldungen senden und Übertragungsfehler entdecken)
- C2 (das Gerät ist der System-Controller und kann IFC-

- Meldungen senden)
- C3 (das Gerät kann Service-Anforderungsmeldungen beantworten, die um die Schleife gehen)
 - C4 (das Gerät kann die Schleifenkontrolle übertragen und annehmen)
 - C5 (das Gerät kann andere Geräte für die parallele Abfrage konfigurieren und diese durchführen)
 - C6 (das Gerät kann asynchrone Schleifenoperationen durchführen)
 - C7 (das Gerät kann automatisch Adressen zuweisen)

Beachten Sie, daß diese Untergruppen der Controller-Funktion nicht hierarchisch geordnet sind. Alle Controller müssen zwar die C1 Fähigkeit haben, aber alle anderen Fähigkeiten können sie je nach Bedarf beinhalten. So können Sie ein Gerät konstruieren, das Service-Anforderungen erkennen und die Kontrolle auf ein anderes Gerät übertragen kann. Ein solches Gerät würde dann als C1, C3, C4 bezeichnet.

Es soll noch erwähnt werden, daß nicht alle der Operationen und Abfolgen, die in den vorhergehenden Kapiteln beschrieben wurden, von jedem System durchgeführt werden können. Wenn der Controller Ihrer Schleife z.B. die Fähigkeit der parallelen Abfrage (C5) nicht beinhaltet, kann diese Operation auf der Schleife nicht durchgeführt werden. Deshalb wäre es auch nicht sinnvoll, die parallele Abfrage-Interface-Funktion (die später erklärt werden soll) in eines ihrer Geräte zu integrieren, wenn Sie ein Gerät für diese bestimmte Schleife konstruieren. Auch an dieser Stelle kann man wieder feststellen, daß die Genauigkeit der HP-IL Spezifikation sowohl die Flexibilität als auch die Wirtschaftlichkeit in die Konstruktion miteinbezieht. Sie sind in der Lage, die genauen Fähigkeiten, die sie für ihr Gerät benötigen auszuwählen, und Sie wissen, daß Ihr Gerät, soweit es die Funktionen enthält, die mindestens erforderlich sind, jederzeit in der Lage sein wird, auf der Schleife zu arbeiten.

Interface-Funktionen zur Gerätekontrolle

Wie schon der Name sagt, beziehen sich diese Interface-Funktionen eher darauf, wie die auf der Schleife miteinan-

der verbundenen Geräte zusammenarbeiten, als darauf, wie die Schleife selbst arbeitet. Sicherlich sind Ihnen die Namen dieser Funktionen im Vergleich zu den anderen in diesem Kapitel am wenigsten geläufig. Dies liegt daran, daß die anderen Funktionen festlegen, wie ein HP-IL Gerät mit der Schleife zusammenarbeiten soll.

Power-Down

Diese Funktion ermöglicht es dem Gerät, sich durch einen Befehl, der über die Schleife empfangen wird, selbst abzuschalten bzw. in einen Zustand mit niedriger Leistungsaufnahme zu versetzen. Sie erlaubt dem Gerät ebenfalls, sich unter Schleifenkontrolle wieder voll zu aktivieren. Die Power-Down-Funktion kann wahlweise vom Konstrukteur vorgesehen werden und hat keine Untergruppen. Entweder hat ein Gerät diese Fähigkeit oder nicht.

Geräte, bei denen diese Fähigkeit vom Konstrukteur vorgesehen ist, müssen sich, wenn sie die LPD-Meldung (Loop Power-Down) erhalten, selbst vollständig abschalten oder in einen Zustand mit niedriger Leistungsaufnahme versetzen. Die Einzelheiten dieses Zustands bestimmt der Konstrukteur des Gerätes. Für die Schleifenfunktionen selbst ist es völlig gleichgültig, ob sich das Gerät völlig abschaltet oder sich nur in einen Zustand mit niedriger Leistungsaufnahme versetzt.

Auf jeden Fall ist es für die Schleife notwendig, daß ein Gerät, das diese Funktion beinhaltet, indem es auf die LPD-Meldung reagiert, in der Lage ist, sich dann wieder voll zu aktivieren, wenn es den entsprechenden Befehl erhält. Normalerweise wird der Controller die Geräte wieder "aufwecken", indem er wiederholt einen Befehl aussendet, bis der Befehl wieder beim Controller ankommt. Geräte, die sich in einem Ruhezustand befinden, müssen die Schleife auf einen eventuell eingehenden Impuls hin überwachen. Wenn die erste Flanke irgendeiner Meldung ermittelt wurde, muß das Gerät sich selbst reaktivieren und damit anfangen, Meldungen zum nächsten Gerät auf der Schleife weiterzugeben.

Geräte, bei denen die Power-Down-Funktion vom Konstrukteur nicht vorgesehen ist, geben die LPD-Meldung, wenn sie sie erhalten, einfach weiter. Beachten Sie, daß die LPD-

Meldung nicht an ein bestimmtes Gerät adressiert ist. Alle Geräte, die diese Fähigkeit haben, werden sich, wenn sie die LPD-Meldung erhalten, in den Ruhezustand versetzen. (Natürlich müssen sie die LPD- und die RFC-Meldung erst an das nächste Gerät auf der Schleife weitergeben, bevor sie sich in den Ruhezustand versetzen). Beachten Sie gleichfalls, daß jede Aktivität auf der Schleife nach der LPD-Meldung automatisch alle im Ruhezustand befindlichen Geräte reaktiviert.

Bei unbeaufsichtigten Anwendungen, z.B. außer Haus, ist diese Fähigkeit sehr nützlich. Wenn ein bestimmtes Gerät von einer Batterie versorgt wird oder viel Energie verbraucht oder beides, ist es sinnvoll, es abzuschalten, wann immer es auf der Schleife nicht gebraucht wird.

Device Clear

Dieser Befehl erlaubt es dem Controller, alle Geräte auf der Schleife in ihre Ausgangsposition zu versetzen, indem er ihnen eine "Löschen"-Meldung schickt. Diese Funktion kann wahlweise vorgesehen werden, obwohl Sie ihr Gerät in den meisten Fällen damit ausstatten werden.

Obwohl sich die Geräte normalerweise, wenn sie zum ersten Mal aktiviert werden, in einer Art Ausgangsbedingung befinden, ist es eine gute Sache, wenn der Controller eigenständig die Möglichkeit hat, die Geräte zu initialisieren.

Die HP-IL Spezifikation definiert die Reaktionen eines Gerätes auf den Empfang der Device Clear-Meldung nicht; der Konstrukteur hat vollständige Freiheit in der Entscheidung, was das Gerät tun soll, um sich selbst zu löschen. Da die Methoden, wie sich ein Gerät in die Ausgangsposition versetzt, stark variieren, ist dies nur logisch. Ein Kassettenlaufwerk und ein Datenmonitor haben ganz offensichtlich in Bezug auf ihren Ausgangszustand wenig gemeinsam.

Es gibt zwei Arten, wie die Device Clear-Funktion implementiert werden kann. Eigentlich gibt es sogar drei Arten, wenn man die Möglichkeit miteinschließt, bei der die Funktion überhaupt nicht beinhaltet ist. Wenn ein Gerät diese Funktionen nicht durchführt, ist es in der HP-IL Nomenklatur als DC0 definiert.

Wenn ein Gerät auf die Device Clear (DCL)-Meldung ant-

wortet, besagt die Definition, daß es die DC1 Fähigkeit hat. Die DCL-Meldung versetzt alle Geräte in den Ausgangszustand zurück, die die DC1 Fähigkeit haben.

Wenn ein Gerät auf die Selected Device Clear (SDC)-Meldung antwortet, so ist diese Fähigkeit als DC2 definiert. Die SDC-Meldung ist immer an ein bestimmtes Gerät auf der Schleife adressiert, so daß der Schleifencontroller ein einzelnes Gerät zurücksetzen oder löschen kann. Wenn ein bestimmtes Gerät Fehler erzeugt oder häufig menschlichen Eingriff benötigt (wie z.B. ein Drucker), ist diese Fähigkeit sehr nützlich.

Beachten Sie, daß ein Gerät, das die DC2 Fähigkeit hat, in der Lage ist, sowohl auf die DCL- als auch auf die SDC-Meldung zu antworten. Es ist nicht möglich, nur die SDC-Fähigkeit im Gerät vorzusehen.

Device Trigger

Diese Funktion erlaubt es dem Controller, eine elementare Funktion eines Gerätes auszulösen, indem er die GET (Group Execute Trigger)-Meldung auf die Schleife sendet. Diese Funktion ist wahlfrei und hat keine Untergruppen. Entweder hat ein Gerät sie oder nicht.

Die Device Trigger-Funktion wird typischerweise benutzt, um die Tätigkeit eines Geräts mit der Echtzeit zu verknüpfen, oder um die Tätigkeiten eines Gerätes auf der Schleife mit denen der anderen Geräte zu synchronisieren. Wenn Sie z.B. mehrere Temperatur- oder Druckmeßgeräte auf der Schleife haben, können Sie deren Messungen zur (ungefähr) gleichen Zeit auslösen, indem Sie die GET-Meldung um die Schleife schicken.

Wenn ein Gerät die GET-Funktion ausführen soll, muß es ebenfalls die Listener-Funktion beinhalten. Weiterhin muß ein Gerät, bevor es auf die GET-Meldung antworten kann, als aktiver Listener adressiert werden. Das erlaubt es dem Controller, entweder ein einzelnes Gerät zu adressieren, oder mehrere Geräte als Listener zu bestimmen, so daß sie zur gleichen Zeit ausgelöst werden können.

Beachten Sie wieder, daß die Reaktion des Gerätes auf die Trigger-Funktion allein beim Systemkonstrukteur liegt. Für die Schleifenfunktion selbst ist diese Einzelheit ohne Belang.

Remote Local (RL)

Diese Funktion erlaubt dem Controller, ein Gerät dazu anzuweisen, Befehle entweder von der Schleife (Remote) oder von seinem eigenen Schaltpult oder Einstellknöpfen (Local) anzunehmen. Diese Fähigkeit ist für Geräte nützlich, die zu verschiedenen Zeitpunkten ihrer Tätigkeit manuell kontrolliert und bedient werden müssen. Programmierbaren Geräten erlaubt diese Funktion, unter Schleifenkontrolle oder manuell eingestellt zu werden. Als einfaches Beispiel haben Sie so die Möglichkeit, den Meßbereich eines Voltmeters einzustellen, indem Sie ihm die passende Datenmeldung über die Schleife schicken, sofern es sich im Remote-Modus befindet. Oder Sie können die entsprechenden Meßbereichsschalter einstellen, wenn es sich im Local-Modus befindet.

Die Remote Local-Funktion ist wahlfrei. Ein Gerät muß sie nicht beinhalten, wenn es sie nicht braucht.

Wenn ein Gerät für diese Funktion vorgesehen ist, muß es auch in der Lage sein, die Rolle des Listener zu übernehmen, denn es muß als aktiver Listener adressiert werden, bevor es den Remote-Modus aktivieren kann oder auf die Go To Local-Meldung (GTL) antworten kann, die bei dieser Funktion benutzt wird.

Geräte, die diese Funktion beinhalten, müssen direkt, nachdem sie eingeschaltet werden, in den Local-Modus gehen. Sie verbleiben im Local-Modus, bis sie die REN-Meldung erhalten haben und als Listener adressiert sind. An dieser Stelle gehen sie in den Remote-Modus, und ihre Arbeitsweise kann vom Controller eingestellt werden.

Das Gerät verbleibt im Remote-Modus (selbst wenn es im folgenden die Rolle des Listeners ablegen muß), bis es die Go To Local-Meldung (GTL) erhält. Die GTL-Meldung ist an kein bestimmtes Gerät gerichtet - sie ist an alle Geräte gerichtet, die gerade als aktive Listener adressiert sind.

Die Remote Local-Funktion kann wahlweise vom Konstrukteur vorgesehen werden. Wenn sie in einem Gerät nicht vorgesehen ist, bezeichnet man es als RL0. Wenn es für die Basis Remote Local-Funktion, wie beschrieben, vorgesehen ist, bezeichnet man es als RL1.

Es gibt eine erweiterte Version dieser Funktion, die man als RL2 bezeichnet. Sie hat eine "Ausschluß"-Fähigkeit. Hier kann der Controller einem Gerät die Local Lockout

(LLO)-Meldung senden, die vollständig verhindert, daß Bedienungspersonal die Einstellung eines Instrumentes über die Einstelltastatur zu einem kritischen Zeitpunkt aus Versehen verändert, indem die Einstelltasten wirkungslos werden. Wenn das Gerät sich einmal in diesem "ausgeschlossenen" Zustand befindet, verbleibt es darin, bis es eine Go To Local (GTL)-Meldung empfängt.

Adressen/Status/Service-Anforderung-Funktionen

Diese Interface-Funktionen sollten ein wenig vertrauter klingen, als die eben erläuterten gerätebezogenen Funktionen. Die Nomenklatur und die Meldungen, die jetzt erläutert werden, sind schon in früheren Kapiteln aufgetaucht. Im allgemeinen handeln diese Funktionen von den Fähigkeiten eines Gerätes auf Befehle zu reagieren, die an ein bestimmtes Gerät adressiert sind, von der Art und Weise, wie ein Gerät auf die Anfrage nach Statusinformation reagiert, und über die Methode, wie ein Gerät Service vom Controller anfordern kann.

Zugegeben, dies ist eine Art Allkategorie, damit alle diese Funktionen erfaßt werden. Es verbleiben uns nur noch ein paar Funktionen, und es ist einfacher, sie an dieser Stelle gemeinsam abzuhandeln.

Parallele Abfrage

Diese Funktion erlaubt es dem Gerät, auf die parallele Abfrage eines Controllers zu antworten, indem es ein Bit Statusinformation innerhalb der IDY-Meldung setzt. Die Funktion der parallelen Abfrage ist wahlfrei. Geräte müssen nicht in der Lage sein, sie durchzuführen. Die Funktion hat keine Untergruppen. Wenn ein Gerät diese Funktion jedoch ausführen soll, muß es ebenfalls in der Lage sein, die Rolle des Listeners zu übernehmen und auf die folgenden Meldungen zu antworten: Parallel Poll Enable (PPE), Parallel Poll Disable (PPD), Parallel Poll Unconfigure (PPU) und Identify (IDY).

Wenn Sie eine gründlichere Erläuterung der parallelen Abfrage wünschen, schlagen Sie zurück zum Kapitel 5. Wir wollen die Operation der parallelen Abfrage hier nur nochmals kurz zusammenfassen.

Der Controller legt die Bit-Nummer und Polarität der Antwort fest, die ein Gerät für die Reaktion auf die parallele Abfrage benutzen muß, indem er die PPE-Meldung sendet. Der Controller kann einzelne Geräte ausschalten, indem er ihnen die PPD-Meldung sendet, damit sie auf eine Abfrage nicht reagieren. Die Fähigkeit aller Geräte zur parallelen Abfrage kann ausgeschaltet werden, indem die PPU-Meldung gesendet wird.

Die tatsächliche Abfragetätigkeit wird eingeleitet, indem die IDY-Meldung über die Schleife geschickt wird. Die Geräte, die die parallele Abfrage durchführen sollen, sind dann in der Lage zu antworten, indem sie ein Bit in der IDY-Datenbyte-Meldung setzen (oder löschen).

Service-Anforderung

Diese Funktion erlaubt dem Gerät, Service vom Controller anzufordern, indem es das Service-Anforderungsbit in einer Data or End-Meldung (DOE) oder in einer Identify-Meldung (IDY) einsetzt. Wenn der Controller die Meldung mit einem gesetzten Service-Anforderungsbit zurückerhält, weiß er, daß ein Gerät auf der Schleife Service braucht. Aber er weiß nicht, welches. Der Controller führt nun eine parallele Abfrage durch, um herauszufinden, welches Gerät (oder welche Geräte) seinen Eingriff benötigen.

Die Service-Anforderungsfunktion ist wahlfrei. Wenn Sie vom Konstrukteur vorgesehen ist, muß das Gerät ebenfalls die Talker Interface-Funktion beinhalten, damit das Gerät als Antwort auf die Abfrage des Controllers Statusbytes senden kann.

Wenn Sie eine ausführlichere Erläuterung der Service-Anforderungsfunktion wünschen, schlagen Sie zurück zum Kapitel 5, das auch Abbildungen der typischen Schleifenabfolgen zeigt.

Die Service-Anforderungs-Funktion hat eine Untergruppe. Wenn sie überhaupt nicht in einem Gerät vorgesehen ist, bezeichnet man dies als SR0. Wenn es die Basis Service-Anforderungs-Funktion, wie eben beschrieben besitzt, bezeichnet man es als SR1. Die Untergruppe der Service-Anforderungs-Funktion, SR2 genannt, erlaubt dem Gerät die asynchrone Service-Anforderung, indem es eine eigene IDY-Meldung erzeugt. Im Kapitel 4 haben wir asynchrone Schleifenoperatio-

nen kurz gestreift und rieten davon ab, diese Fähigkeit im Gerät vorzusehen, wenn es nicht unbedingt nötig ist. Diesen Ratschlag wollen wir hier wiederholen. Wenn Sie eine ausführlichere Darstellung dieser Fähigkeit wünschen, orientieren Sie sich bitte an der HP-IL Spezifikation.

Automatische Adressierung

Diese Funktion erlaubt das Adressieren eines Gerätes per Befehl von der Schleife (als Gegensatz zur manuellen oder festverdrahteten Adressierung). Die Funktion der automatischen Adressierung ist wahlfrei. Wenn sie aber in allen Geräten auf der Schleife vorgesehen ist, muß der Anwender nie mehr der Notwendigkeit des manuellen Adressierens nachkommen, so daß man diese Funktion nur empfehlen kann.

Wenn die Funktion der automatischen Adressierung in einem Gerät nicht vorgesehen ist, ignoriert es die damit verbundenen Meldungen (AAU und AAD), und gibt sie an das nächste Gerät auf der Schleife weiter. Wenn die Funktion vorgesehen ist, nimmt das Gerät die Adresse an, die in der AAD-Meldung als seine Adresse enthalten ist, erhöht die Adresse und gibt sie an das nächste Gerät auf der Schleife weiter (ausführliche Darstellung siehe Kapitel 4).

Die Funktion der automatischen Adressierung hat keine Untergruppen. Aber zwei Interface-Funktionen variieren diese Fähigkeit. Wenn ein Gerät die Fähigkeit der automatischen Adressierung erhalten soll, muß es nur eine dieser drei Funktionen der automatischen Adressierung beinhalten. Hier sind die beiden anderen Möglichkeiten:

Automatisch Erweiterte Adresse (AE)

und Automatische Mehrfach-Adresse (AM)

Die Basis Auto-Adress-Funktion, die in den vorherigen Abschnitten erläutert wurde, erlaubt die Adressierung von bis zu 31 Geräten auf der Schleife. Die Automatisch Erweiterte und Automatische Mehrfach-Adressen-Funktion erlauben die Adressierung von bis zu 961 Geräten.

Die Wirkungsweise der Automatisch Erweiterten Adressierung entspricht im Allgemeinen dem, was wir über die Funktion der automatischen Adressierung ausführten. Statt

daß die Adresse jedoch vollständig in der Auto Address-Meldung (AAD) enthalten ist, wird dem Gerät noch ein zusätzliches Byte Adreßinformation zugesendet. Die erste Meldung nennt man die Auto Extended Secondary (AES)-Meldung. Jedes Gerät, in dem diese Funktion vorgesehen ist, speichert die in der Meldung enthaltene Adresse, erhöht sie und gibt sie an das nächste Gerät auf der Schleife weiter. Diese Tätigkeit ist mit der identisch, die wir als Standard Auto-Adressierungsweise bezeichnet haben. Wenn die in der AES-Meldung enthaltene Adresse jedoch ihr Maximum von 31 erreicht, werden die folgenden Geräte nicht mehr auf die Meldung reagieren, sondern sie geben sie weiter bis zum Controller.

Jetzt sendet der Controller die Auto Extended Primary (AEP)-Meldung. Diesmal nehmen jedoch nur die Geräte diese primäre Adresse als höchstwertigen Teil ihrer Adresse an, die bereits durch die AES-Meldung ordnungsgemäße Adressen erhalten haben. Des weiteren erhöhen sie die Adresse, die in der AEP-Meldung enthalten ist, nicht, sondern geben sie einfach zum Controller weiter. Diese Gerätegruppe ist nun konfiguriert und reagiert solange nicht mehr auf eine AES- oder AEP-Meldung, bis sie die Auto Address Unconfigure-Meldung erhält.

Jetzt kann der Controller eine neue AES-Meldung an die nächste Gerätegruppe senden (die bei der vorherigen Runde keine zulässige Adresse durch die AES-Meldung erhalten hat). Hierauf folgt wieder die AEP-Meldung. Diese Abfolge wird wiederholt, bis die AES-Meldung mit einer zulässigen Adresse, die in der Meldung erhalten ist, an den Controller zurückgegeben wird.

Die Funktion der Automatischen Mehrfach-Adresse ist den eben erläuterten beiden Funktionen ähnlich, ist aber für die Anwendung bei Geräten gedacht, die im Gerät selbst mehrere funktionelle Möglichkeiten vorsehen. Die Funktion der Automatischen Mehrfach-Adresse erlaubt, daß jeder Funktion innerhalb des Geräts eine besondere Schleifenadresse zugewiesen wird. Es folgt die Abfolge, die gesendet werden muß wenn die Mehrfach-Funktion durchgeführt werden soll.

Zuerst sendet der Controller die Auto Multiple Primary (AMP)-Meldung. Jedes Gerät, in dem die Mehrfach-Funktion vorgesehen ist, nimmt diese Adresse an, erhöht sie und gibt sie weiter um die Schleife. Als nächstes sendet der Con-

troller die Zero Extended Secondary (ZES)-Meldung. Wenn das erste Gerät die ZES-Meldung erhält, beginnt es mit der Adressenzuweisung an jede seiner internen Funktionen, indem es mit der Adresse 0 anfängt. Während es diese internen Adressen zuweist, erhöht es die Adresse in der ZES-Meldung. Wenn alle internen Adressen zugewiesen worden sind, gibt das Gerät die ZES-Meldung weiter zum Controller, der jetzt feststellen kann, wie viele funktionelle Adressen dieses Gerät hat. Dann sendet der Controller die ZES-Meldung an das nächste Gerät auf der Schleife, bis alle Mehrfach-Geräte geantwortet haben.

Device Dependent (DD)

Diese Funktion ermöglicht es dem Controller, solche Befehle zu senden, deren Inhalt völlig von dem Gerät abhängt, das sie erhalten soll. Bevor ein Gerät auf die Device Dependent-Meldung antworten kann, muß es als Talker oder Listener adressiert sein. Zu diesem Zweck benutzt der Controller zwei verschiedene Meldungen für Talker und Listener: Device Dependent Listener (DDL) und Device Dependent Talker (DDT).

Die Device Dependent Interface-Funktion ist wahlfrei, da sie aber eine einfache Art darstellt, Geräteaktivitäten einzuleiten und zu kontrollieren, wird sie wahrscheinlich in den meisten Geräten vorgesehen sein.

Die niedrigstwertigen fünf Bits der DDL- und DDT-Meldung enthalten, die Device Dependent Befehlsinformation. So stehen dem Gerätekonstrukteur bis zu 31 Befehle zur Verfügung, die für ein bestimmtes Gerät festgelegt werden können. Die HP-IL Spezifikation setzt dem Gebrauch der Device Dependent-Meldungen keine Schranken. Die Wirkung eines Befehls auf das Gerät bleibt ganz dem Konstrukteur überlassen.

In einem Kassettenlaufwerk können Sie Device Dependent-Befehle benutzen, um Aufgaben wie das Rück- und Vorspulen der Kassette ausführen zu lassen. Ein Drucker kann mit Device Dependent-Befehlen Aktivitäten einleiten wie den Papiertransport zum Blattanfang oder die Einstellung eines doppelt breiten Druckmodus.

Beachten Sie, daß es nie Unklarheiten darüber geben kann, an welches Gerät die DDT-Meldung gerichtet ist, da es zu jeder Zeit immer nur einen aktiven Talker auf der Schleife

geben kann. Es kann jedoch eine Vielzahl von aktiven Listenern geben. Deshalb müssen Sie sicherstellen, daß nicht mehr als ein aktiver Listener für die Device Dependent-Funktion auf der Schleife vorgesehen ist, außer Sie wünschen, daß die Meldung von allen aktiven Listenern angenommen wird.

Kapitel 7

Was Sie außerdem wissen sollten

Die Absicht dieses Buches ist, Ihnen von Kapitel zu Kapitel ein tieferes Wissen über das HP-IL zu vermitteln. Unser Zweck war es jedoch nie, die HP-IL Spezifikation überflüssig zu machen. Stattdessen haben wir Ihnen hier das Basis-Konzept und die Fähigkeiten des HP-IL vorgestellt, so daß sie beurteilen können, welche Möglichkeiten es für Ihre speziellen Anwendungen bietet, und damit Sie die Arbeitsanforderung einschätzen können, die nötig ist, wenn Sie das HP-IL in Ihr System einfügen wollen.

Wir beanspruchen deshalb für dieses Buch auch keine Vollständigkeit. In der Tat haben wir einige Einzelheiten absichtlich ausgelassen, da sie nur für diejenigen unter Ihnen von Belang sind, die die HP-IL Spezifikation mit allen ihren Möglichkeiten nutzen möchten.

Dieses Kapitel soll Ihnen nun einen Einblick in die Einzelheiten geben, die bis jetzt noch ausgelassen wurden. Wir wollen damit zweierlei erreichen: zum einen wollen wir auch dem flüchtigen Leser einen Eindruck über das Gesamtbild des HP-IL vermitteln und ihm ein Verständnis für die Vollständigkeit der Spezifikation geben; zum anderen wollen wir denjenigen, die gelegentlich in dieser Differenziertheit mit dem HP-IL arbeiten müssen, mit dem Entschlüsseln der Zustandsdiagramme, die die Interface-Funktionen des HP-IL vollständig definieren, eine vorbereitende Einführung geben.

 Einfache Zustandsdiagramme

Zustandsdiagramme werden besonders in der Phase der Konstruktion oft benutzt, um jeden ordnungsgemäßen Zustand, in denen ein Gerät oder System arbeiten kann darzustellen, und um die Bedingungen zu definieren, die bestehen müssen, damit ein Übergang von einem Zustand in einen anderen stattfinden kann.

Abb. 7.1 zeigt ein einfaches Zustandsdiagramm.

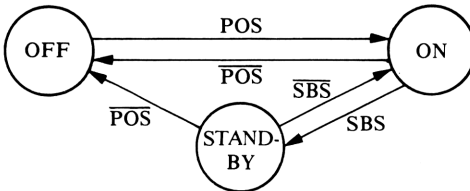


Abb. 7.1 Ein einfaches Zustandsdiagramm

Das Gerät in dieser Abbildung hat drei mögliche Zustände: OFF - ON - STANDBY (Ruhezustand). Wir haben diese Stadien als Kreise, die mit ihrem Namen (oder mnemotechnischen Abkürzungen) bezeichnet sind, dargestellt.

Pfeile, die die Kreise verbinden, zeigen die zulässigen Wege, um von einem Zustand zum anderen zu gelangen. Die Bedingungen, die diesen Übergang ermöglichen, sind durch Buchstaben entlang der Pfeile bezeichnet. In Abb. 7.1 ist POS die Abkürzung für Power On Switch (Schalter steht auf ON) und SBS heißt STANDBY Switch (Schalter steht auf Ruhezustand). Beachten Sie auch, daß POS ohne Linie über den Buchstaben in unserer Nomenklatur anzeigt, daß diese Bedingung "wahr" ist (der Schalter steht auf ON), während ein Ausdruck mit Linie darüber anzeigt, daß diese Bedingung "falsch" ist (z.B. der Schalter steht nicht auf ON, d.h. er steht auf OFF).

Damit sich dieses Gerät also vom OFF-Zustand in den ON-Zustand versetzen kann, muß der POS-Ausdruck wahr sein (der Schalter muß auf ON stehen). Der Übergang vom ON-Zustand zum STANDBY-Zustand kann nur eintreten, wenn der Schalter auf STANDBY steht (SBS).

Beachten Sie, daß dieses Zustandsdiagramm den Übergang von STANDBY zum OFF-Zustand erlaubt, wenn der ON-Schalter falsch gesetzt ist. Ein Übergang von OFF nach STANDBY ist nicht erlaubt.

Die Zustandsdiagramme für HP-IL sind, wie Sie erwarten werden, wesentlich komplizierter als in Abb. 7.1. Es werden jedoch dieselben einfachen Prinzipien verwendet, gleichgültig, wie kompliziert die Diagramme auch erscheinen mögen.

Local-Meldungen und Pseudo-Meldungen

Noch eine Hürde müssen wir überwinden, bevor wir mit den Zustandsdiagrammen des HP-IL fortfahren. Die meisten der Ausdrücke, die als Bedingungen aufgeführt sind, um Übergänge zwischen den verschiedenen Zuständen zu ermöglichen, werden Ihnen bekannt sein. Sie bestehen aus Meldungen und/oder Ausdrücken aus Interface-Funktionen, die in den vorhergehenden Kapiteln erläutert wurden. Gelegentlich werden diese Übergangsbedingungen Ausdrücke miteinschließen (immer in Kleinbuchstaben gedruckt), die als Local- oder Pseudo-Meldungen bezeichnet werden.

HP-IL definiert Remote-Meldungen (Steuermeldungen), die selbst um die Schleife gehen. Dies sind die Meldungen, die in den vorhergegangenen Kapiteln erläutert wurden. Die HP-IL Spezifikation definiert daneben auch noch Local- und Pseudo-Meldungen. Dies sind Meldungen oder Signale, die zwischen den Interface-Funktionen eines Gerätes und den völlig geräteabhängigen Funktionen ausgetauscht werden. Die einfachsten Beispiele für eine Local-Meldung sind die pof- (power off) und pon (power on)-Meldungen. Diese beiden Local-Meldungen werden in Zustandsdiagrammen der Power Down-Interface-Funktion (siehe Kapitel 6) benutzt und zeigen einfach an, daß die Interface-Funktionen und die Geräteelektronik eine Möglichkeit haben müssen, sich gegenseitig den Zustand des Netzschalters des Gerätes anzuzeigen.

Die HP-IL Spezifikation definiert eine ganze Anzahl von Lokalen und Pseudo-Meldungen. Sie brauchen jedoch nur dann alle zu kennen, wenn Sie mit ihnen arbeiten müssen. Anstatt nun alle zu definieren, wollen wir nur die näher erläutern, die hier abgebildet sind.

Driver-Zustandsdiagramm

Da alle Geräte auf der Schleife die Driver Interface-Funktion beinhalten müssen, wollen wir sie hier erläutern. Abb. 7.2 zeigt das Zustandsdiagramm dieser Interface-Funktion. Es gibt vier definierte Zustände:

- DACS (Driver Transmit from Acceptor State(34))
- DIDS (Driver Idle State(35))
- DSCS (Driver Transmit from Source State (36))
- DTRS (Driver Transfer State (37))

Wenn Sie an dieser Stelle noch einmal die Einzelheiten über die Driver-Interface-Funktion nachlesen wollen, blättern Sie bitte zurück zum Kapitel 6. Kurz zusammengefaßt ist diese Interface-Funktion für die Übertragung aller (Remote-)Meldungen auf der Schleife zuständig, gleichgültig, ob die Meldung von diesem Gerät erzeugt oder einfach auf der Schleife weitergegeben wurde.

An dieser Stelle müssen wir noch eine Frage der Nomenklatur erörtern. Die vierbuchstabigen mnemonischen Zeichen, die wir in diesem Diagramm in geschweiften Klammern erscheinen (z.B. PONS), stellen eine Verbindung von oder zu einem anderen Interface-Zustand in einer anderen Interface-Funktion dar.

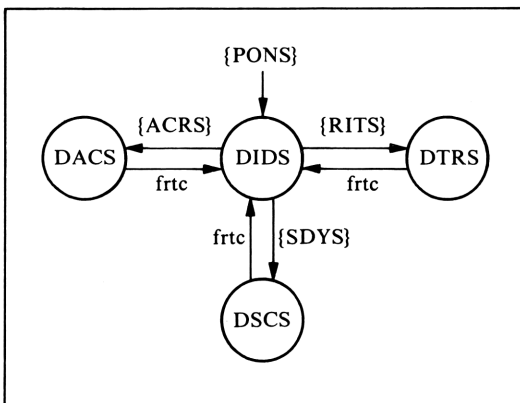


Abb. 7.2 Zustandsdiagramm der Driver-Interface-Funktion

Der Driver Idle-Status (DIDS) wird vom Power On (PONS)-Status aus erreicht, der eine getrennte Interface-Funktion ist (die PD-Funktion). In diesem Ruhezustand wartet die

Driver-Funktion auf eine neue Weisung und überträgt keine Meldung. Es ist der Status, in den die Driver-Funktion geht, wenn das System zum ersten Mal eingeschaltet wird.

Vom Ruhezustand aus führen drei verschiedene Wege. Jeder davon repräsentiert eine andere Interface-Funktion (in diesem Falle drei unterschiedliche Quellen von Meldungen, die übertragen werden sollen), die von der Interface-Funktion bedient werden muß. Die drei anderen Funktionen, die vom Driver "bedient" werden, sind die Source Handshake (SH)-, Acceptor Handshake (AH)- und Receiver (R)-Interface-Funktionen.

Der RITS (Receiver Immediate Transfer State)-Ausdruck veranlaßt den Übergang vom Ruhezustand in den DTRS-Zustand und zeigt an, daß es eine Meldung gibt, die sofort, wenn das Gerät sie dekodiert hat, weiterübertragen werden muß und nicht verzögert werden darf.

Der ACRS (Acceptor Ready State)-Ausdruck veranlaßt den Übergang vom Ruhezustand zum DACS-Zustand und zeigt an, daß eine Meldung erhalten und angenommen wurde, die jetzt weiterübertragen werden kann.

Der SDYS (Source Delay State)-Ausdruck veranlaßt den Übergang vom Ruhezustand in den DSCS-Zustand und zeigt an, daß der Driver eine Meldung übertragen soll, die vom Gerät selbst erzeugt wurde.

Beachten Sie, daß alle drei der erwähnten Ausdrücke (RITS, ACRS und SDYS) selbst Zustände innerhalb anderer Interface-Funktionen darstellen, und in den Zustandsdiagrammen dieser Funktionen definiert sind. Jetzt haben Sie einen ersten Eindruck über die Komplexität der HP-IL Spezifikation (und von dem Konstruktionsaufwand auf dieser Ebene). Es gibt eine Unzahl von Interaktionen zwischen all diesen Interface-Funktionen. Der Ansatz, ganze Systeme und Gerätefähigkeiten nach streng definierten funktionellen Gesichtspunkten zu unterscheiden, kann zu großer Flexibilität und Wirtschaftlichkeit führen. Er kann jedoch auch recht kompliziert sein.

Der gemeinsame Terminus für alle drei Übertragungswege zum Ruhezustand ist "frtc" (Frame Transmission Complete). Der frtc-Ausdruck ist eine Pseudo-Meldung, die vom Kodierungsschaltkreis des Gerätes erzeugt wird, um der Driver-Funktion zu signalisieren, daß ein vollständiger Rahmen ge-

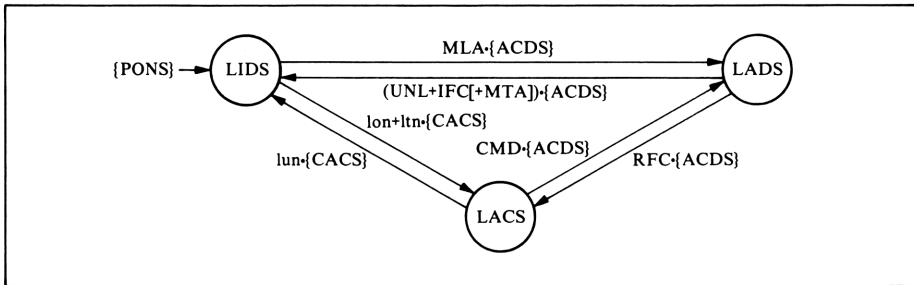


Abb. 7.3 Zustandsdiagramm der Listener-Interface-Funktion

sendet wurde, und deshalb zum Ruhezustand zurückgekehrt werden kann. So kehrt die Funktion in den Wartezustand zurück, der solange andauert, bis eine weitere Meldung übertragen werden soll.

 Listener-Zustandsdiagramm

Abb. 7.3 zeigt das Zustandsdiagramm für die Listener-Interface-Funktion. Dieses Diagramm ist komplizierter als das des Drivers und enthält viel mehr Ausdrücke, die Übergänge zwischen den verschiedenen Zuständen verursachen. Wir wollen mit dem Listener Idle-Status(38) beginnen.

Wie Sie vielleicht schon erwarten, geht diese Funktion beim Einschalten in den Ruhezustand (Beachten Sie die Verbindung vom PONS-Zustand). Der Übergang zu LADS (Listener Addressed State) ist sehr geradlinig. Es unterstellt, daß dieses Gerät seine Listen Address (MLA = My Listen Address) von der Schleife erhalten hat und daß sich die Acceptor Handshake-Funktion (AH) im Acceptor Daten-Status (ACDS) befindet, der anzeigt, daß die MLA-Meldung vollständig dekodiert wurde und für dieses Gerät gültig ist. Beachten Sie, daß der Punkt, der den MLA- und ACDS-Ausdruck miteinander verknüpft "und" bedeutet; d.h. damit der gesamte Ausdruck wahr ist, müssen beide Ausdrücke wahr sein.

Der Übergang vom LADS-Zustand zum Listener Active State (LACS) ist ebenfalls geradlinig. Wenn sich das Gerät bereits im Listener Addressed-Status (LADS) befindet und es eine Ready For Command (RFC)-Meldung empfängt, dann kann die Funktion weiter die Rolle des aktiven Listeners auf der

Schleife übernehmen. Wieder werden Sie feststellen, daß der andere notwendige Ausdruck für den Übergang von LADS zu LACS der Acceptor-Daten-Status (ACDS) ist, der anzeigt, daß die RFC-Meldung von der Acceptor Handshake-Funktion vollständig dekodiert wurde.

Wenn sich die Funktion im Listener Active-Status (LACS) befindet, verbleibt sie dort und empfängt Meldungen von der Schleife, bis sie eine Befehlsmeldung (CMD) erhält.

Wenn eine CMD-Meldung empfangen wird, ist das Gerät noch immer als Listener adressiert und kehrt in den LADS- (Listener Addressed Status) zurück. Von dort kann sie, wenn eine RFC-Meldung erhalten wird, wieder zu LACS zurückgehen, oder sie kann sich in den Ruhezustand (LIDS) versetzen. Der Ausdruck, der die Rückkehr zum Ruhezustand bewirken kann, lautet (UNL+IFC [+MTA])-{ACDS}.

Obwohl dieser Ausdruck abschreckend erscheinen mag, ist er nicht allzu kompliziert. Er besagt, daß die Funktion zum Ruhezustand (LIDS) zurückkehrt, wenn die Unlisten-Meldung (UNL) "oder" (das Pluszeichen bedeutet "oder") die Interface Clear-Meldung (IFC) empfangen wird. Natürlich muß dazu auch der schon bekannte ACDS-Ausdruck wahr sein.

Wir haben einen Begriff dieses Ausdrucks noch nicht besprochen [+MTA]. Sie sehen, daß dieser Begriff in eckigen Klammern steht, statt in runden oder geschweiften Klammern. In der Konvention der HP-IL Zustandsdiagramme zeigen die eckigen Klammern wahlfreie Ausdrücke an, die in dieser Funktion vorgesehen werden können oder auch nicht. In diesem Fall heißt dies, daß ein Gerät, das als Listener adressiert ist, nun als Talker adressiert wird. MTA ist die My Talk Address-Meldung, die anzeigt, daß dieses Gerät als Talker adressiert ist.

Die beiden anderen Übergänge im Listener-Zustandsdiagramm sind hauptsächlich für diejenigen Geräte wichtig, die auch aktive Controller der Schleife sind.

Acceptor Handshake-Zustandsdiagramm

Die Acceptor Handshake (AH)-Interface-Funktion muß, wie die vorne beschriebene Driver-Interface-Funktion, in allen Geräten vorgesehen sein. Die Funktion empfängt Meldungen, die für ein Gerät bestimmt sind, von der Receiver-Interface-Funktion. Ist die Meldung für eine Weiterver-

arbeitung bestimmt, zeigt die AH-Funktion diese Tatsache den anderen Funktionen an. (Sie werden sich an den Acceptor Daten-Status (ACDS) erinnern, der bei der Erläuterung des Listener-Zustandsdiagrammes auftauchte).

Die AH-Interface-Funktion legt ebenfalls fest, ob eine Meldung nach der Verarbeitung weiterübertragen werden soll (dies nennt man "repeat"(39)) oder ob sie ausgesondert werden kann (dies nennt man norepeat(40)).

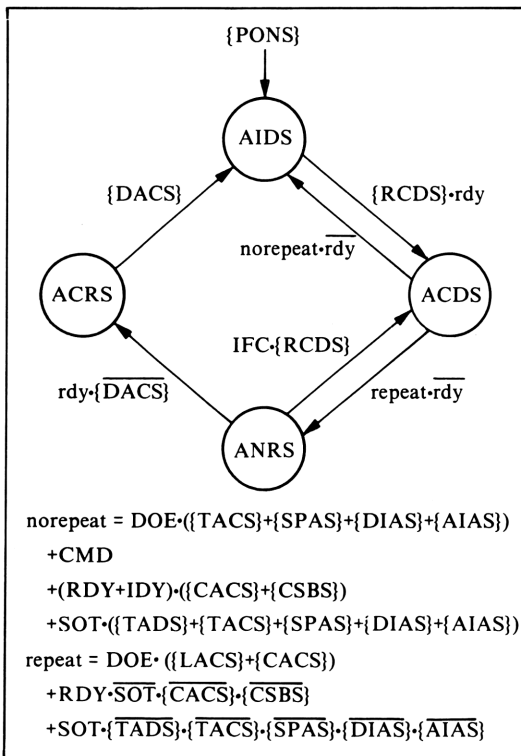


Abb. 7.4 Zustandsdiagramm der Acceptor Handshake-Interface-Funktion

Abb. 7.4 zeigt das Zustandsdiagramm der Acceptor Handshake-Interface-Funktion. Diese Abbildung sieht nicht komplizierter als die des Listeners aus, denn sie wurde dadurch vereinfacht, daß die Begriffe "repeat" und "norepeat" eingefügt wurden. Die Ausdrücke, für die die beidem verein-

fachenden Begriffe eingefügt wurden, finden Sie unter dem Zustandsdiagramm; sie sind von großer Bedeutung. Wir wollen diese Ausdrücke jedoch zunächst beiseite lassen und uns der vereinfachten Form des Zustandsdiagrammes zuwenden.

Wieder ist der Ausgangspunkt der Ruhezustand (AIDS), bei dem das Gerät zum ersten Mal eingeschaltet wird. Von dort aus geht der einzige zulässige Übergang zum Acceptor-Daten-Status (ACDS). Dieser Übergang findet statt, wenn die Local Ready-Meldung (rdy) wahr ist und die Receiver-Interface-Funktion anzeigt, daß eine Meldung für dieses Gerät beim Eintritt in den Receiver-Data-Status (RCDS) verfügbar ist. Die Local rdy-Meldung kommt vom Gerät und zeigt der Acceptor Handshake-Funktion an, daß das Gerät bereit ist, ein weiteres Datenbyte zu empfangen.

Nachdem das Gerät die Meldung zur Verarbeitung erhalten hat, setzt es die Local rdy-Meldung falsch. Wenn die Meldung nicht weiterübertragen werden soll (norepeat), so kehrt die AH-Funktion jetzt in den Ruhezustand (AIDS) zurück, oder sie geht in den Acceptor Not Ready-Zustand (ANRS), wenn die Meldung weiterübertragen werden muß (repeat).

Wenn sich die Funktion im Acceptor Not Ready-Status (ANRS) befindet, wartet sie darauf, daß das Gerät seine Bereitschaft anzeigt, das nächste Datenbyte von der Schleife zu erhalten. Dies zeigt das Gerät an, indem es die Local rdy-Meldung auf wahr setzt. Dies geschieht normalerweise, nachdem ein Gerät alle durch die gerade ablaufende Meldung notwendig gewordenen Aktionen abgeschlossen hat. Wenn das Gerät damit fertig ist, kann die AH-Funktion den Übergang zum Acceptor Ready-Status (ACRS) vornehmen, in dem die Weiterübertragung der Meldung dann beginnen kann.

Sie werden feststellen, daß es zusätzlich zur rdy-Meldung noch einen anderen Ausdruck gibt, der beim Übergang vom ANRS zum ACRS beteiligt ist. Die rdy-Meldung muß von einer nicht gesetzten DACS begleitet werden. Wenn Sie sich an unsere Erläuterung der Driver-Interface-Funktion erinnern, werden Sie feststellen, daß DACS (Driver Transmit from Acceptor State) wahr ist, wenn der Driver gerade eine Meldung sendet. Deshalb kann die AH-Funktion erst dann zum ACRS-Zustand übergehen, wenn die Übertragung der vorhergehenden Meldung durch den Driver abgeschlossen ist. Dort muß der Driver damit beginnen, eine neue Meldung zu senden.

Wenn sich die Funktion im Acceptor Ready-State (ACRS) befindet, zeigt sie der Driver-Funktion an, daß die Weiterübertragung der gegenwärtigen Meldung beginnen soll. Wenn die Driver-Funktion anzeigt, daß sie mit der Weiterübertragung dieser Meldung begonnen hat (indem sie DACS auf wahr setzt) kann die AH-Funktion den Übergang zurück zum Ruhezustand (AIDS) vollziehen.

Kehren wir für einen Augenblick zum Acceptor Not Ready-Status (ANRS) zurück. Wir haben den Übergang von diesem Zustand zum ACRS-Zustand verfolgt, aber wenn Sie sich Abb. 7.4 ansehen, kann die Funktion ebenso vom ANRS zum Acceptor Data-State (ACRS) zurückgehen. Dieser Übergang wird gewählt, wenn der Receiver (RCDS) eine Interface Clear Meldung (IFC) erhält, während er sich im ANRS-Zustand befindet.

Nun wenden wir unsere Aufmerksamkeit den langen Ausdrücken unterhalb des Zustandsdiagrammes in Abb. 7.4 zu, die wir durch repeat und norepeat abgekürzt haben. Da die meisten der Begriffe dieser Ausdrücke sich auf Zustände innerhalb anderer Interface-Funktionen beziehen, wollen wir diese nicht alle erläutern. Denn dazu müßte man sämtliche Zustandsdiagramme der betroffenen Interface-Funktionen erläutern. Stattdessen wollen wir nun die Wirkung von jedem einzelnen Ausdruck zusammenfassen, um Ihnen einen Eindruck davon zu geben, welche Faktoren dafür verantwortlich sind, daß die Acceptor Handshake-Funktion festlegt, ob eine Meldung weiterübertragen werden muß.

Grundsätzlich gibt es drei verschiedene Typen von Meldungen, die nicht weiterübertragen werden müssen.

- DOE (Data or End)-Meldungen, die von diesem Gerät gesendet wurden (das Gerät ist der Talker)
- CMD (Command)-, RDY (Ready)- oder IDY (Identify)-Meldungen, die von diesem Gerät gesendet wurden (das Gerät ist der Controller)
- SOT (Start Of Transmission)-Meldungen, die für dieses Gerät bestimmt sind (wieder ist das Gerät der Talker)

Alle die Ausdrücke, die diesen mnemonischen Abkürzungen folgen, repräsentieren verschiedene Zustände innerhalb anderer Interface-Funktionen, die mit der Acceptor Handshake-Funktion zusammenarbeiten.

Umgekehrt gibt es drei verschiedene Meldungskategorien, die von der AH-Funktion weiterübertragen werden müssen.

- DOE-Meldungen, die nicht von diesem Gerät gesendet wurden
- RDY-Meldungen, die nicht von diesem Gerät gesendet wurden
- SOT-Meldungen, die nicht an dieses Gerät gerichtet sind

Selbst wenn wir alle Begriffe, die in den vereinfachten repeat/norepeat-Ausdrücken vorkommen, nicht erläutern, werden Sie verstehen, daß die Zusammenhänge zwischen verschiedenen Funktionen, wie Abb. 7.5 zeigt, sehr kompliziert sein können.

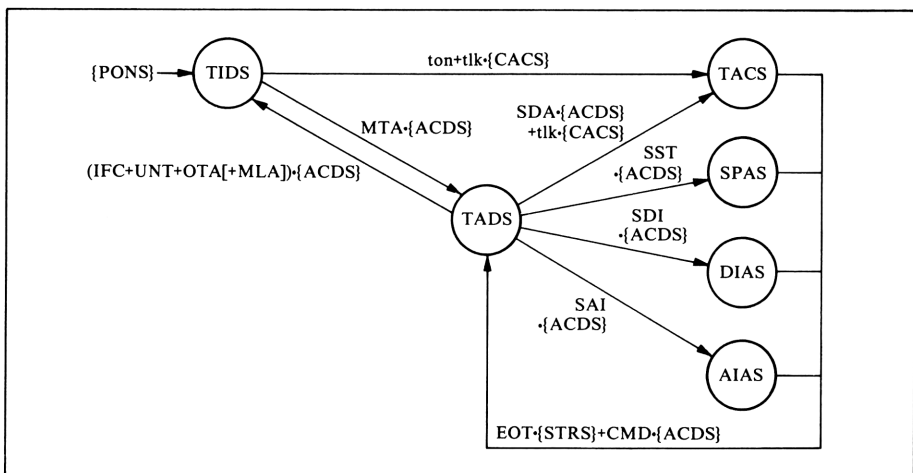


Abb. 7.5 Zustandsdiagramm der Talker-Interface-Funktion

 Talker-Zustandsdiagramm

Abb. 7.5 zeigt das Diagramm für die Talker-Interface-Funktion. Wie Sie sehen, ist es noch komplizierter als die bisher erläuterten Diagramme.

Denken Sie daran, daß das Diagramm deshalb komplizierter ist, weil es mehr Zustände besitzt, in denen es arbeiten

kann. Dies bedeutet jedoch nicht, daß es schwieriger zu verstehen ist oder, was noch wichtiger ist, daß es schwieriger zu implementieren ist als einige der einfacher aussehenden Diagramme. Als Faustregel kann man sagen, daß die Anzahl der Begriffe im Übergangsausdruck – besonders wenn viele Begriffe von anderen Interface-Funktionen stammen – festlegt, wie kompliziert es ist, alle Fähigkeiten einer gegebenen Funktion durchzuführen.

Da Sie nun einen Einblick in diese Arbeitsweise haben, wollen wir das Talker-Zustandsdiagramm nicht ausführlich erläutern. Als Orientierungshilfe werden im folgenden die sechs Zustände aus Abb. 7.5 aufgeführt:

- TIDS (Talker Idle State)(41)
- TADS (Talker Addressed State)(42)
- TACS (Talker Active State)(43)
- SPAS (Serial Poll Active State)(44)
- DIAS (Device Identify Active State)(45)
- AIAS (Accessory Identify Active State)(46)

Da in diesem Zustandsdiagramm viele der Begriffe der Übergangsausdrücke aus Remote-Meldungen bestehen, sollten Sie in der Lage sein, mit Hilfe des Meldungsglossars am Ende dieses Buches wenigstens ansatzweise herauszufinden, was in diesem Diagramm vor sich geht.

Anhang

Der HP-IL Befehlsvorrat

Jetzt, da Sie einen Bezug zu den HP-IL Konzepten gewonnen haben, wollen wir uns einen Überblick über die gesamten Meldungen verschaffen, die über das Schleifen-Interface gesendet werden können. Die Gesamtheit der in dieser Meldungsliste enthaltenen Meldungen nennt man den HP-IL Befehlsvorrat.

Auf den ersten Blick mag die Meldungsliste sehr kompliziert erscheinen, wie es oft der Fall ist, wenn man mit einem neuen Befehlsvorrat arbeiten muß. Wenn Sie sich jedoch genauer ansehen, wie die Meldungen zusammengestellt sind, werden Sie schnell feststellen, daß sie sich in vier elementare Klassen einordnen lassen: Data or End, Command, Ready und Identify (siehe Abb. A.1).

Die folgende Erläuterung soll Ihnen helfen, den HP-IL Befehlsvorrat zu verstehen. Wir wollen mit einem Rückblick auf die allgemeine Struktur einer HP-IL Meldung beginnen und dann in einfachen Worten erklären, auf welche Art die HP-IL Meldungen organisiert sind.

Jede HP-IL Meldung in Abb. A.2 ist nach ihrem dreibuchstabigen mnemonischen Kürzel alphabetisch aufgeführt. Jeder Befehl und jede Meldung wird im Glossar am Ende dieses Buches noch einmal ausführlich definiert.

Die Meldungsstruktur des HP-IL

Man kann sich den Meldungsrahmen des HP-IL als ein Paket von Informationen vorstellen, das aus elf Bits zusammen-

C2	C1	C0	Klasse	Meldung
0	END	SRQ	Data or End	DOE
1	1	SRQ	Identify	IDY
1	0	0	Command	CMD
1	0	1	Ready	RDY

Abb. A.1 Die Kodierung der Bits C2, C1, C0

Abb. A.2 Tabelle der HP-IL Meldungen

Name	Code	Klasse	Funktion	Untergruppe
AAD	101 100 AAAAA	Ready	Auto Address 0-30	AAG
AAG	101 100 XXXXX	Ready	Auto Address Group	—
AAU	100 1001 1010	Command	Auto Address Unconfigure	UCG
ACG*	100 X000 XXXX	Command	Addressed Command Group	—
AEP	101 101 AAAAA	Ready	Auto Extended Primary	AAG
AES	101 110 AAAAA	Ready	Auto Extended Secondary	AAG
AMP	101 111 AAAAA	Ready	Auto Multiple Primary	AAG
ARG	101 01XX XXXX	Ready	Addressed Ready Group	—
CMD	100 XXXX XXXX	Command	Command Class Message	—
DAB	00X XXXX XXXX	Data or End	Data Byte	—
DCL	100 0001 0100	Command	Device Clear	UCG
DDL	100 101X XXXX	Command	Device Dependent Listener	ACG
DDT	100 110X XXXX	Command	Device Dependent Talker	ACG
DOE	0XX XXXX XXXX	Data or End	Data or End Class	—
EAR	100 0001 1000	Command	Enable Asynchronous Requests	UCG
END	01X XXXX XXXX	Data or End	End Byte	—
EOT	101 0100 000X	Ready	End of Transmission	ARG
ETE	101 0100 0001	Ready	End of Transmission — Error	ARG
ETO	101 0100 0000	Ready	End of Transmission — OK	ARG
GET	100 0000 1000	Command	Group Execute Trigger	ACG
GTL	100 0000 0001	Command	Go to Local	ACG
IAA	101 100 11111	Ready	Illegal Auto Address	AAG
IDY	11X XXXX XXXX	Identify	Identify	—
IEP	101 101 11111	Ready	Illegal Extended Primary	AAG
IES	101 110 11111	Ready	Illegal Extended Secondary	AAG
IFC	100 1001 0000	Command	Interface Clear	UCG
IMP	101 111 11111	Ready	Illegal Multiple Primary	AAG
LAD	100 001 AAAAA	Command	Listen Address (0-30)	LAG
LAG	100 001X XXXX	Command	Listen Address Group	—
LLO	100 0001 0001	Command	Local Lockout	UCG
LPD	100 1001 1011	Command	Loop Power Down	UCG
MLA	100 001 AAAAA	Command	My Listen Address	LAG

AAAAA = 5-Bit Adresse
 XXXXX = Bits ohne spezifische Bedeutung
 S = Polarität (Sense): 0 = gesetzt, wenn kein SRQ
 1 = gesetzt, wenn SRQ
 BBB = Bit: 000 = D0 bis 111 = D7

* Enthält ebenfalls die DDL- und DDT-Befehle

Abb. A.2 Tabelle der HP-IL Meldungen (Fortsetzung)

Name	Code	Klasse	Funktion	Untergruppe
MSA	100 011 AAAAA	Command	My Secondary Address	SAG
MTA	100 010 AAAAA	Command	My Talk Address	TAG
NAA	101 100 AAAAA	Ready	Next Auto Address	AAG
NES	101 110 AAAAA	Ready	Next Extended Secondary	AAG
NMP	101 111 AAAAA	Ready	Next Multiple Primary	AAG
NRD	101 0100 0010	Ready	Not Ready for Data	ARG
NRE	100 1001 0011	Command	Not Remote Enable	UCG
NUL	100 0000 0000	Command	Null Command	ACG
OSA	100 011 AAAAA	Command	Other Secondary Address	SAG
OTA	100 010 AAAAA	Command	Other Talk Address	TAG
PPD	100 0000 0101	Command	Parallel Poll Disable	ACG
PPE	100 1000 SBBB	Command	Parallel Poll Enable	ACG
PPU	100 0001 0101	Command	Parallel Poll Unconfigure	UCG
RDY	101 XXXX XXXX	Ready	Ready Class	–
REN	100 1001 0010	Command	Remote Enable	UCG
RFC	101 0000 0000	Ready	Ready for Command	–
SAD	100 011 AAAAA	Command	Secondary Address	SAG
SAG	100 011X XXXX	Command	Secondary Address Group	–
SAI	101 0110 0011	Ready	Send Accessory ID	ARG
SDA	101 0110 0000	Ready	Send Data	ARG
SDC	100 0000 0100	Command	Selected Device Clear	ACG
SDI	101 0110 0010	Ready	Send Device ID	ARG
SOT	101 0110 0XXX	Ready	Start of Transmission	ARG
SRQ**	0X1 XXXX XXXX	DOE or IDY	Service Request	–
SST	101 0110 0001	Ready	Send Status	ARG
TAD	100 010 AAAAA	Command	Talk Address	TAG
TAG	100 010X XXXX	Command	Talk Address Group	–
TCT	101 0110 0100	Ready	Take Control	ARG
UCG	100 X001 XXXX	Command	Universal Command Group	–
UNL	100 0011 1111	Command	Unlisten	LAG
UNT	100 0101 1111	Command	Untalk	TAG
ZES	101 110 00000	Ready	Zero Extended Secondary	AAG

AAAAA = 5-Bit Adresse
XXXXX = Bits ohne spezifische Bedeutung
S = Polarität (Sense): 0 = gesetzt, wenn kein SRQ
 1 = gesetzt, wenn SRQ
BBB = Bit: 000 = D0 bis 111 = D7
** Auch 111 XXXX XXXX

gesetzt ist. Die Information, die in diesem Paket enthalten ist, strukturiert sich wie in Abb. A.3 dargestellt.

Diese 11-Bit-Meldung ist das, was vom Controller zum Gerät oder vom Talker zum Listener über die Schleife geht. Abbildung A.4 zeigt einen typischen Meldungsrahmen, der über die Schleife geht. Bei jeder Meldung werden die höchstwertigen Bits des Rahmens zuerst gesendet. Da die Meldungen hintereinander über das Schleifen-Interface gesendet werden, kommen die ersten Bits beim jeweiligen Gerät in der Reihenfolge C2, C1, C0 an. So hat jedes Gerät auf der Schleife die Möglichkeit, diese Bits zu dekodieren und sich auf die jeweilige Reaktion auf die Meldung vorzubereiten.

Wenn eine Meldung einen Befehl enthält, der sich auf ein bestimmtes Gerät bezieht, wird die Reaktion normalerweise nur solange verzögert, bis das Gerät für seine eigenen Zwecke eine Kopie fertiggestellt hat. Dies erhöht die Durchsatzgeschwindigkeit auf der Schleife und erlaubt es den Geräten, den Befehl auf lokaler Ebene fast gleichzeitig auszuführen.

Die Bits C2, C1 und C0 klassifizieren den Typ der Meldung, die in den Bits D7 bis D0 enthalten ist.

Bit C2

Bit C2 dient zwei Zwecken: Da es das zuerst gesendete Bit einer Meldung darstellt, muß es als Sync- (Synchronisations-) Bit oder Anfangsbit dienen. Es zeigt dem Gerät den Beginn einer Meldung an. Diese Synchronisation ist deshalb

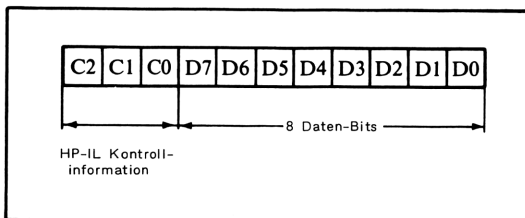


Abb. A.3 Ein HP-IL Meldungsrahmen

notwendig, weil Meldungen zu einem nicht vorhersehbaren Zeitpunkt bei einem Gerät ankommen können (eine Meldung ist ein asynchrones Ereignis). Bit C2 zeigt dem Gerät gleichfalls an, ob diese Meldung ein Datenbyte enthält.

Wenn Bit C2 null ist, wird dem Gerät angezeigt, daß diese Meldung acht Datenbits enthält (D7-D0). Wenn Bit C2 eins ist, wird dem Gerät auf der Schleife damit mitgeteilt, daß diese Meldung aus einer der drei verbleibenden Meldungsklassen besteht: Command, Ready oder Identify.

Bit C1

Wenn ein Meldungsrahmen eine Meldung der Datenklasse enthält, zeigt Bit C1 das Ende eines logischen Datensatzes an. Wenn Datenbytes auf der Interface-Schleife vom Talker zum Listener übertragen werden, ist es oft notwendig, dem Listener mitzuteilen, daß dieses Byte das letzte in einem logischen Zusammenhang ist. Zum Beispiel werden ASCII-Dateien oft mit einem Wagenrücklauf und Zeilenvorschub gesendet, um das Ende einer Textzeile anzuzeigen. In diesem Fall ist der logische Zusammenhang eine Textzeile. Wenn eine Textdatei über die Schleife übertragen wird, z.B. zu einem Drucker, kann das Bit C1 dazu benutzt werden, das Ende einer Zeile anzuzeigen. In Rahmen, die keine Daten enthalten, zeigt das Bit einem Gerät an, ob eine Meldung der Command- oder Ready-Klasse (C1=0) oder der Identify-Klasse (C1=1) angehört.

Bit C0

Bit C0 wird von einem Gerät benutzt, um dem Controller anzuzeigen, daß ein Gerät so bald wie möglich Service braucht. Diese Methode, dem Controller ein Signal zu geben, bezeichnet man als Service-Anforderung.

Weil Meldungen, die an verschiedene Geräte auf der Schleife gerichtet sind, durch das Interface des Gerätes gehen, muß es die Möglichkeit haben, den aktiven Controller schnell zu alarmieren. Es kann bei einem Eingabegerät vorkommen, daß es Service benötigt, um einen Überlauf zu verhindern (weitere Eingabedaten gehen dann verloren), oder ein Drucker muß dem Talker (über den Controller) mitteilen, daß sein Puffer voll ist und daß er keine neuen Informa-

tionen mehr annehmen kann. Beachten Sie, daß C0 nur für Daten und Identify-Rahmen das Service-Anforderungsbit ist. Wenn C2 und C1 jeweils 1 bzw. 0 sind, zeigt C0 an, ob die Meldung eine Command (C0=0)- oder eine Ready (C0=1)-Meldung ist.

Im allgemeinen werden die Bits C2, C1 und C0 benutzt, um den Meldungstypus, der über die Schleife geht, zu kategorisieren. Die Bits C2, C1, und C0 legen die HP-IL Meldungshierarchie fest. Dies ist eine einfache Methode, Meldungen zu kategorisieren. Wir wollen sie im folgenden erläutern.

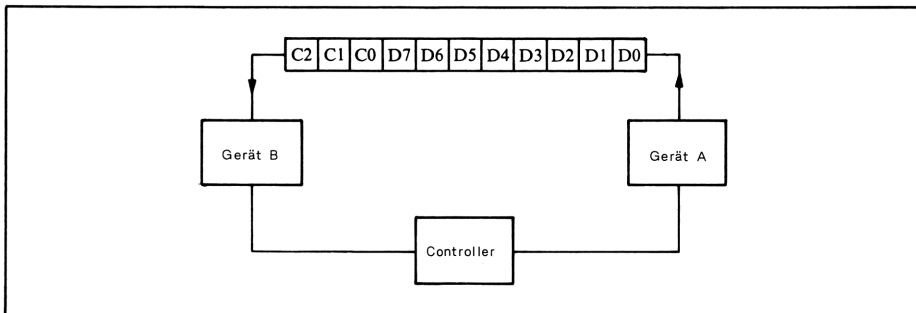


Abb. A.4 Eine Meldung auf der Schleife

Die HP-IL Meldungshierarchie

Wir haben die Struktur einer HP-IL Meldung und die Meldungstypen bereits kurz erläutert. Die Meldungen werden innerhalb einer Hierarchie eingeordnet.

Diese Hierarchie ist in zweierlei Hinsicht nützlich. Zunächst hilft sie, die Meldungen schnell mit ihrer Basisfunktion zu assoziieren. Zum anderen legt sie eine angemessene logische Methode fest, bei der HP-IL Funktionen sowohl in Software als auch in Hardware implementiert werden können.

HP-IL Meldungen sind (aus organisatorischen Gründen) in vier Haupttypen eingeteilt: Die Command-Klasse (CMD), die Data or End-Klasse (DOE), die Ready-Klasse (RDY) und die Identify-Klasse (IDY).

Diese Klassen oder Kategorien basieren auf der Kodierung der Bits C2, C1 und C0 des Meldungsrahmens (vgl. Abb. A.1). Abb. A.5 zeigt die Gesamtorganisation der HP-IL Meldungen, die wir als HP-IL Meldungshierarchie bezeichnen.

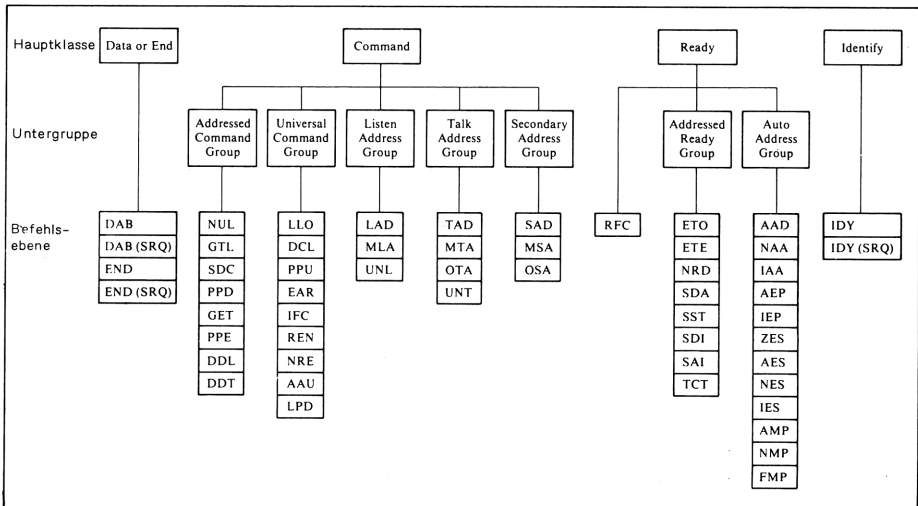


Abb. A.5 Die Hierarchie der HP-IL Meldungen

Command-Klasse

Die Command-Klasse (CMD) umfaßt alle Meldungen, die vom aktiven Controller an die anderen Geräte auf der Schleife gesendet werden. Diese Meldungen enthalten Befehle, die entweder die Interface- oder die Geräte-Funktionen kontrollieren. Die Befehle der Command-Klasse können sich auf ein einzelnes Gerät oder auf alle Geräte der Schleife beziehen.

Ein Gerät, das eine Meldung der Command-Klasse empfängt, überträgt sie sofort zum nächsten Gerät auf der Schleife weiter und behält seine eigene Kopie des Befehls, bevor es mit dessen Ausführung beginnt. Der aktive Controller der Schleife muß der Command-Meldung eine Ready For Command- (RFC) Meldung folgen lassen (sobald der Befehl zurückgekehrt ist), um sicherzustellen, daß er ausgeführt wurde.

Instruktionen der Command-Klasse umfassen:

- die Befehle, die an ein bestimmtes Gerät gerichtet sind (adressierte Command-Gruppe)
- die Befehle, die an alle Geräte auf der Schleife gerichtet sind (universelle Command-Gruppe)
- die Befehle, die den Geräten auf der Schleife die Listener und Talkerrolle zuweisen (LAG, TAG und SAG).

Data- oder End-Klasse (DOE)

Die Meldungen dieser Klasse umfassen sämtliche Daten, die vom aktiven Talker zum aktiven Listener geschickt werden. Das letzte Datenbyte eines Datensatzes wird dabei Endbyte genannt. Der Talker kann das Ende des Satzes anzeigen, indem er das C1 Bit in der Datenmeldung setzt. Beachten Sie, daß das Endbyte etwas anderes ist als die End Of Transmission (ETO, ETE)-Meldung. Das Endbyte veranlaßt nicht die Beendigung einer Übertragung zwischen Talker und Listener. Normalerweise wird C1 benutzt, um einen logischen Endzustand anzuzeigen. Dies kann das Ende eines Datensatzes sein, oder im Falle einer ASCII-Datei, das Ende einer Zeile.

Ready-Klasse

Die Meldungen der Ready-Klasse sind ein Instrument, um die Schleifenoperationen zu kontrollieren. Innerhalb der Ready-Klasse gibt es drei elementare Gruppen: die RFC (Ready For Command)-Meldung, die Meldungsgruppe, die wir unter dem Namen Addressed Ready Group kennen und die Auto Addressed-Gruppe.

Die Ready For Command (RFC)-Meldung erlaubt dem aktiven Controller festzustellen, wann die Geräte die Ausführung eines gegebenen Befehls abgeschlossen haben.

Die Addressed Ready Group-Befehle kontrollieren normalerweise die Daten, die vom Talker zum Listener gesendet werden. Diese Befehle legen genau fest, wann ein Gerät damit beginnen soll, Daten zu senden (SOT-Gruppe), und zeigen dem Controller die Beendigung einer Übertragung (EOT-Gruppe) oder die Tatsache an, daß der Controller eine

Datenübertragung unterbrechen muß (NRD-Meldung).

Die Ready Class-Meldungen schließen auch eine Befehlsgruppe ein, die Auto Addressed Group (AAG) genannt wird. Auto Address-Meldungen erlauben dem Controller, ohne menschlichen Eingriff die Schleife zu konfigurieren. So werden durch die Meldungen der AAG-Gruppe den Geräten Adressen zugewiesen.

Identify (IDY)-Klasse

Die Meldungsklasse unter dem Namen Identify (IDY) wird vom Controller benutzt, um festzustellen, ob ein Gerät auf der Schleife Service benötigt. Der Prozeß, bei dem man feststellt, welches Gerät Aufmerksamkeit verlangt, nennt man "Polling" (Abfrage). Die HP-IL Befehlsstruktur sieht zwei Abfragemethoden vor: die serielle und die parallele Abfrage.

Glossar

Fachbegriffe und Steuermeldungen

Nachfolgend finden Sie eine Liste der Fachbegriffe und Steuermeldungen, die im HP-IL benutzt werden. Die Meldungen sind alphabetisch nach ihren dreibuchstabigen mnemonischen Kürzeln angeordnet. Dann folgt die ausgeschriebene Meldung in englischer Originalfassung, sowie die deutsche Übersetzung in Klammern. Am Ende der Erläuterung jeder Meldung steht ihre binäre Darstellung, in der "X" ein beliebiges Bit und "A" ein Adressenbit repräsentiert.

AAD Auto-Address (Automatische Adressenzuweisung).
Diese Meldung ermöglicht dem Controller, bis zu 31 Geräten auf der Schleife Adressen zuzuweisen. Der Controller sendet die AAD-Meldung an das erste Gerät auf der Schleife. Dieses Gerät übernimmt die Zahl AAAAA als seine Adresse, erhöht dann die Zahl AAAAA und gibt sie an das nächste Gerät auf der Schleife weiter.
(101 100 AAAAA)

AAG Auto-Address Group (Automatische Adressierungsgruppe). Diese Gruppe von Meldungen wird vom Controller gesendet, um den Geräten auf der Schleife Adressen zuzuweisen.
(101 100 XXXXX)

AAU Auto-Address Unconfigure (Löschung der automatisch zugewiesenen Adressen). Diesen Befehl sendet der

Controller, damit alle Geräte, die dazu in der Lage sind, ihre gegenwärtige Adresse löschen. Wenn ein Gerät einmal den AAU-Befehl erhalten hat, kann es durch den AAD-Befehl wieder neu adressiert werden.

(100 100 11010)

Abfrage: Eine Methode, die es dem Controller ermöglicht, ein Gerät herauszufinden, das Service benötigt. In HP-IL Systemen gibt es dazu zwei Methoden: die serielle und die parallele Abfrage.

ACG Addressed Command Group (Befehlsgruppe mit Adressen). Diese Befehlsgruppe wird vom Controller an ein Gerät gesendet, das vorher als Talker oder Listener adressiert wurde. Befehle dieser Gruppe werden dazu benutzt, einzelne Geräte auf der Schleife zu kontrollieren.

(100 X00 0XXXX)

oder (100 101 XXXXX)

oder (100 110 XXXXX)

AEP Auto-Extended Primary (Automatisch erweiterte Primäradresse). Diese Meldung wird vom Controller gesendet, um einer Gruppe von Geräten eine einzelne Primäradresse (AAAAA) zuzuweisen. Diese Gerätegruppe muß in der Lage sein, erweitertes Adressieren durchzuführen, und es müssen ihr zuvor, durch die AES-Meldung, erweiterte Sekundäradressen zugewiesen worden sein.

(101 101 AAAAA)

AES Auto-Extended Secondary (automatisch erweiterte Sekundäradresse). Diese Meldung wird vom Controller gesendet, um den Geräten, die in der Lage sind erweitertes Adressieren durchzuführen, eine Gruppe von Sekundäradressen zuzuweisen. Ein Gerät, das die AES-Meldung erhält, wird AAAAA als seine Sekundäradresse übernehmen, AAAAA um eins erhöhen und sie an das nächste Gerät auf der Schleife weitergeben.

(101 110 AAAAA)

AMP Auto-Multiple Primary (Automatische Mehrfach-Primäradresse). Diese Meldung wird vom Controller gesendet, um den Geräten, die in der Lage sind mehrfache Adressie-

rung durchzuführen, Primäradressen zuzuweisen. Der Controller sendet die AMP-Meldung, und nacheinander nimmt jedes Gerät die Primäradresse (AAAAA) an, erhöht sie um eins und gibt sie an das nächste Gerät auf der Schleife weiter.
(101 111 AAAAA)

ARG Addressed Ready Group (Adressierte Bereit Gruppe). Diese Gruppe von Meldungen wird hauptsächlich vom Controller gesendet, um einen Talker dazu zu veranlassen, Daten oder Statusinformation zu senden oder um den Datenfluß vom Talker zum Listener zu unterbrechen. Innerhalb der ARG-Meldung gib es zwei Meldungen, die vom Talker gesendet werden, um einem Controller mitzuteilen, daß die Datenübertragung beendet ist.
(101 01X XXXXX)

CMD Command class (Befehlsklasse). Die Klasse der Befehlsmeldungen wird vom Controller an die Geräte auf der Schleife gesendet. Befehle werden benutzt, um Geräte- oder Interface-Funktionen zu kontrollieren.
(100 XXX XXXXX)

Controller Ein Controller ist das Gerät auf der Schleife, das die Funktion hat, die ordnungsgemäße Zusammenarbeit der Geräte auf der Schleife zu überwachen. Es gibt jeweils nur einen aktiven Controller auf der Schleife. Die Kontrolle kann jedoch auf eine Anzahl von anderen Geräten auf der Schleife übertragen werden.

DAB Data Byte (Datenbyte). Das Datenbyte ist die Einheit, die über den HP-IL Bus von einem Talker zu einem Listener gesendet wird. Eine Datenbyte-Meldung beinhaltet Daten in den Bits D7-D0 des Meldungsrahmens. Das höchstwertige Datenbit D7 wird zuerst gesendet. Damit Kompatibilität zwischen den verschiedenen Gerätetypen besteht, werden die Daten normalerweise in ASCII kodiert.
(00X XXX XXXXX)

DCL Device Clear (Gerät löschen). Dieser Befehl wird vom Controller gesendet, um alle Geräte in ihren Ausgangszustand (gelöscht) zurückzusetzen. Damit Geräte auf den

DCL-Befehl reagieren, müssen sie nicht adressiert sein.
(100 000 10100)

DDL Device Dependent Listener (Geräteabhängiger Listener). Dieser Befehl wird vom Controller an die Geräte gesendet, die als Listener adressiert sind. Insgesamt 32 mögliche DDL-Befehle werden über die Bits D4 bis D0 des Meldungsrahmens kodiert. Die Wirkung jedes der 32 DDL-Befehle hängt vom Gerät ab. DDL-Befehle ändern sich von Gerät zu Gerät.
(100 101 XXXXX)

DDT Device Dependent Talker (Geräteabhängiger Talker). Dieser Befehl wird vom Controller an alle Geräte gesendet, die als Talker adressiert sind. Insgesamt 32 mögliche DDT-Befehle werden über die Bits D4-D0 des Meldungsrahmens kodiert. Die Wirkung jedes einzelnen der 32 DDT-Befehle hängt vom Gerät ab.
(100 110 XXXXX)

Device siehe Gerät

DOE Data or End (Daten oder Endmeldung). Die Klasse der Daten- oder Endmeldungen umfaßt acht Datenbits (Bits D7 bis D0), die vom Talker zum Listener über die Schleife gesendet werden. Die Endmeldung ist mit der Datenmeldung identisch, mit der Ausnahme, daß Bit C1 gesetzt ist, um das Ende eines Datensatzes anzuzeigen.
(0XX XXX XXXXX)

EAR Enable Asynchronous Requests (Ermöglichung von asynchronen Service-Anforderungen). Der aktive Controller sendet diese Meldung, um allen Geräten, die in der Lage sind, ihre eigene IDY-Meldung zu erzeugen zu erlauben, diese Meldung selbständig auszusenden, um eine Service-Anforderung anzuzeigen. Wenn ein EAR-Befehl einmal gesendet wurde löscht jeder andere Befehl, außer der EAR- oder LPD-Meldung diese Fähigkeit wieder.
(100 000 11000)

Einfachadresse Eine Einfachadresse besteht aus einem Byte im Adreßfeld eines Meldungsrahmens. Einfachadressen

können bis zu 31 Geräten zugewiesen werden.

END Endbyte. Das Endbyte ist ein Datenbyte mit gesetztem C1 Bit, um das Ende eines Datensatzes anzuzeigen. Das Endbyte veranlaßt nicht das Ende einer Übertragung zwischen Talker und Listener; es signalisiert nur den logischen Endzustand.
(01X XXX XXXXX)

Erweitertes Adressieren erlaubt die Adressierung von bis zu 961 Geräten auf der Schleife. Den ersten 31 Geräten werden durch die AES-Meldung erweiterte Sekundäradressen zugewiesen. Dann wird denselben Geräten durch die AEP-Meldung eine gemeinsame Primäradresse zugewiesen. Dieser Prozeß wird solange wiederholt, bis allen Geräten Adressen zugewiesen worden sind.

ETE End of Transmission Error (Ende der Übertragung Fehler). Diese Meldung sendet der Talker, um einem Controller mitzuteilen, daß er die Datenübertragung abgeschlossen hat und daß ein Fehler in der Übertragung gefunden wurde.
(101 010 00001)

ETO End of Transmission OK (Ende der Übertragung OK). Diese Meldung wird vom Talker an den Controller gesendet, um ihm mitzuteilen, daß die Datenübertragung beendet ist und daß alle Daten wieder korrekt zurückerhalten wurden.
(101 010 00000)

Extended Addressing siehe Erweitertes Adressieren

Gerät Eine physikalische Einheit. Geräte und Controller kommunizieren miteinander, indem sie Meldungen über ein gemeinsames Schleifen-Interface schicken.

GET Group Execute Trigger (Auslösen einer Gerätegruppe). Dieser Befehl wird vom Controller an die adressierten Listener auf der Schleife gesendet. Wenn ein Gerät die GET-Meldung erhält, beginnt es seine speziellen Funktionen auszuüben. Die Funktionen, die auf

diese Art und Weise ausgeführt werden können, sind geräteabhängig.

(100 000 01000)

GTL Go To Local (Umstellung auf manuelle Bedienung)

Dieser Befehl dient dazu, allen Geräten, die Listener sind mitzuteilen, daß sie jetzt, statt geräteabhängige Befehle über das Schleifen-Interface zu empfangen, über ihre Einstellknöpfe instruiert werden.

(100 000 00001)

IAA Illegal Auto Address (Unzulässige automatische Adresse). Diese Meldung ist eine Form der AAD-Meldung, die dem Controller anzeigt, daß die Anzahl der Geräte auf der Schleife größer oder gleich 31 ist (die Höchstzahl, die die einfache Adressierung erlaubt). Sie ist als AAD-Meldung definiert, deren Adreßfeld AAAAA die Adresse 31 enthält.

(101 100 11111)

IDY Identify (Identifikation einer Service-Anforderung. Diese Klasse von Meldungen sendet der Controller, um festzustellen, ob ein Gerät auf der Schleife Service benötigt. Diejenigen Geräte, die Service benötigen, können die Aufmerksamkeit des Controllers erwecken, indem sie Bit C0 des Meldungsrahmens setzen. Wenn die parallele Abfrage durchgeführt wird, können die Geräte außerdem die Bits D0 bis D7 setzen, um spezielle Service-Anforderungen anzuzeigen.

(11X XXX XXXXX)

IEP Illegal Extended Primary (Unzulässige Primäradresse). Dies ist die Form der AEP-Meldung, deren Adressfeld AAAAA die Adresse 31 enthält. Geräte antworten nicht auf die IEP-Meldung.

(101 101 11111)

IES Illegal Extended Secondary (Unzulässige erweiterte Sekundäradresse). Eine AES-Meldung, die um eins erhöht wurde, so daß ihr Adreßfeld AAAAA die Adresse 31 enthält, nennt man IES-Meldung. Sie zeigt dem Controller an, daß mehr Geräte als die bereits adressierten eine Ad-

- resse benötigen.
(101 110 11111)
- IFC Interface Clear (Interface löschen). Der System-Controller sendet diese Meldung an alle Geräte und Controller auf der Schleife. Sie veranlaßt alle aktiven Geräte, sich in einen Ruhezustand zurückzusetzen, zerstört jedoch nicht die den Geräten vorher zugewiesenen Adressen.
(100 100 10000)
- IMP Illegal Multiple Primary (Unzulässige Mehrfach-Primäradresse). Dies ist die Form der AMP-Meldung deren Adreßfeld die Adresse 31 enthält. Sie zeigt dem Controller an, daß die Anzahl der Geräte auf der Schleife mit der Fähigkeit einer Mehrfach-Adresse größer oder gleich 31 (der maximal zulässigen Anzahl) ist.
(101 111 11111)
- LAD Listen Address (Listener Adresse). Der Controller sendet diesen Befehl, um ein Gerät zum aktiven Listener auf der Schleife zu machen. Aktive Listener können Daten vom aktiven Talker empfangen. Wenn ein Gerät eine LAD-Meldung erhält, deren Adreßfeld AAAAA mit seiner ihm zugewiesenen Adresse übereinstimmt, wird es zum aktiven Listener auf der Schleife. Wenn ein Gerät für erweitertes oder mehrfach Adressieren konfiguriert wurde, enthält das LAD-Adreßfeld nur die Primäradresse. In diesem Fall wird das Gerät erst dann aktiv, wenn es den entsprechenden SAD-Befehl erhält.
(100 001 AAAAA)
- LAG Listen Address Group (Gruppe der Listener Adressierung). Der Controller benutzt diese Befehlsgruppe, um Geräte auf der Schleife zu aktiven Listenern zu machen.
(100 001 XXXXX)
- Listener (Hörer). Ein Listener ist ein Gerät, dem die Fähigkeit zugewiesen wurde, Daten zu empfangen, die über die Schleife gesendet werden. Listener werden durch die Listen Address Group (LAD) Befehle aktiviert.

LLO Local Lockout (Blockieren der Einstellknöpfe)
Dieser Befehl dient dem Controller dazu zu verhindern, daß das Gerät über seine Einstellknöpfe manuell beeinflusst werden kann. Wenn das Gerät den LLO-Befehl einmal erhalten hat, wird es nicht mehr auf seine Einstellknöpfe reagieren.
(100 000 10001)

LPD Loop Power-Down (Schleife auf niedrige Leistungsaufnahme). Dieser Befehl, den der Controller sendet, veranlaßt alle Geräte, die diese Fähigkeit besitzen, sich in einen Zustand mit niedriger Leistungsaufnahme zu versetzen. Wenn sich ein Gerät in diesem Zustand befindet, wird es sofort wieder aktiviert, wenn irgendeine andere Meldung über die Interface-Schleife kommt. Ein Gerät, das sich in diesem Ruhezustand befindet, verbraucht sehr wenig Energie.
(100 100 11011)

Mehrfach-Adressierung Durch die Mehrfach-Adressierung können einem einzelnen Gerät bis zu 31 spezielle Adressen zugewiesen werden, die aus einer einzelnen Primäradresse und bis zu 31 Sekundäradressen besteht. Mehrfach-adressierten Geräten werden durch die AMP- und ZES-Meldung Adressen zugewiesen.

Meldung Eine Meldung ist die Information innerhalb eines Meldungsrahmens. Wenn die Meldung ein Befehl ist, können die Bits D7-D0 als Teil des Befehls benutzt werden. Enthält die Meldung Daten, enthalten die Bits D7-D0 das acht-Bit-Datenwort, das vom Talker zum Listener gesendet wird.

Meldungsrahmen ist die logische Einheit, die vom Controller zum Gerät oder vom Talker zum Listener über die Schleife gesendet wird. Er umfaßt elf Bits. Die ersten drei Bits (C2, C1, C0) bestimmen die Klasse und Art der Meldung.

MLA My Listen Address (Meine Listener Adresse).
Wenn ein Gerät eine LAD-Meldung empfängt, deren Adressenbits AAAAAA identisch mit der ihm vorher zugewiesenen Adresse sind, wird es dadurch zum aktiven Listener. Aktive

Listener empfangen Daten von einem aktiven Talker.
(100 001 AAAAA)

MSA My Secondary Address (Meine Sekundäradresse).
Wenn ein Gerät, das eine zwei-Byte erweiterte oder Mehr-
fach Adresse benötigt, eine SAD-Meldung erhält, deren
Adressenbits AAAAA identisch mit seiner eigenen Sekundär-
adresse sind, wird es aktiv.
(100 011 AAAAA)

MTA My Talk Address (Meine Talker Adresse). Empfängt
ein Gerät eine TAD-Meldung, bei der die Adressenbits AAAAA
identisch sind mit seiner eigenen Sekundäradresse, wird es
aktiv.
(100 010 AAAAA)

Multiple Address siehe Mehrfach Adressierung

NAA Next Auto Address (nächste automatische Adresse).
Eine AAD-Meldung, deren Adresse von einem Gerät um eins
erhöht worden ist, wird als NAA-Meldung bezeichnet.
(101 100 AAAAA)

NES Next Extended Secondary (Nächste erweiterte
Sekundäradresse). Eine AES-Meldung, deren Adreßfeld AAAAA
von einem Gerät um eins erhöht wurde, nennt man NES-
Meldung.
(101 110 AAAAA)

NMP Next Multiple Primary (Nächste Mehrfach-Primär-
adresse). Diese Meldung ist eine AMP-Meldung, deren
Adreßfeld AAAAA von einem Gerät, das in der Lage ist,
mehrfach adressiert zu werden, um eins erhöht wurde.
(101 111 AAAAA)

NRD Not Ready For Data (Nicht bereit für Daten).
Der Controller kann eine gerade zwischen Talker und
Listener ablaufende Datenübertragung unterbrechen, indem
er ein Datenbyte durch die NRD-Meldung ersetzt. Wenn der
Talker die NRD-Meldung erhält, beendet er die Übertragung.
Wenn der Controller die NRD-Meldung von der Schleife
zurück erhält, gibt er das jeweils letzte Datenbyte wieder

frei.

(101 010 00010)

NRE Not Remote Enable (Nicht fernsteuerbar). Der Controller sendet diesen Befehl an alle Geräte auf der Schleife. Er befähigt ein Gerät, das zuvor alle Meldungen über das Schleifen-Interface erhalten hat, nun über seine Einstellknöpfe gesteuert zu werden. Wenn eine NRE-Meldung einmal empfangen wurde, reagiert dieses Gerät nicht mehr auf Befehle, die über das Schleifen-Interface gesendet werden.

(100 100 10011)

NUL Null Command (Leerbefehl). Der Leerbefehl wird auf der Schleife einfach weitergegeben. Er löst keine Aktivität aus (keine Operation). Er wird in Fällen benutzt, in dem Geräte auf der Schleife aus einem Ruhezustand zurückgerufen werden, ohne daß das Gerät eine bestimmte Aktivität ausführen soll. Der Leerbefehl kann auch in Fällen gesendet werden, in denen es nötig ist, eine Meldung um die Schleife zu schicken, ohne ein bestimmtes Gerät zu betreffen.

(100 000 00000)

OSA Other Secondary Address (andere Sekundäradresse). Ein Gerät, das eine SAD-Meldung erhält, deren Adressenbits AAAAA nicht mit seiner eigenen Sekundäradresse übereinstimmem, kann nicht aktiv werden. Talker, die die OSA-Meldung erhalten, werden inaktiv.

(100 011 AAAAA)

OTA Other Talker Address (Andere Talker Adresse). Ein Talker, der eine TAD-Meldung erhält, deren Adressenbits AAAAA nicht mit der ihm vorher zugewiesenen Talker-Adresse übereinstimmen, werden inaktiv, da nur jeweils ein Talker aktiv sein kann.

(100 010 AAAAA)

Parallele Abfrage erlaubt es dem Controller, schnell ein Gerät herauszufinden, das Service benötigt. Der Controller sendet eine IDY-Meldung. Geräte setzen die vorher zu-

gewiesenen Bits in die IDY-Meldung, um die Service-Anforderung anzuzeigen.

Parallel Polling siehe Parallele Abfrage

Polling siehe Abfrage

PPD Parallel Poll Disable (Abschalten der Fähigkeit zur parallelen Abfrage). Diese Meldung sendet der Controller, um Geräte, die gerade als Listener adressiert sind, zu veranlassen, die parallele Abfrage einzustellen.
(100 000 00101)

PPE Parallel Poll Enable (Befähigung zur parallelen Abfrage). Diese Meldung sendet der Controller, um ein Gerät, das als Listener adressiert wurde, für die parallele Abfrage zu befähigen. Um einem Gerät ein Service-Anforderungsbit zuzuweisen, werden die Bits D2-D0 benutzt (000=D0, 111=D7). Die Polarität des D3 Bits bestimmt, ob das Gerät das parallele Abfrage-Bit setzt, wenn es Service benötigt (D3=1) oder wenn es keinen Service benötigt (D3=0).
(100 100 0SBBB)

PPU Parallel Poll Unconfigure (Löschen der Fähigkeit zur parallelen Abfrage). Indem der Controller diese Meldung sendet, verhindert er, daß die Geräte auf weitere parallele Abfragen antworten. Damit Geräte den PPU-Befehl befolgen, müssen sie zuvor nicht als Listener adressiert worden sein.
(100 000 10101)

Primäradresse ist das höchstwertige Byte einer zwei-Byte Adresse. Primäradressen werden benutzt, um Geräte zu adressieren, denen erweiterte Adressen zugewiesen wurden, oder die die Fähigkeit der Mehrfach-Adressierung besitzen.

Primary Address siehe Primäradresse

RDY Ready (Bereit). Diese Meldungsklasse wird normalerweise benutzt, um Schleifenoperationen zu kontrollieren. Die Meldungen der Ready-Klasse umfassen den

RFC-Befehl, die Befehle der adressierten Ready-Gruppe und die Befehle der automatischen Adressengruppe.
(101 XXX XXXXX)

REN Remote Enable (Befähigung zur Fernsteuerung). Diesen Befehl sendet der Controller an alle Geräte auf der Schleife. Er befähigt die Geräte dazu, Befehle über das Schleifen-Interface zu empfangen, wenn sie adressiert sind. Wenn der REN-Befehl gesendet wurde, und das Gerät die MLA-Meldung empfangen hat, haben die Einstellknöpfe keinen Einfluß mehr auf das Gerät.
(100 100 10010)

RFC Ready For Command (Befehl ausgeführt). Diese Meldung der Ready-Klasse wird vom aktiven Talker gesendet, um sicherzustellen, daß alle Geräte auf der Schleife einen vorher gesendeten Befehl ausgeführt haben. Jeder Meldung, die ein Controller sendet, muß eine RFC-Meldung folgen.
(101 000 00000)

SAD Secondary Address (Sekundäradresse). Der Controller sendet diesen Befehl an Geräte, die eine zwei-Byte Adresse benötigen. Ein Gerät, das eine Primäradresse erhalten hat, wird beim Erhalt der SAD-Meldung, deren AAAAA mit der ihm vorher zugewiesenen Sekundäradresse identisch ist, aktiv.
(100 011 AAAAA)

SAG Secondary Address Group (Sekundäradressen Gruppe). Diese Gruppe von Befehlen wird von einem Controller gesendet, um denjenigen Geräten eine Sekundäradresse zuzuweisen, die zwei-Byte-, erweitertes oder mehrfaches Adressieren durchführen können.
(100 011 XXXXX)

SAI Send Accessory ID (Sende zusätzliche Identifikation). Der Controller sendet diese Meldung, um den aktiven Talker zu veranlassen, seine Accessory ID zu senden. Die Accessory ID ist normalerweise ein einzelnes Byte, in dem die vier höchstwertigen Bits die Geräteklasse angeben (z.B. Drucker, Massenspeicher usw.) und die vier niedrigstwertigen Bits ein bestimmtes Gerät.
(101 011 00011)

SDA Send Data (Sende Daten). Diese Meldung sendet der Controller, um einem Talker den Beginn einer Datenübertragung zu gestatten.

(101 011 00000)

Secondary Address siehe Sekundäradresse

SDC Selected Device Clear (Löschen eines bestimmten Gerätes). Wenn der Schleifen-Controller diesen Befehl sendet, versetzen sich alle Geräte, die als aktive Listener adressiert sind, in ihren gelöschten Zustand zurück.

(100 000 00100)

SDI Send Device ID (Sende Geräte-Identifikation). Diese Meldung sendet der Controller, damit ein aktiver Talker damit beginnt, seine Geräte-Identifikation zu senden. Normalerweise ist diese Geräte-Identifikation eine ASCII-Zeichenfolge, die aus den zweibuchstabigen Initialen des Herstellers, einer fünfstelligen Geräte-Nummer, einer Modell-Revisions-Nummer und beliebiger zusätzlicher Information des Geräteherstellers besteht.

(101 0110 0010)

Sekundäradresse Das niedrigstwertige Byte einer zwei-Byte Adresse. Sekundäradressen werden benutzt, um Geräte zu adressieren, dessen erweiterte Adressen zugewiesen wurden oder die zur Mehrfach-Adressierung in der Lage sind.

Serielle Abfrage ist eine Abfragemethode, die dem Controller dazu dient, die Geräte auf der Schleife herauszufinden, die Service benötigen. Der Controller sendet eine IDY-Meldung. Diejenigen Geräte, die Service anfordern, setzen das Bit C0 in der Meldung. Bei der seriellen Abfrage muß der Controller jedes Gerät einzeln befragen, um die Quelle der Service-Anforderung herauszufinden.

Serial Polling siehe serielle Abfrage

Simple Address siehe Einfachadresse

SRQ Service Request (Service-Anforderung). Eine Service-Anforderung wird von einem Gerät eingeleitet, um dem Controller anzuzeigen, daß es seinen Eingriff benötigt. Das Gerät setzt das Service-Anforderungsbit C0 des Meldungsrahmens, um den SRQ-Zustand zu signalisieren. Dies kann sowohl in der Klasse der DOE- als auch in der Klasse der IDY-Meldungen geschehen.

(0X1 XXX XXXXX)

oder (111 XXX XXXXX)

SST Send Status (Sende Status). Der Controller sendet diese Meldung, um einem Talker mitzuteilen, daß er nun seine Statusbytes senden soll. Statusinformation kann sich sowohl auf eine Service-Anforderung des Gerätes beziehen als auch auf gerätespezifische Information.

(101 011 00001)

TAD Talk Address (Talker Adresse). Der Controller sendet diesen Befehl, um ein Gerät auf der Schleife zum aktiven Talker zu machen. Wenn ein Gerät den TAD-Befehl erhält, vergleicht es dessen Adresse AAAAA mit der ihm zugewiesenen Talker Adresse. Wenn die beiden Adressen übereinstimmen, wird es zum aktiven Talker der Schleife. Wenn das Gerät für erweitertes oder mehrfaches Adressieren konfiguriert wurde, übernimmt es das Adreßfeld als Primäradresse. Das Gerät wird erst dann aktiv, wenn es den SAD-Befehl erhält.

(100 010 AAAAA)

TAG Talk Address Group (Gruppe der Talkeradresse). Diese Befehlsgruppe wird vom Controller an die Geräte auf der Schleife geschickt, die als Talker aktiv werden sollen.

(100 010 XXXXX)

Talker (Sprecher). Ein Talker ist dasjenige Gerät, dem die Fähigkeit zugewiesen wurde, Daten auf die Interface-Schleife zu senden. Talker werden durch die Gruppe der Talk Address-Befehle (TAG) befähigt.

TCT Take Control (Übernimm die Kontrolle). Der aktive Controller sendet diese Meldung, um die Kontrolle auf

einen anderen Controller auf der Schleife zu übertragen.
(101 011 00100)

UCG Universal Command Group (Universelle Befehlsgruppe). Diese Befehlsgruppe wird vom Controller an alle Geräte auf der Schleife gesendet. UCG-Befehle betreffen alle Geräte, nicht nur diejenigen, die adressiert wurden.
(100 X00 1XXXX)

UNL Unlisten (Aufheben des Listener-Zustands). Der Controller sendet diesen Befehl, um alle adressierten Listener in den inaktiven Zustand zurückzusetzen.
(100 001 11111)

UNT Untalk (Aufheben des Talker Zustands). Der Controller sendet diesen Befehl, um den aktiven Talker in den inaktiven Zustand zurückzusetzen.
(100 010 11111)

ZES Zero Extended Secondary (erweiterte Sekundäradresse Null). Der Controller sendet diese Meldung, um den Geräten Sekundäradressen zuzuweisen, die in der Lage sind, Mehrfach-Adressen anzunehmen. Nachdem er den Geräten mit der AMP-Meldung Primäradressen zugewiesen hat, sendet der Controller die ZES-Meldung nacheinander an alle Geräte, die mehrfach adressiert sind. Ein Gerät, das die ZES-Meldung erhält, erhöht die fünf niedrigstwertigen Bits der ZES-Meldung um eins, um dem Controller die Anzahl der Adressen, die für das Gerät reserviert sind, mitzuteilen.
(101 110 00000)

Liste der Fußnoten

- 1 Interface: engl. f. Schnittstelle
- 2 Loop: engl. f. Schleife
- 3 Controller: engl. f. Aufsicht, Kontrolle
- 4 Handshake: engl. f. Hände schütteln
- 5 Bus: engl. f. Sammelschiene
- 6 General Purpose Interface Bus: engl. f. Allzweck
 Interface Bus
- 7 IEEE: Institute of Electrical and Electronics Engineers,
 amerikanisches Standardisierungskomitee
- 8 Talker: engl. f. Sprecher
- 9 Listener: engl. f. Hörer
- 10 Driver: engl. f. Treiber
- 11 Receiver: engl. f. Sender/ Empfänger
- 12 power-down: engl. f. niedrige Leistung

- 13 trigger: engl. f. auslösen

- 14 zwischenzeitlich ersetzt durch das HP 82165A
 HP-IL/GPIO-Interface

- 15 ROM: Abkürzung f. Read Only Memory, Festspeicher

- 16 Receiver: engl. f. Sender/ Empfänger

- 17 Driver: engl. f. Treiber

- 18 Integrierter Schaltkreis mit hoher Bau-
 teildichte

- 19 PLA: Programmierbarer Logik-Baustein

- 20 power-up: engl. f. einschalten

- 21 Command: engl. f. Befehl

- 22 Ready: engl. f. bereit, fertig

- 23 Bereit-Meldungen

- 24 Modus, in dem der Drucker bestimmte Zeichen zu
 seiner eigenen Steuerung verwendet und sie nicht
 ausdruckt.

- 25 Aufnahmelogik

-
- 26 Remote-Kommando: Umstellung auf Fernsteuerung
Local-Kommando: Umstellung auf manuelle Bedienung
- 27 engl.: Parallel Polling
- 28 Meldungen der Bereitklasse
- 29 Auto-Address: engl. f. automatische Adresse
- 30 Acceptor: engl. f. Aufnahme
- 31 Source: engl. f. Quelle, Sender
- 32 Auslösen der Geräte
- 33 Device Dependent: engl. f. Geräteabhängigkeit
- 34 Driver-Übergang aus dem Acceptor-Zustand
- 35 Driver Übergang aus dem Ruhezustand
- 36 Driver-Übergang aus dem Source-Zustand
- 37 Driver-Übergang aus dem Transfer-Zustand
- 38 Ruhezustand des Listener
- 39 repeat: engl. f. Wiederholung

- 40 norepeat: engl. f. keine Wiederholung
- 41 Ruhezustand des Talkers
- 42 Adressierter Talker-Zustand
- 43 Aktiver Talker-Zustand
- 44 Aktiver Zustand der seriellen Abfrage
- 45 Aktiver Zustand der Geräte-Identifikation
- 46 Aktiver Zustand der zusätzlichen
Geräte-Identifikation

Index

A

AAD 84,115
AAU 85
Abfolge 90
Abfrage 140
Abfrageoperation 67,92
Abfragewort 69
Acceptor Daten-Status 124
Acceptor Handshake 99,103,123
Acceptor Handshake-Interface 126
Acceptor Handshake-Zustandsdiagramm 125
Acceptor-Logik 59
Acceptor Not Ready-Status 127
Acceptor Ready State 123
Accessory Identify Active State 130
ACDS 124
ACG 132
ACRS 123
Adapter 17
Addressed Command Group 132
Addressed Ready Group 133
Adressen-Vergleicher-Logik 58
Adressenzuweisung 8
Adressierungsweise 84,85
Adreßmeldung 96
Adreßzuweisung 96
AE 114
AEP 115,132
AES 115,132
AH 123
AIAS 130
AIDS 127
Allzweck-Interfaces 29
alternativer Zeichensatz 19
AM 114
AMP 115,132,145
Anforderung-Funktionen 112
ANRS 127
ARG 133
ASCII-Standardzeichen 19

asynchrone 85, 86
Ausgaberegister 61
Auto Address 115
Auto-Address-Meldungen 84
Auto-Address Unconfigure 85
Auto-Extended Address 100
Auto Extended Primary 115,132
Auto Extended Secondary 115,132
Automatische Adressierung 11,114
Automatische erweiterte Adresse 100,114
Automatische Mehrfach-Adresse 100,114
Auto Multiple Address 100
Auto Multiple Primary 115,132

B

Bar Codes 20
Basisfunktionen 27
Befehlsabfolge 88
Bitmuster 82
Bus 4

C

CMD 72,133
Command class 133
Command-Klasse 33,63
Controller 6,8,105,133
Controller-Funktion 107

D

DAB 46,133
DAB1 89
DAB2 76
DACS 122
Data Byte 133
Data or End 52,128,134
Datenbyte 46
Datenerfassung 12
Datenfernübertragung 15

Datenklasse 52
Datenleitungen 2
Datenmeldungen 82
Datenpuffer 31
Datenübertragung 13,60,90,91
Datenübertragungsgeschwindigkeit 3
Datenübertragungsrates 3,9
DC2-Fähigkeit 110
DCL 133
DD 116
DDL 116,133
DDT 116,134
Dependent Listener 116
Device Clear 109,133
Device Clear (DCL) 33,37
Device Dependent 100,116
Device Dependent Listener 134
Device Dependent Talker 116,134
Device Identify Active State 130
Device Trigger 100
DIAS 130
DIDS 122,122
DIR 21
DOE 128,134
Drei-Pegel-Codes 9
Driver 28
Driver Idle State 122
Driver-Interface 122
Driver-Logik 61
Driver Transfer State 122
Driver Transmit from Acceptor State 122
Driver Transmit from Source State 122
DSCS 122
DTRS 122

E

EAR 134
Einfachadresse 134
Eingabepuffer 56
Enable Asynchronous Requests 134
END 53,135
Endbyte 135
End Data Byte 53
End of Transmission 135
End of Transmission, Error 71
End of Transmission Error (ETE) 43

End of Transmission, OK 71
End of Transmission OK (ETO) 43
erweiterte Adressierungsmethode 9
Erweitertes Adressieren 135
ETE 71,135
ETO 71,135
Extended Addressing 135

F

Fachbegriffe 131
Fehlerüberprüfung 5,62
Fehlerüberprüfungslogik 62
Fernsteuermodus 65
fertiggekaufte Geräte 17
frame 56
Frame Transmission Complete 123
Fremdgerät 22
frtc 123

G

gekaufte Geräte 17
Gerät 36,143
Geräteanschluß 22
Geräteanzahl 9
Geräte-Interface-Logik 31
Gerätekontrolle 107
Gerätenummer 80
GET 110,135
Go To Local 63,136
GPIB 6
Group Execute Trigger 110,135
GTL 63,111,136

H

Handshake 3
Handshake-Leitungen 24
HP-IB 6
HP-IL 7
HP-IL Allzweckadapter 30
HP-IL Interfaces 27
HP-IL Meldungsprotokoll 15
HP-IL Spezifikation 1
HP-Minidatenkassetten 19

I

IAA 136
Identify 69,136
IDY 69,136
IEEE-488 6
IEP 136
IES 136
IFC 63,88,88,137
Illegal Auto Address 136
Illegal Extended Primary 136
Illegal Extended Secondary 136
Illegal Multiple Primary 137
IMP 137
Impulstransformator 10
Initialisierung 88
Interface 60
Interface-Adapter 41,46
Interface-Chip 55
Interface Clear 137
Interface Clear (IFC) 33
Interface Clear-Meldung 63
Interface-Funktionen 99
Interface-Schleife 8

K

Kommando 65
Kommunikation 1
Kompatibilität 5,6
Konfiguration 94
Kontrolle 97
Kontrollleitungen 2
Kontroll-Logik 31
Konverter 17

L

LAD 137
LAD2 67
LADS 124
LAG 137
LCD-Anzeige 19
Lesesignale 60
Listen Address 137
Listen Address Group 137
Listen Address (LAD) 38
Listener 8,104,137

Listener Addressed State 124
Listener-Interface 124
LL 60
LLO 112,138
Local 65
Local Lockout 111,138
Local-Modus 65
Local Ready 127
Loop Power-Down 138
Loop Power-Down (LPD) 33
löschen 143
LPD 41,109,138
LSI 55

M

Mehrfach-Adressierung 138
Meldungsabfolgen 87
Meldungsprotokoll 5
Meldungsrahmen 56,138
Meldungsverarbeitung 28
Meldungsverarbeitungsblock 28
Meßwerterfassung 14
MLA 124,138
Mobilität 12
Modul 18
MSA 139
MTA 50,139
Multiple Address 139
My Listen Address 124,138
My Secondary Address 139
My Talk Address 50,139

N

NAA 139
NES 139
Next Auto Address 139
Next Extended Secondary 139
Next Multiple Primary 139
niedrige Leistungsaufnahme 41
NMP 139
Nomenklatur erörtern 122
norepeat 126
Not Ready For Data 77,139
Not Ready for Data 71,86
Not Remote Enable 63,140
NRD 71,77,86,139

NRE 63,140
 NUL 140
 Null Command 140

O

Operationen 85,86
 OSA 140
 OTA 51,140
 Other Secondary Address 140
 Other Talker Address 51,140

P

Parallel 140
 Parallele Abfrage 48,66,94,112,141
 paralleles Interface 2
 Parallel Poll Disable 63,70,112
 Parallel Poll Enable 63,67,112
 Parallel Polling 141
 Parallel Poll Unconfigure 63,94,112
 PD 122
 physikalische Verbindung 2
 Polarität 24
 PONS 122
 POS 120
 Power-Down 108
 power-down-Zustand 11
 Power On 122
 Power On Switch 120
 PPD 63,70,112,141
 PPE 63,67,68,112,141
 PPE-Meldungsformat 68
 PPU 63,70,94,141
 Primäradresse 141
 Protokoll 5
 PRREG 22

R

Rahmen-Dekodierungs-Logik 57
 RCDS 128
 RDY 141
 rdy 127
 READR 22
 Ready 141
 Ready For Command 71,142
 Ready For Command (RFC) 43

Ready-Klasse 71
 Receiver 28,102
 Receiver Immediate Transfer State 123
 Receiver-Kontroll-Logik 56,57
 Registerauswahlleitungen 61
 Registerbereich 59
 Registern 19
 Remote 65
 Remote Enable 63,142
 Remote Local 110
 Remote-Modus 65
 REN 63,111,142
 repeat 126
 RFC 68,71,71,88,142
 Ringinterface 4
 RITS 123
 RL 110
 RS-232 23
 RSO-RS 61

S

SAD 142
 SAG 142
 SAI 142
 SBS 120
 Schleifen-Interface-Funktionen 27
 Schleifen-Nomenklatur 34
 Schreibsignale 60
 SDA 71,74,143
 SDC 63,110,143
 SDI 71,80,143
 SDLC 6
 Secondary Address 142,143
 Secondary Address Group 142
 Sekundäradresse 143
 Sekundär 85
 Selected Device Clear 63,100
 Send Accessory ID 142
 Send Data 71,74,143
 Send Device ID 71,80,143
 Send Status 71,144
 Send Status (SST) 43
 Serial Poll Active State 130
 seriell 92
 Serielle Abfrage 48,143

serielles Interface 3
serielles Kommunikationsinterface 7
Service-Anforderung 82,113
Service-Anforderungsbit 83
Service Request 144
Simple Address 143
Sonderzeichen 20
SOT 128
Source Handshake 99,104
SPAS 130
Spulgeschwindigkeit 19
SRQ 82,144
SS 60
SST 66,71,144
STANDBY 120
Standby-Fähigkeit 41
Standby-Zustand 32
Start Of Transmission 128
Statuswort 24
Steuermeldungen 131
Suchgeschwindigkeit 19
Synchronisationsbit 56
System 7

T

TACS 130
TAD 49,144
TADS 130
TAG 144
Take Control 71,80,144
Talk Address 144
Talk Address Group 144

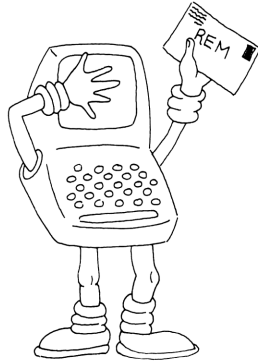
Talk Address (TAD) 33
Talker 8,105,144
Talker Active State 130
Talker Addressed State 130
Talker-Befehle 49
Talker Idle State 130
Talker-Interface 129
Taschencomputer 18
TCT 71,80,144
TIDS 130
To Local 111
Triggern 11

U

Übertragung 97
UCG 145
Unconfigure 70
Universal Command Group 145
UNL 40,145
Unlisten 145
UNT 49,145
Untalk 145
Unterbrechung 91
Untergruppen 107

Z

Zeichenketten 19
Zero Extended Secondary 116,145
ZES 116,145
Zustandsdiagramm 120,122,124,126,129



L. Poole
Praktische
BASIC-Programme
für den IBM Personal Computer

1983. ISBN 3-89028-000-5,
172 Seiten, DM 29,80

Die Anzahl der im Augenblick erhältlichen IBM-PC-Software ist noch sehr begrenzt. Mit diesem Buch stehen nun den Benutzern von IBM-PCs 34 praktische Programme in deutscher Sprache zur Verfügung, die unmittelbar so eingegeben werden können, wie sie im Buch abgebildet sind. Für wenig Geld erhält man damit nützliche Anwendungen für das Büro, für Übungskurse und für zuhause.

A. Osborne / G. Eubanks / M. McNiff
CBASIC Anwenderhandbuch

1984. ISBN 3-89028-006-4, 215 Seiten

Der hier beschriebene populäre BASIC-Compiler für 8080-, Z80- oder 8085-Mikrocomputer ist ebenfalls leicht zu erlernen wie BASIC. Eine einfache und schnelle Programmiersprache, um hochorganisierte, lesbare und modular aufgebaute, strukturierte Programme zu erstellen. Dieses Standardwerk wurde unter Mitarbeit des Erfinders von CBASIC, Gordon Eubanks, verfaßt.

H. Peckham
BASIC für den
IBM Personal Computer

1984. ISBN 3-89028-001-3, 348 Seiten

Mit diesem Buch erlernt man leicht und schnell die Programmiersprache BASIC des IBM-PCs. Jedes Kapitel beginnt mit einer kurzen Darstellung der Lernziele. Übungen begleiten die folgenden Lernabschnitte und vermitteln praktische Fähigkeiten im Umgang mit dem Computer. Durch die selbständige Erarbeitung erwirbt der Leser bei einem Minimum an Unterweisung ein solides theoretisches und praktisches Wissen. Das Buch ist für den Kursunterricht ebenso geeignet wie für das Selbststudium.

Weitere Bücher zum Thema:

Eberhard Bötzel
Statistische Problemlösungen mit
dem SHARP PC 1500/1500A

1984. ISBN 3-89028-013-7,
164 Seiten, DM 35,—

Mit der in den Modulen CE-501 B, CE 502A und CE 502B enthaltenen Standardsoftware wird der SHARP PC-1500 zu einer äußerst leistungsfähigen Datenverarbeitungsanlage im Taschenformat, die die Lösung aller gängigen statistischen Probleme bis hin zur Multiplen Regression ermöglicht. Dieses Buch beschreibt die Arbeitsweise und Anwendungsmöglichkeiten der SHARP-Module. Die Programme sind so ausführlich dargestellt, daß zur Anwendung keine wesentlichen Vorkenntnisse benötigt werden. Jedes Programm wird Schritt für Schritt erklärt und der Programmablauf im Detail erläutert. Dazu wird jeweils ein Anwendungsbeispiel gegeben.

T. Hogan
CP/M Anwenderhandbuch

1984. ISBN 3-89028-005-6, 286 Seiten

Dieses Anwenderhandbuch zum bekanntesten Mikrocomputer-Betriebssystem der Welt diskutiert die neuesten CP/M-Entwicklungen einschließlich CP/M 86 und das Betriebssystem für 8086- und 8088-Microcomputer, wie zum Beispiel den IBM-PC. Der Band stellt auch CP/M-Ableitungen wie MP/M und CP/NET vor.

H. McGilton/R. Morgan
Einführung in das UNIX-System

1984. ISBN 3-89028-003-X, 480 Seiten

Ein Lehrbuch für den Neuling und ein vollständiges Nachschlagewerk für den erfahrenen Anwender. Das Buch stellt die Möglichkeiten des UNIX-Systems anhand aktueller und volausgetesteter Beispiele des UNIX-Codes vor.



Hamburg

Das HP-IL-System Einführung in die Hewlett-Packard Interface-Schleife

von Gerry Kane, Steve Harper & David Ushijima

Ein verbindlicher Führer zum Verständnis des neuen Hewlett-Packard Interface Loop (HP-IL), einem 8-Bit seriellen Interface, das für tragbare, kostengünstige, batteriebetriebene Systeme entworfen wurde. Das HP-IL ermöglicht eine neue Generation von tragbaren Systemen zum Anschluß von Meßgeräten und Peripherie.

Das HP-IL verbindet programmierbare Taschencomputer, wie den HP-41, und die Personal Computer der Serie 80 mit kostengünstigen Peripheriegeräten und Testinstrumenten mit niedriger Leistungsaufnahme. Für die Anwendung außer Haus, wo Tragbarkeit eine Notwendigkeit darstellt, ist es ein unersetzbares Werkzeug für die Datenerfassung, für die Vermeidung von unnötiger Schreibeinheit, für Unterwegsberechnungen und für Laboraufbauten.

Der Coautor Steve Harper arbeitet in der Forschungs- und Entwicklungsabteilung von Hewlett-Packard in Corvallis, Oregon. Er begleitet Gerry Kane und David Ushijima bei der detaillierten Beschreibung des HP-IL und seiner Funktionen. Es wird der Anschluß von HP-IL Geräten und von Fremdgeräten ohne HP-IL Interface angesprochen. Weiterhin werden Zusammenfassungen aller HP-IL Steuermeldungen und Meldungsabfolgen gegeben.

ISBN 3-89028-009-9

