

Interfaces und digitale Datenübertragung (GPIB)

1. Einführung

In der modernen physikalischen Meßtechnik und Prozeßtechnik werden in vielfältiger Weise verschiedene Daten ausgetauscht. So kommuniziert z.B. ein Labor-PC mit unterschiedlichen Meßgeräten oder ein Prozeßrechner erhält Meßdaten von verschiedenen Sensoren und gibt Steuersignale zur Steuerung eines Prozesses aus. Die Datenkommunikation erfolgt über entsprechende Verbindungen, die allgemein als Schnittstellen bezeichnet werden. Die dafür erforderliche elektronische Schaltung und dazugehörige hardware-nahe Software faßt man unter dem Begriff Interface zusammen.

Aus naheliegenden praktischen Gründen (Vereinheitlichung und Austauschbarkeit) sind häufig verwendete Schnittstellen genormt bzw. standardisiert worden. Das betrifft die mechanische Ausführung der elektrischen Verbindung (Stecker, Buchsen, Verbindungskabel) und deren Belegung ebenso wie die elektrischen Eigenschaften (Strom- bzw. Spannungspegel, Frequenzen, Signalformen) und die Funktionen der einzelnen Verbindungsleitungen (Daten-, Steuer-, Adreß- Leitungen). Für das sog. Schnittstellen-Protokoll, das die Datenart und den Übertragungsablauf genau definiert, gilt ebenso eine jeweilige Norm.

Wahrscheinlich sind Ihnen aus eigener Erfahrung mit der Arbeit am PC schon einige Schnittstellen namentlich bekannt, wie etwa die RS232/COMx-Port (z.B. zum Anschluß einer seriellen Maus), die parallele Schnittstelle LPTx zum Anschluß eines Druckers oder die neuere USB-Schnittstelle.

Mit diesem Versuch sollen Sie die Grundprinzipien einer digitalen Datenübertragung kennenlernen und die Funktion eines ausgewählten Datenbus-Systems verstehen. Aufgrund der Vielzahl praktisch bedeutsamer Schnittstellensysteme muß eine Auswahl getroffen werden, für diesen Versuch beschränken wir uns auf die bei meßtechnischen Anwendungen häufig anzutreffende Schnittstelle IEEE488 (IEC60625-Bus/GPIB/HPIB). Diese Architektur ist weit verbreitet für die Verbindung von Meßgeräten untereinander und mit Rechnern.

2. Grundbegriffe

Kodierung

Zu verarbeitende und zu übertragende Daten liegen in binärer Form, d.h. in einzelnen Bits bzw. gruppiert in Bytes, vor. Mehrere Bits bzw. auch Bytes in gruppierter Form nennt man Datenworte oder Binärworte. Diese Worte können z.B. eine Zahl, einen Buchstaben oder ein Steuerzeichen bezeichnen. Zur Datenübertragung ist eine Kodierung aller Signale erforderlich. Die Kodierung von Ganzzahlen in Binärkode hatten wir bereits bei der AD/DA-Wandlung kennengelernt. Ein weiteres Beispiel ist der bekannte ASCII-Kode.

Parallele Übertragung

Für diese Form der Datenübertragung ist für jedes einzelne Bit eines Wortes eine Leitung erforderlich, die jeweils ein datensendendes Logik-Gatter mit einem empfangenden verbindet. Die Daten werden zeitlich parallel übertragen, daher die Bezeichnung. Die erzielbaren Datenraten dieser Übertragung sind durch die Gattergeschwindigkeiten und die Grenzfrequenz des verwendeten Kabels bestimmt. Bei kurzen Entfernungen oder innerhalb von Geräten ist diese Übertragungsart häufig anzutreffen. Für den Fall, daß die Daten bidirektional ausgetauscht werden müssen, ist der Aufwand an erforderlichen Leitungen doppelt so hoch.

Serielle Übertragung

Hier sind bedeutend weniger Übertragungsleitungen erforderlich, da die Einzelbits jedes Datenwortes zeitlich nacheinander (seriell) übertragen werden. Allerdings muß vorher eine Umwandlung parallel vorliegender Daten in einen sog. Bit-Strom vorgenommen werden, was einen etwas höheren Schaltungsaufwand erfordert. Die Übertragungszeit für ein Datenwort ist abhängig von der Anzahl der Bits. Im Vergleich zu einer parallelen Übertragung wird man für ein Datenwort gleicher Länge mehr Zeit benötigen.

Simplex/Duplex/Halbduplex-Übertragung

Die Übertragungsrichtung ist ein weiteres Merkmal einer Schnittstelle. Mit Simplex-Übertragung bezeichnet man die Übertragung nur in einer Richtung. Ist die Richtung umschaltbar, spricht man von Halbduplex-Übertragung. Wenn eine Übertragung gleichzeitig in beiden Richtungen erfolgen kann, liegt eine Duplex- Übertragung vor. Häufig kann man mit ein und derselben Schnittstelle alle drei Modi realisieren, so z.B. auch mit der bekannten RS232.

Stern-Systeme/Bus-Systeme

Einzelkomponenten eines Systems können nach zwei unterschiedlichen Strukturkonzepten verbunden sein:

Bei einem Sternsystem wird von der Zentraleinheit zu jeder Komponente eine eigene Leitungsgruppe geführt. Ein Beispiel dafür wäre die serielle Kopplung mehrerer Geräte (wie Maus, Modem) an einen PC über die seriellen Ports COM1..COM4 - jede Komponente hat einen eigenen seriellen Anschluß.

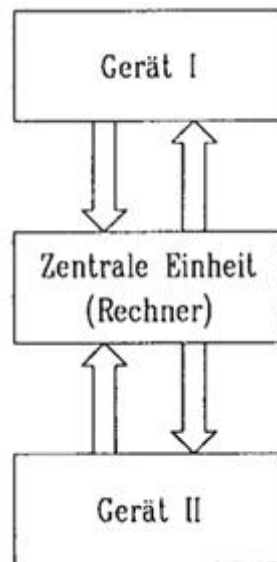


Abb.1 Sternverbindung

Dagegen sind bei einer Bus-Struktur alle Einzelkomponenten durch dieselbe Leitungsgruppe verbunden. Damit läßt sich der Verkabelungsaufwand beträchtlich reduzieren, da die Leitungen von einem Gerät zum nächsten einfach weitergeführt werden.

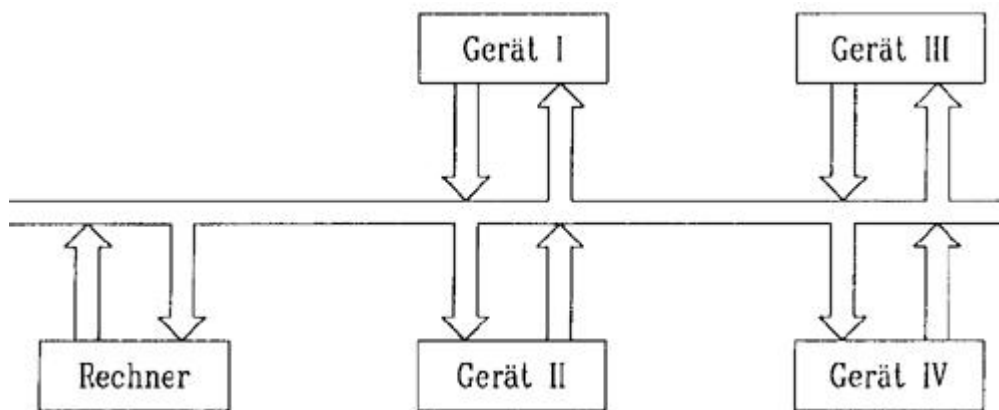


Abb.2 Buskonzept

Es ist leicht einzusehen, daß man hier aus praktischen Gründen die nötigen Verbindungen wie Stecker, Steckerleisten und Kabel standardisieren muß. Ein Bus kann entweder seriell oder parallel Daten übertragen. Weil an einem Bus alle Geräte untereinander mit den gleichen Leitungen verbunden sind, können sie alle die gleichen Leitungen beeinflussen. Außerdem ist die Richtung der Datenübertragung meist nicht fest, so daß alle Geräte im allgemeinsten Fall am Bus sowohl Daten senden als auch empfangen können müssen. Um nun einen gezielten (eindeutig gerichteten) Datenaustausch zwischen mehreren verschiedenen Komponenten zu ermöglichen, muß ein Bus-System entsprechend weiter strukturiert werden: Man unterscheidet zwischen Datenbus, Adreßbus und Steuerbus. Über den Datenbus werden die eigentlichen Daten übertragen; über den Adreßbus festgelegt (adressiert), welche Komponente Daten senden oder empfangen soll. (Manche Bussysteme verfügen über keinen eigenen Adreßbus - hier werden Daten und Adressen über den Datenbus übermittelt und mit dem logischen Pegel einer zusätzlichen Leitung jeweils im Einzelfall festgelegt, ob gerade eine Adresse oder ein Datenwort übertragen wird.) Mit dem Steuerbus wird der Datenaustausch kontrolliert (insbesondere Zeitpunkt der Übertragung gültiger Daten, Bestätigung korrekter Übertragung).

Synchrone/Asynchrone Übertragung

Die Datenübertragung kann synchron oder asynchron erfolgen. Bei einer synchronen Datenübertragung wird ein Takt- oder Synchronsignal über eine gesonderte Leitung übertragen. Dagegen ist bei asynchroner Übertragung keine zeitliche Festlegung gegeben, das Signal "Daten gültig" wird aus den Daten selbst hergeleitet.

Handshaking

Ein wichtiges Verfahren für asynchrone Datenübertragung ist das sog. Handshaking. Für einen einfachen Handshake kommt man mit zwei Leitungen aus: Auf der ersten signalisiert der Datensender, daß ein Datenwort auf dem Datenbus übertragen wird. Auf der zweiten Leitung wird durch den Datenempfänger signalisiert, daß das Datenwort übernommen wurde.

3. IEC625-/IEEE488-Bus/GPIB/HPIB

3.1 Vorbemerkungen

Dieser Bus ist weltweit quasi das Standard- Bussystem der Meßtechnik, das sich insbesondere für die Verbindung von Meßgeräten untereinander und mit Rechnern durchgesetzt hat. Dafür existieren im internationalen Sprachgebrauch die verschiedenen (aber synonymen) Bezeichnungen

<i>Name</i>	<i>Ursprung</i>	<i>URL</i>
IEC60625-Bus	Standard der IEC (International Electrotechnical Commission)	http://www.iec.org
IEEE488-, IEEE488.1-, IEEE488.2-Bus	Standards des IEEE (Institute of Electrical and Electronics Engineers, USA)	http://www.ieee.org
GPIB	General Purpose Interface Bus, National Instruments	http://www.gpib.com
HPIB	Hewlett Packard Interface Bus, Hewlett Packard	http://www.hp.com

Das Bussystem wurde ursprünglich von Hewlett-Packard (HP) entwickelt (als HPIB), um programmierbare Meßgeräte zu verbinden und zu steuern. Aufgrund der hohen Datenübertragungsgeschwindigkeit von 250 Kilobyte bis 1 Megabyte pro Sekunde setzte sich der GPIB schnell in solchen Anwendungen wie Computerverbindungen und Peripheriesteuerung durch. Er wurde später als Industriestandard IEEE-488 bzw. IEC60625 angenommen - einer der (i.a. leider seltenen) Fälle, bei dem die Vorarbeit eines Herstellers relativ schnell in einen Standard umgesetzt wurde. Die Vielseitigkeit des Systems führte zur weit verbreiteten Bezeichnung "General Purpose Interface Bus" (GPIB).

Die Fa. National Instruments (NI) erweiterte die Nutzung des GPIB auf Computer anderer Hersteller als HP und leistete weitere Entwicklungsarbeit zur Erweiterung der hohen Leistungsfähigkeit: Es entstanden High-Speed- Hardware-Schnittstellen und vielseitige Software mit umfangreichen Funktionen zur Unterstützung der Anwender beim Aufbau und der Programmierung von komplexen und leistungsfähigen Meßsystemen.

Der IEEE/IEC-Standard definiert das Interface als byte-serielle, bit-parallele und asynchrone sowie bidirektionale Datenübertragung. Dieser Standard umfaßt elektrische Spezifikationen, Kabel/Stecker/Buchsen/Verbinder, Kontrollprotokoll und Kommando- Sequenzen zum Datenaustausch zwischen Geräten. Dadurch können Geräte verschiedenster Hersteller eingesetzt werden, sofern sie diesem Standard genügen. Für den Anwender ergibt sich der große Vorteil der Universalität, für eine konkrete Applikation können unterschiedliche vorhandene bzw. am besten geeignete Geräte eingesetzt werden - ein weiterer Ausbau mit zusätzlichen Geräten ist vergleichsweise einfach.

3.2 Grundlegende Spezifikationen des GPIB

Mit dem GPIB liegt ein Bussystem vor, auf dem die Daten parallel übertragen werden. Die Verbindungskabel können von einem Gerät zum anderen durchgeschleift werden, mehrere Geräte können parallel an das Bussystem geschaltet und einzeln adressiert werden. Im Unterschied zum Bus in einem Computer werden im GPIB völlig eigenständige und im Prinzip voneinander unabhängige Geräte über einen Kabelbus verbunden.

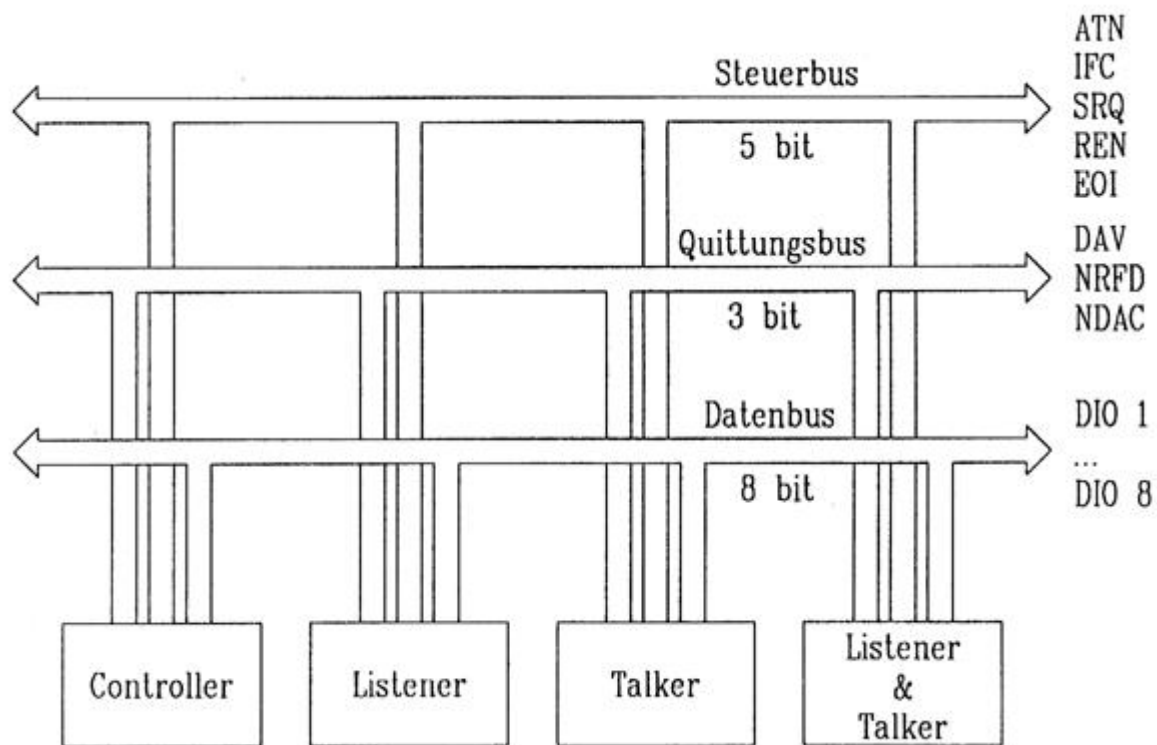


Abb.3 Geräte am GPIB / Struktur des GPIB (schematisch)

3.2.1 Talker, Listener und Controller

Ein sog. Talker (Sprecher) sendet am GPIB Datennachrichten zu einem oder mehreren Listnern (Hörern). Der Controller steuert den Informationsfluß auf dem GPIB durch das Senden von entsprechenden Kommandos zu allen Geräten.

Einzelne Geräte am GPIB können Listener, Talker und/oder Controller sein. So kann z.B. in einer Anwendung ein Digitalvoltmeter sowohl als Talker als auch Listener betrieben werden.

Der GPIB-Controller überwacht die Kommunikation im Bus in ähnlicher Weise wie eine Telefonvermittlung: Bemerkt er, daß ein Gerät eine Datennachricht senden will, so wird dieser Talker mit dem entsprechenden Listener verbunden. Meist werden dabei durch den Controller Talker und Listener adressiert, noch bevor der Talker seine Nachricht zum Listener absetzen kann. Nach Abschluß dieser einzelnen Übertragung werden beide Geräte gewöhnlich wieder entadressiert. U.U. ist für einige Buskonfigurationen kein Controller erforderlich; es gibt Geräte, die nur als Talker (Talk-only) oder Listener (Listen-only) arbeiten. Ein Controller ist aber immer dann notwendig, wenn der aktive oder adressierte Talker oder Listener geändert werden muß. Die Controllerfunktion wird i.a. vom Computer geleitet. Eine GPIB-Interfacekarte und entsprechende Software gestatten einem Computer, alle drei Geräterollen auszuführen: als Controller (Leitung/Kontrolle des GPIB), als Talker (Daten senden) und als Listener (Daten empfangen).

3.2.2 GPIB-Signale und Leitungen

Der GPIB enthält einen Datenbus, einen Quittungsbus und einen Steuerbus. Das Interface-System verfügt insgesamt über 16 Signalleitungen und 8 Massrückleitungen oder abgeschirmte Ableitungen. Die 16 Signalleitungen sind in folgende drei Gruppen unterteilt: 8 Datenleitungen, 3 Handshakeleitungen, 5 Interfaceleitungen.

Datenleitungen (8)

mit der Bezeichnung DIO1..8 (data in/out) übertragen die Daten- und Kommando-Nachrichten. Alle Kommandos und die meisten Daten benutzen den 7-Bit ASCII- oder ISO-Kode; das 8.Bit (DIO8) wird entweder gar nicht oder als Paritätsbit benutzt.

Handshakeleitungen (3)

leiten asynchron die Übertragung von Nachrichtenbytes unter den Geräten. Dieses Verfahren ist ein ineinander verschachteltes Drei-Leiter-Handshaking und garantiert Senden und Empfangen von Nachrichtenbytes auf den Datenleitungen ohne Übertragungsfehler.

- **NRFD** (not ready for data / nicht bereit für Daten) zeigt an, ob ein Gerät bereit oder nicht bereit ist, Nachrichtenbytes zu empfangen. Die Leitung wird zum Empfang von Kommandos und Datennachrichten betrieben.
- **NDAC** (no data accepted / Daten nicht angenommen) zeigt an, ob ein Gerät ein Nachrichtenbyte angenommen

- hat oder nicht. Die Leitung wird ebenfalls beim Empfang von Kommandos und Datennachrichten betrieben.
- **DAV** (data valid / Daten gültig) teilt mit, ob die Signale auf den Datenleitungen stabil sind und von Geräten sicher übernommen werden können. Diese Leitung wird vom Controller zum Senden von Kommandos und vom Talker zum Senden von Datennachrichten benutzt.

Die Datenübertragung erfolgt im GPIB asynchron. Der eingesetzte Handshake mit 3 Leitungen ist flexibel und paßt sich universell an die angeschlossenen Geräte an. Dabei können Daten von einem Talker an mehrere Listener übertragen werden.

Der zeitliche Signalverlauf beim Handshaking ist in Abb.3 schematisch dargestellt.

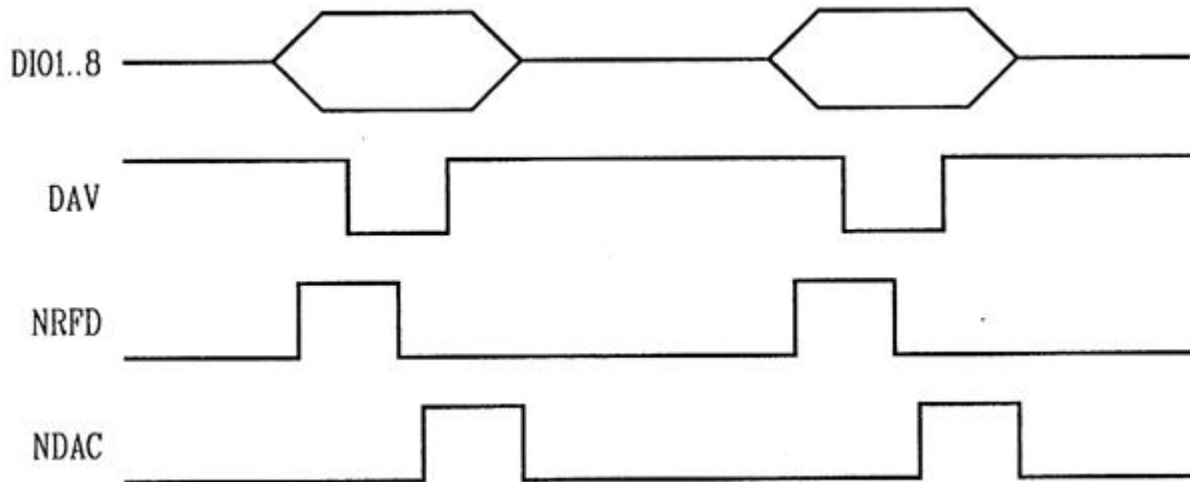


Abb.3 Signalverlauf/Handshaking bei einer Datenwortübertragung (schematisch)

Interfacemanagementleitungen (5)

werden dazu benutzt, um den Informationsfluß über das Interface zu leiten.

- **ATN** (attention / Achtung) wird durch den Controller auf "wahr" gesetzt, wenn er die Datenleitung benutzt, um Kommandos zu senden, und auf "falsch", wenn er einem Talker erlaubt, Datennachrichten zu senden.
- **IFC** (interface clear / Interface frei machen) wird durch den System-Controller verwendet, um den Bus zu initialisieren.
- **REN** (remote enable / Fernsteuerung erlauben) ist durch den System-Controller belegt und setzt Geräte in den Fernsteuer- oder lokalen Programmiermodus.
- **SRQ** (service request / Bedienungsanfrage) kann durch jedes Gerät am GPIB verwendet werden, um asynchron eine Bedienung vom Controller anzufordern.
- **EOI** (end or identify / Ende oder Identifizierung) hat zwei Aufgaben: ein Talker markiert damit das Ende einer Nachrichten- Zeichenkette. Der Controller teilt Geräten mit, ihre Antwort in einer parallelen Umfrage auszuweisen.

3.2.3 Nachrichtenarten

Der GPIB überträgt geräteabhängige Nachrichten und Interfacenachrichten:

- Die geräteabhängigen Nachrichten (Daten oder Datennachrichten) sind Anweisungen, Meßergebnisse, Gerätestatus- Meldungen und Datendateien.
- Die Interfacenachrichten steuern den Bus selbst und werden meist als Kommandos oder Kommandonachrichten bezeichnet. Sie dienen zur Initialisierung des Busses, Adressierung und Entadressierung von Geräten und zum Setzen des Gerätestatus.

3.2.4 Konfiguration und elektrische Spezifikationen

Die GPIB-Geräte sind gewöhnlich mit einem Kabelgefüge verbunden, das aus 24 abgeschirmten Leiterkabeln mit jeweils einem Stecker und einer Kupplung an jedem Ende besteht. Diese Bauart erlaubt es, Geräte in einer linearen oder einer Stern- Konfiguration oder einer Kombination von beiden zu verbinden.

Die Standard-Steckverbindung ist der Amphenol der Cinch Serie 57 MICRORIBBON oder AMP CHAMP-Type, dessen Belegung hier gezeigt ist:

IEEE 488 Interface

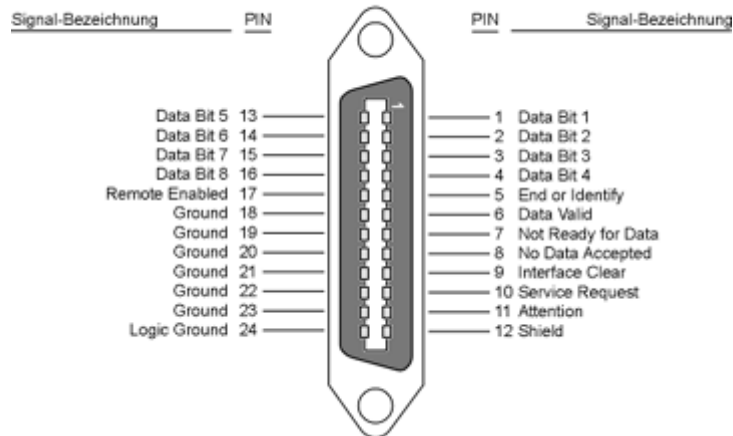


Abb.4 Steckverbindung des GPIB

Erforderlichenfalls kann ein Adapterkabel eingesetzt werden, um Fremdkabel und/oder Steckverbindungen für spezielle Verbindungsanwendungen zu benutzen.

Der GPIB benutzt negative Logik mit Standard-TTL-Pegeln. So ist z.B. beim Handshaking für DAV "wahr" ein TTL-Pegel von 0,8 V (LOW) gegeben, dagegen für DAV "falsch" ein TTL-Pegel von 2,0 V (HIGH) .

Zur Sicherung der hohen Datenübertragungsgeschwindigkeit des GPIB (1 MBit/s) ist der physikalische Abstand zwischen Geräten und die Anzahl der Geräte am Bus begrenzt:

- An einem einzelnen Bus können maximal 15 Geräte betrieben werden.
- Die gesamte Kabellänge darf maximal 20 m erreichen.
- Zwischen zwei Geräten kann höchstens 4 m Kabel liegen, im Normalfall 2 m Kabel.

Sofern eine Anwendung die Überschreitung dieser Grenzen erfordert, müssen entsprechende Buserweiterungen eingesetzt werden, die bei NI und anderen Herstellern erhältlich sind.

4. Aufgabenstellungen

Noch bevor Sie irgendwelche praktischen Tätigkeiten ausführen, lesen Sie bitte aufmerksam alle zur Verfügung stehenden Handbücher (zur GPIB-PC-Karte und zum GPIB-Meßgerät) und arbeiten Sie dabei besonders aufmerksam die relevanten Kapitel zum Hardware-Anschluß und zur Programmierung (entsprechend der von Ihnen gewählten Programmiersprache) durch.

- Untersuchen Sie die GPIB-PC-Karte und notieren Sie sich evtl. vorhandene Jumper- Einstellungen! Prüfen Sie die richtigen Einstellungen unter Beachtung der Angaben im Handbuch nach und nehmen Sie evtl. erforderliche Änderungen vor!
- Öffnen Sie den PC unter Beachtung der im vorigen Versuch gegebenen Hinweise und bauen Sie die GPIB-PC-Karte in einen freien Slot ein! Achten Sie dabei darauf, daß genügend Platz für den Steckverbinder am Kabel zur Verfügung steht! Prüfen Sie die korrekte Installation der Karte mit Hilfe der Test-Software auf der Diskette und nehmen Sie evtl. erforderliche Änderungen vor!
- Installieren Sie die mit der Karte mitgelieferte Software auf dem PC! Überprüfen Sie die Systemkonfiguration und testen Sie mit den zur Verfügung stehenden Mitteln die korrekte Funktionsweise der Karte und des Systems! Welche System- Ressourcen belegt die GPIB-Karte?
- Schließen Sie das Meßgerät mit einem Kabel an die GPIB-Karte an und prüfen Sie erneut mit dem Testprogramm! Finden Sie die Konfiguration für die Karte und das Meßgerät heraus und notieren Sie Ihre Feststellungen!
- Senden Sie ein Kommando zur Ermittlung des Status an das Meßgerät und prüfen Sie die Antwort darauf!
- Erstellen Sie unter Nutzung der in den Handbüchern enthaltenen Hinweise und Beispiele ein eigenes kleines Programm, bei dem in einer Schleife ein Kommando vom PC an das Meßgerät geschickt wird! Oszillografieren Sie die Datensignale und das Handshaking unter Beachtung der Belegungen in Abb. 4 und verifizieren Sie die Darstellung in Abb.3! Bestimmen Sie dabei auch den Zeitverlauf (Pulsängen, Frequenzen)!
- Speisen Sie ein Signal in das Meßgerät ein und versuchen Sie, den Meßwert mit einem eigenen kleinen Programm abzufragen!