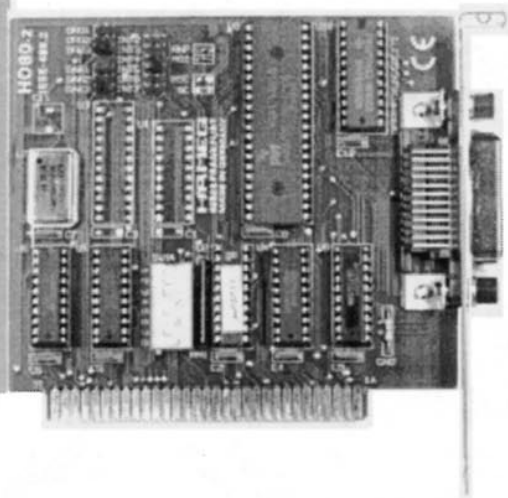


Manual HO80-2

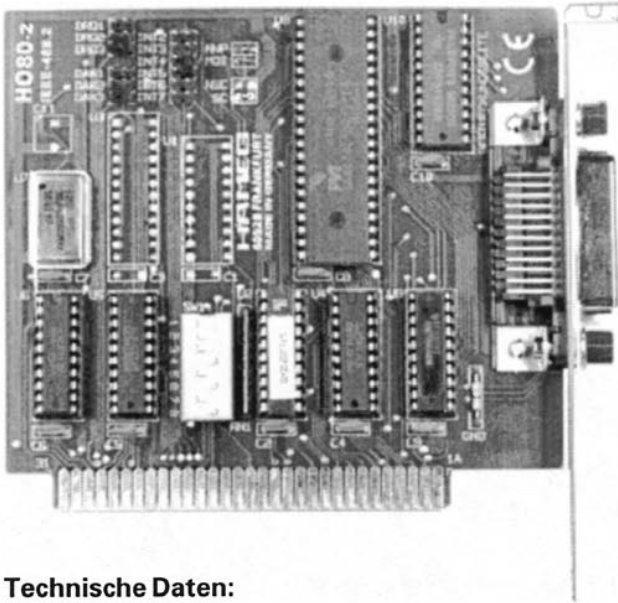
IEEE-488.2
PC-Interface Karte



IBM PC, PC/AT sind eingetragene Warenzeichen der International Business Machines Corporation
Microsoft ist ein eingetragenes Warenzeichen der Microsoft Corp.
Borland ist eingetragenes Warenzeichen der Borland Corp.
CEC PC-488 ist eingetragenes Warenzeichen der Capital Equipment Corp.
National Instruments PC-2A ist eingetragenes Warenzeichen der National Instruments
INES ist eingetragenes Warenzeichen der INES GmbH

Allgemeine Einführung	5
Struktur des IEEE-Interface-Systems	5
Talker, Listener und Controller	6
Geräte Adresse und Verbindungart	7
INSTALLATION	7
Hinweise zur Kartenkonfigurierung	9
Inhalt der Softwarediskette	10
Befehlsliste HO80-2 (mit geladener Firmware)	11
Zusammenfassung der Direktkommandos	12
Testprogramme	13
Portumleitung	14
Basic-Interpreter und Compiler	14
Quick-Basic & Basic-Compiler	14
Turbo-Pascal 4.0 bis 7.0	14
Programmbeschreibungen	14
Programmbeispiel	17
Statusmitteilungen	18
Demoprogramme	18
ANHANG	31
DIP-Schalter und Basis-Adresse	32
ASCII & IEEE (GPIB) CODE CHART	35

Allgemeiner Hinweis:
Änderungen in Hard- oder Software, welche nach Drucklegung dieses Handbuchs erfolgten, entnehmen Sie bitte den Liesmich oder Readme Files auf der Diskette.



HAMEG HO80-2 IEEE-488.2

Technische Daten:

Bus:	PC/AT 1/2 Slot Karte
Anschluß:	24-polige Buchse
Controller:	INES I-GPIB, kompatibel zu NECµPD7210
Systemadresse:	H2B8, H2E1, etc. mit DIP-Schalter wählbar
ROM:	Software ladbare Firmware ab HD000 - HE000
Lieferumfang:	Interface-Karte, Software und Handbuch.

Um Oszilloskope oder andere Meßgeräte mit **IEEE488-Bus** an einen Rechner anzuschließen, muß dieser mit einem entsprechenden Interface ausgerüstet sein. Für 100% kompatible PC's steht hierfür die preiswerte **HAMEG-Option HO80-2** zur Verfügung. Die in eine der PC-Slots einsetzbare Steckkarte gestattet den **Datenverkehr mit max. 15 angeschlossenen IEEE-488 fähigen Geräten** jeden Fabrikats.

Sämtliche Interrupt- und DMA-Möglichkeiten sind, unter Einbindung der im Handbuch beschriebenen Routinen bzw. auf der Diskette befindlichen Programme, nutzbar.

Die **HO80-2** Interface-Karte ist in der Lage, schnellen Daten-Blocktransfer (DMA) mit einer Geschwindigkeit - je nach Rechner-typ und Leitungslänge - bis ca. 25 KByte/Sek (PC 4.7 MHz Takt) zu übertragen. Der Datenblock darf max. 64 KByte lang sein.

Die auf der mitgelieferten Diskette befindliche Treiber-, Test- und Demo-Software (HAMEG-Geräte bezogen) unterstützen

QUICKBASIC, MS-BASIC Compiler, **TURBO PASCAL** und **MS-QUICK C**. Es wird das **Betriebssystem MS-DOS** ab Version 5.0 vorausgesetzt.

Mit dem **HO80-2** können sowohl Steuerbefehle wie auch Daten vom Rechner empfangen bzw. an diesen gesendet werden. Der maximal zu übertragende Datenblock darf dabei bis **64kByte** lang sein.

Im mitgelieferten Handbuch ist der vollständige Befehlssatz, mit dem das **HO80-2** ansprechbar ist, aufgelistet. Diese Syntax wird auch bei der Software SP91 verwendet. Das für den Anschluß von **IEEE-488** gesteuerten Geräten erforderliche Kabel gehört nicht zum Lieferumfang. Dieses ist unter der Bezeichnung **HZ72** auch von **HAMEG** erhältlich.

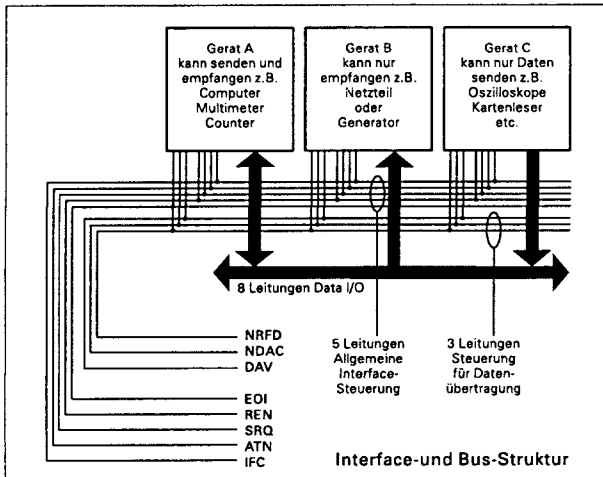
Die Hardware unterstützt den **IEEE-488.2** Standard und kann somit unter entsprechendem Softwareeinsatz Verwendung finden.

Technische Änderungen bei Hard- und Software vorbehalten.

Allgemeine Einführung

Im Jahre 1975 drangen die ersten Informationen über den IEC-Bus an die Öffentlichkeit. Heute hat sich dieses System auf dem Gebiet der Meßtechnik weltweit durchgesetzt. Ein nicht unwesentlicher Anteil der heute neu auf den Markt kommenden Geräte sind systemfähig mittels IEC-Bus.

Jeder Meßtechniker kommt früher oder später in die Lage, sich mit IEC-Bus gesteuerten Automaten auseinanderzusetzen. Mit der Verabschiedung der Publikation 625-1 "Standard Interface System for programmable measuring equipment" durch die IEC (Internationale Electrotechnical Commission) steht solch ein Interface-System zur Verfügung. Für dieses Bus-System haben sich im Laufe der Zeit leider verschiedene Namen eingebürgert. Neben der international genormten Bezeichnung IEC-625-Bus (kurz: IEC-Bus) ist auch der amerikanische Name IEEE488/75 üblich. Bis auf den Stecker sind beide Normen identisch. Die Stecker der amerikanischen Norm sind 24polig, während die internationale Norm einen 25poligen Stecker vorschreibt. Auf dem Markt sind jedoch Adapter dafür erhältlich. Entgegen der Empfehlung des Normenausschusses setzt sich jedoch der Stecker der amerikanischen Norm immer mehr durch. Die HAMEG HO80-2 Schnittstelle besitzt ebenfalls den 24poligen Stecker nach amerikanischer Norm. Die Bus-Elektronik ist jeweils im Gerät untergebracht. Der IEEE-Bus arbeitet mit TTL-Pegel in der "Negativ-True"-Logik (logisch 0 entspricht high Pegel). Die Ausgangstreiber haben ein "Fan-Out" von 30mA. Damit werden die "Thresh-Hold-Pegel" (Schwellwerte) auch bei voller Geräteanzahl erreicht. Das Impedanzverhalten der Verbindungskabel (spezifiziert in der Norm) unter Berücksichtigung der Signalzeiten läßt eine maximale Bus-Länge von 20m zu, mit der Einschränkung, daß pro Gerät eine Kabellänge von 2m nicht überschritten werden soll. Das bedeutet, daß alle 2m eine ohmsche Last (Wirklast) vorhanden sein muß. Die Übertragungsgeschwindigkeit von einem Mega-Byte/sec. wird erreicht bei einer Kabellänge pro Gerät von 0,5m und wenn 48mA Tri-State-Treiber verwendet werden. Die tatsächliche Maximalgeschwindigkeit hängt natürlich sehr stark von dem Zeitverhalten der angeschlossenen Geräte ab. Die höchsten Datenraten werden nur dann erzielt, wenn in den Geräten entsprechende Zwischenspeicher vorhanden sind. An eine Schnittstelle HO80-2 können bis zu 15 Geräte gleichzeitig angeschlossen werden. Die Anschlußfolge kann linear oder sternförmig erfolgen. Als Anhaltspunkt für die Übertragungsgeschwindigkeit von der Controllerseite kann angenommen werden, daß beim Einsatz eines PC-XT Rechners mit einer Übertragungsrate von ca. 5 bis 25 K-Byte/sec. zu rechnen ist.



Struktur des IEEE-Interface-Systems

Die Verbindung der einzelnen Geräte wird von einem Bus-Kabel mit 16 parallelen Leitungen vorgenommen. Über diese Leitungen werden Daten ausgetauscht, Steueranweisungen übermittelt und der Datenfluß überwacht (Handshaking). Die angeschlossenen Geräte werden nach ihren Funktionen in Controller, Talker und Listener eingeteilt.

Talker, Listener und Controller

Ein Gerät mit Talker-Funktionen kann Daten an andere Teilnehmer aussenden, z.B. ein Voltmeter oder Zähler. Geräte, die nur Daten empfangen können, werden als Listener bezeichnet (z.B. Stromversorgungen und Signalgeneratoren). Es gibt auch Geräte, die beide Funktionen beherrschen. Der Controller, ein Rechner, überwacht und steuert den Datenaustausch zwischen den Geräten. Das notwendige Computer-Interface ist die Hameg Schnittstelle HO80-2 für IBM PC/AT und kompatible MS DOS Computer.

Datenbus

Acht der 16 Bus-Leitungen bilden den Datenbus und werden zum Senden von Daten, Adressen, Programmbefehlen, Statusbyte und speziellen Buskommandos verwendet. Auf dem Datenbus unterscheidet man zwei Übertragungsarten: den Datenmode und den Befehlsmode. Die Kontrolleitung ATN signalisiert mit ihrem Zustand Low= Datenmode High= Adressmode. Adressen und Buskommandos werden immer vom Controller ausgesendet.

Kontroll-Bus

Jede der fünf Leitungen des Kontroll-Busses hat eine besondere Funktion:

ATN = Attention (Controller Befehl zur Deklaration)

- a) einer Adresse
- b) "unlisten"-Befehle
- c) "polling"-Befehle

IFC = Interface clear (Controller Befehl an alle Geräte)

Setzen aller angeschlossenen Geräte in einen definierten Zustand.

REN = Remote enable

Dient der Überwachung und Fernsteuerung von (remote control) der angeschlossenen Instrumente.

EOI = End or Identify

Kann als letztes Zeichen einer Gesamtinformation gesendet werden.

SRQ = Service request

Wird von irgend einem Gerät aktiviert. Wenn diese Leitung High ist, weiß der Controller, daß ein Gerät ihm etwas mitteilen möchte (seriell-polling). Der Controller kann dann das laufende Programm unterbrechen und die Nachricht des Gerätes abholen.

Der Handshake-Bus

Die Übertragung von Informationen auf dem Datenbus wird durch die Aktivitäten des Handshake-Bus begleitet. Es gibt drei Handshake-Leitungen:

DAV = Data Valid (Daten sind gültig)

Mit "low" wird vom "talker" signalisiert, daß die auf den Datenleitungen befindlichen Daten gültig sind

NRFD = Not ready for Data (nicht empfangsbereit)

Die "listener" geben über diese Leitung ihre Empfangsbereitschaft bekannt. Es gilt: High= ready (empfangsbereit), Low= **Not Ready For Data** (nicht empfangsbereit).

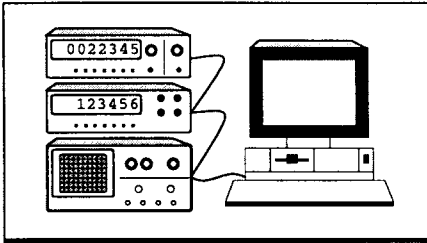
NDAC = Not data accepted (Information nicht aufgenommen)

Mit "high" quittiert ein "listener" die Übernahme der anliegenden Daten.

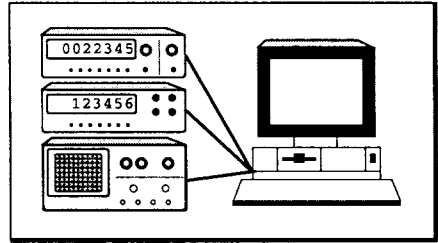
Geräte Adresse und Verbindungart

Alle an den IEEE-Bus angeschlossenen Geräte müssen unterschiedliche Geräteadressen aufweisen. In den Demoprogrammen verwenden wir für Oszilloskopie Adresse 6. Dem Computer (IEEE-Controller) wurde die Adresse 21 zugeordnet.

Verbindungsart Linear



Verbindungsart Stern



INSTALLATION

Vorbereitungen

Vor der Installation der Hardware sind unbedingt folgende Fragen zu klären

1. Welche I/O-Adressbereiche sind auf den Erweiterungssteckplätzen des Rechners verfügbar und werden von der verwendeten Software unterstützt.
2. Soll bzw. muß ein Interruptkanal benutzt werden? Falls ja, welcher ist verfügbar.
3. Soll ein DMA-Kanal benutzt werden? Falls ja, welcher ist verfügbar.
4. Ist das Interface der System-Controller oder nicht?

Soft- und Hardwarekompatibilität

Um die Softwarekompatibilität mit der HO80-1 Karte herzustellen wird der Inhalt des Eproms der HO80-1 Karte in den oberen Bereich des RAM geladen (Himem). Dies geht nur, wenn in der Mindestkonfiguration des Rechners 1MB RAM vorhanden ist. Durch die Programme HIMEM.SYS und EMS386.EXE mit der Option /NOEMS wird dieser Speicherbereich aktiviert. Diese Firmware ist nur dann erforderlich, wenn die Anwendungsprogramme die Unterstützung benötigen. Etliche Programme haben integrierte Softwareunterstützung für den NEC µPD7210. In diesem Falle wird das Laden der Software überflüssig.

Software: LOADHO80.EXE

Das Programm LOADHO80.EXE ist ein Werkzeug zur Installation der HO80-Firmware. Diese Routine untersucht vor dem Laden den Speicherbereich auf seinen Inhalt. Als sinnvolle Adressen haben sich Werte oberhalb DE00H bis EE00H erwiesen. Sollte das BIOS nicht geladen werden können, erfolgt eine Fehlermeldung "unable to load firmware". In diesem Falle muß überprüft werden, ob andere Programme, wie z.B. Smartdrive, SCSI-Bios, Virenüberwachung oder ähnliche Hi-Load Programme diesen Speicherbereich nutzen. Für eine einwandfreie Funktion ist es sinnvoll, zur Ermittlung eines freien Bereiches diese Programme abzuschalten und mit vorgegebener Segmentadresse einen Programmstart durchzuführen. Die eigentliche Programmlänge beträgt 200H Bytes. Das Programm läßt sich nicht in einen mit EMS386.EXE /X vorgegebenen freien Bereich laden.

Benötigte Vorkenntnisse

Personen, die diese Interfacekarte installieren, müssen über die Technik des benutzten Rechners informiert sein. Insbesondere sind die vom Rechner empfohlenen Sicherheitsmaßnahmen unbedingt zu berücksichtigen.

Installation der Karte.



(Beachten Sie die Anweisungen Ihres Rechnerherstellers zur Installation von Zusatzkarten, nur sie sind verbindlich).

1. Nehmen Sie alle Disketten heraus.
2. Schalten Sie den Netzschalter an der Rückseite der Systemeinheit aus.
3. Schalten Sie den Bildschirm aus.
4. Schalten Sie alle anderen mit dem System verbundenen Einheiten aus.
5. Ziehen Sie das Netzkabel aus der Systemeinheit heraus.
6. Ziehen Sie alle anderen Kabel und Verbindungen aus der Systemeinheit heraus.
7. Entfernen Sie das Gehäuse
8. Wählen Sie einen leeren Einsteckplatz unter folgenden Gesichtspunkten aus:
Alle Erweiterungsteckplätze sind in ihrer Funktion identisch; im allgemeinen kann jede Karte an jeden Platz gesteckt werden.
 - Wegen der Beschaffenheit des Rechners oder anderer Karten kann es manchmal unmöglich sein, die Karte an bestimmte Plätze zu stecken. Achten Sie deshalb immer darauf, daß physikalische Eigenschaften von Karte und Einsteckplatz zusammenpassen.
 - Wenden Sie keine Gewalt an! Wenn die Karte nicht leichtgängig einzusetzen ist, stoppen Sie den Versuch und suchen Sie die Ursache.
 - Bei jedem nicht verwendeten Steckplatz muß die Rückenabdeckung eingesetzt sein. Andernfalls ist das ordnungsgemäße Funktionieren der Luftkühlung nicht mehr gewährleistet.
9. Zum Einbau der Karte entfernen Sie zuerst die Rückenabdeckung des ausgewählten Steckplatzes. Werfen Sie die Abdeckung nicht weg. Sie benötigen sie wieder, wenn die Karte wieder einmal herausgenommen werden soll. Entfernen Sie ggf. die Rückenabdeckungen der angrenzenden Steckplätze. Führen Sie die Karte senkrecht von oben in den Einsteckplatz ein. Befestigen Sie die Karte an der Gehäuserückwand mit einer Schraube.
10. Vergewissern Sie sich jetzt, daß die Karte korrekt eingesetzt ist, die Schalter und Steckbrücken ordnungsgemäß gesetzt sind und alle Zusatzkarten fest in ihren Steckplätzen sitzen.
11. Schließen Sie das Gehäuse
12. Schließen Sie alle Kabel an.
13. Schalten Sie den Rechner ein und lassen Sie ihn seinen Selbsttest durchführen. Sollte der Test einen Fehler aufdecken, so überprüfen Sie den Einbau nochmals und achten dabei darauf, daß alles ordnungsgemäß eingesetzt wurde und alle Verbindungen korrekt hergestellt worden sind.
14. Wenn der Selbsttest einwandfrei durchgeführt wurde, fahren Sie mit der Installation der Software fort.

Hinweise zur Kartenkonfigurierung

I/O - Adreßauswahl

Die Interface-Karte belegt acht Adressen im I/O-Adreßbereich - des Rechners (Beispiel: 2E0H bis 2E7H). Diese Adressen dürfen nicht mit denen anderer Baugruppen kollidieren! Drei Adreßleitungen dienen zur Selektion der Register und werden nicht decodiert. Die Basisadresse wird durch den achtpoligen DIP-Schalter auf der Karte eingestellt. Üblicherweise werden die Adressen im I/O-Bereich des Rechners in hexadezimaler, Schreibweise angegeben (gekennzeichnet durch ein 'H' am Ende der Ziffernfolge). Die Basisadresse der Karte im Auslieferungszustand ist 2B8H. Daraus resultiert folgende Einstellung der DIP-Schalter SW1-8:

SCHALTER:	1	2	3	4	5	6	7	8
ON:	0	0	0	■	0	■	0	■
OFF:	■	■	■	0	■	0	■	0

Nachdem eine geeignete Basisadresse gewählt wurde (vgl. Anhang), werden die Schalter gemäß der Tabelle „Einstellung SW 1-8“ gesetzt.

Interruptkanal

Die Auswahl des Interruptkanals geschieht durch Setzen einer Brücke (waagrechtes Verbinden zweier entsprechender Pfsien) an der Stiftleiste INT,

Sie ist wie folgt organisiert:

INT = Interruptkanal

INT2 o o
INT3 o o
INT4 o o
INT5 o o
INT6 o o
INT7 o ——— o

INT7 = Interruptkanal 7 gesetzt

Soll kein Interruptkanal benutzt werden, muß die Steckbrücke entfernt werden!

Bei Lieferung ist kein Interrupt gesetzt.

DMA - Kanal

Die DMA-Funktion des Rechners ermöglicht den direkten Transfer von Daten zwischen Peripherie und Systemspeicher ohne den Eingriff der CPU. Diese Möglichkeit ist insbesondere bei der Verarbeitung großer Datenmengen sinnvoll. Die Aktivierung der DMA-Funktion erfolgt durch das Setzen von zwei Steckbrücken an den Stiftleisten DRQ und DAK.

Diese sind wie folgt organisiert:

DRQ1 o o
DRQ2 o o
DRQ3 o ——— o

DRQ = DMA REQUEST (DMA-ANFORDERUNG)

DAK1 o o
DAK2 o o
DAK3 o ——— o

DAK = DMA ACKNOWLEDGE(DMA-BESTÄTIGUNG)

über die üblicherweise belegten DMA-Kanäle informiert Sie Ihr Computerhandbuch. Sollen die DMA-Möglichkeiten nicht benutzt werden, müssen die Steckbrücken entfernt werden.

Bei Lieferung ist DMA3 und DAK3 gesetzt

SYSTEM-CONTROLLER

Sie können Ihre Interface-Karte als System-Controller oder IEEE-BUS-Device betreiben. Beachten Sie, daß es nur einen (!) System-Controller in einem IEEE-BUS-System geben darf.

In den meisten Fällen wird IEEE488 der System-Controller sein. Die eingelötete Brücke "SC" ist auf diesen Fall fest voreingestellt:

NSC o o
SC o———o Computer ist **System Controller**

Möchten Sie IEEE488 als Device betreiben, so konfigurieren Sie die 'SC'Brücke wie folgt:

NSC o———o
SC o o **kein System Controller (Device)**

WARNUNGEN!

Wenn Sie die Interfacekarte als IEEE-488-BUS-DEVICE SC passiv) konfiguriert haben, dann dürfen Sie nie seitens des BUSCONTROLLERS NEC 7210 das Kommando request system control auslösen! Wird dies dennoch getan, dann kann dies zur Zerstörung von Teilen der Interfacekarte führen.



Die Selektion falscher Basisadressen, Interrupt- oder DMAKanäle kann unter ungünstigen Umständen zur Zerstörung von Teilen Ihres Rechners führen!

Lassen Sie die nötige Sorgfalt walten!!
Wie zeigen sich Fehler?

Falsche Basisadresse:

Interface funktioniert mit Sicherheit nicht. Beim Überlappen von Adressen zeigen sich noch andere Fehlfunktionen, beispielsweise läuft das gesamte System nicht an.

Falscher Interruptkanal:

Systemzeit zählt nicht hoch, Datenübertragungen von/zur Tastatur, Peripheriegeräten oder Diskette/Platte funktionieren nicht.

Falscher DMA-Kanal:

Datenübertragungen Diskette/Platte funktionieren nicht.
SCHALTEN SIE DAS GERÄT AUS UND SUCHEN SIE DIE URSACHE.

ACHTUNG!

Stellen Sie sicher, daß die einwandfreie Einstellung der Basisadresse, des Interruptkanals und des DMA-Kanals erfolgt ist (siehe vorhergehenden Abschnitt)!

Inhalt der Softwarediskette

Basic Konfigurations und Demoprogramme

Konfigurations Programme LOAD8000-LOADDE00 in **Sub-DIR BASICDRV**

Basic-Programme in **Sub-DIR BASIC**

Initialisierung, Senden und Empfangen von Daten	ISE.BAS
Service Request und Seriell Poll	SRQ.BAS
Sekundäradresse und Datenblock lesen	IRDA.BAS

Datenblock schreiben

IWDA.BAS

Turbo-Pascal Programme in Sub-DIR PASCAL

HO80_1.DOC, HO80_1.P40, bis HO80_1.P70 und READ.ME

IEEEPAS1.PAS

IEEEPAS2.PAS

C-Programme in Sub-DIR C

CL-, CM- und CSCOMMON.ASM

CL-, CM- und CSCOMMON.OBJ

DEMO208.C und DEMO208.EXE, HM8112.C und HM8112.EXE für MS-C

HM8112.C und HM8112.EXE für Turbo-C

Bezeichnerliste MC.H für MS-C und Turbo-C

Object-Codes in Sub-DIR CVMASM für Microsoft MACRO-ASSEMBLER

Object-Codes in Sub-DIR CXTASM für Borland Turbo-Assembler

Befehlsliste HO80-2 (mit geladener Firmware)

		<i>Offset der Adressen Interpreter / Compiler</i>	
Basic Befehl:			
Call INIT	(adress%,pegel%) Initialisierung adress% = Adresse vom PC, z.B. 21 pegel% = 0 System Controller 1 Aktiver Controller 2 Gerät	0	0
Call IWSD	(komando\$,status%) Sende sekundär Adresse komando\$ = Befehls- oder Datenwort Beispiel: "MLA TALK 6 SEC 15"	3	30
Call IRSD	(e\$,lang%,status%) lese sekundär adresse e\$ = Datensatzeingang lang% = Datensatzlänge status% = gibt den Status für alle Routinen 0 = Datentransfer ohne Fehler 8 = Zeitüberlauf 32 = Transfer ende mit EOI	6	33
Call IWD	(instrument%,befehl\$,status%) Sende Datenwort instrument% = Geräte Adresse befehl\$ = Befehlsstring	9	36
Call IRD	(e\$,lang%,instrument%,status%) lese Datenwort e\$ = empfangener Datensatz lang% = Datensatzlänge instrument% = Geräte Adresse	21	39
Call ISPL	(instrument%,pol%,status%) instrument% = Geräteadresse für Poll pol% = Status - Byte Poll	12	12

Basic Befehl:

Call IPPL (pol%)	15	15
pol% = Status - Byte Poll		
Call IWDA (segment%,offset%,anzahl%,eoi%,status%)	200	200
Sende Datenarray		
segment% = Segment Adresse der Daten		
offset% = Offsetadresse der Daten		
anzahl% = Anzahl der zu übertragenen Byts		
eoi% = 1 oder 0 für mit und ohne Endzeichen		
Call IRDA (segment%,offset%,anzahl%,lang%,status%)	203	203
lese Datenarray		
segment% = Segment Adresse der Daten		
offset% = Offsetadresse der Daten		
anzahl% = Array gröÙe		
lang% = gelesene Datensatzlänge		

Zusammenfassung der Direktkommandos

Die Direktkommandos des IEEE-Bus teilen sich in vier Gruppen auf. Sie unterscheiden sich in: Universalbefehle, Mehrdrahtnachrichten, adressierte Befehle und Sekundärbefehle.

Universalbefehle

- IFC = Interface Clear / Schnittstellenfunktion rücksetzen.
- REN = Remote Enable / Fernsteuerung freigeben.
- ATN = Attention / Achtung.
- EOI = End of Output/identify / Erkennung des Datenendes
- END = EOI / mit zusätzlichem LF CHR\$(13).

Mehrdrahtnachrichten

- DCL = Device Clear / Gerät rücksetzen.
- LLO = Local Lockout / Steuerung verriegeln.
- PPU = Parallel Poll / Parallelabfrage abschalten.
- SPE = Seriell poll enable / Serienabfrage freigeben.
- SPD = Seriell poll disable / Serienabfrage sperren.
- UNL = Unlisten / Listener wird inaktiviert.
- UNT = Untalk / Talker wird inaktiviert.

Adressierte Befehle

- GET = Group Execute Trigger / Gerätegruppe auslösen.
- SDC = Selected Device Clear / angewähltes Gerät rücksetzen.
- GTL = Goto Local / Auf Eigensteuerung schalten.
- PPC = Parallel Poll Configure / zur Parallelabfrage konfigurieren.
- TCT = Take Control / Steuerung übernehmen.

- DATA = Senden einer Dateninformation.
- LISTEN = Definiert ein Listener (Hörer) oder Listener mit Adresse.
- TALK = Definiert ein Talker (Sprecher) mit Adresse.

Sekundärbefehle

PPE = Parallel Poll Enable / *Parallelabfrage freigeben*
PPD = Parallel Poll Disable / *Parallelabfrage sperren.*
SEC = Sekundäradressen-Übergabe / *Als Erweiterung reiner Listener*
oder Talker Geräte wie z.B. HAMEG Oscilloscope HM305

PC-Befehle

PC Befehle sind für die Einleitung besondere Betriebsarten erforderlich. Sie leiten den Datentransfer zwischen zwei PC's ein oder das Senden von SEC-Befehlen

MLA = MY.Listen Adresse. PC wird Listener

MTA = MY.Talk Adresse. PC wird Talker

Testprogramme

LOOKHO80.EXE

Schnittstellentestprogramm zur Initialisierungsprüfung:

Als Antwort erhalten Sie bei einwandfreier Funktion **** Ver. 1.1 **** sowie die auf der Schnittstellenkarte eingestellte Adresse des Eproms. Wird die Karte nicht gefunden, so wird eine Fehlermeldung ausgegeben.

TESTHO80.EXE

Senden von einzelnen Befehlen und Empfang von Daten in Einzelschritten mit dem Programm:

Nach dem Programmstart können Sie unter 3 verschiedenen Landessprachen wählen. Durch Eingabe des dem Landeskenner zugeordneten Buchstabens, wird das Programm in der Landessprache gestartet. Zur Ermittlung der Kartenadresse läuft zuerst ein Selbsttest. Wird die HO80 Karte gefunden, wird dies auf dem Bildschirm mit zugehöriger Adresse angezeigt. Sollte auf der Rechnerseite ein Defekt vorliegen, wird eine Fehlermeldung ausgegeben. Defekte im Bereich des IEEE-Buses werden nicht erfasst. Für die Funktion SRQ und seriell POLL ist die Adresse des INES I-GPIB Controllers auf den Wert &H2B8 fest eingestellt. Der Computer belegt die Adresse 21. Nach dem Selbsttest wird ein Input verlangt. Wird an dieser Stelle ein Fragezeichen '?' eingegeben, so werden die Datentransfer-Modi angezeigt.

Folgende Eingaben bewirken den Aufruf des Programmteils:

'S' für sende Befehl zum Gerät
'E' für empfangen Daten vom Gerät
'T' für sende Direkt- und SEC- Kommandos
'O' lese Datenarray vom Oscilloscope
'R' für empfangen Direkt- und SEC- Kommandos
'P' für SRQ und seriell Poll
'Q' für Ende

Die zu übertragenden Datensatzlängen für Senden und Empfang sind auf 80 Zeichen eingestellt. Eine Ausnahme macht dabei das Programmteil 'O'. Es wurde speziell auf die Bedürfnisse der Hameg Oscilloscope HM 205-2 und HM 208 eingestellt. Hier kann ein Datenblock bis zu einer Gesamtlänge von 2048 Bytes übertragen und als Zahlenwerte angezeigt werden. Für den Datentransfer mit SEC (Sekundäradressen) gilt folgendes Beispiel:

MLA = Computer sendet SEC-Kommando zum Einlesen

Eingabe SEC Kommando **MLA TALK 6 SEC 10**

TALK 6 = an Geräteadresse 6

SEC 10 = Sekundärkommando 10 (Oscilloscope-Daten als Daten Block auslesen).

MTA = Computer sendet SEC-Kommando zum Ausführen einer Funktion.

Eingabe SEC Kommando **MTA TALK 6 SEC 4**

SEC 4 = Sekundärkommando 4 (Oscilloscope RESET setzen).

Portumleitung

Anwendung der HO80-2 Schnittstelle mit einem IEEE Drucker oder Plotter. Möchten Sie einen Drucker oder Plotter mit IEEE-Bus an Ihrer Systemeinheit installieren, so ist dies möglich. Mit dem Programm PRN_IEEE.EXE, welches sich auf der Diskette befindet, können Sie den Printer Port LPT1: auf den IEEE-Bus umschalten. Verwenden Sie folgende Syntax:
Für ein Gerät mit Listener Adresse 5 starten Sie

PRN_IEEE 5

alle nun zum Port PRN: oder LPT1: gesendeten Daten werden von PRN: über das HO80-2 IEEE Interface zur IEEE-Adresse 5 umgeleitet.

Basic-Interpreter und Compiler

Konfigurierung der Schnittstelle bei BASIC-Programmierung mit dem Programm **LOADHO80.EXE**

LOADHO80.EXE SEG=EE00

Initialisiert die Schnittstellenkarte mit der Adresse &HEE00, (Basic Schreibweise)

Achten Sie darauf, daß nach dem Start des Programms die richtige Adresse ausgegeben wird.

Quick-Basic & Basic-Compiler

Die HO80-2 IEEE Karte kann unter Quick-Basic und Basic-Compiler betrieben werden. Hierfür sind alle Anwendungsprogramme nur in 2 Punkten umzustellen.

1. Die Offsetadressen müssen von Interpreter- in Compiler-Werte geändert werden.
2. Der Befehl steht nicht mehr hinter dem CALL Aufruf, sondern ist Teil des Kommandostrings.

zum Beispiel:

Interpreter Basic

10 IWD = 9

20CALL

IWD(instrument%,befehl\$,status%)

Compiler

IWD% = 36

CALL **ABSOLUTE** (instrument%,befehl\$,status%,**IWD%**)

Für den Einsatz mit dem Quick-Basic Compiler 2.0 befindet sich ein USERLIB.EXE Files auf der Diskette in der Sub-Directory QBASIC2. Dieses von uns erstellte USERLIB.EXE Programm beinhaltet den ABSOLUTE Befehl. für die QBASIC Versionen 2.0 und 2.01. Verwenden Sie Quick-Basic 4.0 oder 4.5 so gelten einige Einschränkungen. Der Befehl Call Absolute in Verbindung mit Sekundäradressen (IWS%) wird nicht von der QB4.xx-Benutzeroberfläche ausgeführt. Erst nach der Generierung eines EXE-Files ist das Programm lauffähig.

Turbo-Pascal 4.0 bis 7.0

Für die Anwendung der Hameg HO80 IEEE-Schnittstellenkarte befindet sich ein Assembler Programm mit dem Namen **HO80_1.TPU** auf der Softwarediskette. Dieses in Turbo Pascal einzubindende Programm stellt die automatische Erkennung und Initialisierung, sowie die gesamten Befehle der HO80-2 Schnittstelle zur Verfügung. Programmierhinweise finden Sie in den mitgelieferten Demoprogrammen IEEEPAS1 und IEEEPAS 2.

Sekundärbefehle

PPE = Parallel Poll Enable / *Parallelabfrage freigeben*
PPD = Parallel Poll Disable / *Parallelabfrage sperren.*
SEC = Sekundäradressen-Übergabe / *Als Erweiterung reiner Listener oder Talker Geräte wie z.B. HAMEG Oscilloscope HM305*

PC-Befehle

PC Befehle sind für die Einleitung besondere Betriebsarten erforderlich. Sie leiten den Datentransfer zwischen zwei PC's ein oder das Senden von SEC-Befehlen
MLA = MY.Listen Adresse. PC wird Listener
MTA = MY.Talk Adresse. PC wird Talker

Testprogramme

LOOKHO80.EXE

Schnittstellentestprogramm zur Initialisierungsprüfung:
Als Antwort erhalten Sie bei einwandfreier Funktion **** Ver. 1.1 **** sowie die auf der Schnittstellenkarte eingestellte Adresse des Eproms. Wird die Karte nicht gefunden, so wird eine Fehlermeldung ausgegeben.

TESTHO80.EXE

Senden von einzelnen Befehlen und Empfang von Daten in Einzelschritten mit dem Programm:

Nach dem Programmstart können Sie unter 3 verschiedenen Landessprachen wählen. Durch Eingabe des dem Landeskenner zugeordneten Buchstabens, wird das Programm in der Landessprache gestartet. Zur Ermittlung der Kartenadresse läuft zuerst ein Selbsttest. Wird die HO80 Karte gefunden, wird dies auf dem Bildschirm mit zugehöriger Adresse angezeigt. Sollte auf der Rechnerseite ein Defekt vorliegen, wird eine Fehlermeldung ausgegeben. Defekte im Bereich des IEEE-Buses werden nicht erfasst. Für die Funktion SRQ und seriell POLL ist die Adresse des INES I-GPIB Controllers auf den Wert &H2B8 fest eingestellt. Der Computer belegt die Adresse 21. Nach dem Selbsttest wird ein Input verlangt. Wird an dieser Stelle ein Fragezeichen '?' eingegeben, so werden die Datentransfer-Modi angezeigt.

Folgende Eingaben bewirken den Aufruf des Programmteils:

- 'S' für sende Befehl zum Gerät
- 'E' für empfangen Daten vom Gerät
- 'T' für sende Direkt- und SEC- Kommandos
- 'O' lese Datenarray vom Oscilloscope
- 'R' für empfangen Direkt- und SEC- Kommandos
- 'P' für SRQ und seriell Poll
- 'Q' für Ende

Die zu übertragenden Datensatzlängen für Senden und Empfang sind auf 80 Zeichen eingestellt. Eine Ausnahme macht dabei das Programmteil 'O'. Es wurde speziell auf die Bedürfnisse der Hameg Oscilloscope HM 205-2 und HM 208 eingestellt. Hier kann ein Datenblock bis zu einer Gesamtlänge von 2048 Bytes übertragen und als Zahlenwerte angezeigt werden. Für den Datentransfer mit SEC (Sekundäradressen) gilt folgendes Beispiel:

MLA = Computer sendet SEC-Kommando zum Einlesen

Eingabe SEC Kommando **MLA TALK 6 SEC 10**

TALK 6 = an Geräteadresse 6

SEC 10 = Sekundärkommando 10 (Oscilloscope-Daten als Daten Block auslesen).

MTA = Computer sendet SEC-Kommando zum Ausführen einer Funktion.

Eingabe SEC Kommando **MTA TALK 6 SEC 4**

SEC 4 = Sekundärkommando 4 (Oscilloscope RESET setzen).

Portumleitung

Anwendung der HO80-2 Schnittstelle mit einem IEEE Drucker oder Plotter.
Möchten Sie einen Drucker oder Plotter mit IEEE-Bus an Ihrer Systemeinheit installieren, so ist dies möglich. Mit dem Programm PRN_IEEE.EXE, welches sich auf der Diskette befindet, können Sie den Printer Port LPT1: auf den IEEE-Bus umschalten. Verwenden Sie folgende Syntax:
Für ein Gerät mit Listener Adresse 5 starten Sie

PRN_IEEE 5

alle nun zum Port PRN: oder LPT1: gesendeten Daten werden von PRN: über das HO80-2 IEEE Interface zur IEEE-Adresse 5 umgeleitet.

Basic-Interpreter und Compiler

Konfigurierung der Schnittstelle bei BASIC-Programmierung mit dem Programm **LOADHO80.EXE**
LOADHO80.EXE SEG=EE00
Initialisiert die Schnittstellenkarte mit der Adresse &HEE00, (Basic schreibweise)
Achten Sie darauf, daß nach dem Start des Programms die richtige Adresse ausgegeben wird.

Quick-Basic & Basic-Compiler

Die HO80-2 IEEE Karte kann unter Quick-Basic und Basic-Compiler betrieben werden. Hierfür sind alle Anwendungsprogramme nur in 2 Punkten umzustellen.

1. Die Offsetadressen müssen von Interpreter- in Compiler-Werte geändert werden.
2. Der Befehl steht nicht mehr hinter dem CALL Aufruf, sondern ist Teil des Kommandostrings.

zum Beispiel:

Interpreter Basic

```
10 IWD = 9  
20 CALL  
IWD(instrument%,befehl$,status%)
```

Compiler

```
IWD% = 36  
CALL ABSOLUTE (instrument%,befehl$,status%,IWD%)
```

Für den Einsatz mit dem Quick-Basic Compiler 2.0 befindet sich ein USERLIB.EXE Files auf der Diskette in der Sub-Directory QBASIC2. Dieses von uns erstellte USERLIB.EXE Programm beinhaltet den ABSOLUTE Befehl, für die QBASIC Versionen 2.0 und 2.01. Verwenden Sie Quick-Basic 4.0 oder 4.5 so gelten einige Einschränkungen. Der Befehl Call Absolute in Verbindung mit Sekundäradressen (IWSW%) wird nicht von der QB4.xx-Benutzeroberfläche ausgeführt. Erst nach der Generierung eines EXE-Files ist das Programm lauffähig.

Turbo-Pascal 4.0 bis 7.0

Für die Anwendung der Hameg HO80 IEEE-Schnittstellenkarte befindet sich ein Assembler Programm mit dem Namen **HO80_1.TPU** auf der Softwarediskette. Dieses in Turbo Pascal einzubindende Programm stellt die automatische Erkennung und Initialisierung, sowie die gesamten Befehle der HO80-2 Schnittstelle zur Verfügung. Programmierhinweise finden Sie in den mitgelieferten Demoprogrammen IEEEPAS1 und IEEEPAS 2.

Erstellung eines TPU-Files

Für die Versionen Turbo-Pascal 4.0 bis 7.0 sind fünf Programme auf der Softwarediskette mit den Namen HO80_1.P40, bis HO80_1.P70. Sie müssen durch COPY das Programmfile HO80_1.TPU erzeugen. Verwenden Sie hierzu den COPY Befehl:
z.B. für Turbo-Pascal 4.0

COPY HO80_1.P70 HO80_1.TPU

zur Erstellung des .TPU Files.

Softwaretools für MS-C und Turbo-C

Zur Unterstützung vom Microsoft C, Quick C Version 1.01B und Borland Turbo C sind auf der Softwarediskette im Unterverzeichnis C folgende Programme enthalten:

CSCOMMON.ASM,
CSCOMMON.OBJ,
MC.H. DEMO208.C,
HM8112.C und
HM8122.C als Source-Code,
*.OBJ Datei und
*.EXE File.

In den Unterverzeichnissen MASM und TASM befinden sich die Assembler-Codes entsprechend MASM=Microsoft- oder TASM=Turbo-Assembler. Das Programm CSCOMMON.ASM im Sourcetext dient fortgeschrittenen C-Programmierern als Referenzmöglichkeit zur Lösung eigener Problemstellungen. Es enthält die Assembler - Routinen für Microsoft QuickC 1.01B, zur Unterstützung der HAMEG Interface-Karte HO80. Bei Verwendung der Bezeichnerliste, im Programm MC.H enthalten, genügt es, das vorhandene CSCOMMON.OBJ File mit einzubinden. Das hier verwendete Speichermodell ist **small**. Zum Kompilieren und Linken der Module können Sie folgende Befehlszeile eingeben :

; qcl /AS /F 1000 DEMO208.C CSCOMMON.OBJ

Gleiches gilt für die Speichermodelle Medium und Large. Sie werden von den Modulen "CMCOMMON.OBJ" bzw. "CLCOMMON.OBJ" unterstützt. Die Option für "QCL" ist /AM für "Medium" und /AL für "Large".

Das Programm DEMO208.C ist ein Demonstrationsprogramm für die Steuerbefehle des HAMEG Oscilloscope HM 208 und HM205-2. Zur Einbindung ist dieses Programm gleichzeitig auch als .OBJ File auf der Diskette vorhanden.

Programmbeschreibungen

INITIALISIERUNG

In jedem BASIC-Programm muß stets ein besonderer Syntax eingehalten werden. Dieser besteht aus der Speicheradresse, dem Unterroutinenoffset und der Angabe über die Controllerpriorität. Die Speichersegmentadresse ist der Wert, welcher als Antwort nach erfolgreichem Laden der Firmware ausgegeben wird. Der Unterroutinenoffset, ein Wert zwischen 1 und 30, im Normalfall 21, teilt der Schnittstelle eine Adresse zur Initialisierung zu. Die Controller-Priorität wird auf verschiedene Pegelwerte eingestellt. Pegel% = 0: HO80 Interface ist Systemcontroller. Pegel% = 1: HO80 Interface ist ein Aktiver Controller und Pegel% = 2: HO80 Interface ist ein Gerät Lötbrücke auf "NSC". Mit diesen Werten wird bei einem Multicontrollerbetrieb festge-

legt welcher Rechner der Systemcontroller ist. Ein Controller kann auch die Funktion eines Gerätes annehmen z.B. zur Meßdatenablage, oder als Subcontroller für bestimmte Aufgaben bei der Meßdatenerfassung als aktiver Controller. Nach dem Befehl CALL INIT mit Pegel% = 0 sendet die HO80-2 Schnittstelle ein (IFC) Interface Clear zur Initialisierung aller am IEEE-BUS befindlichen Geräte. Für das Arbeiten mit BASIC-Interpreter ist unbedingt die Firmware ins RAM zu laden.

SENDEN und EMPFANGEN

Mit den Befehlen IWD und IRD wird der Datentransfer vorgenommen. Diesen Variablen wird ein Unterroutinen-Offset zugewiesen als direkte Adressen, wobei der Syntax folgendes bedeutet: IWD = Interface Write Data (SENDEN) bzw. IRD = Interface Read Data (EMPFANGEN). (Beispiel Seite 18).

```
210 IWSD = 3: IRSD = 6
220 '
230 ADRESS% = 21                'PC Controller Adresse 21
240 PEGEL% = 0                  'Systemcontroller
250 '
260 INSTRUMENT% = 6
```

SERVICE REQUEST mit SERIELL POLL

Für die Abfrage des Service Request gibt es keinen direkten Interruptbefehl. Das Überwachen der Funktion SRQ muß mittels Direktabfrage der Adressleitung am IC μ PD7210C erfolgen. Das Programmbeispiel zeigt die SRQ-Abfrage mit zwei Geräten HM8112 mit unterschiedlich eingestellter Integrationszeit und seriell Poll-Auswertung. Nach der Initialisierung und den Geräte-Befehlen wird für beide Geräte seriell Poll aktiviert. Danach wartet das Programm in Zeile 420, bis sich eines der Geräte mit SRQ meldet. In den Zeilen 450-540 wird nun die Auswertung des seriell Poll vorgenommen, um das entsprechende Gerät zu ermitteln, welches sich mit SRQ gemeldet hat. Nach dem Abfragebefehl ISPL (Interface seriell Poll) wird automatisch der Zustand zurückgesetzt, so daß eine erneute Abfrage möglich wird. Über die Statusvariablen in den Zeilen 580 und 640 wird selektiert und in der nachfolgenden Routine dann die Geräte bedient. Haben beide Geräte gleichzeitig SRQ gesendet, so werden die Geräte nacheinander bedient. Diese Routine ist geräteunabhängig und kann in ihrer Struktur auch auf andere Anwendungen übertragen werden. Hierbei ist zu beachten, daß die eingestellte Controlleradresse des IC μ PD7210 (bei Lieferung eingestellt auf &H2B8) im Programm die gleiche Wertigkeit besitzt. (Beispiel Seite 31).

SEKUNDÄRADRESSE und direkt Kommandos

In diesem Programmteil zeigen wir Ihnen, wie ein nur als TALKER spezifiziertes Gerät mittels der SEKUNDÄR - Adresse in der Lage ist, verschiedene Modi anzunehmen. Als Beispiel wurde hierfür das Oscilloscope HM1007 ausgewählt. (Beispiel Seite 20).

Initialisierung

Nach der Initialisierung Zeile 590-610 wird ein SEC Befehl mit CALL IWSD (Interface-Write-Sekundär-Data) an das angeschlossene Gerät in Zeile 780 gesendet. Nach erfolgter Datenübertragung, ist es empfehlenswert ein Reset an den IEEE-Bus mittels Sekundärkommando zu senden.

Befehlsstring Direktkommandos

In einem Befehlsstring wird dem Controller mitgeteilt (MLA), daß der PC zum Listener wird, daß der Befehl an das Gerät mit der Nr. 11 (TALK 11) gesendet werden soll, und daß das angesprochene Gerät sich in den Betriebszustand 10 zu setzen hat (SEC 10). Diese Syntax gilt für alle Direkt-Kommandos, welche auf Seite 8 aufgeführt sind.

DATENBLOCK LESEN (Array transfer)

In den Zeilen 420 bis 500 folgen die OFFSET und Variablenangaben für den Block-Transfer. Diese Übertragungsart ist die schnellste. Die Daten werden im BINÄR-Format übergeben. Es erfolgen nur am Anfang und am Ende der Übertragung die HAND-SHAKE Routinen. Einen 2-KByte großen Datenblock zu übertragen dauert in der Regel nicht länger als 0,2 Sekunden. Mittels eines PEEK Kommandos können dann die Werte aus dem Speicher direkt gelesen werden. Siehe hierzu das Programm ab Zeile 1210. (Beispiel Seite 21).

DATENBLOCK SCHREIBEN (Array transfer)

Das Senden eines Datenblocks im Binärformat gehört zu der schnellsten Übertragungsart. Sie erfordert genaue Kenntnisse über die zu übertragenen Daten. Da beim Speichern der Werte in einem Integer-Array die Werte im PC-Speicher als 2 mal 8 Bit Information abgelegt wird, können beim Übertragen Differenzen auftreten. Die Übertragung zum Beispiel des Wertes für "A" (65) hätte zur Folge, daß das erste Byte mit dem Wert 65 belegt ist. Das Zweite Byte jedoch 0 enthält. Nach der Übertragung eines Speicherbereiches hat nun jedes 2. Byte beim Empfänger den Wert 0. Es ist nun hier die Aufgabe des Programmierers, die Werte lückenlos in den Speicher zu schreiben zum Beispiel mit POKE.

Das Programmbeispiel

Senden von einem Datenblock zeigt nach der Initialisierung eine Sekundär Adressen Zuweisung MTA (Computer wird Talker 'Sprecher') und gibt für das Gerät Nr. 6 eine LISTEN (Listner 'Hörer') Anweisung. Gerätebezogen kann zusätzlich für das als Hörer angesprochene Gerät noch ein Direktbefehl notwendig sein, welcher mit CALL IWDA übertragen werden muß. Nach der Ermittlung der Offsetadressen wird der Datenblock mit dem Befehl CALL IWDA (Interface write Data Array) gesendet.

Programmierbeispiel

Arraydaten generieren für die Zahlenwerte 1 bis 255.
Gesamtgröße des Datenblocks 512 Byte.

```
100 DIM A(512), B%(256)
110 DEF SEG
120 FOR I = 1 TO 255      'Datenblock mit den Werten 1 und 255
130 A(I*2) = 1          'füllen
140 A(I*2+1) = 255
150 NEXT I
160 '
170 FOR I = 1 TO 512
180 IF INT(A(I*2+1)/128) = 1 THEN A% = 1 ELSE A% = 0
190 IF A% = 1 THEN B%(I) = ((A(I*2)-1 XOR 255) + ((A(I*2+1)
-128) XOR 127) * 256) * -1 : GOTO 210
200 B%(I) = A(I*2) + A(I*2+1) * 256
210 NEXT I
220 '
230 'Datenblock auf Bildschirm ausgeben
240 '
250 ADR = VARPTR(B%(1))
260 '
270 FOR AUSLESE = ADR TO ADR + 512
280 PRINT PEEK(AUSLESE);
290 NEXT
```

Direktprogrammierung des IEEE-Buscontrollers

Der Bus-Controllerbaustein uPD7210C kann mittels INP und OUT Befehlen direkt angesprochen werden. Die dazu notwendigen Register-Adressen können Sie der Produktbeschreibung des Herstellers entnehmen.

Statusmitteilungen

Die von der Schnittstelle ausgegebenen Statusmitteilungen können bei der Fehlersuche sehr nützlich sein. Die Variable STATUS kann entsprechend der Übertragungsart folgende Werte besitzen.

Mode	Wert	Syntax
Allgemein IWSD	0	Kein Fehler
	1	Falscher Befehl
	2	END Kommando in Sende oder Empfangsstring
	4	END Kommando in Sende oder Empfangsstring
	8	Zeitüberschreitung
IRSD	16	Reihenfolge Kommandostring fehlerhaft
	2	Empfangen ohne definierten Empfänger
IWD	8	Zeitüberschreitung
	8	Zeitüberschreitung
ISPL	8	Zeitüberschreitung
IRD	8	Zeitüberschreitung
IWDA	2	Senden ohne definierten Sender
	8	Zeitüberschreitung
IRDA	2	Empfangen ohne definierten Empfänger
	8	Zeitüberschreitung
	32	Datenübertragungsende EOI bei falscher ANZAHL%

Demoprogramm Init, S + E

```
100'-----
110'
120'Beispiel Senden und Empfang von Daten
130'Send = IWD und Empfang = IRD
140'
150'-----
160'
170 DEF SEG = &HDE00                'Speicheradresse
180 INIT = 0 : IWD = 9 : IRD = 21    'Unterroutinen Offset
190 '
200 '
210 ADRESS% = 21                    'HO-80 Adresse
220 PEGEL% = 0                       'Systemcontroller
230 '
240 '
250 INSTRUMENT% = 7                  'Geräteadresse
260 '
270 '
280 CALL INIT (ADRESS%,PEGEL%)      'Schnittstelle aktivieren
290 '
300 BEFEHL$ = "VDR1T3"              'Geräte Funktion HM 8112
310 ' Volt DC
320 ' Bereich 0,2V
330 ' Integrationszeit 1 Sek
340 '-----
```

```

350 CALL IWD (INSTRUMENT%,BEFEHL$,STATUS%)
360 '
370 '
380 '
390 LESE$ = SPACES$(30)
400 CALL IRD (LESE$,LANG%,INSTRUMENT%,STATUS%)
410 '
420 '
430 '
440 '
450 PRINT LEFT$(LESE$,LANG%)

```

Demoprogramm SRQ

```

100'-----
110'
120'Beispiel: SRQ und Seriell Poll mit zwei Geräten
130'
140'-----
150'
160 DEF SEG = &HDE00
170 INIT = 0 : IWD = 9 : IRD = 21 : ISPL = 12
180 ADRESS% = 21 : PEGEL% = 0
190 CALL INIT (ADRESS%,PEGEL%)
200 '
210 INSTRUMENT1% = 7
220 INSTRUMENT2% = 8
230 S1$="VDR1T3L0S0Q1"
240 S2$="VDR1T4L0S0Q1"
250
260
270
280
290
300
310 '
320 CALL IWD (INSTRUMENT1%,S1$,STATUS%)
330
340 CALL IWD (INSTRUMENT2%,S2$,STATUS%)
350
360'
370'
380 CALL ISPL (INSTRUMENT1%,pol%,STATUS%)
390 CALL ISPL (INSTRUMENT2%,pol%,STATUS%)
400'
410'
420 IF (INP(&H2BA) AND &H40) = 0 THEN 420
430
440 ST1 = 0 : ST2 = 0
450 '
460 CALL ISPL (INSTRUMENT1%,pola%,STATUS%)
470 i = (pola% AND &H40)
480 LOCATE 3,5 : PRINT "poll 1 = "; pola%
490 IF pola% > 0 THEN ST1 = 1
500 '

```

```

510 CALL ISPL (INSTRUMENT2%,polb%,STATUS%)
520 i = (polb% AND &H40 )
530 LOCATE 4,5 : PRINT "poll 2 = "; polb%           'Status poll 2
540 IF polb% > 0 THEN ST2 = 1
550 '
580 IF ST1 = 0 THEN 640
590 L$ = SPACES(40)
600 CALL IRD (L$,LANG%,INSTRUMENT1%,STATUS%)
610 LOCATE 6,5 : PRINT "Gerät 1 " L$               'Meßwert
630 '
640 IF ST2 = 0 THEN 420
650 L$ = SPACES(40)
660 CALL IRD (L$,LANG%,INSTRUMENT2%,STATUS%)
670 LOCATE 8,5 : PRINT "Gerät 2 " L$               'Meßwert
690 GOTO 420

```

Demoprogramm SEC und IRDA

```

10 INITIALIZ:
20 DEF SEG = &HDE00                               'GPIB ROM address
30 ADRESS% = 21
40 INSTRUMENT% = 11
50 LEVEL% = 0
60 INIT% = 0
70 IWSD% = 30
90 IWD% = 36
100 IRD% = 39
160 IRDA% = 203
220 '_____
230 CLS
240 GOSUB GPIBINIT
250 KEY(1) ON
275 KEY(2) ON
280 KEY(10) ON
285 ON KEY(1) GOSUB PRINT1
287 ON KEY(2) GOSUB BLOCK
290 ON KEY(10) GOSUB GOHOME
310 GOSUB FKTLINE
320 WAITING:
330 GOTO WAITING
340 GOHOME:
370 END
380 '_____
390 FKTLINE:
400 LOCATE 1, 1
410 PRINT "DEMO-Auslesen der Daten HM 1007 mit HO79 mittels Sekundäradresse „
420 PRINT "(Interfaceadresse: HDE00) HO80-2"
430 PRINT "F1-Werte Auslesen           F2-Grafische Darstellung           F10-Beenden „
450 RETURN
460 '_____
590 GPIBINIT:
600 DEF SEG = &HDE00
610 CALL ABSOLUTE(ADRESS%, LEVEL%, INIT%) 'Initialization
620 '_____
630 DEF SEG

```

```

640 T$$ = TIMES$
650 SLEEP1:
660 IF T$$ = TIMES$ GOTO SLEEP1
670 T$$ = TIMES$
680 SLEEP2:
690 IF T$$ = TIMES$ GOTO SLEEP2
700 RETURN
710 '-----
720 BLOCK:
730 DIM DA(2050), BLOCK(2050)
735 DEF SEG = &HDE00
740 BEF$ = „DIG“
750 CALL ABSOLUTE(INSTRUMENT%, BEF$, STATUS%, IWD%)
770 BEF$ = „MLA TALK 11 SEC 10“
780 CALL ABSOLUTE(BEF$, STATUS%, IWSD%)
830 ANZAHL% = 2048
840 SEGMENT% = -1
850 OFFSET% = VARPTR(BLOCK(0))
860 CALL ABSOLUTE(SEGMENT%, OFFSET%, ANZAHL%, LANG%, STATUS%, IRDA%)
870 DEF SEG
880 CLS : SCREEN 2
890 S = 1: X = 1
900 FOR X = 1 TO 2048 STEP 4
910 DA(X) = PEEK(OFFSET% + X)
920 PSET (S, ((256 - DA(X)) / 2))
930 S = S + 1
940 NEXT X
950 S = 1: X = 2048
960 PSET (S, ((256 - DA(X)) / 2))
970 FOR X = 2048 TO 4096 STEP 4
980 S = S + 1
990 NEXT X
1050 RETURN
1100 PRINT1:
1110 DEF SEG = &HDE00
1130 BEF$ = „DIG“
1140 CALL ABSOLUTE(INSTRUMENT%, BEF$, STATUS%, IWD%)
1150 BEF$ = „MLA TALK 11 SEC 10“
1160 CALL ABSOLUTE(BEF$, STATUS%, IWSD%)
1170 ANZAHL% = 2048
1180 SEGMENT% = -1
1190 OFFSET% = VARPTR(BLOCK(0))
1200 CALL ABSOLUTE(SEGMENT%, OFFSET%, ANZAHL%, LANG%, STATUS%, IRDA%)
1210 FOR I = 1 TO 2048
1220 DA(I) = PEEK(OFFSET% + I)
1230 PRINT DA(I);
1240 NEXT I
1300 RETURN

```

Demoprogramm IWDA

```

100'-----
110' Beispiel: Betriebsart Senden von einem Datenblock
120' binärer Daten (Array transfer).
130'-----

```

```

140 '
150 DIM BLOCK% (1024)           'Dimensionierung für
160 '                           Block senden
180 DEF SEG=&HDE00              'Speicheradresse
190 INIT = 0 : IWD = 9 : IRD = 21 'Unterrountinen Offset
200 IWSD = 3 : IRSD = 6
210 '
220 ADRESS%=21                  'PC Controller Adresse 21
230 PEGEL%=0                    'Systemcontroller
240 '
250 INSTRUMENT%=6              'Geräteadresse LISTNER
260 '
270 '
280 CALL INIT(ADRESS%,PEGEL%)   'Schnittstelle aktivieren
290 '
300 '-----Direkt Kommando ausführen-----
310 '
320 DEF SEG=&HDE00
330 '
340 BEFEHL$= "MTA LISTEN 6"     'Aktivieren Listener
350 '
360 CALL IWSD (BEFEHL$,STATUS%) 'Eingabe der Sekundäradresse
370 '                           zum Daten - Block schreiben
390 '-----Datenblock erzeugen-----
400 FOR I = 1 TO 1024 : BLOCK%(I) = 68 : NEXT
410 '   Datenblock füllen mit dem ASCII-Wert von A
420 '   Hier ist die Speicherverwaltung des Computers zu beachten.
430 '   Integerwerte belegen 2 Byte im RAM. Ist der Wert 8 Bit
440 '   oder kleiner, wird zwischen jedem übertragenem Datenbyte eine
450 '   Null mit übertragen.
460 '-----Datenblock schreiben-----
470 '
480 IWDA = 200
490 '
500 ANZAHL%=1024                'Anzahl der zu Übertragenen
510 '                           Datenbyts.
520 '
530 SEGMENT% = -1 : OFFSET% = 0 : OFFSET = 0 : STATUS% = 0
540 '
550 'OFFSET% = VARPTR (BLOCK%(1)) 'Hauptspeicheradresse
560 '   erstes Datenbyte
570 '
580 EOI% = 1                     'Endzeichen letztes Datenbyte EOI = 0 nein
590 '                           EOI = 1 ja
610 OFFSET% = VARPTR (BLOCK%(1)) 'Hauptspeicheradresse
620 CALL IWDA (SEGMENT%,OFFSET%,ANZAHL%,EOI%,STATUS%)
630 '   Befehl zum Übertragen eines
640 '   Datenblocks
650 '
660 '-----RESET IEEE - Bus-----
670 DEF SEG = &HDE00
680 BEFEHL$ = "UNL UNT DCL"
690 CALL IWSD (BEFEHL$,STATUS%)
700 END

```


Programmbeschreibung Turbo-Pascal Units

UNIT HO80_1;

INTERFACE

VAR

{Die beiden Variablen IEEE_card und IEEE_SEG dürfen nur gelesen werden und sind nicht zu verändern!}

IEEE_card: boolean; { true : IEEE-Karte ist installiert }
{ false : IEEE-Karte ist nicht installiert }

IEEE_SEG : word; { enthält Segment-Adresse der IEEE-Kartenadr. }
{ wird automatisch bestimmt, nicht verändern ! }

{ Es folgen vordefinierte Variablen für die Verwendung der Procedures. }

Pegel,
Status,
Instrument,
Pol,
Anzahl,
Lang,
Eoi,
Adress : integer;
Kommando,
Befehl : string;

{ Folgende Pointer sind Einsprungadressen für Assemblerrouitinen }
{ nicht verändern ! }

ADR_INIT, ADR_IWSD, ADR_IRSD, ADR_IWD, ADR_IRD,
ADR_ISPL, ADR_IPPL, ADR_IWDA, ADR_IRDA, ADR_DEVICE : pointer;

Initialisierung

PROCEDURE INIT

(VAR Adress, Pegel : integer);
{ Adress = PC Geräte Adresse, z.B. 21 [0..30 }
{ Pegel = Controllereinstellung }
{ 0 : System - Controller }
{ 1 : aktiver - Controller }
{ 2 : Gerät; kein Controller }

Übermittlung von Gerätenachrichten, bzw. Datenworten

PROCEDURE IWD

(VAR Instrument : integer; VAR Befehl : string; VAR Status : integer);
{ sende Gerätenachricht, bzw. Datenwort }
{ Instrument = Adresse des Empfangsgerätes }
{ Befehl = Befehlsstring, Gerätenachricht }

```
{ Status = Übermittlungsstatus }
```

PROCEDURE IRD

```
(VAR Befehl : string; VAR Instrument : integer; VAR Status :integer);
```

```
{ empfangen Gerätenachricht, bzw. Datenwort }  
{ Befehl = Befehlsstring, Gerätenachricht }  
{ Instrument= Adresse des Sendegerätes }  
{ Status = Übermittlungsstatus }
```

Übermittlung von Schnittstellen - und Gerätenachrichten, bzw. Datenworten

PROCEDURE IRSD

```
(VAR Kommando : string; VAR Status : integer);
```

```
{ empfangen Schnittstellen- und Gerätenachricht, bzw. Datenwort }  
{ Kommando=Befehlsstring, Schnittstellen- bzw. Gerätenachricht }  
{ Status = Übermittlungsstatus }
```

PROCEDURE IWSD

```
(VAR Kommando : string; VAR Status : integer);
```

```
{ sende Schnittstellen- und Gerätenachricht, bzw. Datenwort }  
{ Kommando=Befehlsstring, Schnittstellen- bzw. Gerätenachricht }  
{ Status = Übermittlungsstatus }
```

Übermittlung von Datenarrays

```
{ Vor Aufruf der Prozeduren IRDA und IWDA ist einem Pointer }  
{ (PtrDA) die Adresse des Arrays zu übergeben }  
{ PtrDA := addr (DA); }  
{ wobei DA das Datenarray ist, zB DA : array [0..1023] of byte }
```

PROCEDURE IRDA

```
(VAR PtrDA : pointer; VAR Anzahl, Lang, Status : integer);
```

```
{ empfangen Datenarray }  
{ PtrDA = Zeiger auf das Datenarray }  
{ Anzahl = Grösse des Datenarray }  
{ Lang = übermittelte Datensatzlänge }  
{ Status = Übermittlungsstatus }
```

PROCEDURE IWDA

```
(VAR ptrDA : pointer; VAR Anzahl, Eoi, Status : integer);
```

```
{ sende Datenarray }  
{ PtrDA = Zeiger auf das Datenarray }  
{ Anzahl = Anzahl der zu übertragenden Daten }  
{ Eoi = Übertragung von EOI }  
{ 0 : nein }  
{ 1 : ja }  
{ status = Übermittlungsstatus }
```

Abfrage von Geräten

PROCEDURE ISPL

(VAR Instrument : integer; VAR Pol : integer; VAR Status : integer);

```
{Serienabfrage eines bestimmten Gerätes      }
{Instrument      = Geräteadresse              }
{Pol            = Gerätestatus                }
{Status        = Übermittlungsstatus         }
```

PROCEDURE IPPL

(VAR Pol : integer);

```
{ Parallelabfrage      }
{ Pol = poll-Antwort   }
```

IMPLEMENTATION

Demoprogramm Turbopascal Sende und Empfang

PROGRAM IEEEPAS1;

```
{ DEMO-PROGRAMM IEEEPAS1                      }
{ BEISPIEL                                    }
{ SENDE UND EMPFANG VON DATEN                 }
{ SEND = IWD; EMPFANG = IRD                   }
{                                             }
{ UNIT HO80_1                                 }
{ STELLT FUER TURBO-PASCAL DIE VERWENDETEN   }
{ VERFÜGUNG                                   }
{ DIE KARTENADRESSE DES HO80 WIRD AUTOMATISCH }
{ ALLGEMEINE BEZEICHNER SIND VORDEFINIERT.-  }
{ beschreibung Turbopascal - Units           }
```

USES

HO80_1;

BEGIN { IEEEPAS1 }

IF not IEEE_card then HALT; { Ausführung beenden falls
kein IEEE }

```
Adress := 21;      { HO80 Adresse          }
Pegel := 0;        { Systemcontroller      }
INIT (Adress, Pegel); { Schnittstelle aktivieren }
Instrument := 7;    { Geräteadresse          }
```

```
Befehl := 'VDR1T3';{Geräte Funktion HM 8112}
                { Volt DC              }
                { Bereich 0,2 V          }
                { Integrationszeit 1 sec }
```

```
IWD (Instrument, Befehl, Status);      { sendet Befehl zum
                                        Instrument          }
```

```

FILLCHAR (Befehl, 80, ' '); Befehl[0] := #80;    { erzeugt
                                                Leerstring      }
IRD (Befehl, Instrument, Status);    { lese Messwert Multimeter }
writeln (Befehl);                    { Ausgabe Datensatz      }
END.                                  { IEEEPAS1                }

```

Demoprogramm Turbopascal Sekundäradressen und Blocktransfer

```

PROGRAM IEEEPAS2;

{ DEMO-PROGRAMM IEEEPAS2 }
{ BEISPIELPROGRAMM FUER INITIALISIERUNG, SENDUNG VON }
{ SEKUNDAERADRESSEN UND DATENARRAYTRANSFER }
{ }
{ UNIT HO80_1 }
{ STELLT FÜR TURBO-PASCAL DIE VERWENDETEN PROCEDUREN ZUR }
{ VERFÜGUNG }
{ DIE KARTENADRESSE DES HO80 WIRD AUTOMATISCH FESTGESTELLT ! }
{ ALLGEMEINE BEZEICHNER SIND VORDEFINIERT.- SIEHE PROGRAMM- }
{ BESCHREIBUNG TURBOPASCAL - UNITS }

USES
HO80_1;
VAR
  I : INTEGER;

DA : ARRAY [0..2050] OF BYTE;           {dataArray }
PtrDA : pointer;                       { Zeiger für array-Transfer IRDA,IWDA }

BEGIN
  { IEEEPAS2 }
  IF not IEEE_card then HALT; { Ausführung beenden falls }
  { kein IEEE-Bus vorhanden }
  PtrDA := addr (DA);           { Zuordnung Datenarray zu }
  { Zeiger-Var. }

{ ----- Initialisierung des IEEE-Interface ----- }

  Adress := 21;                 { HO80 Adresse }
  Pegel := 0;                   { Systemcontroller }
  INIT (Adress, Pegel);        { Schnittstelle aktivieren }

{— Direkt Kommando SEKUNDAER ADRESSE AUSFUEHREN —}

  Kommando := 'MLA TALK 6 SEC 10'; { SEC-Kommando an }
  { Geräteadresse 6 }
  { Sekundäradresse 10 }
  { aktivieren }
  IWSD (Kommando, STATUS);      { sendet Kommando }

{ ----- Datenblock lesen ----- }

ANZAHL := 2046;                 { Anzahl der array-Grösse }

```

```

IRDA (PtrDA, ANZAHL, LANG, STATUS);    { Befehl zum Übertragen      }
                                         { von einem array          }
{ _____ Datenblock anzeigen _____ }
FOR I := 1 TO LANG DO
  BEGIN
  WRITE (DA[I]:4);
  IF I MOD 16 = 0 THEN WRITELN;
  END;
{ _____ Reset IEEE - Bus _____ }
Kommando := 'UNL UNT DCL';
IWSD (Kommando, STATUS);
END. { IEEEPAS2 }

```

Programmbeschreibung C Bezeichnerliste: "MC.H"

INCLUDE-Datei für "QuickC 1.01"
 (geschütztes Warenzeichen von Microsoft)
 INCLUDE-Datei für "TURBO C"
 (geschütztes Warenzeichen von Borland International)
 Bezeichnerliste für Assembler-Funktionen
 zur Unterstützung der HAMEG Interface-Karte HO80-IEEE488

in den Beispielen aufgeführt Bezeichner
 sind folgendermaßen zu definieren :

```

int    controller,
        my_adress,
        status,
        instument,
        anz,
        eoi,
        lang,
        pol;

char   far *nachricht,
        far *antwort;

*/

#define ho80_seg HO80_SEG
#define IEEE_se IEEE_SE
#define init INIT
#define iwd IWD
#define ird IRD
#define iwsd IWSD
#define irsd IRSD
#define iwda IWDA
#define irda IRDA
#define ispl ISPL
#define ippl IPPL

extern unsigned int HO80_SEG;
/* HO80_SEG = 0xC400; */
extern int IEEE_SE (unsigned int);
/* if (IEEE_SE (HO80_SEG) != 0) exit (1); */
extern void INIT(int far *,int far *);

```

```

/* INIT (&controller, &my_adress);                               */
extern void IWD(int far *,char far *,int far *);                 */
/* IWD (&status, nachricht, &instrument);                       */
extern void IRD(int far *,int far *,int far *,char far *);     */
/* IRD (&status, &instrument, &anz, antwort);                 */
extern void IWSD (int far *, char far *);                       */
/* IWSD (&status, nachricht);                                   */
extern void IRSD (int far *, int far *, char far *);           */
/* IRSD (&status, &anz, antwort);                               */
extern void IWDA(int far *,int far *,int far *,int far *,int far *);
/* IWDA (&status, &eoi, &lang, data, &dummy)                 */
extern void IRDA(int far *,int far *,int far *,int far *,int far *);
/* IRDA (&status, &anz, &lang, data, &dummy)                 */
extern void ISPL(int far *,int far *,int far *);               */
/* ISPL (&status, &pol, &instrument);                         */
extern void IPPL(int far *);                                   */
/* IPPL (%pol);                                               */
/*

```

Die Funktionsbeschreibung ist dem Handbuch aus der Befehlsliste zu entnehmen.
Hier werden nur Abweichungen zu diesen Erklärungen aufgeführt.

HO80_SEG

Speicheradresse der HO80 Interfacekarte
Die Adresse ist auf C400H vordefiniert.
Wird eine andere Adresse verwendet, so ist diese
HO80_SEG zuzuweisen. (z.B.: HO80_SEG = 0xC000)

IEEE_SE

Die Funktion testet ob an der übergebenen Segmentadresse eine HO80 Interfacekarte installiert ist, und übergibt die Segmentadresse ho80_seg. Diese Funktion sollte vor IEEE-Routinen aufgerufen werden. (Siehe Programm HM8112.C)

Ist die Segmentadresse unbekannt, kann sie folgendermaßen ermittelt werden:

```

HO80_SEG = 0x0000;
while ((IEEE_se (HO80_SEG)) && (HO80_SEG < 0xFE00))
{ HO80_SEG += 0x0200; }IWDA, IRDA

```

Die Funktion entspricht der Beschreibung im Handbuch; die Segmentübergabe ist jedoch nur aus Kompatibilitätsgründen aufgeführt und kann ein beliebiger Zeiger sein.

Demoprogramm MS-C

```

/*
Beispielprogramm für Interfacekarte HO80 IEEE unter Quick C 1.01
*/

```

```

/*
Je nach Speichermodell sind unterschiedliche Module beim Kompilieren einzubinden :
    SMALL : CSCCOMMON.OBJ
    MEDIUM : CMCOMMON.OBJ
    LARGE : CLCOMMON.OBJ
*/

```

```

#include "MC.H"
main ()
{int my_adress, system_controller;
 int instrument;
 int status, pol;
 int segdata;
 int anz, eoi, lang;
 unsigned char data [2050];
 int bst;
 char recv[80];
 char *cmd0 = "MLA TALK 8";
 char *cmd1 = "MLA TALK 8 SEC 1";
 char *cmd2 = "MLA TALK 8 SEC 4";
 char *cmd3 = "MLA TALK 8 SEC 10";
 char *cmd4 = "MLA TALK 8 SEC 15";
 int ende;

HO80_SEG = 0x0000;
while ((IEEE_se (HO80_SEG)) && (HO80_SEG < 0xFE00))
 { HO80_SEG += 0x0200; }

 if (IEEE_se(HO80_SEG))
 { printf ("HO80-IEEE nicht gefunden");
 exit (1); }
 else
 { printf ("HO80-IEEE auf Adresse %X installiert,\n\n",HO80_SEG); }
 my_adress = 21; system_controller = 0;
 instrument = 8;
 init (&system_controller,&my_adress);
 ende = 0;
while (!ende)
 { printf ("Versions-Nummer                (0)\n");
 printf ("Datenbyte von Scope-Speicher lesen    (1)\n");
 printf ("Trigger-Rücksetzung und Statusmeldung      (2)\n");
 printf ("Scope-Speicher auslesen                    (3)\n");
 printf ("Statusmeldung                             (4)\n\n");
 printf ("Programm beenden (Q)\n\n");
 bst = getche();
 printf ("\n");
 switch (bst)
 {
 case '0':
 /* IRD - Interface Read data */
 strcpy (recv, " ");
 ird (&status,&instrument,&anz,recv);
 printf ("empfangene Daten = '%s', Anzahl = %d\n",recv,anz);
 printf ("Status = %X\n",status);
 break;
 case '1':
 /* IWSD - Interface Write Secondary Data */
 printf ("gesendete Daten : '%s'\n",cmd1);
 iwsd (&status,cmd1);
 printf ("Status = %X\n",status);
 /* IRDA - Interface Read Data Array */
 lang = 1;

```

```

irda (&status,&anz,&lang,data,&segdata);
printf ("Anzahl der empf. Daten : %d (%d)\n",anz,data[0]);
printf ("Status    = %X\n\n",status);
break;
case '2':
/* IWSD - Interface Write Secondary Data */
printf ("gesendete Daten : '%s'\n",cmd2);
iwsd (&status,cmd2);
printf ("Status = %X\n\n",status);
/* IRSD - Interface Read Secondary Data */
strcpy (recv,"");
irsd (&status,&anz,recv);
printf ("empfangene Daten = '%s', Anzahl = %d\n",recv,anz);
printf ("Status    = %X\n\n",status);
break;
case '3':
/* IWSD - Interface Write Secondary Data */
printf ("gesendete Daten : '%s'\n",cmd3);
iwsd (&status,cmd3);
printf ("Status = %X\n\n",status);
/* IRDA - Interface Read Data Array */
lang = 2049;
irda (&status,&anz,&lang,data,&segdata);
printf ("Anzahl der empf. Daten : %d\n",anz);
printf ("Status    = %X\n\n",status);
break;
case '4':
/* IWSD - Interface Write Secondary Data */
printf ("gesendete Daten : '%s'\n",cmd4);
iwsd (&status,cmd4);
printf ("Status = %X\n\n",status);
/* IRSD - Interface Read Secondary Data */
strcpy (recv,"");
irsd (&status,&anz,recv);
printf ("empfangene Daten = '%s', anz = %d\n",recv,anz);
printf ("Status    = %X\n\n",status);
break;
case 'Q': ende = 1; }
}
exit (0);
}

```


Weitere Informationen

Über die Technik des auf dieser Interfacekarte eingesetzten Controller-Bausteins NEC µPD7210c informiert Sie das NEC Data Book "Microprocessor and Peripherals".

Kompatibilität mit anderen IEEE-488 Kartenherstellern.

Dies wird dadurch erreicht, daß die Karten funktionskompatibel mit dem Capital Equipment Corp. Board PC-488 (Model Nr. 01000-00300) sind.

Beispiel DIP-Schalter: 2B8H

SCHALTER:	1	2	3	4	5	6	7	8
ON:	0	0	0	■	0	■	0	■
OFF:	■	■	■	0	■	0	■	0

Einstellung bei Lieferung

oder

National Instruments PC2a

Die Baugruppe enthält erweiterte Eigenschaften, die eine weitgehende Funktionskompatibilität mit dem NATIONAL INSTRUMENTS PC2A (DIP-Schalter 8 = OFF) ermöglichen. Auf diese Weise können Softwareprodukte, die diese Hardwarebasis benutzen, mit der HAMEG HO80-2 IEEE-488.2 Karte betrieben werden.

Beachten Sie, daß sich die Kompatibilität nur auf die HAMEG HO80-2 Interface Karte bezieht, nicht auf die gelieferte Software!

Die folgenden Eigenschaften der o. g. Produkte werden unterstützt:

- Interfacebaustein NEC 7210 Adreßdecodierung,
- DMA-Fähigkeiten,
- INTERRUPT-Fähigkeit (außer 'level-sharing').

Nicht unterstützt werden bzw. Unterschiede bestehen:

- mehr als eine Karte pro Rechner,
- keine CLOCK-Option,
- Interrupt-Sharing
- DMA-Transfer der AT-GPIB.

Im Auslieferungszustand ist die Baugruppe register-kompatibel zur CEC-PC488, wenn Sie den DIP-Schalter 8 auf die Position OFF stellen ist die HO80-2 Karte register-kompatibel zur PC2a GPIB Adapter.

DIP-Schalterstellung für CEC-488 Kompatibilität: **2B8H**

SCHALTER:	1	2	3	4	5	6	7	8
ON:	0	0	0	■	0	■	0	■
OFF:	■	■	■	0	■	0	■	0

DIP-Schalterstellung für PC2a-Kompatibilität: **2E1H**

SCHALTER:	1	2	3	4	5	6	7	8
ON:	■	■	0	0	0	■	0	0
OFF:	0	0	■	■	■	0	■	■

Wenn Sie die HO80-2 Karte mit der Treiber-Software von National Instruments (NI-488-Handler) betreiben möchten, müssen Sie den Jumper für die Interruptleitungen entfernen und im Konfigurationsprogramm 'IBCONF' von National Instruments für den Interrupt 'NONE' angeben.

DIP-Schalter und Basis-Adresse

Einstellung SW1-7

Die zur Einstellung einer bestimmten Basisadresse notwendige Stellung der Schalter SW1-7 entnehmen Sie folgender Tabelle,

HEX	DEZ	SW1	SW2	SW3	SW4	SW5	SW6	SW7
100	256	ON	ON	ON	ON	ON	OFF	ON
108	264	OFF	ON	ON	ON	ON	OFF	ON
110	272	ON	OFF	ON	ON	ON	ON	ON
118	280	OFF	OFF	ON	ON	ON	OFF	ON
•								
1E0	480	ON	ON	OFF	OFF	OFF	OFF	ON
1E8	488	OFF	ON	OFF	OFF	OFF	OFF	ON
•								
220	544	ON	ON	OFF	ON	ON	ON	OFF
•								
260	608	ON	ON	OFF	OFF	ON	ON	OFF
•								
2B8	696	OFF	OFF	OFF	ON	OFF	ON	OFF
•								
300	768	ON	ON	ON	ON	ON	OFF	OFF
•								
330	816	ON	OFF	OFF	ON	ON	OFF	OFF
•								
3E8	1000	OFF	ON	OFF	OFF	OFF	OFF	OFF

Schalter 8 wird wie folgt ausgewertet:

ON = Register-kompatibilität zu **PC-488** (CEC) und **PC2** National Instruments

OFF = Register-kompatibilität zu **PC2a** National Instruments

Bei den INTEL 25MHz Motherboards wurde festgestellt, daß das Adreßbit 15 nicht freigelegt ist. Aus diesem Grund empfehlen wir die Interface-Karten low-level-decodiert einzusetzen. Dies bedeutet, die Einstellung des DIP-Schalters Nr. 8 ON.

Tabelle HEX - BINÄR Umrechnung

HEX	BINÄR			
	MSB		LSB	
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

HEX	BINÄR			
	MSB		LSB	
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

HEX = hexadezimale Ziffer

MSB = höchstwertige Binärstelle (most significant bit)

LSB = niederwertige Binärstelle (least significant bit)

Adreßdecodierung

Die Adreßdecodierung-der Baugruppe erfolgt abhängig von der Stellung des DIP-Schalters 8. Ist Schalter 8 'OFF', dann wird wie folgt decodiert:

BIT 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
 NS ?? ?? R2 R1 RO C7 C6 C5 C4 C3 C2 C1 ?? ?? ??

Wenn Schalter 8 'ON' ist (Voreinstellung im Auslieferungszustand), dann wird folgendes Decodiermuster angewandt:

BIT 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
 ?? ?? ?? ?? NS ?? C7 C6 C5 C4 C3 C2 C1 R2 R1 RO

Abkürzungen:

- ?? Don't care - Der Wert dieses Adreßbits ist für die Decodierung bedeutungslos.
- Cn Der Wert dieses Bits wird mit dem Wert des in 'n' angegebenen DIP-Schalters verglichen. Wenn der Schalter 'ON' ist, dann ist sein Wert binär 0, ansonsten 1.
- Rn Mit diesen Bits werden die Register des Controller-Bausteins ausgewählt.
- NS Mit diesem Bit wird der Controller-Baustein auf der Baugruppe ausgewählt. Wenn NS=0 ist, wird der Controller ausgewählt, ansonsten wird ein Register angesprochen, das den Status der Baugruppe enthält.

Adreßabbild IBM PC I/O

HEXADEZIMAL- BEREICH	VERWENDUNG
000 - 100	Reserviert für Systemplatine
100 - 1F0	frei
1F0 - 1F8	Festplatte (nur PC/AT)
1F8 - 200	frei
200 - 20F	Computerspiele
210 - 217	Erweiterungseinheit (nur PC/XT)
218 - 220	frei
220 - 24F	reserviert (IBM)
250 - 277	frei
278 - 27F	Parallel Drucker 2 (nur PC/AT)
280 - 2EF	frei
2F0 - 2F7	reserviert (IBM)
2F8 - 2FF	Serieller Port 2
300 - 31F	Prototypenkarte
320 - 32F	Festplatte (nur PC/XT)
33G - 35F	frei
360 - 36F	reserviert (IBM)-
370 - 377	frei
378 - 37F	Parallel Drucker 1
380 - 38F	SDLC - Adapter
390 - 39F	frei
3A0 - 3AF	BISYNC 1 (nur PC/AT)
3B0 - 3BF	Monochromer Bildschirmadapter
3C0 - 3CF	reserviert (IBM)
3D0 - 3DF	Color/Graphic Bildschirmadapter
3E0 - 3E7	reserviert (IBM)
3E8 - 3F0	frei
350 - 3F7	Diskette
3F7 - 3FF	Serieller Port 1

(Verbindlich sind die jew. Hersteller-Angaben)

Interruptkanäle IBM PC

Nummer	Verwendung
2	PC-XT: frei PC-AT:nicht benutzen
3	Asynchron 2 Bisync 2 SDLC
4	Asynchron 1 Bisync 1 SDLC
.	Hard Disk (PC-XT)i Parallel Port 2 (PC-AT)
.	Diskette
.	Parallel Port 1

(Verbindlich sind die jeweiligen Hersteller-Angaben)

DMA - Kanäle IBM PC

Nummer	Verwendung
1	SDLC oder frei
.	Diskette
.	Hard Disk

(Verbindlich sind die jeweiligen Hersteller-Angaben)

ASCII & IEEE (GPIB) CODE CHART

key:

ASCII character

octal	25	PPU	GPIB code
hex	NAK	21	decimal

B7 B6 B5 BITS	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
B4 B3 B2 B1	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0	20 DLE 16	40 SP LA 0 32	60 0 LA 16 48	100 TA 0 64	120 TA 16 80	140 SA 0 96	160 SA 16 112
0 0 0 1	1 SOH 1	11 DC1 17	41 LA 1 33	61 LA 17 49	101 TA 1 65	121 TA 17 81	141 SA 1 97	161 SA 17 113
0 0 1 0	2 STX 2	12 DC2 18	42 LA 2 34	62 LA 18 50	102 TA 2 66	122 TA 18 82	142 SA 2 98	162 SA 18 114
0 0 1 1	3 ETX 3	13 DC3 19	43 LA 3 35	63 LA 19 51	103 TA 3 67	123 TA 19 83	143 SA 3 99	163 SA 19 115
0 1 0 0	4 EOT 4	14 DC4 20	44 LA 4 36	64 LA 20 52	104 TA 4 68	124 TA 20 84	144 SA 4 100	164 SA 20 116
0 1 0 1	5 ENQ 5	15 NAK 21	45 LA 5 37	65 LA 21 53	105 TA 5 69	125 TA 21 85	145 SA 5 101	165 SA 21 117
0 1 1 0	6 ACK 6	16 SYN 22	46 LA 6 38	66 LA 22 54	106 TA 6 70	126 TA 22 86	146 SA 6 102	166 SA 22 118
0 1 1 1	7 BEL 7	17 ETB 23	47 LA 7 39	67 LA 23 55	107 TA 7 71	127 TA 23 87	147 SA 7 103	167 SA 23 119
1 0 0 0	8 BS 8	18 CAN 24	48 LA 8 40	68 LA 24 56	110 TA 8 72	130 TA 24 88	150 SA 8 104	170 SA 24 120
1 0 0 1	9 HT 9	19 EM 25	49 LA 9 41	69 LA 25 57	111 TA 9 73	131 TA 25 89	151 SA 9 105	171 SA 25 121
1 0 1 0	A LF 10	1A SUB 26	52 LA 10 42	72 LA 26 58	112 TA 10 74	132 TA 26 90	152 SA 10 106	172 SA 26 122
1 0 1 1	B VT 11	1B ESC 27	53 LA 11 43	73 LA 27 59	113 TA 11 75	133 TA 27 91	153 SA 11 107	173 SA 27 123
1 1 0 0	C FF 12	1C FS 28	54 LA 12 44	74 LA 28 60	114 TA 12 76	134 TA 28 92	154 SA 12 108	174 SA 28 124
1 1 0 1	D CR 13	1D GS 29	55 LA 13 45	75 LA 29 61	115 TA 13 77	135 TA 29 93	155 SA 13 109	175 SA 29 125
1 1 1 0	E SO 14	1E RS 30	56 LA 14 46	76 LA 30 62	116 TA 14 78	136 TA 30 94	156 SA 14 110	176 SA 30 126
1 1 1 1	F SI 15	1F US 31	57 LA 15 47	77 UNL 63	117 TA 15 79	137 UNT 95	157 SA 15 111	177 SA 31 RUBOUT (DEL) 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

HAMEG®

Instruments

Oscilloscopes

Multimeters

Counters

Frequency

Synthesizers

Generators

R- and LC-

Meters

Spectrum

Analyzers

Power Supplies

Curve Tracers

Time Standards

Germany

HAMEG GmbH

Kelsterbacher Str. 15-19
60528 FRANKFURT am Main
Tel. (069) 67805-0
Fax (069) 6780513

France

HAMEG S.a.r.l

5-9, av. de la République
94800-VILLEJUIF
Tél. (1) 4677 8151
Telefax (1) 4726 3544

Spain

HAMEG S.L.

Villarroel 172-174
08036 BARCELONA
Teléf. (9) 3 4301597
Telefax (9) 3 3212201

Great Britain

HAMEG LTD

74-78 Collingdon Street
LUTON Bedfordshire LU1 1RX
Phone (01582) 413174
Telefax (01582) 456416

United States of America

HAMEG, Inc.

1939 Plaza Real
OCEANSIDE, CA 92056
Phone (619) 630 4080
Telefax (619) 630 6507

HAMEG, Inc.

266 East Meadow Avenue
EAST MEADOW, NY 11554
Phone (516) 794 4080
Toll-free (800) 247 1241
Telefax (516) 794 1855

Hong Kong

HAMEG LTD

Flat 1, 4/F.
Crown Industrial Building
106 How Ming St., Kwun Tong
Kowloon, Hong Kong
Phone (852) 2 793 0218
Telefax (852) 2 763 5236