

# Die VECTOR Connection Multi - I/O Karte

- *4 serielle Ports*
- *2 parallele Ports*
- *MIDI-kompatibel*
- *für Amiga2000/  
3000/4000*

**VECTOR**<sup>®</sup>  
HARD & SOFT



**VECTOR Connection**  
**Multi I/O-Karte**

Zusätzliche serielle und parallele Schnittstellen  
für Commodore Amiga 2000,3000,4000

Alle Rechte vorbehalten. Das Platinenlayout, die PAL/GAL-Listings, die Software in den Eproms, Proms und auf Diskette, das VECTOR-Logo und dieses Handbuch unterliegen dem Urheberrechtsschutz. Kein Teil dieser Anleitung, der Programme, der Schaltung oder des Layouts darf ohne unsere ausdrückliche schriftliche Genehmigung in irgendeiner Form reproduziert, vervielfältigt oder verarbeitet werden. Zuwiderhandlungen werden straf- und zivilrechtlich verfolgt.

Wir behalten uns vor, ohne weitere Ankündigung Änderungen der Hardware, der Software und der Anleitung vorzunehmen.

Alle verwendeten Markenzeichen und -namen sind eingetragene Warenzeichen ihrer jeweiligen Inhaber.

**Copyright HK-Computer GmbH (C)1992/93**

Inhaltsverzeichnis:

I.	Allgemeines	3
II.	Einbauanleitung	3
II.1	Zusätzliche Abschlußbleche	3
II.2	Anschlüsse Seriell	4
II.3	Anschlüsse Parallel	4
II.4	Einstecken der Schnittstellenkarte	4
III	Installation der Software	4
III.1.1	Seriell	4
III.1.2	Parallel	4
III.2	Installation bei mehreren Karten im System	5
III.3	De-Installation der Software	5
IV	Das serielle Device	6
IV.1	hkduart.device	6
IV.2	Der Preferences-Mechanismus	6
IV.3	HKDprefs	6
IV.3.1	Verhalten von HKDprefs unter Amiga DOS1.3	7
IV.3.2	Format der Steuerfiles unter Amiga DOS1.3	7
IV.3.3	Verhalten von HKDprefs unter AmigaOS 2.0	9
IV.4	Das Duart-Preferences Programm unter OS 2.0	9
IV.5	Die DOS-Handler	9
IV.5.1	hkduart-handler	10
IV.5.2	hkduart-aux-handler	10
V	Das parallele Device	11
V.1	pio.device	11
V.2	Die DOS-Handler	11
VI	Tools	12
VI.1	Seriell	12
VI.1.1	ReReadPrefs	12
VI.1.2	ResetChannel	12
VI.1.3	ReDirect	12
VI.2	Parallel	13
VI.2.1	PReDirect	13
VI.2.2	ParNet	13
VII	Installation der Header-Files für Entwickler	13
VII.2 1	Programmierung des pio.device	16
VIII.	Technische Daten	19
VIII.1	Pinbelegungen Parallele Schnittstelle	19
VIII.2	Pinbelegungen serielle Schnittstellen	19
VIII.3	Spezifikationen	20
VIII.4	Kompatibilitätsliste	20
	Bild	21

## **I. Allgemeines**

Die **VECTOR** Connection Multi I/O-Karte ist eine ZorroII-Schnittstellenkarte für den Amiga 2000/3000/4000 und stellt zusätzlich vier serielle und zwei parallele Ports zur Verfügung. An diese können Modems, MIDI-Interfaces, bzw. Drucker, Plotter, Scanner, Digitizer etc. angeschlossen werden.

Solange Software nach Commodore-Richtlinien programmiert wurde, wird es keine Probleme machen, diese Geräte statt an die interne Schnittstelle, an eine der Schnittstellen auf dem Interface-Board anzuschließen. Die Karte ist kompatibel zu AmigaOS 1.3, 2.x und 3.x.

### **\*WARNUNG\***

Es sollte selbstverständlich sein, daß Sie niemals während des Betriebs Stecker abziehen oder einstecken. Dies kann zu Beschädigungen an Ihrer Hardware führen. Möglicherweise kennen Sie das ja schon von Ihrem AMIGA-Computer. Sollten Sie sich Ihre VECTOR Connection einmal auf diese Art zerstören, geben wir Ihnen hier einen Tip zur Selbsthilfe.

Ist die parallele Schnittstelle beschädigt, können Sie wahrscheinlich nichts selbst machen. Allerdings hat sich diese auch als sehr robust erwiesen.

Die seriellen Schnittstellen nehmen so etwas aber schon mal übel. Allerdings können Sie solche Fehler für ein paar Groschen auch selbst beheben. In der Regel wird einer der Treiber oder Buffer der jeweiligen seriellen Schnittstelle zerstört. Diese sind gesockelt und haben die Bezeichnung 1489 und 1488. Tauschen Sie immer ein Pärchen aus, wenn ein solcher Fehler auftreten sollte. Oder noch besser, gehen Sie pfleglich mit den Schnittstellen um. Dann bleiben Sie wahrscheinlich von solchen Ausfällen verschont.

## **II. Einbauanleitung**

### **II.1 Zusätzliche Abschlußbleche**

Auf den beiden zusätzlichen Abschlußblechen finden Sie jeweils einen 25poligen parallelen Port und einen 9poligen seriellen Port, die mit Flachbandkabeln mit der **VECTOR** Connection Platine verbunden werden.

## **II.2 Anschlüsse Seriell**

Die erste und zweite serielle Schnittstelle ist über das Abschlußblech bereits nach hinten herausgeführt. Dabei ist die erste Schnittstelle (UNIT 0) der untere 25polige Stecker und die zweite Schnittstelle (UNIT 1) der obere 9polige Stecker. Für die dritte und vierte serielle Schnittstelle sind auf Ihrer Karte jeweils zwei 10polige Pfostenstecker vorgesehen, die mit den Zusatzports verbunden werden. Die markierte Leitung (Pin1) zeigt dabei zur Unterseite der Karte (siehe Bild 1). Achten Sie beim Aufstecken darauf, daß die Stecker richtigerum auf der Karte stecken.

## **II.3 Anschlüsse Parallel**

Die beiden parallelen Ports werden ebenfalls über Flachbandkabel an den 26poligen Steckverbindern auf der *VECTOR* Connection Platine angeschlossen. Die markierte Leitung (Pin1) zeigt dabei zur Rückseite der Karte (siehe Bild 1). Achten Sie beim Aufstecken darauf, daß die Stecker richtigerum auf der Karte stecken.

## **II.4 Einstecken der Schnittstellenkarte**

Schalten Sie Ihren Computer aus, ziehen Sie das Netzkabel ab und öffnen Sie das Gehäuse Ihres Amigas. Entfernen Sie drei Abschlußbleche an der Rückseite Ihres Amigas und stecken Sie die Karte in einen dazu gehörenden freien Amiga-Slot.

Montieren Sie die beiden zusätzlichen Abschlußbleche und verlegen Sie die Kabel. Vergewissern Sie sich, daß sich keine benachbarten Teile (z.B. bei zwei nebeneinander steckenden Karten) berühren. Schrauben Sie Ihr Gehäuse wieder zu.

## **III Installation der Software**

Bitte booten sie von einer Workbench-Disk oder von ihrer Festplatte. Dann legen sie die beiliegende Diskette ein.

Klicken Sie auf das Icon 'Install\_Deutsch'. Folgen Sie den Anweisungen des Scripts. Nach Beendigung der Installation schließen Sie das Textfenster durch Klicken auf das 'Schließen'-Symbol in der linken oberen Ecke.

Das Script installiert im einzelnen folgende Files auf ihrer Arbeitsdisk:

hkduart.device im Ordner DEVS:

-Dies ist das Treiberprogramm für die Karte. Es wird automatisch von Software, die die Karte ansprechen will, nachgeladen.

HKDprefs im Ordner C:

-Dieses Programm kommuniziert mit dem duart.device und kümmert sich um die Voreinstellungen.

hkduart-aux-handler und hkduart-handler im Ordner L:

-Dieses sind Hilfsprogramme, um die Karte von AmigaDOS aus ansprechen zu können.

pio.device im Ordner DEVS:

-Des ist das Treiberprogramm für die Karte. Es wird automatisch von Software, die die Karte ansprechen will, nachgeladen.

pio-handler im Ordner L:

-Dieses ist ein Hilfsprogramm, um die Karte von AmigaDOS aus ansprechen zu können

Wenn die Software für AmigaOS 2.04 installiert wird, dann wird außerdem noch das Programm 'hkduart' in den Ordner 'SYS:Prefs' kopiert.

### **III.2 Installation bei mehreren *VECTOR* Connection Karten im System**

Wenn mehrere Karten in einem Amiga System installiert werden, dann braucht die Software nur einmal installiert werden. Die Software erkennt automatisch, wenn mehr als eine Karte im System ist und konfiguriert sich entsprechend.

### **III.3 De-Installation der Software**

Bitte booten Sie von einer Workbench-Disk oder von ihrer Festplatte. Dann legen sie die Begleitdiskette ein. Klicken Sie auf das Icon 'Deinstall Deutsch'. Folgen Sie den Anweisungen des Scripts. Nach Beendigung der Installation schließen Sie das Textfenster durch Klicken auf das 'Schließen'-Symbol in der linken oberen Ecke.

## **IV Das serielle Device**

### **IV.1 hkduart.device**

Dies ist das Haupt-Treiberprogramm für die Serielle Karte. Es wird von Anwendungen mit Hilfe des Standard-Amiga-Lademechanismus nachgeladen. Sie sollten dieses Programm nicht in einen anderen Ordner als nach 'DEVS:' kopieren, da es sonst vom System nicht erkannt wird.

### **IV.2 Der Preferences-Mechanismus**

Das Device wird von einem Hilfsprogramm mit den Preferences-Einstellungen versorgt. Dieses Programm, 'HKDprefs', muß schon laufen, bevor das Device in den Speicher geladen wird. Es empfiehlt sich deshalb, dieses Programm schon in der Startup-Sequence aufzurufen.

Das HKDprefs-Programm liest die Daten beim Start entweder aus Text-Files (unter Amiga DOS1.3) oder aus einem System-File (unter AmigaOS 2.04). Unter AmigaOS 2.04 kann es vom Duart-Preferences Programm aufgefordert werden, die Daten erneut einzulesen.

Das Device liest die Preferences-Daten zu zwei Zeitpunkten aus dem HKDprefs-Programm:

- Beim Laden des Devices in den Speicher (normalerweise ein OpenDevice() Aufruf)
- Beim Senden eines CMD\_RESET Befehls an das Device.

### **IV.3 HKDprefs**

Dies ist ein sog. 'daemon'-Programm, d.h. ein Programm, das im Hintergrund laufen sollte. Es stellt die Preferences-Informationen für das Device bereit.

Hierzu sollte es in der Startup-Sequence bzw. unter AmigaOS 2.04 in 'User-Startup' wie folgt aufgerufen werden:

```
RUN >NIL: <NIL: C:HKDprefs >NIL: <NIL:
```

Bitte modifizieren Sie Ihre Startup-Sequence entsprechend, z.B. mit dem Editor "ed", weiteres hierzu finden Sie im Systemhandbuch. Sollte das Programm nicht laufen, dann kann das Device nicht die Preferences-Informationen laden und konfiguriert jeden Kanal wie folgt:

9600 Baud. RTS/CTS Handshake, 8 Bits, NO Parity.



Die eigentliche Funktion des Devices wird nicht beeinträchtigt, wenn das HKDprefs-Programm nicht läuft. Sollten Sie also knapp an Speicherplatz sein, dann brauchen sie das Programm nicht aufzurufen. HKDprefs benötigt etwa 25KByte Speicherplatz, wenn es im Hintergrund gestartet wird.

### **IV.3.1 Verhalten von HKDprefs unter Amiga DOS1.3**

**\*WARNUNG\*** Amiga DOS1.3 ist überholt und sollte nicht mehr verwendet werden. Aus Kompatibilitätsgründen funktioniert aber die Software zur seriellen Karte auch unter Amiga DOS1.3. Unter Amiga DOS1.3 liest das HKDprefs-Programm beim Starten folgende Datenfiles ein:

```
DEVS:duart/0.Prefs    für Duart-Kanal 0
DEVS:duart/1.Prefs    für Duart-Kanal 1
DEVS:duart/2.Prefs    für Duart-Kanal 2
DEVS:duart/3.Prefs    für Duart-Kanal 3
usw..
```

Sollten weniger Kanäle im System sein, dann liest das Programm natürlich auch weniger Files ein. Dieses sind Text-Files, die die Konfiguration für einen Kanal in lesbarer Form enthalten. Der Ordner 'DEVS:duart/' wird bei der Installation für Amiga DOS1.3 automatisch angelegt. Sollte ein Steuerfile für einen Kanal nicht vorhanden sein, so wird der Kanal auf folgende Parameter initialisiert:

9600 Baud. RTS/CTS Handshake, 8 Bits, No Parity.

### **IV.3.2 Format der Steuerfiles unter Amiga DOS1.3**

Eine Zeile, die mit einem '#' beginnt, ist eine Kommentarzeile. Am Anfang einer Zeile können folgende Schlüsselwörter stehen:

'Baud'

-Dies legt die Baudrate für den Kanal fest. Erlaubt sind folgende Werte: 57600, 38400, 31250, 19200, 9600, 4800, 2400, 2000, 1800, 1200,600, 300, 150, 134, 110 und 75

Sollte diese Angabe fehlen, dann wird '9600' verwendet.

'Buffer'

-Hiermit wird die Größe des Empfangspuffers festgelegt. Gültige Werte sind: 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536. Sollte diese Angabe fehlen, dann wird '8192' verwendet.

'Handshake'

-Hiermit wird die Art des verwendeten Handshakes festgelegt.

'RTS' - RTS/CTS (7-Wire) Handshake

'xon' - XON/XOFF (Software) Handshake

'none' - kein (3-Wire) Handshake

Sollte diese Angabe fehlen, dann wird 'RTS' festgelegt.

'Parity'

-Hiermit wird die Art der Parity-Erzeugung festgelegt:

'none' - keine Parity-Erzeugung

'odd' - 'odd' Parity

'even' - 'even' Parity

Sollte diese Angabe fehlen, dann wird 'none' festgelegt.

'Bits'

-Hiermit wird die Anzahl der Bits für die Übertragung eingestellt. Mögliche Angaben sind '7' und '8'. Fehlt diese Angaben, dann wird '8' eingestellt.

'StopBits'

-Hiermit wird die Anzahl der StopBits eingestellt. Mögliche Werte sind '1' und '2'. Fehlt diese Angabe, dann wird '1' angenommen.

'BreakTime'

-Hiermit wird die Länge eines 'Break'-Signals in usec festgelegt. Möglich sind alle Werte zwischen 10000 und 500000. Fehlt diese Angabe, dann wird '250000' angenommen.

Beispiel:

#

# Dies sind Kommentarzeilen für das Beispiel

#

Baud 19200

Buffer 16384

Handshake xon  
Parity even  
Bits 7  
StopBits 1  
BreakTime 250000

In diesem Beispiel wird ein Kanal auf 19200 Baud, 7Bits, even Parity mit einem StopBits initialisiert. Die Buffergröße wird auf 16 KBytes eingestellt und die Breaktime auf 250 ms eingestellt.

Diese Files sollten mit einem Texteditor, z.B. "ed" erstellt werden. Einige Beispiele sind auf der Diskette in 'SerialConnection:DEV5/duart' zu finden.

#### **IV.3.3 Verhalten von HKDprefs unter AmigaOS 2.04**

Unter AmigaOS 2.04 werden alle Angaben aus 'ENV:SYS/duart.Prefs' eingelesen. Diese File wird automatisch mit Hilfe des Duart-Preferences Programms erzeugt und verändert. Dieses File wird beim Starten des Rechners automatisch aus 'ENVARC:SYS/duart.Prefs' kopiert. Sollten Angaben für einen Kanal nicht eingestellt sein, so werden folgende Standardwerte angenommen:

9600 Baud. RTS/CTS Handshake, 8 Bits, No Parity.

#### **IV.4 Das Duart-Preferences Programm unter AmigaOS 2.04**

Dieses Programm läuft nur unter AmigaOS 2.04 und wird auch nur bei der Installation auf AmigaOS 2.04 von der Diskette kopiert. Beim Doppelklick auf das Icon öffnet sich ein Fenster, in dem alle Funktionen bequem eingestellt werden können. Mit Hilfe des Schiebers 'UNIT' kann zwischen den verschiedenen Kanälen gewählt werden. Mit der Funktion 'save' werde die Daten abgespeichert, mit 'use' nur bis zum nächsten Booten verwendet. Mit 'Cancel' wird das Programm abgebrochen.

#### **IV.5 Die DOS-Handler**

Damit die Duart-Kanäle von AmigaDOS aus angesprochen werden können, gibt es zwei sogenannte 'DOS-Handler'. Einer dieser Handler ist mit dem 'SER:' Device für die interne Schnittstelle vergleichbar, der andere mit dem 'AUX:' Device.

### **IV.5.1 hkduart-handler**

Dieser Handler ist vergleichbar mit dem SER:Handler für die interne Schnittstelle. Um einen Kanal unter AmigaDOS zu verwenden, muß ein Eintrag in das File 'DEVS:Mountlist' wie folgt gemacht werden:

DU0:

Handler = L:hkduart-handler

StartUp = hkduart/0

StackSize = 5000

Priority = 5

GlobVec = 1

#

Nachdem der Befehl 'MOUNT DU0:' eingegeben wurde, kann dieser Name als AmigaDOS-Device angesprochen werden, z.B. mit 'list >DU0:'. Die Daten werden dann über Kanal 0 des Devices ausgegeben.

Um andere Kanäle verwenden zu können, muß die Zeile 'StartUp' wie folgt geändert werden:

StartUp = hkduart/1 für Kanal 1

StartUp = hkduart/2 für Kanal 2

StartUp = hkduart/3 für Kanal 3 usw.

Es empfiehlt sich, die Nummer im Device-Namen entsprechend der Kanalnummer zu verändern.

### **IV.5.2 hkduart-aux-handler**

Dieser Handler ist vergleichbar mit dem AUX:Handler für die interne Schnittstelle. Um einen Kanal unter AmigaDOS zu verwenden, muß ein Eintrag in das File 'DEVS:Mountlist' wie folgt gemacht werden:

DA1:

Handler = L:hkduart-aux-handler

StartUp = hkduart/1

StackSize = 5000

Priority = 5

GlobVec = 1

#

Nachdem der Befehl 'MOUNT DA1:' eingegeben wurde, kann dieser Name als AmigaDOS-Device angesprochen werden. Z.B. kann eine neue Shell mit newshell DA1: gestartet werden, die Eingaben und Ausgaben laufen dann über Kanal 1 der seriellen Karte.

Um andere Kanäle verwenden zu können, muß die Zeile 'Startup' wie folgt verändert werden:

```
StartUp = hkduart/0 für Kanal 0
StartUp = hkduart/2 für Kanal 2
StartUp = hkduart/3 für Kanal 3 usw.
```

Es empfiehlt sich, die Nummer im Device-Namen entsprechend der Kanalnummer zu verändern.

## **V Das parallele Device**

### **V.1 pio.device**

Dies ist das Haupt-Treiberprogramm für die Parallele Karte. Es wird von Anwendungen mit Hilfe des Standard-Amiga-Lademechanismus nachgeladen. Sie sollten dieses Programm nicht in einen anderen Ordner als nach 'DEVS:' kopieren, da es sonst vom System nicht erkannt wird.

### **V.2 Die DOS-Handler**

Damit die Parallelen Ports von AmigaDOS aus angesprochen werden können, gibt es einen sogenannten 'DOS-Handler'. Dieser ist mit dem Commodore-Handler für 'PAR:' vergleichbar. Um einen Kanal unter AmigaDOS zu verwenden, muß ein Eintrag in das File 'DEVS:Mountlist' wie folgt gemacht werden:

```
PIO0:
Handler = L:pio-handler
StartUp = pio/0
StackSize = 5000
Priority = 5
GlobVec = 1
```

Nachdem der Befehl 'MOUNT PIO0:' eingegeben wurde, kann dieser Name als AmigaDOS-Device angesprochen werden, z.B. mit 'list >PIO0:' wird ein Directory auf einen an Port 0 angeschlossenen Drucker ausgegeben. Um andere Kanäle verwenden zu können, muß die Zeile 'StartUp' wie folgt verändert werden:

Startup = pio/1 für Kanal 1 usw.

Es empfiehlt sich, die Nummer im Device-Namen entsprechend der Kanalnummer zu verändern.

## **VI Tools**

### **VI.1 Seriell**

#### **VI.1.1 ReReadPrefs**

Dieses Tool zwingt das HKDprefs Programm, die Konfigurationsdaten erneut einzulesen. Unter Amiga DOS1.3 kann dieser Effekt erreicht werden, indem dem HKDprefs-Prozeß ein CTRL-F Signal geschickt wird (break<process>f).

#### **VI.1.2 ResetChannel**

Dieses Programm setzt einen Kanal der seriellen Karte zurück und liest die Preferences Daten aus dem HKDprefs-Programm erneut ein.

Von der Workbench aus kann ein Tooltype 'UNIT= <Kanalnummer >' gesetzt werden, womit ein bestimmter Kanal ausgewählt werden kann. Von der Shell aus kann man 'ResetChannel <Kanalnummer >' eingeben.

#### **VI.1.3 ReDirect**

Dieses Programm ist nur von der Shell aus aufrufbar und sorgt dafür, daß Programme, die von Hause aus nur auf das serial.device zugreifen können, auch Daten über die Serielle Karte schicken können.

Nach dem Aufruf 'ReDirect<Nummer>', wobei <Nummer> die Nummer des zu verwendenden seriellen Kanals ist, werden alle Ausgaben auf das interne serielle Device auf diesen Kanal umgelenkt.

Dieser Patch kann mit 'ReDirect -' wieder abgeschaltet werden.

## VI.2 Parallel

### VI.2.1 PReDirect

Dieses Programm ist nur von der Shell aus aufrufbar und sorgt dafür, daß Programme, die von Hause aus nur auf das parallel.device zugreifen können, auch Daten über die parallele Karte schicken können.

Nach dem Aufruf 'PReDirect <Nummer>', wobei <Nummer> die Nummer des zu verwendenden parallelen Ports ist, werden alle Ausgaben auf das interne parallele Device auf diesen Kanal umgelenkt.

Dieser Patch kann mit 'PReDirect -' wieder abgeschaltet werden.

### VI.2.2 ParNet

Eine der häufigsten Anwendungen für parallele Schnittstellen ist das ParNet Filesystem der Software Distillery. Damit auch diese Anwendung mit der parallelen Karte verwendet werden kann, liegen der Karte Treiber für beide PIO-Kanäle bei. Diese befinden sich in 'ParallelConnection:ParNet/Channel0' und 'ParallelConnection:ParNet/Channel1'.

Eine Dokumentation über das ParNet findet sich in 'ParallelConnection:ParNet/README' und 'ParallelConnection:ParNet/Original/README'

#### **\*WARNUNG\***

ParNet ist für erfahrene Amiga-Anwender gedacht, besonders da hierfür ein Kabel gelötet werden muß. Bei unsachgemäßer Anwendung oder falschem Anschließen des Kabels kann sowohl der Amiga als auch die parallele Karte beschädigt werden. HK-Computer GmbH übernimmt hierfür *\*keine\** Verantwortung.

## VII Installation der Header-Files für Entwickler

Zur Programmierung der Devices sind Include-Files für Assembler und C beigelegt. Kapitel VII ist nur für Programmierer interessant, wer keine Software für das Device entwickeln oder anpassen will, kann ihn getrost überspringen.

Zusammen mit der Software werden je zwei Header-Files für C und Assembler Programmierer geliefert, die helfen sollen, Software für die **VECTOR** Connection Multi I/O-Karte zu entwickeln.

Für die seriellen Ports befinden sich diese in 'SerialConnection:Includes/Includes.h", bzw. 'SerialConnection:Includes/Includes.i":

Aztec-C:        -    INCLUDE:Devices/ und ASM:Devices/  
Lattice-C:        INCLUDE: Devices/  
DICE:            DINCLUDE:Devices/

Für die parallelen Ports befinden sich diese in 'ParallelConnection: Includes.  
Kopieren sie sich diese Files in einen passenden Ordner, z.B.:

Aztec-C:            INCLUDE:Devices/ und ASM:Devices/  
Lattice-C:        INCLUDE: Devices/  
DICE:              DINCLUDE:Devices/

### **VII.1.1 Programmierung des hkduart.device**

Prinzipiell verhält sich das Device genau wie das serial.device. Fast alles, was Commodore in der Dokumentation zum serial.device angegeben hat gilt auch für das hkduart.device. Die Abweichungen werden hier besprochen (es handelt sich hierbei nur um kleine Unterschiede für sehr ausgefallene Betriebsarten).

io\_ExtFlags:

Mark und Space Parity werden nicht unterstützt.

io\_Baud:

nur 57600, 38400, 31250, 19200, 9600, 4800, 2400, 2000, 1800, 1200, 600, 300, 150, 134, 110 und 75 Baud sind möglich.

io\_BrkTime:

Nur Zeiten zwischen 10000 und 500000 Mikrosec. werden unterstützt.

io\_ReadLen io\_WriteLen:

Dürfen nicht unterschiedlich sein, die Hardware unterstützt keine verschiedenen Baudraten beim Senden und Empfangen. Nur 5 bis 8 sind erlaubt.

io\_StopBits:

Nur 1 und 2 erlaubt.



io\_SerFlags:

SERF\_RAD\_BOOGIE ist ohne Funktion. SERF\_QUEUEDBRK wird nicht unterstützt.

io\_Status:

Nur Bits 3-7 und 9-12 sind gültig.

Alle anderen Modi werden unterstützt.

### VII.1.2 DUCMD\_SETCTRLLINES

Das Device unterstützt das SETCTRLLINES Kommando nach dem ASDG-Standard; wenn ein Programm diesen Modus anbietet, dann können Sie ihn unbeschadet einschalten.

Das Standard-Device von Commodore bietet nicht die Möglichkeit, die Steuerleitungen DTR und RTS frei zu setzen und zu löschen. Beim normalen Device muß z.B. für DTR die Sequenz CloseDevice()/OpenDevice() verwenden.

Dies wird durch dieses Kommando verhindert, es ermöglicht das freie Setzen und Löschen von Steuerleitungen eines Kanals.

DUCMD\_SETCTRLLINES (CMD\_NONSTD+7)

in io\_Offset wird eine Maske gesetzt, welche Bits verändert werden soll, in io\_Length wird der Pegel angegeben.

DTR setzen:

```
#include <Devices/hkduart.h>
SerRequest->IOSer.io_Command = DUCMD_SETCTRLLINES;
SerRequest->IOSer.io_Offset = DUIOCTLF_DTR;
SerRequest->IOSer.io_Length = DUIOCTLF_DTR;
DoIO(SerRequest);
```

DTR löschen:

```
#INCLUDE <Devices/hkduart.h>
SerRequest->IOSer.io_Command = DUCMD_SETCTRLLINES;
SerRequest->IOSer.io_Offset = DUIOCTLF_DTR;
SerRequest->IOSer.io_Length = 0;
DoIO(SerRequest);
```

RTS setzen:

```
#include <devices/hkduart.h>
SerRequest->IOSer.io_Command    = DUCMD_SETCTRLINES;
SerRequest->IOSer.io_Offset     = DUIOCTLF_RTS;
SerRequest->IOSer.io_Length    = DUIOCTLF_RTS;
DoIO(SerRequest);
```

RTS löschen:

```
#include <devices/hkduart.h>
SerRequest->IOSer.io_Command    = DUCMD_SETCTRLINES;
SerRequest->IOSer.io_Offset     = DUIOCTLF_RTS;
SerRequest->IOSer.io_Length    = 0;
DoIO(SerRequest);
```

Das jeweils andere Bit wird nicht beeinflusst. Natürlich können auch beide Bits gleichzeitig verändert werden.

Alle Defines befinden sich in hkduart.h, bzw. hkduart.i

### **VII.2.1 Programmierung des pio.device**

Prinzipiell verhält sich das Device genau wie das parallel.device. Fast alles, was Commodore in der Dokumentation zum parallel.device angegeben hat gilt auch für das pio.device. Die Abweichungen werden hier besprochen (es handelt sich hierbei nur um kleine Unterschiede für sehr ausgefallene Betriebsarten).

io\_ParFlags:

PARF\_ACKMODE, PARF\_SLOWMODE, PARF\_FASTMODE werden nicht unterstützt.

Alle anderen Modi werden unterstützt.

### **VII.2.2 PIOCMD\_INACTIVE und PIOCMD\_ACTIVE**

Diese Kommandos dienen dazu, das direkte Programmieren der Hardware zu erleichtern. Mit Hilfe des Kommandos 'PIOCMD\_INACTIVE' kann ein Kanal als 'privat' deklariert werden.

Nach Ausführung des folgenden Code-Segments:

```
#include <devices/pio.h>  
SerRequest->IOSer.io_Command = PIOCMD_INACTIVE;  
DoIO(SerRequest);
```

ist der Kanal für private Zugriffe reserviert. Jeder Versuch, vom AmigaOS aus auf diesen Kanal zuzugreifen wird mit dem Fehler 'PIOErr\_Inact' beantwortet. Jetzt kann der Kanal frei programmiert werden. Dieser Zustand kann mit Hilfe von 'PIOCMD\_ACTIVE' aufgehoben werden. Nach folgendem Code-Segment:

```
#include <devices/pio.h>  
SerRequest->IOSer.io_Command = PIOCMD_ACTIVE;  
DoIO(SerRequest);
```

ist der Kanal wieder bei AmigaOS angemeldet und kann jetzt wieder ganz normal mit Hilfe der Device-Kommandos angesprochen werden.

Um das direkte Programmieren zu erleichtern liefert PIOCMD\_INACTIVE noch zwei Werte zurück:

- Zeiger auf den Kanal innerhalb des parallelen Bausteins in IO\_ACTUAL
- Zeiger auf die Basisadresse des Bausteins in IO\_LENGTH

### VII.3 Direkte Hardware-Programmierung

Als Hardware wird ein 68230 PI/T Interfacebaustein der Firma Motorola verwendet. Dokumentation über diesen Baustein findet man z.B. in den Programmierhandbüchern zu diesem Baustein sowie in den Datenblättern.

Auf der parallelen Karte sind folgende Leitungen mit folgenden Baustein-Pins verbunden:

	PAR1	PAR2	25pol.Leiste
Bedeutung			
/Strobe	H2	H4	1
D1	PA0	PB0	2
D2	PA1	PB1	3
D3	PA2	PB2	4
D4	PA3	PB3	5
D5	PA4	PB4	6
D6	PA5	PB5	7
D7	PA6	PB6	8
D8	PA7	PB7	9
/Ack	H1	H3	10
Busy	PC6	PC2	11
PaperOut	PC7	PC1	12
Select	PC4	PC0	13

Weiterhin ist der Pin PC5 (kann als Interrupt-Output programmiert werden) mit der Level 2 Interrupt-Leitung verbunden. Es kann also programmgesteuert ein solcher Interrupt ausgelöst werden.

## VIII. Technische Daten

### VIII.1 Pinbelegungen Parallele Schnittstellen

	PC	Amiga	Parallel Connection
Bedeutung			
/Strobe	1	1	1
D1	2	2	2
D2	3	3	3
D3	4	4	4
D4	5	5	5
D5	6	6	6
D6	7	7	7
D7	8	8	8
D8	9	9	9
/Ack	10	10	10
Busy	11	11	11
PaperOut	12	12	12
Select	13	13	13
+5 Volt	-	14	14
/Reset	-	16	16
Sign Gnd	17-25	17-25	17-25

### VIII.2 Pinbelegungen serielle Schnittstellen

	PC 9pol	Conn. 9pol	PC 25pol	Amiga Serie	Conn. 25pol
Bedeutung					
Schutzerde GND	-	-	-	-	-
Sendedaten TxD	3	3	2	2	2
Empfang RxD	2	2	3	3	3
Sendeteil ein RTS	7	7	4	4	4
Sendebereit CTS	8	8	5	5	5
Betriebsbereit DTR	6	6	6	6	6
Signal/Betr.Erde GND	5	5	7	7	7
Empfangsign. Pegel RI	1	1	8	8	8
+12 Volt	-	-	-	9	9
-12 Volt	-	-	-	10	10
Terminal Bereit DSR	4	4	20	20	20

### **VIII.3 Spezifikationen**

Maximale garantierte Baudrate:       38.400 Baud 68000 CPU  
  57.600 Baud 68020/68030/68040 CPU

Handshake-Modi:   RTS/CTS, XOn/XOff, kein Handshake

Parity-Modi:       Even, Odd, None

Aufgrund der verwendeten Hardware ergeben sich folgende Einschränkungen:  
Die Kanäle 0/1 bzw. auch 2/3 können nicht mit unterschiedlichen Baudraten über 19200 Baud betrieben werden. Folgende Kombinationen sind also nicht möglich:

Kanal 0: 38400   57600

Kanal 1: 57600   38400

Alle anderen Baudraten (außer MIDI) können frei kombiniert werden.

Die serielle Karte ist MIDI-tauglich. Aufgrund der Hardware kann aber immer nur ein ganzer Baustein (zwei Kanäle) auf MIDI umgestellt werden. D.h. wenn nur ein Kanal für MIDI gebraucht wird, dann kann der andere nicht mit einer andern Baudrate betrieben werden.

### **VIII.4 Kompatibilitätsliste**

Mit folgenden Programmen wurde das Device ausprobiert und funktioniert:

- vlt vt100 Terminal-Emulator V 5.045 und größer
- NComm V 1.9 Terminal Programm
- NComm 1.921
- NComm 2.0
- TrapDoor 1.80 (FIDO)
- JRComm
- Dillon UUCP Packet V 1.15 und V 1.16
- AmigaUUCP + V 1.02
- DNetV2.10
- Handshake Terminalprogramm
- AmigaCall V 5.0
- Term V 2.2, 3.1, 3.2, 3.3 und höher
- TermLite
- Commodore SLIP Software

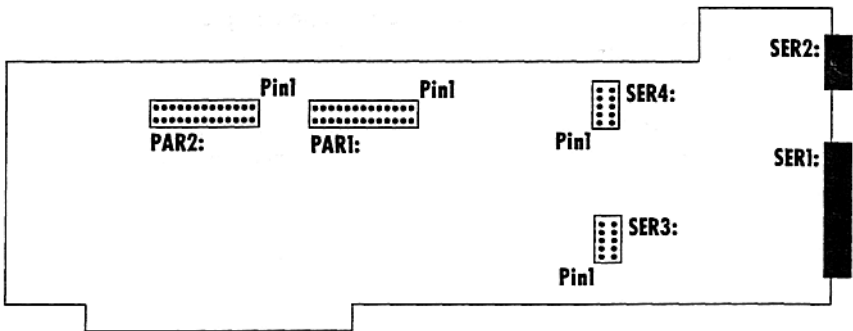
Unter MIDI:

midi.library von Fish-Disk 227 (mit redirect Programm)

Für eine Funktion mit folgenden Programmen wird *\*keine\** Garantie übernommen: (Grund könnten gravierende Programmierfehler oder Verletzungen der Commodore-Software-Richtlinien sein)

Multiterm Btx/Terminal-Programm und sämtliche in GfA-Basic geschriebene Programme.

Bild 1:





**VECTOR**<sup>®</sup>  
HARD & SOFT



# *Amiga Hardware World*

*Everything about Amiga hardware...*

~

*<http://amiga.resource.cx>*