



Picasso II

Die Grafikkarte

Handbuch

# Inhalt

<b>1. Einleitung</b> .....	<b>5</b>
1.1. Zu diesem Handbuch .....	5
1.2. Voraussetzungen .....	6
1.3. Schreibweisen .....	7
1.4. Letzte Änderungen .....	7
1.5. Ein Dankeschön .....	8
<b>2. Nach dem Auspacken</b> .....	<b>9</b>
<b>3. Schnelleinstieg</b> .....	<b>10</b>
3.1. Einbau der Picasso II in den Rechner .....	10
3.2. Software installieren .....	11
<b>4. Einbau der Karte ...</b> .....	<b>12</b>
4.1. ... in einen Amiga 2000 .....	12
4.2. ... in einen Amiga 3000 .....	14
4.3. ... in einen Amiga 4000 .....	16
<b>5. Von Auflösungen und Farbtiefen</b> .....	<b>18</b>
5.1. Wie ein Monitor arbeitet .....	18
5.2. Wenn's farbig wird .....	19
5.3. Die verschiedenen Frequenzen .....	19
5.4. 24Bit mit 1280 mal 1024 Punkten? .....	20
5.5. RAM-Geschwindigkeiten .....	24
5.6. Segmentierung .....	27
<b>6. Software installieren</b> .....	<b>29</b>
<b>7. Der Intuition-Treiber</b> .....	<b>31</b>
7.1. village.library .....	31
7.2. Picasso-Monitordatei .....	33
<b>8. Programme</b> .....	<b>37</b>
8.1. PicassoMode .....	37
8.2. ChangeScreen .....	48
8.3. PicassoSwitch .....	56
8.4. PicassoPhoto .....	57
8.5. StyxBlank .....	60

8.6. IntuiView .....	62
8.7. IntuiSpeed .....	71
8.8. VilVersion .....	74
8.9. VilReport .....	75
<b>9. Bildanzeige-Programme .....</b>	<b>79</b>
9.1. ViewIFF .....	79
9.2. ViewGIF .....	82
9.3. ViewJPEG .....	83
<b>10. Treiber für andere Programme .....</b>	<b>84</b>
<b>11. Probleme beheben .....</b>	<b>85</b>
<b>12. Programmierung der Picasso II .....</b>	<b>89</b>
12.1. Einleitung .....	89
12.2. Speicherorganisation .....	90
12.3. Einen Bildschirm öffnen .....	93
12.4. Zeichnen .....	94
12.5. Die Funktionen .....	95
<b>13. Programmierung mit C .....</b>	<b>97</b>
13.1. SAS C Version 5.1 und 6.x .....	98
13.2. Aztec 5.2a .....	98
13.3. DICE - registrierte Version 2.06.39 .....	99
13.4. Maxon C/C++ Version 1.02 .....	99
13.5. GNU C/C++ Version 2.2.2 oder höher .....	99
<b>A. Technische Daten .....</b>	<b>100</b>
<b>B. Jumper und Segmentierung .....</b>	<b>101</b>
<b>C. Steckerbelegung .....</b>	<b>103</b>
<b>D. Anschluß eines 1084 und 1081 .....</b>	<b>104</b>
<b>E. Monitor-Anschluß über BNC-Kabel .....</b>	<b>106</b>
<b>F. RAM-Aufrüstung .....</b>	<b>107</b>
<b>G. Glossar .....</b>	<b>110</b>

## 1. Einleitung

Grafik, Bewegungen, Simulationen haben den Menschen schon immer fasziniert. Versuche, künstliche Welten zu schaffen, hat es genug gegeben. Am Ende dieses Jahrhunderts sind wir fast am Ziel: Noch nie war es für jedermann so einfach möglich, täuschend echte Grafiken und Filme künstlich herzustellen wie heute. Der Computer spielt dabei *natürlich* die wichtigste Rolle.

Natürlich braucht man dazu spezielle Hardware. Der Amiga ist als erschwingliches Gerät bereits von Haus aus bestens für Video-Anwendungen geeignet. Was ihm bisher gefehlt hat, war eine Grafik mit großer Auflösung und vielen Farben. An Geschwindigkeit hat es – bis auf Auflösungen mit maximaler Farbtiefe – bisher nicht gemangelt.

Unter diesem Gesichtspunkt können wir von der Village Tronic Marketing GmbH Sie nur beglückwünschen. Sie haben eine Grafikkarte gekauft, die mit den Mängeln aufräumt, ohne es an Geschwindigkeit fehlen zu lassen. Wir möchten uns an dieser Stelle herzlich bedanken, daß Sie uns Ihr Vertrauen geschenkt haben. Sie werden es garantiert nicht bereuen.

Sollte es für Sie irgendeinen Anlaß zur Klage oder eine Anregung geben, die Sie uns mitteilen möchten, zögern Sie nicht, uns anzurufen oder uns zu schreiben. Wir wünschen uns zufriedene Kunden.

### 1.1. Zu diesem Handbuch

Dieses Handbuch beschreibt, wie man die Picasso II einbaut, was sie kann, die Installation der Software und die Bedienung der mitgelieferten Programme. Außerdem finden sich im hinteren Teil Informationen für diejenigen, die die besonderen Modi der Karte in eigenen Programmen nutzen möchten.

**Bitte** lesen Sie die Kapitel zum Einbau der Karte und zur Installation der Software sorgfältig durch. Beide Kapitel! Leider können wir nicht ausschließen, daß Sie durch unsachgemäße Handhabung der Karte oder der Software den Rechner oder den Monitor beschädigen. Wir haben dennoch alles so vorbereitet, daß Sie durch richtiges Einstecken der Karte und einfaches Starten der mitgelieferten Software weder Rechner noch Monitor zerstören können.

Die ersten Kapitel sind für den reinen Anwender gedacht, bewußt einfach gehalten und sollen das Lesen so unbeschwert wie möglich machen. Sie benötigen keine Stunde dafür. Und dennoch ist es wichtig, die Kapitel zu lesen. Nehmen Sie sich die Zeit!

Wer genauere technische Daten oder Informationen zur Programmierung benötigt, sei auf die hinteren Kapitel und den Anhang verwiesen.

Kundige Amiga-Besitzer, die bereits Erfahrungen mit dem Einbau von Karten und der Installation von systemnaher Software haben, können den Schnelleinstieg wagen und die Kapitel zur Installation der Hard- und Software getrost überblättern.

## 1.2. Voraussetzungen

Die Picasso II ist eine moderne, zeitgemäße Entwicklung. Leider hat Commodore es vor der Betriebssystemversion 2.0 versäumt, verschiedene Auflösungen und Farbtiefen der Workbench durch geeignete Aufrufe und Preference-Programme zu unterstützen. Dies geht erst, seitdem es die Workbench 2.0 gibt. Deshalb benötigen Sie zwingend eine Workbench der Version 2.0 oder höher.

Weiter kann die Picasso II den Blitter des Amigas nicht direkt nutzen, da dieser nur im sogenannten ChipMem funktioniert. Die Aufgaben des Blitters muß also ein anderer Baustein übernehmen. Um die CPU zu entlasten, haben wir auf der Karte einen Grafik-Chip mit Blitter eingebaut, der die Funktionen des Amiga-Blitters weitgehend übernimmt. Sie können die Picasso II deshalb auch in einem Rechner mit einer einfachen 68000-CPU einsetzen.

Ganz wichtig ist die Wahl des richtigen Monitors. Sie können einen Commodore 1084 oder 1084s an der Picasso II betreiben. Dazu sind aber besondere Vorkehrungen zu treffen. Sie benötigen ein besonderes Kabel und müssen einige voreingestellte Werte verändern.

Betreiben Sie einen Commodore 1084 oder 1084s nie mit den voreingestellten Werten, so wie die Software ausgeliefert wird. Sollten Sie es dennoch versuchen, werden Sie bestenfalls nichts sehen. Wahrscheinlich hören Sie aber einen Knall, bemerken

einen stechenden Geruch und können ihren Monitor zum Sondermüll tragen. Bitte lesen Sie für die Benutzung eines 1084 unbedingt das Kapitel über PicassoMode ab Seite 37 durch.

Sonst reicht ein 14-Zoll-Multiscan-Monitor wie ein NEC 3D, den man heutzutage – 1994 – für unter 1000 DM kaufen kann. Für die höheren Auflösungen ab 1024x768 Pixel (non-interlaced) benötigen Sie einen Bildschirm, der Zeilenfrequenzen bis 57kHz beherrscht (siehe auch *Von Auflösungen und Farbtiefen* ab Seite 18). Dazu gehört zum Beispiel der NEC 4FG. Sollten Sie noch keinen Monitor besitzen, wenden Sie sich einfach an uns. Wir können Sie beim Kauf eines Bildschirms beraten.

## 1.3. Schreibweisen

In diesem Handbuch werden verschiedene Schriftarten dazu verwendet, bestimmte Textteile hervorzuheben.

**Wie Sie sicher bereits gemerkt haben, erscheint besonders Wichtiges fettgedruckt.**

Dateien und Verzeichnisse, die von Bedeutung sein können, erscheinen in Schreibmaschinenschrift, wie zum Beispiel

`s:startup-sequence.`

Schräggestellte Wörter sollen eine besondere Betonung dieses Wortes hervorheben und können auch auf einen *Fachbegriff* hinweisen, der eventuell im Glossar näher erläutert wird.

Mehr Schriften kommen nicht vor, um Sie nicht zu verwirren. Denn schließlich sollen die Schriften eine Hilfe sein und nicht ein Punkt, an dem es nachzudenken gilt, welcher Schrifttyp denn nun welche Bedeutung hatte.

## 1.4. Letzte Änderungen

Das Handbuch kann nicht immer so aktuell sein wie die Software. Deshalb ist es wichtig, daß Sie bei Fragen, Unklarheiten, vermeintlichen Fehlern oder anderen Problemen zuerst einmal einen Blick auf die erste Diskette werfen, ob sich dort eine Datei namens *Liesmich* befindet, in der eventuelle Änderungen beschrieben sind.

Sollten Sie trotz der Lektüre dieser Datei noch Probleme haben, wenden Sie sich bitte zuerst an Ihren Händler. Kann auch dieser nicht helfen, setzen Sie sich bitte mit uns in Verbindung. Am besten erreichen Sie uns per Telefon. Natürlich können Sie uns auch ein Fax schicken, eine email zusenden oder eine Nachricht in unserer Mailbox hinterlassen.

### 1.5. Ein Dankeschön

Von der Idee zum fertigen Produkt ist es ein langer Weg. Neben technischen Problemen gab es zeitliche und räumliche zu überwinden. Deshalb gilt unserer besonderer Dank denjenigen, die neben ihrer Arbeit auch viel Zeit geopfert und manche Reisestrapaze auf sich genommen haben, um die Picasso II zu einem gelungenen Produkt zu machen.

Im einzelnen sind das:

<i>Software</i>	Michael Balzer Olaf 'Olsen' Barthel David Göhler Georg Heßmann Frank Mariak Uwe Röhm Thomas Sontowski Stefan Sticht
<i>Hardware</i>	Andreas Bober Klaus Burkert Hubert Neumeier Holger Wollny
<i>Support</i>	Holger Bumke Elke Grosse Yvonne Ritzka

sowie Angela Schmidt, Henning Schmiedehausen, Birgit Neumeier, Michael Hohmann, und allen ungenannten, die unser Produkt ein gutes Stück weitergebracht haben. Und vergessen darf man natürlich auch nicht Mr. Joseph Hsu, der auch die weitesten Reisen nicht gescheut hat, um die Produktion in Trab zu halten.

## 2. Nach dem Auspacken

Wenn Sie die Packung geöffnet haben, sollten sie sich vergewissern, daß Sie auch tatsächlich alle Teile erhalten haben und diese in Ordnung sind.

Zu der Lieferung gehören:

- vier Disketten<sup>1</sup>
- die Picasso II
- ein kurzes Monitorkabel
- dieses Handbuch
- evtl. weitere Handbücher



Wenn Sie etwas vermissen oder etwas offensichtlich beschädigt ist, wenden Sie sich bitte an Ihren Händler oder direkt an die Village Tronic Marketing GmbH.

Ein eventuell bestellter Video-Encoder wird separat ausgeliefert. Der Picasso II-Karton bietet dafür nicht genügend Platz.

Das mitgelieferte, kurze Monitorkabel hat einen sogenannten 15poligen VGA-Stecker. Sollten Sie keinen Flickerfixer oder Amiga 3000 besitzen, so benötigen Sie zum korrekten Anschluß des Kabels an Ihren Amiga noch einen Adapter, wie ihn Commodore zum Amiga 4000 und Amiga 1200 liefert. Sollten Sie solch einen Stecker nicht besitzen, wenden Sie sich bitte an Ihren Händler.

Der Karton, in welchem Sie die Lieferung erhalten, besteht ausschließlich aus Papier und Pappe. Er ist damit vollständig recyclebar, und kann problemlos ins Altpapier gegeben werden. Sollten Sie jedoch über genug Platz verfügen, empfehlen wir Ihnen, den Karton zu lagern. Er ist gut für den Transport von Steckkarten geeignet und vielleicht kommen Sie einmal in die Verlegenheit, eine Karte verschicken zu müssen.

<sup>1</sup> Die Anzahl der Disketten kann sich ändern

### 3. Schnelleinstieg

Sollten Sie ein Kenner des Amigas sein, so können Sie sich die ausführlichen Beschreibungen der Installation der drei nächsten Kapitel sparen und in diesem Abschnitt alles nötige erfahren, um die Picasso II in Ihrem Rechner zu installieren und die Software bedienen zu können.

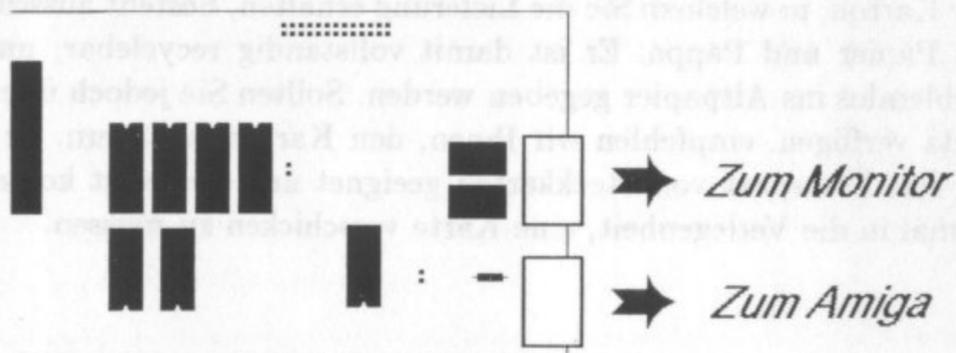
Sollten Sie dagegen noch nie oder nur selten eine Karte in Ihren Rechner eingebaut haben, dann überspringen Sie bitte diesen Abschnitt und lesen Sie bei dem Kapitel über die Hardware-Installation ab Seite 12 weiter.

#### 3.1. Einbau der Picasso II in den Rechner

Öffnen Sie Ihren Rechner und suchen Sie sich einen freien Steckplatz. Die Picasso II wird **nicht** in den Video-Slot, sondern in einen normalen Zorro-II/III-Bus-Steckplatz gesteckt. Dabei ist es egal, ob dieser links oder rechts (im Amiga 2000), oben oder unten (im Amiga 3000/4000) liegt.

Achten Sie jedoch darauf, daß die Karte fest im Steckplatz sitzt und Sie die Picasso II gut mit dem Halteblech am Rechner verschraubt haben.

Das kurze, mitgelieferte Monitorkabel wird auf der einen Seite in den Videoausgang des Amigas gesteckt, auf der anderen Seite in einen der beiden Stecker auf der Karte; und zwar in den, der näher an den Steckkontakten liegt.



Sie können nichts zerstören, sollten Sie die Kabel falsch herum einstecken. Sie werden dann nur nichts sehen, sobald Sie auf die Picasso II schalten.

#### 3.2. Software installieren

Die Software-Installation erfolgt mit dem **Installer** von Commodore, der Ihnen zu Beginn die Auswahl läßt, sich als *Experten*, *geübten Benutzer* oder *Einsteiger* einzuschätzen. Als *Einsteiger* werden Sie nach dieser Abfrage nichts mehr gefragt. Das Programm installiert dann alles nötige, Extra-Programme verbleiben allerdings auf der Diskette.

Als *geübter Benutzer* können Sie festlegen, ob und wohin Sie diverse Programme kopieren möchten. Als *Experte* schließlich können Sie auch Einstellungen vornehmen, die nicht ganz ungefährlich sind. Bevor Sie die Installation als *Experte* vornehmen, lesen Sie bitte das Kapitel über Auflösungen, Frequenzen und grundlegende Monitortechnik ab Seite 18 durch, da Sie am Ende der Installation vor Fragen gestellt werden könnten, die man ohne technisches Know how nur schwer beantworten kann.

Die neuen Auflösungen sollten anschließend im **ScreenMode-Requester** erscheinen. Wenn dies nicht der Fall ist, lesen Sie bitte im Kapitel *Probleme beheben* ab Seite 85 nach, wie Sie ein eventuell vorhandenes Problem beheben können.

Während der Installation werden unter anderem – falls von Ihnen nicht anders gewünscht – die zwei Commodities **ChangeScreen** und **StyxBlank** nach **Sys:WBStartup** kopiert. Zu **ChangeScreen** gibt es eine AmigaGuide-Datei, die Ihnen eine interaktive Hilfe anbietet, sobald Sie auf die **HELP**-Taste drücken (geht leider erst ab der Betriebssystemversion 3.0) und das Konfigurations-Fenster von **ChangeScreen** zu sehen und aktiv ist.

Wie die mitgelieferte Software zu steuern ist, können Sie einerseits durch Ausprobieren herausfinden oder in diesem Handbuch nachlesen. Das Ausprobieren hat den Vorteil, daß man seiner Neugier nachgeben darf und schneller lernt, das Handbuch dagegen bietet einen umfassenden Blick auf alle Funktionen und Bedienungselemente der Programme.

## 4. Einbau der Karte ...

Das folgende Kapitel erklärt Ihnen, wie Sie die Picasso II in Ihren Rechner einbauen können. Bitte gehen Sie dabei sehr sorgfältig vor. Wenn Sie die Arbeiten nicht selbst vornehmen möchten, kann dies auch Ihr Händler für Sie erledigen.

### 4.1. ... in einen Amiga 2000

Für den Einbau der Karte in Ihren Amiga 2000 benötigen Sie lediglich einen Kreuzschlitz-Schraubendreher. Der Karte liegt nur ein kurzes Monitorkabel für einen Flickerfixer und den Amiga 3000 bei. Sollten Sie keinen Flickerfixer im Rechner eingebaut haben, benötigen Sie einen speziellen Stecker, der im Fachhandel erhältlich ist. Commodore liefert diesen Stecker für den Amiga 1200 und Amiga 4000 aus.

Bevor Sie die Karte einbauen können, sollten Sie alle Kabel vom Rechner abziehen und das Rechnergehäuse öffnen. Dazu müssen Sie fünf Schrauben lösen. Zwei befinden jeweils an den Seiten in der Nähe des Gerätebodens, die letzte sitzt an der Rückseite in der Mitte direkt unter dem Gehäusedeckel. Bewahren Sie die Schrauben gut auf!

Wenn Sie alle Schrauben gelöst haben, müssen Sie den Gehäusedeckel ein wenig nach vorne ziehen. Anschließend können Sie ihn einfach abheben.

Wenn Sie nun in den Rechner blicken, sehen Sie rechts hinten das Netzteil, rechts vorne ein oder zwei Diskettenlaufwerke und auf der linken Seite die Steckplätze.

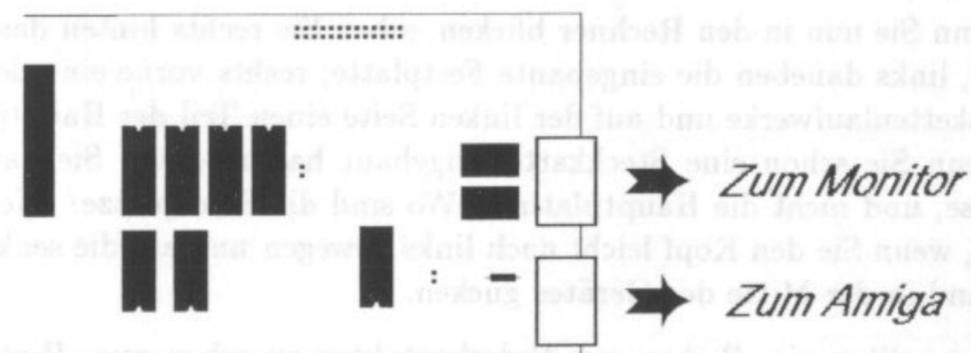
Vorne links sollten fünf gleiche Reihen mit Federkontakten zu sehen sein. Hinten, an der Gehäuserückwand, sind mit jeweils einer Schraube sieben Haltebleche angeschraubt (fünf für Amiga-Steckkarten und zwei für PC-Karten). Die Haltebleche sorgen dafür, daß Steckkarten fest im Gehäuse stecken und bei einem Transport des Rechners oder Stößen nicht durch den Rechner fliegen.

**Bevor Sie die Picasso II oder Teile innerhalb des Rechners berühren, fassen Sie bitte kurz an eine Heizung oder das Rechnergehäuse. Sie könnten sich elektrisch aufgeladen haben. Sollten Sie diese Ladung über die ICs Ihres Rechners ableiten, könnten diese Schaden nehmen. Deshalb: Vor dem Einstecken der Karte etwas Geerdetes anfassen.**

Lösen Sie nun mit dem Schraubendreher eines der Haltebleche und entfernen Sie das Halteblech. Schieben Sie anschließend die Picasso II von oben so in den Rechner, daß das Halteblech der Picasso II in die freigewordene Lücke paßt. Eventuell brauchen Sie etwas Kraft, um die Picasso II fest in den Steckplatz zu drücken.

Sitzt die Picasso II fest im Steckplatz, können Sie das Halteblech anschrauben (mit der Schraube, die ja irgendwo liegen mußte). Ist das geschehen, müssen noch zwei Verbindungen geschaffen werden: eine zwischen Rechner und Picasso II und eine zwischen Picasso II und Monitor.

Für die erste Verbindung benötigen Sie das mitgelieferte kurze Kabel. Stecken Sie es auf der einen Seite an den Videoausgang des Flickerfixers (oder in den Adapterstecker, falls Sie keinen Flickerfixer benutzen), die andere Seite gehört in die untere Buchse der Picasso II. In die noch freie Buchse der Picasso II kommt der Stecker des Monitorkabels.



Bitte kontrollieren Sie vor dem Zuschrauben des Rechners noch einmal alle Kabel und den richtigen Sitz der Picasso II.

Nun müssen Sie nur noch den Rechner zuschrauben. Bitte achten Sie beim Draufschieben des Gehäusedeckels darauf, daß keine Kabel am Deckel hängenbleiben. Nach dem Festziehen aller fünf Schrauben müssen Sie nur noch die zu Beginn abgezogenen Kabel anstecken, und dann kann die Karte in Betrieb genommen werden. Sollte es Probleme geben, lesen Sie bitte im Kapitel *Probleme beheben* nach, wie die Probleme zu lösen sind.

## 4.2. ... in einen Amiga 3000

Für den Einbau der Karte in Ihren Amiga 3000 benötigen Sie lediglich einen Kreuzschlitz-Schraubendreher.

Bevor Sie die Karte einbauen können, sollten Sie alle Kabel vom Rechner abziehen und das Rechnergehäuse öffnen. Dazu müssen Sie fünf Schrauben lösen. Zwei befinden jeweils an den Seiten in der Nähe des Gerätebodens, die letzte sitzt an der Rückseite in der Mitte direkt unter dem Gehäusedeckel. Bewahren Sie die Schrauben gut auf!

Wenn Sie alle Schrauben gelöst haben, können Sie den Gehäusedeckel nach vorne (zu Ihnen hin) abziehen. Wenn es beim Abziehen leicht hakt, dann sollten Sie den Gehäusedeckel vorne leicht nach unten neigen. Sie können den Deckel nicht abheben, sondern nur nach vorne abziehen!

Wenn Sie nun in den Rechner blicken, sehen Sie rechts hinten das Netzteil, links daneben die eingebaute Festplatte, rechts vorne ein oder zwei Diskettenlaufwerke und auf der linken Seite einen Teil der Hauptplatine (wenn Sie schon eine Steckkarte eingebaut haben, sehen Sie natürlich diese, und nicht die Hauptplatine). Wo sind die Steckplätze? Die sehen Sie, wenn Sie den Kopf leicht nach links bewegen und auf die senkrechte Wand in der Mitte des Gerätes gucken.

Dort sollten vier Reihen mit Federkontakten zu sehen sein. Hinten, an der Häuserückwand, sind mit jeweils einer Schraube vier Haltebleche angeschraubt. Vorne links kann man hinter dem Frontmetall vier Plastikschienen erkennen. Die Plastikschienen und die Haltebleche sorgen dafür, daß Steckkarten fest im Gehäuse stecken und bei einem Transport des Rechners oder Stößen nicht durch den Rechner fliegen.

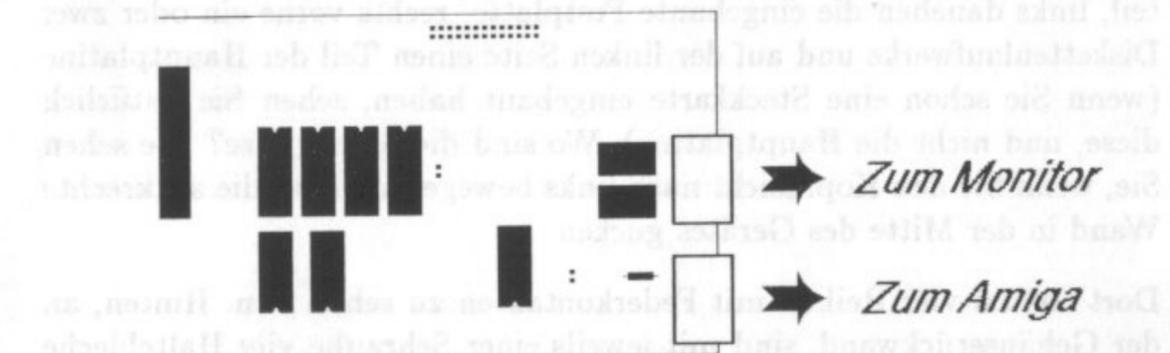
**Bevor Sie die Picasso II oder Teile innerhalb des Rechners berühren, fassen Sie bitte kurz an eine Heizung oder das Rechnergehäuse. Sie könnten sich elektrisch aufgeladen haben. Sollten Sie diese Ladung über die ICs Ihres Rechners ableiten, könnten diese Schaden nehmen. Deshalb: Vor dem Einstecken der Karte etwas Geerdetes anfassen.**

Lösen Sie nun mit dem Schraubendreher eines der Haltebleche und entfernen Sie das Halteblech. Schieben Sie anschließend die Picasso II von links so in den Rechner, daß das Halteblech der Picasso II in die freigewordene Lücke und das rechte Kartenende in die Plastikschiene paßt.

Eventuell brauchen Sie etwas Kraft, um die Picasso II fest in den Steckplatz zu drücken.

Sitzt die Picasso II fest im Steckplatz, können Sie das Halteblech anschrauben (mit der Schraube, die ja irgendwo liegen mußte). Ist das geschehen, müssen noch zwei Verbindungen geschaffen werden: Eine zwischen Rechner und Picasso II und eine zwischen Picasso II und Monitor.

Für die erste Verbindung benötigen Sie das mitgelieferte kurze Kabel. Stecken Sie es auf der einen Seite an die 15polige Video-Buchse des Amiga 3000, die andere Seite gehört in die Buchse, die näher am Netzteil ist – die rechte, wenn man von vorne in den Rechner guckt. In die noch freie Buchse der Picasso II kommt der Stecker des Monitorkabels. Sollten Sie die beiden Kabel vertauscht anstecken, geht nichts kaputt. Allerdings scheint es dann so, als ob die Karte nicht funktioniert.



Bitte kontrollieren Sie vor dem Zuschrauben des Rechners noch einmal alle Kabel und den richtigen Sitz der Picasso II.

Nun müssen Sie nur noch den Rechner zuschrauben. Bitte achten Sie beim Draufschieben des Gehäusedeckels darauf, daß keine Kabel am Deckel hängenbleiben. Eventuell müssen Sie auch beim Zusammenschieben den Gehäusedeckel vorne leicht nach unten drücken, während Sie ihn nach hinten schieben. Nach dem Festziehen aller fünf Schrauben müssen Sie nur noch die zu Beginn abgezogenen Kabel anstecken, und dann kann die Karte in Betrieb genommen werden. Sollte es Probleme geben, lesen Sie bitte im Kapitel *Probleme beheben* nach, wie die Probleme zu lösen sind.

### 4.3. ... in einen Amiga 4000

Für den Einbau der Karte in Ihren Amiga 4000 benötigen Sie lediglich einen Kreuzschlitz-Schraubendreher.

Bevor Sie die Karte einbauen können, sollten Sie alle Kabel vom Rechner abziehen und das Rechnergehäuse öffnen. Dazu müssen Sie zwei Schrauben lösen. Diese befinden sich oben an der Rückseite des Gerätes. Bewahren Sie die Schrauben nach dem Lösen gut auf!

Der Gehäusedeckel läßt sich am besten abheben, indem Sie ihn hinten ein wenig anheben und ihn dann nach hinten abziehen.

Wenn Sie nun in den Rechner blicken, sehen Sie rechts hinten das Netzteil, links daneben die eingebaute Festplatte, rechts vorne ein oder zwei Diskettenlaufwerke und auf der linken Seite einen Teil der Hauptplatine (wenn Sie schon eine Steckkarte eingebaut haben, sehen Sie natürlich diese, und nicht die Hauptplatine). Wo sind die Steckplätze? Die sehen Sie, wenn Sie den Kopf leicht nach links bewegen und auf die senkrechte Wand in der Mitte des Gerätes gucken.

Dort sollten vier Reihen mit Federkontakten zu sehen sein. Hinten, an der Gehäuserückwand, sind mit jeweils einer Schraube vier Haltebleche angeschraubt. Vorne links kann man hinter dem Frontmetall vier Plastikschiene erkennen. Die Plastikschiene und die Haltebleche sorgen dafür, daß Steckkarten fest im Gehäuse stecken und bei einem Transport des Rechners oder Stoßen nicht durch den Rechner fliegen.

**Bevor Sie die Picasso II oder Teile innerhalb des Rechners berühren, fassen Sie bitte kurz an eine Heizung oder das Rechnergehäuse. Sie könnten sich elektrisch aufgeladen haben. Sollten Sie diese Ladung über die ICs Ihres Rechners ableiten, könnten diese Schaden nehmen. Deshalb: Vor dem Einstecken der Karte etwas Geerdetes anfassen.**

Lösen Sie nun mit dem Schraubendreher eines der Haltebleche und entfernen Sie das Halteblech. Schieben Sie anschließend die Picasso II von links so in den Rechner, daß das Halteblech der Picasso II in die freigewordene Lücke und das rechte Kartenende in die Plastikschiene paßt. Eventuell brauchen Sie etwas Kraft, um die Picasso II fest in den Steckplatz zu drücken.

Sitzt die Picasso II fest im Steckplatz, können Sie das Halteblech anschrauben (mit der Schraube, die ja irgendwo liegen müßte). Ist das geschehen, müssen noch zwei Verbindungen geschaffen werden: Eine zwischen Rechner und Picasso II und eine zwischen Picasso II und Monitor.

Für die erste Verbindung benötigen Sie das mitgelieferte kurze Kabel und den Adapter für die Video-Buchse, die den 23poligen Video-Ausgang des Amiga 4000 in eine 15polige VGA-Buchse umsetzt. Diesen Stecker liefert Commodore mit aus. Stecken Sie das eine Ende des kurzen VGA-Kabels in den Adapter und das andere Ende in die Buchse der Grafikkarte, die näher am Netzteil ist – die rechte, wenn man von vorne in den Rechner guckt. In die noch freie Buchse der Picasso II kommt der Stecker des Monitorkabels. Sollten Sie die beiden Kabel vertauscht anstecken, geht nichts kaputt. Allerdings scheint es dann so, als ob die Karte nicht funktioniert.



Bitte kontrollieren Sie vor dem Zuschrauben des Rechners noch einmal alle Kabel und den richtigen Sitz der Picasso II.

Nun müssen Sie nur noch den Rechner zuschrauben. Bitte achten Sie beim Draufschieben des Gehäusedeckels darauf, daß keine Kabel am Deckel hängenbleiben. Der Deckel des Rechners läßt sich am einfachsten einsetzen, indem Sie ihn zuerst vorne hinter die Frontverkleidung schieben und dann hinten absenken. Nach dem Festziehen der zwei Schrauben müssen Sie nur noch die zu Beginn abgezogenen Kabel anstecken, und dann kann die Karte in Betrieb genommen werden.

Sollte es Probleme geben, lesen Sie bitte im Kapitel *Probleme beheben* nach, wie die Probleme zu lösen sind.

## 5. Von Auflösungen und Farbtiefen

Die Picasso II bietet Ihnen eine Vielzahl von Möglichkeiten. Sie können zwischen verschiedenen Auflösungen wählen, die Anzahl der Farben bestimmen, mit denen Sie arbeiten wollen, und Sie müssen sich auch noch um die Wahl eines Monitors kümmern – wenn Sie dies noch nicht getan haben.

Hierbei gibt es jedoch ein paar Dinge zu beachten. Die Auflösungen und die Anzahl der Farben kann man zum Beispiel nicht frei kombinieren. Auch kann man bestimmte hohe Auflösungen non-interlaced nur mit guten – sprich großen – Monitoren nutzen. Wer es trotzdem mit einem kleinen Monitor und großen Auflösungen versucht, kann diesen dabei zerstören. Deshalb folgt jetzt eine Einführung in die Technik, damit Sie verstehen, wie die Picasso II und ihr Monitor funktionieren.

### 5.1. Wie ein Monitor arbeitet

Wenn Sie wissen, wie ein Fernseher funktioniert, dann wissen Sie im Prinzip auch, wie ein Monitor funktioniert.

Hinter der Mattscheibe sitzt eine Kathodenstrahlröhre. Klingt kompliziert, ist es aber nicht. Durch hohe Spannungen werden am hinteren Ende der Bildschirnröhre (in der Kathode) Elektronen freigesetzt – im Prinzip wie bei einer Glühbirne –, um anschließend über starke Magnetfelder in Richtung Benutzer geschleudert zu werden. Dieser hat Glück, denn die Elektronen bleiben an der Mattscheibe hängen, wo ein kleiner Lichtblitz entsteht.

Statt nun die Elektronen chaotisch durch die Gegend zu schießen, erfolgt dies nach einem festen Muster. Die Elektronen werden zu einem Strahl gebündelt und dieser Strahl wird von links nach rechts, Zeile für Zeile über den Bildschirm geschwenkt. Dort, wo der Strahl auf die Mattscheibe trifft, leuchtet ein Bildpunkt (Pixel) kurz auf, um danach bald wieder zu verblassen.

Damit der Benutzer von dieser Schwäche der Mattscheibe nichts mitbekommt, muß man die einzelnen Pixel oft genug in der Sekunde erneut treffen, quasi auffrischen. Der Strahl fetzt also Zeile für Zeile von links nach rechts über den Bildschirm und kaum rechts unten angekommen darf er wieder links oben loslegen. Die Anzahl solcher Durchläufe pro

Sekunde nennt man Bildwiederholfrequenz. Je größer sie ist, desto besser ist es für das Auge. Denn je öfter das Bild in der Sekunde aufgefrischt wird, um so weniger flimmert es.

Jeder kennt sicher das Interlace-Flackern des Amigas. Hierbei beträgt die Bildwiederholfrequenz 25 oder 30 Hz – 25 oder 30 Bilder pro Sekunde. Ein Noninterlace-Bild hat normalerweise 50 oder 60 Hz. Aber auch bei diesen Bildwiederholgeschwindigkeiten bemerkt das Auge noch ein leichtes Flackern. Mittlerweile ist man sich einig, daß 70 oder mehr Hz notwendig sind, damit das Auge nicht mehr durch das Flackern gestreßt wird.

### 5.2. Wenn's farbig wird

Die bisherige Beschreibung bezieht sich nur auf einen Schwarzweiß-Monitor. Die meisten Monitore sind mittlerweile aber farbig. Die Technik ist aber fast gleich. Statt eines Bildpunktes gibt es derer drei: einen roten, einen grünen und einen blauen. Und für jeden der drei Punkte existiert ein eigener Strahl. Genau wie einem Schwarzweiß-Gerät leuchten die Punkte verschieden stark auf (je nachdem, wie gut *sein* Strahl den jeweiligen Punkt getroffen hat) und es ergibt sich eine Mischfarbe.

Treffen die drei Strahlen ihre Punkte alle voll, ergibt das Weiß. Wird kein Punkt gestreift, sieht man Schwarz.

### 5.3. Die verschiedenen Frequenzen

Ein Monitor wird an eine Grafikkarte wie zum Beispiel die Picasso II angeschlossen. Diese muß nun in der richtigen Geschwindigkeit und Reihenfolge die Steuerinformationen an den Monitor schicken, die zu einem Bild führen sollen. Neben den reinen Farbinformationen der einzelnen Bildpunkte muß die Karte dem Monitor auch mitteilen, wann eine Zeile zu Ende und wann ein kompletter Bilddurchlauf abgeschlossen ist. Bei einem Zeilenende soll der Monitor schließlich am Anfang der nächsten Zeile weitermachen und nicht irgendwo am Rand hängenbleiben. Dazu gibt es zwei spezielle Leitungen, die dem Monitor mitteilen, wann eine Zeile abgeschlossen und ein Bild komplett aufgebaut ist. Diese werden meist mit H-Sync (horizontaler Synchronimpuls) und V-Sync (vertikaler Synchronimpuls) bezeichnet.

Neben der schon erwähnten Bildwiederholfrequenz ist die sogenannte Zeilenfrequenz von Bedeutung. Das ist die Anzahl der Zeilen, die der Monitor – im Zusammenspiel mit der Grafikkarte – pro Sekunde auf den Bildschirm bringt. Nehmen wir mal an, sie haben im Preference-Programm **ScreenMode** die Auflösung **NTSC:HighRes Interlace** gewählt und besitzen einen Flickerfixer. Mit Overscan erhalten Sie dann eine Bildschirm-Auflösung von 720x480 Punkten. Die Bildwiederholfrequenz beträgt bei NTSC-Einstellungen 60 Hz. Die Zeilenfrequenz beträgt also 480 Zeilen mal 60 = 28.800 Hz, also 28.8 kHz (in Wirklichkeit ist sie etwas höher, siehe unten).

Leider können Monitore nicht jede beliebige Zeilenfrequenz verarbeiten. Es gibt technisch bedingte Höchstwerte, die sich auch im Preis niederschlagen. Einfache 14-Zoll-Monitore funktionieren meist bis 38 kHz. Das reicht zum Beispiel gerade noch für 800 mal 600 in 60 Hz aus. Wer 1024 mal 768 Punkte noch in augenschonenden 70 Hz sehen möchte, sollte mit einem guten Multiscan-Monitor liebäugeln. Dieser sollte Frequenzen bis 57kHz beherrschen. Rechnen Sie doch einfach mal durch, wie hoch die Zeilenfrequenz bei einer Auflösung von 1024 mal 768 Punkten bei 70 Hz ist.

Da Sie mit PicassoMode Auflösungen innerhalb der technischen Grenzen frei wählen können, lassen sich nicht alle möglichen Auflösungen in einer Tabelle auflisten. Wenn Sie sich jedoch eine Bildwiederholfrequenz wählen, unterhalb derer Sie nicht mehr arbeiten wollen, und die maximale Zeilenfrequenz Ihres Monitors kennen, können sich leicht die Anzahl der Zeilen errechnen, die noch möglich ist.

#### 5.4. 24Bit mit 1280 mal 1024 Punkten?

Oft genug kann man in Anzeigen technische Daten lesen, mit denen ein Anbieter wirbt. Meist sollen die Daten der Information dienen, damit man sich – auch technisch gesehen – ein Bild von dem jeweiligen Produkt machen kann. Manchmal steigen dabei Zweifel in einem auf und man fragt sich: „Kann das sein?“. Für Erklärungen der Fachbegriffe ist in einer Anzeige oft kein Platz. Dabei ist die Video-Technik nicht so kompliziert, wie es scheinen mag; die folgenden Absätze sollen Ihnen beim Verständnis eine Hilfe sein.

Neben der Bildwiederholfrequenz, die in Hertz gemessen wird, und der Zeilenfrequenz – ein Wert in kHz – gibt es noch den sogenannten Pixeltakt (auch Videobandbreite), der in MegaHertz (MHz) gemessen wird. Dieser Wert gibt an, wieviele Pixel pro Sekunde auf den Bildschirm kommen. Manchmal gibt dieser Wert aber auch an, wieviele Bytes pro Sekunde aus dem Bildschirmspeicher gelesen werden, um am Bildschirm zu erscheinen. Dies hängt oft mit der Organisation des Speichers der Grafikkarte zusammen.

Bei Karten mit sogenannten DRAMs (dynamischen RAMs), gilt das 1 Byte = 1 Pixel-Prinzip. Neben DRAMs kann man auch VRAMs (Video RAMs) einsetzen, die aber wesentlich teurer sind als DRAMs, dafür aber auch einen wesentlich höheren Pixeltakt erlauben. Bei Karten unter 1000 DM (Stand Frühjahr 1993) kann man bei der Verwendung von mehr als einem MByte RAM auf DRAMs tippen.

Wie wird dieser Pixeltakt nun berechnet? Er ergibt sich aus der Zeilenfrequenz \* Anzahl der Bytes pro Zeile \* 1.1. Der letzte Faktor ist die Zeit, die der Bildschirmstrahl für den Rücklauf von rechts unten nach links oben benötigt. Was da herauskommen kann, sind Werte wie 60, 70, 80 oder gar 90 MHz. Etwas über 90 MHz ist dabei oft die Grenze, was mit DRAMs bestückte Karten *legal* noch können. Einige Kartenhersteller übertakten die Chips auch gerne mal, was sie dann mit Kühlkörpern zu kaschieren versuchen. Was ein Grafikchip an Pixeltakt zu leisten vermag, kann man natürlich nur dem entsprechenden Datenblatt entnehmen – was Sie meist nicht haben. Andererseits setzen die Zugriffszeiten der DRAMs der Auslesegeschwindigkeit auch Grenzen, so daß man selbst bei moderneren Chips keine Wunder erwarten kann.

Die Picasso II hat nun ein oder zwei MByte Speicher auf der Platine. Ein MByte reicht aus, um 640 mal 480 Pixel in 16 Millionen Farben – *TrueColor* genannt – noch darzustellen, denn

$$640\text{Spalten} * 480\text{Zeilen} * 3\text{Bytes/Pixel} = 921.600$$

was fast ein MByte Speicher belegt. Für jedes Pixel werden dabei drei Byte benötigt. Ein kurze Berechnung zeigt, was dabei für eine Videobandbreite benötigt wird. Nehmen wir mal an, wir hätten gerne eine Bildwiederholfrequenz von 72 Hertz.

$$72\text{Hz} * 480\text{Zeilen} * 1,1 \approx 38.000\text{Hz}$$

Das entspricht also ziemlich genau einer Zeilenfrequenz von 38 kHz. Um nun auf die Videobandbreite zu kommen, müssen noch die Pixel, die Anzahl der Bytes pro Pixel (BpP) und ein Faktor von 1,25 eingerechnet werden. Diese 1,25 rühren daher, daß der Bildschirmstrahl pro Zeile noch etwas Zeit für den Randbereich und den Rücklauf zur nächsten Zeile benötigt. Das ergibt dann

$$38.000\text{Hz} * 640\text{Pixel} * 3\text{BpP} * 1,25 = 91.200.000\text{BpS}$$

Hierbei steht *BpS* für *Bytes pro Sekunde*, das ist also die Videobandbreite mit etwas über 90 Millionen Bytes pro Sekunde oder etwas über 90 Mhz.

Für die nächst bessere Auflösung von 800 mal 600 Punkten – für die man schon zwei MByte Speicher braucht – muß man also mit der Bildwiederholfrequenz heruntergehen, da

$$72\text{Hz} * 600\text{Zeilen} * 1,1 \approx 47,5\text{kHz}$$

und

$$47.500\text{Hz} * 800\text{Pixel} * 3\text{BpP} * 1,25 = 142.500.000\text{BpP}$$

bereits jenseits aller gängigen Hardware liegt. Aber auch dann, wenn man sich mit 60 Hertz begnügt, ergibt sich noch ein Wert von

$$60\text{Hz} * 600\text{Zeilen} * 1,1 = 39,6\text{kHz}$$

$$39.600\text{Hz} * 800\text{Pixel} * 3\text{BpP} * 1,25 = 118.800.000\text{BpP}$$

was sich wohl nur mit Kühlkörpern erreichen läßt und keine Garantie für ein langes Lebens des Chips darstellt. Es geht gerade so in 50 Hertz; rechnen Sie selbst einmal!

Wenn nun jemand behauptet, er könne noch 1280 mal 1024 Pixel in TrueColor darstellen – dazu braucht man 4 MByte Speicher –, dann kann das wohl richtig sein, die Frage ist nur, in welcher Zeilen- und Bildwiederholfrequenz. Eine kurze Berechnung gibt da Aufschluß. Man nehme die maximale Videobandbreite, die ja bekanntlich bei 90 MHz

liegt, und teile Sie durch die Anzahl der Pixel, dem Faktor 1,25 und der Anzahl der Bytes pro Pixel, um an die Zeilenfrequenz zu gelangen:

$$\frac{90.000.000\text{Hz}}{1280\text{Pixel} * 3\text{BpP} * 1,25} \approx 18.700\text{Hz}$$

Mit dieser Zeilenfrequenz von fast 19 kHz werden nur ältere 14-Zoll-Monitore fertig, da sie sehr langsam ist (VGA-Frequenzen beginnen bei 31 kHz). Andererseits haben solche Monitore meist nicht das Auflösungsvermögen, um 1280 Pixel noch sauber darzustellen. Nun gut, das sei dahingestellt; wie sieht es mit der Bildwiederholfrequenz aus?

$$\frac{18.700\text{Hz}}{1024\text{Zeilen} * 1,1} = 16,6\text{Hz}$$

16 Hertz flimmern doppelt so schlimm wie PAL-Interlace-Auflösungen. Ganz abgesehen davon, daß sowenig Bilder pro Sekunde wohl gar kein Monitor mehr fängt.

Darüber hinaus sollte man sich überlegen, ob man für 4 MByte Bildspeicher überhaupt die richtige CPU im Rechner hat. Die wollen nämlich auch bewegt werden. Über den Zorro-II-Bus lassen sich aber bei guter Programmierung maximal etwas über 3 MByte pro Sekunde transferieren. Daran ändert auch ein 68040 nichts.

Stellen Sie ruhig einmal eigene Berechnungen an. Mit ein wenig Grundschulmathematik kann man so überprüfen, ob die eine oder andere Karte überhaupt das halten kann, was die Werbung verspricht. Darüber hinaus sind auch Transferratenberechnungen für Animationen recht interessant. So wird einem erst klar, daß man bei 320-mal-200-Animationen in TrueColor für 25 Bilder pro Sekunde rund 4,8 MByte pro Sekunde in den Bildspeicher schaufeln und von irgendeiner Platte besorgen müßte.

Wir haben uns deshalb dafür entschieden, die Picasso II mit einem oder zwei MByte RAM auszustatten. Ein MByte RAM reicht aus, um die höchste Noninterlace-Auflösung noch in 256 verschiedenen Farben darstellen zu können (bei gutem Monitor noch in über 65 Hz), und es reicht aus, um die Modi mit vielen Farben noch in vernünftigen Bildwiederholfrequenzen anzeigen zu können.

Zwei MByte Speicher machen vor allem dann Sinn, wenn man auch in HiColor (32768 oder 65536 Farben) und TrueColor höhere Auflösungen benötigt.

### 5.5. RAM-Geschwindigkeiten

Nicht nur die Videobandbreite des VGA-Chips stellt eine Grenze dar, die man nicht überschreiten sollte. Manchmal sind es auch die DRAM-Bausteine und deren Zugriffszeiten, die einer theoretisch möglichen Auflösungen den Garaus machen. Wenn Sie möchten, können Sie jetzt einen kleinen Ausflug in die Funktionsweise von RAMs machen.

Bei RAMs kann man ganz grob zwei verschiedene Arten unterscheiden: Statische RAMs, auch SRAMs genannt, und Dynamische RAMs, kurz DRAMs. Erstere sind schnell und teuer, zweitere billiger und mit größeren Kapazitäten erhältlich.

Den Aufbau von DRAMs kann man sich so vorstellen: in einem großen Rechteck aus Zeilen (engl.: Rows) und Spalten (engl. Columns) befinden sich kleine Zellen, die entweder 1 oder 0 sind. 1 bedeutet hierbei, daß in einer Zelle Strom gespeichert ist, 0 heißt, daß in einer Zelle kein Strom gespeichert ist.

Aus technischen Gründen besteht eine Zelle aus einem kleinen Kondensator, der zwar sehr gut isoliert ist, aber dennoch über kurz oder lang Strom verliert. Man muß also zusehen, daß regelmäßig wieder Strom in die Zellen gelangt, die eine 1 darstellen. Eine Logik muß also dafür sorgen, daß regelmäßig alle Zellen abgeklappert werden, um deren Inhalt aufzufrischen. Diese Dynamik gab den DRAMs den Namen.

Neben den Zugriffen der CPU auf die RAMs müssen diese sich also selbst immer wieder auffrischen. Die CPU kann deshalb nicht beliebig schnell und zu jeder Zeit auf die Inhalte eines RAM-Bausteins zugreifen.

Beliebig schnell geht sowieso nichts. Wie schon erläutert kann man sich einen Baustein wie ein großes Rechteck vorstellen. Tatsächlich ist ein RAM-Baustein auch so aufgebaut. Bevor man nun ein Bit auslesen kann, muß man dem RAM mitteilen, aus welcher Zeile das Bit zu entnehmen ist. Hierzu wird die Zeilenadresse an die Adressleitungen des Bausteins gelegt und ein spezielles Signal, das RAS (Row Address Strobe) auf gültig gesetzt.

Daraufhin kopiert der Baustein die komplette Zeile in einen speziellen Zwischenspeicher, der genau eine Zeile aufnehmen kann und den man nicht ständig auffrischen muß. Das dauert schon ein wenig. Wenn das geschehen ist, kann man nun an die gleichen Adressleitungen des RAM-Bausteins die Spaltenadresse anlegen und mittels des CAS-Signals (Column Address Strobe) klarmachen, daß die Spaltenadresse gültig ist. Das so ausgewählte Bit liefert der RAM-Baustein nun an einem Ausgang ab.

Der Zugriff ist damit aber noch nicht beendet. Beim Kopieren der Zeile in den Zwischenspeicher wurde die Zeile im Zellenpeicher komplett gelöscht. Die Zeile muß also zurückgeschrieben werden, damit der Inhalt nicht verloren geht.

Nun kann die CPU leider nicht gleich wieder zulangem, da zwischendurch immer mal wieder Zeilen refreshed werden müssen. Neben der eigentlich Zugriffszeit, das ist Zeit zwischen dem Anlegen der ersten Adresse bis zur Ausgabe des gewünschten Bits, gibt es also eine Erholzeit. Beide Zeiten zusammen nennt man die Zykluszeit.

Die Zykluszeit ist – über den Daumen gepeilt – meist knapp doppelt so groß wie die Zugriffszeit. Bei einem RAM mit einer Zugriffszeit von 70 Nanosekunden beträgt die Zugriffszeit also 130 Nanosekunden = 0,000 000 130 Sekunden. Wieviel Zugriffe sind damit pro Sekunde möglich?

$$\frac{1,0}{0,000000130} = 7692307$$

Das sind etwas mehr als 7,5 Millionen Zugriffe. Selbst wenn man immer 32 Bit auf einmal (in einem Zugriff) liest, kommt man so nur auf 30 MByte/s. Da stimmt was nicht, wie kommt man sonst auf eine Videobandbreite von 90 MHz! Auch beim Einsatz von 60-ns-RAMs wie auf einer Picasso II reicht man auch nicht annähernd an die 90 MHz heran.

Der Trick ist eigentlich ganz einfach: Oft kommt es vor, daß man bei einem RAM-Zugriff nicht wahllos ein Byte, Wort oder Langwort benötigt. Oft liegen diese direkt hintereinander. Beim Auslesen der RAMs einer Grafikkarte liegen die Bits fast immer direkt hintereinander.

Was passiert nun im RAM? Zuerst wird eine Zeile in den Zwischenspeicher kopiert, die Spalte in der Zeile ausgewählt, das Bit ausgegeben und die Zeile zurückgeschrieben. Für das nächste direkt dahinter liegende Bit

wird die gleiche Zeile in den Zwischenspeicher kopiert, das nächste Bit ausgewählt, ausgegeben und die Zeile zurückgeschrieben. Beim nächsten Bit ist es wieder die gleiche Zeile: An sich ist die Kopiererei der immer gleichen Zeile überflüssig. Und genau das macht man sich zunutze.

Die Zeilenadresse wird nur einmal angelegt und anschließend werden nacheinander mehrere Bits ausgelesen, indem man nur noch die Spaltenadresse ändert. Typische Vertreter von RAMs, die diesen Trick beherrschen sind Page-Mode- und Static-Column-RAMs. Bei Page-Mode-RAMs wird immer nur eine neue Spaltenadresse angelegt und das CAS-Signal gesetzt. Bei Static-Column-RAMs verzichtet man auf das CAS-Signal, was den Zugriff nochmals beschleunigt.

Leider kann man keine ganze Zeile mit diesem Trick abfahren. Nach einer gewissen Zeit müssen ein paar Refresh-Zyklen eingelegt werden, damit das RAM nicht die Inhalte der restlichen Zeilen vergißt. Der Zugriff eines 70-ns-RAMs kann man durchschnittlich auf 40 ns (immer etwas unter  $2/3$  der Zeit<sup>2</sup>) drücken. Und von diesen Page-Mode-Zugriffen kann man durchaus 10 oder noch ein paar mehr nacheinander ausführen, ehe ein Refresh nötig ist.

Ein Beispiel: der Cirrus auf der Picasso II kann 13 Zyklen nacheinander ausführen, ehe er eine Zugriffspause einlegen muß. Das sind also mit 60-ns-RAMs

$$\frac{12 * 35ns + 1 * 110ns}{13} \approx 40ns$$

gut 40 Nanosekunden pro Zugriff – und keine 110 wie sonst üblich. Damit sind knapp 25 Millionen Zugriffe pro Sekunde möglich. Liest man dabei immer ein Langwort, kann man auf diese Art 100 MByte/s aus dem Video-Speicher lesen. Das liegt sogar noch etwas über der Videobandbreite, die dem Cirrus bereits von der Frequenz-Bandbreite Grenzen auferlegt.

Versucht man aus dem RAM schneller Daten zu holen, als es dieses verträgt, gehen einige Zugriffe daneben und die Pixel fangen an zu

<sup>2</sup> Die genauen Werte muß man den Datenbüchern zu den verwendeten RAMs entnehmen

flackern und umzukippen. Das Ergebnis sind Punkte, die in falschen Farben erscheinen.

Anders als bei den Grafikchips, die gewisse Toleranzen bei der Videobandbreite kennen, kann man RAMs nicht übertakten. Die meisten 60-ns-RAMs benötigen eine Zugriffszeit von 60 Nanosekunden, nicht 59! Wer es trotzdem versucht, greift ins Nichts.

Abhilfe kann in einem solchen Fall zum Beispiel ein 64-Bit breites Speicherinterface bieten, so daß in einem Zugriff zwei Langwörter gelesen werden. Auch der Einsatz von VRAMs (spezielle Video-RAMs) bringt einiges, da bei diesen Bausteinen der Refresh und der Zugriff gleichzeitig erfolgen kann.

## 5.6. Segmentierung

Besitzer eines Amiga 2000 stoßen mittlerweile immer öfter an eine feste Grenze: acht MByte FastMem. Mehr lassen sich nicht einbinden, es sei denn, man besitzt eine Turbo-Karte mit eigenem Speicher.

Möchte man nun in ein solches System mit voll ausgebautem Speicher die Picasso II einsetzen, so wird diese nicht eingebunden. Denn sie benötigt ebenfalls einen Speicherbereich, in welchem das Video-RAM angesprochen werden kann. Eine schnelle aber nicht sehr schöne Lösung dieses Problems ist das Herausnehmen von etwas FastMem. Oft können Sie dies aber nur in Schritten von zwei oder vier MByte tun, was nicht gerade ein befriedigende Lösung darstellt.

Deshalb muß ein anderer Weg beschritten werden: die Segmentierung. Dabei wird der Video-Speicher in 16 oder 32<sup>3</sup> Teile aufgeteilt, von denen jeweils nur ein Teil sichtbar ist. Dieser Speicherbereich wird nicht als normaler Speicher eingebunden, sondern im I/O-Bereich des Amiga eingependelt. Der Vorteil ist, daß man nun auch in einem voll ausgebauten Amiga 2000 die Picasso II betreiben kann.

Nachteile gibt es leider auch ein paar. Zum einen muß jeweils der richtige Speicherbereich eingependelt sein, wenn man etwas zeichnen möchte.

<sup>3</sup> Mit einem MByte RAM sind es 16, bei zwei MByte sind es 32

Dies hat zur Folge, daß oft ein Einblenden des richtigen Bereichs im I/O-Bereich zu erfolgen hat, was Zeit kostet und das Zeichnen von Linien, Texten und anderen Objekten verkompliziert und verlangsamt.

Zum zweiten erwarten einige zusätzliche Programme für die Picasso II, daß der Bildschirmspeicher nicht zerteilt – neudeutsch: segmentiert – ist. Diese Programme arbeiten deshalb nicht, wenn die Karte segmentiert betrieben wird.

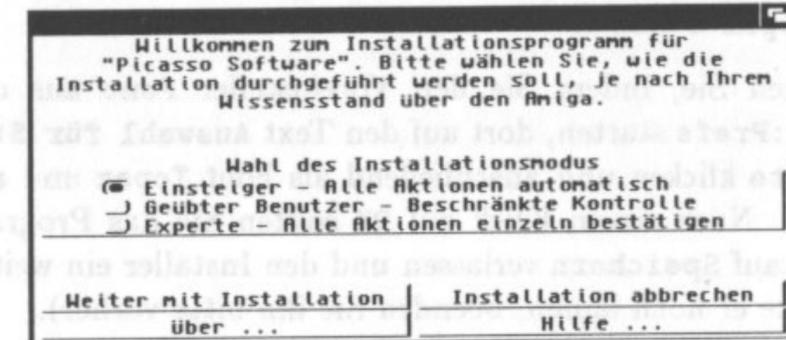
Da die Segmentierung in der Programmierung viele Schwierigkeiten mit sich bringt und die Ausgabe verlangsamt, kann diese Art der Einbindung nur eine Zwischenlösung sein. Wir werden versuchen, alle Programme so anzupassen, daß sie auch im segmentierten Modus funktionieren. Nur werden sicher die meisten Anbieter von 24-Bit-Software den Aufwand scheuen, für diesen Modus Anpassungen vorzunehmen.

Sollten Sie also das beschriebene Problem haben, so empfehlen wir Ihnen auf lange Sicht sich eine Turbo-Karte anzuschaffen, auf der Sie den Hauptspeicher erweitern können. Auf diese Weise schaffen Sie Platz für die Picasso II und können dann alle Programme ohne Einschränkung nutzen. Die A2630 von Commodore hilft Ihnen diesbezüglich leider nicht, da auch sie ihren Speicher in den unteren acht MByte einblendet.

Sollten Sie die Picasso II segmentiert betreiben wollen, so ist ein Jumper umzusetzen. Bitte lesen Sie dazu das Kapitel *Jumper und Segmentierung* auf Seite 101.

## 6. Software installieren

Die Installation der Software ist denkbar einfach. Ihre Aufgabe beschränkt sich auf einen Doppelklick auf das Installations-Icon `Install_Deutsch` der mitgelieferten Diskette `Install_Picasso`. Es erscheint darauf der typische Anfangs-Requester des Installers von Commodore.



Die Installation verändert **keine** wichtigen Systemdateien wie die `S:user-startup` oder `S:Startup-sequence`, ohne Sie zu fragen. Sie kopiert lediglich ein paar Dateien an die richtigen Stellen. Unter anderem werden

die <code>village.library</code>	in das Verzeichnis <code>Sys:Expansion</code> ,
die <code>Picasso-Monitordatei</code>	nach <code>Sys:WBStartup</code> oder auch nach <code>DEVS:Monitors</code> ,
<code>StyxBlank</code>	nach <code>Sys:WBStartup</code> ,
und <code>ChangeScreen</code>	nach <code>Sys:WBStartup</code>

kopiert.

Sollten Sie sich noch nicht mit dem Amiga auskennen, so klicken Sie im Installer auf den Knopf vor *Einsteiger*. Das Programm nimmt dann alle nötigen Aktionen selbst vor, ohne Sie mit Fragen zu belästigen.

Als Benutzer, der schon mehr mit dem Amiga gemacht hat, können Sie die Installation als *Geübter Benutzer* vornehmen. Sie haben dann jeweils die Wahl, ob Sie bestimmte Programme installieren wollen, und wohin diese kopiert werden sollen.

Als *Experte* werden Sie alles gefragt. In diesem Modus können Sie auch die kHz-Frequenzen einstellen, die Ihr Monitor noch verarbeiten kann.

Doch seien Sie dabei vorsichtig, denn bei zu hohen Angaben können Sie Ihren Monitor zerstören.

Sollten Sie nach dem Start des Installers durch einen Doppelklick auf das Installations-Icon nicht das abgebildete Fenster erscheinen, sondern ein Fenster, in dem die runden Knöpfe vor **Einsteiger**, **Geübter Benutzer** und **Experte** fehlen, dann müssen Sie den Font für die Standard-System-Texte auf **Topaz 8** stellen.

Dies erreichen Sie, indem Sie den Voreinsteller **Font** aus dem Verzeichnis **Sys:Prefs** starten, dort auf den Text **Auswahl für Standard-System-Texte** klicken und anschließend als Font **Topaz** und als Größe **8** auswählen. Nach einem Klick auf **OK** sollten Sie das Programm mit einem Klick auf **Speichern** verlassen und den Installer ein weiteres Mal starten (sollte er noch laufen, beenden Sie ihn bitte vorher).

## 7. Der Intuition-Treiber

Was nutzt die schönste Grafikkarte, wenn man sie nicht mit fast jedem Programm nutzen kann? Richtig, fast nichts. Schließlich möchte man – wo immer möglich – viele Farben und eine große Auflösung mit guten Bildwiederholffrequenzen auf seinem Monitor sehen. Man benötigt also Software, die die Grafikkarte so einbindet, als wäre sie *eingebaut*.

Leider hat Commodore das Betriebssystem nicht so erweitert, daß man Grafikkarten wie die Picasso II problemlos einsetzen kann. Ein modulares Grafikkarten-Konzept wie beim Apple Macintosh™ oder wie bei Windows™ fehlt bisher. Daher kann auch die Lösung der Picasso II nur vorübergehend ein modulares Konzept ersetzen und auch nicht 100 Prozent kompatibel sein – obwohl wir uns um die größtmögliche Kompatibilität bemüht haben.

Um die Grafikkarte so systemkonform wie möglich einzubinden, haben wir uns zweier wohl bekannter Mechanismen des Amiga-Betriebssystems bedient: Die Karte und ihre Funktionen werden über eine Library, die **village.library**, angesprochen; die Einbindung der neuen Auflösungen geschieht mittels einer Monitor-Datei, die **Picasso** heißt.

### 7.1. village.library

Die **village.library** hat zwei Aufgaben:

1. Sie ist die Schnittstelle zwischen der Karte und dem Intuition-Treiber.
2. Sie wählt die richtigen Auflösungen für den Monitor und stellt dem Benutzer nur die Auflösungen zur Verfügung, die möglich sind.

Die **village.library** muß unbedingt in **Sys:Expansion** zu finden sein. Normalerweise kopiert das Installations-Programm die Library in das Verzeichnis **Sys:Expansion**, so daß Sie sich hierum nicht kümmern müssen, wenn Sie die Installation per Doppelklick gestartet haben.

**Wichtig:** In Ihrer **S:Startup-sequence** muß das Programm **Bind-drivers** aufgerufen werden! Sonst finden die anderen Programme die Library nicht und die Karte ist nicht ansprechbar.

Dies ist normalerweise der Fall. Sollten Sie aber die S:Startup-sequence verändert haben, stellen Sie bitte sicher, daß dort Bindrivers aufgerufen wird.

Im Kapitel über die technischen Zusammenhänge war die Rede von Monitoren, die Zeilenfrequenzen bis 38kHz und bis 64kHz vertragen (Erklärungen dazu siehe Seite 20). Im Icon der *village.library* gibt es den Eintrag

```
MONITOR=38kHz
```

Sie können ihn ansehen, wenn Sie einfach auf das Icon der Library klicken und anschließend AMIGA-I drücken. Sollten Sie einen Monitor besitzen, der 57kHz oder mehr als Zeilenfrequenz verträgt (**bitte unbedingt in der Monitorokumentation nachsehen!**) ändern Sie den Eintrag in

```
MONITOR=57kHz
```

**Wichtig: Versichern Sie sich, daß der Monitor die höheren Frequenzen wirklich verträgt. Es kann sonst passieren, daß Ihr Monitor schwer Schaden nimmt. Die Village Tronic Marketing GmbH übernimmt hierfür keinerlei Haftung!** Nach der Änderung müssen Sie den Rechner neu starten, um die neue Einstellung zu aktivieren.

Mögliche Zeilenfrequenzen sind: 15, 31, 35, 38, 48, 57 und 64 kHz. Wenn Sie einen anderen Wert eintragen, werden Sie beim Hochfahren des Systems einen gelben Guru zu sehen bekommen. Es ist daher nicht zu empfehlen, hier einen anderen Wert als die aufgeführten einzutragen.

Statt eines konkreten Wertes kann man den Eintrag auch in

```
MONITOR=CUSTOM
```

ändern. Bei Hochfahren des Rechners sucht die *village.library* dann nach einer Datei mit Monitor-Daten, die von dem Programm *PicassoMode* angelegt wurde. Mit diesem Programm können Sie sich selbst Auflösungen in verschiedenen Zeilenfrequenzen erzeugen (dies wird ausführlich in einem späteren Kapitel auf Seite 37 erklärt).

Den Eintrag

```
PRODUCT=...
```

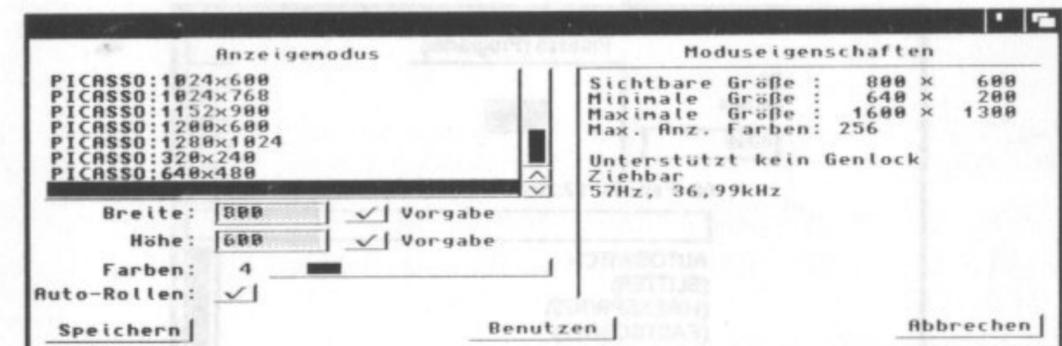
dürfen Sie auf keinen Fall verändern, da sonst die Library nicht vom Betriebssystem eingebunden wird.

## 7.2. Picasso-Monitordatei

Ohne den Intuition-Treiber *Picasso* könnten Sie die Picasso II kaum nutzen. Wie bei anderen Grafikkarten oft noch üblich, bräuchten Sie speziell auf die Grafikkarte abgestimmte Programme, die nur auf solch einer Grafikkarte laufen. Diese Lösung ist aber unbefriedigend und teuer.

Deshalb haben wir einen anderen Weg eingeschlagen und die relevanten Betriebssystem-Routinen so erweitert, daß sie auch mit der Picasso II zusammen funktionieren.

Um den Treiber *Picasso* zu aktivieren, brauchen Sie ihn einfach nur doppelt anzuklicken. Anschließend können Sie mit dem Programm *Screen-Mode* aus dem Verzeichnis *Sys:Prefs* die gewünschte Auflösung einstellen und aktivieren. Dort erscheinen in dem sogenannten Listview links oben die neuen Auflösungen:



Der ScreenMode-Voreinsteller mit den Picasso-Auflösungen

Welche neuen Auflösungen dort erscheinen, hängt davon ab, mit welcher Zeilenfrequenz Sie Ihren Monitor betreiben (können). Bei besseren Monitoren erscheinen dort mehr, da diese in der Lage sind, höhere Auflösungen darzustellen.

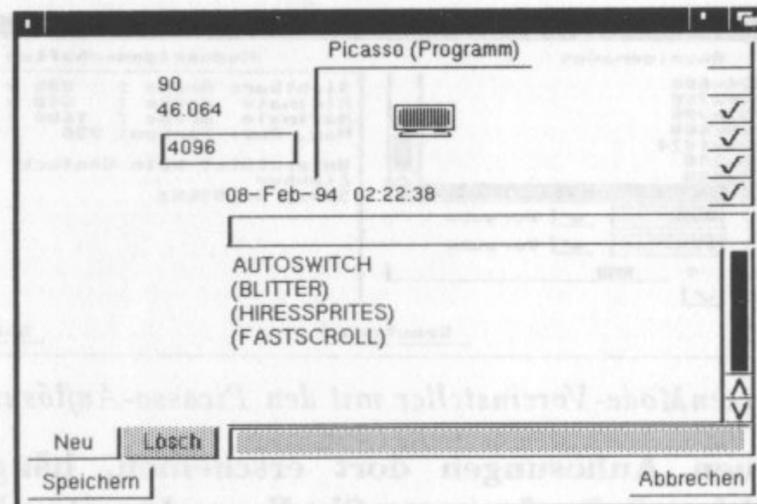
Einige Programme, die erst ab der Betriebssystemversion 2.0 laufen, bieten dem Benutzer die Möglichkeit an, den Screenmode zu wählen. Auch in diesen Auflistungen können Sie dann einfach die gewünschte Auflösung auswählen.

Für Programme, die noch für Betriebssysteme der Versionen 1.2 oder 1.3 gedacht sind, gibt es das Programm **ChangeScreen** (Beschreibung siehe Seite 48), das es ermöglicht, auch Screens älterer Programme auf der Picasso II darzustellen.

Mit der neuen Monitor-Datei erhalten Sie wesentlich mehr Vorteile als nur eine größere Auflösung. Auch ohne spezielle *AA-Chips* können Sie ab der Betriebssystemversion 3.0 Screens mit bis zu 256 Farben öffnen und benutzen. Weiterhin benötigen Sie für die Screens kein ChipMem mehr. Der Intuition-Treiber nutzt nur dann ChipMem, wenn kein FastMem mehr vorhanden ist. Normalerweise haben Sie also fast das gesamte ChipMem noch frei für Klangdaten oder Diskettenpuffer..

Wie schon bei der Library können auch beim Intuition-Treiber über das Icon vier Einstellungen gewählt werden. Zum einen ist dies

#### AUTOSWITCH



Nach einem Klick auf *DEVS:Monitors/Picasso* und Drücken von *AMIGA*

Befindet sich dieser Text in der ToolType-Liste, schaltet die Software automatisch zwischen der Amiga-Grafik und der Picasso II hin und her, je nachdem, welcher Screen gerade sichtbar ist. Wenn Sie diese Einstellung ausschalten wollen, so löschen Sie einfach dieses ToolType oder klammern Sie es einfach ein, zum Beispiel so:

(AUTOSWITCH)

Weiterhin können Sie sich – leider nur als Benutzer einer Workbench 3.0 oder höher – aussuchen, welche Art von Mauszeiger Sie benutzen möchten. Hierüber entscheidet der Eintrag

#### HIRESSPRITES

Taucht dieser Eintrag auf und ist er nicht eingeklammert, bekommen Sie einen kleinen (hochauflösenden) Mauszeiger. Fehlt dieser Eintrag oder steht er in Klammern, wird der *normale*, 16 Pixel breite LoRes-Mauszeiger in doppelter Größe dargestellt. Letzteres ist bei hohen Auflösungen sinnvoll, sonst findet man den Mauszeiger kaum noch auf dem Bildschirm.

Als dritten Eintrag kann man

#### BLITTER

eintragen. Damit erlauben Sie dem Intuition-Treiber, den Amiga-Blitter einzusetzen. Dies lohnt sich dann, wenn Sie keine schnelle CPU wie eine 68030 im Rechner haben. Im Gegenzug erkaufen Sie sich damit aber auch einen Nachteil: Die Screens müssen ins ChipMem ausgelagert werden.

Der vierte Eintrag ist mit Vorsicht zu genießen. Er heißt

#### FASTSCROLL

Er bringt Ihnen starke Geschwindigkeitsvorteile beim Scrollen, die allerdings bei wenigen Programmen zu Inkompatibilitäten führen können. Ist diese Option angeschaltet, werden immer alle *Planes* eines Fenster gescrollt – also genau andersherum als man denken würde. Da dies dem Blitter auf der Karte weniger Arbeit macht, als eine einzelne Plane zu scrollen, arbeitet der Intuition-Treiber mit **FASTSCROLL** wesentlich schneller.

Einige Programme wie zum Beispiel CygnusEd scrollen über ein paar Tricks manchmal aber nur eine Plane eines Fenster. Man erkennt das daran, daß der Cursor ohne ein Flackern stehenbleibt, während der Text unter ihm hindurchscrollt. Ist die Option **FASTSCROLL** eingeschaltet, so wird auch der Cursor gescrollt. Sollten Sie dieses Verhalten nicht wünschen, so entfernen Sie dieses ToolType oder klammern Sie es ein.

Sollten Sie die eine oder alle der Einstellungen verändern, bedenken Sie bitte, daß diese erst nach einem Neustart wirksam werden.

Eventuell gibt es bei neueren Software-Versionen noch weitere Tooltypes. Bitte lesen Sie hierzu die **Liesmich**-Datei auf der Installations-Diskette.

## 8. Programme

### 8.1. PicassoMode

**Warnung:** Das Programm PicassoMode kann Ihre Hardware beschädigen, wenn Sie nicht wissen, wie Sie es zu bedienen haben. Bitte verlassen Sie das Programm sofort, wenn Sie diese Anleitung noch nicht komplett gelesen haben oder Sie noch Zweifel haben, was die Bedienung einzelner Elemente im Programm-Fenster für Folgen haben.

**Bitte nehmen sie diese Warnung ernst !**

Der Jammer ist groß, wenn ein Schaden erst einmal da ist. Die Village Tronic Marketing GmbH übernimmt keine Kosten für Schäden, die durch die Anwendung von PicassoMode entstanden sind.

#### 8.1.1. Zweck von PicassoMode

Seit der ersten Auslieferung der Picasso II sind immer wieder Wünsche an uns herangetragen worden. Der am häufigsten genannte war der Wunsch nach mehr Auflösungen oder gar einem Programm, mit dem man sich selbst Auflösungen erzeugen kann.

Das große Problem bei solch einem Programm ist - nach wie vor - die Sicherheit. Wir haben uns unerdentlich viel Mühe gegeben, den Benutzer davor zu bewahren, seinen Monitor in die ewigen Jagdgründe zu schicken. Leider kann jedem Auflösungs-Wunsch nicht immer entsprochen werden (mancher Monitor hat da gewisse Grenzen), andererseits ist die algorithmische Berechnung aller nötigen Werte aus ein paar Grundwerten nicht einfach, damit fehlerträchtig und somit nicht unkritisch.

Oberstes Gebot war höchstmögliche Sicherheit bei einfachster Bedienung mit möglichst wenig Vorkenntnissen. Mit PicassoMode hoffen wir, unseren eigenen Ansprüchen möglichst gut gerecht worden zu sein. Wenn nicht, üben Sie Kritik und stellen Sie uns Fragen.

#### 8.1.2. Was Sie wissen sollten

Sie sollten vor der Lektüre dieser Dokumentation unbedingt das Kapitel *Von Auflösungen und Farbtiefen* ab Seite 18 dieses Handbuches durchgelesen haben. Dort wird Ihnen erklärt, was Bildwiederholffrequenz,

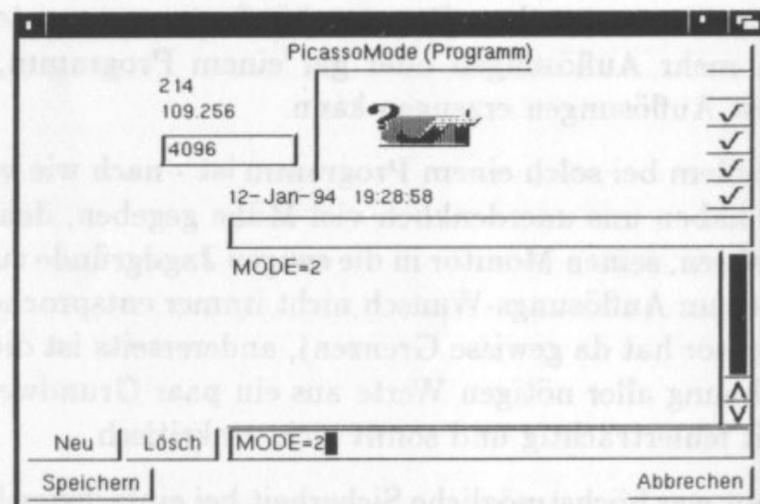
Zeilenfrequenz und Pixeltakt sind und wie sie zusammenhängen. Dieses Kenntnis ist zwar nicht zwingend nötig, aber doch überaus hilfreich, um zu verstehen, warum sich so viele Regler ändern, wenn man nur an einem gezogen hat.

Weiterhin sollten Sie schon mal nach Ihrem Monitor-Handbuch fahnden, denn Sie werden daraus einige Werte entnehmen wollen. Die Sync-Polarität hat nämlich kaum ein Computer-Benutzer im Kopf.

Nicht zuletzt sollte vor der Benutzung von PicassoMode eine Picasso II eingebaut und die Software installiert sein.

### 8.1.3. Die zwei Modi

PicassoMode kann in zwei verschiedenen Modi gestartet werden. Einen dieser Modi kann man beim Start des Programms auswählen. Im CLI geschieht das mit der Option **MODE=0** oder **MODE=2**, beim Start per Workbench ist die gleiche Option als Tooltype einzutragen.



Mit **MODE** geben Sie an, wieviel geändert werden kann

**MODE=0** In diesem Modus kann man nur einen Monitor aus der Liste der Monitore auswählen (keine eigenen Monitorbeschreibungen erzeugen) und auch nur die Position und Ausdehnung von vorhandenen Auflösungen ändern, aber keine neuen Auflösungen erzeugen.

Dieser Modus ist für denjenigen gedacht, der lediglich schnell eine Auflösung neu positionieren will oder zu Beginn ohne ge-

nauere Kenntnis seiner Hardware einfach nur seinen Monitor anhand des Namens auswählen möchte.

Startet man PicassoMode in diesem Modus, kann man nicht allzuviel falsch machen.

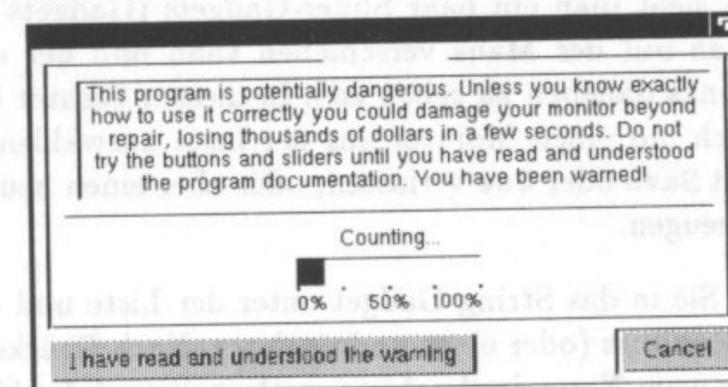
**MODE=2** In diesem Modus hat man alle Freiheiten, die das Programm einem läßt. Man kann neue Monitorbeschreibungen anlegen und mit einem Namen versehen, als auch neue Auflösungen erzeugen.

In diesem Modus sollte man aber gut überlegen, was man tut. Wer seinen Monitor als 64-kHz-Gerät einträgt, obwohl er nur 57 kHz Zeilenfrequenz verträgt, unterzieht diesen einer harten Prüfung.

Voreingestellt ist **MODE=0**, damit erst einmal nichts schief gehen kann.

### 8.1.4. Die Bedienung

PicassoMode läßt sich über drei Fenster steuern. Das erste sehen Sie, wenn Sie das Programm starten und Sie darauf hingewiesen werden, daß die Benutzung dieses Programms gefährlich ist. Dieser Fenster bleibt etwa 10 Sekunden stehen, bevor Sie es per Klick schließen können.

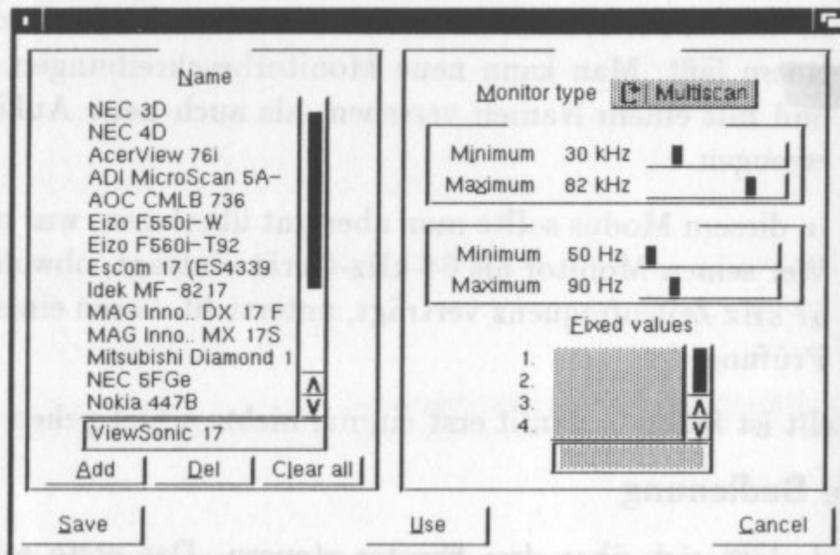


Nach dem Start von PicassoMode...

Die beiden anderen Fenster dienen der Wahl des Monitors und der Wahl der Auflösungen. Nach dem Start und dem Verschwinden des Warnfensters befinden Sie sich im Haupt-Einstellungsfenster, in dem man die Auflösungen wählt und einstellt.

Nach dem ersten Start, können Sie in diesem Fenster fast nichts anklicken, bis auf das Gadget links oben, daß mit **Monitor** überschrieben

ist. Rechts davon befindet sich ein Pfeil, der nach unten zeigt. Wenn Sie auf diesen Pfeil klicken oder die Taste **m** drücken, erscheint das Monitor-Konfigurationsfenster, das mit **Monitor definitions** überschrieben ist.



Die Qual der Wahl: Der richtige Monitor

Links sieht man eine (eventuell noch leere) Liste, in der Monitornamen stehen. Rechts sieht man ein paar Slider-Gadgets (Gadgets mit einem Knopf, den man mit der Maus verschieben kann und der einen Wert darstellt, der links daneben zu sehen ist). In diesem Fenster können Sie entweder einfach nur einen Monitor aus der Liste auswählen und dann das Fenster mit **Save** oder **Use** verlassen, oder aber einen neuen Eintrag in der Liste erzeugen.

Hierzu klicken Sie in das String-Gadget unter der Liste und tragen den Namen Ihres Monitors (oder eines anderen) ein. Nach Drücken von **RETURN** sollte der neue Name in der Liste erscheinen und die Gadgets auf der rechten Seite des Fensters freigeschaltet sein.

Nun sollten Sie die Grenzwerte Ihres Monitors festlegen. Dazu gehört, ob es ein Multiscan-Monitor oder ein Festfrequenz-Monitor ist. Dies erreichen Sie durch Klicken auf das Cycle-Gadget **Monitor Type**. Stellen Sie es auf **Multiscan**, wenn Sie einen Multiscan-Monitor angeschlossen haben.

Betreiben Sie einen A1084, A1081 oder einen ähnlichen Festfrequenz-Monitor an Ihrer Picasso II, so stellen Sie das Gadget bitte auf **Fixed-**

**Scan** (Sie können hier auch **MultiScan** wählen und den Bereich auf 15 bis 16 kHz beschränken).

Im Falle eines Multiscan-Monitors können Sie nun die minimale und maximale Zeilenfrequenz (engl.: Line frequency) und die minimale und maximale Bildwiederholfrequenz (engl.: Scan rate) einstellen. Bitte gehen Sie hierbei sehr sorgfältig vor und tragen Sie hier nicht irgendwelche Wunschwerte ein. Die Werte, die Sie hier angeben, verwendet PicassoMode dazu, Sie davon abzuhalten, dem Monitor mehr zuzumuten, als er verträgt. Zu deutsch: Alle Versuche, bestimmte Frequenzen oder Auflösungen zu erzeugen, die außerhalb dieser hier eingestellten Grenzen liegen, werden fehlschlagen. PicassoMode wacht darüber, daß Ihrem Monitor nur Zeilen- und Bildwiederholfrequenzen zugemutet werden, die er verträgt.

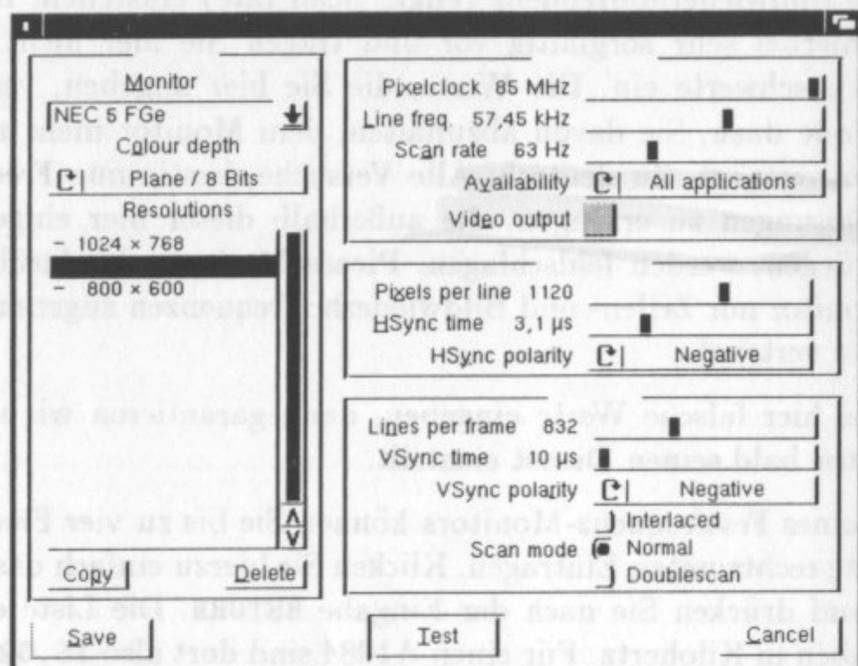
Geben Sie hier falsche Werte eingeben, dann garantieren wir nur, daß Ihr Monitor bald seinen Dienst einstellt.

Im Falle eines Festfrequenz-Monitors können Sie bis zu vier Frequenzen in die Liste rechts unten Eintragen. Klicken Sie hierzu einfach das String-Gadget und drücken Sie nach der Eingabe **RETURN**. Die Liste erwartet die Eingaben in Kilohertz. Für einen A1084 sind dort also **15,625** einzugeben (man beachte das Komma, Punkte werden nicht angenommen!).

Damit ist die Monitor-Konfiguration abgeschlossen. Mit **Save** können Sie Ihre Einstellungen dauerhaft sichern. **Use** ist nur dann zu empfehlen, wenn Sie einen Monitor aus der Liste ausgewählt haben, die vorher schon gesichert wurde. Mit **Cancel** schließlich verlassen Sie das Fenster, ohne das sich etwas geändert hätte.

### 8.1.5. Das Haupt-Einstellungsfenster

Nach dem Start oder nach der Monitor-Konfiguration finden Sie sich immer wieder im Haupt-Einstellungsfenster mit dem Titel **PicassoMode**.



Das Haupt-Einstellungsfenster von PicassoMode

Man kann es grob in drei Bereiche teilen: Unten sind drei Gadgets über die ganze Fensterbreite verteilt, und mit **Save**, **Test** und **Cancel** beschriftet. Mit **Save** kann man alle gemachten Einstellungen sichern und das Programm verlassen. Sollte man jedoch irgendetwas geändert haben, so fragt PicassoMode, ob man den Rechner neu starten möchte. Dies ist auf jeden Fall nötig, wenn man Einstellungen auf Platte gesichert hat. Diese können nicht mehr rückgängig gemacht werden.

**Cancel** läßt sich nur anklicken, wenn man noch nicht gesichert hat. Damit verläßt man PicassoMode, ohne irgendeine Änderung vorgenommen zu haben. **Test** dient dazu, eine Auflösung zu testen, die man sich erzeugt hat. Dazu später mehr.

Links sieht man unter dem Monitor-Namen ein Cycle-Gadget und eine Liste. Das Cycle-Gadget kennt drei Zustände:

Plane / 8 Bits  
15 Bits / 16 Bits  
24 Bits

Je nachdem, welche Stellung das Cycle-Gadget hat, enthält die Liste darunter eine Aufzählung von Auflösungen, die für Bildschirme mit bis zu 256 Farben (8 Bit Farbtiefe), bis zu 65536 Farben (16 Bit Farbtiefe) oder 16.777.216 Farben (24 Bit) zur Verfügung stehen.

Sie können in diese Liste selbst neue Auflösungen eintragen, bestehende ändern oder auch löschen. Je nachdem, welcher Eintrag gerade angewählt ist, positionieren sich die Schieberegler und alle anderen Gadgets auf der rechten Seite des Fensters.

Wenn Sie auf **Copy** klicken, wird der aktuelle, angewählte Eintrag an die Liste als Kopie an das Ende der Liste angehängt; mit **Delete** läßt sich der ausgewählte Eintrag wieder löschen.

Rechts oben, unter **Parameters** lassen sich die **Pixelclock**, **Line frequency** (Zeilenfrequenz) und **Scan rate** (Bildwiederholfrequenz) einstellen. Sie werden schnell merken, daß diese alle voneinander abhängen. Am interessantesten dürfte für Sie die **Scan rate** sein, da man hiermit die Bildwiederholfrequenz festlegt.

Mit dem Cycle-Gadget **Availability** macht man der village.library klar, für welche Applikationen die gewählte Auflösung verfügbar sein soll. Mit **All applications** wird eine Auflösung sowohl auf der Workbench als auch über spezielle Applikationen wie TVPaint verfügbar sein. Mit **Not available** kann niemand auf die ausgewählte Auflösung zugreifen. Bei **Special applications** können nur Programme mit dieser Auflösung arbeiten, die speziell dafür programmiert wurden, auf der Picasso II zu arbeiten. Dazu gehören alle Treiber, Viewer, TVPaint, VDPaint, MainActor, usw.

Unter dem Cycle-Gadget sieht man eine Check-Box mit der Bezeichnung **Video Output**. Dieses Gadget wird erst dann bedienbar, wenn die Zeilenfrequenz nahe 15,625 Kilohertz liegt. Es dient dazu, bei einer bestimmten Auflösung den (optional bei uns erhältlichen) Video-Encoder anzuschalten. Die Auflösung ist dann fest mit dem Video-Encoder gekoppelt. Nutzt man die Auflösung, wird der Video-Encoder eingeschaltet, nutzt man eine andere, bei der dieser Schalter ausgeschaltet ist, wird der Video-Encoder abgeschaltet.

Neben den Frequenzen sind bei der Wahl einer Auflösung natürlich die Screen-Ausmaße in X- und Y-Richtung wichtig. Diese können mit

den Slider-Gadgets **Pixels per Line** (in X-Richtung) und **Lines per Frame** (in Y-Richtung) ausgewählt werden.

Die Regler, die mit **HSync time** und **VSync time** beschriftet sind, lassen Sie am besten in Ruhe, wenn Sie nicht wissen, was Sie damit tun können. Die sind nur für Kenner der Materie von Belang. Im allgemeinen werden Sie von PicassoMode auf die richtigen Werte gesetzt. Sie bestimmen die Länge des Synchronisations-Impulses. Bei Monitoren sind diese im allgemeinen recht kurz, bei Fernsehern recht lang.

Noch von Interesse könnten die Cycle-Gadgets zur VSync- und HSync-Polarität sein. Diese müssen Sie wahrscheinlich nur ändern, wenn Sie einen alten FixedScan-Monitor anschließen. Sollten Ihnen Zweifel kommen, blicken Sie bitte in Ihr Handbuch zu Ihrem Monitor.

Die Radio-Buttons hinter Scan-Mode **Interlaced**, **Normal** und **DoubleScan** sind in den seltensten Fällen wirklich schaltbar und dienen mehr als Anzeige denn als Einstellmöglichkeit.

### 8.1.6. Die Menüs

Von den beiden Menüs ist nur das rechte interessant: Es bietet Standard-Werte für eine feste Zeilenfrequenz, wie man Sie bisher von der *village.library* her kannte.

Project	Pre-set	
Village 15 kHz	1	
Village 31 kHz	2	
Village 35.5 kHz	3	
Village 38 kHz	4	
Village 48 kHz	5	
Village 57 kHz	6	
Village 64 kHz	7	

*Einige Default-Monitor-Werte sind per Menü erreichbar*

Statt einen Monitor zu bestimmen, kann man einfach einen Menüpunkt wie **Village 38kHz** auswählen und damit die Liste auf der linken Seite mit den Auflösungen füllen, die einem zur Verfügung stehen, wenn man im Icon der *village.library*

**MONITOR=38kHz**

eingetragen hätte. Mit dieser Hilfe ist es sehr einfach, für einen gängigen Monitor schnell ein paar Anpassungen zu machen, ohne alle Auflösungen selbst einstellen zu müssen.

### 8.1.7. Das Test-Bild

Mit den bisherigen Kenntnissen sollten kaum noch Schwierigkeiten bestehen, mal einen Testdurchlauf zu machen. Wählen Sie dazu aus dem Menü einen kHz-Eintrag, dessen Wert unter der Maximal-kHz-Größe bleibt, die ihr Monitor noch verträgt (31 kHz ist selten verkehrt, außer bei einem A1084).

Sind die Auflösungsparameter alle gewählt, ist es an der Zeit, sich mit einem Klick auf **Test** oder Drücken von RETURN ein Testbild in der gewählten Auflösung anzeigen zu lassen. Daraufhin wird der Bildschirm kurz dunkel und es erscheint ein Testbild, wie man es von Zeitschriften oder auch einem Fernseh-Apparat kennt.

Was man sehen sollte, ist ein stabiles Bild, mit einem Grau- und Farbverlauf in der Mitte, flankiert von verschiedenen Graumustern (um die Schärfe beurteilen zu können). Darum sieht man einen Kreis, der bei einem korrekten Seitenverhältnis auch rund sein sollte. Den restlichen Bildschirm füllt ein gleichmäßiges Karo-Muster. Den Rand in jeder Richtung markiert ein weißes Rechteck, das jeweils gleichmäßig weiß aussehen sollte.

Nun haben Sie folgende Möglichkeiten:

Pfeil rechts	bewegt den Bildschirm nach rechts
Pfeil links	bewegt den Bildschirm nach links
Pfeil oben	bewegt den Bildschirm nach oben
Pfeil unten	bewegt den Bildschirm nach unten
Shift Pfeil rechts	macht den Bildschirm in der Breite größer
Shift Pfeil links	macht den Bildschirm in der Breite kleiner
Shift Pfeil oben	macht den Bildschirm höher
Shift Pfeil unten	macht den Bildschirm in der Höhe kleiner
RETURN	übernimmt die Einstellungen und schließt den Testschirm
ESC	vergißt alle die Einstellungen und schließt den Testschirm
Mausklick	wie bei ESC

Bitte bedenken Sie, daß die Einstellungen nur dann übernommen werden, wenn Sie RETURN drücken, in allen anderen Fällen nicht.

PicassoMode versucht, unsinnige Bildlagen und -größen zu verhindern. Dies klappt jedoch nicht immer. Zum einen sind Monitore zu schieden, zum anderen kann man nicht die Werte als Begrenzung nehmen, mit der alle Monitore laufen würden. Dann könnte man nämlich gar nichts mehr einstellen.

Sollte das Testbild kaum zu erkennen sein, weil das Bild nur wild flimmert, der Monitor nur einen Pfeifton von sich gibt oder warum auch immer, **drücken Sie bitte sofort ESC oder eine Maustaste!**

Wenn der Monitor pfeift, kommt es auf jede Zehntel-Sekunde an! Der pfeift meist nicht sehr lange. Und dann pfeift er wahrscheinlich nie wieder.

Haben Sie alle Auflösungen zu Ihrer Zufriedenheit geändert, klicken Sie bitte auf **Save**. PicassoMode wird Sie anschließend auffordern, den Rechner neu zu starten, oder aber zum Programm zurückzukehren. Sie können das Programm regulär anschließend nur noch mit einem Reset verlassen.

Bevor Sie dies allerdings tun, sollten Sie im nächsten Kapitel nachlesen, wie eine Einstellung im Icon der village.library zu verändern ist, damit Ihre erstellten Auflösungen überhaupt eingebunden werden.

### 8.1.8. Der Monitor-Eintrag der village.library

Bisher mußte man in den ToolTypes der village.library angeben, welche Maximal-Zeilenfrequenz der angeschlossene Monitor noch verträgt. Beim Booten des Rechners und dem Einbinden der village.library laß diese den Wert aus den Tooltypes und baute sich intern eine Tabelle auf, die alle nötigen Parameter für die Farbtiefen 8, 16 und 24 Bit enthielt. Bei Bit-Tiefen unter 8 wurden die Werte von 8 genommen, bei einer Farbtiefe von 15 Bit wurden die Werte für 16 Bit verwendet.

Die village.library ab Version 2.53 kann eine Datei einlesen, in der Werte für 8, 16 und 24 Bit stehen. Diese Datei kann nur mit PicassoMode erzeugt und verändert werden. Es ist keine Textdatei. Bitte sehen Sie davon ab, in dieser Datei etwas per Hand zu ändern, *weil PicassoMode dies oder jenes nicht zuläßt*.

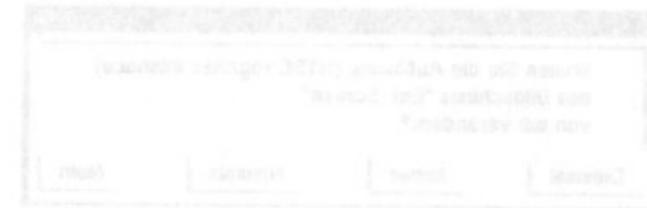
Statt einer Kilohertz-Angabe ist in der village.library und

MONITOR=CUSTOM

einzutragen. Nach dem nächsten Booten sucht die village.library dann nach einem File mit dem Namen

DEVS:Village/<xyz\_bla\_usw>

wobei für <xyz\_bla\_usw> ein sinnvoller Monitorname wie NEC 3D stehen sollte. In dem Verzeichnis DEVS:Village können auch mehrere Dateien stehen. Welche davon genommen wird, darüber entscheidet ein String in der Environment-Variable PicassoMonitor. Diese Variable legt PicassoMode automatisch an, wenn Sie das Programm einmal gestartet und verlassen haben. Sie befindet sich als Datei auch in dem Verzeichnis ENVARC:.



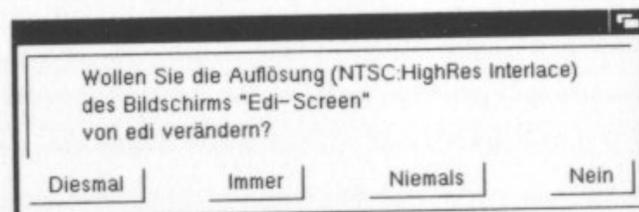
## 8.2. ChangeScreen

Das Commodity **ChangeScreen** erlaubt es Ihnen, auch Programme auf der Picasso II laufen zu lassen, die einen eigenen Screen öffnen und dem Benutzer keine Möglichkeit bieten, die Auflösung und Farbtiefe dieses Screens zu wählen.

Die Installation kopiert **ChangeScreen** nach **Sys:WBStartup**, so daß es normalerweise immer aktiv ist. Wenn Sie es nur gelegentlich benutzen wollen, dann verschieben Sie es am besten in einen anderen Ordner und starten Sie es bei Bedarf mit einem Doppelklick.

### 8.2.1. Funktion

Jedesmal, wenn ein Programm einen eigenen Screen öffnen möchte, erscheint folgender Requester:



Die vier Gadgets (Elemente zum Anklicken) haben dabei folgende Bedeutung:

- Diesmal:** Klickt man auf dieses Gadget, wird der Screen einmalig auf der Picasso II geöffnet. Bevor dies passiert, erscheint noch ein weiterer Requester, der die Auflösung erfragt (siehe unten). Beim nächsten Start des Programms, das wieder einen solchen Screen öffnen möchte, erscheint wieder dieser Requester.
- Immer:** Klickt man auf **Immer**, kann man den zu öffnenden Screen dauerhaft umleiten. Sollte man dieses Programm ein weiteres Mal starten, erscheint dieser Requester nicht mehr. Statt dessen wird der Screen sofort umgeleitet, ohne noch einmal nachzufragen. Wenn man auf **Immer** geklickt hat, erscheint wie bei **Diesmal** ein weiterer Requester (siehe unten).

- Nie:** Klickt man auf **Nie**, wird **ChangeScreen** nie mehr mit diesem Requester erscheinen und fragen, ob er diesen Screen umleiten soll. Das Programm merkt sich anhand des Namens, daß ein Screen mit diesem Namen nicht umgeleitet werden soll und daß es auch nicht mehr danach fragen soll.
- Nein:** Steht für *Nur dieses Mal nicht umleiten*. Beim nächsten Start des Programms erscheint dieser Requester erneut.

Im Fall von **Diesmal** und **Immer** erscheint dann ein weiterer Requester, der wie folgt aussieht:



Sie können nach Belieben eine neue Auflösung wählen. Dies muß nicht unbedingt eine Picasso II-Auflösung sein. Falls Sie den Screen doch nicht umleiten wollten, können Sie die Auswahl mit **Abbruch** verlassen. Möchten Sie den Screen umleiten, so klicken Sie bitte nach der Wahl einer neuen Auflösung auf **Ok**. Sollte der neue Bildschirm kleiner als der angeforderte sein, wird automatisch auf **Auto-Rollen** geschaltet.

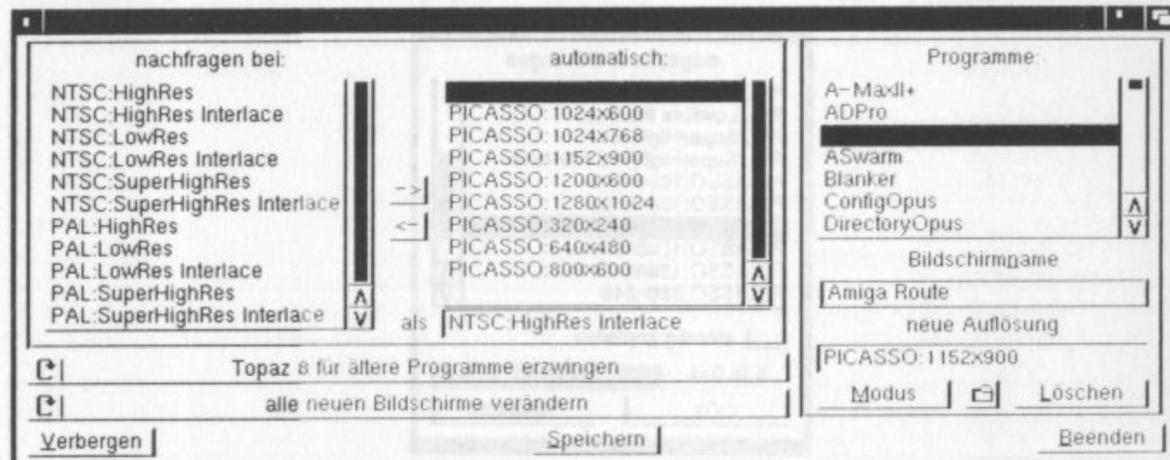
Unter der Liste der Auflösungen finden sich noch zwei weitere Gadgets (Elemente zum Anklicken). Wenn Sie **ständig kopieren** anklicken, wird der neue Bildschirm nicht *wirklich* umgeleitet. Vielmehr wird dann ein Bildschirm im ChipMem im Hintergrund geöffnet und dessen Inhalt regelmäßig in einen Picasso II-Bildschirm kopiert. Dies ist nötig bei Programmen, die direkt in den Bildschirmspeicher schreiben, statt dafür Systemfunktionen zu benutzen. Alte DPaint-Versionen sind Kandidaten

für solch eine Wahl. Auch Animations-Programme gehören oft zu der Kategorie.

Wie oft kopiert werden soll, können Sie über das Zieh-Gadget einstellen. Sie geben dabei die Anzahl von Kopien pro Sekunde an.

### 8.2.2. Das Konfigurationsfenster

Verständlicherweise vertut man sich manchmal, oder ein Programm *verträgt* die Umleitung wider Erwarten doch nicht. Dann möchte man die gemachten Einstellungen ändern. Das Konfigurationsfenster



erlaubt Ihnen, alle Einstellungen zu verändern, die Sie per **Immer** oder **Nie** dauerhaft angeordnet hatten. Darüber hinaus können Sie auch generelle Umleitungen festlegen. Das Fenster öffnet sich, sobald Sie eine spezielle Tastenkombination – voreingestellt ist **CTRL ALT c** – gedrückt haben.

Links oben im Konfigurationsfenster sehen Sie eine Liste mit allen in Ihrem System eingebundenen ScreenModes. Darüber steht **nachfragen bei:**. Wenn ein Programm einen Screen öffnen möchte, der in dieser Liste steht, fragt **ChangeScreen** nach, ob es diesen Screen umleiten soll. Im allgemeinen soll das Programm diese Arbeit ja machen. Bei einigen Screens ist dies aber nicht sinnvoll, zum Beispiel dann, wenn ein Programm einen Screen mit dem ScreenMode **PICASSO:800x600** öffnen möchte. Offensichtlich gibt es hierbei nichts mehr umzuleiten, da der Screen schon auf der Picasso II aufgehen soll.

Für diesen Fall kann man einen ScreenMode von der linken Liste in die mittlere verschieben, über der **automatisch:** steht. Hierzu klicken Sie in der linken Liste den gewünschten ScreenMode an und verschieben ihn, indem Sie auf den Pfeil nach rechts klicken, der sich zwischen den beiden Listen befindet. Möchte ein Programm einen Screen mit einem ScreenMode öffnen, der in der mittleren Liste steht, erfolgt keine Nachfrage mehr.

Die mittlere Liste hat aber neben dieser Funktion noch eine zweite. Unter der Liste finden Sie eine Zeile, in der nach einem Klick auf einen ScreenMode der gleiche ScreenMode noch einmal auftaucht. Dieser ScreenMode ist sozusagen der *Ersatz-ScreenMode*. Jedesmal, wenn der oben in der Liste aufgeführte ScreenMode für einen neuen Bildschirm benutzt werden sollte, wird der dafür angegebene Ersatz-ScreenMode gewählt.

Den Ersatz-ScreenMode können Sie ändern, indem Sie in der Liste auf einen ScreenMode doppelklicken. Es erscheint ein ScreenMode-Requester, in dem Sie einen ScreenMode angeben können, der immer dann gewählt werden soll, wenn ein Programm einen Screen im Ursprungs-ScreenMode öffnen wollte. Sie können zum Beispiel mit dieser Funktion den ScreenMode **NTSC:HighRes Interlace** generell auf **PICASSO:640x480** umleiten. Alle Programme, die einen **NTSC:HighRes Interlace** öffnen wollen, werden diesen dann automatisch auf der Picasso II öffnen.

In der rechten Liste sehen Sie – vielleicht – eine Reihe von Namen der Programme, die Sie bereits mit **Immer** oder **Nie** umgeleitet haben. Darunter ist eine Zeile, die mit **Bildschirmname** überschrieben ist. Hier steht der Titel eines Screens, wenn Sie in der Liste darüber einen Programmnamen ausgewählt haben. Wiederum darunter sehen Sie die Auflösung, die Sie für das ausgewählte Programm und den angegebenen Screen ausgewählt hatten. Hier kann auch **<unverändert>** stehen, falls Sie einmal bei einer Nachfrage zwecks Umleitung auf **Nie** geklickt haben sollten.

Die neue Auflösung können Sie in diesem Fenster jederzeit ändern, indem Sie ein Programm auswählen und auf **Modus** klicken. Es erscheint ein ScreenMode-Requester, in dem alle ScreenModes und der Eintrag **<unverändert>** aufgeführt sind. Wählen Sie einen ScreenMode aus und klicken Sie auf **Ok**.

Selbstverständlich können Sie Einträge in der rechten Liste auch löschen. Wählen Sie dazu einfach einen Eintrag aus und klicken Sie auf **Löschen**.

Zwischen den beiden Gadgets **Ändern** und **Löschen** sehen Sie ein Gadget mit einem Verzeichnis-Symbol. Über dieses können Sie Programme in die rechte Liste aufnehmen, ohne das diese gestartet werden müssen. Nach einem Klick auf das Verzeichnis-Symbol erscheint ein Datei-Requester, in dem Sie ein Programm auswählen können. Da ChangeScreen nicht wissen kann, wie der Screen-Titel lautet, müssen Sie diesen unter der Liste der Programme selbst eintragen. Hierbei können Sie jedoch die vom AmigaDOS bekannten Platzhalter wie **#?** einsetzen. Geben Sie als Titel nur **#?** an, werden alle Screens des angegebenen Programms umgeleitet.

Dieser Platzhalter-Mechanismus ist auch dann sehr praktisch, wenn ein Programm fälschlicherweise seinen Screens keinen Namen gibt und als Screen-Titel stets wechselnde, komische Kürzel auftreten. Setzen Sie in einem solchen Fall **#?** als Screen-Titel ein. Für Programm-Namen gilt das gleiche: Auch hier können Sie die typischen AmigaDOS-Platzhalter verwenden.

Unter den beiden Listen weiter links befinden sich noch zwei Cycle-Gadgets. Das obere kann zwischen den beiden Einstellungen **Topaz 8 für ältere Programme erzwingen** und **Standard-System-Font für alle Programme** hin- und hergeschaltet werden. Viele alte Programme gehen davon aus, daß sie – wenn sie keinen Font anfordern – den Font **Topaz 8** bekommen. Ab Workbench 2.0 ist dies aber nicht unbedingt mehr der Fall, da man die Fonts für die Workbench, Screens und den Rest einstellen kann. Wenn Sie **Topaz 8 für ältere Programme erzwingen** wählen, dann bekommt ein umgeleiteter Screen den Font **Topaz 8** verpaßt, was dazu führen sollte, daß Menüs und einige Texte wieder gut zu lesen sein sollten, selbst wenn Sie als ScreenFont einen viel größeren gewählt haben.

Das zweite Cycle-Gadget kann zwischen **alle neuen Bildschirme verändern** und **nur bei Aufrufen von OpenScreen()** nachfragen hin- und hergeschaltet werden. In der ersten Einstellung wird bei allen Versuchen, einen Screen zu öffnen, nachgefragt; bei der zweiten Einstellung nur dann, wenn das Öffnen über die alte Funktion **OpenScreen** und nicht über die seit Workbench 2.0 verfügbare **OpenScreenTagList**

erfolgt. Das hat dann zur Folge, daß nur bei alten Programmen nachgefragt wird, ob ein Screen umgeleitet werden soll.

### 8.2.3. Menüs

Alle Einstellungen können Sie laden oder auch speichern. Dies geschieht über die Menü-Punkte **Projekt/Öffnen ...** und **Projekt/Speichern als ...**. Wenn Sie einen dieser Menü-Punkte anwählen, erscheint ein Datei-Requester, in welchem Sie eine Datei auswählen können.

Normalerweise speichert **ChangeScreen** seine Einstellungen in der Datei **ENVARC:ChangeScreen.prefs**. Dies passiert zum Beispiel automatisch, wenn Sie beim Umleiten **Nie** oder **Immer** ausgewählt oder im Fenster auf das Gadget **Speichern** geklickt haben. Die Einstellungen werden dort als einfache Texte vermerkt, die Sie auch mit einem Text-Editor verändern können. Dies ist aber nicht zu empfehlen, da dies zu Inkonsistenzen führen kann und **ChangeScreen** die Datei später eventuell nicht lädt.

Aus diesem Grund wird der Aufbau der Datei auch nicht näher beschrieben, da er sich in späteren Versionen durchaus noch ändern kann.

**Über ...** bringt ein Fenster auf den Bildschirm, dem Sie den Namen des Autors, das Erstellungsdatum und eine Versionsnummer des Programms entnehmen können. Dies ist für den Fall interessant, daß Sie uns einen Fehler mitteilen möchten oder aber feststellen wollen, ob Sie die neueste Version des Programms besitzen.

**Verbergen** schließt das Fenster, ohne das Programm zu beenden. Da es als Commodity läuft, können Sie es jederzeit mit der Tastenkombination **CTRL ALT c** öffnen. Diese Kombination können Sie auch ändern (siehe dazu den nächsten Abschnitt).

Mit **Beenden** schließen Sie nicht nur das Fenster, sondern beenden auch das Programm.

Im nächsten Menü gibt es nur zwei Menüpunkte: **Bildschirmmodi neu lesen** dient dazu, die im Betriebssystem gespeicherte Liste aller ScreenModes noch einmal einzulesen. Dies braucht man, nachdem man ein Monitor-File zwischendurch aktiviert hat. **Piktogramme erzeugen** schließlich regelt, ob für die Datei mit den Einstellungen auch ein Icon erzeugt werden soll.

### 8.2.4. ToolTypes

ChangeScreen läßt sich über die Workbench oder auch in einer Shell starten. Hierbei kennt es diverse Parameter, die sich jeweils als ToolTypes oder Shell-Parameter angeben lassen.

Die Parameter sind

SETTINGS	Name der Konfigurationsdatei
NOICON	Konfigurationsdateien ohne Icons abspeichern
ONLYOLDSTYLE	ob nur OpenScreen() – Aufrufe gepatched werden
FORCETOPAZ	alte Programme bekommen Topaz 8
CX_PRIORITY	zum Angeben der Commodity-Priorität
CX_POPKEY	Kommandotaste, mit der das Einstellfenster geöffnet wird.
CX_POPUP	um das Einstellfenster beim Starten sofort zu öffnen

Hinter SETTINGS können Sie einen Dateinamen angeben, in welchem nach den Einstellungen gesucht werden soll. Sie können auf diese Art mehrere Dateien mit Einstellungen benutzen und auch im Betrieb die Datei wechseln.

Normalerweise speichert ChangeScreen seine Konfigurationsdateien immer mit Icon ab. Soll es dieses nicht tun, so ist der Parameter NOICON anzugeben.

Sollen nur alte OpenScreen()-Aufrufe umgeleitet werden, so ist die Option ONLYOLDSTYLE anzugeben. Diese macht zusammen mit FORCETOPAZ Sinn. Sie führt dazu, daß beim Öffnen des Screens der Screenfont auf Topaz 8 gesetzt wird, unabhängig davon, was Sie als Screen-Font im Font-Preference-Tool gewählt haben.

CX\_PRIORITY, CX\_POPKEY und CX\_POPUP schließlich sind die Standard-ToolTypes, die jedes Commodity versteht (siehe dazu die von Commodore gelieferten Handbücher).

### 8.2.5. Interaktive Hilfe

Sollten Ihnen diese Erklärungen nicht ausreichen oder Sie dieses Handbuch irgendwann einmal verlegen, dann können Sie sich trotzdem immer eine Hilfe geben lassen. Ein Druck auf die Taste HELP reicht aus und es

öffnet sich ein neues Fenster des Programms AmigaGuide, in welchem Sie sich weitere Informationen holen können.

Eine Hilfe zur Benutzung von AmigaGuide erhalten Sie, wenn Sie nach dessen Start einfach auf HELP drücken. Da es sich komplett mit der Maus und über Menüs steuern läßt, sollten Sie keine Probleme damit haben.

### 8.2.6. Warnung

**Wichtig: ChangeScreen ist zwar sehr nützlich, aber nicht unbedingt die Lösung für alle Fälle. Es kann nicht dafür sorgen, daß alle Programme fehlerlos auf der Picasso II arbeiten.**

Viele Programme erwarten, daß sie genau den Screen erhalten, den sie öffnen möchten – und nicht einen, der deutlich größer ist. Sie müssen selbst ausprobieren, ob ein Umleiten erfolgreich ist oder nicht. Funktioniert die einfache Umleitung nicht, so sollten Sie den Kopier-Modus ausprobieren. Es kann aber auch dabei passieren, daß der Amiga abstürzt oder hängenbleibt. Für Schäden übernimmt die Village Tronic Marketing GmbH keine Haftung.

### 8.3. PicassoSwitch

Manchmal kann es sinnvoll sein, zwischen der Amiga-Grafik und der Picasso II-Grafik hin und herschalten zu können. Im allgemeinen ist dies nicht nötig, da der Intuition-Treiber die Umschaltung automatisch vornimmt.

Zu Testzwecken oder nach dem Absturz eines Programms kann dies nötig sein. Für diesen Fall gibt es das Programm **PicassoSwitch**. Es ist ein Commodity und bietet dem Benutzer drei Funktionen an, die man per Tasten aktivieren kann. Dies sind:

Zur Picasso II (SHIFT AMIGA P) schaltet auf die Picasso II-Grafik  
 Zum Amiga (SHIFT AMIGA A) schaltet auf die Amiga-Grafik  
 Wechseln (SHIFT AMIGA T) schaltet um auf die jeweils andere Grafik

Diese Einstellungen lassen sich natürlich ändern. Sie können dies über die Workbench erreichen, in dem Sie auf das Icon klicken und anschließend AMIGA-I drücken. Wenn das Programm schon läuft, können Sie ein Konfigurationsfenster mit dem Commodity **Exchange** erscheinen lassen. Das Konfigurationsfenster sieht so aus:

#### *Das Konfigurationsfenster von PicassoSwitch*

Die in diesem Fenster gemachten Einstellungen sind aber nur vorübergehend und gehen nach dem Ausschalten des Rechners verloren. Dauerhaft können Sie die Einstellungen nur im Icon von **PicassoSwitch** ändern.

### 8.4. PicassoPhoto

Normalerweise kann man mit diversen Programmen ein sogenanntes SnapShot vom Bildschirm machen. Dabei wird aus dem Screen eine IFF-Datei erzeugt, die man mit einem IFF-Anzeige-Programm wieder ansehen kann.

Solange man nur mit dem Intuition-Treiber und normalen Intuition-Screens arbeitet, erledigt diese Aufgabe jedes x-beliebige Snapshot-Programm wie zum Beispiel **GraphicDump** von Commodore.

Bei den Screens der Picasso II, die man mit den Anzeige-Programmen ViewIFF, ViewGIF, ViewJPEG und den Treibern zu diversen Bildverarbeitungsprogrammen öffnen kann, geht das aber nicht. Diese sind gänzlich anders aufgebaut. Normale Snapshot-Programme werden es einerseits ablehnen, von solch einem Screen eine Datei zu erzeugen, andere werden nur Unsinn speichern oder abstürzen.

Für solche Fälle, in denen man von einem TVPaint-Screen ein Abbild braucht, um einem Freund etwas zu zeigen oder einen Tip zu geben oder auch um sich vom Macintosh-Emulator A-MaxIV-Color eine schöne Grafik zu schnappen, gibt es **PicassoPhoto**. Es erzeugt Bilddateien im IFF-Format von Picasso-Screens in 8, 15, 16 und 24 Bit Farbtiefe.

Um Frustrationen vorzubeugen: **PicassoPhoto** macht **nur** von **Nicht-Intuition-Screens** Abzüge. Wenn Sie einen normalen Intuition-Screen festhalten wollen, so **müssen** Sie dazu ein anderes Programm verwenden.

#### 8.4.1. Aufruf vom CLI

Das Programm kann sowohl vom CLI als auch von der Workbench aus gestartet werden. Vom CLI aus kann man mehrere Optionen angeben. Nach der Eingabe von

```
PicassoPhoto ?
```

erscheint die folgende Parameter-Schablone wie bei Commodore-Programmen üblich:

```
FILENAME/A,D=DELAY/N,V=VERBOSE/S:
```

Für **FILENAME** muß man einen Dateinamen angeben. In die so spezifizierte Datei wird das Bild gespeichert. Gibt man keine weiteren Optionen

an, erfolgt die Speicherung sofort nach dem Start. Da beim Start vom CLI wohl meist ein Workbench-Screen der vorderste Screen ist, sollte man noch eine Wartezeit einlegen, um den abzuspeichernden Screen in der Vordergrund zu holen. Hierfür gibt es die Option **DELAY**. Hinter ihr kann man eine Wartezeit in Sekunden angeben. Ein Beispiel

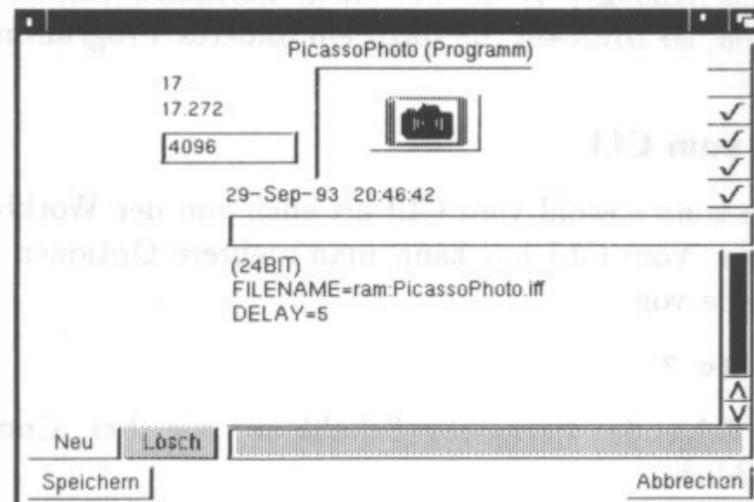
```
PicassoPhoto ram:TestBild DELAY 20
```

Nach 20 Sekunden wird in der Datei `ram:TestBild` eine Kopie des vordersten Screens geschrieben. Bis zu dem Zeitpunkt sollten Sie keine Veränderungen an dem Screen vornehmen. Nach dem Speichern sollte der Bildschirm kurz aufblitzen. Bei 15/16- und 24-Bit-Screens ist dies meist aber nicht sichtbar.

Mit der Option **VERBOSE** kann man dem Programm mitteilen, daß man gerne gewußt hätte, wie die Datei heißt, in der er das Bild ablegt und wie tief (Farbtiefe) der Screen ist, von dem ein Photo geschossen wurde.

#### 8.4.2. Aufruf von der Workbench

Der Aufruf von der Workbench ist denkbar einfach. Ein Doppelklick genügt. Die Wartezeit sowie der Name der Datei kann über ein Tooltype eingestellt werden.



Die Tooltypes von PicassoPhoto

Die beiden möglichen Tooltypes (Merkmale) heißen **FILENAME** und **DELAY**. Wird für **FILENAME** nichts angegeben, wählt PicassoPhoto den Namen `ram:picassophoto.iff` als Dateinamen. Als **DELAY** sind 5 Sekun-

den voreingestellt, wenn man nichts an den Tooltypes seit der Installation geändert hat. Für den Parameter **VERBOSE**, den man beim Aufruf vom CLI aus angeben kann, gibt es kein entsprechendes Gegenstück.

#### 8.4.3. Tips für den Umgang

In den neueren Betriebssystemversionen von Commodore (ab 2.1) kann man mit dem Sound getauften Voreinsteller einstellen, daß bei jedem DisplayBeep – dem Aufblitzen eines Screens – ein Ton abgespielt werden soll. So ist es einfacher festzustellen, wann das Sichern eines Screens abgeschlossen ist.

Man kann dieses Programm auch dazu verwenden, Graphiken in das IFF-Format umzuwandeln, denn es lassen sich auch Bilder sichern, die mit dem ViewJPEG oder ViewGIF angezeigt werden.

Die Viewer reagieren auf jeden Mausklick und jede Taste, was das Starten von PicassoPhoto etwas erschwert: vor dem Starten muß man irgendeine Taste drücken. Also startet man am besten vor dem Anzeigen PicassoPhoto mit `run` und einer ordentlichen Wartezeit. Ein Beispiel:

```
run PicassoPhoto ram:testfile DELAY=60
```

Mit 60 Sekunden sollte man in den allermeisten Fällen hinkommen. Sonst wählen Sie die Wartezeit einfach etwas größer.

## 8.5. StyxBlank

Zur Picasso II liefern wir als Bildschirmschoner ein Programm mit aus, das nach einer gewissen Zeit den Bildschirm schwarz schaltet und ein sehr buntes Linienbüschel über den Bildschirm wandern läßt. Dies passiert allerdings nur, wenn Sie eine einstellbare Zeit lang weder eine Taste gedrückt, noch die Maus bewegt oder eine Diskette eingelegt oder entnommen haben.

Der Sinn des ganzen liegt darin, den Monitor vor einem *Einbrennen* zu schützen. Oft sitzt man immer wieder vor dem gleichen Bild – so passiert es im Laufe der Jahre, das sich dieses Bild quasi in den Bildschirm eingraviert. Das liegt daran, daß ein Monitor einer gewissen Alterung unterworfen ist. Die Elemente in der Mattscheibe leuchten jeweils kurz auf, wenn der Elektronenstrahl sie trifft. Nach mehreren Millionen von Durchläufen wird das Aufleuchten immer schwächer, so daß einige Partien des Schirmes, die immer helle Flächen oder Zeichenketten darstellen, mit der Zeit immer schwächer leuchten als andere.

Um diese Alterung ein wenig hinauszuzögern, hat man sogenannte Bildschirmschoner erfunden, die den Bildschirm nach einer gewissen Wartezeit einfach dunkel schalten. Damit man aber nun nicht meint, der Monitor wäre aus, läßt man etwas Buntes, sich Bewegendes über den Schirm huschen.

Die größte Fläche ist damit dunkelgeschaltet und nur ein kleiner Teil zeigt etwas an. Da die Grafik sich bewegt, kann es auch nicht zum Einbrennen spezifischer Partien auf dem Bildschirm kommen.

Als ganz einfaches Programm dieser Art liefern wir **StyxBlank** mit aus. Es kennt drei Parameter, die man über die ToolTypes im Icon des Programms oder über die Tastatur angeben kann.

Dies sind

<b>LINES</b>	Anzahl der Linien auf dem Schirm
<b>SPEED</b>	Wartezeit zwischen dem Zeichnen zweier Linien in 1000stel Sekunden
<b>BLANKTIME</b>	Zeit in Sekunden, bis der Blanker den Bildschirm dunkel schaltet.

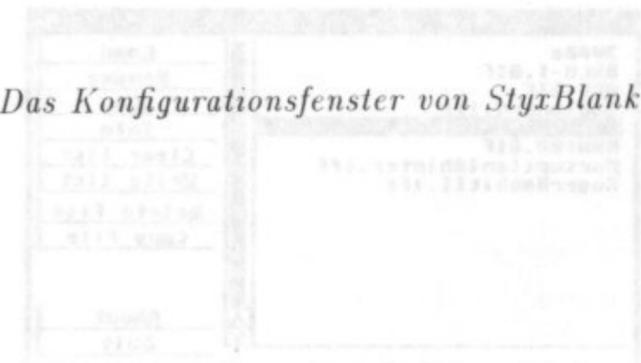
Ein Beispielaufruf könnte wie folgt aussehen:

```
run styxblank LINES=200 SPEED=2 BLANKTIME=120
```

Nach 2 Minuten (=120 Sekunden) des Nichtstuns erscheint ein schwarzer Bildschirm, auf dem anschließend 200 Linien herumhuschen, die im Abstand von 2000 tausendstel Sekunden gezeichnet werden.

Da StyxBlank ein Commodity ist, kann es mittels des Steuerungsprogramms **Exchange** aktiviert, deaktiviert und beendet werden.

Außerdem kann man temporär die Einstellungen über ein Fenster verändern, das sich per Tastendruck oder über **Exchange** öffnen läßt:



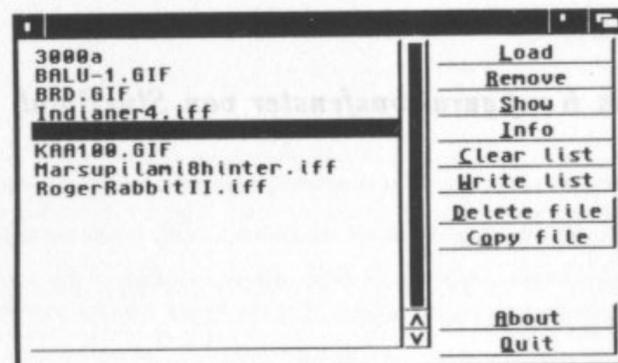
Das Konfigurationsfenster von StyxBlank

## 8.6. IntuiView

IntuiView ist eigentlich kein eigenständiges Programm, es kann nur in Verbindung mit anderen Programmen richtig funktionieren. Es dient dazu, Bilder, Texte, Archive und was es sonst noch in verschiedenen Formaten gibt, bequem von der Workbench aus anzusehen. Dazu verfügt es über ein leicht zu konfigurierendes Analysesystem, daß Sie beliebig erweitern können.

Ursprünglich war IntuiView dazu gedacht, CLI-Unkundigen den Umgang mit den Bildanzeige-Programmen zu erleichtern. Dank des flexiblen Konzepts können Sie IntuiView auch für andere Zwecke einsetzen.

Nach dem Start des Programms von der Workbench oder einer Shell heraus sehen Sie folgendes Fenster vor sich:



IntuiViews Fenster, in das man auch etwas werfen kann

Links ist ein sogenanntes ListView zu erkennen, indem Dateinamen stehen (können). Rechts daneben sieht man verschiedene sogenannte *Gadgets* (rechteckige Boxen, auf die man klicken kann), die Aktionen starten.

### 8.6.1. Die Bedienung

Um eine Datei anzusehen – egal, welchen Inhalt Sie nun gerade hat –, klicken Sie einfach auf den Dateinamen links in der Liste und anschließend auf **Show** (ein Doppelklick bewirkt das gleiche). IntuiView versucht darauf hin herauszubekommen, was für eine Art von Datei die angeklickte ist und startet dann ein Programm, welches die Datei anzeigt.

Es gibt verschiedene Möglichkeiten einen Dateinamen in die Liste einzufügen:

- Sie klicken auf **Load** und es erscheint ein Datei-Requester, in welchem Sie eine Datei auswählen können.
- Sie klicken auf der Workbench auf das Icon einer Datei, halten die Maus gedrückt, ziehen das Icon mit der Maus über das Fenster von IntuiView und lassen es dort fallen. Sie können auch gleich mehrere Dateinamen auf einmal anklicken und in das Fenster von IntuiView ziehen.
- Sie wählen **Load** im Menü aus, worauf wiederum ein Datei-Requester erscheint. Diese Funktion können Sie auch über die Tastenkombination **AMIGA 1** erreichen.
- Sie drücken einfach auf **1** und schon erscheint der Datei-Requester.

Um einen Dateinamen aus der Liste zu löschen, klicken Sie auf **Remove** oder Drücken Sie die Taste **r**. Dies funktioniert jedoch nur, wenn Sie vorher einen Eintrag in der Liste ausgewählt haben. Mit **Remove** löschen Sie aber nur den Eintrag in der Liste, nicht die Datei auf der Festplatte oder Diskette.

**Info** kann Ihnen ein paar nützliche Informationen zu der ausgewählten Datei zeigen. Hierzu gehört der komplette Dateiname, die Länge der Datei, das Datum der letzten Änderung und die Dateiattribute.

Der Button **Clear List** hat eine ähnliche Aufgabe wie **Remove**, nur das nach einem Klick alle Einträge aus der Liste entfernt werden. Wiederum wird keine Datei auf der Festplatte oder einer Diskette gelöscht.

Möchten Sie die aktuelle Liste speichern, so klicken Sie einfach auf das Gadget **Write List**. IntuiView versucht dann die Liste in **EN-VARC:Intuiview.list** zu speichern. Soll die Liste unter einem anderen Namen gespeichert werden, können Sie den Namen beim Aufruf oder über ein Tooltype (siehe weiter unten in diesem Text) ändern.

Einzig gefährlich ist der Klick auf **Delete file**. Hat man einen Eintrag in der Liste ausgewählt und klickt man auf **Delete file**, dann fragt IntuiView noch einmal nach, ob die ausgewählte Datei wirklich gelöscht werden soll und löscht nach der positiven Bestätigung die Datei und ein eventuell vorhandenes Icon. Bei dieser Funktion ist also Vorsicht geboten.

Sollte Ihnen ein Bild so gut gefallen, daß Sie es sich kopieren oder auf eine Diskette sichern möchten, dann geht dies per **Copy file**. Hierzu ist allerdings das Programm **Copy** im Verzeichnis **C:** nötig, da es beim Kopieren aufgerufen wird.

**About** bringt ein Fenster auf den Bildschirm, dem Sie den Autor des Programms, das Erstellungsdatum und eine Versionsnummer entnehmen können. Dies ist für den Fall interessant, daß Sie uns einen Fehler mitteilen möchten oder aber feststellen wollen, ob Sie die neuste Version des Programms besitzen.

Mit **Quit** – wer hätte es gedacht – kann man das Programm beenden.

Statt einen Eintrag auszuwählen und dann auf **Show** zu klicken, können Sie den Eintrag in der Liste auch doppelt anklicken. Dies hat den gleichen Effekt, als hätten Sie auf **Show** geklickt.

Sollten Sie kein Freund der Maus sein, so können Sie die Einträge in der Liste auch mit der Tastatur auswählen. Benutzen Sie einfach die Pfeil-Hoch- und Pfeil-Runter-Tasten. Mit einem Tastendruck auf **RETURN** (die Taste über der rechten Umschalttaste) wird die Datei angezeigt.

Mit der Taste **F1** ist es möglich, das Fenster zu *zippen*, also zwischen zwei verschiedenen Positionen und Fenstergrößen hin- und herzuschalten.

### 8.6.2. Die Menüs

**IntuiView** besitzt ein Menü, das bis auf einen Ausnahme Aktionen auslösen kann, die auch über die Gadgets erreichbar sind.

Die Ausnahme ist **Reload List**. Damit wird die gespeicherte Liste aus der eingestellten Datei (normalerweise **ENVARC:Intuiview.list**) der aktuellen Liste hinzugefügt.

Diese Funktion ist nur in selten Fällen gefragt (deswegen kann man sie nur über das Menü erreichen). Sie ist dann nützlich, wenn man kurzzeitig mit **Clear List** die aktuelle Liste löschen möchte, ein paar Aktionen startet und anschließend wieder die *normale* Liste laden möchte.

### 8.6.3. Die Konfiguration

Woher weiß nun **IntuiView**, welche Datei welchen Typs ist und was für ein Programm zum Anzeigen zu starten ist? Dazu benötigt das

Programm eine Konfigurationsdatei, in der die nötigen Informationen stehen müssen. Die Konfigurations-Datei ist normalerweise in **ENVARC:Intuiview.prefs** zu finden und kann mit jedem Texteditor verändert und ergänzt werden.

Als Beispiel für weitere Erklärungen folgen ein paar Zeilen, aus denen eine solche Datei bestehen könnte:

```
IFF: // dient nur als Kommentar
#?,FORM????ILBM#?,bin:viewiff %s C
#?,FORM????8SVX#?,bin:sound %s
```

```
GIF: // da gibt es zwei Formate
#?,GIF87a#?,bin:ViewGIF.dcc %s C
#?,GIF89a#?,bin:ViewGIF %s C
```

```
JPEG: // dafür haben wir auch ein Viewer
#?,??????JFIF#?,bin:viewJPEG %s
```

```
MPEG: // etwas zum Abspielen
#?.mpg,#?,bin:playMPEG %s
#?.mpeg,#?,bin:playMPEG %s
#?.play,#?,bin:play %s
```

```
TeX: // man kann das ja auch angucken
#?.dvi,#?,tex:bin/ShowDVI %s
```

```
Archive: // auch für LHA-Dateien geeignet
#?.(lha|lzh),#?,bin:lha >t:temp v %s,bin:most t:temp
#?.zoo,#?,progs:bin/zoo >t:temp v %s,bin:most t:temp
```

```
Exec: // Programme lassen sich auch starten
#?,??'03'F3#?,run %s
```

```
Default: // wenn gar nichts mehr paßt, ist es wohl Text
#?,#?,bin:most %s
```

Leerzeilen und solche Zeilen, die weniger als zwei Kommas enthalten, werden als Kommentar betrachtet. Sie können also reichlich Erklärungen

in die Datei einfügen, solange die Zeilen nicht mehr als ein Komma enthalten.

Eine Zeile mit zwei oder mehr Kommas läßt sich in drei Bereiche unterteilen:

```
<file-pattern>,<byte-pattern>,<action>,<action>,...
```

Das `file-pattern` ist noch am einfachsten zu verstehen. Viele Dateien kann man an der Datei-Endung erkennen. So enden LHA-Dateien meist mit einem `.lha` oder `.lzh`. Wenn TeX eine Datei übersetzt hat, hat diese immer die Endung `.dvi`. Wenn dies soweit eindeutig ist, kann man für das zweite Pattern einfach die gewohnte Kombination für *alles* `#?` einsetzen und muß dann nur noch angeben, was mit einer Datei zu geschehen hat, die an der Endung erkannt worden ist. Im Falle der TeX-Datei soll sie mit dem Programm `ShowDVI` angezeigt werden, das sich im Verzeichnis `TeX:bin` befindet. Der Dateiname wird für die Zeichenfolge `%s` eingesetzt.

Möchte man also eine Datei ansehen, die `Hallo.dvi` heißt, dann wird an der Endung erkannt, daß es sich um eine DVI-Datei handelt und die Aktion gestartet. Aus der Zeile

```
TeX:bin/ShowDVI %s
```

wird dann

```
TeX:bin/ShowDVI Hallo.dvi
```

Nun kann es aber sein, daß man Dateien nicht an der Endung erkennen kann. So haben IFF-Bilder meist keine eindeutige Kennung am Schluß des Namens. Mal enden sie auf `.pic`, mal auf `.iff` oder haben gar keine Endung. Für diese Fälle kann man das `byte-pattern` einsetzen. Hier gibt man eine Kombination von Buchstaben, Zahlen und Platzhaltern an, die mit den ersten 80 Bytes der zu testenden Datei verglichen werden.

Beispielsweise fangen alle IFF-ILBM-Dateien mit den Buchstaben `FORM` und `ILBM` an. Zwischen diesen beiden Zeichenketten sind vier Bytes, die für uns ohne Bedeutung und je nach Dateilänge unterschiedlich sind. Wie man dem Beispiel entnehmen kann, kann man also mit

```
#?,FORM????ILBM#?,BIN:viewiff %s C
```

eine IFF-ILBM-Datei sicher erkennen, egal was für eine Dateinamen-endung sie hat. Da dies keine Rolle spielt, gibt man vor dem ersten Komma auch ein `#?` an.

Die Aktion, die nach dem Erkennen auszuführen ist, folgt wie üblich nach dem zweiten Komma. Manchmal kann es nötig sein, eine Aktion in mehreren Schritten auszuführen. Wie dies geht, kann man dem Beispiel bei den Zeilen für die Archive sehen: Einfach weitere Kommandos durch Komma getrennt aufführen. Sie können so viele Kommandos angeben, wie sie möchten. Pro Kommando können Sie den Platzhalter `%s` bis zu fünfmal einsetzen.

Wie man dem Pattern unter `Exec` entnehmen kann, lassen sich statt Buchstaben auch Byte-Werte angeben. Diese hat man mit einem Rückwärtshäkchen anzugeben wie in

```
#?,??'03'F3#?,run %s
```

Diese Zeile bedeutet, daß Datei-Endungen egal sind, die ersten zwei Bytes in einer Datei keine Rolle spielen und dann zwei Bytes folgen sollten, die die Werte `$3` und `$F3` haben. Mit dieser Kombination beginnen alle ausführbaren Programme. `IntuiView` läßt sich also auch zum Programmstarten benutzen. Einfach hineinwerfen und doppelt draufklicken oder `RETURN` drücken.

#### 8.6.4. ToolTypes

Für den Fall, daß Sie die Konfigurations-Datei nicht in `ENVARC:` speichern möchten, können Sie dies auch ändern. Hierzu gibt es im Icon von `IntuiView` ein sogenanntes `ToolType` mit Namen `CLASSFILE`. Ein möglicher Eintrag wäre:

```
CLASSFILE=S:meine_Definitionen
```

Sie können sich auch zwei Icons anlegen, je eins mit verschiedenen Konfigurations-Dateien; eins zum Angucken von Dateien und eins zum Verändern mit dem jeweils passenden Programm. Lassen Sie Ihrer Fantasie freien Lauf.

Ein zweites `ToolType` trägt den Namen

```
WINDOWTITLE=Mein Anzeigefenster
```

Damit kann man den Fenstertitel festlegen und so verschiedene IntuiViews auseinanderhalten.

IntuiView speichert die Liste normalerweise automatisch, wenn Sie das Programm beenden und liest die Liste automatisch ein, wenn Sie das Programm starten. Dies Verhalten läßt sich (in Grenzen) beeinflussen. Zum einen können Sie den Dateinamen ändern, falls Ihnen die Vorgabe von `ENVARC:Intuiview.list` nicht zusagt:

```
INPUTFILE=S:meine_Bilderliste
```

Außerdem können Sie IntuiView verbieten, beim Verlassen des Programms die Liste zu schreiben. Die passende Option lautet:

```
DONTWRITEONEXIT=TRUE
```

Statt `TRUE` können Sie auch `FALSE` einsetzen. Dann wird die Liste beim Verlassen geschrieben.

Um IntuiView gleich nach dem Start an der richtigen Stelle auf dem Bildschirm erscheinen zu lassen, können Sie festlegen, wo das Fenster erscheinen soll, wo die Position im verkleinerten Zustand liegt und ob das Fenster in normaler Größe oder gleich verkleinert erscheint. Die Optionen dazu sind:

```
WINDOWPOS=xxx,yyy ZOOMPOS=xxx,yyy ZOOM=TRUE
```

Für `xxx` und `yyy` sind Koordinaten wie `200,200` anzugeben. Sie die Angaben größer als der Bildschirm, auf dem IntuiView erscheinen soll, so wird der größtmögliche Wert gewählt.

Da Sie nicht nur einzelne Dateien, sondern auch Ordner in das Fenster von IntuiView werfen können, ist eine Option notwendig, die regelt, ob auch die Inhalte von Ordnern in diesem Ordner mit in die Liste aufgenommen werden sollen. Die Option lautet:

```
RECURSION=TRUE
```

Wenn Sie das nicht wünschen, so sollten Sie dort

```
RECURSION=FALSE
```

eintragen.

### 8.6.5. Aufruf von der Shell

Sie können IntuiView nicht nur über die Workbench starten, sondern auch in einer Shell. Wenn Sie das Programm mit einem Fragezeichen als Parameter starten, erhalten Sie als Ausgabe folgende Zeile:

```
FILES/M,C=CLASSFILE/K,T=WINDOWTITLE/K,I=INPUTFILE/K,  
RECURSION/S,NOW/S
```

Die Optionen `C=CLASSFILE/K`, `T=WINDOWTITLE/K`, `I=INPUTFILE/K`, `RECURSION/S` entsprechen den gleichlautenden Tooltypes.

Der Parameter `FILES` steht für eine beliebige Anzahl von Dateinamen und Platzhaltern (englisch *Pattern*), die direkt nach dem Start in der Liste auftauchen sollen. Diese werden nach dem Start nicht sofort angezeigt, sondern nur in die Liste übernommen.

Ein Beispiel:

```
IntuiView #?.c readme CLASSFILE s:Klassen T=Test
```

Nach dem Aufruf befinden sich alle Dateinamen, die auf `.c` enden sowie `readme` in der Liste. Als Konfigurations-Datei wird die Datei `s:Klassen` benutzt, das Fenster hat den Titel `Test`.

Die Option `NOW/S` macht das Programm zu einem vollwertigen Bildanzeigeprogramm, das alle Formate kennt. Statt ein Fenster zu öffnen und die Dateien in die Liste zu übernehmen, öffnet es kein Fenster und zeigt die Bilder nacheinander an. Dazu untersucht es wie gehabt jede Datei und benutzt zur Anzeige das passende Programm.

Um nicht viel tippen zu müssen, empfiehlt es sich, ein Alias einzurichten, wie zum Beispiel folgendes:

```
Alias IV Work:Picasso-II/Viewer/Intuiview [] NOW
```

### 8.6.6. Besonderheiten

Richtig nützlich ist IntuiView nur, wenn man es ständig offen oder auch im Zoom-Zustand auf der Workbench liegen hat. Um Ihnen möglichst viel abzunehmen, sind zwei Besonderheiten eingebaut.

Ist das Fenster des Programms verkleinert, und werfen Sie ein Programm-Icon auf dieses Fenster, so wechselt es in die normale Größe,

aktiviert das eigene Fenster, aktualisiert die Liste mit dem neuen Eintrag und wählt diesen automatisch an. Sie brauchen nur noch auf **Show** zu klicken oder **RETURN** zu drücken, um sich die Datei anzusehen.

Die zweite Besonderheit macht sich bemerkbar, wenn Sie die Auflösung der Workbench wechseln wollen und dabei *vergessen*, **IntuiView** zu beenden. Das Programm merkt selbst, daß sich die Größe der Workbench ändern soll, schließt sein Fenster, wartet darauf, daß ein neuer Workbench-Screen geöffnet wird und öffnet danach sein Fenster.

Es wird wenige Fälle geben, in denen diese Automatik nicht funktioniert. Normalerweise treten diese Fälle aber nicht auf.

## 8.7. IntuiSpeed

„Vertrauen ist gut, Kontrolle ist besser“ soll Lenin einmal gesagt haben. Egal, wem man diesen Spruch nun zubilligt, etwas Wahres ist schon dran. Nachdem die Picasso II das Prototypenstadium erreicht hatte, wollten wir genau wissen, wie schnell wir denn mit unserer Karte sind, vor allem im Vergleich zur eingebauten Grafik, aber auch im Vergleich zu anderen Karten.

Aus diesem Grund haben wir das Programm **IntuiSpeed** erstellt, das die Ausführungszeiten für diverse grafische Operationen mißt. Nach dem Start des Programms erscheint ein Fenster – das je nach gewählten Default-Font – etwa so aussehen könnte:

Computer: A3000, 25 MHz, 8 MByte Fast, Picasso II			
Write Pixel	63541	Open/Close Window	48
Draw Lines	14404	Change Window size	119
Fill Rectangle	9364	Move Window	631
Scroll Vertical	1073	Lines Pattern	1795
Scroll Horizontal	1060	...	0
Draw Circle	371	...	0
Text Without Scroll.	7341	...	0
Draw Rectangle	8007	...	0
ScreenMode: PICASSO:800x600			
Colors: 256			
Start all	Save	Print	Show Graphic

Intuispeed nach einem Testdurchlauf

Ganz oben sehen Sie ein String-Gadget hinter dem Text **Computer**, in welches Sie einen Text wie **Mein A3000, 43 MHz, 7.5 MByte RAM, handoptimiert mit Picasso II** eintragen können. Spätestens danach müssen Sie vor allen anderen Aktionen einen Bildschirmmodus auswählen.

Ist das geschehen, werden alle Gadgets angeschaltet, so daß Sie die Tests starten können. Jeder Test läuft exakt 10 Sekunden. In dieser Zeit wird gezählt, wie oft die zu testende Aktion abgelaufen ist. Der Wert erscheint nach dem Test in der Box hinter dem jeweiligen Gadget.

Wenn Sie alle möglichen Tests nacheinander durchführen möchten, können Sie auch links unten auf **Start all** klicken. Die Werte speichern können Sie mit **Save**. Wenn Sie das erste mal auf Speichern klicken, erscheint ein Datei-Requester, in welchen Sie den Dateinamen angeben oder auswählen können. Wenn Sie weitere Testläufe absolviert haben und auf **Save** klicken, werden die Ergebnisse an die anfangs ausgewählte Datei einfach angehängt. Wahlweise können Sie die Ergebnisse auch mit Klick auf **Print** auf dem Drucker ausgeben.

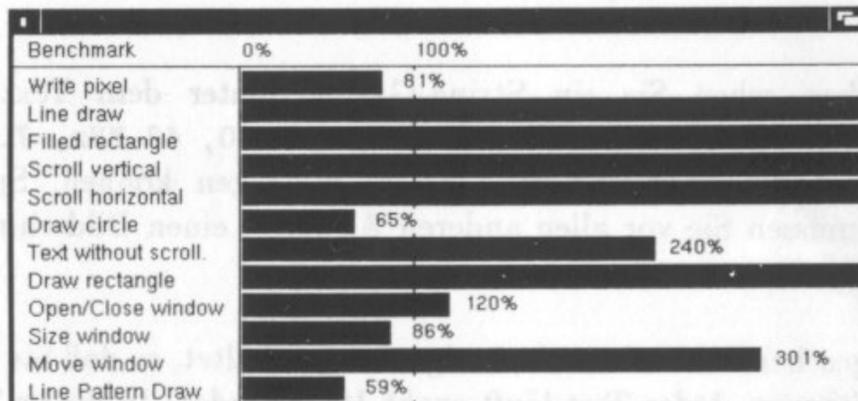
Wenn Sie die Anzahl der Farben oder den ScreenMode verändern, werden alle gemessenen Ergebnisse gelöscht. Dies passiert, damit man immer konsistente Ergebnisse auf dem Bildschirm und in der gespeicherten Datei hat.

Die Sourcen zum Programm kann man auszugsweise in unserer Mailbox herunterladen. Aus Gründen der Vergleichbarkeit sind sie nicht komplett. Wären Sie komplett, hätte das wahrscheinlich zur Folge, daß es bald danach *optimierte* Fassungen des Programms gäbe und die Ergebnisse dann nicht mehr vergleichbar wären.

### 8.7.1. Das Grafikfenster

Zahlen an sich sind ja ganz nett, aber leider kann man sich oft von den unterschiedlichen Werten erst ein *Bild* machen, wenn man eine Grafik sieht, die die Werte in Form von Balken zeigt.

Hierzu dient das Gadget **Show Graphic**. Nach einem Klick auf dieses Gadget erscheint ein weiteres Fenster, wie zum Beispiel folgendes:



Die Ergebnisse in grafischer Form

Die Angaben sind nun nicht mehr absolute Zahlenwerte, sondern relative in Bezug auf einen Amiga 3000 oder Amiga 4000. Das Programm ermittelt intern die CPU, die in Ihrem Amiga ihre Arbeit verrichtet und vergleicht die ermittelten Ergebnisse dann mit denen, die vorher auf einem A3000 oder A4000 gestoppt wurden.

Zur Ermittlung der Referenz-Werte wurde die Auflösung **PAL:HighRes Interlaced** mit allen möglichen Farbtiefen herangezogen. Die Messungen fanden auf einem frisch installierten System statt, auf der noch kein Programm in **Sys:WBstartup** lag und keine **S>User-Startup** existierte.

Sollten Sie einen Rechner mit 68030-CPU besitzen und einen Test mit mehr als 16 Farben machen, so werden als Vergleichswerte die für **PAL:HighRes Interlace** in 16 Farben verwendet.

## 8.8. VilVersion

VilVersion ist ein sehr einfaches Programm. Es hat lediglich die Aufgabe, Ihnen die aktuelle Version der Software mitzuteilen. Nach einem Doppelklick auf das Icon erscheint ein Fenster wie zum Beispiel dieses:

*So kann ein Fenster von VilVersion aussehen.*

Diesem Fenster können Sie entnehmen, welche Software-Version Sie erhalten haben und wann die Software zusammengestellt wurde. Diese Angaben sind aber nur als Hinweis gedacht, damit Sie feststellen können, ob Ihre Software noch up-to-date ist.

VilVersion können Sie auf der ersten Diskette **Install Picasso** im Wurzel-Verzeichnis finden und per Doppelklick starten.

Wenn Sie Fehler im Intuition-Treiber oder anderen Programmen feststellen, und wenn Sie uns das mitteilen wollen, so sollten Sie hierzu das Programm VilReport (siehe Seite 75) verwenden, daß automatisch wichtige Daten über Ihren Computer sammelt.

## 8.9. VilReport

Das fehlerlose Produkt können wir (leider!) nicht garantieren. Aber wir möchten diesem Ideal so nahe wie möglich kommen. Damit sind wir aber auch auf Ihre Mitarbeit angewiesen. Natürlich nur, wenn Sie auch dazu bereit sind. Sie müssen selbstverständlich nicht!

Bei vielen Briefen und Faxen in der Vergangenheit fehlten leider oft die wichtigsten Angaben. Zuerst sind da natürlich die verschiedenen Versionen der von uns gelieferten Programme zu nennen. Aber auch das Zusammenspiel mit anderen Programmen kann zu Fehlern führen, die sonst nicht auftreten. Damit Sie beim Sammeln der Angaben sich kaum Arbeit machen müssen, ist VilReport entstanden.

VilReport generiert einen Report über die aktuelle Systemkonfiguration Ihres Rechners inklusive **s:startup-sequence**, **s:user-startup**, dem Inhalt der Schublade **Sys:WBStartup**, der aktuellen Library-Liste und anderen System-Informationen. Am Ende der Liste können Sie eine Fehlerbeschreibung mit einem Editor vornehmen.

Wenn alle Informationen beisammen sind, besteht die Möglichkeit, sie auszudrucken oder als Datei zu speichern, um sie uns zukommen zu lassen. Bitte verstehen Sie uns nicht falsch: Wir wollen nicht Ihren Rechner ausspionieren, sondern möglichst viele Informationen sammeln, um Fehler besser eingrenzen zu können. Da das Programm reine ASCII-Daten erzeugt, haben Sie stets die Möglichkeit sich anzusehen, was das Programm als Ausgabe erzeugt hat.

### 8.9.1. Bedienung

Nach dem Start von VilReport sollte ein Fenster in der Mitte der Workbench auftauchen, das ungefähr so aussieht (in dem abgebildeten Fenster wurden schon ein paar Daten eingetragen, lassen Sie sich dadurch nicht stören):

Das fertige Produkt können wir (leider!) nicht garantieren. Aber wir meinen, dass Sie es so nahe wie möglich kommen. Bitte sind wir aber auch auf Ihre Mitarbeit angewiesen. Natürlich nur, wenn Sie auch dazu bereit sind. Sie müssen selbstverständlich nicht.

Die Fehlerberichte und Faxen in der Vergangenheit liefen leider als die richtigen Angaben. Somit sind die natürlich die verschiedenen Versionen der von uns gelieferten Programme zu nennen. Aber auch das Zusammenlagern mit anderen Programmen kann zu Fehlern führen, die nicht auftreten. Damit Sie beim Sammeln der Angaben sich keine Sorgen machen müssen, ist VilReport entstanden.

#### VilReport erwartet nur ein paar einfache Angaben...

Bitte tragen Sie in die entsprechenden Felder Ihren Namen, Adresse, Rechner-Typ (wie A3000), Monitor (genaue Bezeichnung) und einen Zeilenfrequenz-Bereich (wie 30 bis 64) sowie eine Bildwiederholrate ein. Die letzten beiden Angaben sind optional.

Nachdem alles ausgefüllt, klicken Sie bitte auf **Report**. Das Programm durchsucht nun einige Verzeichnisse und nimmt einige System-Dateien in die Liste auf. An was es gerade arbeitet, kann man der Liste im unteren Teil des Fenster entnehmen.

Nach einer kurzen Zeit, erscheint ein weiteres Fenster mit einer Frage,

ob beim Hochfahren des Rechners noch andere Script-Dateien ausgeführt werden, als dies normalerweise der Fall ist. Wenn Sie lediglich eine **S:User-Startup** am Ende der **S:Startup-Sequence** ausführen lassen, können Sie die Frage mit **Nein** beantworten. Bei **Ja** müssen Sie anschließend die Datei angeben, die noch ausgeführt wird.

Zum Schluß der Analyse-Phase werden Sie gefragt, ob Sie eine Fehlerbeschreibung eingeben wollen (was wohl immer der Fall sein wird).

Hierzu wird der Editor gestartet, dessen Namen in der Environment-Variable **EDITOR** verzeichnet ist. Gibt es diese Variable nicht, lädt VilReport den Editor **C:Ed** von Commodore. Der Editor sollte sich nicht vom startenden Prozeß *abnabeln*, damit VilReport nicht fälschlicherweise annimmt, der Benutzer hätte den Text schon getippt, und dann eine leere Datei als Fehlerbeschreibung übernimmt.

Der so erstellte Report kann anschließend ausgedruckt oder gespeichert werden. Sie können uns den Text sowohl zufaxen, per Brief oder Diskette schicken oder am einfachsten per Modem in unsere Mailbox legen (am besten dem Sysop schicken).

#### 8.9.2. Aufruf per CLI oder Workbench

Statt alle Angaben immer wieder beim Start des Programms einzugeben, können diese schon beim Aufruf als Parameter angegeben werden:

```
FILE,USER/K,ADDRESS/K,AMIGA/K,MONITOR/K,KHZ/K,
HZ/K,BUGREPORT/K,NOSCRIPTS/S:
```

Dabei stehen die Parameter für

<b>FILE</b>	Dateiname für den erzeugten Report; voreingestellt ist <b>T:VilReport.tmp</b>
<b>USER</b>	Ihr Vor- und Zuname
<b>ADDRESS</b>	Ihre Adresse (eventuell mit Telefonnummer)
<b>AMIGA</b>	Ihr Amiga, zum Beispiel <b>A4000</b>
<b>MONITOR</b>	Die Bezeichnung Ihres Monitors, z.B.: <b>NEC 5 FGe</b>
<b>KHZ</b>	Der Zeilenfrequenzbereich, <b>xx bis yy</b>
<b>BUGREPORT</b>	Name einer Datei mit der Fehlerbeschreibung
<b>NOSCRIPTS</b>	um VilReport mitzuteilen, daß es nicht noch nach weiteren Scripten fragen soll

Die gleichen Parameter können Sie auch als Tooltypes in das Icon des Programms eintragen, wie Sie das wahrscheinlich schon von anderen Programmen zur Picasso II kennen.

## 9. Bildanzeige-Programme

Was nützt einem die Farbvielfalt, wenn man keine Programme hat, die diese nutzen. Nicht jeder hat *Art Department Professional* oder *ImageFX*. Deshalb haben wir ein paar Programme entwickelt, mittels derer Sie sich Bilder anzeigen lassen können. Diese Bilder können im IFF-ILBM-, GIF- oder JPEG-Format gespeichert sein.

Abhängig vom Format gilt es dann, das richtige Anzeigeprogramm – englisch *Viewer* – für das jeweilige Bild auszuwählen. Die Programme können von einer Shell oder von der Workbench aus gestartet werden und kennen jeweils einige Parameter, die sich teilweise gleichen.

Einfacher ist es jedoch, sich des Programms *IntuiView* zu bedienen, daß dank seiner Analyse-Fähigkeiten immer das richtige Anzeigeprogramm zu einer Bilddatei auswählt (siehe Seite 62) und sich ebenfalls von einer Shell oder der Workbench aus starten läßt.

### 9.1. ViewIFF

Das wichtigste Bild-Format auf dem Amiga ist das IFF-ILBM-Format. Kaum ein Programm, welches nicht sogenannte *IFF-Bilder* speichern oder sie lesen könnte.

**ViewIFF** liest alle gängigen IFF-Formate. Dazu gehören Bilder bis zu 24 Bitplanes. Sollten Sie Bilder in dem Programm unbekanntem Format haben, so lassen Sie uns das wissen. Wir werden dann versuchen, das Programm so zu erweitern, daß es auch dieses IFF-Format liest.

**ViewIFF** kennt einige Parameter, die es Ihnen auf Wunsch auch auflistet, wenn Sie das Programm mit einem ? als Parameter starten:

```
FILENAME/M/A,C=CENTER/S,W=WAIT/K/N,R=RESOLUTION/K/N,
B=BEHIND/S,I=INFOONLY/S
```

Da **ViewIFF** ständig weiterentwickelt wird, kann es durchaus sein, daß Sie bei einem Aufruf mit einem Fragezeichen mehr Parameter aufgelistet sehen. Sollte das der Fall sein, entnehmen Sie die Erweiterungen bitte der zugehörigen Dokumentation.

Für **FILENAME/M/A** können Sie eine beliebige Anzahl von Dateinamen und Platzhaltern wie **#?.c** einsetzen. **ViewIFF** wird dann versuchen, alle

Dateien, die Sie angegeben haben oder deren Namen auf eines der Pattern passen, anzuzeigen.

Wenn Sie die Option **CENTER** angeben, werden Bilder, die kleiner als eine der möglichen Auflösungen sind, in der Mitte des Bildschirms zentriert. Statt des langen Namens können Sie auch einfach nur ein **C** angeben.

Normalerweise wartet **ViewIFF** auf einen Mausklick, um die Darstellung des Bildes zu beenden. Wenn Sie aber so etwas wie eine DiaShow oder ähnliches abspielen möchten, ohne nach jedem Bild eine Maustaste drücken zu müssen, können Sie den Parameter **WAIT** und eine Zahl angeben. Die Zahl ist die Wartezeit in Sekunden, bis **ViewIFF** den Bildschirm schließt und das nächste Bild aufbaut.

Manche Bilder sind größer als die größte Auflösung, die Sie auf Ihrem Monitor darstellen können. **ViewIFF** versucht aber immer einen Bildschirm zu öffnen, der sowohl in der Horizontalen, als auch in der Vertikalen mindestens so groß wie das anzuzeigende Bild ist. Hat ein Bild die Ausmaße 600 mal 800 Punkte, wird **ViewIFF** versuchen, einen Screen der Größe 1120 mal 832 zu öffnen. Ist dieser ScreenMode nicht nutzbar (vielleicht weil Sie keinen 16-Zoll-Monitor haben), bricht das Programm mit einem Fehler ab. Sie können jedoch **ViewIFF** dazu zwingen eine bestimmte Auflösung zu wählen, egal wie groß das Bild ist. Hierfür gibt es die Option **RESOLUTION**, hinter der Sie die *Breite* der Auflösung in Pixel angeben. Diese ist exakt anzugeben. Für das obige Beispiel könnte ein Aufruf wie folgt aussehen:

```
ViewIFF MeinBild.iff RESOLUTION=1024 CENTER
```

Um den Aufbau nicht zu sehen, oder aber zwei Aufrufe von **ViewIFF** verzahnt ausführen zu lassen, gibt es den Parameter **BEHIND**. Damit wird ein Bild im Hintergrund aufgebaut. Es kann anschließend in den Vordergrund geholt werden, indem man dem Programm ein **CTRL-F** schickt. Sollte man zusätzlich noch zu **BEHIND** den Parameter **WAIT** angeben haben, so kommt das Bild automatisch in den Vordergrund, sobald es komplett aufgebaut ist.

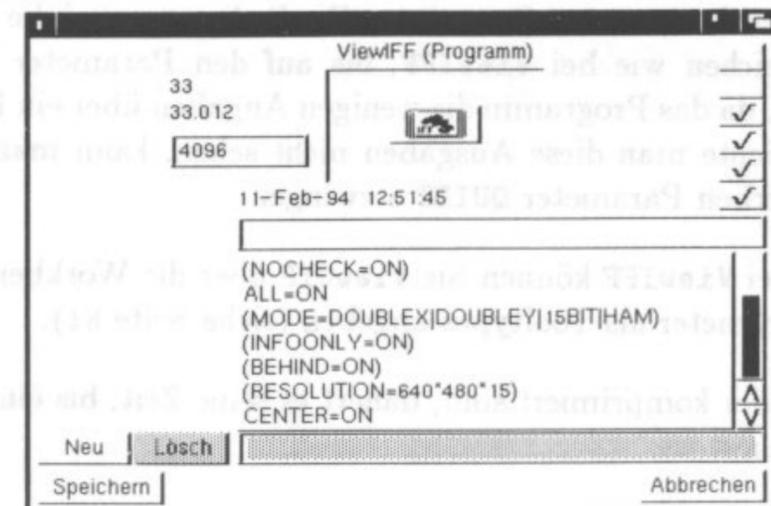
**INFOONLY** schließlich dient dazu, sich ein paar Daten über ein Bild geben zu lassen. Dies ist vor allem dann nützlich, wenn man sich darüber wundert, warum **ViewIFF** ein Bild nicht anzeigen will. Die Angaben sind

aber nur für denjenigen nützlich, der sich näher mit dem IFF-Format beschäftigt hat.

Der Start von der Workbench ist denkbar einfach: Ein Doppelklick auf das Icon genügt und es erscheint ein Datei-Requester, in welchem Sie eine Datei auswählen können, die Sie sich ansehen möchten. Wenn die Datei angezeigt worden ist, reicht ein Klick mit der linken Maustaste, um den Bildschirm mit dem Bild zu schließen. Danach erscheint wie nach dem Start erneut der Datei-Requester. Haben Sie genug gesehen, klicken Sie bitte auf **Abbruch**.

Statt nun alle möglichen Bilddateien mit dem Datei-Requester auszuwählen, können Sie auch das Icon von **ViewIFF** einfach und dann der Reihe nach alle Icons der Bilddateien anklicken, die Sie betrachten wollen. Wenn Sie das letzte Icon einer Bilddatei zweifach angeklickt haben, zeigt Ihnen **ViewIFF** der Reihe nach alle Bilder an. Nach einem einfachen Klick auf ein betrachtetes Bild, erscheint das nächste, bis sie alle angezeigt worden sind.

In einer Shell konnte man **ViewIFF** Parameter angeben, um eventuell die Auflösung zu ändern oder die Zentrierung einzuschalten. Dies geht auch, wenn man das Programm von der Workbench aus startet. Hierzu hat man in dem Icon von **ViewIFF** die Parameter zu setzen. Klicken Sie dazu einfach auf das Icon und drücken Sie die Tasten **AMIGA i**. Es erscheint ein Fenster wie dieses:



In der Liste rechts unten sehen Sie einige Einträge, die den Schlüsselwörtern der Shell-Parameter gleichen – zum Beispiel das Wort **CENTER=ON**. Wenn ein Eintrag eingeklammert ist, dann bleibt er ohne Wirkung. Er steht nur da, um Sie daran zu erinnern, daß es ihn gibt. Einigen Einträgen kann man einen Wert zuweisen, wie etwa **RESOLUTION** oder **WAIT**, andere können dagegen nur *an* oder *aus* sein wie **CENTER**, **BEHIND** und **INFOONLY**. Die Wirkung ist die gleiche, wie bei den Shell-Parametern.

Ein besonderer Vorteil ist, daß Sie diese sogenannten Tooltypes – so heißen nämlich diese Einträge in einem Icon – nicht nur im Icon von **ViewIFF** stehen können, sondern auch in einem Bild-Icon. So können Sie jedem Bild die gewünschte Auflösung aufzwingen oder einige zentriert anzeigen lassen, andere dagegen nicht.

Leider ist der Aufbau von IFF-Dateien völlig gegensätzlich zum Aufbau des Bildspeichers in den Modi mit 256 oder mehr Farben. Für die Anzeige muß deshalb jedes einzelne Bit – und zu einem Pixel gehören meist 8 oder mehr Bit – *gedreht* werden. Bei einem Bild mit 800 mal 600 Punkten und 256 Farben sind das etwas über 3.8 Millionen, bei 640 mal 480 in 16 Millionen Farben schon über 7.3 Millionen Bit. Das dauert natürlich ein wenig. Die Anzeige-Programme profitieren deshalb am meisten von einer schnellen CPU.

## 9.2. ViewGIF

Neben IFF-Bildern findet man recht häufig auch GIF-Bilder, die vor allem auf größeren Rechnern an Universitäten zu finden sind. Für diese Bilder gibt es das **ViewGIF**-Programm. Die Bedienung und die Parameter sind die gleichen wie bei **ViewIFF**, bis auf den Parameter **INFOONLY**. Dieser fehlt, da das Programm die wenigen Angaben über ein Bild immer ausgibt. Möchte man diese Ausgaben nicht sehen, kann man dies über den zusätzlichen Parameter **QUIET** erzwingen.

Wie auch bei **ViewIFF** können Sie **ViewGIF** über die Workbench starten und die Parameter als Tooltypes angeben (siehe Seite 81).

Da GIF-Bilder komprimiert sind, dauert es seine Zeit, bis ein Bild komplett aufgebaut ist.

## 9.3. ViewJPEG

Im allgemeinen verbrauchen Bilder ziemlich viel Platz auf der Festplatte oder Diskette. Um diesen Verbrauch so gering wie möglich zu halten, hat es in letzter Zeit einige Entwicklungen gegeben, um speziell Bilder sehr effizient zu packen. Ein in diesem Sinne sehr gutes Format ist JPEG. JPEG lohnt sich vor allem bei TrueColor-Bildern, die es sehr gut packt, bei 8-Bit-Bildern ist man mit GIF meist besser bedient.

Die Parameter von **ViewJPEG** sind:

**FILE/M,8BIT/S,15BIT/S,16BIT/S,24BIT/S,VERBOSE/S**

Wie schon von den anderen Viewern bekannt, kann man für **FILE/M** beliebig viele Dateien mit und ohne Platzhaltern angeben.

Über die Qualität der Anzeige entscheidet einer der Parameter **8Bit**, **15BIT**, **16BIT** und **24Bit**. Mittels dieser gibt man die Farbtiefe in Bit an. Dabei steht **8BIT** für 256 Farben, **15BIT** für 32768 Farben, **16BIT** für 65536 Farben und **24BIT** schließlich für 16 Millionen Farben. Gibt man keinen dieser Parameter an, wählt **ViewJPEG 15BIT**.

**VERBOSE** schließlich führt zu einer Ausgabe mit Angaben zur Größe des Bildes und in welcher Farbtiefe es angezeigt wird.

**CTRL-C**, **q**, **Q** oder auch **ESC** bricht die Anzeige jederzeit ab, ein Klick auf das fertige Bild oder ein Tastendruck auf **n** führt zur Anzeige des nächsten Bildes, falls Sie mehrere Bilder beim Start angegeben haben.

Wie auch bei den anderen beiden Anzeige-Programmen können Sie die CLI-Parameter auch als ToolType im Icon des Programms eintragen. Wenn Sie beim Start keinen Dateinamen angeben, erscheint ein Datei-Requester, in welchem Sie eine Datei auswählen können. Ab Workbench 3.0 können Sie auch mehrere Dateien auf einmal auswählen.

Auch für JPEG gilt: Das Entkomprimieren benötigt Zeit und Rechenleistung. Wer eine schnelle CPU in seinem Rechner hat, muß nicht so lange warten.

## 10. Treiber für andere Programme

Bereits viele Programme, die hochauflösende oder vielfarbige Grafiken darstellen könnten – aber nicht können, da die Amiga-Grafik nur weniger Farben oder geringere Auflösungen unterstützt – gibt es in speziellen Versionen für verschiedene Grafikkarten.

Einige davon haben ein Modul-Konzept, so daß man auch nachträglich – ohne das eigentliche Programm zu verändern – ein Modul erstellen kann, um eine neuere Grafikkarte zu unterstützen. Verständlicherweise sind wir sehr bemüht, möglichst viele solcher Programme zu unterstützen.

Bei Programmen ohne Modulkonzept, wie zum Beispiel **Imagine**, können wir wenig machen, außer uns mit dem Hersteller solch eines Programms in Verbindung zu setzen, um ihn davon zu überzeugen, daß eine Anpassung sinnvoll und erwünscht ist. Einen Erfolg dieser Art konnten wir beispielsweise bei TecSoft – dem Hersteller von **TVPaint** erzielen, der eine Anpassung vorgenommen hat.

Programme wie **Maxon Paint** oder **Personal Paint** laufen glücklicherweise ohne Anpassung, so daß für diese Fälle keine zusätzliche Arbeit nötig ist.

Die Anpassung von Programmen mit Anzeige-Modulen dagegen macht weniger Arbeit und ist im allgemeinen konzeptbedingt einfach zu implementieren. Im Laufe der Zeit sind recht viele Treiber entstanden, so daß wir im Handbuch darauf verzichten, diese näher zu beschreiben. Im allgemeinen finden Sie über die Einbindung von Grafikkarten bei jedem Produkt ein spezielles Kapitel.

Wie die Einbindung zu erfolgen hat, können Sie dort erfahren. So wird ein Picasso II-Treiber bereits bei **AdPro**, **ImageFX** und **Cinema 4D** mit ausgeliefert. Der Installer ist in der Lage die Treiber an die jeweils richtigen Stellen zu kopieren.

Bei Fragen zur Bedienung werfen Sie bitte einen Blick in die Dokumentation, die für einige Treiber in Form eines Dokumentes verfügbar ist.

## 11. Probleme beheben

Sollte die Karte nicht gleich funktionieren: Nur die Ruhe! Die Karten, die wir ausliefern, sind alle getestet und es ist eher unwahrscheinlich, daß sie beim Transport beschädigt wurden. Meist sind es ganz simple Sachen, die ein Funktionieren verhindern.



Deshalb sollten Sie sich bei einer Fehlfunktion oder vermeintlichem Totalausfall die folgenden möglichen Fehlerursachen und mögliche Lösungen durchlesen.

Sollten Sie tatsächlich einen Fehler in der Software gefunden haben, so benutzen Sie bitte das Programm **VilReport**, um uns davon in Kenntnis zu setzen. Die Bedienung ist einfach.

### Überhaupt kein Monitorbild

Kontrollieren Sie zuerst, ob der Monitor eingeschaltet ist und die Kabel zwischen Rechner, Grafikkarte und Monitor fest sitzen. Vergewissern Sie sich mit Hilfe des Handbuches, ob die Kabel in den richtigen Steckern sitzen. Siehe dazu auch die Zeichnung auf Seite 10

### Die Picasso II liefert kein Bild

Nach dem Booten (Einschalten des Rechners) haben Sie schon ein Bild, so wie sie es vom Amiga gewohnt sind. Nur beim Umschalten auf die Picasso II ist der Bildschirm schwarz. Dann haben Sie wahrscheinlich die beiden Stecker an der Picasso II vertauscht angesteckt. Versuchen Sie es einfach einmal andersherum.

Sollten Sie einen ganz neuen oder sehr guten Monitor besitzen, kann es auch sein, daß Ihr Monitor nicht in der Lage ist, die gewählte Auflösung darzustellen, da die Zeilenfrequenz zu niedrig ist. Diese kann man im Icon der **village.library** in **Sys:Expansion** einstellen. In diesem Fall, versuchen Sie den Rechner so zu starten, daß nach dem Booten eine Auflösung gewählt wird, die der Monitor anzeigen kann. Eventuell erreichen Sie dies bereits, indem Sie von der von Commodore mitgelieferten **Workbench-Diskette** booten.

### Es ist nur ein Flimmern zu sehen

Möglicherweise haben Sie die Software nicht richtig konfiguriert oder/und der Monitor ist nicht in der Lage, die gewünschte Auflösung darzustellen, da die Zeilenfrequenz der gewählten Auflösung zu niedrig oder zu hoch ist. Diese kann man im Icon der `village.library` in `Sys:Expansion` einstellen.

Versuchen Sie den Rechner so zu starten, daß nach dem Booten eine Auflösung gewählt wird, die der Monitor darstellen kann. Eventuell erreichen Sie dies bereits, indem Sie von der von Commodore mitgelieferten Workbench-Diskette booten.

### Das Bild läuft durch

Der Monitor stellt das Bild nicht richtig dar. Möglicherweise haben Sie die Software nicht richtig konfiguriert oder/und der Monitor ist nicht in der Lage, die gewünschte Auflösung darzustellen, da die Zeilenfrequenz der gewählten Auflösung zu niedrig oder zu hoch ist. Diese kann man im Icon der `village.library` in `Sys:Expansion` einstellen.

Versuchen Sie den Rechner so zu starten, daß nach dem Booten eine Auflösung gewählt wird, die der Monitor fangen kann. Eventuell erreichen Sie dies bereits, indem Sie von der von Commodore mitgelieferten Workbench-Diskette booten.

### Der Rechner bootet nicht

Entweder hat der Rechner keinen Strom (kontrollieren Sie die Kabel!) oder aber die Picasso II steckt nicht richtig im Steckplatz. Im letzten Fall müssen Sie den Rechner noch einmal öffnen und prüfen, ob das Halblech angeschraubt ist und die Picasso II auch richtig im Steckplatz sitzt. Da die Amiga-Rechner mechanisch einige Toleranzen aufweisen, reicht es manchmal schon, wenn man ein wenig an der Karte wackelt.

### Die Picasso II-Auflösungen fehlen im ScreenMode-Requester

Das kann mehrere Gründe haben. Kontrollieren Sie, ob die `village.library` in `Sys:Expansion` steht, das Programm `Binddrivers` in der `S:Startup-Sequence` gestartet wird, ob die Monitor-Datei `Picasso` in `DEVS:Monitors` (ab Workbench 2.1) oder in `Sys:WBStartup` (Workbench 2.0) zu finden ist.

Sollte das alles der Fall sein, dann ist die Picasso II beim System-Start nicht eingebunden worden. Dies kann daran liegen, daß Sie kaputt ist oder nicht richtig im Slot steckt. Um sicherzugehen, sollten Sie zuerst das Programm `ShowConfig` starten und mit der folgenden Ausgabe vergleichen:

#### BOARDS:

Board (...): Prod=2167/11(\$877/\$B) (0\$200000 2meg)

Board (...): Prod=2167/12(\$877/\$C) (0\$E90000 64K)

Wichtig bei dieser Auflistung sind die Produkt-IDs, die hier die Werte 2167/11 und 2167/12 haben. Diese beiden müssen auftauchen, sonst arbeitet die Karte nicht richtig. Statt der 2167/12 kann auch der Wert 2167/13 auftauchen, wenn Sie die Karte segmentiert betreiben (siehe dazu Seite 101).

Fehlen diese Einträge, dann müssen Sie den Rechner öffnen und den korrekten Sitz der Karte kontrollieren.

Für den Fall, daß die Karte gar nicht funktioniert, möchten wir Sie bitten, die Picasso II eventuell bevor Sie sie zurückschicken, noch bei einem Freund oder Bekannten zu testen. Dies hilft uns eventuell bei der Fehlersuche weiter.

### Das Bild sieht ein wenig zerstückelt aus

Wenn Sie nach einigen Verschiebeaktionen plötzlich bemerken, daß Teile des Bildschirms nicht neu gezeichnet werden, dann haben Sie vermutlich das Programm `CPUBlit` in Benutzung. Wenn `CPUBlit` nach dem Einbinden der `Picasso-Monitor-Datei` eingebunden wird – das ist der Normalfall –, dann können diese Fehlfunktionen auftreten.

Abhilfe: Entfernen Sie `CPUBlit` aus dem System und sorgen Sie dafür, daß es beim Starten des Rechners nicht automatisch eingebunden wird. Sie können auch getrost auf das Programm verzichten, da zeitraubende Aktionen auf dem Bildschirm sowieso vom Blitter auf der Picasso II übernommen werden und somit die Notwendigkeit entfällt, diese Arbeiten mit der CPU zu erledigen.

### Fehlermeldung: Konnte vilintuisup.library nicht öffnen

Dies kann die Ursache dafür sein, daß die vilintuisup.library in LIBS: fehlt, sich die village.library nicht öffnen läßt oder aber auch der Speicher der Picasso II segmentiert eingebunden wurde. Ältere Versionen der vilintuisup.library sind nicht in der Lage, mit segmentiertem Speicher zu arbeiten. Sollte letzteres der Fall sein, lesen Sie hierzu im Anhang ab Seite 101 nach, wie man die segmentierte Einbindung wieder rückgängig macht.

Ein anderer Grund könnte sein, daß Ihre Picasso II nicht richtig im Rechner sitzt oder nicht arbeitet. Überprüfen Sie durch einen Aufruf von ShowConfig – wie oben beschrieben – ob die Picasso II korrekt eingebunden wurde.

## 12. Programmierung der Picasso II

Die Picasso II leistet viel mehr, als ihr der Intuition-Treiber abverlangt. Wie auf anderen Rechner-Systemen bereits üblich können Sie auch Bildschirme öffnen, die 32.768, 65.536 oder gar 16.777.216 verschiedene Farben auf einmal darstellen können.

Diese Bildschirme sind aber komplett anders organisiert, als dies unter Intuition üblich ist. Das bedeutet, alle Funktionen, die die normale Grafikspeicherorganisation voraussetzen, müßten für die neuen Modi umgeschrieben werden. Dies war uns doch ein wenig zu viel Arbeit (dazu braucht man Jahre!), und wir haben die anderen Modi so eingebunden, daß Sie als Screens im System auftauchen, alle zeichnenden Intuition-Funktionen jedoch ohne Wirkung bleiben, wenn man Sie auf einen solchen Screen anwendet.

Von diesen besonderen Screens machen alle jene Programme und Treiber Gebrauch, die Screens mit mehr als 256 Farben benutzen. Damit auch Sie selbst Programme für diese besonderen Modi schreiben können, haben wir einiges an Beispielprogrammen, Includes, Libraries und Dokumentation zusammengetragen.

Die folgenden Ausführungen geben Ihnen – hoffentlich – genügend Informationen, um die Karte selbst zu programmieren. Die AutoDocs finden Sie aus Platz- und Aktualitätsgründen nicht im Handbuch. Sie liegen als Datei vor. Um selbst Programme zu schreiben, seien Ihnen die Beispielprogramme 8BitLineDemo.c, 8BitBlitterDemo.c und 8BitFillDemo.c ans Herz gelegt, in denen einige wichtige Funktionen zum Einsatz kommen.

### 12.1. Einleitung

Die Picasso II ist eine Mischung zwischen Framebuffer und echter Grafikkarte. Ein Framebuffer ist reiner Videospeicher, in den die CPU im schlimmsten Fall noch nicht einmal direkt schreiben kann. Dieser Speicherinhalt wird über spezielle Hardware so an den Monitor geschickt, daß ihn der Monitor als Grafik anzeigt. Bei besseren Framebuffern kann man auch aus dem Bildschirmspeicher schreiben und lesen, so daß manche Anbieter schon von einer Grafikkarte sprechen.

Eine Grafikkarte besitzt aber Eigenintelligenz, sonst verdient Sie den Namen nicht. So sollte sie bestimmte Operationen wie Linien ziehen, Bereiche verschieben und Flächen füllen selbst erledigen, um somit die CPU zu entlasten.

Die Picasso II ist keine vollgültige Grafikkarte gemäß der obigen Definition. Dennoch besitzt sie einen Blitter, der Kopier- und Verschiebeaktionen selbständig erledigt. Das Zeichnen von Linien muß jedoch die CPU übernehmen.

Um einem Programmierer der Picasso II alle Freiheiten zu lassen, kann er – nach einer Anforderung – direkt in den Bildschirmspeicher schreiben. Dies ist kein Freibrief, an bestimmte Konventionen sollte man sich schon halten. Außerdem ist der Speicher nicht immer gleich organisiert. Kurz und gut: trotz der einfachen Programmierung sollte man sich an ein paar wenige, aber wichtige Regeln halten. Was Ihnen nun noch fehlt, sind nur ein paar Informationen. Und die soll Ihnen der folgende Text vermitteln.

## 12.2. Speicherorganisation

Die Picasso II läßt sich in verschiedenen Modi betreiben. Das sind

1. Planar (bis zu 16 Farben)
2. Packed Pixel (256 Farben, auch Chunky Pixel genannt)
3. Zwei 16-Bit-Modi (32768 bzw. 65536 Farben)
4. TrueColor Modi (16777216 Farben)

In jedem Modi wird der Videospeicher anders verwaltet.

### 12.2.1. Planar

Dies ist eine andere Bezeichnung für die Organisation, die man vom Amiga schon kennt: Viele schöne Bitplanes. Die Farbinformation für ein Pixel ist auf mehrere Bitplanes verteilt. Aus jeder Plane nimmt man ein Bit, ordnet diese nebeneinander in einem Byte an, und nimmt die so entstandene Nummer, um in einer Farbtabelle nach den Rot-, Grün- und Blauwerten zu gucken. Diese gelangen dann zum Monitor, der das Pixel mit der richtigen Farbe anzeigt.

Die Bitplane mit der niedrigsten Nummer (=0), enthält dabei das niederwertigste Bit des Farbindex. In diesem Modi wird die Picasso II betrieben, wenn der Intuition-Treiber läuft. Die Organisation entspricht also genau der des Amiga.

### 12.2.2. Packed Pixel

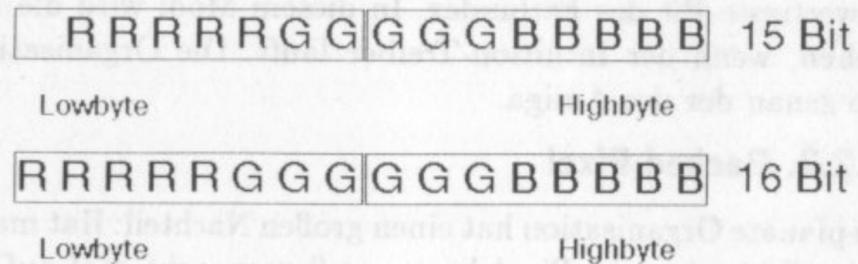
Die planare Organisation hat einen großen Nachteil: Hat man acht Planes und möchte nur ein Pixel lesen, muß man acht mal auf den Speicher zugreifen, da die acht zugehörigen Bits alle verteilt im Speicher liegen. Beim Verändern sind es eventuell sechzehn Zugriffe. Dies Problem hat auch der Amiga 4000 noch und macht ihn unnötig langsam.

Mittlerweile hat sich eine andere Organisationsform durchgesetzt: Ein Pixel, ein Byte. Das Index-Byte für die Farbtabelle wird nicht aus verteilten Bits zusammengesucht, sondern befindet sich schon in einem Byte im Bildschirmspeicher. Möchte man ein Pixel setzen oder dessen Farbe bestimmen, reicht ein einfacher Zugriff. Das geht natürlich schneller bei 256 Farben. Diese Organisation nennt man Packed Pixel. Mit ihr kann man bis zu 256 verschiedene Farben auf dem Bildschirm darstellen.

Sie hat aber auch Nachteile: Braucht man nur vier Farben, so muß man trotzdem immer ein Byte pro Pixel opfern. Beispielsweise verbrauchen 2 Bitplanes (im Planarmodus) nur ein Viertel des Speichers, den man im Packed Pixel Modus braucht: denn dieser benötigt immer ein Byte für ein Pixel. Auch das Ausmaskieren einzelner Bits ist im Packed-Pixel-Modus aufwendiger als bei Bitplanes.

### 12.2.3. 16-Bit-Modi (auch HiColor genannt)

Auch 256 Farben können zuwenig sein. Deshalb hat man weitere Modi ersonnen. Im PC-Bereich gab es zuerst sogenannte HiColor-Auflösungen. Hierbei wird ein Word (= zwei Bytes) für ein Pixel verwendet. Der Wert des Words ist aber kein Index in eine Farbtabelle, sondern enthält die Rot-, Grün- und Blauwerte direkt. Dabei bekommt jeder Farbanteil 5 Bits zugestanden, macht insgesamt 15 Bits; eins bleibt übrig. Und an diesem übriggebliebenen Bit unterscheiden sich auch die zwei 16-Bit-Modi, die die Picasso II unterstützt: Der eine läßt das 16. Bit ungenutzt, der andere verwendet für den Grün-Anteil nicht 5, sondern 6 Bits. Die Organisation in einem Wort ist dabei folgende:



Da das Wort jedoch PC-typisch abgelegt wird, kommt das Byte mit den Rotwerten in die niedrigere Adresse, das Byte mit den Blauwerten in die höhere Adresse.

#### 12.2.4. TrueColor (24Bit)

Das Bithierumgeschiebe in den HiColor-Modi kann ganz schön aufwendig werden und letztlich ist die Anzahl der möglichen Farben der wählbaren Farben kaum größer als bei einem normalen ECS-Amiga (Statt vier Bit pro Farbkomponente nur fünf).

TrueColor räumt mit dem Bitgefriemel auf und spendiert jeder Farbkomponente ein Byte. Man braucht also für ein Pixel drei Byte. Das Byte mit der niedrigeren Adresse enthält den Blauanteil, das nächste den Grünanteil, das darauffolgende den Rotanteil (also genau umgekehrt als das Kürzel RGB vermuten läßt).

Die Datenmenge für ein Pixel steigt dadurch an, die Arbeit wird aber einfacher und die mögliche Farbzahl auf einem Bildschirm steigt auf 16.777.216 Farben. Das sind mehr als das menschliche Auge differenzieren (auseinanderhalten) kann. In diesem Modus sehen Fotos wie echt und Bilder wie vom Fernseher aus. Da man allerdings in diesem Modus für jedes Pixel drei Byte benötigt, sind große Auflösungen nicht möglich. Einerseits fehlt meist das RAM, andererseits bekommt die Grafikkarte Schwierigkeiten, die großen Mengen an Daten noch schnell genug an den Monitor zu liefern. Und wird es zu langsam, kriegt man wieder dieses schöne Flickern zu sehen.

### 12.3. Einen Bildschirm öffnen

Mit der Picasso II können Sie alle der beschriebenen Modi nutzen. Sie müssen allerdings den Intuition-Treiber gestartet haben. Dann erlauben ihnen die beiden Funktionen

`OpenVillageScreenTagList (...)` und  
`CloseVillageScreen(...)`

das Öffnen und Schließen eines Screens mit den erwähnten Farbtiefen und bestimmten Auflösungen. Die Funktionen `OpenVillageScreenTagList()` und `CloseVillageScreen()` benötigen Sie allerdings nicht, wenn Sie einen Screen in einem Planar-Modi öffnen wollen. Dies geschieht wie üblich, indem Sie beim Öffnen eines Picasso II-Screens die gewünschte ViewModeID übergeben (wenn Ihnen das nicht geläufig ist, informieren Sie sich zum Beispiel im Amiga ROM Kernel Reference Manual:Libraries).

Die Funktion `OpenVillageScreenTagList()` verlangt als Parameter einen Zeiger auf eine TagList, wie dies auch bei Intuition üblich ist. Der Aufruf ähnelt dem von `OpenScreenTagList()` weitestgehend.

Die möglichen Auflösungen sind nicht genau zu bestimmen. Dies hängt damit zusammen, daß Sie mit `PicassoMode` beliebige Auflösungen erzeugen können (siehe auch Seite 37). Wenn Sie diese allerdings erzeugen, so sollten Sie dafür Sorge tragen, daß zumindest eine für jede Farbtiefe vorhanden ist.

Die früher eingeführte Funktion `OpenVillageScreen()` sollte in Zukunft nicht mehr verwendet werden. Bitte werfen Sie unbedingt einen Blick in die AutoDocs, um die nicht ganz einfache Funktion zu studieren.

Gelingt das Öffnen des entsprechenden Screens, erhalten Sie einen Zeiger auf den Screen zurück. Sollten Sie anschließend versuchen, auf dem so neu entstandenen Bildschirm ein Fenster zu öffnen, so gelingt dies zwar (!), aber Sie werden nichts sehen. Der Intuition-Treiber verhindert, daß irgendeine Intuition- oder Graphics-Funktion auf solch einen Screen zeichnet. Dies ist auch sinnvoll, da er ja völlig anders organisiert ist, als Intuition-typische Screens.

## 12.4. Zeichnen

Hüten Sie sich davor, sich nun die Startadresse des Bildschirmspeichers irgendwie zu greifen und im Speicher herumzumalen. Bevor Sie auf den Screen zugreifen können, muß ein Programm die Funktion `LockVillageScreen(ScreenPointer)` aufrufen. Diese liefert als Rückgabewert den Anfang des Bildschirmspeichers des übergebenen Screens zurück. Erst dann kann man in dem Bildschirm herummalen und nach Lust und Laune Bytes manipulieren. Ist man fertig, oder legt man eine Pause ein, sollte man mit `UnlockVillageScreen(ScreenPointer)` den Zugriff wieder freigeben.

Während man einen Bildschirm mit `LockVillageScreen()` gesperrt (ge-Locked) hat, kann zwischen verschiedenen Picasso II-Screens nicht hin- und hergeschaltet werden.

Hier ein kurzes Beispiel

```
#include <other includes>
#include "vilintuisup.h"

void HandleVillageCard()
{
    struct Screen *ScreenPointer;
    struct Dimensions dm = { 0, 800, 600, 8 };
    UBYTE *ScreenStartAdr; /* start of screen */

    /* for simplicity I use the old OpenVillageScreen() */
    if (ScreenPointer = OpenVillageScreen(&dm))
    {
        ScreenStartAdr = LockVillageScreen(ScreenPointer);

        // ... draw into the screen and use ScreenStartAdr
        UnlockVillageScreen(ScreenPointer);

        // ... let the user wait and see
        CloseVillageScreen(ScreenPointer);
    }
    else
    { PutStr("Can't open desired screen\n");
    }
}
```

## 12.5. Die Funktionen

Dort wo die // stehen, kann man eigene Routinen einfügen. Die Beispielfunktion öffnet einen Screen mit den Ausmaßen 800 mal 600 Punkten im Packed Pixel Modus. Die Werte 800, 600 und 8 werden deshalb in die Struktur `dm` geschrieben:

```
struct Dimensions dm = { 0, 800, 600, 8 };
```

Der Rest sollte eigentlich klar sein. Die Anordnung der Pixel im Bildschirm ist recht einfach. Der Pixel links oben in der Ecke hat die Adresse, die `LockVillageScreen()` zurückliefert. Der Pixel rechts daneben hat die nächst höhere Adresse (ein Byte höher bei Packed Pixel, zwei Byte bei einem der beiden HiColor-Modi und drei beim TrueColor-Modus). Ist der rechte Rand einer Zeile erreicht, befindet sich in der nächst höheren Adresse das Pixel ganz links in der zweiten Zeile. Und so geht es dann weiter. Etwas unbeholfen ausgedrückt: Die Pixel laufen von links nach rechts, die Zeilen von oben nach unten.

## 12.5. Die Funktionen

In der Funktionensammlung `vilintuisup.c` kann man sich ansehen, wie man mit dem Bildschirmspeicher umzugehen hat. Zum einen gibt es da Routinen, mit denen man ein Pixel setzen kann:

```
void SetTrueColorPixel(Screen, x, y, r, g, b);
void SetPackedPixel (Screen, x, y, color);
void Set15BitPixel (Screen, x, y, r, g, b);
void Set16BitPixel (Screen, x, y, r, g, b);
```

Bevor man diese Routinen benutzt, muß man wie bereits geschildert, den Zugriff mit `LockVillageScreen()` und `UnlockVillageScreen()` reservieren und freigeben. Dies geschieht aus Geschwindigkeitsgründen nicht in den Funktionen selbst.

Dann gibt es zwei Linienroutinen, eine für das Linienziehen im PackedPixel-Modus, `LinePacked()` genannt, und eine für das Zeichnen in TrueColor, `LineTrueColor()` genannt:

```
void LinePacked (Screen, xstart, ystart, xende, yende, color);
void LineTrueColor(Screen, xstart, ystart, xende, yende, rgbcolor);
```

Diese Funktionen erledigen das Locking selbst. Man kann es sich also sparen, vorher `LockVillageScreen()` und hinterher `UnlockVillageScreen()` aufzurufen.

Im Packed-Pixel-Modus arbeitet man mit Farbregistern, da ein Byte im Bildschirmspeicher nicht die Rot-Grün-Blau-Werte enthält, sondern nur die Nummer eines RGB-Eintrags in einer Farbtabelle. Nun braucht man eine Funktion, um die Farben in dieser Tabelle zu setzen. Dies ist die altbekannte `SetRGB4()`-Funktion. Sie funktioniert auch noch bei Packed-Pixel-Screens.

Wichtig für die Programmierung ist immer noch das Studium von Beispielprogrammen. Aus diesem Grund liefern wir einen ganzen Schwung solcher Beispiele mit aus. Bitte machen Sie sich die Mühe und drucken Sie sich einige davon aus, um Sie in Ruhe anzusehen. Es lohnt sich.

## 13. Programmierung mit C

Das Kapitel *Programmierung der Picasso II* sollte Sie soweit auf den Stand gebracht und neugierig gemacht haben, daß Sie jetzt selbst einmal ein Programm schreiben möchten. Hierzu ist jedoch noch eine weitere Hürde zu nehmen. Sie müssen die Includes, Libraries und andere Dateien an die für Ihren Compiler richtigen Orte kopieren, um die Demo-Programme erfolgreich übersetzen zu können.

Für Programmierer, die sich einer anderen Sprache als C bedienen, haben wir leider noch keine Umsetzungen fertig. Sie sollten aber durch die mitgelieferte FD-Datei in der Lage sein, mit Hilfe eines Utilities ihres Compilers sich alles nötige zu generieren, so daß Sie zumindest die Library-Funktionen der `vilintuisup.library` nutzen können. Die C-Strukturen müssen Sie per Hand übersetzen.

Bevor Sie sich die Arbeit machen, wäre es sicher von Vorteil, kurz bei uns anzurufen, um festzustellen, ob mittlerweile nicht doch eine Umsetzung existiert. Vielleicht arbeitet auch schon jemand daran, so daß wir Ihnen eine Adresse vermitteln können. Im schlimmsten Fall hat die Umsetzung noch niemand vorgenommen und wir könnten eventuell von Ihrer Arbeit profitieren – wenn Ihnen das Recht ist.

Die Demo-Programme in C sind mit den fünf gängigen C-Compilern

- SAS C Version 5.1 und 6.x
- Aztec 5.2a
- DICE Version 2.06.39
- Maxon C/C++ Version 1.02
- GNU C/C++ Version 2.2.2

getestet worden. Sie lassen sich alle mit den genannten Compilern übersetzen.

Die Sourcen der Demo-Programme, Includes und Libraries befinden sich in dem Ordner, den Sie bei der Installation angegeben haben. Sollten Sie die Installation als **Einsteiger** vorgenommen haben, sind diese allerdings *nicht* kopiert worden. Sie befinden sich dann noch auf der Installations-Diskette. Bitte starten Sie in dem Fall die Installation erneut, klicken Sie bei der ersten Frage auf den Knopf vor **Geübter Benutzer** und bestätigen Sie alle Aktionen, wie es der Installer von Ihnen verlangt.

Für die korrekte Funktion der kompilierten Programme ist die `vilintuisup.library` notwendig. Sie sollte sich in `LIBS:` befinden, wo Sie bei der Installation normalerweise hinkopiert wird.

Bitte werfen Sie vor der Programmierung einen Blick in die mitgelieferten AutoDocs. Insbesondere die Texte unter der Rubrik **BUGS** sind lesenswert, möchte man nicht stundenlang nach einem Fehler suchen, der dann gar nicht im eigenen Programm liegt.

Da die Aktionen, die für die Einrichtung nötig sind, von Compiler zu Compiler verschieden sind, folgen nun fünf Unterkapitel, in denen gezielt beschrieben wird, was bei welchem Compiler wohin zu kopieren ist.

### 13.1. SAS C Version 5.1 und 6.x

Die Installation für den SAS C ist denkbar einfach. Sie haben schlicht und ergreifend nichts zu tun. Da der Compiler die `#PRAGMA`-Direktive dazu benutzt, um Library-Aufruf-Konventionen festzulegen, brauchen Sie nicht einmal eine Link-Library. Ein Aufruf von

```
smake -f makefile_SAS6
```

im Verzeichnis, wo sich die Sourcen befinden, genügt und schon sollte der Compiler alle Demo-Programme übersetzen. Für die 5er-Version heißt der Aufruf entsprechend:

```
lmk -f makefile_SAS5
```

### 13.2. Aztec 5.2a

Beim Aztec C verhält es sich wie beim SAS C: Sie haben nichts weiter zu tun, als lediglich das `make` mit dem entsprechenden `makefile` zu füttern, um alle Demos zu übersetzen. Wie auch der SAS-C-Compiler verfügt der Aztec-C-Compiler über spezielle `#PRAGMA`-Anweisungen, um Library-Einsprünge zu spezifizieren. Damit entfällt das bei anderen Compilern nötige Kopieren von Link-Libraries.

Der Aufruf sieht damit wie folgt aus:

```
make -f makefile_Aztec5
```

Hierzu sollten Sie sich im Verzeichnis mit den Sourcen befinden.

### 13.3. DICE - registrierte Version 2.06.39

Beim DICE haben Sie die Library `vilintuisup.lib` und `vilintuisupr.lib` aus dem Verzeichnis `linklibs` nach `DLIB:` zu kopieren, da dieser Compiler keine Direktive kennt, mit der man einen Einsprung in eine Library spezifizieren könnte.

Danach brauchen Sie nur noch das `DICE-make` aufzurufen:

```
dmake -f makefile_DICE
```

Hierzu sollten Sie sich im Verzeichnis mit den Sourcen befinden.

### 13.4. Maxon C/C++ Version 1.02

Für das Maxon C/C++ gilt das gleiche wie für SAS C und Aztec C: Sie haben nichts weiter zu tun, als lediglich ein `make` mit dem entsprechenden `makefile` zu füttern, um alle Demos zu übersetzen. Wie auch der SAS-C-Compiler verfügt der Maxon-C/C++-Compiler über spezielle `#PRAGMA`-Anweisungen, um Library-Einsprünge zu spezifizieren. Damit entfällt das bei anderen Compilern nötige Kopieren von Link-Libraries.

Um die Beispieldateien allerdings übersetzen zu können, müssen Sie sich ein `Make` besorgen. Dies gibt es PD zum Beispiel auf Fish 523 (BMake inklusive Source). Der Aufruf mit solch einem `Make` hieße dann entsprechend:

```
make -f makefile_MaxonC
```

Hierzu sollten Sie sich im Verzeichnis mit den Sourcen befinden.

### 13.5. GNU C/C++ Version 2.2.2 oder höher

Wie beim DICE haben Sie beim GNU C eine Library namens `libvilintuisup.a` aus dem Verzeichnis `linklibs` nach `gcc:lib` zu kopieren.

Um die Beispieldateien übersetzen zu können, müssen Sie sich ein `Make` besorgen. Dies gibt es PD zum Beispiel auf Fish 523 (BMake inklusive Source). Der Aufruf mit solch einem `Make` hieße dann entsprechend:

```
make -f makefile_gcc
```

Hierzu sollten Sie sich im Verzeichnis mit den Sourcen befinden.

## A. Technische Daten

Ausgangssignale	Analog Rot, Grün, Blau: max. 1V VSS H-Sync, V-Sync: TTL-Pegel 15pol. High-Density VGA-Stecker (weiblich)
Eingangssignale	Analog Rot, Grün, Blau: max. 1V VSS H-Sync, V-Sync: TTL-Pegel 15pol. High-Density VGA-Stecker (weiblich)
Monitoranschluß	2 mal 15poliger VGA-Stecker
Videospeicher	1MByte oder 2MByte DRAM 256×4 (Fast Page Mode), Zugriffszeit 60 Nanosekunden oder schneller
VGA-Chip	Cirrus Logic 5426
Bus-Interface	Amiga-ZORRO-II
Farbauswahl	256 Farben aus 262144 im Workbench-Modus 256 Farben aus 262144 im Chunky-Pixel-Modus 32768 Farben gleichzeitig im HiColor-Modus 65536 Farben gleichzeitig im HiColor2-Modus 16777216 Farben gleichzeitig im TrueColor-Modus
Auflösungen	frei wählbar von 320×200 bis 1600×1300
Monitor	fast jeder Monitor, besser ein MultiScan-Monitor. Monitore, die ein Sync-Signal mit einem Farbsignal mischen, können derzeit nicht angesteuert werden.

## B. Jumper und Segmentierung

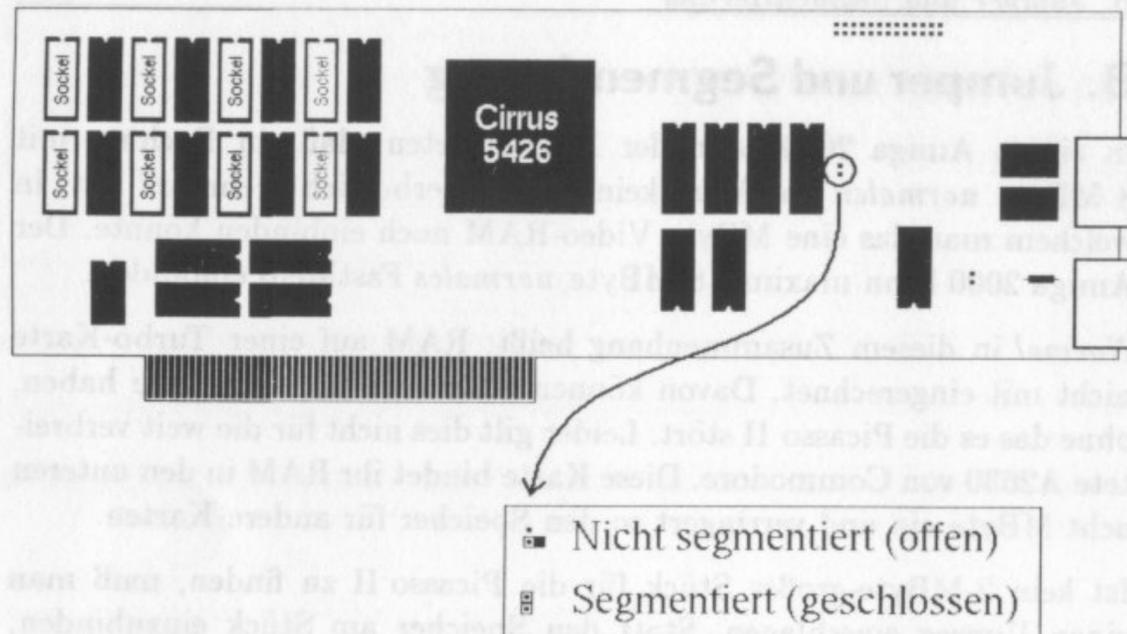
In einem Amiga 2000 kann der Fall eintreten, daß ein Rechner mit 8 MByte *normalen* FastMem keinen Speicherbereich mehr frei hat, in welchem man das eine MByte Video-RAM noch einbinden könnte. Der Amiga 2000 kann maximal 8 MByte *normales* FastMem einbinden.

*Normal* in diesem Zusammenhang heißt: RAM auf einer Turbo-Karte nicht mit eingerechnet. Davon können Sie weit über ein GByte haben, ohne das es die Picasso II stört. Leider gilt dies nicht für die weit verbreitete A2630 von Commodore. Diese Karte bindet ihr RAM in den unteren acht MByte ein und verringert so den Speicher für andere Karten.

Ist kein 2-MByte-großes Stück für die Picasso II zu finden, muß man einen Umweg einschlagen. Statt den Speicher am Stück einzubinden, wird jeweils nur ein kleines Stück davon im sogenannten I/O-Bereich eingebunden. Dazu wird der Video-Speicher in kleine Stücke (die Segmente) zerteilt. Möchte man nun in den Bildschirm schreiben (um beispielsweise einen Punkt zu setzen oder einen Bereich zu füllen), so muß man vorher das richtige Segment ermitteln und einblenden. Dies verlangsamt Grafikoperationen ein wenig und verkompliziert diverse Algorithmen ungemein – die Segmentierung wird deshalb von Programmierern nicht unbedingt geliebt.

Der Vorteil ist klar: Auch mit vollausgebautem Speicher kann man die Picasso II nutzen. Allerdings nur eingeschränkt. Der Intuition-Treiber läuft zwar uneingeschränkt, einige Programme jedoch, die sich auf die `vilintuisup.library` stützen, funktionieren nicht mit der segmentierten Einbindung zusammen. Zu nennen wären in diesem Zusammenhang TV-Paint und TruePaint. Unsere eigenen Treiber und Anzeige-Programme arbeiten mittlerweile auch segmentiert.

Normalerweise bindet sich die Karte immer nicht-segmentiert ein. Soll sie segmentiert eingebunden werden, so ist hierzu ein Jumper (eine Steckbrücke) umzustecken (siehe nächste Seite).



Wie man dem Bild entnehmen kann, befindet sich der Jumper auf der rechten Seite der Platine neben vier gleichlangen, senkrecht angeordneten ICs. Zwei Steckkontakte gucken in die Höhe, auf dem einen sollte ein – meist einfarbiger – Jumper stecken.

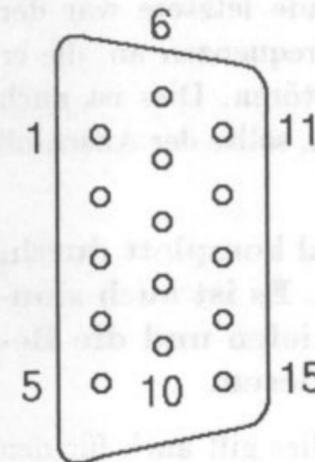
**Bevor Sie diesen Jumper umstecken, fassen Sie bitte das Gehäuse Ihres Rechners oder einen Heizkörper an, um eine eventuelle elektrische Aufladung Ihres Körpers abzuleiten.**

Ziehen Sie nun diesen Jumper ab und stecken Sie ihn so auf, daß er auf beiden Steckkontakten sitzt.

Wie Sie Ihren Rechner öffnen, schließen und die Karte in den Rechner einbauen können, ist ab Seite 16 in diesem Handbuch beschrieben.

### C. Steckerbelegung

Die VGA-Stecker der Karte haben folgende Belegung:



1. Rot
2. Grün
3. Blau
4. – (nicht verbunden)
5. – (nicht vorhanden)
6. Rot-Masse
7. Grün-Masse
8. Blau-Masse
9. 12 Volt (Pin meist nicht vorhanden)
10. Sync-Masse
11. – (nicht verbunden)
12. – (nicht verbunden)
13. H-Sync (oder C-Sync für BNC-Kabel<sup>4</sup>)
14. V-Sync
15. – (nicht verbunden)

<sup>4</sup> siehe Anhang E ab Seite 106

## D. Anschluß eines 1084 und 1081

Ansich war die Picasso II schon immer in der Lage, einen Commodore Monitor 1084 anzusteuern. Was fehlte, war nur das richtige Kabel und eine auf den Monitor abgestimmte Software. Gerade letztere war der Knackpunkt, denn steuert man den Monitor mit Frequenzen an, die er nicht verarbeiten kann, ist es möglich, ihn zu zerstören. Dies ist auch nach wie vor **möglich**, wenn Sie allerdings aufpassen, sollte der Anschluß reibungslos vonstatten gehen.

**Bitte lesen Sie den folgenden Text erst einmal komplett durch, bevor Sie den A1084 oder A1081 anschließen. Es ist auch sinnvoll das Kapitel Von Auflösungen und Farbtiefen und die Beschreibung zu PicassoMode (ab Seite 37) zu lesen.**

Leider ist es bei den bestehenden Monitoren (und dies gilt auch für den 1084) nicht möglich, abzufragen, was für ein Typ von Monitor an der Picasso II angeschlossen ist. Dies muß der Benutzer bestimmen und der Software, die den Monitor ansteuert, mitteilen. Hierbei können natürlich Fehler passieren. Deshalb ist die Monitor-Auswahl mit Bedacht vorzunehmen.

**Wichtig: Wir leisten keinerlei Schadensersatz, wenn Sie Ihren Monitor durch unsachgemäße Bedienung der Software oder falschen Anschluß der Hardware-Komponenten beschädigen oder zerstören. Sollten Sie sich die Installation nicht zutrauen, wenden Sie sich bitte an Ihren Händler.**

Für den Anschluß eines A1081 oder A1084 benötigen Sie ein spezielles Kabel, das Sie bei Ihrem Händler oder auch bei uns erwerben können. Neben der Umsetzung der Signale auf die richtigen Pins enthält das Kabel in einem Stecker noch ein IC, um den Commodore-Monitor richtig ansteuern zu können.

Der Anschluß sollte Ihnen keine Probleme bereiten: Die Seite mit dem VGA-Stecker gehört in die Picasso II, der Stecker an der anderen Seite des Kabels in das zum Monitor mitgelieferte Kabel.

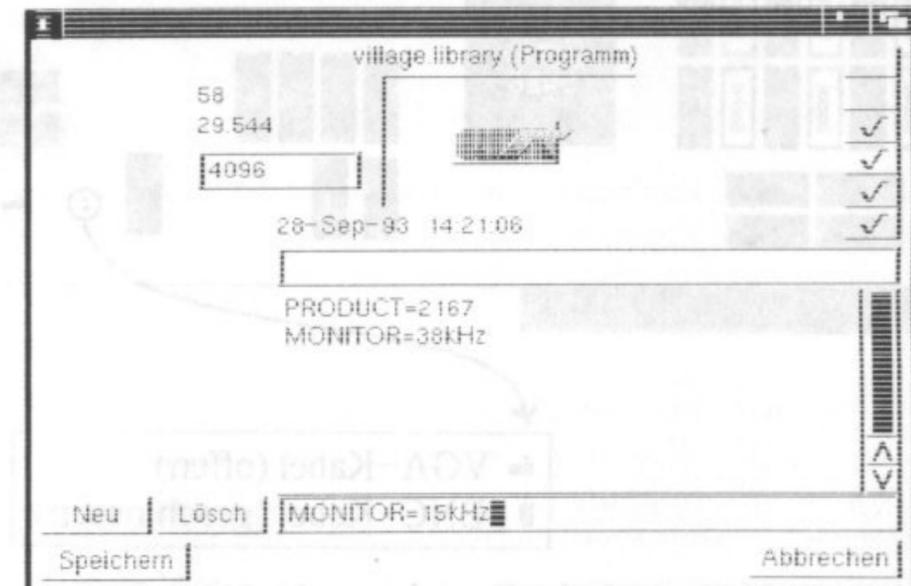
Neben den Einstellungen mit PicassoMode gibt es noch eine einfache Möglichkeit, einen A1084 anzusteuern. Hierzu verändern Sie einfach nur einen Tooltype-Eintrag in der `village.library`. Klicken Sie einfach auf das Icon der `village.library`, die sich in der Schublade `Expansion`

Ihrer Boot-Partition befindet. Drücken Sie bitte **AMIGA i**, um das Informationsfenster der Workbench zu öffnen.

Dort kann man in der Mitte eine Liste sehen, die zwei Einträge enthält. Für uns ist nur der zweite

**MONITOR=xxkHz**

interessant. Hinter **MONITOR** ist der Kilohertz-Wert anzugeben, den der Monitor maximal verkraftet. Da der 1084 (und 1081) jeweils nur 15.625 Hertz verarbeiten können, tragen Sie dort den Wert **15** ein:



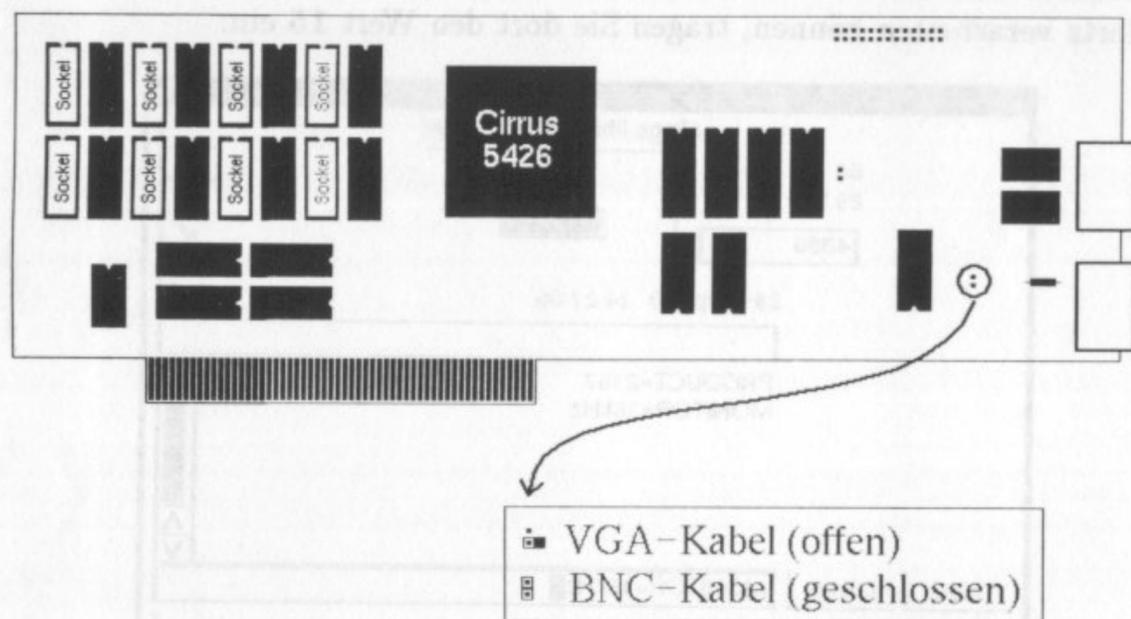
Das Informations-Fenster der `village.library`

Damit wird der Commodore-Monitor fortan nur noch mit dieser Frequenz angesteuert. Nach einem erneuten Hochfahren des Rechners können Sie nun die im ScreenMode-Requester erscheinenden Auflösungen auswählen.

## E. Monitor-Anschluß über BNC-Kabel

Wenn Sie einen recht guten Monitor einsetzen, kann es für die Bildqualität von Vorteil sein, ein sogenanntes BNC-Kabel zu benutzen, das für die Farbanteile Rot, Grün, Blau und die Sync-Impulse ein separat abgeschirmtes Koaxkabel vorsieht.

Hierbei gibt es Monitore, die ein gemischtes H-Sync/V-Sync-Signal (auch Composite-Sync genannt) auf einer Leitung erwarten. Für diesen Fall müssen Sie einen Jumper auf der Picasso II umstecken.



Sie finden diesen Jumper mit der Bezeichnung JP 301 rechts unten, direkt neben dem VGA-Konnektor, der mit dem Amiga verbunden ist. Normalerweise ist dieser Jumper nicht gesteckt. Bitte stecken Sie den Jumper so, daß beide Kontakte miteinander verbunden werden.

Beim Erstellen von eigenen Auflösungen mit PicassoMode ist bei gestecktem Jumper unbedingt darauf zu achten, daß die H-Sync- und V-Sync-Polarität immer auf Negative steht.

Sonst wird der Monitor entweder nichts fangen oder aber sogar die normalen Zeilen für Sync-Impulse halten (und umgekehrt), was nicht jeder gute Monitor mitmacht.

## F. RAM-Aufrüstung

Die Picasso II gibt es schon etwas länger in einer 1MByte- und einer 2MByte-Ausführung. Die Aufrüstung ist zwar denkbar einfach, man sollte jedoch auf ein paar Dinge achten. Sollten Sie sich die Aufrüstung nicht selbst zutrauen, setzen Sie sich bitte mit Ihrem Händler in Verbindung oder schicken Sie uns – nach vorheriger Absprache – die Karte zur Aufrüstung zu.

### F.0.1. Bausteine-Typen

Zur Aufrüstung können nur Bausteine des Typs 256 × 4 Verwendung finden. Die minimale Zugriffszeit sollte mindestens 60 Nanosekunden betragen. Die RAMs müssen im Fast Page Mode laufen (keine Static Column RAMs).

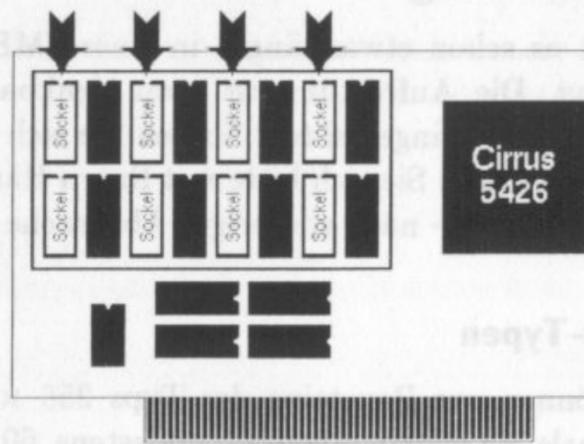
Folgende Typen sind derzeit freigegeben, für die Funktion anderer RAM-Typen können wir keine Funktionsgarantie übernehmen:

Fujitsu MB81C4256A-60p  
Mosel-Vitelec V53C104HP45

Bei Bausteinen anderer Hersteller sollte man unbedingt auf die Zeitanzeige – 60 Nanosekunden-Bausteine – und den Typ achten. Das Timing der Karte ist hochgradig optimiert und es kann zu Problemen kommen, wenn die Bausteine sich nicht exakt an die Spezifikationen halten. Deshalb setzen Sie bitte nur RAMs ein, die von uns freigegeben wurden.

Die Bausteine sind in die noch leeren Sockel auf der Karte links oben zu stecken. Bitte achten Sie darauf, daß die Einkerbung (oder der eingravierte runde Punkt) in die gleiche Richtung wie bei den bereits eingesetzten RAMs zeigen. **Wenn Sie die RAMs falschherum in die Sockel stecken, zerstören Sie damit die Bausteine und beschädigen wahrscheinlich auch die Karte.**

*Achten Sie auf die Einkerbung*



*Achten Sie beim Aufrüsten auf die Einkerbung*

Bevor Sie die Picasso II oder auch nur andere Komponenten im Rechner berühren, fassen Sie bitte eine metallische Fläche im Rechner oder einen Heizungskörper an. Sonst kann es zu einer Aufladung über die RAM-Bausteine kommen, die diese möglicherweise schädigt.

Die RAM-Bausteine lassen sich nicht so einfach in die Sockel drücken. Die Beinchen sind etwas mehr gespreizt, als nötig wäre. Dies hat seinen Sinn darin, daß die Beinchen der Bausteine etwas unter mechanischer Spannung stehen, wenn sie in den Sockeln sitzen und somit nicht so leicht herausfallen können.

Am besten biegt man die Beinchen eines ICs ein wenig nach innen, in dem man eine Reihe ganz leicht in einen Sockel setzt, schräg drückt und dabei die andere Reihe in die Fassung drückt. Anschließend muß das IC fest in den Sockel gepreßt werden (richtig fest drücken, es darf dabei auch ein wenig knacken).

Nach dem Einsetzen kontrollieren Sie bitte, ob alle Beinchen des ICs korrekt in der Fassung sitzen. Sollte dies nicht der Fall sein, benötigen Sie einen breiten Schraubendreher (kein Kreuzschlitz-Dreher!), um das IC wieder aus der Fassung zu holen. Setzen Sie dazu den Schraubendreher zwischen IC und Sockel an und schieben Sie ihn bei ganz leichtem Hin- und Herdrehen zwischen das IC und den Sockel. Aber nur schieben, nicht hebeln!

Wenn Sie den Schraubendreher anheben, verbiegen Sie die Beinchen des ICs. Deswegen den Schraubendreher nur schieben (nehmen Sie dazu einen wirklich großen Schraubendreher mit einer Klingenbreite von ungefähr 5mm).

Anschließend versuchen Sie das (wahrscheinlich) verknickte Beinchen wieder geradezubiegen und erneut in die Fassung zu setzen.

Statt die Beinchen beim Einsetzen passend ein wenig zu verbiegen, können Sie dies auch vorher auf einem Tisch tun, in dem Sie das IC auf die Seite legen – alle Beinchen einer Seite liegen gleichzeitig auf dem Tisch – und dann alle Beinchen einer Seite ein wenig nach innen biegen.

Achten Sie unbedingt darauf, daß die Kerbe (oder ein kleiner runder Kreis) auf der gleichen Seite zu sehen ist, an der auch die Kerbe im Sockel vorhanden ist (oben bei einer Platine, an der unten die Kontakte für den Zorro-Bus zu sehen sind).

## G. Glossar

Nicht jedem ist jeder Fachbegriff geläufig oder gerade in Erinnerung (auch beim Handbuch-Schreiben lernt man einiges!). Um Sie nicht im Regen stehen zu lassen, haben wir dem Handbuch ein kleines Glossar angefügt, in dem Sie hoffentlich den Begriff finden, den Sie suchen.

Das Glossar ist alphabetisch aufgebaut. Wenn Sie Begriffe im Index suchen und dort eine Seitennummer in *kursiver* Schrift aufgeführt ist, bezieht sich diese Seitenzahl auf einen Begriff im Glossar.

### AmigaGuide

Bezeichnung für das Hilfesystem von Commodore. Es erlaubt dem Benutzer, auf speziell markierte Begriffe zu klicken und sich Informationen zu Fachbegriffen holen zu können.

ChangeScreen unterstützt dieses System.

### Animation

Andere Bezeichnung für eine schnelle Abfolge von Bildern, landläufig als Film bekannt. Diese stellen durch große Datenmengen hohe Ansprüche an Speicher und Rechengeschwindigkeit.

### Auflösung

Die Anzahl der Punkte auf dem Bildschirm in waagerechter und senkrechter Richtung. Man spricht von hoher Auflösung, wenn man viele Pixel in beiden Eichtungen darstellen kann. Die Angabe einer Auflösung ist zum Beispiel: 800 x 600 Punkte.

### Bildwiederholfreq.

Anzahl der Bilder pro Sekunde, die der Monitor pro Sekunde auf der Mattscheibe erzeugt. Je höher dieser Wert ist, um so weniger *flackert* das Bild für das menschliche Auge. Erst ab einer Bildwiederholfrequenz von 70 Hz ist ein ergonomisches Arbeiten möglich.

### Blanker

Andere Bezeichnung für Bildschirmschoner: ein Programm, daß nach einer gewissen Zeit

### Blitter

den Bildschirm dunkel schaltet, um ein Einbrennen zu verhindern.

Ein besonder Baustein im Computer oder auf einer Steckkarte, der darauf spezialisiert ist, schnell Bits zu kopieren, zu bewegen oder zu manipulieren. Der Amiga hat einen solchen Baustein standardmäßig eingebaut; dieser kann Linien ziehen, rechteckige Bereiche kopieren, aber auch Disketten-Rohdaten bearbeiten.

Auf der Picasso II dagegen sitzt ein Blitter, der vor allem auf das Kopieren von Rechtecken optimiert ist. Ihm hat man die hohe Scroll-Geschwindigkeit zu verdanken, die die Picasso II auszeichnet.

### Booten

Englisches Wort für das Starten des Rechners. Dies passiert automatisch, wenn man den Computer einschaltet. Man kann diesen Vorgang aber auch erzwingen, wenn man im normalen Betrieb die Tasten CTRL AMIGA AMIGA drückt.

### Button

Andere Bezeichnung für ein einfaches Gadget

### ChipMem

Speicherbereich, in welchem die speziellen Amiga-Chips zugreifen können. Die eingebaute Amiga-Grafik benötigt zum Beispiel ChipMem, um einen Screen zu speichern. Bei den neusten Amiga-Modellen ist das ChipMem auf zwei MByte begrenzt.

### CLI

siehe Shell

### Commodity

Eine spezielle Art von Programm, was stets im Hintergrund läuft und oft auf bestimmte Tastendrucke, Mausbewegungen oder Zeitpunkte wartet. Diese Programme können mittels des mitgelieferten Programms **Ex-Change** gesteuert und beendet werden.

**CPU**

Zentraler Prozessor des Rechners. Das ist der Chip, der die Arbeit im Computer erledigt. Er rechnet, verschiebt und verändert Daten, wertet Tastendrücke und Mausbewegungen aus. In Amiga-Computern werden Prozessoren der Firma Motorola eingesetzt, wie zum Beispiel die 68000, 68020, 68030 oder 68040.

**C-Sync**

Ein Signal, daß sowohl die H-Sync- als auch die V-Sync-Impulse beinhaltet und dafür nur eine Leitung zum Monitor benötigt. Wird beim Einsatz eines BNC-Kabels verwendet.

**Cycle-Gadget**

Eine besondere Form eines Gadgets. Klickt man auf solch ein Gadget, bekommt es einen neuen Inhalt (Text). Klickt man es erneut an, wechselt der Text wiederum. Dies passiert solange, bis man wieder bei dem ersten Text angelangt ist. Man *cycled* durch die verschiedenen Zustände des Gadgets.

**DPaint**

Name des wohl bekanntesten Malprogramms für den Amiga.

**DRAM**

Allerwelts-Speicher. Wenn man von RAM spricht, meint man fast immer DRAM. Dies ist preiswerteste Form von RAM, leider auch die langsamste.

**Farbtiefe**

Kompliziert ausgedrückt: der Logarithmus zur Basis 2 von der Anzahl der möglichen Farben. Oder  $2^{\text{Farbtiefe}} = \text{Farbenanzahl}$ . Bei einer Farbtiefe von 4 hat man zum Beispiel die Auswahl aus 16 Farben, bei einer Farbtiefe von 8 die Auswahl aus 256, bei einer Farbtiefe von 24 die Auswahl aus 16777216 Farben.

**Flickerfixer**

Steckkarte, die das Flackern bei einer Interlace-Darstellung beseitigt (siehe Interlace). Ein Flickerfixer ermöglicht allerdings

keine höhere Auflösung oder mehr Farben.

**Floppy**

Anderes Wort für Diskette

**Font**

Bezeichnung für eine Schriftart, in der Buchstaben gedruckt oder angezeigt werden.

**Gadget**

Element in Fenstern, auf das man klicken und damit Aktionen auslösen kann.

**Guru**

Eigentlich die Bezeichnung für einen Meister der asiatischen Meditationskunst. Beim Amiga ist hiermit die rot oder gelb blinkende Fehlermeldung nach einem Absturz des Rechners gemeint.

**H-Sync**

spezieller Impuls, den eine Grafikkarte an den Monitor schickt, um das Ende einer Bildschirmzeile anzuzeigen.

**Hardware**

Alles am Computer, was man anfassen kann. Dazu gehören der Computer selbst, die Festplatte, die Tastatur, die Maus, aber nicht Programme (siehe Software)

**Icon**

Bezeichnet ein kleines, stilisiertes Bild, auf das man klicken und eventuell mit der Maus auch verschieben kann. Heißt manchmal auch Piktogramm.

**Installation**

Damit etwas korrekt funktioniert, muß die Hard- und Software so eingerichtet werden, daß jedes Teil an seinem Platz ist. Diesen Vorgang des Einrichtens nennt man Installation.

**Interlace**

englisch für *verflechten*, *verweben*. Interlace ist die Bezeichnung für eine Darstellungsverfahren, bei dem der Bildschirmstrahl ein komplettes Bild nicht in einem Durchgang von links oben nach rechts unten aufbaut, sondern dafür zwei Durchgänge benötigt. In jedem der zwei Durchgänge baut er dabei nur

ein Halbbild auf, in welchem jede zweite Zeile fehlt.

Dieses Verfahren hat den Vorteil, daß man größere Auflösungen darstellen kann, allerdings fängt das Bild für das menschliche Auge gesehen deutlich zu flackern an.

### Intuition

Der Teil des Betriebssystems der Screens, Fenster, Gadgets und Icons erzeugt und verwaltet.

### Jumper

Eine kleine, meist bunte Steckbrücke, die man auf zwei Kontakte auf einer Platine stecken kann. Mit solch einem Jumper kann man das Verhalten einer Steckkarte beeinflussen (siehe auch Seite 101).

### kompatibel

laut Fremdwörterduden: miteinander vereinbar, zusammenpassend.

Etwas ist kompatibel zu etwas anderem, wenn es sich genauso verhält wie das andere, dabei aber eventuell bessere Eigenschaften besitzt.

### Komprimierung

Um Platz zu sparen, hat man einige Verfahren entwickelt, um Dateien *zusammenzuquetschen* – zu komprimieren. Diese belegen nach der Komprimierung häufig nur noch die Hälfte oder ein Drittel des ursprünglichen Platzes. Leider hat das den Nachteil, daß man eine Datei erst wieder entkomprimieren muß, bevor man des Inhalt anschauen oder weiterverarbeiten kann.

### Library

Eine Sammlung von Funktionen, die Programme gemeinsam nutzen können. Auf dem Amiga liegen diese meist als Datei in dem Verzeichnis LIBS: vor und werden erst bei Bedarf in den Hauptspeicher geladen.

### Listview

spezielles Gadget in Form einer Liste, aus der

man eine Zeile auswählen kann.

### Multiscan

steht für *mehrfrequenzfähiger* Monitor. Monitore dieses Typs sind in der Lage, verschiedene Auflösungen in verschiedenen Bildwiederholungsfrequenzen darzustellen. Sie werden gelegentlich auch als MultiSync-Monitore bezeichnet. *MultiSync* ist allerdings ein geschützter Begriff der NEC Corporation.

### Overscan

Randbereich des Video-Bildes, der von der Amiga-Grafik noch genutzt werden kann.

### Packer

Ein Programm, mit dem man Dateien komprimieren kann; siehe Komprimierung

### Picasso

Pablo, spanischer Maler, Bildhauer, Grafiker geboren in Malaga 25.10.1881, gestorben 18.4.1975. 1904 in Paris niedergelassen, nach Bildern im kubistischen und klassizistischem Stil bekannt durch Werke mit grellen Farben und starken Verzerrungen (Hauptwerk dieser Zeit *Guernica*).

### Pixel

andere Bezeichnung für einen Bildpunkt

### Pixeltakt

siehe Videobandbreite

### Plane

Der Amiga speichert seine Bildschirm-Inhalte in sogenannten Planes. Hierzu werden die Bildinformationen in Schichten gespeichert, als hätte man mehrere durchsichtige Folien übereinandergelegt. Hätte man beispielsweise eine Folie mit gelben Punkten, eine mit blauen und eine mit roten Punkten übereinandergelegt, könnte man Bilder mit bis zu acht verschiedenen Farben darstellen.

### Preference

englisches Wort für Voreinstellung. Mit den sogenannten Voreinstellern des Betriebssystems kann man das Aussehen der Fenster, der Zeichensätze, das Datum und die bevorzugte Sprache einstellen.

- RAM** Random Access Memory. Bezeichnung für schnellen Hauptspeicher
- Requester** Andere Bezeichnung für Fenster. Ursprünglich war ein Requester etwas weniger als ein Fenster. Mittlerweile hat sich dieser Unterschied allerdings verwischt.
- Software** Alle Programme, Treiber und Libraries. Im Gegensatz zur Hardware also alles, was man nicht anfassen oder direkt sehen kann.
- Screen** Englisch Wort für Bildschirm.
- ScreenMode** Bezeichnung einer Auflösung. Dies kann zum Beispiel *NTSC:HighRes Interlace* oder auch *PICASSO:800x600* sein.
- Scroll** Verunglimpfter Trinkspruch (*Skol*) eines leicht angeschlagenen Entwicklers auf einer Konferenz in den USA.
- Scrollen** englischer Begriff für das schnelle Verschieben eines rechteckigen Bereiches. Dies braucht man zum Beispiel, wenn man einen langen Text anzeigen läßt und bereits gelesener Text nach oben aus dem Fenster geschoben wird. Das nennt man dann Scrollen.
- Segmentierung** Ist kein Speicherbereich im Rechner mehr frei, um den Grafikkarten-Speicher am Stück einzubinden, so muß man die Picasso II segmentiert betreiben. Hierbei wird immer nur ein kleiner Teil des gesamten Video-Speichers in einem Speicherfenster eingeblendet. Möchte man also einen bestimmten Speicherbereich verändern, hat man vorher den entsprechenden Bereich in dem Fenster einzublenden. Diese Betriebsart kann man per Jumper erzwingen (siehe Seite 101).
- Shell** Ein Fenster, in dem man Kommandos eingeben und starten kann und in welchem Pro-

- gramme Ergebnisse ausgeben können. Eine Shell wird oft auch als CLI bezeichnet.
- Tooltype** Klickt man auf ein Icon und drückt man anschließend die Tasten **AMIGA i**, erscheint ein Fenster, in welchem man Parameter einstellen kann. Diese haben meist die Form **OPTION=<wert>** und werden Tooltypes genannt.
- Transferrate** Anzahl der Bytes, die man pro Sekunde transferieren - bewegen oder kopieren - kann.
- TrueColor** Steht für eine Darstellung auf dem Monitor, bei der man für jeden Bildpunkt die Farbe frei aus 16 Millionen Farben wählen kann. Da dies mehr verschiedene Farben sind, als das menschliche Auge noch differenzieren kann, spricht man von Echtfarbdarstellung, oder eben neudeutsch von TrueColor.
- VRAM** Speicher, der speziell für Grafikkarten ausgelegt ist. Sehr schnell und noch sehr teuer.
- V-Sync** spezieller Impuls, den eine Grafikkarte an den Monitor schickt, um das Ende eines kompletten Bildes anzuzeigen. Nach einem solchen V-Sync beginnt der Bildschirmstrahl von neuem, links oben das nächste Bild aufzubauen.
- Video** kommt aus dem lateinischen und heißt eigentlich *ich sehe*. Das Wort kommt meist in allen Zusammenhängen vor, wo es darum geht, mittels Elektronik Bilder aufzuzeichnen oder darzustellen.
- Video-Slot** Ein Steckplatz, der speziell für Karten mit Video-Fähigkeiten gedacht ist. Dieser wird zum Beispiel von einem Genlock benötigt. Die Picasso II braucht den Video-Slot nicht.

- Videobandbreite** Menge an Bytes die pro Sekunde ausgelesen werden müssen, um in ein Monitorbild umgewandelt zu werden. Dies ist meist ein Wert in MHz (MegaHertz).
- Viewer** Programm zum Anzeigen von Bild-Dateien
- VGA** Virtuuell Graphic Adapter. Bezeichnung von IBM für eine spezielle Art von Grafikkarten, die bei IBM-kompatiblen Computern verbreitet sind.
- Workbench** Oft meint man mit einer Angabe wie **Workbench 2.0** das Betriebssystem des Amigas. Die Nummer hinter **Workbench** gibt dabei die aktuelle Version an. Je höher die Nummer ist, um so neuer ist das Betriebssystem. Ab Version 3.0 können Sie mehr als 16 Farben auf Screens und Fenstern darstellen. Das Wort hat aber noch ein zweite Bedeutung: Es steht auch für die Oberfläche des Amigas mit Bildchen für Programme und Verzeichnisse, mit Fenstern und der Möglichkeit, Icons in Fenster fallen zu lassen und alle Aktionen mit der Maus zu erledigen. Das Programm, was diese Möglichkeiten dem Benutzer eröffnet nennt sich ebenfalls **Workbench**. Nicht immer lassen sich beide Bedeutungen auseinanderhalten, was manchmal aber auch gar nicht nötig ist.
- Zeilenfrequenz** Die Anzahl der Bildschirmzeilen, die der Elektronenstrahl im Monitor pro Sekunde überstreicht. Die ist meist ein Wert der Größe *kHz* (Kilohertz).
- Zoll** Einheit wie zum Beispiel Zentimeter. Ein Zoll entspricht genau 2,54cm und wird gerne bei der Größenangabe von Monitoren verwendet. Dabei hat ein 14-Zoll-Monitor eine