

Poseidon-Anleitung

Michael Böhmer and Chris Hodges

COLLABORATORS

	<i>TITLE :</i> Poseidon-Anleitung		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Michael Böhmer and Chris Hodges	August 26, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Poseidon-Anleitung	1
1.1	Anleitung zum Poseidon USB Stack	1
1.2	Rechtliche Hinweise...	2
1.3	Beschränkungen der Demoversion	4
1.4	Einführung in Poseidon...	6
1.5	Systemvoraussetzungen...	9
1.6	Installationsanleitung...	10
1.7	Arbeiten mit Poseidon...	11
1.8	Arbeiten mit Trident...	11
1.9	Trident General Panel	13
1.10	Trident Hardware Panel	13
1.11	Trident Devices Panel	14
1.12	Trident Classes Panel	15
1.13	Trident Config Panel	16
1.14	Poseidon Shell-Kommandos	16
1.15	AddUSBHardware	17
1.16	AddUSBClasses	18
1.17	PsdStackloader	18
1.18	PsdDevLister	19
1.19	PsdErrorLog	19
1.20	PsdRestart	20
1.21	PencamTool	20
1.22	Pencam Treiber für VHI Studio	21
1.23	DRadioTool	22
1.24	UproarTool	23
1.25	Klassentreiber...	24
1.26	Der Hub Klassen-Treiber	25
1.27	Der HID Klassen-Treiber	26
1.28	Der Bootmouse Klassen-Treiber	35
1.29	Der Bootkeyboard Klassen-Treiber	36

1.30	Der Printer Klassen-Treiber	37
1.31	Der Mass Storage Klassen-Treiber	39
1.32	Der Communication Device Class Treiber für USB Modems	40
1.33	Der Klassentreiber für PL2303	41
1.34	Der RawWrap Klassentreiber	41
1.35	Hardware-Treiber	43
1.36	MorphOS Input Device V50	43
1.37	Der Boot Menu Patch	44
1.38	LoadModule und PsdLoadModule	44
1.39	RomTags - Wie mache ich Poseidon resetfest?	45
1.40	Bekannte Fehler	46
1.41	Ein Dankeschön geht an...	46
1.42	Die Zukunft	48
1.43	Fragen und Antworten (FAQ)	48
1.44	Entwicklerunterlagen	49
1.45	Version History	49
1.46	Kontakt zum Autor	60

Chapter 1

Poseidon-Anleitung

1.1 Anleitung zum Poseidon USB Stack

```

Willkommen zu
V/\V.      /\
mMMnw,    || []
(o o)W    () || ||
| / |Mw   || ||//
\ -'_/mw  \\||/
~%%/WM"   \||
~\ ~\     _||
|"( " \ ()/\ " \ ()/"_ ) |"( _ ) ) |"( " \ ()/\ " \ ( _ )/" ) /" ) " \ / _ )O
| ) ) )/" \ \ ( _/" \ / | )_ ( ( | )_ ) /" \ \ / /|/ / \ \ \ / , |O
| ( _ ) / ( ( _ \ ) _ \ \ | ( _ ) ) | ( _ ) | ( ( _ \ ) / /" / / | \ ' _ |O
| | _ \ / / /" \ / ) | " )_ ( ( | ) " ) \ / // /|/ / . | / \ _ / ||
| _ | ( _ ) \ / _ / ( _ _ / | _ ( _ ) ) | _ ( _ / . \ _ / ( _ / ( _ / . : : | ||

|" _ \ Poseidon -- Der göttliche USB Stack für Amiga Computer
| ( _ ) Version: 2.2 (14.12.03)
| _ ( Entworfen und entwickelt von Chris Hodges.
|" ( _ )
| _ / Copyright ©2002-2003
Chris Hodges
. Alle Rechte vorbehalten.

WARNUNG: Bitte lesen Sie die
rechtlichen Hinweise
'
bevor Sie mit dieser Software arbeiten!

-----

Rechtliche Hinweise

Beschränkungen der Demoversion

Einleitung

Voraussetzungen

```

Installation

Arbeiten mit dem Poseidon Stack

Arbeiten mit Trident

Klassen-Treiber

Hardware-Treiber

Shell-Kommandos

Bekannte Fehler

Fragen und Antworten

Die Zukunft

Entwicklerunterlagen

Ein Dankeschön geht an...

Kontakt zum Autor

Version History

Diese Anleitung wurde freundlicherweise von Michael Böhmer ←
aus dem

Englischen übersetzt. Vielen Dank.

1.2 Rechtliche Hinweise...

Urheberrechtshinweise

Die Software "USB Stack Poseidon" (Hauptprogramm, Klassen-Treiber, Hardware-Treiber, Trident, zusätzliche Tools und andere Teile der Software in diesem Paket) sowie die Anleitung wurden von Chris Hodges geschrieben und sind urheberrechtlich geschützt.

Kein Teil dieses Programms oder der Anleitung darf in irgendeiner Art und Weise verändert werden (einschliesslich Editieren, Verändern, Nachprogrammieren, auszugsweiser Verwendung, Packen, Disassemblieren usw.), ausdrücklich ausgenommen einer Archivierung. Es ist insbesondere illegal, Keyfiles zu verbreiten, modifizieren oder zu erzeugen oder mit anderer Software Ausstattungsmerkmale freizuschalten, die ohne Lizenz oder persönliches Keyfile nicht zugänglich sind.

Das "Application Programmers Interface" (API) des "USB Stacks Poseidon" ist urheberrechtlich geschützt. Software, die es erlaubt, Klassen- oder Hardwaretreiber ohne das ganze Poseidon-Paket zu benutzen, verstößt gegen diese Lizenzvereinbarung und den Schutz geistigen Eigentums.

Namen oder Produkte, die in dieser Anleitung verwendet werden, können

registrierte Markennamen oder Produkte der entsprechenden Firmen sein, auch wenn sie in dieser Anleitung ohne weitere Kennzeichnung benutzt werden.

Disclaimer

Diese Software wurde im täglichen Betrieb intensiv getestet und hat sich als stabil erwiesen. Der Autor kann in keiner Weise verantwortlich gemacht werden für Datenverluste, Schäden an Hard- und oder Software, Einflüsse auf die Gesundheit der Menschen weltweit oder Ihr Liebesleben, die direkt oder indirekt aus der Benutzung dieses Programms hervorgehen könnten.

Der Autor behält sich ausdrücklich das Recht vor, ohne weitere Bekanntmachung Änderungen an Software und/oder Anleitung vorzunehmen.

Modifikationen an dieser Software und/oder dem Keyfile können zu unerwarteten Ergebnissen führen. Der Autor ist für Schäden jedweder Art, die aus solchen Modifikationen ergeben, in keiner Weise verantwortlich, unabhängig davon ob die Modifikationen absichtlich oder unabsichtlich vorgenommen wurden.

Verbreitung

Diese Software ist nicht frei vertreibbar, sofern dies nicht ausdrücklich anders vermerkt ist. Die Distributionsrechte verbleiben beim Autor. Es ist ausdrücklich verboten, dieses Produkt ohne vorherige schriftliche Einwilligung des Autors kommerziellen Softwarepaketen beizulegen.

Nochmals: Keine der Dateien in diesem Paket dürfen verändert oder weggelassen oder mit anderen Produkten ohne die ausdrückliche schriftliche Genehmigung des Autors mitgeliefert werden. Es dürfen weiterhin keine Dateien hinzugefügt werden, die nichts mit diesem Programm zu tun haben.

Demo-Version

Falls dieses Softwarepaket ausdrücklich als unlicenzierte Demoversion ohne Keyfile gekennzeichnet ist, darf diese Software in einem Zeitraum von bis zu 30 Tagen getestet werden. Nach Ablauf spätestens dieser Frist muss die Software entweder deinstalliert oder ein legales Keyfile erworben werden.

Die Demoversion ist frei verteilbar, solange keine Gebühren dafür erhoben werden, sie nicht in einem kommerziellen Produkt enthalten ist und es nicht anderslautend erklärt wird.

Lizenzvereinbarung für registrierte Software

Der Autor dieser Software gibt dem Lizenznehmer ein beschränktes,

nicht-exklusives Recht zur Benutzung des Programms "Poseidon USB Stack" und seiner zugehörigen Dateien auf einem einzelnen Rechner. Der Autor vergibt eine spezielle Identifikationsdatei, genannt Keyfile, an den Lizenznehmer. Dieses Keyfile befähigt den Lizenznehmer zur vollständigen Benutzung des Softwarepakets nach den Lizenzbestimmungen. Dieses Keyfile ist ausdrücklich nur für den persönlichen Gebrauch durch den Lizenznehmer bestimmt. Es ist nicht erlaubt, ein Keyfile zu vervielfältigen, vertreiben, verleihen oder verkaufen. Die Lizenz und das Keyfile selber werden ungültig, sobald das Keyfile auf irgendeine Art und Weise veröffentlicht oder Dritten zugänglich gemacht wird. Der Lizenznehmer muss selbst die nötigen Vorkehrungen treffen, dass die Datei nicht von ihm unabsichtlich veröffentlicht wird (z.B. als Teil einer anderen Datei, speziell einer Konfigurationsdatei).

Der Lizenznehmer wird für alle Schäden, die durch die Veröffentlichung eines Keyfiles entstehen mögen, haftbar gemacht. Dieses Produkt wird 'wie es ist' und ohne irgendeine Garantie, weder ausdrücklicher, unausgesprochener, noch gesetzlicher Art, inklusive ohne Einschränkung irgendeiner unausgesprochenen Garantie der Nicht-Verletzung von Copyrights, Handelbarkeit und die Eignung für einen speziellen Zweck, ausgeliefert. Das vollständige Risiko in Bezug auf den Gebrauch, seinen Resultaten und Leistungen liegt bei Ihnen und sollte das Produkt sich als fehlerhaft erweisen, so liegen die gesamten Kosten aller von Ihnen in Anspruch genommenen nötigen Serviceleistungen, Reparatur oder anderer Vergütungen bei Ihnen. Keinesfalls soll der Hersteller dieses Produkts oder seine Weiterverkäufer haftbar gemacht werden für Sach- und Personenschäden, Verlust der Nutzung oder andere indirekte, zufällige oder kausale Schäden, inklusive ohne Einschränkung irgendwelche Schäden für entgangene Gewinne, Unterbrechung der Berufstätigkeit oder Daten, die verloren gegangen oder ungenau berechnet wurden, auch wenn ich auf die Möglichkeit solcher Schäden hingewiesen wurde.

Diese Vereinbarung soll exklusiv durch die Gesetze der Bundesrepublik Deutschland geregelt werden.

Durch das Kopieren, Verteilen und/oder das Benutzen dieser Software zeigen Sie, dass Sie die obenstehenden Regeln akzeptieren.

1.3 Beschränkungen der Demoversion

Beschränkungen der Demoversion

Große Anstrengungen wurden gemacht, um den Poseidon USB Stack zu verwirklichen. Diese Mühen wurden unterstützt vom Hersteller und dem Vertreiber der Highway, Subway und Algor USB Karten E3B und auch bPlan, die verschiedene USB Geräte zu Verfügung gestellt haben, nicht zu vergessen die große Mengen Schokolade. Deshalb wird diese Software mit den Highway-, Subway- und Algor-Karten verkauft und dem Pegasos für dessen onboard-USB Chipsatz mitgeliefert.

Meiner Meinung nach wäre es ziemlich unfair gegenüber obengenannten, wenn andere USB-Kartenhersteller oder Hersteller von PCI Brückenkarten diese Software völlig umsonst bekämen. Denn alles, was sie dazu tun müssten, ist

einen Treiber für die offengelegte Hardwareschnittstelle zu schreiben. Allerdings wäre es ebenso unfair zu verlangen, dass alle Hersteller eine Lizenz erwerben müssen, auch wenn sie vielleicht eigene Software mit ihren Karten mitliefern.

Aus diesen Gründen muss Poseidon durch den Erwerb einer Einzelnutzerlizenz registriert werden, wannimmer keine OEM Lizenz für einen Hardwaretreiber verfügbar ist. Dieses Keyfile ist meiner Ansicht nach nicht sehr teuer (siehe unten).

Wenn die Lizenz fehlt, operiert Poseidon im Demomodus. In diesem Modus kann überprüft werden, ob der Stack den Anforderungen genügt und die Hardware funktioniert, aber die Software hört nach kurzer Zeit auf zu funktionieren, indem Sie "response timeout" Fehlermeldungen bekommen.

ACHTUNG: Nutzer von Highway, Subway und Algor, sowie des Pegasos müssen sich NICHT registrieren!

Bitte beachten Sie, dass inzwischen eine elektronische Registrierung über IOSpirit (<http://www.iospirit.com>) möglich ist. Dort werden z.B. auch Kreditkarten akzeptiert.

Registrierungsformular

An: Chris Hodges
Kennedystraße 8
D-82178 Puchheim
Germany

An: poseidon@platon42.de (im Falle einer Überweisung und E-Mail)

JA, ich will eine Einzellizenz für Poseidon erwerben.

Ich habe die Lizenzgebühr beigelegt (bitte ankreuzen):

- als Bargeld (EUR 25)
- als Scheck (innerhalb Deutschlands: EUR 23)
- als Scheck (außerhalb Deutschlands: EUR 27)
- per Banküberweisung (EUR 23) to
 - Konto : 359 68 63
 - BLZ : 700 530 70
 - Bank : Sparkasse Fürstentfeldbruck

Füllen Sie bitte die folgenden Zeilen aus:

Firma: _____

Name: _____

Straße: _____

PLZ, Ort: _____

Land: _____

E-Mail: _____

Tel.: _____

Das Keyfile wird an die obige Adresse geschickt per:

- E-Mail (keine Transport/Verpackungskosten)
- Diskette (fügen Sie EUR 5 hinzu für Porto/Verpackung)

Dies ist freiwillig:

USB Karte: _____

Kickstart: _____

Computer: _____

(CPU, RAM

Erweit.) _____

Wie bewerten Sie diese Software:

	sehr gut-1	2	3	4	5	6-ganz schlecht
Eigenschaften	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Installation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Anleitung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Preis/Leistung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Support	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spaß-Faktor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kommentare, Vorschläge, neue Ideen, Bugs...:

Durch das Ausfüllen und Abschicken dieses Registrierungsformulars akzeptieren Sie die Lizenzbedingungen.

(Ort, Datum)

(Unterschrift)

1.4 Einführung in Poseidon...

Einführung in Poseidon

Der USB Stack Poseidon ist eine Software-Lösung, die die Möglichkeiten des Universal Serial Bus (USB) und der Geräte mit USB-Anschluss (Mäuse, Tastaturen, Grafiktablets, Joysticks, Drucker, Scanner, Webcams, Digitalkameras, Flashkartenleser, Memorysticks, Netzwerkadapter, Zip- und Floppy-Laufwerke, Festplatten, Audioadapter sowie nicht ganz alltägliche Geräte wie GPS- und Fingerabdrucksensoren) dem Amiga zugänglich macht. Poseidon wurde als Lösung für alle Systeme entworfen und programmiert.

Das modulare Design von Poseidon erlaubt eine nahtlose Integration in das AmigaOS Betriebssystem. Es ist keine Anpassung bestehender Systeme (wie dem Linux USB Stack), sondern wurde unter Berücksichtigung der einzigartigen Möglichkeiten des AmigaOS entwickelt, die dieses Betriebssystem so effizient machen.

Es gibt keinerlei Beschränkungen bezüglich der verwendeten USB Controller Hardware, solange ein Hardwaretreiber dafür existiert -- die API zur Programmierung eigener Gerätetreiber ist öffentlich und gut dokumentiert.

Poseidon wird als OEM-Lizenz mit dem Highway/Algor Zorro II bzw. dem Subway Uhrenport USB Controller ausgeliefert.

Poseidon benutzt externe Klassen-Treiber, um Standard USB-Geräte zu unterstützen. Dies erlaubt wirkliches Plug & Play -- einfach das Gerät anstecken und ohne weitere Treiberinstallation loslegen. Die Spezifikationen für Klassen-Treiber sind ebenfalls sehr einfach zu verstehen und zu programmieren. Beispielsweise entstand die erste funktionsfähige Version des Maustreibers innerhalb weniger Stunden.

Hier gibt es eine Liste der bereits existierenden Klassen-Treiber

.

Programmierer von Anwendungsprogrammen können auch direkt mit bestimmten USB-Geräten kommunizieren, ohne das Klassen-Interface zu nutzen. Es wird empfohlen, klassenspezifische Geräte mit den entsprechenden Klassentreibern anzusprechen.

Als Benutzeroberfläche wird Trident mitgeliefert. Das vollständig MUI basierte Frontend erlaubt es in komfortabler Art und Weise, Start und Konfiguration der angeschlossenen USB-Geräte vorzunehmen. Zum Betrieb des USB-Stacks wird Trident jedoch nicht unbedingt benötigt. Zwei Shell-Kommandos reichen zur kompletten Nutzung des Stacks ohne GUI.

Poseidon kann durch Nutzung von
RomTags
resetfest gemacht werden.

Dadurch steht es bereits beim Warmstart zur Verfügung und erlaubt damit die Benutzung von USB Mäusen und Tastaturen im Bootmenu. Ein

Patch
ist für

das Boot Menu in Systemen <= OS3.9 notwendig, um das dort verwendete Hardwarehacking zur Abfrage der Maustasten zu ersetzen. Ohne Patch kann das Boot Menu nur mit der original Amiga-Maus aufgerufen werden.

Der Universal Serial Bus (USB) wurde 1996 vom USB Konsortium (Compaq, Intel, Microsoft und NEC) im PC Konsumerbereich eingeführt. USB1.0/1.1 richtet sich an Low Cost bzw. Mid Cost Ansprüche und erlaubt Transferraten von 1.5Mbit/s (LowSpeed) bzw. 12Mbit/sec (Fullspeed). Um mit dem neuen FireWire-Standard gleichzuziehen, wurde ein Highspeed-Modus mit 480Mbit/s (Rohdatenrate) im neuen Standard USB2.0 eingeführt.

USB besitzt eine baumartige Busstruktur. An der Wurzel des Baums sitzt der Root-Hub des Host-Computers. Geräte können an die Ports dieses Hubs angeschlossen werden. Durch Hinzufügen weiterer Hubs kann die Baumstruktur um weitere Äste und Zweige erweitert werden.

Für die Stromversorgung von USB-Geräten gibt es zwei Möglichkeiten: Entweder erfolgt die Versorgung über den Bus (bus powered, d.h. der Hub versorgt das Gerät mit Strom) oder das Gerät besitzt ein eigenes Netzteil (self powered). Der USB-Standard setzt allerdings Grenzen bei der Stromaufnahme: jedes Gerät darf maximal 500mA vom angeschlossenen Hub-Port ziehen. Dies mag ausreichend erscheinen, man darf aber nicht vergessen, dass dem bus powered Hub selber nur 500mA zur Verfügung angeschlossener Geräte zur Verfügung stehen. Die Verwendung von self powered Hubs wird (wo immer möglich) empfohlen.

Der nächste Teil ist etwas technisch, kann aber auch gut ausgelassen werden, wenn kein Interesse am internen Aufbau von USB-Geräten besteht.

USB-Geräte können mehrere Konfigurationen unterstützen (ich habe allerdings selbst noch keines gesehen).

Jede Konfiguration enthält ein oder mehrere Interfaces mit (möglicherweise) unterschiedlichen Klassendefinitionen. Verbundgeräte (z.B. eine kabellose Funktastatur mit Maus) können ein Interface für die Maus und ein weiteres für die Tastatur besitzen. USB definiert Standard-Klassen für bestimmte Anwendungen. Leider sind die Hersteller von USB-Geräten jedoch frei in der Wahl des verwendeten Protokolls -- es können auch herstellereigene Protokolle anstelle der Klassen-Protokolle verwendet werden. Interfaces können mehrere alternative Settings haben, die sich aber jeweils gegenseitig ausschließen (so kann z.B. ein USB-Drucker unidirektional oder bidirektional angesteuert werden).

Momentan sind Standard-Klassen und Unterklassen auf Geräte- und Interface Ebene definiert. Ein Gerät kann z.B. keine definierte Klasse haben, jedoch mehrere Interfaces, die zu unterschiedlichen Klassen gehören. Eine Tastatur könnte auch ein Interface für die normalen Tasten sowie ein weiteres für spezielle Applikationstasten unterstützen.

Jedes Interface besitzt üblicherweise einen oder mehrere Endpoints. Diese erlauben es jeweils, Daten zu senden oder zu empfangen.

Jetzt wird es wirklich technisch. Es gibt vier Typen von Endpoints, und damit auch vier unterschiedliche Arten von Datentransfers.

Kontrol-Transfers sind kurz (normalerweise Pakete mit 8 Bytes) und werden bidirektional zum Konfigurieren des Geräts bzw. für Befehle an das Gerät genutzt. Jedes Gerät muss mindestens einen Endpoint 0 haben, der immer ein Kontrol-Endpoint ist.

Interrupt-Transfers sind kurz (Pakete mit 1 bis 8 Bytes) und treten periodisch auf. Diese sehr zuverlässige Übertragungsart wird zur Abfrage von Ereignissen (wie bei Mausbewegung und Tastaturanschlag) genutzt.

Bulk-Übertragungen sind von mittlerer Größe (Pakete mit bis zu 64 Bytes), burstartig und ebenfalls sehr zuverlässig. Eine Bandbreitengarantie für diese Übertragungen ist jedoch nicht gegeben (d.h. der Transfer kann lange dauern, wenn der Bus mit anderen Übertragungen ausgelastet ist).

Isochron-Übertragungen sind normalerweise groß (Pakete mit bis zu 1023 Bytes) und werden für zeitkritische Datenübertragungen genutzt. Im Gegensatz zu den anderen drei Übertragungsmethoden bieten sie jedoch keine Fehlerüberwachung; sie werden meist für Echtzeit-Audio- bzw. Video-Daten genutzt.

1.5 Systemvoraussetzungen...

Systemvoraussetzungen

Systemvoraussetzungen:

- Kickstart 3.0 (V39) oder höher
- MC68020 oder größer
- ca. 200KB freier RAM-Speicher
- ein USB Hardwarekontrolller samt passendem Hardwaretreiber
- Geräte mit USB-Anschluss ; -)
- MUI für die Trident GUI

oder

- MorphOS 0.4 (V50) oder höher
- PPC603e oder größer
- ca. 300KB freier RAM-Speicher
- ein USB Hardwarekontrolller samt passendem Hardwaretreiber
- Geräte mit USB-Anschluss ; -)
- MUI für Trident GUI

Empfohlen:

- Kickstart 3.1 (V40) oder höher
- MC68040 oder größer
- ca. 1MB freier RAM-Speicher
- ein Highway/Subway/Algor USB Kontrolller
- USB-Geräte, die sich wirklich an den USB-Standard halten ; -)

oder

- Pegasos mit MorphOS
- PPC604 oder größer
- ca. 1MB freier RAM-Speicher
- Onboard USB Kontrolller
- USB-Geräte, die sich wirklich an den USB-Standard halten ; -)

Entwickelt auf:

- A4000TE/60, 100MB, 40GB HD, CDR, ZIP, CV64, FliFix, Highway, Subway, Algor, Unity, MelodyPro, Silversurfer.
-

1.6 Installationsanleitung...

Installationsanleitung

Bitte benutzen Sie das beiliegende Installerskript zur Installation der Poseidon Software.

Wenn Sie sich sehr gut mit Ihrem System auskennen, können Sie die Dateien auch per Hand kopieren:

- Copy Libs/poseidon.library TO LIBS:
Dies ist das Hauptprogramm mit dem USB-Stack.
 - Copy Devs/USBHardware/#? TO DEVS:USBHardware
Dies sind die
Hardware-Treiber
.
 - Copy Classes/USB/#? TO SYS:Classes/USB
Die beiliegenden [Klassentreiber]. KOPIEREN SIE KEINE ANDEREN DATEIEN IN DIESES VERZEICHNIS (z.B. Libraries oder Gerätetreiber). Dies wird fast sicher zum Systemabsturz führen, wenn Poseidon diese Dateien zu laden versucht.
 - Copy Tools/#? TO C:
-
 AddUSBClasses
 -
 AddUSBHardware
 -
 PsdErrorLog
 -
 PsdDevLister
 -
 PenCamTool
- Copy Trident#? TO SYS:Prefs

 Trident
 ist die Benutzeroberfläche für Poseidon. Trident benötigt MUI zum Betrieb. Starten Sie Trident mindestens einmal, um die Konfigurationsdatei ENVARC:PsdStackloader zu erzeugen.
 - Falls Poseidon beim Booten automatisch starten soll, fügen Sie die Zeile "ENVARC:PsdStackloader" in Ihrer S:User-Startup ein.
 - Installieren Sie LoadModule aus dem Archiv im "Extra" Verzeichnis, wenn Sie nicht bereits ein
 LoadModule
 in Ihrem C-Verzeichnis haben.
 - Copy PsdRomTag#? TO SYS:Utilities
Dieses kleine Shell-Skript übernimmt das Resetfest-Machen per
 RomTags
-

- .
- Copy Devs/input.device TO DEVS:
Fügen Sie die Zeile "LoadModule DEVS:input.device QUIET" in Ihrer Startup-Sequenz ein, am besten vor SetPatch. Wenn Sie THOR's LoadModule benutzen, sollte die eingefügte Zeile "LoadModule DEVS:input.device NOREBOOT QUIET" lauten.
 - Copy Storage/#? TO SYS:Storage
Dies ist ein exemplarischer DOSDriver für for
USB Massenspeicher
- .

1.7 Arbeiten mit Poseidon...

Arbeiten mit Poseidon Stack

Poseidon braucht hauptsächlich drei Dinge, um zu starten:

- A) Mindestens einen
Hardware-Treiber
 - B)
Klassen-Treiber
für die angeschlossenen USB-Geräte
 - C) die
Konfigurationsdatei
- Normalerweise startet der PsdStackloader automatisch auch die \leftrightarrow
Hardware- und
die Klassen-Treiber, falls diese in
Trident
eingetragen wurden.

Sobald diese drei Dinge installiert sind, können USB-Geräte an Ihrem System verwendet werden. Stecken Sie einfach Ihre USB-Geräte an die Ports Ihres Root hubs. Mäuse, Tastaturen und Drucker können sofort benutzt werden, Massenspeicher müssen vor der Benutzung noch gemountet werden (siehe auch die zugehörige

Klassen-Beschreibung
).

1.8 Arbeiten mit Trident...

Arbeiten mit Trident

Trident ist die zu Poseidon gehörenden graphische Benutzeroberfläche (GUI). Sie wird zur Konfiguration des USB-Stacks eingesetzt und benötigt MUI.

Trident besitzt fünf Karteikarten:

General

Hardware

Devices

Classes

Config

Informationsbereich

Im unteren Teil des Trident-Fensters werden Fehlermeldungen und andere Mitteilungen des Stacks aufgelistet. Der gesamte Informationsfluss zwischen Anwender und dem Stack selber kann dort eingesehen werden. Beim Einstecken eines USB-Geräts wird dort eine kurze Mitteilung angezeigt, ebenso wieder beim Ausstecken.

Die Art und Menge der angezeigten Informationen kann über das Gadget 'Information level' eingestellt werden.

Bereits aufgelaufene Informationen können durch das Gadget 'Flush error messages' aus dem Speicher entfernt werden.

Sie können auch verhindern, dass überhaupt Nachrichten erzeugt werden. Wie das geht, steht im
Konfigurationspanel

.

Online / Offline

Das Anklicken von des 'Offline'-Buttons hält den Stack an und setzt alle angeschlossenen USB-Geräte in den Powerdown-Modus. Die Hardware-Treiber werden aus dem Speicher entfernt. Der Neustart des Stacks ist durch Anklicken von 'Online' möglich.

Konfiguration

Trident lädt beim ersten Start automatisch eine bereits existierende Konfigurationsdatei und startet den USB-Stack.

Die jeweils aktuelle Konfiguration kann mit dem 'Save'-Knopf (unten rechts im Fenster) bzw. 'Save' oder 'Save as...' aus dem Menu (Punkt: 'Settings') abgespeichert werden. Die Konfigurationsdatei selbst ist eine ausführbare Datei (siehe auch

```
PsdStackLoader
).
```

Eine bereits gespeicherte Konfiguration kann über das Menu ('Load') geladen werden.

Registrierung / Lizenz

Falls Sie legal ein Poseidon Keyfile erworben haben, können Sie dieses aktivieren, indem Sie den Menüpunkt "Inject keyfile..." auswählen. Bitte beachten Sie unbedingt, dass danach in jeder Konfigurationsdatei, die Sie speichern, das Keyfile mitenthalten ist. Seien Sie deshalb außerordentlich vorsichtig, dass Sie diese Dateien nicht versehentlich verbreiten.

Die Lizenzinformationen, die im Keyfile gespeichert sind, können im Hardwaretreiber Informationsfenster eingesehen werden.

1.9 Trident General Panel

Trident General Panel

Hier gibt es noch nicht viel zu sehen, außer dem hübschen Poseidon-Logo und einigen Textzeilen aus Liedern von Aimee Mann. Ihre neue CD "Lost In Space" ist der absolute Oberhammer! :)

1.10 Trident Hardware Panel

Trident Hardware Panel

Hier können

```
Hardware-Treiber
für Poseidon hinzugefügt und entfernt
werden. Diese Seite sollte selbsterklärend sein.
```

Durch einen Doppelklick auf einen Hardwaretreibereintrag oder das Drücken auf den Info-Knopf öffnet sich ein weiteres Informationsfenster, das weitere Details zum Hardwaretreiber zeigt. In diesem Fenster werden auch Informationen zum verwendeten Lizenzmodell dargestellt.

Einige Hardwaretreiber für den USB Controller benötigen mehrere Einträge für unterschiedliche Units, damit alle Ports funktionieren.

1.11 Trident Devices Panel

Trident Devices Panel

Auf dieser Karteikarte werden alle derzeit angeschlossenen USB-Geräte angezeigt. Wann immer ein Gerät ein- oder ausgesteckt wird, sollte es (hoffentlich) hier erscheinen. Falls nicht, ist irgendetwas schiefgegangen. Zusätzlich ist hier erkennbar, welche Klassen-Treiber an welches Gerät gebunden sind; damit weiß man auch sofort, ob ein bestimmtes USB-Gerät vom Stack akzeptiert wurde oder nicht.

Mit dem 'Class Scan'-Knopf kann versucht werden, per Hand USB-Geräte ohne Klassen-Bindung an eine Klasse zu binden.

Der 'Release Binding'-Knopf entfernt alle Bindungen von einem USB-Gerät (sowohl Geräte- als auch Interface-Bindungen). Das USB-Gerät wird dadurch temporär unbenutzbar.

Ist ein Gerät bereits an eine bestimmte Klasse gebunden und unterstützt diese Klasse eine Konfigurationsoberfläche, so kann diese mit dem 'Settings' Knopf geöffnet werden. Dabei werden sowohl Geräte- als auch Interface-Bindungen berücksichtigt. Wollen Sie nur die GUI eines bestimmten Interfaces öffnen, so können Sie das über das Geräteinformationsfenster machen (siehe unten).

Ein Doppelklick auf ein Gerät bzw. Anwählen des 'Information'-Knopfs öffnet ein Fenster mit detaillierten Informationen über das jeweilige Gerät. Dort werden auch die einzelnen Interfaces angezeigt. Auch hier bietet sich wieder die Möglichkeit, einen 'Class Scan' durchzuführen, sowie einzelne Bindungen zu deaktivieren ('Release Binding'). Durch einen Doppelklick auf einem Interface oder einen Klick auf dem 'Settings' Knopf, wird ein Konfigurationsfenster der Klasse geöffnet, sofern diese das unterstützt.

Gewaltsame Bindungen (Forced Bindings)

Seit V1.32 von Poseidon und Trident V1.0 gibt es die Möglichkeit, ganze Geräte oder Interface-Bindungen an ganz bestimmte Klassen mit Gewalt an eine Klasse zu binden, auch wenn die Klasse das normalerweise nicht tun würde. Das kann für Klassentreiber nützlich sein, die sich an herstellerspezifische Interfaces binden oder an USB Geräte mit fehlerhaftem Interface Descriptor.

Jedoch bedeutet das nicht, dass eine Bindung mit jeder beliebigen Klasse auch wirklich eingegangen werden kann (diese Entscheidung obliegt immer noch der Klasse selbst). Falls das nicht unterstützt wird, bekommen Sie eine Fehlermeldung.

Um eine gewaltsame Bindung zu erstellen, wählen Sie den entsprechenden Geräteeintrag (im Devices-Panel) or den Interfaceeintrag (im Fenster mit den Gerätedetails). Drücken und halten Sie dann die rechte Maustaste und wählen aus dem Popup-Menü die entsprechende Klasse aus, an die das Gerät

oder Interface binden wollen. Bitte beachten Sie, dass diese Bindung nicht sofort aktiv wird, sondern möglicherweise erst ein lösen der aktiven Bindung ('Release Binding') und einen Klassenscan~('Class Scan') benötigt. Verwechseln Sie bitte auch nicht Geräte- mit Interfacebindungen -- die meisten Klassentreiber binden ohnehin nur an Interfaces (für weitere Details siehe die

Klassentreiberbeschreibung
)

Um eine gewaltsame Bindung wieder zu entfernen, wählen Sie einfach "None" (keine) in der Klassenliste im Popup-Menü. Seien Sie vorsichtig mit diesen Bindungen, da sie bei falschem Einsatz Ihre USB Geräte zeitweise nutzlos machen können.

Dass eine gewaltsame Bindungen eingegangen werden soll, wird ganz normal mit den Einstellungen mitabgespeichert.

Sie sehen bestimmt noch nicht den Sinn dieser gewaltsamen Bindungen, oder? Hier ein paar Beispiele:

- Bindungen an USB Scanner mit der `rawwrap.class`, um sie mit BetaScan nutzen zu können.
- Erzwingen von Bindungen zwischen HID Geräten zu `bootmouse/bootkeyboard` und umgekehrt.
- Zuweisen von MSD Geräten mit defekter Unterklasse/Protokoll (Sony?) zur `massstorage.class`.

1.12 Trident Classes Panel

Trident Classes Panel

Diese Karteikarte enthält eine Liste mit allen Klassentreibern, die dem USB Stack hinzugefügt wurden. Ein Druck auf den 'Dir Scan'-Knopf fügt alle verfügbaren Klassen-Treiber im Verzeichnis `SYS:Classes/USB` hinzu.

Klassen können manuell auch über den 'Add'-Knopf hinzugefügt bzw. über den 'Remove'-Knopf entfernt. Der 'Configure'-Knopf erlaubt benutzerdefinierte Einstellungen, falls die jeweilige Klasse dies unterstützt. Meistens befinden sich hier die Default-Einstellungen für Klassenbindungen.

Vorsicht: mit dem Entfernen der `hub.class` werden alle angeschlossenen USB-Geräte deaktiviert!

Eine Liste der fügaren Klassen finden Sie [hier](#).

.

1.13 Trident Config Panel

Trident Config Panel

Hier werden Konfigurationsmöglichkeiten für den Stack selbst und die vorhandenen Einstellungen eingeblendet.

Im Moment können folgende Dinge verändert werden:

SubTask Priority: Hier kann eingestellt werden, mit welcher Priorität alle vom Stack gestarteten Untertasks laufen sollen. Der Defaultwert ist 5, wenn Sie wollen können sie ihn aber auch auf 20 oder 21 stellen, damit z.B. die Mausbewegung durch keine anderen Tasks gestört wird.

Boot Delay: Erhöhen Sie dieses Zeitinterval, wenn Sie von Massenspeichern booten können möchten. Da der Stack 100% asynchron beim Booten gestartet wird, kann er nicht darauf warten, dass USB Geräte rechtzeitig bereit werden, bevor der Amiga das DOS bootet. Besonders ein USB Ziplaufwerk braucht lange Zeit um hochzufahren. Falls also ein USB Gerät nicht von sich booten lässt, so versuchen Sie, diesen Wert zu erhöhen. Anderenfalls können Sie den Wert bei 0 belassen.

Logging Options: Hier können Sie einstellen, welche Nachrichten überhaupt erzeugt werden sollen.

In der untersten Zeile wird außerdem angegeben, wieviel Speicher Poseidon im Moment in seinem internen Pool reserviert hat (es kann also sein, dass noch ein paar Bytes mehr verbraucht werden, als dort angegeben sind).

1.14 Poseidon Shell-Kommandos

Shell-Kommandos und weitere Software

Im Poseidon-Paket sind auch einige Shell-Kommandos enthalten. Sie sind vor allem für erfahrenere Anwender gedacht.

- AddUSBHardware
-> Hinzufügen von Hardwaretreibern
 - AddUSBClasses
-> Klassenscan durchführen
 - PsdStackloader
-> Laden des Stacks
-

- PsdDevLister
-> Ausgeben umfangreicher USB Systeminformation
- PsdErrorLog
-> Ausgabe eines Fehlerprotokolls
- PsdRestart
-> Neustarten des Stacks
- PencamTool
-> Vielseitiges Webcam-Programm
- pencam.vhi
-> Pencam Treiber für VHI Studio
- DRadioTool
-> D-Link/GemTek USB Radio Tool
- UproarTool
-> Tool für verschiedene MP3 Player
- PsdRomTag
-> Resetfestes Einbinden von Poseidon

1.15 AddUSBHardware

AddUSBHardware

Kommando : AddUSBHardware 1.5

Schablone: DEVICE,UNIT/N,QUIET/S,REMOVE/S,ALL/S

Beispiele:

```
AddUSBHardware DEVS:USBHardware/highwayusb.device
AddUSBHardware DEVS:USBHardware/highwayusb.device 1 REMOVE
AddUSBHardware REMOVE ALL
```

Zweck:

Fügt zum USB-Stack hinzu bzw. entfernt manuell USB Hardware-Kontroller. Nach dem Hinzufügen eines Hardware-Treibers findet automatisch ein Scan der verfügbaren Klassen statt.

Benutzung:

DEVICE - Argument mit dem (absoluten) Pfad des Hardware-Treibers, der hinzugefügt bzw. entfernt werden soll.

UNIT/N - optionale Unit-Nummer, falls mehrere Units unterstützt werden. Standardeinstellung ist Unit 0.

QUIET/S - unterdrückt die Ausgabe von Infos in der Shell.

REMOVE/S - versucht den Eintrag, falls vorhanden, zu entfernen.

Geben Sie hierbei exakt den Pfad/Dateinamen an, der beim Hinzufügen benutzt wurde.

ALL/S - Versucht, alle Units zu öffnen. Wenn REMOVE angegeben wurde, so werden alle Einträge entfernt -- der Stack geht offline.

1.16 AddUSBClasses

AddUSBClasses

Kommando : AddUSBClasses 1.3

Schablone: QUIET/S,REMOVE/S

Beispiele:

```
AddUSBClasses QUIET
AddUSBClasses REMOVE
```

Zweck:

Durchsucht entweder das Verzeichnis SYS:Classes/USB und fügt alle Klassen-Treiber dem Stack hinzu oder entfernt alle installierten Klassen-Treiber aus dem Stack.

Beachten Sie, dass die Klassen-Treiber aus dem Speicher entfernt werden, wenn die Haupt-Library geflusht wird (dies kann vorkommen, wenn kein Hardware-Treiber die poseidon.library geöffnet hat). Die korrekte Abfolge ist deshalb, erst AddUSBHardware und dann AddUSBClasses aufzurufen.

Benutzung:

QUIET/S - unterdrückt die Ausgabe von Infos in der Shell.
REMOVE/S - entfernt die Klassen-Treiber.

1.17 PsdStackloader

PsdStackloader

Kommando : PsdStackloader 2.1

Schablone: keine

Beispiele:

```
ENVARC:PsdStackloader
PsdLoadModule ENVARC:PsdStackloader
```

Zweck:

Dieses Programm (!) enthält die Konfigurationsdaten des USB-Stacks und kann durch einfachen Aufruf den Stack starten. Dabei werden intern

AddUSBHardware

```
    und
    AddUSBClasses
    mit allen in der gespeicherten
Konfiguration vorhandenen Hardware- und Klassen-Treibern aufgerufen.
PsdStackloader wird automatisch von
    Trident
erzeugt.
```

Dieses Programm kann (und sollte) als
 RomTag
 geladen werden,
falls der USB-Stack vor dem Boot-Vorgang gestartet werden soll.

1.18 PsdDevLister

PsdDevLister

Kommando : PsdDevLister

Schablone: keine

Beispiele:

```
PsdDevLister
```

Zweck:

Listet alle angeschlossenen USB-Geräte mit ausführlichen Informationen auf. Bitte senden Sie bei Bugreports stets die Ausgabe dieses Programms mit -- dies erleichtert es uns wesentlich, Ihr Problem nachzuvollziehen.

1.19 PsdErrorLog

PsdErrorLog

Kommando : PsdErrorLog

Schablone: keine

Beispiele:

```
PsdErrorLog
```

Zweck:

Gibt alle Informationen, Warnungen und Fehlermeldungen aus, die Poseidon angesammelt hat. Die ausgegebenen Meldungen werden automatisch aus dem Speicher entfernt; ein weiterer Aufruf von PsdErrorLog zeigt deshalb nur die in der Zwischenzeit aufgelaufenen neuen Meldungen seit dem letzten Aufruf an.

1.20 PsdRestart

PsdRestart

Kommando : PsdRestart

Schablone: keine

Beispiele:

PsdRestart

Zweck:

Dieses Programm wird hauptsächlich als RomTag eingesetzt. Es startet den Stack quasi neu, indem es alle Bindungen auflöst und danach einen Klassenscan durchführt. Der Zweck dieses Programms liegt darin, die Subtasks, die während des Bootens erzeugt wurden, nach dem Booten in vollwertige Prozesse umzuwandeln, die dann auch die dos.library benutzen können. Vermeiden Sie PsdRestart, wenn Sie von einem USB Massenspeicher booten möchten.

PsdRestart kann auch dazu verwendet werden, um Mäuse und Keyboards, die sich während des Bootvorgangs an die bootmouse bzw. bootkeyboard.class gebunden haben, an die hid.class überwechseln zu lassen.

PsdRestart ist teil des
PsdRomTag
Vorgangs.

1.21 PencamTool

PencamTool

Kommando : PencamTool 1.6

Schablone: TO/A, PICNUM/N, INTERVAL/N, UPTO/N/K, NOBEEP/S, GAMMA/K, SHARPEN/S,
TEXT/K, FONT/K, FONTSIZE/N/K, UNIT/N/K

Beispiele:

PencamTool Snap.ppm
PencamTool Snap.ppm 0 GAMMA 0.45 SHARPEN
PencamTool Movie%04ld.ppm INTERVAL 0 GAMMA 0.5
PencamTool Webcam.ppm GAMMA 0.45 SHARPEN TEXT "Platon's Cam"
FONT small.font FONTSIZE 6 NOBEEP
PencamTool Shotseries%03ld.ppm 0 UPTO 79 GAMMA 0.45 SHARPEN

Zweck:

Einfaches Shell-Tool zum Auslesen von Bildern aus USB-Webcams, die den STV680 Chip (Vendor ID = 0x0553, Product ID = 0x0202) beinhalten. Dazu

gehören Kameras aus der Aiptek Pencam Serie sowie einige andere billige Kameras. Die Bilder werden als Echtfarbgrafiken im Portable Anymap Format (PPM) gespeichert (für Konvertierungsmöglichkeiten siehe das NetPBM-Paket im Aminet).

Außerdem können Gammakorrektur, Weißabgleich sowie ein Schärfefilter für die Bilder eingestellt werden. Ein Bildtext (Schriftart frei wählbar) kann ebenfalls ins Bild eingeblendet werden (z.B. für die Nutzung als Webcam).

Benutzung:

- TO/A - Dateiname, unter dem das Bild gespeichert werden soll. Kann auch einen Format-String wie "%ld" enthalten (das 'l' nicht vergessen), um eine fortlaufende Zahl in den Dateinamen einzufügen, wenn die Option INTERVAL benutzt wird.
- PICNUM/N - Anzahl der aus dem Kamera-Speicher auszulesenden Bilder. 0 bezeichnet dabei das erste abgespeicherte Bild; die Angabe eines nicht existenten Bilds liefert Grafikmüll. Bei Weglassen dieses Arguments wird ein Bild aufgenommen und übertragen.
- INTERVAL/N - gibt die Pause zwischen zwei Aufnahmen in Ticks (50 Ticks entsprechen einer Sekunde) an. Nur sinnvoll ohne die Option PICNUM. Bei Betrieb der Kamera mit der Option INTERVAL kann PenCamTool über Ctrl-C beendet werden.
- UPTO/N/K - wenn dieses Argument angegeben wird, so wird eine ganze Reihe von Bildern aus der Kamera heruntergeladen. Dabei gibt der Parameter an, bei welcher Bildnummer gestoppt werden soll. Es sollte beachtet werden, dass der Dateiname einen Format-String wie "%ld" enthält, damit die Bilder nicht alle unter dem selben Namen geschrieben und damit überschrieben werden. Falls PICNUM nicht angegeben wurde, sondern INTERVAL, so gibt der Parameter an, nach welcher Bildnummer das Aufzeichnen der Bilder gestoppt werden soll.
- NOBEEP/S - schaltet den Piepston beim Download der Bilder ab.
- GAMMA/K - schaltet Weißabgleich und Gammakorrektur mit dem angegebenen Fließkommawert an. 0.45 ist erfahrungsgemäß ein guter Wert.
Falls nur ein Weißabgleich ohne Gammakorrektur gewünscht ist, so kann ein Wert von 1.0 angegeben werden.
- SHARPEN/S - schaltet einen hochoptimierten 5x5 Schärfefilter an.
- TEXT/K - fügt optional eine Zeile Text am unteren Rand des Bildes ein. Die Zeile wird abgeschnitten, falls sie zu lang ist. Es findet kein Umbruch statt.
- FONT/K - Schriftart für die Texteinblendung (z.B. xen.font). Die Standardschrift wird benutzt, falls dieses Argument fehlt.
- FONTSIZE/N/K - Größe der Schrift in Pixeln.
- UNIT/N/K - wählt die Kamera aus, falls mehrere angeschlossen sind. Standardeinstellung ist Unit 0.

1.22 Pencam Treiber für VHI Studio

Pencam Treiber für VHI Studio

Datei : Libs:VHI/pencam.vhi
Version: 1.2 (beta)

Zweck:

Hierbei handelt es sich um einen Treiber für die Pencam Kamera Serie für die Verwendung unter VHI Studio. Der Treiber erlaubt es, Schnappschüsse zu machen oder Bilder aus Kameras zu lesen, die den STV680 Chipsatz (Vendor ID = 0x0553, Product ID = 0x0202) benutzen. Dazu gehören Kameras aus der Aiptek Pencam Serie sowie einige andere billige Kameras.

Benutzung:

Der Treiber schnappt sich die erste freie Webcam, sobald VHI Studio gestartet wird. Das bedeutet, dass Poseidon bereits laufen muss, wenn VHI Studio aufgerufen wird.

Im VHI Studio haben Sie sowohl die Möglichkeit, Schnappschüsse zu nehmen, als auch Bilder aus dem Speicher der Kamera herunterzuladen, die zuvor mit der Kamera aufgenommen wurden. Sie können sogar einige Animationen aufzeichnen bei einer Rate von etwa 2-3 Bildern pro Sekunde. Das Format für einen Videostream des STV680 Chips ist nicht öffentlich dokumentiert und kann daher nicht unterstützt werden. Somit sind keine höheren Raten zu erwarten.

1.23 DRadioTool

DRadioTool

Kommando : DRadioTool 1.1

Schablone: ON/S,OFF/S,FREQ/K/N,SCAN/S,AUTO/S,SIGNAL/S,UNIT/N/K

Beispiele:

DRadioTool ON SCAN AUTO
DRadioTool FREQ 104000

Zweck:

Sehr einfaches Shell-Tool zum Kontrollieren eines USB-Radios des Herstellers D-Link bzw. GemTek. Es wird nur das Radio mit folgendem Vendor ID = 0x04b4 und Product ID = 0x1002 unterstützt.

Benutzung:

ON/S - Schaltet das Radio an.
OFF/S - Schaltet das Radio wieder aus.
FREQ/K/N - Einstellen der Frequenz auf den gewünschten Wert. Die Frequenz muss in Kiloherz angegeben werden. Nur Werte zwischen 87 MHz und 108 MHz werden unterstützt.
SCAN/S - Schaltet in den Scan-Modus. Der Scan beginnt bei 87 MHz, sofern FREQ nicht angegeben wurde und endet bei 108 MHz. Für jede gefundene Senderstation wird die Frequenz in KHz auf der Shell ausgegeben. Die letzte gefundene Station bleibt eingestellt. Der Scan kann mit Ctrl-C abgebrochen werden.
AUTO/S - nur bei Nutzung von SCAN interessant. Wenn eine Senderstation

gefunden wurde, gibt das Programm die Frequenz aus und wartet drei Sekunden. Der Nutzer kann dann mit Ctrl-C diesen Sender eingestellt lassen.

SIGNAL/S - setzt den Shell-Rückgabewert auf WARN (5), wenn auf der aktuellen Frequenz kein Trägersignal gefunden wurde oder OK (0) andernfalls. Damit lässt sich ein manueller Scan implementieren.

UNIT/N/K - wählt das Radio aus, falls mehrere angeschlossen sind. Standardeinstellung ist Unit 0.

1.24 UproarTool

UproarTool

Kommando : UproarTool 1.0

Schablone: DEVINFO/S, FILELIST=FL/S, UPLOAD=UL/K, DOWNLOAD=DL/K, DELETE=DEL/K, FORMAT/S, EXT/S, AS/K

Beispiele:

```
UproarTool DEVINFO
UproarTool FL
UproarTool UL AimeeMann-NightmareGirl.mp3 AS NghtMare.mp3
UproarTool DL 1 EXT
UproarTool FORMAT
UproarTool DELETE HormonBn.mp3
```

Zweck:

Dies ist ein Verwaltungsprogramm für MP3 Player, die einen bestimmten und bekannten Chipsatz von Samsung benutzen. Es gibt keine offiziellen und zugänglichen Informationen zum Protokoll, allerdings ist es ähnlich dem, das beim "Samsung Uproar SPH-M100" verwendet wird. Ich glaube, es sollte mit verschiedenen Implementationen der Player funktionieren. Jedenfalls tut es das mit meinem "Valencia MPX Player, den ich damals kurz vor meinem Urlaub in Neuseeland im Dezember 2002 bei Lidl für EUR 99 erworben habe. Das UproarTool bindet an jedes Gerät, das eine VendorID = 0x04e8 verwendet, nachdem die ProduktID wohl deutlich unterschiedlicher ausfällt. Falls Sie einen MP3 Player haben, der kein Mass Storage Gerät ist und die VendorID übereinstimmt, probieren Sie ruhig mal dieses Tool aus -- zumindest mal den DEVINFO Schalter und schicken Sie mir die Ausgabe zu.

Der Quellcode ist verfügbar für alle, die das Programm an ihren MP3 Player anpassen wollen.

Das Shell-Tool erlaubt all diejenigen Dinge, die auch die Windows Software erlaubt. Leider ist der MP3 Player recht empfindlich und gerät schnell aus der Synchronisation. Falls das passiert, stecken Sie ihn ab und wieder an oder probieren sie die Kommandos mehrmals, bis es funktioniert.

Mein MP3 Player hat 64 MB eingebauten Speicher und einen Slot für MMC/SD Karten. Um die Kommandos auf dem externen Speicher auszuführen, geben Sie einfach den EXT-Schalter mit an.

Benutzung:

- DEVINFO/S - Liefert einen ID String. Mein Valenica MP3 Player gibt:
"Device ID String: '2.1.0.10 20000329CPAD00'"
- FILELIST=FL/S - Listet die Dateien auf, die im internen oder externen Speicher liegen. Falls Sie nur Müll bekommen, probieren Sie, das Medium zu formatieren.
- UPLOAD=UL/K - Lädt die angegebene Datei auf den MP3 Player hoch (entweder in den internen oder in den externen Speicher, wenn Sie EXT angeben haben). Falls der Dateiname länger als 8+3 Zeichen ist, wird er entsprechend verstümmelt. Bei der Angabe des AS Schlüsselwortes wird stattdessen dieser Name benutzt.
- DOWNLOAD=DL/K - Lädt die angegebene Datei vom MP3 Player herunter. Die Datei kann entweder durch Angabe ihrer Tracknummer oder über ihren exakten Namen spezifiziert werden. Die heruntergeladene Datei wird dann im aktuellen Verzeichnis gespeichert. Alternativ kann durch das AS Schlüsselwort auch ein anderer Pfad und Name gewählt werden. Um eine Datei aus dem externen Speicher zu laden, nutzen Sie den EXT-Schalter.
- DELETE=DEL/K - Löscht die angegebene Datei aus dem Speicher des MP3 Players. Wie beim Download, kann auch hier wieder entweder die Tracknummer oder der Dateiname verwendet werden. Um eine Datei aus dem externen Speicher zu löschen, nutzen Sie den EXT-Schalter.
- FORMAT/S - Falls diese Option benutzt wird, wird der gesamte interne oder externe Speicher gelöscht (abhängig vom EXT-Schalter).
- EXT/S - Gibt an, ob ein Kommando auf dem internen oder externen Speicher ausgeführt werden soll.
- AS/K - Erlaubt die Angabe eines alternativen Dateinamens für das Hoch- oder Herunterladen von Dateien.

Bugs:

- Es gibt nicht gerade viel Fehlerbehandlung.
- Kommt leicht aus der Synchronisation.
- Erzeugt kein gültiges Datum beim Hochladen von Dateien.
- Die Wirkung einiger Felder im Protokoll ist unbekannt.

1.25 Klassentreiber...

Klassentreiber

Das Poseidon-Paket enthält momentan folgende Klassentreiber:

- - hub.class
 - für den Roothub und alle externen Hubs
 - - hid.class
 - für alle möglichen Eingabegeräte
-

```

-
    bootmouse.class
    für Mäuse und Grafiktablets mit "boot protocol"
-
    bootkeyboard.class
    für Tastaturen mit "boot protocol"
-
    printer.class
    für USB-Drucker
-
    massstorage.class
    für Standard-Massenspeicher (CF-Leser, ZIP usw.)
-
    cdcacm.class
    für USB Modems, die dem ACM Standard entsprechen
-
    rawwrap.class
    um herstellerspez. Schnittstellen verfügbar zu machen
-~
    serialpl2303.class
    für Prolific USB->Seriell-Adapter

```

Bitte beachten Sie, dass nur Geräte unterstützt werden können, die sich vollständig an die USB-Spezifikationen halten. Leider gibt es immer noch viele Hersteller, die ihre eigenen undokumentierten Protokolle statt der Klassen-Protokolle verwenden. Vermeiden Sie diese Geräte, da eine Unterstützung in Poseidon zumindest zweifelhaft ist. Treiberentwicklung ist sehr schwer, wenn keine Informationen zum Protokoll verfügbar sind.

Scanner und Digitalkameras müssen von Drittherstellern unterstützt werden, das sie normalerweise in die oben erwähnte Kategorie von USB-Geräten mit herstellerspezifischen Protokollen fallen.

Klassen-Treiber von Poseidon befinden sich im Verzeichnis `SYS:Classes/USB`. Andere Dateien haben in diesem Verzeichnis nichts verloren und können Abstürze beim Start von Poseidon bewirken.

Es gibt zwei unterschiedliche Arten von Klassen-Treibern: Gerätegebundene und Interfacegebundene. Da es in einem USB-Gerät (z.B. Funktastatur mit Maus) auch mehrer Interfaces geben kann, können sich auch mehrere Klassen an ein USB-Gerät binden.

Klassen können über
 Trident
 dem System hinzugefügt bzw. entfernt werden.
 Alternativ dazu kann auch das Shell-Kommando
 AddUSBClasses
 benutzt
 werden.

1.26 Der Hub Klassen-Treiber

hub.class

Klasse: hub.class 1.22
Bindet an: Gerät, Klassencode 9 (HUB)
Konfigurations-GUI: Keine

Dies ist die Wurzel allen Übels. Die hub.class kontrolliert den Root-Hub und alle angeschlossenen externen Hubs. Wann immer ein Gerät in einen Hub-Port gesteckt wird, kümmert sich diese Klasse um die Enumeration und Konfiguration des Geräts, wobei versucht wird, dem System automatisch die im Gerät verfügbaren Merkmale zur Verfügung zu stellen. Magischerweise wird von dieser Klasse ebenfalls das Entfernen von Geräten gehandhabt.

Externe Hubs fügen dem USB-System mehrere freie Ports hinzu oder erweitern die Möglichkeiten zur Aufstellung von USB-Geräten. Hubs können self powered oder bus powered sein; vermeiden Sie bus powered Hubs, falls Sie stromhungrige Geräte wie USB-Laufwerke anschliessen wollen. Andernfalls kann es passieren, dass diese USB-Geräte den Hub an seiner Arbeit hindern oder den gesamten Hub samt aller angeschlossenen Geräte abschliessen. Hubs sollten zwar Überstrom erkennen und sofort den entsprechenden Port abschalten. So die graue Theorie, die Wirklichkeit ist leider meist anders :-)

Einige Tastaturen und Monitore haben eingebaute Hubs. Normalerweise eine nützlich Sache, aber denken Sie an die Probleme mit der Stromversorgung, die oben diskutiert wurden.

Die hub.class kann als
RomTag
eingebunden werden.

1.27 Der HID Klassen-Treiber

hid.class

Klasse: hid.class 1.6
Bindet an: Interface, Klassencode 3 (HID), jede Unterklasse und Protokoll
Konfigurations-GUI: Für jedes Interface einzeln, sowie standard Klassen-Einstellungen

Die Übersetzung der Änderungen von HID 1.5 zu 1.6 sind leider einem Plattencrash zum Opfer gefallen. Sie werden später nachgereicht. Bitte verwenden Sie bis dahin die englische Anleitung für die fehlenden Details.

Nach Monaten harter Arbeit und über 300 KB Source Code darf ich Ihnen die erste Version der hid.class präsentieren. Es handelt sich dabei um einen generischen und sehr vielseitigen Klassentreiber für menschenbezogene Eingabegeräte (Human Interface Devices, HID): Also alle Geräte, die

zwischen Mensch und Maschine stehen, wie Grafiktablets, Joysticks, magische Teppiche, Baseball-Schläger, Telefone, Monitore, Barcode-Leser, Spannungsversorgungen und auch weniger gebräuchliche Dinge wie Mäuse und Tastaturen.

Ah, ja. Bevor ichs vergesse: Scrollräder werden unterstützt :)

Geräte können beliebig an- und abgesteckt werden, zu jedem Zeitpunkt.

Die hid.class besitzt eine sehr umfangreiche und komplexe GUI. Es gibt vieles, was man falsch machen kann -- aber keine Angst, ich nehme Sie jetzt an der Hand und leite Sie Schritt für Schritt durch.

Das neue

input.device

muss installiert sein, um einen 100% kompatiblen

Ersatz für Maus und Tastatur zu bekommen.

Die hid.class kann als

RomTag

installiert werden, ist aber erst nach dem

Booten verfügbar.

Globale Einstellungen

Die hid.class hat globale Einstellungen sowohl für alle als auch für einzelne Geräte. Die globalen Standardeinstellungen können durch Doppelklick auf den hid.class Eintrag geöffnet werden; die individuellen Einstellungen erreicht man über den 'Settings'-Knopf in der Device-Liste oder im Fenster mit den Detailinformationen des jeweiligen Geräts.

Die globalen Einstellungen enthalten:

Shell console window:

Beim Starten eines Programmes wird dieses Konsolenfenster als Ausgabe benutzt. Wenn dies nicht gewünscht ist, einfach NIL: eintragen.

Shell default stack:

Definiert die Größe des Stackspeichers für den Aufruf externer Programme. Falls dieser Wert zu niedrig eingestellt wird, kann es zu Abstürzen kommen.

Enable keyboard reset:

Erlaubt das Zurücksetzen des Rechners über die normale Tastenkombination Ctrl-Amiga-Amiga, falls dieser Schalter aktiviert ist.

Hijack reset handlers:

Sollten Reset-Handler für die Tastatur im System eingebunden sein, werden diese vor dem Rebooten aufgerufen.

Reset delay:

Wartezeit vor dem Reboot, falls (und nur falls) irgendwelche Reset-Handler im System gefunden wurden.

Die Tastatur-Einstellungen bieten:

Nicht viel im Moment. Die Tastatur-Belegung kann noch nicht geändert werden. Kommt aber noch.

Reports, Collections, Items, Actions

Das einzig wirklich Komplizierte an HID-Geräten ist das Format der Daten. Das USB Implementers Forum hat versucht, alles so vielseitig wie möglich zu machen, um jedes nur denkbare Eingabegerät unterstützen zu können. Aus diesem Grund werden Daten in Form eines Berichts geschickt. Diese sog. Reports sind aus einer oder mehreren Collections aufgebaut. Jede Collection wiederum kann ein oder mehrere Einträge (Items) umfassen (Items werden also nach logischen oder physikalischen Gesichtspunkten in Collections zusammengefasst). Items können entweder Variablen oder Arrays sein, wobei letztere meistens eine ganze Reihe sog. Usages (Einsatzbestimmungen) haben, erstere dagegen nur ein Usage. Kurz gesagt, ein Usage beschreibt, was ein Item repräsentiert.

In Poseidon können jedem Usage mehrere Actions zugeordnet werden. Die hid.class kennt einige (>1500) dieser Usages von sich aus und definiert sinnvolle Standard-Actions für einige davon automatisch. Ihre Tastatur und Maus wird also von sich aus funktionieren.

Wo führt uns das ganze jetzt hin? Hm. Fangen wir am besten mit den Items an. Jedes Item kann einen wohldefinierten Bereich von Werten annehmen. Items können unterschiedliche Typen haben: linear <-> nicht-linear, absolut <-> relativ, Variable <-> Array, um nur einige zu nennen.

Es gibt einen wichtigen Unterschied zwischen Variablen und Arrays. Variablen können nur einen Usage haben, die einen bestimmten Wert innerhalb eines beschriebenen Intervalls zugeordnet wird. Im Gegensatz dazu können Arrays mit einer Vielzahl von unterschiedlichen Usages verbunden sein, die in einem Array auftauchen -- oder eben auch nicht (also können die Werte für Usages in einem Array-Item nur 0 (nicht vorhanden) oder 1 (vorhanden) annehmen). Da Array-Items üblicherweise eine Vielzahl von Usages mit gleichem oder ähnlichem Verhalten definieren, besitzen diese Items einen zusätzlichen Standardeintrag. Dieser dient als als Vorgabe für alle definierten Usages, wenn (und nur wenn) kein anderes Verhalten für diese bestimmte Usage definiert ist. Wir werden später genauer auf dieses Thema zurückkommen.

Auslöser (Triggers)

Sie können so viele Actions für ein einzelnes Item definieren wie Sie wollen. Jede Action kann auf vier unterschiedliche Arten ausgelöst werden:

Trigger-Definitionen:

- Down: Wenn der Wert sich erhöht (d.h. die Taste gedrückt wird), löst dies die Action aus.
 - Up: Wenn der Wert kleiner wird (d.h. der Knopf bzw. die Taste losgelassen wird) triggert das die Action.
 - Any: Eine Kombination aus Up und Down Triggern, d.h. wannimmer sich
-

der Wert ändert, löst dies die Action aus.

- Always: Wenn ein neues Datenpaket mit einem gültigen neuen Datenwert empfangen wird (unabhängig vom Wert), triggert das die Action.

Also: Sie können unterschiedliche Actions für das Drücken und das Loslassen einer Taste definieren. Sie können auch zwischen sogenannten einmaligen Ereignissen (Signalen) und kontinuierlichen Ereignissen (wie z.B. der Mausbewegung) unterscheiden.

Verschiedene Typen von Actions

Kommen wir zum Kern der Sache, den Action-Typen und ihrer Definition:

- No action
Tut nichts. Besonders gut geeignet, um eine Action temporär zu deaktivieren.
- Qualifiers
Die Qualifiers enthalten den Zustand der Shift-, CapsLock-, Ctrl-, Amiga- und ähnlicher Tasten. Mit dieser Action können Sie diese Qualifier auf einen neuen Wert setzen (abhängig vom Modus).

Folgende Modi sind möglich:

- Set: Setzen des Qualifiers unabhängig vom empfangenen Wert.
- Clear: Löschen des Qualifiers unabhängig vom empfangenen Wert.
- Toggle: Löschen des Qualifiers, wenn er gesetzt war und umgekehrt.
- Assign: Wenn der Wert 0 ist, Qualifier löschen, ansonsten Setzen des Qualifiers.

Normalerweise wird das nur für Tastaturen genutzt. Glücklicherweise sind bereits Standard-Actions für eine korrekte Tastaturbelegung definiert.

Beispiel: Zwei unterschiedliche Caps Lock Definitionen:

- Standard-Definition (Umschalten von Caps Lock durch Drücken der Caps Lock-Taste)
 - Beim 'Caps Lock' Tastatur-Item:
Action: Keymapping, Trigger: Any
Action: Qualifiers, Trigger: Down, "Toggle Caps Lock"
 - Bei den 'Shift' Tastatur-Items (beide):
Action: Keymapping, Trigger: Any
Action: Qualifiers, Trigger: Down, "Assign Left/Right Shift"
- Alternative Definition (Einmaliges Drücken von Caps Lock aktiviert es, bis eine der Shift-Tasten gedrückt wird)
 - Beim 'Caps Lock' Tastatur-Item:
Action: Keymapping, Trigger: Any
Action: Qualifiers, Trigger: Down, "Set Caps Lock"
 - Bei den 'Shift' Tastatur-Items (beide):
Action: Keymapping, Trigger: Any
Action: Qualifiers, Trigger: Down, "Clear Caps Lock"
Action: Qualifiers, Trigger: Down, "Assign Left/Right Shift"

Weitere Hinweise:

- Sie können die Shift-Taste auch auf die mittlere Maustaste legen.
 - Durch Hinzufügen dreier Actions zu den Tasten Ctrl, linke und rechte
-

Amigataste kann man den Tastatur-Reset nachbauen.

- Keymapping

Richtet die USB-Tastatur nach der Amiga-Tastatur aus.

Die Tastenbelegung kann noch nicht geändert werden (siehe Keyboardpanel). Im Moment kann nur über die Raw key Actions eine andere Taste zugewiesen werden.

Beachten Sie bitte: Tastaturbelegungen funktionieren wirklich nur mit Tasten. Versuchen Sie nicht, dies mit anderen Items zu probieren.

- Raw key

Sendet ein einzelnes Ereignis vom Tastendrücken oder -loslassen. Einfach auswählen, welche Taste Sie senden wollen, und ob ein Tastenanschlag oder -loslassen generiert werden soll.

Sie sollten sich bei diesen Sachen im Klaren darüber sein, was Sie tun. Vergessen Sie nicht, die entsprechenden Key up Events zu senden, sonst kann dies zu einer großen Anzahl an wiederholten Tasten führen (bis Sie die Taste wieder drücken und wieder loslassen, um ein Up-Ereignis zu generieren).

Falls das ganze zu kompliziert sein sollte, können Sie immer noch die Keystring bzw. die Vanilly Key Action nutzen.

- Vanilla key (detached)

Schmeckt und riecht gut. Aber: Dies erzeugt eine spezifische Taste nach der Beschreibung, die im String-Gadget angegeben ist. Die Beschreibung muss den Regeln für Commodity-Tasten entsprechen. Hier ist eine kurze Liste dessen, was möglich ist (aus den RKM Libraries):

The following regular expression outlines the format of the input event description string:

```
[class] {(qualifier|synonym)} [highmap|ANSICode]
```

Class can be any one of the class strings in the table below. If the class is not explicitl stated, it will assume it is rawkey.

Class String	Input Event Class
-----	-----
"rawkey"	IECLASS_RAWKEY
"newprefs"	IECLASS_NEWPREFS
"diskremoved"	IECLASS_DISKREMOVED
"diskinserted"	IECLASS_DISKINSERTED

Qualifier is one of the qualifier strings from the table below. Notice that there can be more than one qualifier (or none at all) in the input description string.

Qualifier String	Input Event Class
-----	-----
"lshift"	IEQUALIFIER_LSHIFT
"rshift"	IEQUALIFIER_RSHIFT
"capslock"	IEQUALIFIER_CAPSLOCK
"control"	IEQUALIFIER_CONTROL

```

"lalt"           IEQUALIFIER_LALT
"ralt"           IEQUALIFIER_RALT
"lcommand"       IEQUALIFIER_LCOMMAND
"rcommand"       IEQUALIFIER_RCOMMAND
"numericpad"     IEQUALIFIER_NUMERICPAD
"repeat"         IEQUALIFIER_REPEAT
"midbutton"      IEQUALIFIER_MIDBUTTON
"rbutton"        IEQUALIFIER_RBUTTON
"leftbutton"     IEQUALIFIER_LEFTBUTTON
"relativemouse" IEQUALIFIER_RELATIVEMOUSE

```

Synonym is one of the synonym strings from the table below. These strings act as synonyms for groups of qualifiers. Notice that there can be more than one synonym (or none at all) in the input description string.

Synonym String	Synonym Identifier	
"shift"	IXSYM_SHIFT	/* look for either shift key */
"caps"	IXSYM_CAPS	/* look for either shift key or capslock */
"alt"	IXSYM_ALT	/* look for either alt key */

Highmap is one of the following strings:

```

"space", "backspace", "tab", "enter", "return", "esc", "del",
"up", "down", "right", "left", "f1", "f2", "f3", "f4", "f5",
"f6", "f7", "f8", "f9", "f10", "help".

```

ANSIcode is a single character (for example 'a') that Commodities Exchange looks up in the system default keymap.

Here are some example description strings. For function key F2 with the left Shift and either Alt key pressed, the input description string would be:

```
"rawkey lshift alt f2" = "lshift alt f2"
```

Andere, möglicherweise nützliche Strings:

```

"ramiga x" : cut-Operation in den meisten Anwendungsprogrammen.
"ramiga c" : copy-Operation.
"ramiga v" : paste-Operation.

```

- Key string (asynchron)

Der ANSI-String im entsprechenden Gadget wird durch Drücken der entsprechenden Tasten emuliert. Das Programm, dessen Fenster gerade aktiviert ist, erhält die angegebenen Tastenanschläge.

Sie können auch folgende vier Spezial-Zeichen nutzen:

- "\n": CR (Return-Taste)
- "\r": CR (Return-Taste)
- "\t": Tabulator
- "\\": Normaler Backslash

Darüberhinaus können auch andere spezielle Tasten erzeugt werden, wenn diese in spitzen Klammern angegeben werden (z.B. "<alt f4>", siehe auch

die Beschreibung der Vanilla key Action).

Effektiv ist das ganze eine Art Tastaturmakro.

- Mouse position
Ändert die Position Mauszeigers. Benutzt entweder relative oder absolute Koordinaten für jede Achse. Für letztere werden Grafiktablett-Ereignisse erzeugt (die leider nicht von allen Programmen (z.B. Cinema4D) korrekt ausgewertet werden,

Beachten Sie: der Trigger hierfür sollte normalerweise 'Always' sein.

- Buttons
Ändert den Status der linken, rechten oder mittleren Maustaste (eine vierte Maustaste ist durch NewMouse ermöglicht, eine fünfte wird nur für Tablett-Ereignisse erzeugt).

Wie bei den Qualifier Actions gibt es unterschiedliche Modi:

- Press: Drücken der Taste, unabhängig vom Wert.
- Release: Loslassen der Taste, unabhängig vom Wert.
- Flip: Drücken der Taste, wenn sie vorher nicht gedrückt war, bzw. Loslassen der Taste, wenn die vorher gedrückt war.
- Assign: Setzt den Zustand der Taste nach dem empfangenen Wert.

- Tablet data
Dies wird benutzt, um zusätzliche Daten vom Grafiktablett ins System zu schleusen -- wenn das Tablett es erlaubt. Der Wert des Items wird immer einem der folgenden Einträge zugewiesen: Druck, Rotation um die drei Achsen, Annäherung (wenn der Stift gerade in Reichweite ist) und Z-Position (wie hoch ist der Stift über dem Tablett).

Leider senden einige besonders dumme Tablett diese Informationen erst, nachdem sie in einen speziellen Modus geschaltet wurden. Wacom-Tabletts sind noch schlimmer: Sie schicken diese Informationen in einem herstellerspezifischen Report, der nicht mit der HID-Spezifikation konform ist. Schade - für Wacom.

- Digital joystick
- Analogue joystick
Noch nicht implementiert. Dummerweise gibt es keinen vernünftigen Weg, um Daten vom Joystick an eine Anwendung zu schicken, ohne das gameport.device bzw. die lowlevel.library zu patchen (und selbst dann würde es nur mit wenigen Spielen funktionieren). Ein neues Interface für solche Eingabedaten tut not.

- Scrollwheel
Sie haben bis hierher durchgehalten?
Sie haben mich wegen der Scrollräder genervt?
Sehr gut :)
Dieser Punkt erlaubt die Definition eines Scrollrades, entweder vertikal oder horizontal nach dem NewWheel-Standard. Unter AmigaOS brauchen Sie noch MUIWheelPatch, damit MUI-Programme diese Wheel Events auch nutzen, unter MorphOS ist dies bereits implementiert.

Wenn die Wheel-Information nicht als Delta-Daten mit negativem Min.- und positivem Max.-Werten vorliegt, können Sie auch die vier Richtungsoptionen nutzen. Werden diese benutzt, so gibt die Distanz des Schiebers

die Anzahl erzeugter Events wieder. Dieser Modus kann auch dazu genutzt werden, eine BildHoch- bzw. BildRunter-Funktion zu emulieren.

Wichtig: vergessen Sie nicht, der Wheel-Funktion hübsche Sounds zu verpassen. Sie wollen doch Ihre Wintel-Freunde und Nachbarn beeindrucken :)

- Sound (asynchron)

Spielt ein Soundfile ab, wenn diese Action getriggert wird. Alle Formate der in Ihrem System installierten Sound-Datatypes werden unterstützt. Die Lautstärke kann zwischen 0 und 64 eingestellt werden.

Besonders geeignet ist diese Action, wenn Tasks gestartet werden oder wenn der Rechner runtergefahren wird. Oder einfach nur für den nächsten DJ-Abend :)

Ach ja, Sie können natürlich unterschiedliche Sounds für down und up Events einstellen.

- Shell (asynchron)

Eine der mächtigsten Actions -- sie erlaubt das Ausführen eines beliebigen Shell-Kommandos (oder Skripts). Die Klasse kann die entsprechenden Kommandos entweder asynchron starten oder auf das Beenden des aktuell ausgeführten Kommandos warten, ehe der nächste gestartet wird.

Für das Festlegen der Ein- und Ausgabe wird die unter den globalen Einstellungen spezifizierte Konsole zusammen mit den dort zu findenden Stackeinstellungen genutzt

Starke Sache :) Mit dem Rx Kommando können auch kleine Arexx-Befehle geschickt werden.

Beispiele:

- Die 'E-Mail key':
YAM:YAM
- Die 'Scan Previous Track' Taste:
rx "ADDRESS Amplifier.1; playprev"

- Arexx

Sorry, noch nicht implementiert.

- HID output

Sorry, noch nicht implementiert. Hier kann man später die Output Events umdefinieren.

- HID feature

Sorry, noch nicht implementiert. Damit kann man später die Feature Items nutzen.

- Miscellaneous

Enthält einige wenige, aber sehr nützliche Events:

- Activate window:

Aktiviert das Fenster unter der aktuellen Mauszeiger-Position. Wenn Sie diese Action in eine der Mouse Position Items mit Trigger=Always einbauen, bekommen Sie SunMouse/Autopoint-Funktionalität, ganz umsonst.

Genauso einfach: setzen Sie diese Action in das Item für die rechte Maustaste ganz oben in die Liste der Actions. Ergebnis: der Mauszeiger verhält sich ganz ähnlich wie beim RightSunMouse Tool (d.h. Sie erhalten immer das Menu des Fensters, über dem der Mauszeiger steht).

- Window to front:
Holt das aktuelle Fenster in den Vordergrund. Eingesetzt in das Item für die linke Maustaste, ersetzt dies das ClickToFront Commodity.
- Window to back:
Schickt das aktuelle Fenster in den Hintergrund.
- Close window:
Schließt das Fenster durch ein Close Window Ereignis. Recht nützlich.
- Zip window:
Hat denselben Effekt wie das Drücken des Zip/Zoom Gadgets im oberen rechten Eck des Fensters.
- Screencycle:
Wie der Name sagt: Bringt den nächsten Screen nach vorne.
- WB to front:
Setzt die Workbench als vordersten Screen.
- Display beep:
Ruft die DisplayBeep() Funktion von Intuition auf. Doh!
- Reboot machine:
Langsam, mein Freund. Diese Action bootet den Amiga neu.
Es gibt keinen Weg zurück.

Konfigurieren der ganzen Sache

Ich hoffe, dass die Beschreibung bisher halbwegs verständlich war. Und wie macht oder verändert man denn nun Actions? Ganz einfach: wählen Sie eines der Items in der oberen rechten Liste der Collections, die Sie vorher in der oberen linken Liste ausgewählt haben. Dies führt alle bisher definierten Actions für dieses Item in der linken unteren Liste auf. Wenn die Liste leer bleibt, so war bisher keine Action definiert und das Item wird nichts tun, wenn es getriggert wird.

Eine neue Action erzeugen Sie durch Drücken des 'New'-Gadgets. Diese besitzt als Standardeinstellung den Typ 'No action'. Wählen Sie jetzt im Cycle-Gadget die Action aus, die Sie für das gewählte Item haben wollen. Setzen Sie anschliessend die gewünschten Optionen für die Action.

Eine Action wird nach Auswählen der Action in der Liste durch Drücken des 'Del'-Buttons gelöscht.

Actions werden normalerweise immer in der Reihenfolge ausgeführt, in der sie in der Liste stehen. Deshalb kann man die Reihenfolge der Actions mit den 'Up' und 'Down' Gadgets den gewünschten Begebenheiten anpassen. Einige der Actions werden jedoch in einem eigenen Task, unabhängig von der Reihenfolge in der Liste ausgeführt. Diese Actions sind oben in der

Beschreibung mit dem Schlüsselwort 'asynchron' markiert (meist sind dies Actions, deren Ausführung etwas länger dauert). Damit wird verhindert, dass das Eingabegerät (Tastatur, Maus, etc.) während der Ausführung der Action blockiert ist.

Nochmals ein Hinweis zu Array-Items: Es gibt eine Standardactionliste für diese Items. Diese wird aber wirklich nur dann ausgeführt, wenn zu einem Unterereignis keine Aktionen definiert wurden.

Achtung: alle Actions werden direkt nach Einstellung aktiv, nicht erst nach dem Drücken des 'Use'-Knopfes.

Sie können eine Action komplett durch Setzen des Typs 'No action' temporär ausschalten, ohne die anderen Einstellungen der Action zu verlieren.

Zusätzlich gibt es einen Modus, der sehr hilfreich ist, um zu einem Ereignis das entsprechende Item zu finden. Durch Anklicken des 'Track incoming events' Knopfes wird automatisch das letzte aufgetretene Ereignis des entsprechenden Interfaces ausgewählt. Nochmals: es wird nur das letzte Event gezeigt, auch wenn ein Report eine ganze Menge anderer getriggelter Events enthält -- aber es funktioniert sehr gut mit Tastaturen und ähnlichen Geräten.

Last, but not least, es gibt noch zwei Knöpfe, die im Falle eines Falles sehr schnell aus einem verkonfigurierten System helfen können: 'Fill defaults' löscht alle vom Benutzer eingestellten Actions und stellt den Standardzustand wieder her; 'Clear actions' löscht alle Actions einer Collection und deaktiviert diese damit effektiv.

So, das wars. Hoffentlich habe ich nichts vergessen zu erklären. Vielen Dank, dass die trotz der langen Anleitung und der schrecklichen Übersetzungskünste von Michael Böhmer (die ich (CH) dann nochmal überarbeiten durfte ;-)) bis hierher durchgehalten haben. Aber es gibt so vieles, mit dem man sich rumschlagen muss, und noch mehr, das man falsch machen kann, wenn man die hid.class konfigurieren möchte. Trotzdem bin ich definitiv nicht in der Stimmung, alles jedem nochmal zu erklären. Die hid.class ist selbst in der ersten Release bereits ein sehr mächtiges Tool, deshalb: Lesen Sie erst die Anleitung, nutzen Sie sie erst danach.

1.28 Der Bootmouse Klassen-Treiber

bootmouse.class

Klasse: bootmouse.class 1.13

Bindet an: Interface, Klassencode 3 (HID), Unterklassencode 1, Protokoll 2
Konfigurations-GUI: Für jedes Interface einzeln, Default Klassen-GUI

Ein Klassen-Treiber für Human Interface Devices (HID), die auch das sogenannte Boot-Protokoll unterstützen. Die Spanne möglicher Geräte reicht von Mäusen über Grafiktablets bis hin zu sog. Zeige-Geräten.

Die `bootmouse.class` unterstützt nur das Boot-Protokoll, ein einfaches Protokoll speziell zur Einbindung in das PC-BIOS. Da HID wesentlich komplexer und umfangreicher zu implementieren ist, kann damit im PC BIOS auch Unterstützung für USB-Tastaturen und -Mäuse eingebaut werden.

In diesem Modus werden nur drei Mausknöpfe und keinerlei speziellen Features (wie Wheels usw.) unterstützt. Mit V1.11 wird experimentell ein Scrollrad unterstützt. Das funktioniert nicht mit allen Mäusen, da einige die Wheel-Informationen nicht im Bootprotokoll versenden.

Leider besitzen einige USB-Tastaturen mehrere Interfaces, von denen einige vorgeben, das Bootmouse-Protokoll zu unterstützen (was falsch ist). Datenpakete dieser Interfaces werden als Mausbewegungen interpretiert, was zu seltsamen Effekten führen kann. Normalerweise werden diese Seiteneffekte durch Spezialtasten aktiviert. Bitte einfach diese vermeiden.

Geräte dieser Klasse können jederzeit an- und abgesteckt werden.

Um einen 100% kompatiblen Mausersatz zu erhalten, müssen Sie ebenfalls das neue

```
input.device
installieren.
```

Die `bootmouse.class` kann als
RomTag
eingebunden werden.

Wegen des im Kickstart verwendeten Hardware-Bangings können die Tasten von USB-Mäusen leider nicht zum Aufruf des Bootmenüs benutzt werden, ohne das Kickstart zu

```
patchen
.
```

1.29 Der Bootkeyboard Klassen-Treiber

`bootkeyboard.class`

Klasse: `bootkeyboard.class` 1.13

Bindet an: Interface, Klassencode 3 (HID), Unterklassencode 1, Protokoll 1
Konfigurations-GUI: Global für die Klasse

Ein Klassen-Treiber für Human Interface Devices (HID), die auch das sogenannte Boot-Protokoll unterstützen. Dies sind üblicherweise Tastaturen, können aber auch z.B. Fernbedienungen eines Beamers sein.

Die `bootmouse.class` unterstützt nur das Boot-Protokoll, ein einfaches Protokoll speziell zur Einbindung in das PC-BIOS. Da HID wesentlich komplexer und umfangreicher zu implementieren ist, kann damit im PC BIOS auch Unterstützung für USB-Tastaturen und -Mäuse eingebaut werden.

In diesem Modus werden nur die normalen 104 Tasten unterstützt. Die LEDs am

Keyboard werden nicht angesteuert. Es findet aber ein Mapping der Tasten auf die Amiga-Tasten statt. Für die Spezialtasten kann eine der im Aminet erhältlichen Win95 Belegungen genutzt werden.

Die Unterschiede zur normalen Amiga-Tastatur:

```
F11      -> kein Effekt
F12      -> die HELP-Taste (kein Effekt bei ISA Mapping)
Print    -> kein Effekt
Scroll   -> kein Effekt
Break    -> ctrl-c
Insert   -> amiga-c
Pos 1    -> shift-cursor left
End      -> shift-cursor right
Page Up  -> shift-cursor up
Page Down -> shift-cursor down
Num Lock -> kein Effekt
Menu     -> kein Effekt (HELP Taste bei ISA Mapping)
Caps Lock -> Caps Lock bis entweder die linke oder rechte Shift-Taste
          gedrückt wird ODER alternativer Modus:
          Caps Lock bis die Taste erneut gedrückt wird.
```

Das Tastenlayout ist statisch und kann nicht verändert werden. Bitte warten Sie auf die generische HID-Klasse, die Support für alles bieten wird, was Sie sich wünschen.

Achtung: Das Drücken von Ctrl, linker und rechter Amiga Taste gleichzeitig rebootet Ihren Amiga. Falls Reset-Handler existieren, wird versucht, diese vor dem Reset aufzurufen. In diesem Fall wird der Reset um 10 Sekunden verzögert.

Seit Version 1.9 gibt es eine globale Konfigurationsoberfläche. Hier können Sie einstellen, ob die Reset-Handler überhaupt aktiviert werden sollen und wie lange in diesem Fall vor einem Reset gewartet werden soll. Das Caps Lock Verhalten kann dort auch geändert werden. Außerdem gibt es eine Option, die eine alternative Tastenbelegung erlaubt, die den eines Standard PC-Adapters entspricht. Die erweiterte Tastenemulation (wie z.B. Bild rauf/runter) kann mit 'Disable extra key emulation' abgeschaltet werden.

Geräte dieser Klasse können jederzeit an- und abgesteckt werden.

Um einen 100% kompatiblen Mausersatz zu erhalten, müssen Sie ebenfalls das neue

```
input.device
installieren.
```

Die bootkeyboard.class kann als

```
RomTag
eingebunden werden.
```

1.30 Der Printer Klassen-Treiber

```
printer.class
```

Klasse: printer.class 1.17
Bindet an: Interface, Klassencode 7 (Drucker), Unterklassencode 1,
Protokolle 0 und 1
Konfigurations-GUI: Keine

Dies ist ein Standard-Klassen-Treiber für Drucker. Uni- und bidirektionale Protokolle werden unterstützt, das 1284.4-Protokoll ist bislang ungetestet.

Beim Anstecken eines USB-Druckers wird automatisch ein "usbparallel.device" erzeugt. Jeder Drucker erhält eine eigene Unit-Nummer (beginnend mit 0). Beim Ab- und Wiederanstecken eines Druckers wird dem Drucker wieder seine alte Unit-Nummer zugewiesen. Die jeweils vergebene Unit-Nummer wird in den Meldungen des Stacks angezeigt.

Beachten Sie bitte, dass keine Druckertreiber von uns gestellt werden. Vergewissern Sie sich deshalb vor dem Kauf Ihres Wunsch-Druckers, dass entweder im Aminet, in OS3.5/3.9 oder Drittprodukten wie TurboPrint und Studio ein Treiber vorhanden ist. TurboPrint hat bereits Unterstützung für mehrere USB-Drucker am Amiga angekündigt.

In TurboPrint muss nur das "usbparallel.device" als zu benutzendes Device, die jeweilige Unit-Nummer und der korrekte Druckertreiber eingestellt werden. Das wars.

Vorsicht vor sogenannten GDI-Druckern. Sie sind meist sehr billig, aber ohne einen entsprechenden Treiber (den es meist nur für Windows gibt) komplett nutzlos. Viele moderne Drucker unterstützen ebenfalls keinen Textdruck mehr.

Der Epson Stylus 870 Photo benötigt eine spezielle Init-Sequenz, ehe über die USB-Schnittstelle Drucken möglich ist. Die printer.class überprüft die IEEE1284.4 ID auf Epson-Drucker und sendet automatisch den Init-String. Sollten Sie eine Zeile mit seltsamen Zeichen auf ihrem Epson-Drucker erhalten, so schalten Sie diese Init-Sequenz einfach in der GUI ab.

Apropos GUI: Einige Drucker haben Probleme mit dem SOFT_RESET Kommand. Falls Sie also eine derartige Fehlermeldung sehen (oder der Drucker sich weigert zu funktionieren), schalten Sie einfach den SOFT_RESET in der GUI ab.

Wird ein Drucker während des Ausdrucks abgesteckt, so werden alle weiteren Requests dieses Druckauftrags durch den Hyperspace geschickt und an eine fremde Lebensform verfüttert. Wenn gerade kein Druckjob läuft und der Drucker abgesteckt wird, so wird das usbparallel.device nicht entfernt. Es speichert alle dann einlaufenden Requests und wartet darauf, dass der entsprechende Drucker wieder angeschlossen wird.

Die printer.class kann als
RomTag
installiert werden. Dies ist jedoch
nicht unbedingt sinnvoll, denn es existiert keine uns bekannte Anwendung,
die vor dem Booten etwas ausdrucken möchte ;-)

1.31 Der Mass Storage Klassen-Treiber

 masstorage.class

Klasse: masstorage.class 1.23

Bindet an: Interface, Klassencode 8 (masstorage),
 Unterklassencode 6 (SCSI), Protokoll 80 (Bulk-only transport)
 Protokoll 0 (Control-Bulk (beta)),
 Protokoll 1 (CBI (beta))

Konfigurations-GUI: Für jedes Interface einzeln, Einstellungen für LUNs,
 Default Klassen-GUI

Dieser Klassen-Treiber unterstützt Massenspeicher (MSD) mit bulk-only Transfer und dem transparenten SCSI Kommandosatz. Hierunter fallen CF-Leser, ZIP-Laufwerke, USB-Festplatten usw. Beachten Sie, dass noch nicht alle Arten von Protokollen unterstützt werden und dass etliche (billige) Geräte eigene Protokolle implementieren und deshalb nicht unterstützt werden können. Solche Geräte erfordern unter Windows/Mac normalerweise die Installation eines eigenen Treibers, um zu funktionieren.

Denken Sie daran, dass USB1.1 nicht unbedingt für schnelle Transferraten ausgelegt ist. Die theoretisch erreichbare Grenze liegt bei 12Mbits/s, aber durch die Begrenzung von Bulk-Übertragungen Pakete mit maximal 64 Bytes zu generieren, erzeugen diese Transfers einen großen Overhead. Erwarten Sie keine Transferraten, die höher als ca. 950kB/s liegen.

Beim ersten Anstecken eines MSDs wird automatisch ein "usbcsdi.device" erzeugt. Jede Logical UNIT (LUN) erhält eine eigene Unit-Nummer beginnend mit 0. Beim Ab- und Wiederanstecken eines MSD erhält es wieder die vorherige Unit-Nummer zugeordnet. Die jeweils vergebene Unit-Nummer wird in den Meldungen des Stacks angezeigt.

Sie können HDToolBox mit dem usbcsdi.device benutzen. Ebenso können wie gewohnt Mountlists genutzt werden, die das usbcsdi.device ansprechen. Ein Beispiel für einen DOSDriver (UMSD:) ist im Poseidon-Paket enthalten. Kopieren Sie diese Datei und ändern Sie die Unit-Nummer, um alle verfügbaren LUNs eines MSD ansprechen zu können.

Seit Version 1.9 werden FAT formatierte Units automatisch angemeldet, seit V1.15 sogar RDB Medien. Sie brauchen also dafür keine Mountliste mehr. Seit Version 1.13 gibt es die Möglichkeit, den Defaultnamen für FAT-Partitionen zu ändern, sowie die Anzahl der Puffer, die dafür verwendet werden. Schließlich können auch einige Modifikationen vorgenommen werden, u.a. die Zeit eingestellt werden, nachdem ein Datentransfer abgebrochen wird, wenn zuviele NAKs eingehen (wird der Wert auf 0ms eingestellt, so ist NAK Timeout deaktiviert).

Dort kann man auch einstellen, welche Unit des usbcsdi.device für eine LUN verwendet werden soll, falls diese noch frei ist.

Es gibt noch eine Reihe weiterer Schalter, die helfen können, wenn Ihr Gerät sich nicht vollständig an die Spezifikation hält:

- Single LUN: Nur LUN 0 benutzen, auch falls das Geräte mehrere hat.
- No Initial Reset: Verhindert das verschicken des BULK_RESET Kommandos vor dem ersten Befehl. Falls Sie Bulk Reset Fehlermeldungen bekommen, probieren Sie diesen Schalter.
- Simple SCSI: Filtert einige SCSI Kommandos. Falls die Firmware des Geräts abstürzt, schauen Sie mal, ob es so funktioniert.
- Translate CMD6->CMD10: Ändert einige SCSI Kommandos in ihre 10 Byte Versionen. Einige kaputte Geräte können die 6 Byte Befehle nicht richtig handhaben, obwohl diese Standard sind.
- Fake Inquiry: Filtert das Inquiry-Kommando und emuliert es intern.
- Better Removable Support: Schalten Sie das ein, falls das USB Gerät eine gute Firmware besitzt und z.B. einen Schreibschutz unterstützt.
- No Fallback: Mit der Standardeinstellung versucht die Klasse, einige Fehler, die aufgetreten sind, richtig zu deuten und zukünftig zu vermeiden, indem es einige der Patchflags aktiviert. Wenn die MSD Klasse "zurückfällt" (fallback), wird dies durch Meldungen signalisiert. Dann sollten Sie das Gerät ab- und wieder anstecken. Funktioniert es damit besser, so sollten Sie die Einstellungen speichern.
- Debug: Schaltet ein paar weitere Debug-Fehlermeldungen ein.

Bitte kontaktieren Sie mich im Falle, dass Sie durch Umstellen einer Optionen ein Gerät zum Laufen gebracht haben.

AutoMounter von Thore Böckelmann liegt nicht mehr bei, da alle Funktionen auf Basis seines Codes in die Klasse implementiert wurden.

Version 1.15 hat ein neues Feature hinzugefügt, nämlich die Möglichkeit, von RDB formatieren Medien zu booten. Dazu muss Poseidon aber resetfest eingebunden werden (siehe unten).

Die masstorage.class kann als
 RomTag
 installiert werden.

1.32 Der Communication Device Class Treiber für USB Modems

cdcacm.class

Klasse: cdcacm.class 1.6 (beta)
 Bindet an: Interface, Klassencode 2 (Communication Device Class),
 Unterklassencode 2 (Abstract Control Model), Protokoll 1 (Hayes)
 Konfigurations-GUI: Keine

Dies ist ein allgemeiner Klassentreiber für sog. 'Legacy' USB Modems. Dieser Treiber bindet sich an alle Modems, die dem ACM Standard genügen. Leider gibt es eine ganze Reihe billiger Modems, die nicht diesem Standard entsprechen.

Das AmigaOS erhält Zugriff auf das Modem über das "usbmodem.device", welches zur Laufzeit erzeugt wird und eine serielle Schnittstelle emuliert. Baudraten- und Handshakingeinstellungen werden ignoriert.

Es handelt sich hierbei um eine Betaversion, bei der einige Features, wie z.B. die Unterstützung des Break-Signals fehlen.

Die cdcacm.class kann auch als
RomTag
installiert werden.

1.33 Der Klassentreiber für PL2303

serialpl2303.class

Klasse: serialpl2303.class 1.8 (beta)
Bindet an: Gerät, Hersteller/Produkt: 0x067b/0x2303, 0x067b/0x04bb
0x0557/0x2008
0x04bb/0x0a03
0x6189/0x2068

Konfigurations-GUI: Keine

Dies ist ein Klassentreiber für einige USB->Seriell-Adapter, die u.a. von Prolific, Aten und IOData hergestellt werden. Bitte beachten Sie, dass diese Klasse sich nicht an andere USB Produkte binden lässt.

An diese Adapter werden standard RS232 Geräte angeschlossen. Das AmigaOS erhält Zugriff über das "serialpl2303.device", welches zur Laufzeit erzeugt wird.

Es handelt sich hierbei um eine Betaversion, bei der einige Features, wie z.B. die Unterstützung des Break-Signals fehlen.

Die serialpl2303.class kann auch als
RomTag
installiert werden.

1.34 Der RawWrap Klassentreiber

rawwrap.class

Klasse: rawwrap.class 1.7
Bindet an: entweder unbekannte oder herstellerspezifische Interfaces (je nach Einstellung) oder über gezwungene Bindungen.

Konfigurations-GUI: Sowohl globale als auch individuelle Einstellungen pro Interface.

Die `rawwrap.class` ist ein sog. Wrapper, der es erlaubt, einige USB Geräte direkt über ein standard AmigaOS Exec-Device ansprechbar zu machen. Dies erlaubt z.B. die Nutzung von USB Scannern über BetaScan und funktioniert evtl. auch mit anderer Software, die sich mit der Synchronisation von PDAs beschäftigt (dies wurde allerdings noch nicht getestet).

Damit das funktioniert, schnappt sich die `rawwrap.class` einfach die Bulk-In und -Out Endpoints eines Interface. Daraus generiert sie ein `usbraw.device` und schickt `CMD_READ` und `CMD_WRITE` Kommandos einfach weiter. Aus diesem Grund schlägt auch eine Bindung an ein Interface fehl, dass diese beiden Endpoints nicht besitzt.

Jedoch bindet sich die Klasse in der Grundeinstellung nicht automatisch an irgendwelche Interfaces. Sie müssen daher entweder per Hand eine

gezwungene Bindung
einstellen oder das Binden an unbekannte (Klassencode
0) oder herstellerspezifische (Klassencode 255) Interfaces in der GUI
aktivieren.

Apropos GUI: Es gibt eine ganze Reihe verschiedener Optionen, die versuchen, die Klasse mit möglichst vielen Geräten kompatibel zu machen.

Das Panel für die globalen Einstellungen enthält nur zwei Kästchen, die es erlauben, automatische Bindungen mit unbekanntem oder herstellerspezifischen, oder sogar mit allen Interfaces zu erlauben. Nutzen Sie diese Eigenschaft mit Vorsicht, verwenden Sie stattdessen lieber Forced Bindings.

In den Interface Einstellungen befinden sich Optionen, um die Default Gerätenummer für das `usbraw.device` festzulegen und ob nur ein Programm gleichzeitig fähig sein soll, das `usbraw.device` zu öffnen.

Sie können die NAK Timeout-Werte für sowohl den Bulk-In als auch den Bulk-Out Endpoint festlegen. Das Setzen dieser Werte auf 0 schaltet das NAK Timeout Feature aus, was potentiell zu Hängern führen kann.

Es gibt für den Bulk-In Endpoint drei verschiedene Modi:

- Der 'No Buffering' Modus wandelt einfach die Leseaufforderungen in USB anfragen um. Dies funktioniert möglicherweise nur mit wenigen Programmen gut, benötigt aber keinen Pufferspeicher (lassen Sie die Puffergröße bei 2KB).
- Im 'Readahead' (Vorauslese)-Modus wird der Bulk-In Endpoint ständig abgefragt und jedes verfügbare Byte in den zyklischen Puffer geschrieben. Sofern der Puffer groß genug ist, gehen keine Bytes verloren.
- Der 'Read on request' Modus bewirkt, dass nur dann ein Paket vom Gerät angefordert wird, wenn wirklich eine Leseanforderung anliegt. Allerdings werden zuviel gelesene Bytes in einem Puffer zwischengespeichert (es reicht ein Puffer von 2KB).

Die 'Short reads terminate' Option beendet eine wartende Leseanforderung, wenn auf dem USB ein kurzes Paket entdeckt wird. Dies ist die Standardmethode bei USB, um ein Ende der Übertragung zu signalisieren. Falls diese Option nicht genutzt wird, so wartet das Kommando so lange bis

alle angeforderten Bytes gelesen wurden.

Für BetaScan müssen Sie den 'Read on Request' Modus einschalten zusammen mit der 'Short reads terminate' Option.

Die rawwrap.class kann auch als
RomTag
installiert werden.

1.35 Hardware-Treiber

Hardware-Treiber

Diese Release von Poseidon enthält Hardware-Treiber für die Highway-, Subway- und Algor-Karten, sowie für den Pegasos-Onboard USB Chipsatz. Andere Hardware-Treiber für Thylacine, Amithlon, GRex und andere PCI-Buskarten sind durch Dritte in Vorbereitung.

Hardware-Treiber werden in DEVS:USBHardware abgelegt (in Analogie zu SANA Geräten in DEVS:Networks). Wenn mehrere Karten eines Typs in ihrem Rechner vorhanden sind (z.B. zwei Highways), so erhält die erste Karte die Unit 0, die zweite die Unit 1, usw.

Hardware-Treiber können über
Trident
dem System hinzugefügt bzw.
entfernt werden. Alternativ dazu kann auch das Shell-Kommando

AddUSBHardware
benutzt werden.

1.36 MorphOS Input Device V50

MorphOS Input Device V50

Als das originale input.device geschrieben wurde, konnte niemand ahnen, dass mehrere Eingabegeräte eines Typs (z.B. drei Mäuse) gleichzeitig an einen Amiga angeschlossen werden würden. Diese Designschwächen können nur durch komplettes Neuschreiben des input.devices umgangen werden.

Mit einigen kleinen Änderungen konnte das WriteEvent-Interface auf die Unterstützung mehrerer Eingabe-Events hin umgearbeitet werden. Ralph Schmidt hat freundlicherweise Zugang zu dem von ihm für MorphOS geschriebenen input.device gegeben. Dadurch konnte die notwendige Erweiterung eingebaut werden -- jetzt werden externe Events korrekt gehandhabt.

Durch Einsatz dieses neuen `input.device` kann jetzt jede Tastatur wie die original Amiga-Tastatur verwendet werden, einschließlich des Key-Repeats. Die Maus verhält sich ebenfalls in allen Programmen wie gewohnt. Patches wie in meinem Grafiktablett-Treiber `FormAldiHyd` sind damit überflüssig.

Das neue `input.device` ist jedoch nur eine temporäre Lösung, bis das Eingabesystem von AmigaOS/MorphOS durch ein besseres Design ersetzt wird.

Das neue `input.device` wird durch Hinzufügen der Zeile

```
PsdLoadModule DEVS:input.device QUIET
```

in Ihre Startup-Sequence vor `SetPatch` installiert. Für OS-Versionen ≥ 3.5 installiert dies das `input.device V50` vor dem von `SetPatch` ausgelösten Reboot. Ein User teilte uns mit, dass bei OS3.9 der `PsdLoadModule` NACH `SetPatch` stehen muss, da ansonsten der Rechner beim nächsten Booten stehen bleibt.

Überprüfen Sie bitte, ob `PsdLoadModule` in Ihrem `C:-Verzeichnis` liegt, da ansonsten Ihre Startup-Sequence mit einem "command not found"-Fehler abgebrochen wird.

1.37 Der Boot Menu Patch

Der Boot Menu Patch

...ist leider für AmigaOS Versionen $< OS4$ noch nicht fertig. Das Boot Menu wird erst durch den OS3.9 `BoingBag 2` gefixt. Falls Sie dieses Archiv haben, können Sie `BlizKick` von Harry Sintonen benutzen, um das neue Boot Menu mittels aus der `Rom Update` Datei zu extrahieren und mit

```
LoadModule  
zu
```

installieren.

1.38 LoadModule und PsdLoadModule

LoadModule und PsdLoadModule

Um Poseidon und seine externen Dateien als

```
RomTag
```

zu installieren, wird ein spezielles Programm benötigt. Es gibt mindestens drei Programme mit entsprechenden Fähigkeiten:

- LoadV43Module, Autor unbekannt
Erschien mit einer frühen Beta-Version des neuen SCSI-device und ist nicht länger öffentlich erhältlich.
- LoadModule von Thomas 'THOR' Richter
Erhältlich im Aminet, wahrscheinlich Teil von OS3.5/3.9. Leider wurde die Genehmigung, dieses Programm beizulegen, verweigert.
- LoadModule von Torbjörn Andersson
Erhältlich im Aminet, mit freundlicher Erlaubnis des Autors diesem Archiv beigelegt. Es hat ettlliche schöne Features und funktioniert :)

Poseidon benutzt LoadModule von Torbjörn Andersson. Leider benutzt THOR's LoadModule den gleichen Kommando-Namen. Da dieses Tools möglicherweise Teil von OS3.5/3.9 ist, wurde das Tool in PsdLoadModule umbenannt. Das vermeidet ein versehentliches Überschreiben eines bereits existierenden LoadModule-Programms. PsdLoadModule ist immer noch das Programm von Torbjörn.

Falls Sie THOR's LoadModule benutzen wollen, müssen Sie das PsdRomTag-Skript sowie Ihre Startup-Sequence ändern. Dies ist notwendig, da beide Tools eine leicht unterschiedliche Syntax haben: bei THOR's Programm muss das Argument "NOREBOOT" explizit angegeben werden, um das Rebooten der Maschine nach Installation des RomTag zu verhindern. PsdLoadModule hingegen rebootet nur, falls das Argument "REBOOT" angegeben wird.

1.39 RomTags - Wie mache ich Poseidon resetfest?

RomTags - Wie mache ich Poseidon resetfest?

Das Kickstart-ROM des AmigaOS ist als Satz von einzelnen Modulen (sogenannte RomTags) organisiert. Während des Bootvorgangs wird jeder RomTag (geordnet nach seiner Priorität) aufgerufen und kann dann Libraries, Devices oder andere Programme im Speicher starten oder laden. Scout (im Aminet) zeigt die im System vorhandenen RomTags an.

Glücklicherweise kann dieser Mechanismus durch Installation zusätzlicher Module erweitert werden. Entsprechende Programme werden im LoadModule Abschnitt vorgestellt.

Poseidon und seine Klassen-Treiber benötigen nur wenige Libraries. Speziell die dos.library, die erst nach Abschluss des Bootvorgangs verfügbar ist, wird nicht benötigt. Das heißt aber auch, dass Poseidon nichts von Disk laden kann. Deshalb wird die gesamte Konfiguration von Poseidon in einer einzigen ausführbaren Datei (ENVARC:PsdStackloader) mit RomTag gespeichert. Dieses RomTag bootet dann den Stack und seine Konfiguration.

Bitte beachten Sie, dass möglicherweise Klassen- oder Hardware-Treiber von Dritten nicht resetfest installiert werden können.

Poseidon benötigt folgende Dateien als RomTags, um resetfest zu sein:

- LIBS:poseidon.library (die Haupt-Library)
- DEVS:USBHardware/#? (der jeweilige Hardware-Treiber)
- SYS:Classes/USB (die Klassen-Treiber)
- ENVARC:PsdStackloader (der Bootstrap und die Konfiguration)

Das Skript "PsdRomTag" (in SYS:Utilities) übernimmt die automatische Installation der entsprechenden Dateien. Ein Requester fragt nach, ob die RomTags installiert oder entfernt werden sollen. Bitte beachten Sie, daß dabei die jeweils aktuelle Konfiguration aus ENVARC:PsdStackloader resetfest gemacht wird. Nach einer Änderung der Konfiguration in

Trident

sollten die RomTags entfernt und neu installiert werden, um die ←
aktuelle

Konfiguration resetfest zu haben.

1.40 Bekannte Fehler

Bekannte Fehler

Falls Sie einen Bug entdecken, lassen Sie es mich bitte wissen. Geben Sie mir soviele Informationen wie möglich. Bitte schicken Sie mir die Ausgaben von

PsdDevLister
und
PsdErrorLog
.

Bekannte Fehler:

- Epson USB-Drucker benötigen eine spezielle Init-Sequenz. Diese wird immer gesendet, wenn ein Epson-Drucker (irgendein Epson-Drucker) gefunden wird. Dies kann an USB-Parallel-Wandlern zu Datenmüll führen, wenn ein alter Epson-Drucker angeschlossen ist.
- Das CB und CBI Transportprotokoll der Mass Storage Klasse wurden bisher nicht wirklich getestet.

Bitte schauen Sie auch die

FAQ

durch, dort finden sich weitere Hinweise
zu Fehlern und Bugs.

1.41 Ein Dankeschön geht an...

Ein Dankeschön geht an...

Hier ist eine Liste von Leuten, die als (Beta-) Tester, Unterstützer und Beteiligte an diesem Projekt mitgewirkt haben (die Auflistung erfolgt ohne spezielle Reihenfolge):

- Michael 'E3B' Böhmer -- ohne ihn wäre das USB-Projekt niemals zum Leben erwacht. Was wichtiger ist, er ist ein guter Freund auf den man sich verlassen kann. Danke! Von ihm stammt ebenfalls diese fürchterliche Übersetzung der englischen Anleitung.
- Harry 'Piru' Sintonen -- von ihm stammt der cnet.device Quellcode, auf dem alle Devices und Libraries basieren. Dies machte es möglich, von Anfang Hybrid-Software für MorphOS und 68k zu schreiben.
- Ralph 'laire' Schmidt -- für MorphOS und diverse USB-Testhardware. Außerdem hat er den Quellcode zum MorphOS input-device bereitgestellt, was die Erstellung und Mitlieferung der 68k Version des input.devices erst möglich machte.
- Frank 'cyfm' Mariak -- erstellte den USB-Treiber für GRex/Pegasos. Von ihm kamen außerdem eine Menge Vorschläge und Ideen.
- Robert Tsien -- für den Hardwaretreiber für sein Thylacine Board, mit dem er den Poseidon Standard unterstützt.
- Felix Schwarz -- für seine Unterstützung in VHI Studio und die ständigen Rückmeldungen.
- Martin 'Mason' Merz -- für seine Mason Icons. Gute Arbeit, Mann!
- Torsten Jager -- für seine Erlaubnis, fat95 beizulegen und seine großartige Arbeit an dieser Software. Schauen Sie im Aminet, ob es neue Versionen gibt !
- Torbjörn Andersson -- für seine Erlaubnis, LoadModule beizulegen, ein essentieller Beitrag, um Poseidon resetfest zu machen.
- Thore Böckelmann -- für Automounter.
- Gunther Nikl -- Unmengen von Tips, Anmerkungen zur Anleitung und Beispielscode.
- David 'Zapek' Gerber -- eine helfende Hand bei offensichtlichen Fehlern :)
- Gunther Horbach -- für seine USB-Testhardware, seine Geduld und sein Vertrauen in unsere Arbeit.

Ich hoffe, ich habe niemanden vergessen. Falls doch, bitte nicht aufregen, sondern mich drauf aufmerksam mache ;-)

1.42 Die Zukunft

Die Zukunft

Es wird zu Poseidon Updates geben! Zu finden sind sie auf meiner Homepage <http://www.platon42.de/> und wahrscheinlich auch auf <http://www.e3b.de/>. Die Updates werden ebenfalls bald auf dem Aminet zu finden sein.

Eine Liste der Dinge, die für das nächste Release geplant sind:

- Workarounds für "unusual devices" in den Klassen-Treibern
- mehr Klassen-Treiber
- mehr Drittgeräte Tools

Wenn Sie noch Ideen haben, was fehlt, lassen Sie es mich bitte wissen.

1.43 Fragen und Antworten (FAQ)

Fragen und Antworten (FAQ)

Q: Ich habe einen Flashkartenleser angeschlossen, aber es taucht kein Piktogramm auf der Workbench auf!

A: Möglicherweise unterstützt Ihr Kartenleser nicht die MSD-Klasse. Oder es wurde kein Dateisystem auf dem Medium gefunden. Überprüfen Sie, ob Sie FAT95 installiert haben. Lesen Sie den Abschnitt über die

```
masstorage.class
für weitere Einzelheiten.
```

Q: Es geht nicht. Was kann ich tun?

A: Geben Sie mir alle Informationen, die Sie kriegen können (Systemkonfig, was verursacht den Fehler, installierte Programme, Informationen zum USB-Gerät usw.) Schicken Sie mir die Ausgabe von

```
PsdDevLister
und
```

```
PsdErrorLog
```

"Es geht nicht." als Fehlerbeschreibung ist leider zu mager.

Q: Wenn ich den Stack starte und bereits Geräte angeschlossen habe, werden einige nicht erkannt?

A: Manchmal werden einige USB-Geräte beim Hochfahren nicht erkannt. Stecken Sie das Gerät aus und wieder ein. Bitte melden Sie mir derartige Geräte. Schauen Sie auch in den Fehlermeldungen nach.

Q: Tasten auf meiner USB-Tastatur unterstützen kein Tastenwiederholung.

Q: MUI Drag & Drop geht nicht mit meiner USB-Maus.

Q: Wenn ich in IBrowse Links anklicke, passiert nichts.

A: Benutzen Sie bitte das neue
input.device

Q: Ich kann das Boot Menu mit meiner USB-Maus nicht aufrufen.

A: Vergewissern Sie sich, dass Sie das neue
input.device

,
den USB-Stack als
RomTag
und den
boot menu patch
installiert haben.

Q: Ich habe ein USB-Gerät gekauft -- und jetzt funktioniert es nicht.
Schreiben Sie bitte Treiber dafür!

A: Wenn Ihr Gerät einer Standard-Klasse angehört, schicken Sie mir bitte
alle Informationen dazu. Falls das Gerät ein herstellerspezifisches
Protokoll nutzt, kann ich nichts tun. Darüberhinaus ist die Liste der
"ungewöhnlichen Geräte" leider sehr lang. Ich kann nicht für alle
kaputten Firmwares der Geräte da draußen Workarounds schreiben.

1.44 Entwicklerunterlagen

Entwicklerunterlagen

Für interessierte Programmierer sind Entwicklerunterlagen vorhanden. Wegen
der schnellen Weiterentwicklung der Software sind diese jedoch nicht immer
uptodate und deswegen nicht öffentlich verfügbar. Ich entschuldige mich
hierfür.

Dieses Material wird mit der Verfügbarkeit der neuen Version
veröffentlicht, ältere Versionen können via
eMail
angefordert werden.

1.45 Version History

Version History

poseidon.library 2.2

- o Querying DA_CurrCfg now calls the USB command to be sure that the
current config number is up to date (suggested by Uwe Ryssel). Removed
that again, because he reported it to be of no use.
 - o Fixed a race condition for interface and device bindings. Caused some
-

- havoc on my Pegasos with four root hubs during PsdRestart.
- o Added Epson mass storage interface patch for printers with card slots.
- o Poseidon no longer ignores configurations, if the config was set without error but nevertheless returns a bad GetConfiguration number. Fixes problems with broken Olympus cameras.
- o Implemented DA_CloneCount. Mounting two devices with exactly the same VendorID, ProductID and serial number will give it a unique clone count number (e.g. required when plugging in two joypads from the very same vendor -- this still allows different prefs for the second one).
- o Due to popular demand, the "welcome" line has been cosmetically fixed.
- o Now generates more meaningful names for the device name according to the interface class, if the device cannot say its name itself.
- o Empty device descriptor strings are now considered to be invalid.

Trident 2.0

- o More rewriting for subclasses.
- o Lots of internal changes, I cannot remember.
- o Lots of fixes for the subwindow handling.
- o Automatic updating works again.
- o Reintroduced Poseidon-Logo to USB.mprefs.
- o Generally, more robust.
- o Added Online/Offline menu items.

Installer script 1.8

- o Added spanish installer strings and licence translation courtesy of Dámaso Domínguez.
- o Small changes for better ArakAttack support.

serialpl2303.class 1.8

- o Added detection of Sitecom serial adapter (reported by Jens Bagh).

rawwrap.class 1.7

- o Added default settings for epon scanners, as requested by FelixS.

massstorage.class 1.23

- o Simple SCSI patchflag now disables geometry scan.
- o Added two new GUI option for patchflags.
- o Fixed a misplaced CloseLibrary() reported by jacaDcaps
- o Improved fssm check as suggested by Heinz Wrobel.
- o Changed default settings to CMD6->CMD10, no initial reset and simple SCSI.

hid.class 1.6

- o Restricted assignment of mouse buttons and pointer movement to mouse and tablet items (connecting joypads don't block the mouse anymore).
 - o Raw keymap strings enhanced to be user readable and more friendly.
 - o Added missing keyboard mapping stuff.
 - o Added option to disable all actions.
 - o Added option to show the current values of an item.
 - o Added absolute to relative value translation (i.e. for joypads and tablets).
 - o Added clipping of values.
 - o Added scaling of values.
 - o Added pre-condition codes for actions (you probably do not grasp the power that's behind this tiny line, but just read the manual).
 - o Added input redirection for action values.
 - o Added optional tracking of key events for keyboard mapping.
-

- o Generated default actions for joypads.
- o Added patching of `lowlevel.library/ReadJoyPort()` for digital joystick emulation (works great on MAME).
- o Added a button to copy an existing action (might come in handy).
- o Introduced a new trigger if the value returned was out of bound (needed for hatswitch, stupid stupid design flaw of the HID specs IMHO).
- o Added default actions for digital pads (hatswitches).
- o Added default actions for analogue sticks. Warning: if you have got a joystick with hatswitch (digital pad), you need to disable the hatswitch default actions, because they will interfere with the analogue stick handling (or use another port for it? :)).
- o Added a column for the current/last value in the items list. It does not get updated automatically, except for the selected item, if you enable "Report current values".
- o Optimized action updating process when switching between two items (e.g. in track events mode).
- o Fixed reading of item arrays that were not one or eight bits wide. Obviously, this code was never used (as I did not encounter one device yet that uses arbitrarily sized items for an array).
- o Added output and feature items and actions! Yay! This means you can now toggle the leds on a keyboard, toggle the force feedback (rumble) of a joystick, send out all kinds of data to the device.
- o Added default actions for caps lock led.
- o Added a new report/collection with two items to add actions on connection and disconnecting (i.e. variable initialization or sound playback).
- o Lots and lots of internal changes I cannot quite remember.
- o Source code for the HID class is now about 450 KB, just to get the picture.
- o Last Action Hero subtask is now created on demand. This fixes the problems (reported by Hartmut Schulze) of the task not being generated at boot point (FlashRom), due to the lack of some of the required libraries.

printer.class 1.17

- o Accepts now short packets on receiving without outputting an error message.

AddUSBClasses 1.3

- o Now searches in `MOSSYS:Classes/USB` instead of `SYS:Classes/USB`, if found.

AddUSBHardware 1.5

- o Added an ALL option. When used for adding hardware, all units are added until one fails. When used on REMOVE, it removes all active USB controller hardware, effectively switching off the stack.

AddUSBHardware 1.4

- o On removal, checks only the given device name and not the path as requested by Günther Nikl.

algorusb.device 1.7

- o Fixed a bug with packet size calculation on interrupt endpoints with `MaxPktSize != 2^x`. Sorry!!!
- o Optimized interrupt ETD allocation. Turned on immediate interrupts for all endpoints. This should stop the performance loss some people have reported when connecting hubs.

PsdDevlistener

- o Now also returns IDStrings for Devices and Interfaces.
- o Properly reports Application Bindings.
- o Outputs a google search line ;)
- o Replaced printf by Printf (smaller binary).

Trident 1.5

- o Large rewrite for USB.mprefs for MorphOS. Some features were missing, such as automatic updating.

subwayusb.device 2.7

- o Fixed a stack overflow condition due to the increased memory usage and a local dummy struct in the OpenUnit() code.
- o Fixed a problem with the interrupt code (introduced probably with V2.5 or V2.6).

massstorage.class 1.21

- o Mode page reading routine rewritten.
- o GetGeometry completely rewritten. Fixed several bugs. Returns now nice values for all kinds of missing values.
- o Detects write protection change even if no disk change occurs.

poseidon.library 2.1

- o Added psdCheckPipe() requested by UltraGelb.
- o psdFindInterface() now can find alternative interfaces (also requested by UltraGelb).
- o Now also returns truncated string descriptors instead of returning an error.

Trident 1.4

- o Added warning for forced interface bindings, too.
- o DevClass shown is now acquired from the interface classes as people have difficulties understanding that 'None' at DevClass is nothing bad. Also, replaced red exclamation mark by blue question mark.

hid.class 1.5

- o fixed wrong mapping of numeric + as reported by Bernd Gollesch.
- o no delta movements do not cause mouse movement events anymore, as requested by FelixS. This also fixes the problems with YAM2.4 under MorphOS.
- o 'Track incoming events' does not jump back to the zero value, as some devices have zero as a valid item number in array, which actually is out of range.

hub.class 1.21

- o Slight changes not to hang with broken hubs that NAK forever.
- o Will now correctly detect the removal of device shortly after plugging a device in.

massstorage.class 1.20

- o Tweaked CBI even more.
 - o Fixed and improved write protection check, I messed up in V1.19. I'm
-

very sorry the inconveniences I caused.

- o Enabled NAK Timeout for initial GET_MAX_LUN request, because some devices NAK forever here.
- o CreateSegment() was still bugged in the rare case that the segment wanted chip mem (which shouldn't happen with filesystems anyway).
- o Cylinders from Rigid Disk Geometry Page was not read correctly.
- o Fixed a potential division by zero in geometry calculation.
- o If a protection change is detected, a diskchange is now invoked to the unit.

highwayusb.device 2.12 / subwayusb.device 2.6

- o Undone little changes from V2.11, increased priority of interrupt handler again, although I'm not sure what the problem was.
- o Toggle bits are now handles separately for in and out on the same endpoint. There are some devices out there which use the same endpoint for data transfers in both directions, causing the toggle bits to get out of sync. This is a violation of the usb spec, so consider this a workaround for buggy devices.
- o Fixes to ISO IN error cases.
- o ISO IN now also allows more than one packet per request to be received.
- o more little changes.

algorusb.device 1.6

- o Various fixes.

UproarTool 1.0

- o Initial.

DRadioTool 1.1

- o Initial.

poseidon.library 2.0

- o Bumped version to 2. This mainly is due to the anniversary.
- o If device enumeration failed, the assigned usb device address was not freed.
- o During enumeration, the pipe for the target device address was not set to 0. This caused SET_ADDRESS timeouts on retrying the enumeration. Fixed.

hub.class 1.20

- o Fixed wrong GET_STATUS request (non zero wValue).

massstorage.class 1.19

- o Fixed zillion of bugs in CreateSegment() by Ralph Schmidt ;)
 - o When using SimpleSCSI, protection is not queried anymore.
 - o Fixed CB/CBI protocol. I hope, I didn't break other devices, but at least now the MP3 Player sent in by Oliver Mart works. Did I mention that CB/CBI is really a brainf*cked transfer protocol?
 - o Fixed AutoMount crashing, if evil DosList entries are found (thanks to Olli & Laire).
 - o Deactivated Command Status Block Signature check for Olympus cams, as they are broken.
 - o Protection is also updated on each mode sense request.
 - o Fixed CB/CBI transport even more.
-

- o Fixed support for UFI command set. USB Floppy drives should work now!
- o Now the first block is checked for a valid filesystem entry aswell to allow mounting of floppy disks (and similar media without MBR).

Trident 1.3

- o Added balancing bar for information list view (requested by Henes).
- o Will not attempt to run ENV:PsdStackloader anymore with HappyENV installed (because HappyENV has no executable bits set).
- o Added a warning requester for forced device bindings.

hid.class 1.4

- o Fixed resident tag. Now automatic switching from bootmouse/keyboard to HID class on PsdRestart in FlashRom actually works.
- o Help key on USB keyboards was not assigned to help in the AmigaOS (reported by FelixS).

all classes

- o Fixed a potential crash under MorphOS when using multiple MUI windows.
- o Potential danger when attempting to abort quick commands fixed.

PsdRestart 1.2

- o Fixed resident tag.
- o Will now output a message into the errorlog.

algorusb.device 1.4

- o Initial.

poseidon.library 1.33

- o No change, just bumped revision.

Trident 1.2

- o Fixed a wrong format string causing an enforcer hit under MorphOS (only, as the code triggered some locale specific stuff).
- o Changed a gadget label.

hub.class 1.19

- o Packet overflow on GET_HUB_DESCRIPTOR is now ignored. Fixes a broken LG Monitor hub firmware.

hid.class 1.3

- o Get HID descriptor is retried once on request of Stefan Burstroem.
- o Packet overflow on GET_HID_DESCRIPTOR is now ignored. Fixes a broken LG Monitor hub firmware.
- o Failure of setting report type does not bail out anymore. Fixes a broken LG Monitor hub firmware.
- o Done a workaround to avoid MUI writing into the resident module, that would crash the GUI when MuProtectModule was active. Note: This is actually a MUI bug (or flaw).
- o Track incoming event now works better on variables.
- o Fixed an order problem with the qualifier actions that could cause repeated keys. Note: you will have to 'fill the defaults' for keyboards again, if you saved some interface prefs.

massstorage.class 1.18

- o Done a workaround to avoid MUI writing into the resident module,
-

- that would crash the GUI when MuProtectModule was active. Note: This is actually a MUI bug (or flaw).
 - o Added more debug information.
-

poseidon.library 1.32

- o Fixed a 'minor' enforcer hit introduced in V1.31.
- o Added even more vendor strings.
- o Added support for forced bindings.
- o Special Elb*x driver treatment implemented.
- o Various small fixes and tweaks.

Trident 1.1

- o Added support for forced bindings, both for whole devices or single interfaces. They can be set using a context popup menu in the device or interface list views. Warning, this is an advanced user option, using it the wrong way may render your devices useless, until you remove the forced binding again.
- o Removes .elf-extension on DirScan before trying to add that class.
- o Uses full path SYS:Classes/USB/ instead of only USB/ now.
- o Fixed a race condition that could cause a recoverable 01000008 guru on deletion of a hardware entry.
- o Saving the config to ENV:PsdStackloader showed an error if there was none and vice versa (just forgot an '!' in the source :() (reported by Erik Johansson).

AddUSBClasses 1.2

- o Removes .elf-extension on DirScan before trying to add that class.
- o Uses full path SYS:Classes/USB/ instead of only USB/ now.

hid.class 1.2

- o Initial. About ~9000 lines of code (>300KB source code). Lots of features, some are still missing. Depressingly long manual. Great loss of hair. Surprisingly much fun. Enjoy.

rawwrap.class 1.5

- o Initial. Class for wrapping bulk in/out endpoints of an interface to usbraw.device. Can be used for BetaScan, maybe for some other programs aswell (like Spitfire).

bootmouse.class 1.12

- o Forgot filling in the devid and ifid string pointers, therefore failed to load previously saved configuration and caused an enforcer hit.

massstorage.class 1.17

- o Added default unit GUI setting.
 - o Added mode (and GUI patchflag) to translate READ_6/WRITE_6, MODE_SENSE_6, MODE_SELECT_6 to their 10 bytes equivalent. This might help to fix broken devices.
 - o Added debug flag, that will output a few more lines of information.
 - o Fixed a bug that could potentially assign the same unit numbers to different LUNs (that probably did never appear before though).
 - o Buffers setting wasn't saved. Fixed (reported ex-post by Ch. Sauer).
 - o There was one (!) static variable that killed the class when used residently with LoadModule by Thor due to a wrong checksum. Fixed (reported by Ch. Sauer).
-

- o Now also sets Reserved = 1 for mounting FAT partitions. This allows the use of CrossDOS (suggested by Ch. Sauer).
- o Returns the size of the data transferred in io_Actual, although this is not mandatory according to the AutoDocs, but some programs (e.g. ScsiSpeed seem to rely on this).
- o Oops, slight change of code caused the last LUN settings to overwrite all other LUN settings (reported by Erik Johansson). Fixed.

subway.device 2.4

- o Fixed clockport detection.
- o Fixed some bugs (most of them introduced with this version).

highway.device / subway.device

- o Interrupt requests with iouh_Length > iouh_MaxPktSize would never return, if all packets were of iouh_MaxPktSize.
- o Increased reading speed for bulk transfers by about 40% (at least here on my machine) by changing the order of some code. DriveSpeed reports about 735000 bytes/sec for the ZIP250 on Highway.
- o Further code optimizations.

serialpl2303/rawwrap/printer/massstorage/cdcacm.class

- o AbortIO() was broken for more than one read/write request pending, I forgot to walk the linked list in the loop :-\ (reported by Svend Daugaard Pedersen).

all classes

- o Bailing out of a class binding on early binding failure was broken (this was the very last time you will see this message).
- o Oops. GUI exit code forgot to dispose the MUI objects.

poseidon.library 1.31

- o Added stack config support.

Trident 0.12

- o Added stack config stuff.
- o Misc changes.

hub.class 1.17

- o Fixed a bug where the hub task wouldn't necessarily fail out from the outer loop on hub removal.

bootmouse.class 1.11

- o Added experimental quick & hacky wheelmouse support. Will probably only work with a few mice (the ones that report wheel movement in byte 4 even with boot protocol enabled -- works with my Sigma mouse, doesn't work with my Logitech wireless desktop (no wheelmouse data in boot protocol). Be sure you have MUIWheelPatch installed.
- o Added class (defaults) and binding GUI to enable wheelmouse support.

massstorage.class 1.15

- o Booting from RDB implemented (added a flag for this in the GUI aswell). Thanks to Ralph Schmidt for contributing the CreateSegment() code.
 - o Misc internal changes.
 - o RDB Mounting added. Now RDB formatted partitions will be mounted automatically on media insertion aswell.
-

- o Mounting is not done from the device task anymore (as this could cause a deadlock, if the filesystem wanted to do device IO before returning from the mount call), but from a tiny separate task, living only for the moment :)
- o Changed implementation of FIX_INQUIRY to fake inquiry. Old fix inquiry is still present for wrong inquiry requests. Fake Inquiry should fix a lot of broken devices.
- o Added option to disable the initial bulk reset, as some broken devices seem to choke on it (and crash their firmware, stupid thing).
- o Changed NakTimeout prefs value from ms to 1/10secs. This seems more reasonable. Please check your configs.
- o Added option to delay the device startup. This is needed for some broken devices, which crash their firmware, if they receive commands during their setup (Nikon Coolpix).
- o Added possibility to change the DOSType. If the DOSType is NOT 0x464154xx (FATx), LowCyl will not be 0 and autodetection of FAT95 will be disabled. This should allow the use of CrossDOS, but this has not been tested.
- o Added saving of default prefs.
- o Added default prefs GUI (class config).

printer.class 1.14

- o Added implementation of CMD_START/CMD_STOP.
- o Added class GUI with two options: one disables the sending of the init sequence on Epson printers, the other one disables the SOFT_RESET command which causes problems on some broken printers.

poseidon.library 1.30

- o psdFreeVec() was broken (freed 4 bytes less than allocated). I am really sorry about this. Pfuschwork orange.
- o Added class config reading and writing.
- o Added interface/device config reading and writing.
- o Added some more support functions to the library.
- o Internal variables/flags added for USB2.0 support (although, there is no working USB2.0 support right now).
- o Added NAK timeout (5 secs) on EP0 while enumerating, as there seem to be badly broken devices out there, doing NAKs forever, blocking out the stack until the device is removed again.
- o There were some global fields in the library data section which caused LoadModule by THOR to remove the resident tag because the checksum was wrong. Hopefully, this is now fixed.
- o Licence check was broken. I forgot an if-clause, which caused the single user keyfile being killed after the verification (i.e. click on online/offline/online and you were back to demo version). Sorry to Mediator users.

Trident 0.11

- o Added calls to open class config windows.
 - o Fixed some gadgets in device window not being ghosted.
 - o Added 'Settings' gadget in device window.
 - o Added speed output of USB 2.0 highspeed devices (no support in the stack, yet).
 - o Added calls to open binding config windows.
 - o Added some gadgets in device info window.
 - o Increased stack to 16KB as it was nearly running out of stack.
-

all classes

- o Lots of internal changes to be more extensible (via a DoMethod vector).
- o Changed classes to use the new FindEndpoint and FindInterface support functions. Saves a few bytes, makes classes more robust.
- o Bailing out code introduced with release V1.27 was broken and would always have crashed the machine on failing allocation. Fixed.

massstorage.class 1.13

- o Bugfix: With a device with multiple LUNs, Permit() was called more often than Forbid(). Could have caused all kinds of havoc. Fixed.
- o Data overflow is now being ignored in the data phase. This fixes some broken devices.
- o Added device config GUI. I hope you enjoy it. If there's something missing, please let me know. Remember that automounting of RDB partitions still doesn't work, even if there's a switch already.
- o On response timeout errors and multiple LUNs, access to unit could be locked, hanging the other LUN tasks. Stupid bug, which is now fixed.
- o CB/CBI recovery handling didn't work in some cases.
- o Added CBI_RESET (maybe this helps?).

bootkeyboard.class 1.9

- o Added class config GUI. The configuration settings include turning on/off of ResetHandlers, setting the time until the machine reboots (in case of resethandlers installed) and an alternate caps lock mode.
- o Added ISA keymap option, aswell as an option to disable the extra key translation functions (like page up/down).
- o Extra key translation did not send events for up-qualifiers, therefore shift (or amiga or control) were still pressed after using keys like page up/down. Fixed.

highway.device

subway.device

- o Fixed a possibly wrong PPC gate function.

poseidon.library 1.28

- o BrokenConfigFix: Sony Handycam has a wrong MSD subclass.
- o BrokenConfigFix: LaCie USB Floppy drive has a dummy interface.
- o Internal changes to accept user keyfiles.
- o Added restarting function.

AddUsbHardware 1.2

- o Did not call psdEnumerateHardware() when using QUIET. Fixed. Reported by Christian Sauer, thanks a lot.

PsdRestart 1.1

- o Initial.

pencam.vhi 1.2

- o Initial driver for VHI Studio.

PencamTool 1.6

- o Optimized bayer derasterization function.
 - o Derived pencam.vhi driver from PencamTool.
-

Trident 0.9

- o Added hardware driver information window.
- o Added menu item to inject user keyfile.

all classes

- o Fixed them to compile for MorphOS.

highwayusb.device 2.9

- o Slight changes for compiling for MorphOS.

subwayusb.device 2.3

- o Slight changes for compiling for MorphOS.
-

poseidon.library 1.27

- o Internal changes for PPC native compile.

printer.class 1.11

- o Fixed printes with twisted bulk in/bulk out. Should have been more careful with this from the beginning :-\

all classes

- o Bailing out of a class binding on binding failure was broken.
-

poseidon.library 1.26

- o On USB devices with multiple configurations, only the first one was read due to me thinking that I would get all configurations automatically. This is now fixed. Hey, it was the first time I came across a device with multiple configs, so I couldn't test the code before.

massstorage.class 1.11

- o Added information messages on mounting (after three attempts :)).
- o Mounting now again uses lowcyl = 0. But therefore, detection of an already mounted partition has been reduced to unit, device and dostype. This means that no multiple partitions on a harddisk will be mounted, but only the first one. I hope this is not a problem for most of you. (thanks to Jens & Jean-François).
- o After mounting, a diskchange signal was sent. This is now fixed (thanks to Jens).
- o Added NSD and TD64 support. This means you now can uses harddisks >4GB.

serialpl2303.class 1.3

- o Fixed buffer management. Added EOF mode.

cdcacm.class 1.2

- o Initial. Should support state reporting and EOF mode.
-

pencamtool

- o Added UPTO option.
-

- highwayusb.device 2.8
- o Optimized copyloop a bit.
- subwayusb.device 2.2
- o Initial
- serialpl2303.class 1.2
- o Initial
- bootkeyboard.class 1.6
- o Now calls reset handlers of the keyboard.device on reset. This is done in a tricky, but system-friendly way. If any reset handler is found, the reset is delayed by 10 seconds (no way to find out, if the reset handlers have finished).
- massstorage.class 1.9
- o Added 200ms delay for some strange devices before first interaction. I hope this is enough for most drives.
 - o Added automounting capability for FAT95. Took me ages to get this working. Sorry.
 - o Added experimental untested support for CB and CBI transport protocols.
 - o Added special init sequence for eUSCSI bridges.
 - o Disabled internal retry mechanism for HDCMD_SCSI commands. Maybe this fixes the disk changing bugs when using "+s" on FAT95.
 - o Fixed broken disk change detection. Sorry, just forgot some paranthesis.
- hub.class 1.14
- o Changed the init sequence to be more error tolerant, adding several retries
 - o Added locking mechanism (oops! how could I forget) of address 0 to avoid confusion on the bus with multiple hubs connected to a hub at the same time.
 - o Lots of little changes.
- poseidon.library 1.25
- o If SET_CONFIGURATION fails, the enumeration will now still proceed. This seems to be the case with a few USB2.0 devices.
 - o Changed psdSpawnSubTask() to launch a process if started with dos available.
 - o Misc little changes.

1.46 Kontakt zum Autor

Kontakt zum Autor

Falls es Fragen gibt oder Sie Entwicklerunterlagen möchten -- schreiben Sie mir. Es gibt ebenfalls eine Mailing-Liste für neue Entwickler, kontaktieren Sie mich, falls Sie ebenfalls auf diese Liste wollen.

Chris Hodges
Kennedystr. 8
D-82178 Puchheim

Account: 359 68 63
BLZ : 700 530 70
Bank : Sparkasse Fürstenfeldbruck

Germany

Tel.: +49-89/80001543

Email: hodges@in.tum.de

WWW: <http://www.platon42.de/>

IRC: platon42 on EfNet/Arcnet