

Erledigt

Eigenes Audio Layout erstellen (AppleALC)

Beitrag von „e4535030“ vom 21. April 2020, 20:29

Hallo Leute,

Ich suche eine Möglichkeit meine Soundkarte (Onboard Z370M D3H ALC892) mal richtig komplett ans laufen zu bringen.

Mit AppleALC funktioniert der ALC892 zwar aber die Ausgänge/eingänge sind bei keiner Layout ID so wie ich sie mir vorstelle.

Da ich hier nen aufwendiges Audio Setup (mehrere einzel angesteuerte Subs) habe benutze ich Audio Hijack um mir meine Raummoden Korrektur zu berechnen.

Alle anderen Audio Player (TV / Radio usw) gehen dann durch den Hackintosh und werden zusammengemischt.

Die ALC892 hat laut Datenblatt 5 Stereo DACs + 2 digital SPDIF out und 2 stereo ADCs + 1 SPIDF in die chip intern an die 8 verschiedenen Ausgänge geschalten werden können.

Mein Mainboard hat insgesamt mit front audio 8 Stereo Anschlüsse klinken Buchsen und einen SPIDF header onboard.

SPIDF interessiert mich vorerst nicht.

Die Konstellation sollte also ermöglichen das ich alle DACs und alle ADCs verwenden könnte.

Bei allen vorhandenen LayoutIDs sind aber nur maximal 3 Ausgänge und 2 Eingänge vorhanden.

Ich hab schon gegoogelt aber nicht gefunden wie die Layout.xml Dateien erstellt werden.

Ich würde mir gerne ne eigene Layout.xml erstellen die mir ermöglicht alle 5 Dacs und die 2 ADCs zu benutzen.

Hat da jemand ne Idee wie ich die Datei generiere?

Ich hab kein problem damit das Mainboard auszumessen um rauszufinden an welchem ALC892 Port welche Buchse vom Mainboard angeschlossen ist.

Das sollte aber eigentlich nichtmal nötig sein. Welche Buchse dann am ende wie im OS heißt ist mir egal.

Die Datei würde ich dann natürlich irgendwo zur verfügung stellen damit andere mit de Board die Arbeit nicht mehr machen müssen.

Vielen dank, liebe grüße.

Beitrag von „e4535030“ vom 22. April 2020, 23:10

Hallo Leute

es gibt fortschritt:

Ich habe mit der aktuellen version vom PinConfigurator rumgespielt und rausgefunden das keine von den mit Clover gedumpten Codec Dateien funktioniert.

Verfahren war folgendes:

Im Clover Bootloader F8 drücken um Codec zu Dumpen.

Produzierte Datei in PinConfigurator einlesen.

Verbs sanitize anklicken.

AppleALC.kext patchen.

mit gepatchter AppleALC.kext booten.

Beim rumspielen ist mir aber aufgefallen das Layout ID 15 fast perfekt funktioniert. Alle Ausgänge sind da, nur leider ist Speaker intern im switchmode mit dem Kopfhörerausgang konfiguriert (selbiges bei Mic intern hinten + Frontpanel Buchse Mic)

Also hab ich probiert den Switchmode aus den Daten von der 15er Layout ID zu entfernen, bin aber kläglich gescheitert.

ALC1220 hat z.b. ID28 für manualmode und ID29 für switchmode, scheinbar ist diese Modeeinstellung aber nicht in den verbs gespeichert, denn beide layout IDs sind laut PinConfigurator exakt gleich?!?

Kann mir jemand ne kleine Hilfe geben wie ich das mache?

[MacPeet](#) vielleicht? Er scheint super viel Ahnung auf dem gebiet zu haben.

Ich hab mal meinen Dump angehängt.

Vielen Dank!

Beitrag von „wern“ vom 23. April 2020, 10:48

Soweit ich weiss unterstützt AppleHDA nur 8 Geräte, da bei Layout 15 schon 8 integriert sind müssen (falls benötigt) zusätzliche Geräte ersetzt werden.

Da der Codec nur 2 ADCs hat müssen 2 Eingänge im Switchmode arbeiten. Den Kopfhörer kannst du aber im manuellen Modus betreiben. Bin selber kein Fan von diesem AutoSwitchmode, benutze eine App dafür.

Die Pinconfig kannst du belassen wie sie ist. Ändere nur die Platforms15.xml, Pin 20 und 27 sollten wie 21-23 aussehen.

Beitrag von „e4535030“ vom 23. April 2020, 11:39

Hallo,

Da ich 5 DACs und 2 ADCs haben sind das 7 Geräte total, das passt super mit den 8 Anschlüssen und Mic auf switchmode.

Danke für die info.

In der Plattform Datei ist also drin AHA!

Nur leider gibts die in der AppleALC nicht?!?

Da die Datei im Github liegt heißt das also ich muss das mit Xcode neu kompilieren?

So richtig unterschiedlich sieht das bei 20/27 und 21/23 aber nicht aus?

Die Struktur ist gleich, nur die NodeIDs unterscheiden sich, 12 u. 13 fürs eine 38 u. 37 fürs andere.

Müssten die gleich sein?

Meinem beschränkten Verständnis nach sind doch die Nodes quasi die Mixer im chip damit der passende DAC am gewünschtem Anschluss rauskommt oder?

Das würde aber heißen ich kann die IDs nicht gleich setzen, dann müsste das signal ja durch die gleichen Mixer durchgehen?

Überseh ich da was?

Was Laut Block Diagramm vom codec auch gehen müsste:

z.b. Mic Port 18 und Port 19 in hardware summieren und dann auf ADC 09h zu geben gleichzeitig aber Port 1B auf ADC 08h zu geben.

Das konfiguriert man über die nodes?

Was ich auch nicht verstehe:

In den Platform.xml ist immer "CommonPeripheralDSP" erwähnt.

Beutetet das da sind irgendwelche Hardware DSPs aktiv und machen irgendwelche

Klangveränderungen?

Unter Windows gibts ja z.b. Equi und Komische "Badezimmer" / "Höhle" Filter, ich dachte immer diese leben im Windows Treiber bzw der Software.

Im Blockdiagramm sind auch nur "Digital Filter" erwähnt, das ist aber so wie es aussieht nur Signalrekonstruktion / LP für out und HP für in.

Die Macbooks verwenden mit Sicherheit DSPs um den klang der internen Lautsprecher zu verbiegen, wie ist das geregelt?

Wenn ich mir z.b. den Schaltplan vom A1297 anschaue (5 Lautsprecher)

Da sind direkt Verstärker hinter den Codec Ausgängen, wo wird da das Processing gemacht?

Übernimmt das OSX mit AU plugins oder ist da was im codec?

Das Datenblatt vom Codec hab ich natürlich nicht gefunden.

Beitrag von „wern“ vom 23. April 2020, 11:59

8 Geräte insgesamt, also jeden Pin auch wenn im switch modus konfiguriert.

Die NodeIDs musst du nicht ändern, können aber geändert werden. Anbei die kompilierte ALC und xml mit getrenntem Kopfhöreranschluss.

CommonPeripheralDSP ist für Micro und Apple Earbuds, funktionieren aber nur mit dem MikeyDriver. DSP ist auch möglich, da kann man mit den Gleitkommazahlen experimentieren.

Bei den MacBooks wird alles über DSP gesteuert.

Beitrag von „e4535030“ vom 23. April 2020, 15:58

Hallo,

Danke für die Arbeit!

Ich bin aber grad ein wenig verwirrt.

Ich hab die Platform15.xml und die Platform15.xml von GitHub (AppleALC) durch nen text checker geschickt und der findet keine unterschiede außer 2 Leerzeilen Zeile 223?

AppleALC teste ich jetzt mal.

Öh Also die AppleALC funktioniert perfekt!

Kann das sein das du die falsche Platform Datei hochgeladen hast?

Schön wäre noch wenn du kurz erklären könntest wie du es gemacht hast, damit ich bei neuem AppleALC release nicht um Hilfe bitten muss und auch andere aus diesem thread lernen können.

Perfekt vielen dank!

Beitrag von „MacPeet“ vom 23. April 2020, 17:36

[e4535030](#)

Deine Dump-Datei lässt sich nicht komplett wandeln und zeigt Fehler. Damit kann ich nix anfangen.

Womit und wie ist die gemacht? Unter Linux?

Ist aber im Prinzip auch egal, denn [wern](#) hat Dir wohl schon einen ManualMode-Kext gebastelt.

Ich frage mich allerdings, was gegen den SwitchMode spricht. Hierbei muss man nichts mehr tun, wenn man von hinten grün (die Boxen) auf vorn grün (Kopfhörer) wechselt.

Im ManualMode muss man dafür jedesmal ins Menü.

Dem hinteren grünen LineOut-Anschluss ist es dabei völlig egal, ob er als LineOut oder interne

Speaker konfiguriert ist, er funktioniert so oder so.

Ich denke, Dich stört, dass der externe Anschluss als intern angezeigt wird und Du denkst, dass es falsch ist, oder? Sei ehrlich!

Es ist aber nicht falsch, es macht genau so Sinn.

In beiden Konfigurationen (SwitchMode/ManualMode) lassen sich alle LineOut's hinten mittels MidiSetup mit einem "gemeinsamen Gerät" zu einer 5.1 Konfiguration verbinden, falls man 5.1 Boxen betreiben möchte.

Bei den Inputs ist es oft etwas schwieriger. Es gibt drei Ausgänge (hinten pink Mic, vorn pink Mic und hinten LineIn blau), allerdings oft nur zwei Ausgangs-Knoten.

Somit müssen sich die beiden Mic's einen Knoten teilen.

Einige Rechner haben sogar drei Ausgangsknoten, was ich an Hand Deines defekten Dump's so nicht sehen kann.

Treiber, wie unter Windows ("Badezimmer" / "Höhle / Konzertsaal, etc.") gibt's hier nicht. Du kannst zufrieden sein, dass Dein Audio unter macOS so gut läuft mit einem Hackintosh.

Ferner kannst Du keine fertige kompilierte AppleALC.kext ändern. Hier kannst Du zwar die PinConfigs einsehen, aber ändern kannst Du hier nix.

Du musst schon auf GitHub den aktuellen SourceCode laden (entpackt ist dat dann der Ordner "AppleALC-master").

In diesen Ordner kopierst Du noch die neuste Lilu.kext, ansonsten kannst Du mit xcode nicht neu kompilieren, da AppleALC ein Plugin von Lilu ist.

So, der einzige Unterordner, der Dich für Änderungen interessiert, ist der Ordner "Resources".

Die dortigen Dateien (.plist/.xml) kannst Du nicht mit einem normalen Text-Editor bearbeiten.

Ich empfehle die PlistEditPro.app dafür.

In AppleALC-master/Resources/ liegt PinConfigs.kext dort rechte Maustaste "Paketinhalt zeigen" und zur info.plist navigieren.

In dieser Datei liegen alle PinConfigs aller Codecs der AppleALC. Hier kannst Du einen bearbeiten, ggf. einen neuen Codec durch weiteren Eintrag erweitern.

In Deinem Fall ist der nächste Ordner der "Resources/ALC892".

Dort liegt die jeweilige layout?.xml und die dazugehörige Platforms?.xml, die Du ändern kannst.

Wenn Du einen ganz neuen Codec hinzufügen willst, mit einer freien ID, dann musst Du die auch in der info.plist im Ordner "ALC892" erweitern.



Hier z.B. siehst Du, die ID's 15, 16, 17, 18 sind alle von mir, allerdings alle nicht ursprünglich für einen Z370.

Willst Du nun einen neuen Codec im ManualMode hinzufügen, z.B. ID 14 wäre noch frei, dann musst Du die neuen Dateien auch hier eintragen/hinzufügen in der info.plist unter Files/Layouts und Files/Platforms.

Die entsprechenden z.B. layout14.xml und Platforms14.xml natürlich auch in dem Ordner "ALC892" ablegen. Nur mal so als Beispiel.

Dann mit Xcode neu kompilieren und hoffen, dass alles läuft, wie gewünscht.

Alle Änderungen die Du für Dich am SourceCode machst, werden im nächsten Release nicht auftauchen.

Du musst den neuen Codec also verbreiten, mittels PullRequest oder jedes Mal den dann aktuellen SourceCode patchen.

Wenn Du es einmal drauf hast, dann ist die letztere Option kein großer Akt mehr.

Beitrag von „e4535030“ vom 23. April 2020, 19:55

TLDR: Ich brauche die 5 Output Channel wie die Ports heißen ist vollkommen egal, Aber ich bin immer noch nicht viel schlauer 😞

Zitat

Ich frage mich allerdings, was gegen den SwitchMode spricht. Hierbei muss man nichts mehr tun, wenn man von hinten grün (die Boxen) auf vorn grün (Kopfhörer) wechselt.

Im ManualMode muss man dafür jedesmal ins Menü.

Dem hinteren grünen LineOut-Anschluss ist es dabei völlig egal, ob er als LineOut oder interne Speaker konfiguriert ist, er funktioniert so oder so.

Ich denke, Dich stört, dass der externe Anschluss als intern angezeigt wird und Du denkst, dass es falsch ist, oder? Sei ehrlich!

Es ist aber nicht falsch, es macht genau so Sinn

Nein, Das ergibt schon sinn.

Das "Problem" ist das ich kein herkömmliches 5.1 Setup benutze.

Ich hab hier 4 Stereo Verstärker und betreibe damit im Prinzip nen 4.4 Setup.

4 Subwoofer zur Raummoden Kompensation + Beamforming, dafür muss jeder Sub extra angesteuert werden.

Jeder übernimmt einen anderen Frequenzbereich mit anderen delays. Das ergibt dann

sozusagen Beamforming wie man es von WLAN kennt.

Sinn und Zweck davon ist es entweder die Nachbarn nicht zu stänkern = Bass mit gradem Frequenzgang fast nur am Sitzplatz, restlicher Raum weit weniger Pegel.

Alternativ könnte man das Setup auch so einrichten, dass der Bass im Raum gleichmäßiger verteilt ist.

Ich glaub das kennt ja jeder, in einer Ecke dröhnt es und an anderen Stellen vom Raum passiert nix.

Um die Phasen / Frequenzgangs / Pegel Kompensation zu erreichen muss man allerlei Audiofilter benutzen.

Die Audiofilter laufen bei mir in Audio Hijack.

OSX sieht ein normales Stereo Lautsprecher Pärchen, gibt das Signal über z.B. Soundflower oder Loopback an Audio Hijack.

Selbiges berechnet dann die nötigen Änderungen und gibt das Signal auf die Ausgänge.

Dafür brauch ich mindestens 8 einzelne Channel.

Andere Geräte z.B. TV laufen dann an den Line In und werden so über Audio Hijack auch mit den Filtern berechnet und anschließend ausgegeben.

Man kann das natürlich auch für Surround machen, dann brauch man noch mehr Ausgänge, dafür habe ich in späterer Heimkino Ausbaustufe USB Soundkarten vorgesehen.

Das Problem:

Ab und zu höre ich mit Kopfhörern und benutze dafür den Front Output. Wenn der jetzt im Switchmode läuft, fehlt mir hinten plötzlich ein Ausgang.

Das bedeutet, dass die anderen Geräte, die grade über den Line In rein kommen, unter anderem über den grünen Rear Ausgang ausgegeben werden sollten, auf den Kopfhörern landen. Zusätzlich schaltet der Codec dabei den Amp vom grünen Rear Port ab. Der Anschluss ist dann tristate und produziert Brummen.

Zitat

Treiber, wie unter Windows ("Badezimmer" / "Höhle" / "Konzertsaal, etc.") gibt's hier

nicht. Du kannst zufrieden sein, dass Dein Audio unter macOS so gut läuft mit einem Hackintosh.

Die Sound Optionen möchte ich auch garnicht benutzen. Die Frage ist gewesen ob diese in Software oder im Codec berechnet werden. Wenn diese im Codec berechnet werden würde das ermöglichen manche Filter nicht mehr auf der CPU zu berechnen und damit Ressourcen zu sparen.

Das würde z.b. ermöglichen das ich 2 Line in Eingänge im Codec Summiere, das sollte der Codec laut Blockdiagram können und würde die CPU entlasten.

Naja zu <10.6.x Zeiten lief das auch immer wunderbar, das Setup gibts bei mir schon ne weile. Irgendwann wurde der MacMini aber zu klein und ich bin bei erscheinen auf Hackintosh mit Sandy Bridge Intel umgestiegen. Damit Lief das auch schon hervorragend (Gigabyte Board)

Seit 10.15.x hab ich Probleme mit Audio Crashes die vorher nie da waren. (Siehe anderer Thread)

Zitat

Erklärung zum Kompilieren der AppleALC

Danke, soweit ist das jetzt klar, ich werde dann mal nen Pull Request machen wenn ich selber Kompiliert habe.

Die Frage die mir bleibt:

Wie extrahiere ich die Knoten? Wieso sind beide Platform15.xml Gleich aber haben unterschiedliche "Wirkung".

Wie ich vom Dump zur PlatformXX.xml komme ist quasi das was mir fehlt.

Zitat

Deine Dump-Datei lässt sich nicht komplett wandeln und zeigt Fehler. Damit kann ich nix anfangen.

Womit und wie ist die gemacht? Unter Linux?

Ich habe da einen Fehler gemacht, die Datei war nicht der Dump sondern das File vom Pin Configurator.

Ich hab hier nochmal den richtigen Dump Angehängt.

Erstellt wurde der mit Clover (F8 Taste im Bootloader)

Beitrag von „wern“ vom 23. April 2020, 23:04

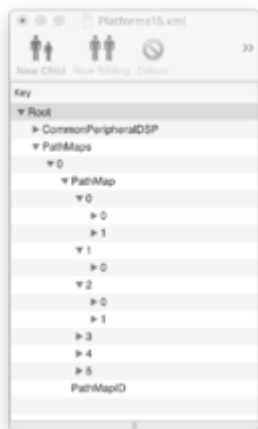
Hier mal optisch der Unterschied.

alt: nach PathMap

0 = 2x Line-In, switch Mode

1 = 1x Line-In

2 = 2x Line-Out + HP switch Mode



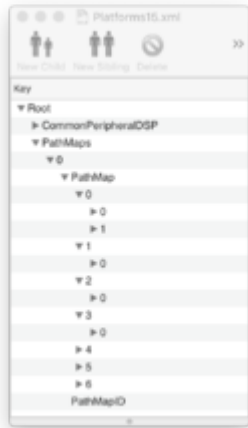
neu:

0 = 2x Line-In, switch Mode

1 = 1x Line-In

2 = 1x Line-Out

3 = 1x HP



Vielleicht hilft dies weiter. Benutze ein Plist Editor.

Zum Auto Switch Mode.

Ich benutze meistens Kopfhörer, da hat mich der Stecker vorne am Gehäuse immer gestört. Mein Mainboard hat hinten auch einen Kopfhöreranschluss da steckt auch immer der Kopfhörer drin. Immer aufstehen um die Lautsprecher zu benutzen ist nicht gerade komfortabel, deswegen kein AutoSwitchMode.

Beitrag von „e4535030“ vom 23. April 2020, 23:24

Vielen dank [uern](#)

Die Datei in .plist geändert und mit xCode geöffnet zeigt es eindeutig.

Keine Ahnung wieso mein Text Compare keine unterschiede gefunden hat. Da sind sie auf jeden fall!

Bleibt nur noch, Wie kommt man vom Dump zur platformXX.xml?

Beitrag von „uern“ vom 24. April 2020, 10:33

Ich benutze dafür keine Tools vielleicht mal den Apple Taschenrechner um die hex. Werte nach decimal umzurechnen.

Kleines Beispiel:

Erster Pin (Suchfunktion)

Node 0x14 [Pin Complex] wcaps 0x0040058D: Stereo Amp-Out

Amp-Out caps: ofs=0x00, nsteps=0x00, stepsize=00, mute=1

Amp-Out vals: [0x80 0x80]

Pincap 0x0001003E: IN OUT HP EAPD Trigger Detect

EAPD 0x2: EAPD

Pin Default 0x01014010: [Jack] Line Out at Ext Rear

Conn = 1/8, Color = Green

DefAssociation = 0x1, Sequence = 0x0

Pin-ctls: 0x20: IN

Connection: 1

0x0C

Node 0x0C [Audio Mixer] wcaps 0x0020010B: Stereo Amp-In

Amp-In caps: ofs=0x00, nsteps=0x00, stepsize=00, mute=1

Amp-In vals: [0x00 0x00] [0x80 0x80]

Connection: 2

0x02 0x0B

Node 0x02 [Audio Output] wcaps 0x0000041D: Stereo Amp-Out

Amp-Out caps: ofs=0x40, nsteps=0x40, stepsize=03, mute=0

Amp-Out vals: [0x40 0x40]

Pin: 0x14 (20)

Mixer: 0xc (12)

DAC 0x2 (2)

Natürlich gibt es noch Feintuning, zB. AMPs usw.

Die Codec Werte stimmen oft nicht. Der Treiber korrigiert das aber in der Regel. Hier zB. Pin Control "0x20 IN" ist falsch. Sollte "0x40 OUT" sein.

Beitrag von „MacPeet“ vom 24. April 2020, 20:07

Die Unterschiede zwischen den beiden Platform-Dateien hat [wern](#) Dir ja schon aufgezeigt.

Sofern der Kext von ihm für ManualMode funktioniert, brauchst Du an der Platforms eigentlich gar nicht mehr fummeln.

Für Deine Multi-Channel-Config musst Du ja nur noch die Outputs verbinden, mittels macOS/Dienstprogramme/Audio-Midi-Setup mittels unten + Zeichen (Gerät mit mehreren Ausgängen erstellen).

Hier kannst Du auch die Pegel für jeden Anschluss einstellen.

hier als Beispiel die ID 15, welche Du ja aktuell verwendest. Hier 20, 21, 22 und 23 verbinden, z.B..

Pin Configurator

01471C10 01471D40 01471E11 01471F91 01B71C20 01479C02 01B71D40 01B71E21 01B71F02 01B79C02 01B71C30 01B71D90
01B71E81 01B71F91 01571C30 01571D10 01571E21 01571F01 01571C60 01571D60 01571E01 01571F01 01771C70 01771D30
01771E01 01771F01 01B71C80 01B71D90 01B71E81 01B71F02 01A71C90 01A71D30 01A71E81 01A71F02

Id	PinDefault	Device	Location	Port	Connector	Color	G	P
30 (14A)	9111401D	Out Speaker	Back	Internal + Jack	1/8" stereo/mono	Green	1	0
27 (18A)	0221402D	Out Headphone	Front	Jack or ATAPI	1/8" stereo/mono	Green	2	0
34 (18A)	91a1903D	In Microphone	Back	Internal + Jack	1/8" stereo/mono	Pink	3	0
21 (18A)	0101105D	Out Line Out	Back	Jack or ATAPI	1/8" stereo/mono	Black	5	0
22 (18A)	0101606D	Out Line Out	Back	Jack or ATAPI	1/8" stereo/mono	Orange	6	0
23 (17A)	0101207D	Out Line Out	Back	Jack or ATAPI	1/8" stereo/mono	Gray	7	0
25 (18A)	0281908D	In Line In	Front	Jack or ATAPI	1/8" stereo/mono	Pink	8	0
26 (1AA)	0181309D	In Line In	Back	Jack or ATAPI	1/8" stereo/mono	Blue	9	0

Fraglich nur, ob Du damit auf 5 Outputs kommst, die auch das entsprechende machen. Jeder Anschluss hat ja eigentlich eine Funktion (Front, Center, etc.).

Eines muss Dir dabei klar sein, die verwendete LayoutID 15 habe ich ursprünglich für einen MSI-Rechner entwickelt.

Bis jetzt ist noch immer unklar, wie viele farbliche Ausgänge Dein Rechner tatsächlich hat für Multi-Channel.

Weder der erste Dump vom PinConfigurator2, noch der letzte Dump von Clover F8 sind für mich zu gebrauchen.

Diese Dumps zeigen ohnehin nur, was aktuell konfiguriert ist und nicht, was der Rechner tatsächlich kann.

Ein echter Codec_Dump wird gemacht, in dem man mit einer Linux-Live-Version (z.B. Ubuntu) bootet, welcher einen ALSA-Audio-Treiber hat.

Dort gibt man im Terminal folgenden Befehl ein und erhält den codec_dump:

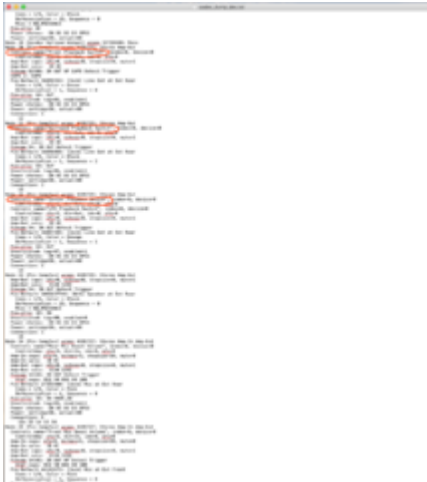
Code

1. `cd ~/Desktop && mkdir CodecDump && for c in /proc/asound/card*/codec#*; do f="{c/V*/card/card}"; cat "$c" > CodecDump/{f//V/-}.txt; done && zip -r CodecDump.zip CodecDump`

Anders als [uern](#) nutze ich sehr gern ein paar Tools dafür, z.B. jage ich jeden Linux-Dump erst einmal durch das Tool `codegraph` und erhalte einige weitere nützliche Dateien mit Infos.

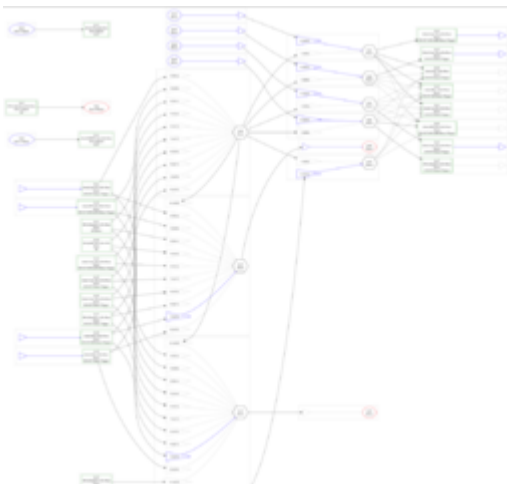
Ich habe zufällig noch den codec_dump vom MSI B150M MORTAR hier, für den ich ebenfalls alc892 ID17 SwitchMode und ID18 ManualMode konfiguriert habe.

nur als Beispiel:



Mit einem echten Dump bekommst Du auch diese Infos. Vergleiche es mal selbst mit Deinen geposteten Dump's!

Ferner hat man dann auch eine .svg - Datei, welche stark verkleinert dann so aussieht:



Mit all diesen Dateien bekommt man auch ziemlich gut die entsprechenden Knotenkettens für alle Anschlüsse.

Natürlich braucht ein erfahrener User wie [wern](#) kein Tool für die Hex-Umrechnung, ich auch nicht, allerdings ist dies eine ganz andere Geschichte.

Fakt ist aber, dass wir ohne echten Dump und entsprechende Hardwareangaben der tatsächlichen Ausgänge für Multi-Channel Deines Rechners eigentlich rein gar nichts sagen können.

Es gibt Rechner, die haben hinten 3 Anschlüsse für Multi, einige haben 4, neuere sogar 5.

Einige nutzen (zumindest unter Windows mittels echtem Treiber) teilweise die Frontanschlüsse oder sogar die Input-Anschlüsse für eine Multi-Channel-Config.

Letzteres mit den Inputs könnte unter OSX schwierig werden.

Was ich eigentlich damit sagen will, wenn Du eine LayoutID verwendest, die nicht explizit für Deinen Rechner gepatcht wurde, dann mag es für Stereo sicher reichen und vielleicht auch im gewissen Rahmen für Mehrfachanschlüsse, jedoch es muss und kann nicht immer perfekt sein.

Ich habe oben echt nicht alles verstanden. Dein 4.4 Setup hebt die Sache evtl. unter macOS auf ein schweren Level.

Ich kann leider nicht sagen, ob dat alles so möglich ist mit macOS, wie Du es oben beschrieben hast.

Beitrag von „e4535030“ vom 25. April 2020, 21:46

Danke für die Erklärung.

Jetzt versteht es auch ein DAU wie ich!

Soweit Funktioniert auch alles.

Der Codec hat 5 Dacs und ich kann alle 5 unabhängig voneinander benutzen. Perfekt!

Beitrag von „atl“ vom 26. April 2020, 22:01

Hallo.

Ich hänge mich hier mal dran, denn ich habe ein, zwei Fragen:

Wenn ich den Thread richtig verstanden habe, kann ich Kopplung des Front-Audio-Ausgang bei mit dem Line-Out am Mainboard ausschalten und dadurch den Line-Out auch benutzen, wenn am Frontausgang ein Gerät angeschlossen ist? Hintergrund: Ich habe einen kleinen Lautsprecher am Frontausgang dran, möchte ggf. (manuell) auf den Line-Out mit angeschlossener Anlage umschalten können.

Kann man den "Kopfhörer"-Ausgang (bei eingestecktem Lautsprecher am Front-Ausgang heißt der so) umbenennen (in "integrierter Lautsprecher", oder so)?

Danke.

Beitrag von „e4535030“ vom 26. April 2020, 22:06

Hey,

Soweit ich das gesehen habe ist die Namensgebung nur von der Pin Config abhängig.

Die kannst du auswechseln ohne die AppleALC neu zu kompilieren.

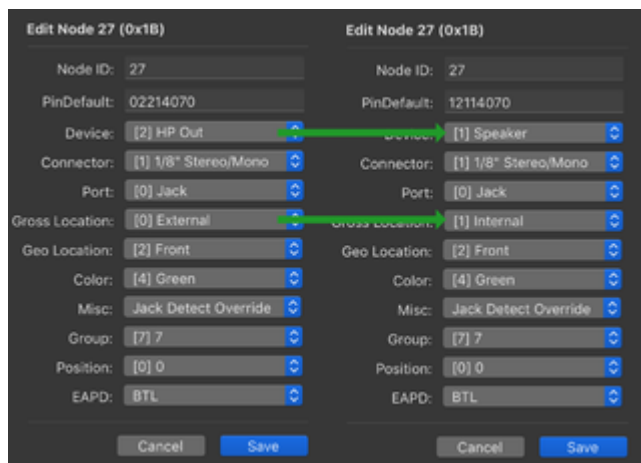
Dazu einfach mit dem PinConfigurator die AppleALC öffnen, deinen Codec+Layout ID wählen.

Dann kannst du im Fenster die Anschlüsse sehen und umbenennen.

Wenn du beide Ausgänge händisch umschalten willst dann musst du die PlatformXX.xml modifizieren und die ALC neu Kompilieren.

Beitrag von „atl“ vom 26. April 2020, 23:33

[e4535030](#) , danke. Okay, dann versuche ich erst einmal das Umbenennen. Ich habe jetzt mal die Einstellungen für den Front-Ausgang angepasst:



Ich hoffe, das paßt so. Dann habe ich die `PinConfigs.kext` aus dem AppleALC-Repo von Github geladen und die Änderungen da drin gespeichert. Die Kernelextension habe ich im OpenCore eingetragen, aber es wird immer noch **Kopfhörer** als Bezeichnung angezeigt. 🤔

Beitrag von „e4535030“ vom 27. April 2020, 01:18

Ich hab das folgendermaßen gemacht:

AppleALC aus GitHub geladen [Link](#)

PinConfigurator aus GitHub geladen [Link](#)

AppleALC auf desktop entpackt.

PinConfigurator gestartet.

File -> Import -> PinConfigs.kext -> Fenster zur Datei Auswahl öffnet sich -> AppleALC vom Desktop angeklickt.

Jetzt Zeigt dir der PinConfigurator die PinConfigs der AppleALC an.

Dort suchst du deinen Codec und wählst dann die Layout ID die du aktuell verwendest.

Dann kannst du das genauso wie dein Screenshot ist ändern.

Fertig.

Nur noch den PinConfigurator schließen und die AppleALC vom desktop mit der die du verwendest ersetzen.

Bei OpenCore weiß Ichs nicht, bei Clover kommt sie ins EFI.

Die PinConfigs.kext brauchst du dafür nicht.

Welchen Codec verwendest du den? Dann würde ich für dich die Manual AppleALC kext Kompilieren, so als Übung.

Beitrag von „atl“ vom 27. April 2020, 01:25

[e4535030](#),

das ist der Realtek ALC892 und Layout-Id 1.

Beitrag von „e4535030“ vom 27. April 2020, 02:11

Hallo,

Also Ich hab jetzt den HP out in Speaker Internal geändert und manuelles umschalten hinzugefügt.

Wenn ich alles richtig gemacht habe sollte das hier funktionieren:

EDIT:

Mir ist grad aufgefallen das das Eventuell verwirrend sein könnte. Mit der ersten Kext hast du 2x Speaker, Ich hab das aktualisiert, der Hintere Anschluss ist nun nen "Line Out" und der Front Audio Anschluss ist nun "Speaker intern", Manuell umschaltbar.

Beitrag von „atl“ vom 27. April 2020, 07:59

[e4535030](#) funktioniert fast. 😊 das Umbenennen hat geklappt, aber Ton höre ich nur, wenn ich auf "Line Out" schalte, dafür aber auf beiden Anschlüssen (vorn und hinten) gleichzeitig. Wähle ich "interne Lautsprecher", bleiben beide stumm. 😞

Beitrag von „e4535030“ vom 27. April 2020, 10:17

Dann passen die Nodes nicht und es wird falsch gerouted.

Das sollte aber kein Problem sein, ich werde gleich nochmal neu kompilieren.

EDIT:

Jap wie gedacht, ich hab die Nodes nicht passend gesetzt. Beide Ausgänge haben die selben DACs benutzt dann ists klar.

Hier jetzt in Korrekt.

Beitrag von „atl“ vom 27. April 2020, 11:52

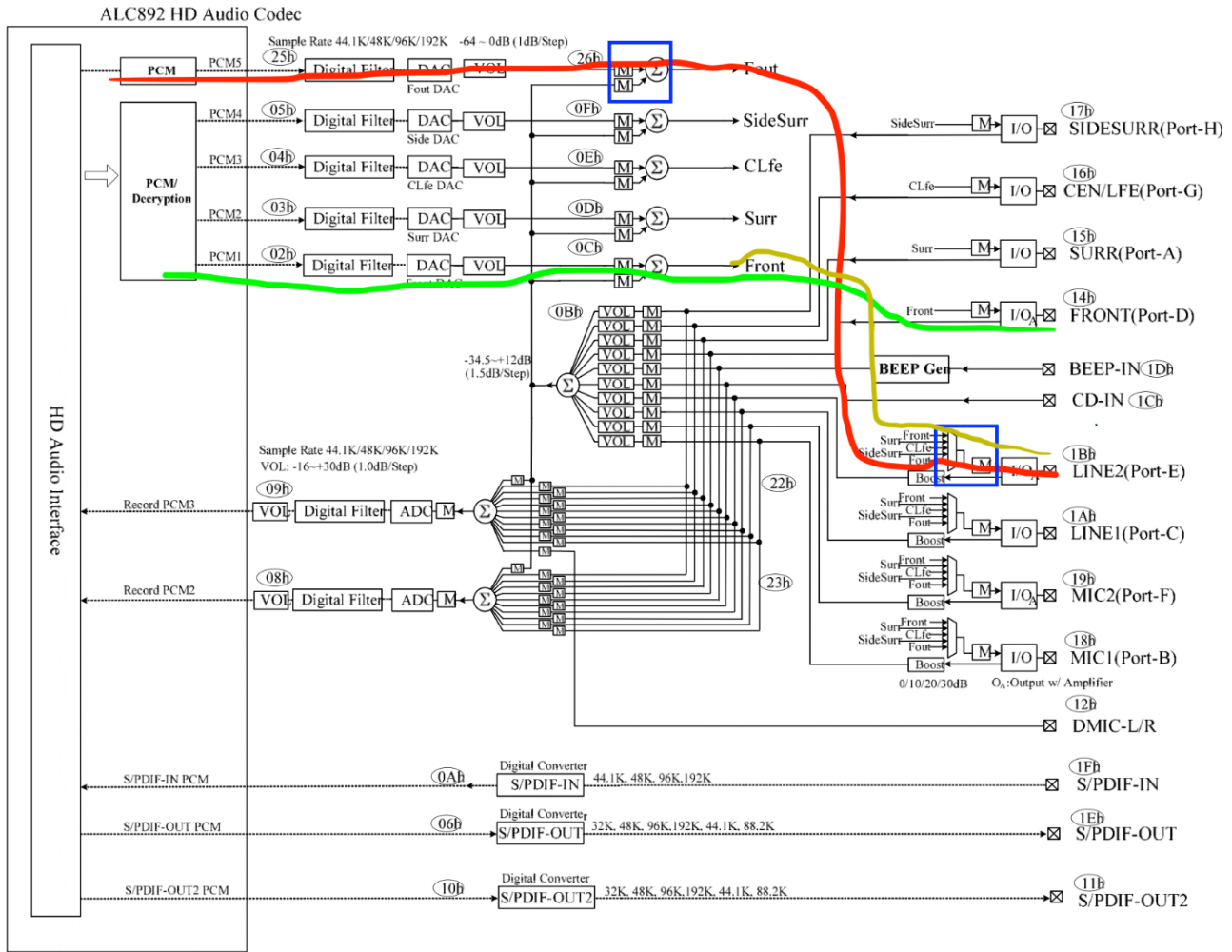
[e4535030](#) , es funktioniert! Vielen Dank! 😄

Kannst du mir noch (kurz) erklären, was du genau gemacht hast bzw. welche Änderungen du vorgenommen hast? Dann kann ich das zukünftig selber machen. Den Thread-Anfang hier habe ich leider nicht wirklich verstanden. 😞

Beitrag von „e4535030“ vom 27. April 2020, 13:45

Huhu, ich kann probieren, bin aber auch noch Anfänger.

Schauen wir uns mal das Blockdiagramm vom ALC892 an:



Ein Blockdiagramm ist das "Innenleben" von dem Codec = Chip der auf dem Mainboard sitzt

Da sieht man das der Codec auf der Linken Seite 5 DACs und 2 ADCs hat.

DAC= Digital -> Analog Converter / ADC Analog -> Digital Converter

Auf der Rechten Seite siehst du die Ausgangs Ports, das sind Pins am Chip die ihn mit der Außenwelt verbinden.

Davon müssen aber nicht alle verbunden sein. Es gibt z.B. Mainboards die nur 3 Klinkenbuchsen an der Rückseite haben. Diese würden dann z.B. nur Port C / D / F verwenden.

Mit dem PinConfigurator stellst du ein welche Ports dein Mainboard hat z.B Front Audio usw

Darin sind die Namen gespeichert die OSX verwendet, auch die Bezeichnung "Intern" kommt daher.

Du kannst natürlich nur Ports konfigurieren die dein Mainboard auch besitzt (oder du musst mit dem Lötkolben dran und Buchsen nachrüsten!).

Inwiefern OSX die Werte verwendet weiß ich nicht. Da wird z.B. auch die Farbe der Buchsen angegeben, ich kenne kein Programm welches diese Werte verwendet.

Die Knoten oder Nodes übernehmen das Routing vom Signal.

z.B. Die Grüne Linie.

Diese geht vom DAC über 2 mit (M) markierte Mischer Knoten an den Port D.

Die Blauen Kästchen markieren die Knotenpunkte.

Da siehst du auch das man z.B. Port B als Mic input oder als Fout konfigurieren kann. Also entweder Eingang oder Ausgang. Je nach dem geht das Signal dann entweder zum DAC oder zum ADC.

In deiner Original Config ist der Port D mit Port E "verbunden" das bedeutet sobald der Codec mitbekommt das du in E einen Stecker einsteckst Routet er das Signal um. Die grüne Linie wird am (M) kurz vor dem Port unterbrochen, wie die Gelbe Linie geführt und am Blau markiertem Knoten auf den Port E gelegt.

Wenn du jetzt beide gleichzeitig verwenden willst musst du die rote Linie realisieren. Diese benutzt einen anderen DAC für den Port E.

Was hab ich gemacht:

PinConfig angepasst

1. Mir das Github von der AppleALC auf den Desktop gelegt.
2. PinConfig geöffnet und bei File die PinConfigs.kext aus dem AppleALC Ordner geöffnet.
3. Deinen Codec und deine LayoutID im Fenster gesucht.

4. PinConfig öffnet jetzt die dazugehörige Port Konfiguration, diese sollte deinem Mainboard entsprechen. Die Daten dafür bekommst du aus dem Dump e.g. welche Physikalische Buchse mit welchem Codec Port elektrisch verbunden ist. Da deine PinConfig ja ok war hab ich nur die Namen geändert:

NID	PinDefault	Device	Connector	Port	Gross Location	Geometric Location	Color	G	P	EAPD
20 (0x14)	01014010	Out Line Out	1/8" Stereo/Mono	Jack	External	Rear	Green	1	0	0x2
21 (0x15)	01011020	Out Line Out	1/8" Stereo/Mono	Jack	External	Rear	Black	2	0	-
22 (0x16)	01016030	Out Line Out	1/8" Stereo/Mono	Jack	External	Rear	Orange	3	0	-
24 (0x18)	90A09040	In Mic In	Unknown	Fixed	Internal	N/A	Pink	4	0	-
26 (0x1A)	01813050	In Line In	1/8" Stereo/Mono	Jack	External	Rear	Blue	5	0	-
25 (0x19)	02819060	In Line In	1/8" Stereo/Mono	Jack	External	Front	Pink	6	0	-
27 (0x1B)	90100070	Out Speaker	Unknown	Fixed	Internal	N/A	Unknown	7	0	0x2
30 (0x1E)	014B6190	Out SPDIF Out	Combination	Jack	External	Rear	Orange	9	0	-
23 (0x17)	400000F0	Out Line Out	Unknown	No Connection	External	N/A	Unknown	15	0	-
31 (0x1F)	400000F0	Out Line Out	Unknown	No Connection	External	N/A	Unknown	15	0	-
17 (0x11)	400000F0	Out Line Out	Unknown	No Connection	External	N/A	Unknown	15	0	-

5. Die Ports so benennen wie du möchtest und es für dich Sinn ergibt. Ich weiß nicht inwiefern man da z.b. auch Speaker Extern konfigurieren könnte und ob OSX damit klar kommt. Ich hab bis jetzt nur Standart sachen eingetragen. Das ist zum teil nur Kosmetisch. Solange die Ports da sind und Eingänge / Ausgänge richtig sind ists scheinbar ok.

6. PinConfigurator -> File->export PinConfigs.kext und damit die alte aus dem AppleALC Ordner überschreiben, bestätigen das dein Layout schon da ist und das du es überschreiben möchtest.

7. Fertig mit PinConfigurator

Knoten geändert

zur Pinconfig gehört auch immer die LayoutID.

Diese findest du im AppleALC->Resources->Codename.

Da du weißt das du Layout ID 1 verändern möchtest schaust du danach, was finden wir da: die layout1.xml aber von der Platform1.xml ist keine spur.

Wie das mit der layoutX.xml funktioniert weiß ich nicht genau. Scheinbar sind da z.b. die Funktionen der knoten festgelegt, manche können z.b. in Hardware Filtern oder Mischen.

In der Info.plist findest du aber das deine LayoutID sich auf die PlatformT.xml referenziert um die Knoten zu bekommen.

In der layout1.xml findest du das diese die Pathmap 100 verwendet.

Jetzt kannst du die PlatformT.xml öffnen und dort die Pathmap 100 öffnen.

Darin findest du die ganzen Ports wieder. Wenn da unter einem Eintrag 2 Untereinträge sind werden diese umgeschaltet. An der ersten NodeID siehst du welcher Ausgang aus dem PinConfigurator das ist. Die 2 nodes drunter sind die Knoten durch die das Signal geht.

Jetzt erzeugst du einen neuen Eintrag und kopierst den Untereintrag der nicht mehr umschalten soll in deinen neuen Eintrag. Danach musst du noch die Knoten für die rote Linie setzen, sonst verwenden beide Ausgänge die selben Knoten und DACs (Grüne Linie geht auf beide Ports) Damit verhält es sich so wie du es beschrieben hast. Logisch soweit.

Die Knoten kannst du aus deinem Dump oder einer anderen Layout ID die alle Ports benutzt extrahieren.

Dann Speichern und mit xCode kompilieren.

Äh kurz ;D

Vielleicht kann jemand von den Profis noch was zur layoutX.xml sagen, bei den DSP functions seh ich nämlich noch garnicht durch. Bei.z.b Mic ist nen SoftwareDSP festgelegt, das ist wahrscheinlich nen AU plugin. Beim LineIn ist aber nen MuteGPIO festgelegt, der kommt irgendwo aus der Hardware.

Credit to [wern](#) und [MacPeet](#)

Beitrag von „atl“ vom 27. April 2020, 13:56

[e4535030](#) Vielen Dank für die ausführliche Erklärung. Ich werde das bei Gelegenheit noch mal selber probieren, um zu sehen, ob ich's verstanden habe. 😊

Beitrag von „MacPeet“ vom 27. April 2020, 17:29

[e4535030](#)

Na, geht doch 😊 und Du konntest sogar schon einem User helfen, prima.

Betreffs, LayoutID's.xml, kannst Du Dich an bereits vorhandenen ID's des jeweiligen ALCxxx richten, was Du ändern möchtest. In der Regel passt dat so.

Zitat: Mic ist nen SoftwareDSP festgelegt

Mic-Anschlüsse sind regelbar in der Software, daher die Einträge.

Ich mache diese Einträge sogar oft auch auf LineIn, da ja oft ein Mic-Anschluss unter MacOS als LineIn konfiguriert wird und nur einer als intern Mic verbleibt.

Der tatsächliche blaue LineIn-Anschluss ist ohnehin nicht regelbar (ist auch unter Windows so), die eingesteckte Quelle bestimmt das Signal.

Es ist vergleichbar mit digitalen Ausgängen (HDMI/Optical Audio) auf dem Mac. Diese sind ohne Zusatzsoftware in der Menüleiste nicht mehr regelbar. Hier übernimmt das angeschlossene Gerät.

Thema MuteGPIO :

In ID1 alc892 ist für mic und line MuteGPIO gesetzt. Diese Werte sind auch in der Regel so richtig. Die kommen nicht von der Hardware, sondern sind hier so gesetzt.

Wenn Du Dich mal durch andere ALC-Devices der AppleALC arbeitest, dann siehst Du, dass es ggf. dort ganz anders aussehen kann.

Einige Devices brauchen gar kein MuteGPIO als Eintrag, andere brauchen wieder MuteGPIO=0 oder sogar MuteGPIO=0 auf den Outputs.

Vieles kann man nicht so einfach beschreiben, es ist oft viel testen und versuchen, bis man das beste Ergebnis hat.

Beitrag von „wern“ vom 1. Mai 2020, 14:14

[Zitat von MacPeet](#)

Weder der erste Dump vom PinConfigurator2, noch der letzte Dump von Clover F8 sind für mich zu gebrauchen.

Diese Dumps zeigen ohnehin nur, was aktuell konfiguriert ist und nicht, was der Rechner tatsächlich kann.

Ich weiss nicht ob HdaCodecDump.efi alles ausliest, zeigt aber das was der Hersteller konfiguriert hat. Meistens unvollständig und fehlerhaft. Das ist auch der Grund warum VoodooHDA schlecht oder fehlerhaft funktioniert, VoodooHDA arbeitet nach den Intel HDA Spezifikationen. Der Linux Treiber nutzt fast immer Patches. [Hier zB. für Realtek](#), es wird fast jeder Codec korrigiert, deswegen ist der Linux-dump auch die bessere Alternative.

Dies ist mein Wissensstand, wenn falsch bitte korrigieren.

Werde später noch andere Kommentare hinzufügen.

Beitrag von „MacPeet“ vom 1. Mai 2020, 14:26

[Zitat von wern](#)

..., deswegen ist der Linux-dump auch die bessere Alternative.

genau so ist dat.

Beitrag von „atl“ vom 2. Mai 2020, 13:34

Hallo [e4535030](#), vielen Dank noch einmal! Ich habe mir das Ganze noch mal einmal komplett durch gelesen und mich - in einer ruhigen Nacht - selbst mal daran versucht. Nach einigen kläglichen Versuchen habe ich jetzt endlich meine eigenen `AppleALC.kext` am Laufen, die genau das Gleiche tut, wie deine. Und ich glaube jetzt verstanden zu haben, wie es funktioniert, so dass ich das beim nächsten Mal alles (hoffentlich) selber hinbekomme.

Aber natürlich auch Dank an die anderen beiden, [wern](#) und [MacPeet](#), deren Beiträge viel zum Verständnis beigetragen haben. 😊

Beitrag von „atl“ vom 12. Juli 2020, 14:51

Hallo an die hier versammelte Kompetenz, ich habe da noch einmal ein paar Fragen:

1. Kann ich das Audio-Layout so verändern, dass ich aus dem Front-Audio/Mic-In einen weiteren Audio-Ausgang mache?
2. Kann man aus einem Stereo-Ausgang einen Mono-Ausgang machen?
3. Bei so vielen Änderungen am Layout macht es doch Sinn ein eigenes zu erstellen, oder? Was muss ich da beachten?

Beitrag von „MacPeet“ vom 12. Juli 2020, 14:57

Wo ist der Sinn bei Deinen Fragen? Wo soll dieses hinführen, bzw. was soll es bewirken?

Beitrag von „e4535030“ vom 12. Juli 2020, 18:44

Hallo,

unabhängig davon wieso du das machen willst ist das je nach Codec möglich.

Manche Leute wollen halt nicht gleich nen Audio Interface benutzen nur um mehr Ausgänge zu bekommen.

Wenn man sich die Datenblätter der AudioCodecs anschaut sind die meisten auch qualitätsmäßig vollkommen ausreichend. (vorausgesetzt der MB Hersteller hat nicht geschlampt!)

Dafür solltest du mal in das Datenblatt von deinem Codec schauen und rausfinden welcher Port was kann.

Auf dem Screenshot vom Blockdiagramm auf dieser Seite sieht man z.b. das die Mic In Eingänge auf nen Ausgang gepatched werden können.

Dafür würde ich erstmal mit dem PinConfigurator schauen welchen Eingang du zum Ausgang machen möchtest. Wenn du die Port Nummer weißt kann man schonmal schauen ob er eignet ist.

In dem aLC892 Blockdiagramm ist ersichtlich das alle Eingänge außer der CD in geeignet wären.

Du brauch aber auf jeden fall noch einen freien DAC im chip, der muss dann passend zur Buchse gerouted werden.

Zu 2. das ist ne gute frage

Du könntest im Audio Midi Setup beide Ausgangskanäle auf den ersten oder zweiten Kanal legen. Dann kommt an beiden Kanälen etwas raus, beides ist das selber. Aber es wird nicht nach Mono gemischt.

Du könntest auf jeden fall Audio Hijack verwenden um aus Stereo Mono zu mischen und auszugeben.

Ansonsten ist das eigentlich selten nötig. Wenn du z.b. ein Mono Gerät einsteckst (was auch einen Mono Stecker hat) wird automatisch elektrisch gemischt. Das ist vorgegeben durch das Buchsen/Stecker design. Davon bekommt deine Soundkarte/OS aber nichts mit.

Nur bei Geräten mit Stereo Stecker die aber nur einen Kanal verwenden wäre es nützlich, oder wenn man Stereo Effekte nicht mag.

3.

Du kannst einfach eine unbenutzte Layout ID verwenden.

Ich benutz einfach eine verwendete Layout ID die grundsätzlich funktioniert. Dann geht alles außer die "extras" nach nem kextupdate sofort. Ich tausch dann einfach meine gespeicherte layout Datei aus und alles läuft.

Beitrag von „5T33Z0“ vom 12. Juli 2020, 18:57

[Zitat von atl](#)

Hallo an die hier versammelte Kompetenz, ich habe da noch einmal ein paar Fragen:

1. Kann ich das Audio-Layout so verändern, dass ich aus dem Front-Audio/Mic-In einen weiteren Audio-Ausgang mache?
2. Kann man aus einem Stereo-Ausgang einen Mono-Ausgang machen?
3. Bei so vielen Änderungen am Layout macht es doch Sinn ein eigenes zu erstellen, oder? Was muss ich da beachten?

1. Ich denke nicht, denn ein Audio Eingang ist ja etwas physisch fix, monodirektional. Die "Leitung" funktioniert nicht bidirektional. Ansonsten hieße es ja "Eingang/Ausgang nach Belieben". 😊

2. Ja, indem du deine Audio-Quelle intern auf Mono summierst und ausgibst. Sollte funktionieren mit Rogue Amoeba Loopback oder Soundflower (kostenlos)

3. I don't know.

Beitrag von „atl“ vom 12. Juli 2020, 21:36

[Zitat von MacPeet](#)

Wo ist der Sinn bei Deinen Fragen? Wo soll dieses hinführen, bzw. was soll es bewirken?

Kurz zum Hintergrund: Ich nutze als Gehäuse das eines PowerMac G4. Dieses hat keine Frontanschlüsse aber einen eingebauten Mono-Lautsprecher. Den habe ich - dank [e4535030](#) - an den den internen Frontanschluß angeschlossen und per Anpassung am Routing so entkoppelt, dass der Line-Out am Board nicht deaktiviert wird. Diesen Ausgang wollte ich auf „Mono“ umstellen, in der Hoffnung, das System erkennt das und die Balanceregung ist nicht mehr möglich.

Desweiteren möchte ich noch einen zusätzlichen Audioanschluß mit Verstärker einbauen, um die Pro Speaker da anzuschliessen. Auch da möchte ich nicht einen hinteren Anschluß verwenden, da ich dann das Kabel nach innen führen müsste.

Danke [e4535030](#), ich schaue mir das mal an. 😊

Beitrag von „e4535030“ vom 13. Juli 2020, 20:23

[Zitat von 5T33Z0](#)

1. Ich denke nicht, deinn ein Audio Eingang ist ja etwas physisch fixes, monodirektionales. Die "Leitung" funktioniert nicht bidirektional. Ansonsten hieße es ja "Eingang/Ausgang nach Belieben". 😊

Das ist auf jeden fall nicht korrekt.

Die meisten Codecs können beliebig konfiguriert werden. Sieh dir die rechte Seite vom Blockdiagramm oben an. Rechts die Ports haben Pfeile rein und raus = Konfigurierbar wie man will.

Beitrag von „5T33Z0“ vom 13. Juli 2020, 20:54

[Zitat von e4535030](#)

Das ist auf jeden fall nicht korrekt.

Die meisten Codecs können beliebig konfiguriert werden. Sieh dir die rechte Seite vom Blockdiagramm oben an. Rechts die Ports haben Pfeile rein und raus = Konfigurierbar wie man will.

Dem Codec mag das vielleicht egal sein, wo das Signal herkommt, aber dem Interface nicht. Zumindest müsste man da was modden am Board, damit das funktioniert. Mit Software Reroutings geht das glaub ich nicht.

Ich weiss ja nicht wofür das genutzt werden soll, aber falls für Musikproduktion, würde ich lieber n Paar Euro in ein vernünftiges Audio-Interface investieren als ma Board rumzufrickeln.



Beitrag von „g-force“ vom 13. Juli 2020, 21:19

Könnt ihr Euch die Vollzitate mal sparen bitte?

Wozu man einen Desktop-Hackintosh mit einem Monolautsprecher-Onboard betreiben möchte, entzieht sich allerdings nun auch meinem Verständnis.

Beitrag von „e4535030“ vom 14. Juli 2020, 14:24

[Zitat von 5T33Z0](#)

Dem Codec..... glaub ich nicht.

Ich glaub du verstehst das Falsch.

Im Codec gibt es ADCs und DACs = Analog zu Digital und umgedrehte Wandler.

(Blockschaltbild Linke Seite)

Diese Wandler sind in der Software/dem OS die ein Ausgabe und Eingabe Kanäle.

Der Codec hat dann auch noch "Ports" oder anders Physikalisch Anschlüsse am Chip (Rechte Seite Blockschaltbild)

Dazwischen sind die Mischer und das Routing.

Auf dem Mainboard ist dann in der Regel nur eine 3,5mm Klinkebuchse und ein paar Entkoppelkondensatoren, das ganze hängt dann an den physikalischen Anschlüssen vom Codec.

Den Entkoppel Kondensatoren und den Klinkebuchsen ist prinzipbedingt egal in welche Richtung das Signal da durch fließt, den Leiterbahnen auf dem Mainboard auch.

Wenn du dir jetzt im Blockdiagramm die rechte Seite z.B. Port-C anschaust siehst du das davor direkt ein Mischer sitzt der auch das Signal "Front" und "Surr" bekommt. Man kann den Codec jetzt so konfigurieren das der Mischer vor Port-C das Front signal vom DAC auf Port-C raus läßt. Dann ist dein Eingang plötzlich nen Ausgang.

Im Datenblatt: [Datasheet](#)

Auf Seite 2 unter 2.1 steht auch ausdrücklich:

Zitat

- All analog jacks (port-A to port-G) are stereo input and output re-tasking

Das hat mit dem Mainboard nichts zutun. Das Mainboard ist in dem fall vereinfacht ein einfaches "Kabel" zwischen den Buchsen hinten und dem Codec. Dem ist das vollkommen egal in welche Richtung die Signale fließen.

Beitrag von „5T33Z0“ vom 14. Juli 2020, 20:55

Ah, ok, ich dachte, das wäre das physische board layout auf dem Diagramm. Danke für die Erläuterungen. Na dann happy coding 😊

Beitrag von „g-force“ vom 14. Juli 2020, 22:05

Ich lasse mich in diesem Thread überraschen, ob ein D/A-Wandler per Konfiguration zu einem A/D-Wandler wird.

Beitrag von „apfelnico“ vom 14. Juli 2020, 23:43

[g-force](#)

Wohl nicht. 😊

Aber beide Einheiten sind ja vorhanden, und so wie ich es verstanden habe, sind die physischen I/O relativ frei konfigurierbar, so dass ein vom Hersteller vorgesehener "Ausgang" nicht unbedingt einer bleiben muss und umgekehrt.

Beitrag von „e4535030“ vom 15. Juli 2020, 01:54

[atl](#)

Gehts immernoch um die ALC892?

Welches Board hast du?

Welchen Eingang möchtest du als Ausgang verwenden?

welche Ausgänge sollen weiterhin normal funktionieren?

Bedenke du hast „nur“ 5 insgesamt zur Verfügung wenn du drüber bist musst du einen anderen Ausgang Opfern oder beide geben das selbe aus.

Dann können wir gemeinsam an einem Layout arbeiten.

Beitrag von „atl“ vom 15. Juli 2020, 02:25

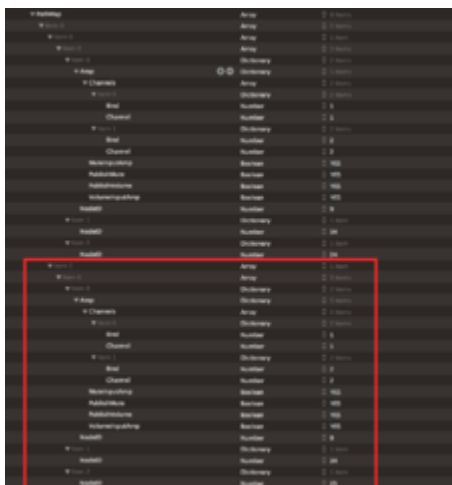
[Zitat von g-force](#)

Wozu man einen Desktop-Hackintosh mit einem Monolautsprecher-Onboard betreiben möchte, entzieht sich allerdings nun auch meinem Verständnis.

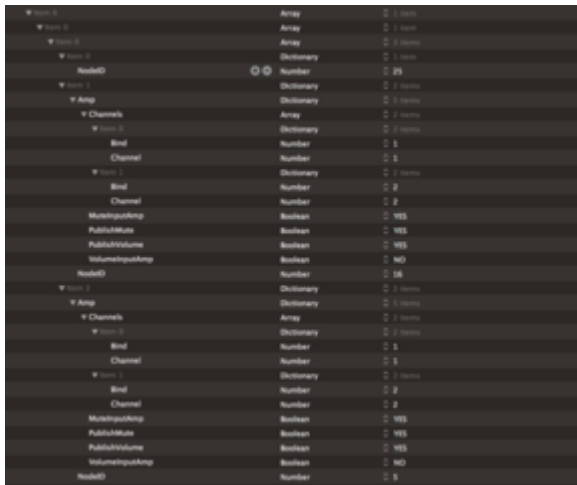
Das ist ganz einfach zu verstehen: Ich habe momentan keine Lautsprecher dran. Die Anlage ist per Airplay angeschlossen und nur aktiv, wenn ich Musik (in Stereo 😊) höre. Wenn ich im Normalfall am Rechner arbeite, dann reicht der Mono-Lautsprecher durchaus für diverse System- und Signaltöne. 😊

[e4535030](#), ja es geht noch um genau das Board, sprich auch den Codec (ALC892) wie in den Beiträgen vom April. Laut momentan verwendeter Layout-ID 1 (PlatformsT.xml, Pathmap 100) ist der Port H (SideSurr, 23 / 17h) nicht durchgeführt. Daraus folgt, DAC 5 (05h) wäre frei und könnte über Mischer / Amplifier 16 (0Fh) weitergereicht werden. Versucht habe ich es im ersten Schritt mit Port F (MIC2, 25 / 19h).

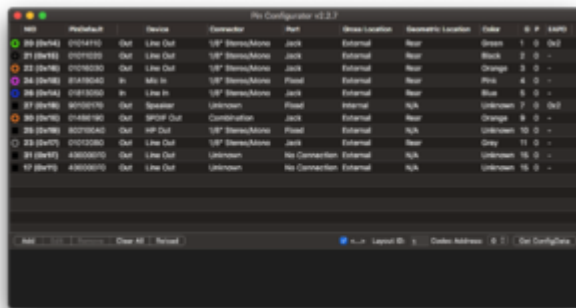
Dieser ist in der Pathmap 100 in PlatformsT.xml schon als Eingang definiert, weshalb ich den roten Bereich entfernt habe:



Danach habe ich ein neues Element / Array hinzugefügt:



Und mit dem PinConfigurator habe ich den Ausgang (als Kopfhörer zur Identifikation) definiert:



Doch leider im Moment noch ohne Erfolg. Allerdings war die Aktivierung von Port-H (17h) durch Routing von 05h über 0Fh auf 17h auch nicht erfolgreich (siehe angehängte PlatformsT.xml.zip). U.U. ist da noch ein anderes Problem. :-/

Beitrag von „e4535030“ vom 16. Juli 2020, 01:00

Hallo Atl!

Das ist super vorarbeit.

Ich bin grade mal dazu gekommen deine PlatformsT.xml zu überprüfen.

Dabei fällt mir auf das grundsätzlich alles richtig ist für deinen "Port-h (17h)" versuch.

Allerdings hast du einen Fehler beim umrechnen oder eintippen gemacht.

17h in dezimal sind (das h läßt man scheinbar immer weg): 23 das ist der output, passt.

05h in dezimal sind: 5, passt!

0fh in dezimal sind: 15!

Hier ist der Fehler, dort hast du 16 eingetragen, das ist in hex 10 und damit der SPIDF out da kommt der codec durcheinander.

Wenn du hier 15 setzt passt das und sollte gehen, Bei mir läuft das zumindest.

Zum Eingang als Ausgang:

Da hast du den selben Fehler gemacht. Ansonsten sollte das so richtig sein!

Richtig wäre von oben nach unten:

25

15

5

Damit würdest du DAC 05h über 0fh zu Port-F (19h) ausgeben.

EDIT:

Da der Port-F auch einen Ausgangs Verstärker enthält (sichtbar im Blockschaltbild "I/Oa", musst du im PinConfigurator auch EAPD setzen.

EAPD steht für external amplifier power down = Was passiert mit dem im Codec integrierten Ausgangsverstärker im standby.

Wenn das nicht gesetzt ist kann es nach Sleep vorkommen das der Ausgang nicht geht da der Verstärker aus ist.

Ich hab das jetzt nicht getestet, kompilieren, Neustarten usw

Probier das erstmal selber, der Tipp/Rechen/Schusselfehler sollte es gewesen sein!

Wenns geht/nicht geht schreib nochmal, dann schau ich genauer.

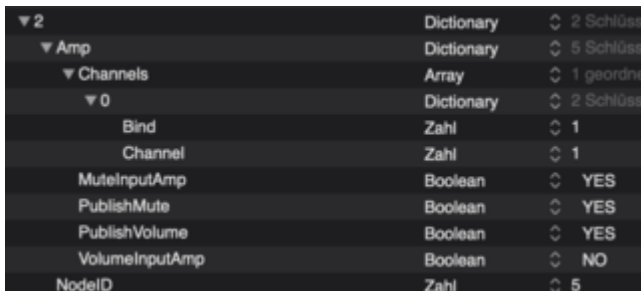
Wegen Mono:

du könntest mal probieren was passiert wenn du das Channel binding änderst. Das sieht für mich so aus als ob der Codec da in Hardware beide Channel zu Mono mischen könnte. Ansonsten mal testen was passiert wenn du das ganze auf einen Channel reduzierst (am DAC 05h) dann erkennt OSX den Ausgang vielleicht als Mono.



▼ 2	Dictionary	⌵ 2 Schlüssel/Wert-Pa
▼ Amp	Dictionary	⌵ 5 Schlüssel/Wert-Pa
▼ Channels	Array	⌵ 2 geordnete Element
▼ 0	Dictionary	⌵ 2 Schlüssel/Wert-Pa
Bind	Zahl	⌵ 1
Channel	Zahl	⌵ 1
▼ 1	Dictionary	⌵ 2 Schlüssel/Wert-Pa
Bind	Zahl	⌵ 1
Channel	Zahl	⌵ 2
MuteInputAmp	Boolean	⌵ YES
PublishMute	Boolean	⌵ YES
PublishVolume	Boolean	⌵ YES
VolumeInputAmp	Boolean	⌵ NO
NodeID	Zahl	⌵ 5

oder



▼ 2	Dictionary	⌵ 2 Schlüsse
▼ Amp	Dictionary	⌵ 5 Schlüsse
▼ Channels	Array	⌵ 1 geordne
▼ 0	Dictionary	⌵ 2 Schlüsse
Bind	Zahl	⌵ 1
Channel	Zahl	⌵ 1
MuteInputAmp	Boolean	⌵ YES
PublishMute	Boolean	⌵ YES
PublishVolume	Boolean	⌵ YES
VolumeInputAmp	Boolean	⌵ NO
NodeID	Zahl	⌵ 5

Beitrag von „atl“ vom 16. Juli 2020, 03:22

[Zitat von e4535030](#)

Ofh in dezimal sind: 15!

Sch..., wenn man doof ist! 🤪 Das war's!

Jetzt funktioniert der Front-MIC als Audio-Ausgang! 😄

Aber macOS scheint nicht mehr als 4 Ausgänge zu nutzen. Der 5. Ausgangspfad (item6) in der PlatformsT.xml funktioniert nicht. An der Buchse 22 (16h) kommt kein Signal an. Vermutlich hängt es mit dem definierten Layout (Layout-ID 1) zusammen. Aber wie und wo ich da einen weiteren Ausgang definieren / hinzufügen kann, habe ich noch nicht gefunden.

Auch nicht geklappt hat, der Versuch den SideSurr gleichzeitig auf zwei Ausgänge (05h -> 0fh -> 19h & 17h) zu legen. Dann könnte ich wahlweise Lautsprecher mit integriertem Verstärker (klassisch) an SideSurr auf der Rückseite anschliessen und Lautsprecher ohne an den (Ex-)Front-MIC.

Das mit dem Mono funktioniert irgendwie auch nicht. Auf der Strecke habe ich ja eine AMP an 37 (25h) und einen AMP an 38 (26h). Was habe ich versucht:

1. 37: Bind 1 => Ch 1 & Bind 1 => Ch 2 und 38: Bind 1 => Ch 1 (gelöscht: Bind 2 => Ch 2)
2. 37: Bind 1 => Ch 1 & Bind 2 => Ch 1 und 38: Bind 1 => Ch 1 & Bind 1 => Ch 2

In beiden Fällen kommt das Signal nur noch aus dem linken Lautsprecher, aber die Lautstärke läßt sich nicht mehr regeln.

Irgendwie fehlt mir das Verständnis dafür, was "Bind" und "Channel" bedeutet. Heißt das rein kommt "Channel" (z.B. 0 -> links) und raus geht "Bind" (d.h. nach dem Mixer "Bind" = "Channel")? Oder umgekehrt? 😞

Beitrag von „e4535030“ vom 16. Juli 2020, 23:45

Hallo atI!

Schön das es das war: Das bedeutet es läuft schonmal!

22/14/4 ist auf jeden fall korrekt. Ich benutz das selbe und bei mir geht der Ausgang (16h) ohne Probleme.

Ich seh da auch kein Problem in der Layout1.xml, das sieht genauso aus wie in der 15 (die benutze ich) abgesehen davon das SPDIF bei der 1 Konfiguriert ist und bei der 15 nicht.

Du kannst mal probieren den Letzten Eintrag (30/6) rauszuschmeißen. Angeblich kann OSX nur 8 In/outputs für einen codec. Ich benutze 7 da SPDIF aus ist.

Vielleicht kann es nur 7?

Ansonsten trag doch mal deine Wunsch Konfiguration unter LayoutID15 ein. Die hab ich bei mir angepasst und ich verwende aktuell 4 Ausgänge gleichzeitig ohne Probleme. Der 5. Funktioniert auch, wird aber aktuell nicht benutzt (ist bei mir 17h).

Ansonsten besteht natürlich die Möglichkeit das Asus auf dem Board nicht den Ausgang 22 für den Orangenen Anschluss benutzt. Manchmal werden da Layout bedingt die Buchsen nicht nach Standart verdrahtet. Es wäre ja möglich das z.b. Port-C (1AH) an der orangenen Buchse angeschlossen ist und du deswegen da nichts rausbekommst. Je nachdem ob genug platz für die Leiterbahnen war hat der Layouter einfach den anderen Port vom codec verwenden. Das kann ja wie man sieht im Treiber einfach korrigiert werden.

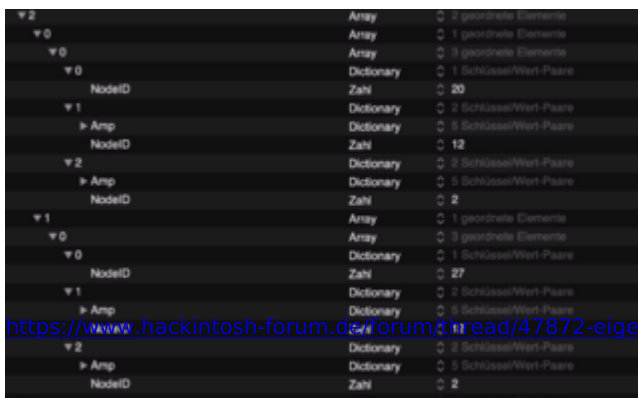
Da könntest du einfach alle Ports ausprobieren bis du weißt welcher Codec Port an welchem Mainboard Anschluss hängt.

Zu 05h auf 19h und 17h:

Wie genau hast du denn Probiert einen DAC auf beide Ausgänge zu Routen?

Bei der ersten version die ich dir erstellt hatte gab es ja ungewollt den Effekt (als ich switch mode falsch konfiguriert hatte) das du Ton auf beiden Outputs hattest als du einen Ausgang angewählt hattest.

Original SM ist ja:



Falsch war das so konfiguriert:

▼ 2	Array	○ 1 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
▼ 0	Array	○ 3 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
NodeID	Zahl	○ 27
▼ 1	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 12
▼ 2	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 2
▼ 3	Array	○ 1 geordnete Elemente
▼ 0	Array	○ 1 geordnete Elemente
▼ 0	Array	○ 3 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
NodeID	Zahl	○ 20
▼ 1	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 12
▼ 2	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 2

Ich würde jetzt entweder das:

▼ 2	Array	○ 1 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
▼ 0	Array	○ 4 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
NodeID	String	○ 27
▼ 1	Dictionary	○ 1 Schlüssel/Wert-Paare
NodeID	Zahl	○ 20
▼ 2	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 12
▼ 2	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 2

Oder das Testen.

▼ 2	Array	○ 1 geordnete Elemente
▼ 0	Dictionary	○ 2 Schlüssel/Wert-Paare
▼ 0	Array	○ 3 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
NodeID	Zahl	○ 20
▼ 1	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 12
▼ 2	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 2
▼ 1	Array	○ 3 geordnete Elemente
▼ 0	Dictionary	○ 1 Schlüssel/Wert-Paare
NodeID	Zahl	○ 27
▼ 1	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 12
▼ 2	Dictionary	○ 2 Schlüssel/Wert-Paare
▶ Amp	Dictionary	○ 5 Schlüssel/Wert-Paare
NodeID	Zahl	○ 2

Ansonsten kannst du beide "normal" einzeln konfigurieren und dann im Audio Midi Setup ein neues "Gerät mit mehreren Ausgängen" erstellen (Links unten + drücken).

Wenn du da jetzt deine beiden Ausgänge anwählst die Parallel ausgeben sollen, dann kannst du in jedem Program oder auch im ganzem OS als Ausgabegerät das "Multiausgangsgerät" wählen. Dann gibt OSX automatisch auf beiden DACs aus. D.h. es kommt aus beiden ausgewählten Ausgängen das selbe.

Zu Binds/Mono:

Dazu kann ich leider nichts genaues sagen, war auch nur raten von mir.

Wenn keiner weiter eine Idee hat kann ich dir eine Elektronische Lösung anbieten:

Du kannst an deinem Mono Lautsprecher auch einfach L und R gleichzeitig anschließen. Die Ausgänge vom Codec werden dann automatisch Elektrisch gemischt. Das würde ein Mono Klinkenstecker in der Stereo Buchse auch tun. L und R sind dort "Kurzgeschlossen" die "Last/Kopfhörer/Verstärker" hängt dann zwischen GND und LR.

Die Ausgänge halten das aus, da der Ausgangswiderstand recht hoch ist.

Beitrag von „atl“ vom 17. Juli 2020, 02:38

[e4535030](#) danke für deine Hilfe. 😊

Also ich habe das Gefühl, dass macOS nur mit 4 Ausgängen umgehen kann - zumindest mit der Layout-ID 1. Die Ports konnte ich alle identifizieren:



Die beiden übrigen sind die Frontanschlüsse. Und es funktionieren auch alle, aber nicht in einer Konfiguration zusammen. Es lassen sich nur 4 (analoge) Ausgänge nutzen, auch wenn ich mehr konfiguriere (mit 5 getestet). macOS zeigt dann auch alle (5) Anschlüsse an, aber bei Wahl des letzten kommt kein Ton und die Wiedergabe bleibt stehen. Das kann natürlich auch ein

Hi, [e4535030](#) ,

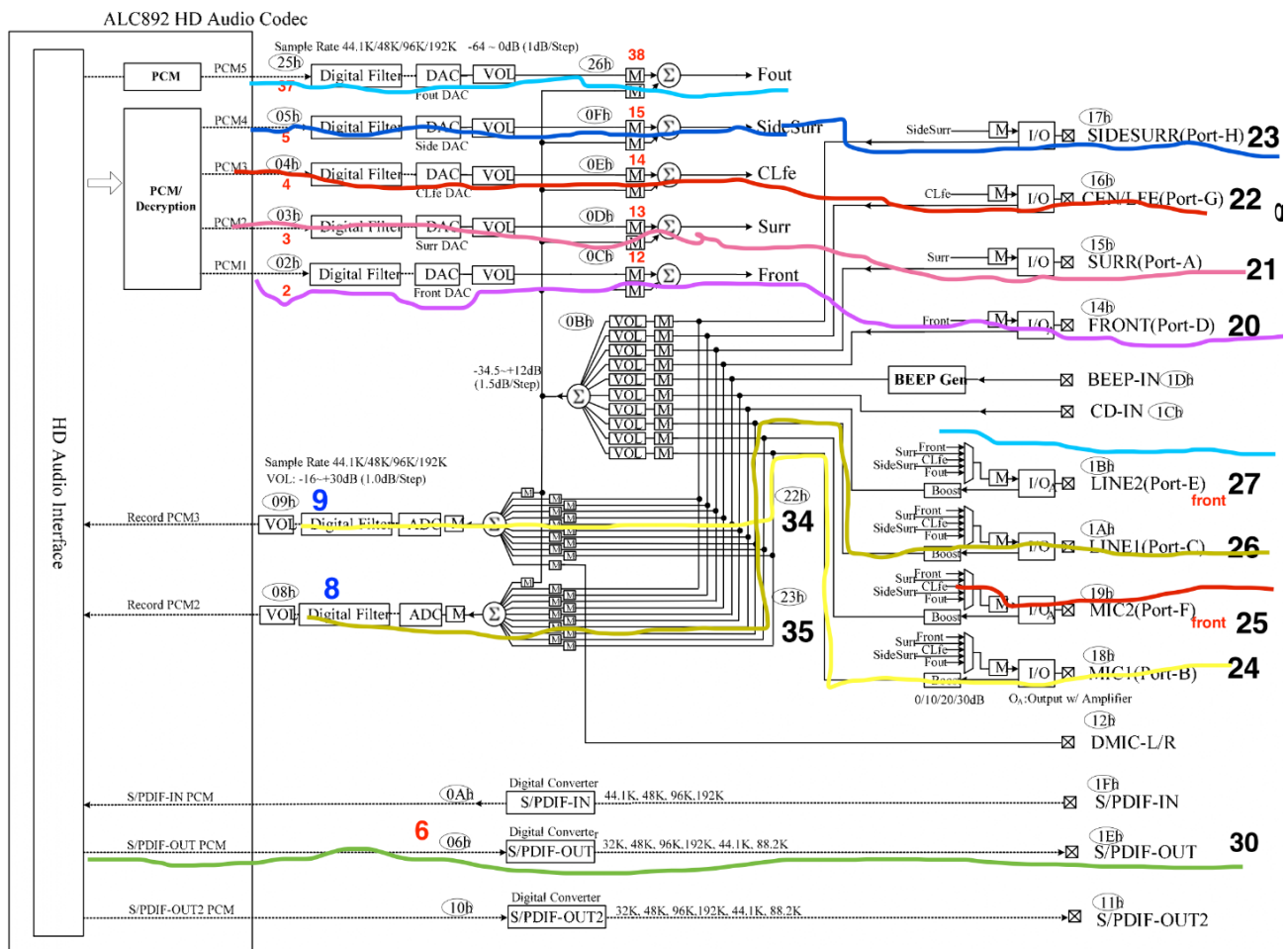
meine „Traum“-Konfiguration wäre folgendes:

- Front-Out (Mono) entkoppelt von Line-Out (internal Speaker)
- Front-In als Stereo-Out und Signal gleichzeitig auf C/Sub
- Line-Out (unabhängig von Front-Out nutzbar)
- Rear (Stereo-Out)
- Side (Stereo-Out)
- SPDIF-Out (optional)
- Mic & Line-In ganz normal als Eingänge nutzbar

Beitrag von „e4535030“ vom 26. Juli 2020, 00:55

Huhu!

Ich hab mal was gebaut:



Das ist das Blockdiagramm mit allen eingezeichneten Routen.

Mono habe ich nicht hinbekommen. OSX erkennt immer 2/Stereo Ausgabe Kanäle.

Gehen würde einen DAC Kanal auf beide Kanäle gleichzeitig zu legen, das eliminiert aber nicht den Balance Regler und gaukelt OSX keinen Mono DAC vor.

Da müssen jetzt die Profis ran.

Wie sagt man OSX wieviele Channel ein Ausgabekanal hat? Das würde auch Modifikationen ermöglichen z.B. aus 3 Stereo DACs ein 5.1 Ausgang in OSX zu machen (Ohne Hauptgerät/Multi Ausgangs Gerät).

Weiß das jemand? [MacPeet](#) vielleicht?

Auch das Rote (Port-F und Port-G) gleichzeitig anzusprechen, scheint schwer zu sein. Die meisten Konfigurationen enden damit das die AppleALC fehlerhaft ist und garkein

Ausgabegerät vorhanden ist.

Ich habe jetzt einfach den DAC für 2 Ausgänge ohne switchmode gesetzt. Das bringt OSX durcheinander, nur der zuerst definierte Ausgang geht, dafür kommt der Ton aber aus beiden Ausgängen. Das bedeutet aber das man einen "kaputten" Eintrag in der Geräteliste hat. Der kann man auch nicht über die PinConfig.kext ausblenden.

Beide Ports als Ausgangsnode zu setzen geht nicht = AppleALC geht garnicht mehr.

[Zitat von atI](#)

Die gleichzeitige Ausgabe auf 17h & 19h hat fast geklappt. Mit der Version habe ich auf beiden etwas gehört:

Aber der Ton spielte ca. 0,5 Sek in doppelter Geschwindigkeit und war dann 0,5 Sek still. Als ob beide Ausgänge im Wechsel Zugriff auf den DAC bekommen. Deshalb habe ich das erst mal wieder zurück gebaut.

Davon hab ich mir am meisten versprochen. So scheint es doch den Codec richtig zu konfigurieren. Leider begrenzt OSX/AppleALC die Funktion. Es scheint so als ob es die Daten 2x zum DAC sendet (der steht ja auch 2 mal drin!)?

Der Ton kommt zeitgleich aus beiden Outputs, aber doppelt so schnell + Lücke. Lücke und Ausgabe sind auch gleichzeitig auf den Outputs, nicht abwechselnd so wie man vermuten würde.

Wenn man den DAC nur einmal einträgt funktioniert die komplette AppleALC nicht mehr. Komisch.

Hat dazu jemand eine Idee?

Interessant wäre auch zu wissen wie man unabhängig von Ausgangsgeräten Routen festlegen kann. Z.b. Line In direkt zu Fout.

Hier mein "fortschritt":

[halb.zip](#)

Auf jeden fall geht:

Alle 5 DACs, auch gleichzeitig.

Front Out als unabhängiger Ausgang leider "nur" Stereo

Front In als Ausgang, Signal kommt gleichzeitig bei C/LFE raus. (Dafür letztes Ausgabegerät wählen)

Line Out als unabhängiger Ausgang

Rear und Side als Ausgang

SPDIF Out als Digitaler Ausgang (Sollte gehen, habe aber kein Gerät zum testen)

Mic und Line In an der Rückseite als Stereo Input

Das bedeutet aber: OSX kann sehr wohl mehr als 8 In/Outputs pro Codec

Beitrag von „atl“ vom 26. Juli 2020, 01:07

[e4535030](#) Vielen Dank! Ich werde es allerdings erst in den nächsten Tagen testen können. Sobald ich es getestet habe, gebe ich Feedback. 😊

Beitrag von „MacPeet“ vom 26. Juli 2020, 11:29

Euer Vorhaben ist ja sehr experimentell.

Generell ist die AppleHDA gar nicht für Hacki-Hardware gemacht und die Optionen, die wir mit AppleALC mitgeben sind oft nur begrenzt.

[e4535030](#)

Natürlich ist OSX bei der Anzahl der In-/Outputs nicht begrenzt, allerdings begrenzt Dich die Anzahl der möglichen [Audio Output]-Nodes und [Audio Input]-Nodes.

Bei Laptops sind dies oft nur Node 2 und 3, wenn SPDIF verbaut ist noch die 6 für 30-6. Hierbei müssen sich dann oft im SwitchMode die Output-Knoten teilen.

Bei Desktop-Rechnern sind es in der Regel ein paar Knoten mehr. Knoten teilen im ManualMode geht nicht mit OSX.

Das MidiSetup ist für eine Verbindung ja gedacht. Ob Du eine Verbindung auch ohne hinbekommst, musst Du versuchen.

Schau Dir dazu als Beispiel mal im SourceCode AppleALC Ordner ALC255 die Platforms20.xml an!

Diese ist für einen Laptop mit Subwoofer, veranschaulicht aber sehr gut, wie man dies verbindet und die Bind/Channel einstellt.

Hierbei sind Speaker und HP im SwitchMode konfiguriert und zeitgleich nochmal Speaker und Subwoofer innerhalb dessen als eine Einheit.

Hierbei hat der Subwoofer Bind 0 und Channel 4, also keine eigene Bindung und Kanal 4 für Mono rechts, da in Laptops die Subwoofer in der Regel immer auf dem rechten Kanal laufen und Bass ohnehin nur Mono ist.

Was in Windows alles geht, sollte ohnehin keine Rolle spielen, denn schließlich arbeiten hier explizite Treiber.

Bei der Konfiguration unter OSX sind wir halt oft eingeschränkt, was das Beispiel SPDIF auch zeigt.

Ein realMac gibt vom SPDIF/Optical zum Receiver feinstes 5.1 weiter. Die Konfigurationen für SPDIF in der AppleALC nur Stereo.

Dies geht nur auf dem Hacki, wenn man in AppleALC die Patches auf 00000000 auch aufs Device patcht. Dann geht SPDIF auch mit 5.1, bricht dann aber das Onboard-Audio.

Ich will damit nur aufzeigen, dass man nicht alles haben kann.

Es gibt ja auch noch z.B. 5.1/7.1 USB-Teile, welche OOB unter OSX arbeiten.

Beitrag von „atl“ vom 9. August 2020, 17:57

[e4535030](#) Vielen, vielen Dank und entschuldige, dass ich mich erst jetzt melde. Ich habe das noch kurz vor meinem Urlaub ausprobiert und es funktionierte tatsächlich so „halb“, wie du es gebaut hast. Ich habe mir das auch in der Platforms15.xml mal angeschaut.

Nach meinem Urlaub ist allerdings meine neue Hardware eingetrudelt und die verwendet den ALC1220A als Codec, zu dem ich kein Blockschaltbild gefunden habe. Aber mit Hilfe der Informationen aus diesem Thread habe ich die einzelnen Knoten anhand des Hdacodecdump unter Linux (Dank an [MacPeet](#)) ermitteln können. Und dank deiner Informationen habe ich die `AppleALC.kext` anpassen können, so dass **ALLES** soweit paßt:

- * Front-In und -Out arbeiten als unabhängige Ausgänge und
- * die rückseitigen Ausgänge arbeiten ebenfalls unabhängig

Zu Hilfe kam dabei, dass das Board einen rückseitigen Anschluß weniger (kein SIDE) hat und somit genügend DACs vorhanden sind. Interessanterweise hat diesmal das hinzufügen eines „Paths“ zur Pathmap problemlos funktioniert. Das komplizierteste war, herauszufinden, dass der ALC nicht **ALC1220** sondern **ALC1220A** ist. Das fehlende Mono nehme ich jetzt einfach so hin, weil es wirklich nur kosmetischer Natur ist.

Deshalb noch einmal vielen Dank an **alle**, die hier in diesem Thread aktiv waren!



Beitrag von „MacPeet“ vom 10. August 2020, 16:48

ja, sehr schön, aber Du brauchst auch kein Blockschaltbild. Ein `codec_dump` von Linux reicht völlig um die nötigen Infos zu lesen für einen neuen Codec.

Bei ALC1220 oder ALC1220A ist es nicht anders. Der Dump zeigt Dir die Device-ID vom Audio. Bei den bereits fertigen Codecs ALC1220 oder ALC1220A brauchst Du eigentlich nur schauen, welche ID passt und schon bist Du schlauer ob nun ALC1220 oder ALC1220A.

Beitrag von „tiloprost“ vom 7. September 2020, 00:54

Darf ich mich hier mit einklinken [MacPeet](#) und [atl](#) ? Ich habe einen NVisen Y-MU01 (chinesischer NUC) mit i5-8265U und der Audio Codec scheint der ALC269VC oder ALC269VB (10ec_0269) zu sein. AppleALC hat bei meinen Hackintoshes der Vergangenheit wunderbar funktioniert, aber hier kann ich alle möglichen layout-ids probieren und es passiert rein gar nicht. Der NUC hat übrigens von außen gesehen nur einen Kopfhöreranschluss - ob das ein kombinierter Ein-/Ausgang ist, kann ich nicht sagen. Online kann ich auch viel suchen und finde nicht wirklich was.

Ich hab mal unter Ubuntu den Dump erzeugt - Wie komme ich von hier an funktionierendes Audio? Und wenn AppleALC das Layout noch nicht kennt, wie kriegt man es dort rein?

Beitrag von „MacPeet“ vom 7. September 2020, 07:15

[tiloprost](#)

Dies ist die falsche Dump-Datei. Diese ist für's HDMI, gebraucht wird aber die vom Onboard-Audio.

Vermutlich fehlen bei Dir aber nur die IRQ-Fixes, denn ALC269 hat bereits sehr viele ID's zur Auswahl in der AppleALC.

Beitrag von „tiloprost“ vom 7. September 2020, 21:18

Hi,

ich hatte deinen Linux Terminal Code benutzt:

Code

1. `cd ~/Desktop && mkdir CodecDump && for c in /proc/asound/card*/codec#*; do f="{c/V*card/card}"; cat "$c" > CodecDump/{f/V/-}.txt; done && zip -r`

Wie kann ich den denn modifizieren um an das Onboard Audio zu kommen? Im Hackintosh -> PCIe sehe ich nirgendwo das Onboard Audio. Unter Windows gab es übrigens ebenfalls keine Tonausgabe (durchgestrichenes Symbol), unter Linux wurde Audio immerhin überhaupt angezeigt, aber ich habe es nicht getestet. VoodooHDA gibt mir nur HDMI Audio, das für mich wertlos ist.

Dass sich für den ALC269 viele IDs finden, glaube ich gern. Ich habe ja genug ausprobiert. Ich halte es dennoch für plausibel, dass das Layout des chinesischen NUCs noch nicht erfasst ist, denn wer soll es denn erfasst haben?

Was sind denn IRQ Fixes?

Ich würde jedenfalls gern das Ding zum Laufen bekommen. Was die Framebuffer Patches angeht, stehe ich aktuell auf halbem Wege. Wenn das Ding 100%ig lief würde ich gern meine Ergebnisse gern veröffentlichen und kann dann auch die Hardware empfehlen.

Beitrag von „MacPeet“ vom 8. September 2020, 10:13

IRQ-Fixes sind im CloverConfigurator FixRTC, FixIPIC, FixTIMR. Auch HPET sollte i.O. sein.

Für Windows fehlt sicher nur ein Audiotreiber.

Deine Bilder zeigen mir aber ein anderes Problem.

Du hast scheinbar einen dieser neuen Rechner der Intel 400 series mit LP-Controller. Dies ist an pci8086,9dc8 zu sehen, denn es ist ein Intel-Device, welches hierbei vor dem eigentlichen Realtek geschaltet ist und somit taucht das Realtek erst gar nicht auf.

Vor einiger Zeit hatte ich dieses Problem bereits mit [griven](#) in irgend einem anderen Thread. Hier ging der Dump via Linux auch nicht.

[griven](#) hatte damals berichtet, dass selbst unter Linux das Audio gefixt werden musste, wofür er eine Lösung im Netz gefunden hatte.

Ich kann aktuell nicht mehr sagen welcher Thread dies war.

Seit einigen Versionen der AppleALC arbeiten die Entwickler bereits an verschiedenen Controller-Fixes.

Erst gestern kamen die neuen Release's AppleALC/Lilu/WEG, wobei bei der AppleALC wieder ein neuer Controller dazu kam.

In wie weit Dein Controller schon dabei ist, vermag ich aktuell nicht zu sagen.

Im AppleALC-Thread auf insanelyMac, wenn man sich von hinten nach vorn durcharbeitet, findet man erste Lösungen für diese Art Controller. Hier arbeiteten einige User mit einer FakeID, um diese Intelbrücke zu überwinden.

Dort hat, ich glaube der User fewtarius darüber einiges geschrieben. Vielleicht kannst Du Dir dies mal anschauen.

Wenn diese Hürde überwunden wurde, dann ist das eigentliche Realtek in der Regel kein Problem mehr, denn viele Rechner mit ALC269VC verwenden die gleichen Knoten und bei ALC269VB ist es gleich, also je nach dem sollten dann einige ID zumindest Teilerfolge geben.

Beitrag von „tiloprost“ vom 9. September 2020, 20:02

Hi, er schreibt [hier](#) etwas, aber aus

Code

1. I've now found success using layout 21 and FakePCIID spoofing 0x8086 0x9DC8, everything lights up and sound works perfectly.

werde ich nicht recht schlau. Ich soll FakePCIID.kext und FakePCIID_Intel_HDMI_Audio.kext installieren? AppleALC deinstallieren? Im Clover Config die device-id für Audio rausnehmen oder drinlassen? Die IRQ fixes rausnehmen oder drinlassen?

EDIT: Hm, wenn ich die beiden FakePCIID Kexts reinwerfe, verändert sich immerhin der Eintrag im IORegExplorer.

Beitrag von „MacPeet“ vom 10. September 2020, 08:52

aktuelle AppleALC natürlich behalten und auch die IRQ-Fixes

Ich habe allerdings nie mit FakePCIID.kext gearbeitet, brauchte es nie.

Er verwendet ID21 und der User ein Post weiter die ID27 und dieser zeigt auch, wie die device-id in die Properties einzutragen ist.

Das ioreg sieht ja schon ganz gut aus. Ich denke Eure Rechner sind auch unterschiedlich.

Selbst wenn das Device gleich ist, die verschiedenen Hersteller verwenden oft auch unterschiedliche Audio-Knoten.

Wenn Du mit FakePCIID und fake device-id alles richtig hast, dann würde ich mal alle ID's nacheinander durchtesten.

Beitrag von „tiloprost“ vom 10. September 2020, 10:40

Die device-id so eintragen? Versteh ich nicht. Warum die device-id des Intel Audio eintragen welches ich gar nicht haben will? Mit der Layout ID des ALC audio?

Und verstehe ich das richtig: Ich brauche beides, die FakePCIID.kext und die FakePCIID_Intel_HDMI_Audio.kext?

Soll ich die device-id wirklich mit Raute als #C89D0000 eintragen?

Beitrag von „tiloprost“ vom 17. Oktober 2020, 21:48

Nachtrag: Ich hab das Ding jetzt eine Weile in Betrieb mit einem billigen USB-Audio [Interface](#), weil ich das onboard Audio auf Gedeih und Verderb nicht zum Laufen bekomme. Ich hab ein wenig mit Unity gearbeitet, das als Videoencode-Maschine genutzt, HEVC Playback und Benchmarks getestet ... Also eigentlich ist das Teil super. Kann ich nur empfehlen. Der Barebone kostet bei banggood 320€, mit Audio Interface, 500GB SSD, 16 GB RAM und ne

weitere HD vielleicht 500€. Kann man doch nicht meckern. Und das Ding ist halt klein genug um es im Rucksack mitzunehmen, dann kann ein Freelancer es an verschiedenen Orten nutzen.

Was ich mit verschiedenen Platform IDs und diversen Framebuffer Patch Experimenten ebenfalls nicht hinbekommen habe, ist die Nutzung beider Monitore gleichzeitig. Ich kann zwar beide in Betrieb nehmen, aber der zweite wacht nicht aus dem Ruhezustand auf. What a shame ...

Hat noch irgendwer Ideen zum Audio?

Beitrag von „atl“ vom 5. September 2021, 22:15

Ich bin's wieder. Mittlerweile habe ich schon mehrfach an meinem Layout herumgebastelt und bin sehr zufrieden. Ich habe es mittlerweile sogar in ein eigenes Layout für den Realtek ALC892 ausgelagert und dieses auf Github als [Pull Request](#) zur Aufnahme in die AppleALC.kext eingereicht. Dann muss ich hoffentlich nicht mehr ständig mergen und selber kompilieren.



Aktuell arbeite ich wieder an einem "neuen" Layout, da ich für den "Power Mac G4" eine neue Hardware habe. Das Board hat den Realtek ALC897 als Codec und an der ATX-Blende nur 3 Anschlüsse (Mic-In, Line-Out, Line-In). Den CodecDump habe ich ausgelesen und mittels diverser Tools und der Anleitung von [MacPeet in einem anderen Forum](#) visualisiert. Die Front-Anschlüsse habe ich schon - wie immer - von den Anschlüssen der Blende entkoppelt und den Kopfhörerausgang als "internal speaker" verfügbar gemacht. Lediglich eines hat nicht funktioniert, den hinteren Mic-In und den Line-In gleichzeitig verfügbar zu haben, so dass ich sie unabhängig als Audio-Quelle auswählen kann. Meine Idee war folgendes Routing:

Mic-In (24) -> 34 -> 9

Line-In (26) -> 35 -> 8



Allerdings scheint es nicht zu funktionieren. Vielleicht hat ja einer noch eine Idee? Vermutlich habe ich etwas übersehen oder es fehlt noch etwas. 😊

Beitrag von „MacPeet“ vom 6. September 2021, 18:55

ok, Dein card0-codec#0.**dec**.txt kann ich nicht mehr wandeln ohne Fehler mit codecgraph, wenn dann bräuchte ich die originale card0-codec#0.txt Datei.

Letztendlich sehe ich aber dennoch durch.

Zitat: Kopfhörerausgang als "internal speaker"

macht man so gar nicht im Normalfall, sondern man konfiguriert bei einem Desktop-Rechner den hinteren **grünen LineOut als interne Speaker** und Kopfhörer dann im SwitchMode an der Front.

Hinten am grün hängen normal die Boxen und wenn vorn Kopfhörer gesteckt werden, dann wird es unter macOS automatisch erkannt und umgeschaltet.

Beide Outputs haben EAPD, was konfiguriert werden muss.

Was Du bei den Inputs genau gleichzeitig verbinden willst, kann ich jetzt auch nicht genau feststellen.

Viele Rechner haben nur zwei Eingangs-Knoten, bzw. Knotenketten. Einige Rechner bieten sogar drei, in dem Fall noch die ...7.

In Deinem Fall 34-9 und 35-8.

Wenn man aber drei Eingänge hat ...

24 Mic hinten

25 Mic front

26 der tatsächliche LineIn, zumeist blauer Anschluss (ungeregelter Anschluss, bzw. die Eingabe bestimmt die Quelle)

Hierbei muss man bei nur zwei Eingangsknoten und drei Eingänge die Mic's auch im SwitchMode konfigurieren.

Hierbei setzt man Knoten 24 auf internal Mic und aus Knoten 25 macht man dann ein LineIn, statt Mic, beides im SwitchMode

Der Knoten 26 bleibt wie bei Dir dann eigenständig.

Meine Konfiguration würde dann so aussehen:

PinConfig:



ID	PinDefault	Device	Connector	Port	Group Location	Geometric Location	Color	S	P	EAPO
30 (9x14)	91114310	Out Speaker	1/8" Stereo/Mono	Fixed	Internal	Rear	Green	1	0	On2
34 (9x18)	90A09050	In Mic In	Unknown	Fixed	Internal	N/A	Pink	0	0	-
35 (9x18)	02B19050	In Line In	1/8" Stereo/Mono	Jack	External	Front	Pink	0	0	-
36 (9x18)	01B13070	In Line In	1/8" Stereo/Mono	Jack	External	Rear	Blue	7	0	-
37 (9x18)	02214080	Out HP Out	1/8" Stereo/Mono	Jack	External	Front	Green	0	0	On2

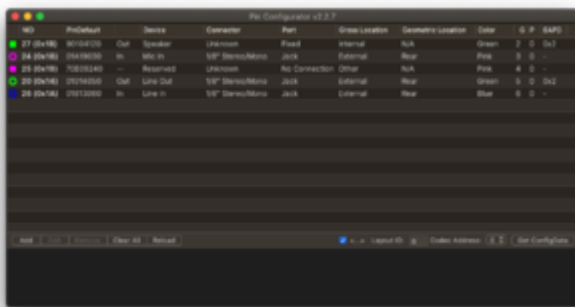
...und layout/Platforms dann so:

[layouts.zip](#)

Falls ich irgendetwas falsch verstanden haben sollte, dann melde Dich halt nochmal!

Beitrag von „atl“ vom 7. September 2021, 12:43

[MacPeet](#), anbei noch die originale `CodecDump_card0-codec#0.txt`. Danke für deine Anmerkungen. Tatsächlich sieht meine PinConfig wie folgt aus:



Den "25 Mic front" habe ich nicht konfiguriert, weil der nicht angeschlossen ist. Daher brauche ich den auch nicht im "SwitchMode" mit einem der hinteren Anschlüsse. Somit müsste ich die hinteren beiden Eingänge (24 Mic-In & 26 Line-In) unabhängig nutzen können, wenn ich aus deiner Pathmap einfach das Segment für den "25 Mic front" entferne. Richtig?



Der "27 Headphone front" ist dauerhaft an den internen Lautsprecher des Power Mac G4 Gehäuses angeschlossen. D.h. im SwitchMode hätte ich gar keine Möglichkeit den hinteren Line-Out zu nutzen. Aus diesem Grund habe ich den SwitchMode "deaktiviert" und kann dadurch beide Ausgänge unabhängig nutzen.

Die Umbenennung habe ich gemacht, da ja an dem Kopfhörerausgang (vorne) ein echter interner Lautsprecher angeschlossen ist. 😊

Beitrag von „MacPeet“ vom 7. September 2021, 16:23

ok, verstehe, aber der interne Lautsprecher des G4 ist ja nicht wirklich der Kracher, aber für Service-Töne reicht es wohl.

Allerdings hätte ich den Umbau sicher anders gemacht, diesen Lautsprecher mit dem hinteren LineOut verkabelt, so dass man die Frontanschlüsse frei hat, wenn man wirklich mal einen Kopfhörer stecken will.

Umgekehrt ist es ja schwierig, sich immer hinter den Rechner legen zu müssen, aber ok, es ist Deine Wahl.

Bezüglich der Mic's hatte ich nicht ganz verstanden, wo das Problem lag. Kam kein Signal bei den Inputs mit Deiner Konfiguration oder was war das Problem?

Wenn Du Knoten 25 (vorderes Mic) nicht brauchst, dann kannst Du die 25 aus der Pinconfig löschen und natürlich die Einträge aus meiner Platforms betreffs diesem Switchmode, was aber dann genau Deiner Konfiguration entspricht.

Ich hatte sonst nix verändert.

Wenn es Probleme mit den Mics, bzw. dem LineIn gibt, bezüglich der Funktion, dann muss man experimentieren.

Man könnte zuerst mal aus der layout diese üblichen Werte für MuteGPIO ganz raus löschen und schauen, was macOS daraus selbst regelt.

Manche Rechner brauchen auch schon mal MuteGPIO 0 bzw. 1 an mancher Position.

Beitrag von „atl“ vom 7. September 2021, 21:10

[MacPeet](#), es ist nur ein Eingang (Mic-In) in den Soundeinstellungen auswählbar und funktional. Beim Line-In tut sich nichts. Ich probiere dann mal rum. Danke für's draufschaun und die Tipps (MuteGPIO). 😊

Beitrag von „errut“ vom 9. Dezember 2021, 10:24

Hi, habe auch den ALC897 und bekomme ihn nicht zum Laufen. Vielleicht kann ich hier einen Hinweis bekommen.

Beitrag von „Hecatomb“ vom 9. Dezember 2021, 11:06

[errut](#) definiere bitte mal das nicht zum laufen bekommen genauer. Nur weil z.B. Die richtige Layout id nicht in der config gesetzt und gefunden wurde, heisst es nicht das es an einer fehlenden Layout in der kext Datei liegt.

Was genau funktioniert denn bei dir nicht?

Beitrag von „errut“ vom 9. Dezember 2021, 11:16

ok, habe Onboardgraphic des MOBO mit HD750 und ALC897, ferner eine NVIDIA GeForce 710 GT mit GK208 Chip. Als Display einen Philips 278E Bildschirm. Ich weiss, dass die Grafik mit der CPU I5-11600 nicht unterstützt wird. Benutze jeweils beide Grafikkarten über HDMI, weil ich immer noch herausfinden möchte, ob Sound möglich ist. Die NVIDIA hat keine separaten Tonausgänge, die Onboard die üblichen Analogausgänge, aber da kommt auch nichts raus. Die AlCIDs habe ich auch schon ausprobiert.

Beitrag von „MacPeet“ vom 9. Dezember 2021, 12:13

Unter DeviceProperties spoofst Du beim Onboard-Audio die DeviceID. Ist dies so richtig und gewollt? Kann ich nicht sagen, da ich diesen Rechner nicht kenne.

Für HDMI-Audio kannst Du auch mal AppleALCU versuchen, statt AppleALC, evtl. in der Reihenfolge etwas nach oben setzen.

Eigentlich sollte HDMI-Audio aber auch mit AppleALC gehen.

Evtl. braucht es noch für die Grafik in DeviceProperties einen hda-gfx Eintrag, aber eigentlich sollten die nativ laufen. Meine GT710 zumindest tat dies.

Die interne Grafik geht auf den Teilen ja wohl ohnehin nicht. daher würde ich die auch im Bios abschalten. Wenn die auch aktiv ist, dann versucht AppleALC evtl. die falsche Karte für HDMI-Audio zu injecten.

Beitrag von „errut“ vom 9. Dezember 2021, 12:20

danke. Ich weiss nicht was "DeviceProperties spoofst" bedeutet, also nichts bewusst gemacht. Wie kann ich das reparieren? Deine anderen Hinweise werde ich ausprobieren.

ok habe gegoogelt , die device-id war so drin, hatte ein ähnliches EFI von GITHub übernommen.

habe jetzt device-is 10DE0E0F aus den Hackintool entnommen und hoffe jetzt, dass das richtig ist. Habe AppleALCU nach vorne gebracht bringt aber nichts.

Beitrag von „MacPeet“ vom 10. Dezember 2021, 00:02

Hier geht einiges durcheinander. Erst einmal die Frage, es geht doch um dieses neue Brett Z-590M-Plus, oder?

Ferner die EFI von Github war explizit für genau diesen Z-590M-Plus???

Der Device-Pfad PciRoot(0x0)/Pci(0x1F,0x3) ist aber explizit nur das Onboard-Audio. Es bringt nichts, wenn man hier Devices der Grafikkarte einträgt für Hoffnung auf HDMI-Audio, denn dies hat mit Onboard-Audio nichts zu tun.

In der Regel braucht man die Device-ID überhaupt nicht eintragen, da AppleALC sie allein findet.

Die Device-ID für Onboard-Audio alc897 wäre ohnehin HEX 10EC0897, Dezimal 283904151 und mit Byte-Flip HEX 9708EC10

An dieser Stelle die Device-ID der Nvidia-Karte einzutragen kann ja nix bringen, da sie mit dem Onboard-Audio rein gar nix zu tun hat.

AppleALCU habe ich selbst nie gebraucht, soll ja für Digital-Audio sein. Möglich aber auch, dass dies nur den SPDIF-Bereich betrifft. Wechsel mal lieber wieder zur AppleALC, in der ja viele Grafikkarten-Controller für's HDMI-Audio eingetragen sind.

Diese Spoof-Geschichte ist im Prinzip so, dass man bei einigen ganz neuen Brettern, die mit macOS noch gar nicht unterstützt sind, auf das Device der Vorgängerversion spoofst, was oft ganz gut klappt.

Wird oft im IntelHD Grafik-Bereich gemacht bei den neueren CPU's, teils auch beim Audio.

Ich selbst habe diese Verfahrensweise nie gebraucht, da ich solche Rechner nicht habe. Ich meine aber dass man dann auch diese FakePCI...Kext's braucht. Bin aber nicht sicher diesbezüglich.

Vielleicht kann diese Sache mal ein User richtig erklären, der sich damit auskennt, bzw. einen dieser neueren Rechner sein Eigen nennt. Viele Nutzer hier machen diese Sache ja und können hoffentlich was dazu sagen.

Ferner hast Du in Deiner EFI/OC/Kexts den HDMIAudio.kext Version 1.0 liegen, welchen ich sogar noch etwas neuer habe in Version 1.1:

[HDMIAudio.kext.zip](#)

Diese Kext's sind eigentlich schon etwas älter, welche HDMI-Audio für AMD- und Nvidia-Grafikkarten injecten konnten, unabhängig von irgendwelchen anderen Lösungen.

Ich kann nicht sagen, ob die noch mit neuem macOS laufen, aber Du kannst den Kext ja testweise mal in der config unter Kernel/Add einbinden und schauen ob Du zumindest HDMI-Audio mit der GT710 bekommst.

Ich hatte früher den HDMIAudio.kext selbst erfolgreich in Verwendung, allerdings zu Zeiten, als es AppleALC noch gar nicht gab, läuft aber vielleicht noch heute.

Bei Änderungen an der config.plist, bzw. an der EFI setze ich aber einen Backup-Boot-Stick voraus, falls mal was nicht geht bei den Test's.

Zur weiteren Lösungsfindung hoffe ich, dass sich ein User mit dieser neueren Hardware hier einschaltet und vielleicht schon mehr Erkenntnisse dazu beisteuern kann.

Beitrag von „errut“ vom 10. Dezember 2021, 08:52

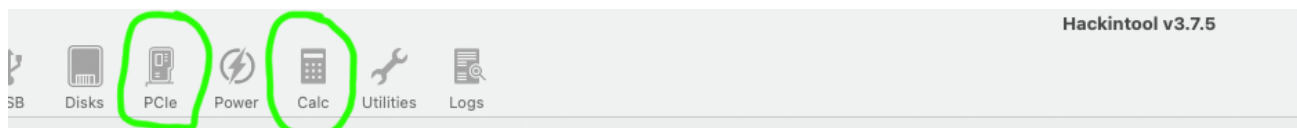
Vielen Dank, du hast mir viel mitgeteilt: Das Board ist korrekt (vor 2 Monaten gekauft - mit I5-11600K). Habe folgendes auf Github gefunden

<https://github.com/uranium81/hackintosh-intel-11th-gen>

Werde den Devicepfad für die Nvidia suchen und einbinden. Komme mit Dezimal und hex etc immer durcheinander. Im Hackintool

wird die NVIDIA angezeigt, wobei ich annehme, dass unter Device die Device-ID als HEXCode für die config.plist angezeigt wird.

Beitrag von „MacPeet“ vom 10. Dezember 2021, 10:31



Hackintool hilft Dir