

# AppleALC - der dynamische AppleHDA Patcher

Beitrag von „al6042“ vom 5. Mai 2016, 12:34

Hallo Leutchens...

sorry für die Abwesenheit, aber die privaten und geschäftlichen Aufgaben haben sich in den letzten Tagen meiner bemächtigt.

Ein wichtiger Punkt im Vorgang der Erweiterung des AppleALC-Kextes ist die Tatsache, dass der AudioCodec auf eine andere Art und Weise mit der AppleHDA schon mal zum Fliegen gebracht werden muss.

Wenn das funktioniert, wird mit einem dedizierten Script namens "Codec-Info.command" der aktuelle Zustand der eingesetzten Layouts, Platforms, Patches und ggf. PinConfigs in einer Info.plist auf dem Desktop abgelegt.

Ausgabe des Scripts am Beispiel meiner Config ALC898 und HDMI der GTX770 vom großen Rechner:

Spoiler anzeigen

Inhalt der erzeugten Info.plist vom Desktop:

Spoiler anzeigen

Der spannende Part ist das Einfügen in den Quellcode des bestehenden AppleALC...

Man lädt sich das komplette Package von <https://github.com/vit9696/AppleALC> herunter und prüft in den folgenden Verzeichnissen die Einträge:

In meinem Beispiel /Resources/ALC898... in dem Ordner befinden sich die folgenden Dateien:

Info.plist

layout1.xml.zlib

layout2.xml.zlib

layout3.xml.zlib

PlatformsT.xml.zlib (das "T" bezieht sich auf den Ursprung aus Toledas Patches)

In der info.plist werden zum einen die Kombinationen aus layout#.xml.zlib und Platforms#.xml.zlib dargestellt:

Spoiler anzeigen

und zum anderen die benötigten On-the-Fly KextPatches, so ähnlich wie man sie aus Clover kennt:

Spoiler anzeigen

Wenn man nun eigene layout#.xml.zlib, Platform#.xml.zlibs benötigt, wie im folgenden Beispiel meiner ALC269 vom Lenovo T530, dann kann man die als funktionierend bekannten zlibs in den Verzeichnissen mit eigenen Namen einbinden:

Spoiler anzeigen

Innerhalb der Info.plist im Ordner /Resources/ALC269 gebe ich dann diese neuen Dateien zur Nutzung bekannt:

Spoiler anzeigen

Sollte das Ergebnis des oben erwähnten Command-Scripts noch zusätzliche Patches feststellen, die noch nicht in der Info.plist des Resource-Ordners bekannt sind, könnt ihr diese einfach mit einbinden.

Im Ergebnis des Command-Scripts kann auch einen Teil namens "PinConfigurations" ausgibt. Diese werden in der Info.plist des Kextes /Resources/PinConfigs.kext eingefügt. Zuerst prüft man natürlich, ob es einen Eintrag für die vorliegende CodecID schon gibt und ob darin nicht der gleiche ConfigData-String hinterlegt ist. Hier wieder ein anders Beispiel für einen ALC293 Codec:

Spoiler anzeigen

Achtet darauf, dass die CodecID als Dezimal Wert eingetragen werden muss.

Danach könnt ihr das XCode-Projekt in XCode laden und eine eigene Version des AppleALC-Kexts erstellen.

Was das Auslesen dieser Infos durch eine Linux-Distribution betrifft, bin ich selbst bisher nurauf die bekannten Geschichten wie hier [http://www.insanelymac.com/for...-applehda-for-your-codec/](http://www.insanelymac.com/forum/thread/26776-applealc-der-dynamische-applehda-patcher/?postID=231298#post231298) gestoßen.