

Erledigt

GA-Z87-UD5H mit Ozmosis und angepasster DSDT

Beitrag von „thomaso66“ vom 22. November 2013, 19:12

Dank dem Ozmosis Bootloaders sind wir ja schon recht nahe mit unserem Hakintosh an einen Original MAC vorgedrungen, einiges last sich ja an Kexten noch im BIOS mit unterbringen, nur beim Audio bedarf es immer noch einer gepatschten appleHDA das und noch einiges mehr möchte ich nun mit Hilfe einer DSDT umgehen, die dann schlussendlich im BIOS gegen die Originale ausgetauscht wird zu Integrieren.

Dies ist nun mal ein Versuch meine herangehensweise etwas zu Dokumentieren vielleicht hat ja der an oder andere geneigte Leser Anregungen, Verbesserungen, oder gar findet ja selbst gefallen daran für sein Board sich seine DSDT zu erstellen.

Aktuell setze ich ein mod Bios für mein Z87X-UD5H in Version F8.a [quelle](#) dieses wurde gegen über dem Originalen F8a in folgenden Punkten verändert.

Original F7/F8a enthält:

Intel® Boot Agent GE v1.5.04 PXE 2.1 Build 092 (WfM 2.0)
Intel® PCI Accelerated SVGA BIOS 2171 PC 14.34 04/08/2013
Intel® GOP Driver 5.0.1032
Marvell 88SE91xx Adapter - BIOS Version 1.0.0.0027
Intel® RAID for SATA - v12.6.0.1867
EFI RAID SataDriver - v12.6.0.1867
ME Firmware 9.0.3.1347
Haswell CPU-Patch 9

mod enthält F8a

Intel® Boot Agent GE v1.5.43 PXE 2.1 Build 092 (WfM 2.0)
Intel® PCI Accelerated SVGA BIOS 2177 PC 14.34 07/25/2013
Intel® GOP Driver 5.0.1035
Intel® RAID for SATA - v12.7.0.1936
EFI RAID SataDriver - v12.7.0.1936
ME Firmware 9.0.21.1462
Haswell CPU-Patch 16

Das habe ich um den [Ozmosis Bootloader](#) erweitert, und beinhaltet des weiteren noch

LAN,fakeSMC-v5.3.820,GPU,CPU,LPC-Sensoren für HWSensor, HDAEnebler, sowie der Spalshsreen (Apple) wurden angepasst.

Damit ist dann schon mal eine Vanilla Installation nach Herstellung eines Installsticks via [Disk Maker X](#) möglich.

Das vor mir eingesetzt BIOS [Z87X-UD5H.mod.F8a-Ozmosis](#)

Daraus habe ich mir dann nach dieser [Anleitung](#) von apfelnico die [AmiBoardInfo.fff.zip](#) und daraus die [dsdt.aml.zip](#) extrahiert.

Mit MaciACL geöffnet, in den Preferences ACPI 4.0 eingestellt (DSDTEse und DSDT Editor arbeiten auch so) und den erstmal auf Compile geklickt um zu sehen was an Fehlern und Warnungen so bei rauskommt.

Compiler Summary		
Line	Code	Message
3262	1105	Possible operator timeout is ignored
4291	5111	Use of compiler reserved name (_T_1)
4292	5111	Use of compiler reserved name (_T_0)
4548	5111	Use of compiler reserved name (_T_1)
4549	5111	Use of compiler reserved name (_T_0)
4805	5111	Use of compiler reserved name (_T_1)
4806	5111	Use of compiler reserved name (_T_0)
5062	5111	Use of compiler reserved name (_T_1)
5063	5111	Use of compiler reserved name (_T_0)
5319	5111	Use of compiler reserved name (_T_1)
5320	5111	Use of compiler reserved name (_T_0)
6352	5111	Use of compiler reserved name (_T_1)
6353	5111	Use of compiler reserved name (_T_0)
6609	5111	Use of compiler reserved name (_T_1)
6610	5111	Use of compiler reserved name (_T_0)
6866	5111	Use of compiler reserved name (_T_1)
6867	5111	Use of compiler reserved name (_T_0)
7338	1088	Not all control paths return a value (RDGI)
7355	1088	Not all control paths return a value (RDGP)
7742	5111	Use of compiler reserved name (_T_0)
7830	5111	Use of compiler reserved name (_T_0)
7918	5111	Use of compiler reserved name (_T_0)
8006	5111	Use of compiler reserved name (_T_0)
8228	5111	Use of compiler reserved name (_T_0)
8316	5111	Use of compiler reserved name (_T_0)
8471	1099	Unknown reserved name (_DEP)
8471	1099	Unknown reserved name (_DEP)
8471	1099	Unknown reserved name (_DEP)
8485	5072	Effective AML package length is zero
9732	1099	Unknown reserved name (_DEP)
9732	1099	Unknown reserved name (_DEP)
9732	1099	Unknown reserved name (_DEP)
9746	5072	Effective AML package length is zero
10318	5111	Use of compiler reserved name (_T_0)
10773	1100	Syntax error, unexpected PARSEOP_ARG0
10773	1100	Statement is unreachable

Da das ganze ja neu für mich ist, hier mal mein Lösungsvorschläge, wenn da jemand etwas besser weis als ich dann Immer her damit.

Fehler Behebung:

Warning 1105 - Possible operator timeout is ignored

Code

1. Acquire (MUT0, 0x0FFF)
- 2.
- 3.
4. ändern zu
- 5.
- 6.
7. Acquire (MUT0, 0xFFFF)

Remark 5111 - Use of compiler reserved name

Code

1. Suchen und ersetzen von `_T_` mit `T_`

Remark 5072 - Effective AML package length is zero

Code

1. Return (Package (0x00) {})
- 2.
- 3.
4. ändern zu
- 5.
- 6.
7. Return (Package (Zero) {})

Ist ja eigentlich das selbe nur stört sich der Compiler nicht mehr daran.

Warning 1088 - Not all control paths return a value

Code

```
1. Method (RDGI, 1, NotSerialized)
2. {
3. If (LLessEqual (Arg0, 0x5E))
4. {
5. Store (Add (Add (GPBS, 0x0100), Multiply (Arg0, 0x08)
6. ), Local0)
7. OperationRegion (LGPI, SystemIO, Local0, 0x04)
8. Field (LGPI, AnyAcc, NoLock, Preserve)
9. {
10. , 30,
11. TEMP, 1
12. }
13.
14.
15. Return (TEMP)
16. }
17. }
18.
19.
20. ändern in
21.
22.
23. Method (RDGI, 1, NotSerialized)
24. {
25. If (LLessEqual (Arg0, 0x5E))
26. {
27. Store (Add (Add (GPBS, 0x0100), Multiply (Arg0, 0x08)
28. ), Local0)
29. OperationRegion (LGPI, SystemIO, Local0, 0x04)
30. Field (LGPI, AnyAcc, NoLock, Preserve)
31. {
32. , 30,
33. TEMP, 1
34. }
35.
36.
37. Return (TEMP)
38. }
39. Return (Zero)
40. }
```

Alles anzeigen

Warning 1099 - Unknown reserved name (_DEP)

Code

1. _DEP
2. ändern zu
3. DEP

Error 4096 - syntax error, unexpected PARSEOP_ARG0

Code

1. native:
2. Method (ADBG, 1, Serialized)
3. {
4. If (CondRefOf (MDBG))
5. {
6. Return (MDBG)
7. Arg0
8. }
- 9.
- 10.
- 11.
- 12.
13. Return (Zero)
14. }
- 15.
- 16.
- 17.
- 18.
19. patched:
20. Method (ADBG, 1, Serialized)
21. {
22. If (CondRefOf (MDBG))
23. {
24. Store (Arg0, MDBG)
25. Return
26. }
27. Return (Zero)
28. }

Alles anzeigen

Die Änderungen führen dann zu einer DSDT ist 0 Fehler 0 Warnungen enthält, da ich die Anpassungen die ich nun vorgenommen habe erst noch alle austesten muss, werde ich diesen Posting nach und nach erweitern, vielleicht hilft es dem ein oder anderen der sich mit der DSDT Bearbeitung nicht auskennt ja etwas, zumindest sind hier ein paar immer wieder vorkommende Fehler und deren Lösungsweg beschrieben.

So da ich meine dstd auch gerne nach ACPI 5.0 zumindest Fehler frei haben wollte habe ich mich auf die suche gemacht und bin dann bei toleda auf Github fündig geworden.

Also erweitern wir erstmal MaciASL

MaciASL -> Preferences -> Sources -> + als Namen habe ich der Einfachheit halber toleda hdmi eingetragen bei der url: https://raw.githubusercontent.com/toleda/audio_hdmi_uefi/master

Preferences -> iASL ACPI 5.0 einstellen

Im Hauptfenster -> Compile und wir sehen es sind doch noch so einige Fehler vorhanden, diese merzen wir nun aus 😊

Patch -> unter toleda hdmi sehen wir nun den gesuchten UEFI-Clean Compile patsch auswählen, ausführen Fenster wieder schließen gefolgt von einem Compile und wir sehen das die Fehler auch unter ACPI 5.0 weg sind.

Wer sich den Patsch an gesehn hat wird feststellen das einige der oben genannten Fehler mit diesem Automatisch behoben werden.

Nicht vergessen die Preferences wieder auf ACPI 4.0 zu stellen.

Thema Sound 😊 ich habe ja bei meinem Board einen ALC898 Chip verbaut aber er ist eigentlich für alle geeignet die bis jetzt mit HDAEnebler Kext gearbeitet haben.

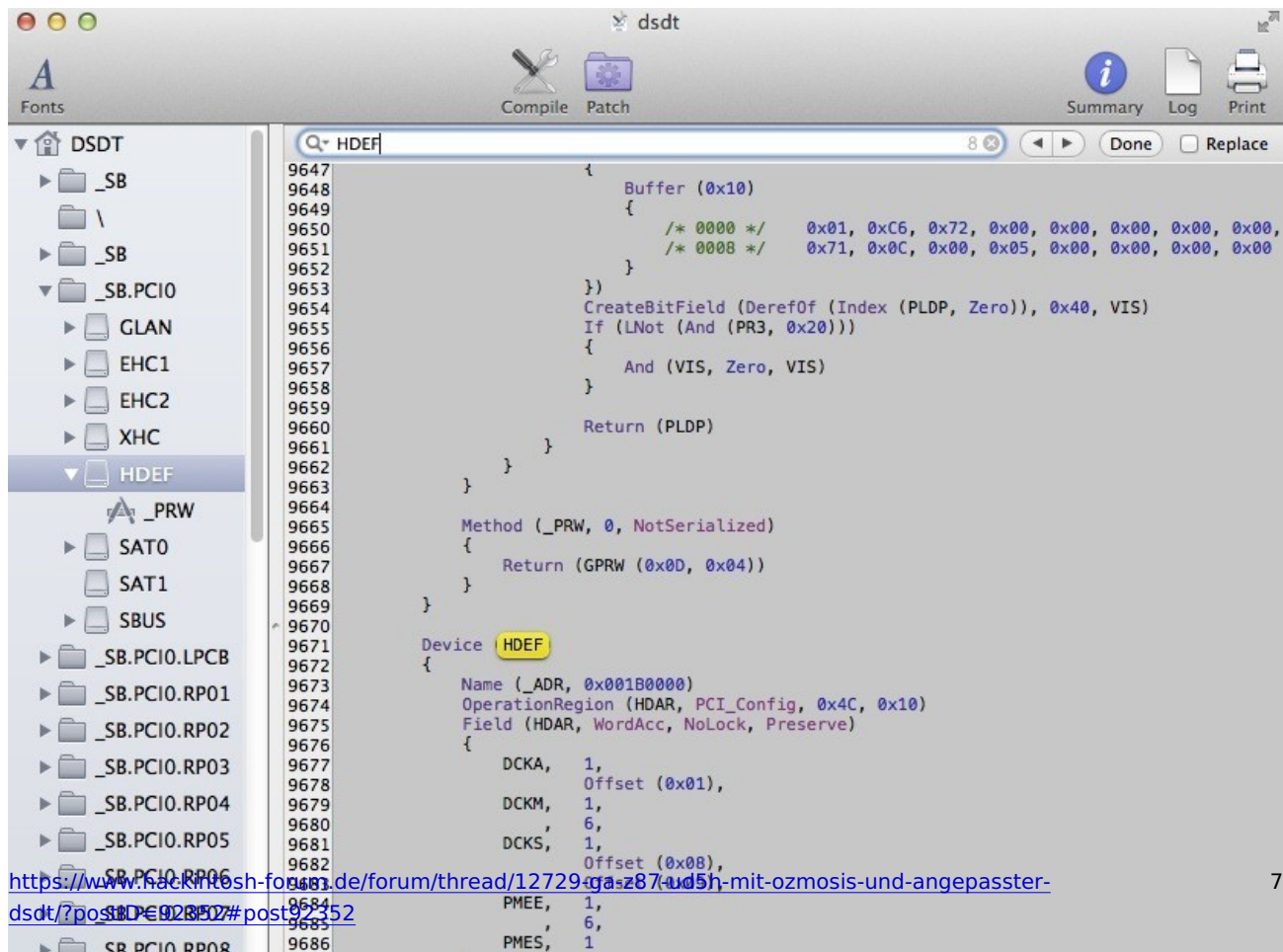
hier der [Link](#) zur Vollständigen Erklärung (Englisch)

Und nun meine Zusammenfassung des ganzen.

Wir erweitern MaciASL genauso wie schon oben beschrieben um die URL: https://raw.githubusercontent.com/toleda/audio_ALCInjection/master damit wird der notwendige Patsch automatisch bereitgestellt.

So nun schauen wir erstmal nach ob wir das Device HDEF schon haben oder nicht, danach entscheidet sich welcher Patsch ausgewählt werden muss.

Entweder per MaciASL->Edit->Find->Find HDEF oder cmd+f HDEF

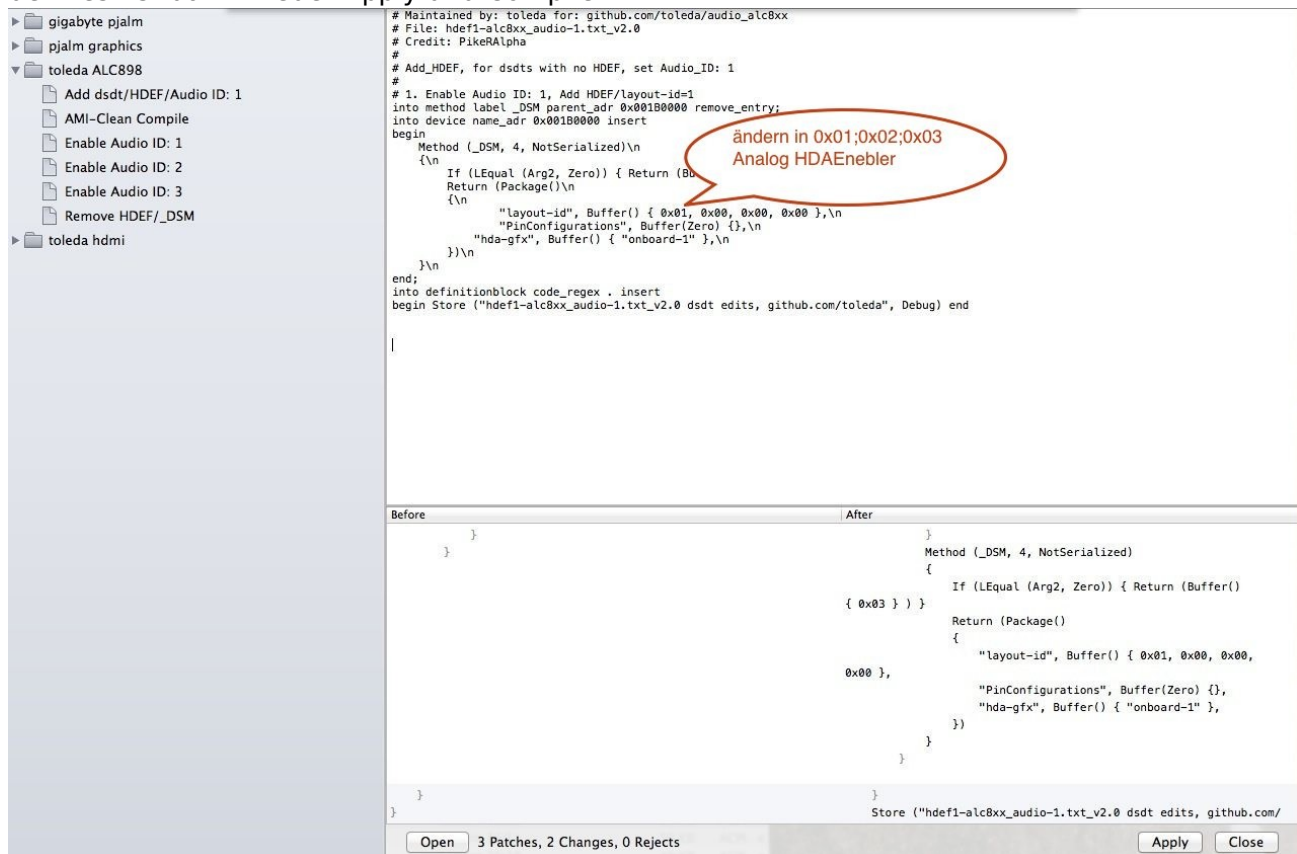


```
9647         {
9648             Buffer (0x10)
9649             {
9650                 /* 0000 */ 0x01, 0xC6, 0x72, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
9651                 /* 0008 */ 0x71, 0x0C, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00
9652             }
9653         })
9654         CreateBitField (DerefOf (Index (PLDP, Zero)), 0x40, VIS)
9655         If (LNot (And (PR3, 0x20)))
9656         {
9657             And (VIS, Zero, VIS)
9658         }
9659         Return (PLDP)
9660     }
9661 }
9662 }
9663 }
9664 }
9665 Method (_PRW, 0, NotSerialized)
9666 {
9667     Return (GPRW (0x0D, 0x04))
9668 }
9669 }
9670 Device (HDEF)
9671 {
9672     Name (_ADR, 0x001B0000)
9673     OperationRegion (HDAR, PCI_Config, 0x4C, 0x10)
9674     Field (HDAR, WordAcc, NoLock, Preserve)
9675     {
9676         DCKA, 1,
9677         Offset (0x01),
9678         DCKM, 1,
9679         , 6,
9680         DCKS, 1,
9681         Offset (0x08),
9682         PMEE, 1,
9683         , 6,
9684         PMES, 1
9685     }
9686 }
```

Ist das Device HDEF wie bei mir schon vorhanden brauchen wir unter Patch nur noch Enable Audio ID:1;2 oder 3 auswählen (1 und 2 entspricht den bisher bekannten HDAEneblern 1 und 2) den patch via Apply ausführen und Compile das wars.

Wenn HDEF **nicht vorhanden** ist, muss der Patch Add dsdt/HDEF/Audio ID:1 genommen werden.

Im Oberen Fenster kann die Anpassung der Layout ID vorgenommen werden also 1;2;3 sie Bild der Rest ist dann wieder Apply und Compile



So das war es zum Thema Audio Integration in die DSdT

Weiter geht es in Posting 31