

Erledigt **DSDT Connector-Patch**

Beitrag von „modzilla“ vom 3. November 2017, 16:19

Hey,

[@kuckkuck](#) und ich wollten ganz gern mit der DSDT den Connector patchen, haben uns dazu auch bereits die SSDT von mit angeschaut, hierbei ist uns dieser Ausschnitt aufgefallen:

Code

1. // Only use this if you need special connectors that are incompatible with the automatic connector detection
2. "connectors",
3. Buffer ()
4. {
5. 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x12, 0x04, 0x04, 0x01,
6. 0x00, 0x08, 0x00, 0x00, 0x04, 0x02, 0x00, 0x00, 0x00, 0x01, 0x02, 0x00, 0x22, 0x05, 0x01, 0x03,
7. 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00, 0x00, 0x01, 0x03, 0x00, 0x10, 0x00, 0x05, 0x06,
8. 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00, 0x00, 0x01, 0x04, 0x00, 0x11, 0x02, 0x06, 0x05
9. },

Deshalb wollte ich jetzt mal einen Thread eröffnen, um das jetzt gemeinsam anzugehen 😊

Dabei ist mir schon aufgefallen, dass sicher der Code/Patch kompiliert immer in 8er Blöcke aufteilt!

Code

1. "connectors",
2. Buffer (0x40)
3. {
4. /* 0000 */ 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00,
5. /* 0008 */ 0x00, 0x01, 0x01, 0x01, 0x12, 0x04, 0x04, 0x01,
6. /* 0010 */ 0x00, 0x08, 0x00, 0x00, 0x04, 0x02, 0x00, 0x00,
7. /* 0018 */ 0x00, 0x01, 0x02, 0x00, 0x22, 0x05, 0x01, 0x03,
8. /* 0020 */ 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
9. /* 0028 */ 0x00, 0x01, 0x03, 0x00, 0x10, 0x00, 0x05, 0x06,
10. /* 0030 */ 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
11. /* 0038 */ 0x00, 0x01, 0x04, 0x00, 0x11, 0x02, 0x06, 0x05
12. },

Alles anzeigen

Mein Patch oder verwechsel ich hier grad was? :

Code

1. Chutoro:
2. 00040000040300000001000000000001204000100000000
3. 00040000040300000001000000000002205000200000000
4. 00080000040200000071000000000001102000300000000
5. 040000001402000000010000000000000000000600000000
6. 00020000140200000001000000000002103000500000000

Ich hab grad ne Idee... Vllt kann man ja die letzten 8en und 8 aus der Mitte entfernen, so käme man ja auch auf 16:

Code

1. Chutoro:
2. 00040000040300000001000012040001
3. 00040000040300000001000022050002

4. 00080000040200000071000011020003
5. 04000000140200000001000000000006
6. 00020000140200000001000021030005

Ich glaube wir kommen dem Ende näher:

Code

1. 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x12, 0x04, 0x00, 0x01
2. 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x22, 0x05, 0x00, 0x02
3. 0x00, 0x08, 0x00, 0x00, 0x04, 0x02, 0x00, 0x00, 0x00, 0x71, 0x00, 0x00, 0x11, 0x02, 0x00, 0x03
4. 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06
5. 0x00, 0x02, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x21, 0x03, 0x00, 0x05

Ich glaube das sieht schon mal ganz gut aus 😊 :

Code

1. "connectors",
2. Buffer (0x50)
3. {
4. /* 0000 */ 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00,
5. /* 0008 */ 0x00, 0x01, 0x00, 0x00, 0x12, 0x04, 0x00, 0x01,
6. /* 0010 */ 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00,
7. /* 0018 */ 0x00, 0x01, 0x00, 0x00, 0x22, 0x05, 0x00, 0x02,
8. /* 0020 */ 0x00, 0x08, 0x00, 0x00, 0x04, 0x02, 0x00, 0x00,
9. /* 0028 */ 0x00, 0x71, 0x00, 0x00, 0x11, 0x02, 0x00, 0x03,
10. /* 0030 */ 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
11. /* 0038 */ 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06,
12. /* 0040 */ 0x00, 0x02, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
13. /* 0048 */ 0x00, 0x01, 0x00, 0x00, 0x21, 0x03, 0x00, 0x05
14. },

Alles anzeigen

Tja so einfach ist es dann leider doch nicht, weiß jmd weiter? Wenn ich das so in meiner DSDT hinzufüge, laufe ich in einen Blackscreen und kann nur noch mit VNC an den Rechner. Grafikbeschleunigung funktioniert dabei leider auch nicht...

UPDATE: Habe ma einen IOREG-Auszug angehängt!

Beitrag von „Harper Lewis“ vom 3. November 2017, 16:19

Soll das denn den selben Effekt bewirken, wie ein Kext-Patch für den AMDXXXXController.kext, oder ist das eine spezielles Viech für WhateverGreen?

Und "deinen Patch" hast du mit dem AMD Framebuffer Utility erstellt?

Die FAQ dazu sind lustig, die versteht glaube ich niemand außer vit.

Beitrag von „kuckkuck“ vom 3. November 2017, 16:41

Ich hab du gerade glaube ich des Rätsels Lösung bei den Verrückten gefunden! 🤪

Schau mal hier: <http://www.insanelymac.com/for...b-clover-injection/page-1>

Das ganze interessiert mich so, weil ich bei meiner Karte nicht beid DVI Ports gleichzeitig benutzen kann... Mit einem DSDT Patch lässt sich der default Framebuffer benutzen, aber gleichzeitig die Anzahl der Ports definieren:

Code

1. "connector-count",
2. Buffer ()
3. {
4. 0x04, 0x00, 0x00, 0x00
5. },

Und daraufhin noch sagen welcher Connector welche Art von Port ist. Dazu muss also die Hex Folge, wie in obigem Link beschrieben, erstellt werden und in den Eintrag "connectors" gesetzt werden.

Weiter im Code:

Code

1. "AAPL,slot-name",
2. Buffer ()
3. {
4. "Slot-1"
5. },
- 6.
- 7.
8. // Write the main main monitor index (currently 0) here, the value does not matter
9. "@0,AAPL,boot-display",
10. Buffer (Zero) {}
- 11.
- 12.

```
13. /*
14.
15.
16. // Only use this if your GPU device identifier is not natively supported in macOS kexts
17. "device-id",
18. Buffer (0x04)
19. {
20. 0x12, 0x34, 0x00, 0x00
21. },
22.
23.
24. // Only use this if automatic detection fails or you are faking your GPU device-id
25. "model",
26. Buffer ()
27. {
28. "AMD Radeon HD 6450"
29. },
30.
31.
32. // Only use this if you are not satisfied with automatic HDMI injection
33. // Make sure not to write the same onboard index for each installed GPU
34. "hda-gfx",
35. Buffer ()
36. {
37. "onboard-1"
38. },
```

Alles anzeigen

"AAPL,slot-name": Slot-1 entspricht dem ersten PCI Steckplatz für eine GPU von oben gezählt, Slot-2 wäre der zweite, sehe ich das richtig? Nur nötig wenn mehrere GPUs installiert?

"@0,AAPL,boot-display": Coole Sache, aber woher weiß ich welches Display welche Nummer hat?

"device-id": Aus DPCIManager mit ByteSwap

"model": Steht dat dann auch unter "Über diesen Mac"? 😬

"hda-gfx": Wenn die iGPU aktiv ist, dann wird eine der beiden zu "onboard-2"... Spielt es wohl eine Rolle ob iGPU oder GFX0 "onboard-2" bzw. 1 ist?

Beitrag von „Harper Lewis“ vom 3. November 2017, 16:50

Das klingt interessant. Der Framebuffer ATY,AMDRadeonFramebuffer wird ja auch mit Whatevergreen genutzt. "connector-count" in der SSDT ohne Framebuffer (bei mir Hamachi oder Futomaki) werde ich später mal antesten...

Beitrag von „modzilla“ vom 3. November 2017, 17:10

1. Ja der Slot Buffer gibt nur an wo das Gerät liegt, kannst also auch wenn das deine einzige Karte ist, Slot-3 nehmen oder so.

2. Das hab ich auch noch nicht hundertprozentig durchblickt... Aber ich glaube das kannst du mit der priority festlegen. Dann muss der erste aus dieser Liste dann 0 sein und somit das boot-display

3. Ja tut es 😊

4. Kann ich leider nicht sagen, da mein Xeon keine iGPU hat 😬 Aber ich nutze dennoch onboard-2 und dat läuft wunderbar.

[@kuckkuck](#) ja genau das möchte ich auch, aber irgendwie würde ich das mit dem FB

ConnectorPatch hinbekommen 🤔😡

[@Harper Lewis](#) Ja richtig, der Einfachkeithalber hab ich das nette kleine Tool genutzt: 😄 Aber mein Ziel ist es primär auf das manuelle patchen des AMD7000Controllers zu verzichten (nutze ja OZM)! Daher versuche ich möglichst alle Parameter mit der DSDT zu injecten... und der WEG Kext ist ja nichteinmal mehr nötig, denn Mieze hat passend zum Thema ein DSDT Patch entwickelt, der das gleiche tut!

Beitrag von „kuckkuck“ vom 3. November 2017, 17:51

Check ich nicht, hast du die verlinkte Anleitung gelesen? Das kommt doch einfach nach GFX0 in die DSDT, da musst du doch nichts am Controller patchen...

Beitrag von „Harper Lewis“ vom 3. November 2017, 18:20

Funktioniert das denn bei dir, [@kuckkuck](#)? Wenn ich in meiner SSDT die vier Einträge für den Framebuffer Hamachi weglassen, wird der Default-Framebuffer geladen. Dafür wird der Geräte name "Sapphire R9 280 Dual-X" nicht aus der SSDT übernommen. Die Connector-Types sind aber identisch: DP, DP, DDVI, HDMI. Jetzt ist die Frage ob es irgendwie zielführend ist, das auf die Ausgänge der R9 umzubiegen: DP, HDMI, DDVI, DDVI

Beitrag von „kuckkuck“ vom 3. November 2017, 19:09

Ich hab's noch nicht probiert...

Wird der Name denn bei Hamachi übernommen?

Default ist doch Futomaki, dann trag doch in die SSDT mal Futomaki ein und dann aber mit Custom Namen etc. und schau obs funzt 😊

Beitrag von „Harper Lewis“ vom 3. November 2017, 19:19

Hamachi und Futomaki funktionieren beide und dann wird auch der in der SSDT eingetragene Gerätenamen ("model") angezeigt. Ich habe jetzt mal folgende connectors eingetragen (und wieder die Framebuffer weggelassen):

Code

```
1. "connectors",
2. Buffer (0x40)
3. {
4. /* 0000 */ 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00,
5. /* 0008 */ 0x00, 0x01, 0x01, 0x01, 0x12, 0x04, 0x02, 0x01,
6. /* 0010 */ 0x00, 0x08, 0x00, 0x00, 0x04, 0x02, 0x00, 0x00,
7. /* 0018 */ 0x00, 0x01, 0x02, 0x00, 0x22, 0x05, 0x03, 0x03,
8. /* 0020 */ 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
9. /* 0028 */ 0x00, 0x01, 0x03, 0x00, 0x01, 0x02, 0x01, 0x04,
10. /* 0030 */ 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
11. /* 0038 */ 0x00, 0x01, 0x04, 0x00, 0x00, 0x00, 0x06, 0x06
12. }
```

Alles anzeigen

Ergebnis:

wie gehabt model: AMD Radeon HD 7xxx, Framebuffer: ATY,AMD,RadeonFramebuffer

Dafür hat sich aber der connector-type bei den vier Anschlüssen geändert:

0x400, 0x800, 0x4, 0x4 = DP, HDMI, DVI, DVI (vermute ich, passt aber auch zu den Ausgängen der R9)

Mit Hamachi oder Futomaki:

0x400, 0x400, 0x4, 0x800 = DP, DP, DDVI, HDMI (vermute ich)

Unter 10.12.6 hat eh alles funktioniert. Egal ob ich WhateverGreen oder wie jetzt aktuell den Patch von Mieke genutzt habe. Unter 10.13 sieht es da wieder ganz anders aus.

Beitrag von „Mork vom Ork“ vom 3. November 2017, 19:30

Warum versucht ihr einen Connectorpatch über die DSDT zu lösen und nicht via "KextsToPatch" ala:

Code

1. <key>KextsToPatch</key>
2. <array>
3. <dict>
4. <key>Comment</key>
5. <string>Connectors Patch for Framebuffer ORINOCO for RX480/580 NITRO</string>
6. <key>Disabled</key>
7. <false/>
8. <key>Find</key>
9. <data>AAQAAQDAAAAQEBAABIEBgEAAAAAAQAAQDAAAAQIBAAAAACIFBAMAAAAAAgAAAC
10. <key>Name</key>
11. <string>AMD9500Controller</string>
12. <key>Replace</key>
13. <data>AAQAAQDAAAAQEBAABIEBgEAAAAAAQAAQDAAAAQIBAAAAACIFBAMAAAAAAgAAAC
14. </dict>
15. <dict>
16. <key>Comment</key>
17. <string>Connectors Patch for Framebuffer ACRE for RX460 NITRO</string>
18. <key>Disabled</key>
19. <false/>
20. <key>Find</key>
21. <data>AAQAAQDAAAAQEBAABIEBgEAAAAAAgAAQCAAAAAQIAAAAAACEDBQQAAAAABAAAA
22. <key>Name</key>

- 23. <string>AMD9500Controller</string>
- 24. <key>Replace</key>
- 25. <data>AAQAAAQDAAAAAQABAAAAABECBQEAAAAAAAgAAAQCAAAAAQABAAAAACEDAwQAAAAABAAAA
- 26. </dict>

Alles anzeigen

was dann dem hier entspräche:

KeystitchPatch	Array	(5 Items)
Item 0	Dictionary	(5 Items)
Comment	String	Connectors Patch for framebuffer ORNOCO for RX480/SB0 NTRG
Disabled	Boolean	NO
Find	Data	<004A0000 04030000 00010101 00000000 12040801 00000000 00040000 04030000 00010201 00000000 2290403 00000000 00080000 04020000 00010300 00000000 11020102 00000000 00080000 04020000 00010400 00000000 21030504 00000000 04000000 04000000 00010500 00000000 00000308 00000000>
Name	String	AMD9500Controller
Replace	Data	<004A0000 04030000 00010101 00000000 12040801 00000000 00040000 04030000 00010201 00000000 2290403 00000000 00080000 04020000 00010301 00000000 11020102 00000000 00080000 04020000 00010400 00000000 21030504 00000000 04000000 04000000 00010500 00000000 10000308 00000000>
Item 1	Dictionary	(5 Items)
Comment	String	Connectors Patch for framebuffer ACHE for RX480 NTRG
Disabled	Boolean	NO
Find	Data	<004A0000 04030000 00010101 00000000 11020201 00000000 00080000 04020000 00010200 00000000 21030504 00000000 04000000 04020000 00010300 00000000 00000308 00000000>
Name	String	AMD9500Controller
Replace	Data	<004A0000 04030000 00010001 00000000 11020501 00000000 00080000 04020000 00010001 00000000 21030304 00000000 04000000 04020000 00010000 00000000 10000408 00000000>

Beitrag von „modzilla“ vom 3. November 2017, 19:59

Ja das wäre ja was für Anfänge. 😊 , nein Spaß... Ich würd es gern so machen, da ich nicht nach jedem Update den Controller mit nem perl Befehl patchen möchte und ich gleichzeitig nicht genügend Platz im ROM habe für den KernextPatcher!

Und außerdem ist meine Interesse geweckt XD

@Harper Lewis ich vermute, dass bei dir der Name nicht übernommen wird, liegt daran, dass die Buffergröße nicht mehr stimmt und er es deshalb verkackt den zu injecten. Deshalb: Entfernen einfach immer nach ner Änderung die Zahl in der Klammer entfernen! Dann wird die Größe neu berechnet

Beitrag von „Harper Lewis“ vom 4. November 2017, 13:07

Ist doch nur Rumspielerei. 😊 Wobei das, was ich da oben geschrieben habe, Unsinn ist: 0x400, 0x800, 0x4, 0x4 gibt's immer, wenn ich keinen Framebuffer angebe.

[@modzilla](#): Das mache ich bei jeder Änderung, funktioniert aber trotzdem nicht. Der Geräte name ändert sich nur mit Framebuffer Hamachi oder Futomaki.

Beitrag von „aufdenschlips“ vom 4. November 2017, 13:40

@"Spielerei"

Wenn es klappt, hat man die komplette Kontrolle über seine GPUs ohne sich sorgen zu machen, ob es beim nächsten Update schief läuft.

Vielleicht auch interessant für problematische QuickSync&Co Kandidaten, falls dies in der iGPU Sektion ebenso realisierbar ist.

[@modzilla](#): Vielen Dank für die Detektivarbeit!

Beitrag von „modzilla“ vom 4. November 2017, 16:16

Ja prinzipiell kannst du alle Grafikprobleme mit der DSDT lösen 😊
Hat denn jetzt noch jmd ne Idee bzgl des Connector Eintrages in der DSDT?

Beitrag von „kuckkuck“ vom 5. November 2017, 02:10

| [Zitat von modzilla](#)

Hat denn jetzt noch jmd ne Idee bzgl des Connector Eintrages in der DSDT?

Kannst du das mal konkret formulieren? Ich verstehe das Problem nicht so ganz



Beitrag von „modzilla“ vom 5. November 2017, 10:35

Wie gesagt, ich möchte gern den Framebuffer otf patchen mit der DSDT, wie oben mit dem "connectors" Eintrag, aber immer, wenn ich meine eigenen Werte eintrage, dann bleiben die Bildschirme schwarz und QE/CI geht verloren...

Beitrag von „kuckkuck“ vom 5. November 2017, 12:27

Verstehe... Was ist der erste Wert den du injectest und zu welchem Port genau gehört er?

Beitrag von „modzilla“ vom 5. November 2017, 13:53

[Zitat von modzilla](#)

Code

1. "connectors",
2. Buffer (0x50)
3. {
4. /* 0000 */ 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00,
5. /* 0008 */ 0x00, 0x01, 0x00, 0x00, 0x12, 0x04, 0x00, 0x01,
6. /* 0010 */ 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00,
7. /* 0018 */ 0x00, 0x01, 0x00, 0x00, 0x22, 0x05, 0x00, 0x02,
8. /* 0020 */ 0x00, 0x08, 0x00, 0x00, 0x04, 0x02, 0x00, 0x00,
9. /* 0028 */ 0x00, 0x71, 0x00, 0x00, 0x11, 0x02, 0x00, 0x03,
10. /* 0030 */ 0x04, 0x00, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,

11. /* 0038 */ 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06,
12. /* 0040 */ 0x00, 0x02, 0x00, 0x00, 0x14, 0x02, 0x00, 0x00,
13. /* 0048 */ 0x00, 0x01, 0x00, 0x00, 0x21, 0x03, 0x00, 0x05
14. },

Alles anzeigen

Ich weiß jetzt noch nicht genau was du damit meinst, aber die Ports sind wie folgt injected: DP, DP, HDMI, DVI-DL, DVI-SL

Beitrag von „kuckkuck“ vom 5. November 2017, 16:21

Die Werte bestehen immer aus 16 Einzelwerten. Der erste Wert ist bei dir also:

Code

1. 0x00, 0x04, 0x00, 0x00, 0x04, 0x03, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x12, 0x04, 0x00, 0x01

Diese Reihenfolge lässt sich zerlegen, wobei jeder teil einen bestimmten Sinn hat.

Die ersten vier Werte sind beschreiben den Typ des Connectors:
0x00, 0x04, 0x00, 0x00 entspricht einem DisplayPort.

Es gibt:

Die nächsten vier sind die ATY,ControlFlags
0x04, 0x03, 0x00, 0x00 entspricht einem Display Port

Es gibt hier wieder einige... Ein paar:

Danach kommen die ATY,Features:

0x00, 0x01, 0x00, 0x00

Aufgesplittet in:

- 0x00 <-- DP port
- 0x01 <-- Internal

Es gibt:

- 0x00 <-- Reihenfolge der Connector Aktivierung: 01 wäre erster aktiver Connector, hier 00???
- 0x00 <-- unknown

Die letzten vier kommen aus dem ROM der GPU:

0x12, 0x04, 0x00, 0x01

Aufgesplittet in:

- 0x12 <-- Transmitter
- 0x04 <-- Encoder
- 0x00 <-- Hotplug ID
- 0x01 <-- Sense ID

Die Werte erhält man wenn man das ROM mit redsock_bios_decoder und redsock_bios_**decode** dekompiert.

Beispiel für einen Teil der Outputs der beiden:

03 (HOTPLUG ID) [DVI_I]

redsock_bios_decoder :

enc obj 0x1e transmitter 0x10 (TRANSMITTER) dual link 0x0 enc 0x0 (ENCODER)

radeon_bios_decode:

Connector at index 2

Type [@offset 43542]: DVI-I (2)

Encoder [@offset 43546]: INTERNAL_UNIPHY (0x1e)

i2cid [@offset 43696]: 0x95, OSX senseid: 0x6 (SENSE ID)

Daraus lässt sich also insgesamt ein Custom Konstrukt für jeden Port der GPU bilden...

Beitrag von „modzilla“ vom 5. November 2017, 23:24

Jap, exakt! Genau das tat ich ja auch! Der Patch aus dem ersten Post funktioniert auch super, jedoch nicht die abgeänderte Version von vit's connectors Eintrag...

Das Zitat ist meine abgeänderte Version 😊

Beitrag von „kuckkuck“ vom 6. November 2017, 07:06

Wieso solltest du denn vits Eintrag benutzen? Der Eintrag muss doch an die Hardware angepasst werden 😞

Beitrag von „Si Vis Pacem“ vom 6. November 2017, 15:56

Laut Aussage von Mieke und Racermaster drüben bei IM [Tracing back the AMD GPU wakeup issue to its origin](#) werden die Connector Einträge nur von Whatevergreen genutzt und nicht vom System.

Beitrag von „Harper Lewis“ vom 6. November 2017, 16:35

Das hatte ich (mich) hier auch schon gefragt:

[Zitat von Harper Lewis](#)

Soll das denn den selben Effekt bewirken, wie ein Kext-Patch für den AMDXXXXController.kext, oder ist das eine spezielles Viech für WhateverGreen?

Beitrag von „modzilla“ vom 6. November 2017, 18:04

Danke! Ja gut da hatte ich dann ja lange probieren können 😄 @kuckkuck nein, nein da versteh du mich wohl nicht ganz richtig 😊 Der Eintrag mit dem Buffer(0x50) ist mein abgeänderter 😄 der von vit hat nur 4 Display Anschlüsse und hat somit natürlich auch nur n 0x40 Buffer 😊

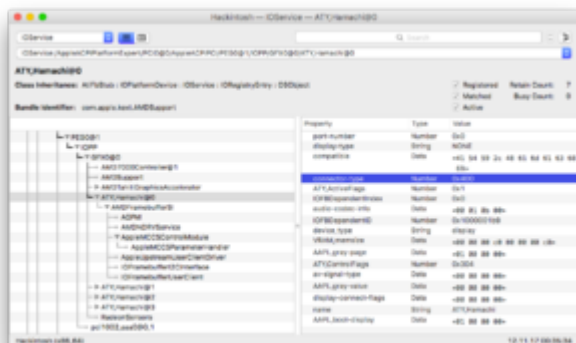
Beitrag von „kuckkuck“ vom 12. November 2017, 00:43

Zitat von Si Vis Pacem

Laut Aussage von Mieke und Racermaster drüben bei IM Tracing back the AMD GPU wakeup issue to its origin werden die Connector Einträge nur von Whatevergreen genutzt und nicht vom System.

Ich will das hier nochmal aufgreifen, denn ich weiß nicht, ob die Aussage so ganz stimmt... Ich meine ja, einfach nur "connectors" und danach alles aufführen, das ist Whatevergreen-Welt, aber mit _DSM Methoden lassen sich doch Properties verändern, was in IOReg ebenfalls zu sehen ist...

Schaut man sich die GPU in IOReg an, sieht man unter dem GFX0/PEGP Device unter anderem den geladenen AMD Controller. Etwas später kommen dann die Einträge für die einzelnen connectors. Sie heißen *ATY,Framebuffer@x* und beinhalten Properties, die sich anscheinend auch mit _DSM Methoden verändern lassen. So habe ich bei meinem ersten Connector den Typ auf DP gesetzt:



Das ganze funktioniert mit

Code

1. "@0,connector-type",
2. Buffer (0x04)
3. {
4. 0x00, 0x04, 0x00, 0x00
5. },

In IOReg wird dies zu 0x400, da ein ByteSwap durchgeführt wird und vorausgehende Nullen wegfallen... Aus 00 00 04 00 (ByteSwap) wird dann 400.

Der Framebuffer lässt sich wie folgt setzen:

Code

1. "@0,name",
2. Buffer (0x0C)
3. {
4. "ATY,Hamachi"
5. },

Ich habe es zwar noch nicht getestet, genauso müssten sich aber auch die ATY,ControlFlags oder das AAPL,boot-display mit einem entsprechenden @x davor in der _DSM-Method, setzen lassen...

Whatevergreen-spezifische Settings wie "connector-priority" oder "connector-count" lassen sich so natürlich nicht ändern, da wir den Framebuffer nicht einfach über ACPI patchen können, aber man könnte zB versuchen die Port-Nummern zu tauschen, und somit Prioritäten zu setzen. Die Anzahl der Connectors können wir ja einfach durch die Wahl eines anderen Framebuffers ändern, überflüssige Anschlüsse kriegen dann halt schön viele 0x00 als connector-type.

Noch kurz zum auslesen SenselD etc, dafür lässt sich super das [AMD Framebuffer Utility](#)

nutzen, wer es nicht schon bereits tut 😊

EDIT: Ich habe gerade bemerkt, dass beim Hamachi der erste Port sowieso schon auf 0x400 steht... Mein IOREG Bild oben ist also noch kein Beweis, dass es funktioniert, ich muss dem nochmal etwas genauer nachgehen.

Beitrag von „modzilla“ vom 12. November 2017, 17:47

Aber ich bin mir relativ sicher, dass es funktionieren wird, nur ich habe es gerade geschafft, dass maciasl nicht mehr startet 😊 Wenn ich mir einen neuen Benutzer anlege, geht es aber....

Beitrag von „Si Vis Pacem“ vom 13. November 2017, 17:50

Falls bei jemanden, so wie bei mir, das Umbenennen der GPU nicht funktioniert:

Code

1. "@3,model",
2. Buffer (0x20)
3. {
4. "Sapphire Nitro R9 390 Backplate"
5. },
- 6.
- 7.
8. "@3,AAPL,slot-name",
9. Buffer (0x0B)
10. {
11. "PCIe-Slot1"
12. },

Alles anzeigen

In den vom Display benutzten Connector. Hat bei mir zumindest geholfen.

<https://www.hackintosh-forum.d...17-11-13-um-17-46-01-png/>

Beitrag von „modzilla“ vom 13. November 2017, 18:41

Ja das mit der "model" Namensänderung ist so eine Sache für sich, denn wenn der RadeonFB gesetzt ist, wird dieser aus welchen Gründen auch immer nicht gesetzt... Wirklich strange

Beitrag von „Harper Lewis“ vom 13. November 2017, 18:47

Bei mir funktioniert die Namensänderung auch nur mit einem zu meiner Karte passenden Framebuffer (Hamachi oder Futomaki).

Beitrag von „Doctor Plagiat“ vom 13. November 2017, 18:57

Kann ich bestätigen. Bei mir ist der Framebuffer-Name "Acre", sobald ich einen anderen nehme wird im System-Profilier eine AMD R9xxx angezeigt.

Beitrag von „kuckkuck“ vom 13. November 2017, 19:20

Habt ihr mal versucht den Namen und Framebuffer für jeden Connector einzeln und generell zu injecten?

Also mit

Code

1. "@x,model"

x=1,2,3,4...

Ebenfalls ist mir aufgefallen, dass Apple selber die Namen in Hex injected... Sie konvertieren den ASCII Namen nach Hex. (Leerzeichen ist 0x20 wenn mich nicht alles täuscht)

Beitrag von „Harper Lewis“ vom 13. November 2017, 19:27

Das ist mir zwar so halb aufgefallen, aber dann eben doch entgangen:



ATY_EFI_InjectBytes	Data	<" "P">
VRAM_InjectMB	Number	0x800
FP_PromiseDummyBackEnd	Number	0x0
model	Data	<41 42 43 20 32 61 64 65 67 68 32 39 20 44 33 20 30>
ATY_EFI_DriverType	Data	<82>
ATY_EFI_Version	Data	<30 31 2e 30 30 2e 30 30 30>

= AMD Radeon R9 M380

Besten Dank, [@kuckkuck!](#)

Beitrag von „Doctor Plagiat“ vom 13. November 2017, 19:49

[Zitat von kuckkuck](#)

Habt ihr mal versucht den Namen und Framebuffer für jeden Connector einzeln und generell zu injecten?

Bei mir steht da zwar @0, name statt model, aber vielleicht ist das ja nicht ganz verkehrt.

```

"@0,AAPL,boot-display",
Buffer (Zero) {},
"@0,name",
Buffer (0x09)
{
  "ATY,Acre"
},

"@1,name",
Buffer (0x09)
{
  "ATY,Acre"
},

"@2,name",
Buffer (0x09)
{
  "ATY,Acre"
},

```

im IOReg sieht es bei mir dann aber so aus

PP_PhmlUseDummyBackEnd	Number	0x0
model	Data	<"Radeon RX 560">
VRAM,totalMB	Number	0x1000
ATY,EFIVersion	Data	<"WEAR-104-2017-08-27">
pcidebug	String	1:0:0

Beitrag von „Si Vis Pacem“ vom 13. November 2017, 19:57

Wenn ich statt Miezes DSDT/SSDT-Ansatz Ozmosis AtiInject nehme, steht bei mir R9 290 8GB mit einem gepatchten Baladi und nvram AtiFramebuffer. Noch will der nicht so ganz mittels DSDT, aber zumindest schaffe ich es schon, dass er mit @3,name ATY,Baladi die restlichen Werte nicht nimmt ;). Scheint also einen Effekt zu haben.

Spiele mich derweil weiter, da es schon sehr verführerisch ist, mittels der DSDT das System auf das i-Tüpfelchen anzupassen.

Schwanke gerade zwischen grauen Boot-Bildschirm und Baladi als Priorität. Wird aber wohl Baladi werden.

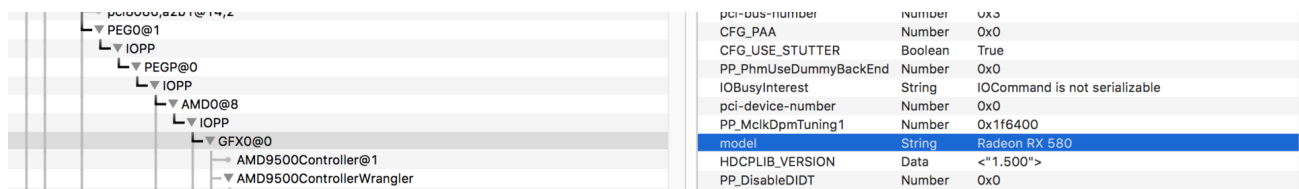
Edit:

Solange ich booten kann, ist es eh nicht wirklich wichtig. Halt spielerisches Erforschen.

Beitrag von „Mork vom Ork“ vom 13. November 2017, 19:58

Mit "@0, name" vergibt man den Framebuffernamen. Mit "model" den Namen der GFX-Karte. Und eigentlich steht "model" immer für sich alleine - ohne ein vorangestelltes "@0," oder "@1," etc.

Klickt man dann im IORegistryExplorer auf das GFX0@0 sollte dort in der Liste auch ein "model, String WieImmerDuDeineKarteBenanntHast" erscheinen:



Beitrag von „Doctor Plagiat“ vom 13. November 2017, 20:02

Ja, so ist das bei mir momentan. Scheint so als wäre meine DSDT ganz gut zusammen geschustert.

Beitrag von „Si Vis Pacem“ vom 13. November 2017, 20:03

[Zitat von Mork vom Ork](#)

Mit "@0, name" vergibt man den Framebuffernamen. Mit "model" den Namen der GFX-Karte. Und eigentlich steht "model" immer für sich alleine - ohne ein vorangestelltes

"@0," oder "@1," etc.

Klickt man dann im IORegistryExplorer auf das GFX0@0 sollte dort in der Lsite auch ein "model, "WielmmerDuDeineKarteBenanntHast" erscheinen.

Tja. nur manchmal nimmt er das dann nicht so, wie er es sollte.

Ich nehme an, dass die model und andere Geschichten unter GFX0 oder PEGP dann funktioniert, wenn auch der erste Connector mit dem/einem Display verbunden ist. Und wenn dies nicht der Fall ist, macOS dieses einfach mal ignoriert.

Jemand anderer hat auch gepostet, es würde reichen den ersten Connector zu definieren, alle anderen würden den FB genauso übernehmen. Wahrscheinlich der Fall, dass wenn der erste funktionierende Connector der Erste ist, er die Richtung vorgibt.

Beitrag von „Mork vom Ork“ vom 13. November 2017, 20:06

[Zitat von Doctor Plagiat](#)

im IOReg sieht es bei mir dann aber so aus

PP_PhMUseDummyBackEnd	Number	0x0
model	Data	<"Radeon RX 560">
VRAM,totalMB	Number	0x1000
ATY,EFIVersion	Data	<"WEAR-104-2017-08-27">
pcidebug	String	1:0:0

Wundert mich nicht, denn Du scheint noch immer LILU und WhatEverGreen zu nutzen. Das sagt mir der Eintrag "ATY,EFIVersion" <-- der stammt DEFINITIV von WhatEverGreen
Und Du hast auch definitiv noch in der CLOVER config.plist "Inject ATI = enabled" gesetzt.
Sachen, die nicht nötig sind, da man all diese Werte, wenn man eh schon eine SSDT nutzt, nun auch in der SSDT setzen kann.

Beitrag von „Si Vis Pacem“ vom 13. November 2017, 20:06

Kann mir jemand sagen, ob/wie man in der DSDT einen Booleschen Wert eintragen kann?

Obwohl meine iGPU nicht primär ist, hat sie ein AAPL,boot-display true unterm @0 connector.

Habe leider keinen Core iMac hm, mal beim Mac mini 2010 schauen. Was dort IOJONES sagt.

Beitrag von „Doctor Plagiat“ vom 13. November 2017, 20:11

[Zitat von Mork vom Ork](#)

Wundert mich nicht, denn Du scheinst noch immer LILU und WhatEverGreen zu nutzen. Das sagt mir der Eintrag "ATY,EFIVersion" <-- der stammt DEFINITIV von WhatEverGreen

Da hast du aber sehr gut aufgepaßt 😄. Ich hatte den Whatevergreen nur auf meiner externen HighSierra-SSD entfernt und in Sierra vergessen. Ich habe ihn jetzt da auch gelöscht, muss mal eben einen Neustart machen. 😊

EDIT: Jetzt ist der Whatevergreen und der Eintrag "ATY,EFIVersion" weg...

PP_MclkDpmTuning1	Number	0x1f6400
model	Data	<"AMD Radeon RX560">
HDCPLIB_VERSION	Data	<"1.500">
PP_DisableDIDT	Number	0x0
acpi-device	String	IOACPIPlatformDevice is
CFG_TRANS_WSRV	Boolean	True

... und NÖÖÖ Inject ATI war bei mir noch nie enable.

Beitrag von „Si Vis Pacem“ vom 13. November 2017, 20:13



Wollte gerade das Selbe schreiben, aber irgendein Fehler beim posten.

Beitrag von „Doctor Plagiat“ vom 13. November 2017, 20:32

Habe hier noch was ergänzt. [EDIT](#)

Beitrag von „Mork vom Ork“ vom 13. November 2017, 20:45

[Zitat von Doctor Plagiat](#)

... und NÖÖÖ Inject ATI war bei mir noch nie enable.

Bitte nicht missverstehen: ich wollte Dir damit NICHTS unterstellen 😊

Mich wunderte halt nur, das "model" bei Dir als DATA <"AMD Radeon..."> angezeigt wird. Hat man es in der SSDT korrekt eingepflegt, sollte es eigentlich als " STRING BalBlaBla" eingefügt werden (siehe dazu auch meinen letzten Screenshot).

Das einfügen als DATA <"BlaBlaBla"> ist ein typisches Verhalten der CLOVER Inject ATI = true Funktion. Denn CLOVER setzt dabei alle eigentlich als STRING definierten Werte als DATA <"XYZ"> ein. Ein Umstand, der mir schon so manches Mal sauer aufgestossen ist.

Beitrag von „Doctor Plagiat“ vom 13. November 2017, 20:53

[Zitat von Mork vom Ork](#)

Bitte nicht missverstehen: ich wollte Dir damit NICHTS unterstellen

So empfindlich bin ich nicht, ist alles OK.

Ich sollte vielleicht noch erwähnen, dass ich das ganze Grafik-Gedöns in der DSDT habe, also keine Extra-SSDT.

Nur wenn du Lust und Zeit hast, kannst du ja mal drübergucken.

[DSDT-GA-Z170M-D3H.zip](#)

Beitrag von „kuckkuck“ vom 13. November 2017, 21:35

[Zitat von Mork vom Ork](#)

Mich wunderte halt nur, das "model" bei Dir als DATA <"AMD Radeon..."> angezeigt wird. Hat man es in der SSDT korrekt eingepflegt, sollte es eigentlich als " STRING BalBlaBla" eingefügt werden

Naja, kommt drauf an wie man es macht, bei Apple selber ist es auch Data:

[Zitat von kuckkuck](#)

Ebenfalls ist mir aufgefallen, dass Apple selber die Namen in Hex injected... Sie konvertieren den ASCII Namen nach Hex.

Aber Data + ASCII String ist sicherlich die falsche Kombi, wenn schon Data, dann Hex 😊

[Zitat von Mork vom Ork](#)

Und eigentlich steht "model" immer für sich alleine - ohne ein vorangestelltes "@0," oder "@1," etc.

Stimmt! Habe gerade auch nochmal bei mir nachgeschaut und da gehört echt kein @x davor, also Kommando zurück! 😊

Beitrag von „Mork vom Ork“ vom 13. November 2017, 21:39

[Zitat von Doctor Plagiat](#)

Nur wenn du Lust und Zeit hast, kannst du ja mal drübergucken.

Sieht sehr gut aus. Aber versuch bitte mal "ATY, Card#" in "ATY,Part#" zu ändern. Wird aller Voraussicht nach keine Änderung in irgendwas bewirken, "ATY,Card#" hiess es noch zu "ATI Radeon HD"-Zeiten.

Aber sonst: erste Sahne, sehr sauber durchstrukturiert. Jibbt et nüscht zu vabessahn, wah ey!



PS: setzte auch mal folgendes testweise, bitte:

Code

1. "model",
2. "AMD Radeon RX560",

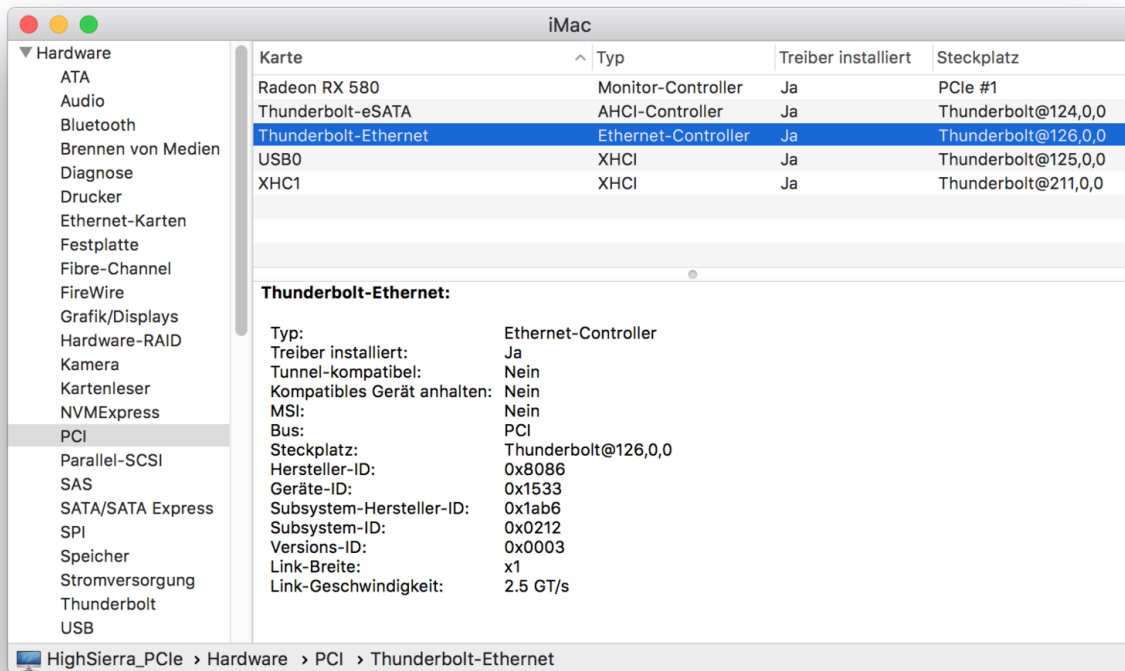
Vergiss dabei aber bitte nicht, auch folgenden Wert anzupassen:

Code

1. Store (Package (0x18)

so habe ich die Namen der einzelnen Devices in meiner Thunderbolt3-SSDT angepasst

bekommen:



Beitrag von „Doctor Plagiat“ vom 13. November 2017, 21:47

Danke dir. Ich bin aber nicht der DSDT-Experte. Ich habe sie mit den vorhandenen Pjalm- und Sourceforge-Repas und natürlich mit den vielen Beispielen hier im Forum gepatched. Den ATY,Card# Eintrag werde ich deiner Empfehlung in ATY,Part# ändern. Leider ist der ATY,Part# String bei mir xxx-xxx-xxx.

Beitrag von „kuckkuck“ vom 13. November 2017, 21:52

In einen original iMac17,1 IOReg Dump heißt es ebenfalls ATY,Card# (= <109-C60051-00>) 🙄

Beitrag von „Mork vom Ork“ vom 13. November 2017, 21:54

[Zitat von Doctor Plagiat](#)

Leider ist der ATY,Part# String bei mir xxx-xxx-xxx.

Macht ja nichts. Im Grunde genommen scheint der ja austauschbar zu sein. Ich habe meinen ja auch via Kext-Patch angepasst.

[@kuckkuck](#)

was hat der original iMac für eine Grafikkarte? ATI Radeon HD 8950 ???

Beitrag von „Doctor Plagiat“ vom 13. November 2017, 22:00

Dann müsste man aber einen passenden string kennen. Sind da in der Mitte nicht immer 5 Werte, also xxx-xxxxx-xxx ?

Ich hatte ja hier gelesen, dass damit Metal2 aktiviert werden kann. Ist mir bis jetzt noch nicht gelungen.

Beitrag von „kuckkuck“ vom 13. November 2017, 22:08

[Zitat von Mork vom Ork](#)

was hat der original iMac für eine Grafikkarte? ATI Radeon HD 8950 ???

Ist nicht meiner, aber nach IOReg hat der eine AMD Radeon R9 M380... Mir ist soweit auch kein iMac 17,1 mit HD 8950 bekannt.

Vielleicht weiß [@Harper Lewis](#) mehr darüber, welcher iMac das genau ist 😊

Beitrag von „Mork vom Ork“ vom 13. November 2017, 22:21

[Zitat von Doctor Plagiat](#)

Dann müsste man aber einen passenden string kennen. Sind da in der Mitte nicht immer 5 Werte, also xxx-xxxxx-xxx ?

Ich hatte ja hier gelesen, dass damit Metal2 aktiviert werden kann. Ist mir bis jetzt noch nicht gelungen.

Versuch mal den String: "113-4E353BU-U02"

Ich bin immer noch der Meinung, das Metal2 nur bei AMD RX580 Karten mit der passenden Part# in Kombination mit dem ORINOCO Framebuffer aktiviert wird. Warum bin ich dieser Meinung?

- a) weil das die Kombi von Apple's eGPU Lösung bislang ist - und
- b) weil ich nur Metal2 bekomme, wenn ich genau diese Kombi unter CLOVER einstelle. Nutze ich einen anderen FB, aber eine der folgenden Part#: 113-4E353BU-xxx (xxx kann beliebig sein) oder 113-4E3531U-xxx (xxx kann beliebig sein) oder aber ORINOCO als FB, aber eine andere Part#, bekomme ich bei mir auch kein Metal2-Support mehr angezeigt.

[@kuckkuck](#)

Ja, die Radeon R9 M380 sollte wohl richtig sein, jedoch lt. dem von Dir angefügten IOReg nutzt er als DeviceID die 6640, welche eigentlich eine HD8950 sein sollte

Beitrag von „Harper Lewis“ vom 14. November 2017, 15:47

AMD Radeon R9 M380 passt, genau.

Geräte-ID: 0x6640

= AMD Radeon™ R9 M380

Beitrag von „Doctor Plagiat“ vom 14. November 2017, 18:57

[Zitat von Mork vom Ork](#)

Ich habe meinen ja auch via Kext-Patch angepasst.

Ich glaube dir die Aussage, dass Metal2 wahrscheinlich nur mit der RX580 funktioniert und würde trotzdem den Test mal machen. Ich habe hier leider nur den Connector-Patch gefunden und würde mich nochmal über deine Hilfe freuen.

Welche Find- und Replace-Werte müsste ich denn einfügen für den AMD9500Controller.kext ? Der Replace-Wert wäre "113-4E353BU-U02" in HEX umgerechnet?

Beitrag von „Mork vom Ork“ vom 14. November 2017, 19:55

[Zitat von Doctor Plagiat](#)

Welche Find- und Replace-Werte müsste ich denn einfügen für den AMD9500Controller.kext ? Der Replace-Wert wäre "113-4E353BU-U02" in HEX umgerechnet?

Kein Find-and-Replace nötig in Deinem Fall: einfach diesen Wert: "113-4E353BU-U02" an der Stelle Deiner DSDT einfügen bzw. ersetzen, an der Du Deinen "ATY,Card#" (ATY,Part#) gesetzt hast. Das sollte schon reichen, da dieser Wert ja aus der AMD9500Controller.kext stammt. Sprich hier gleicht der kext deine Part# mit dieser ab und wenn diese übereinstimmen, nimmt er an, bei Deiner Karte handelt es sich um eine RX580 und schaltet Metal2 frei.

Sollte es der Wert nicht bringen, versuch den anderen, den ich weiter oben noch genannt

hatte, denn in dem AMD9500Controller.kext stehen beide Werte drin.

Beitrag von „Doctor Plagiat“ vom 14. November 2017, 20:14

Na wenn das so einfach ist, dann kriege ich das sogar hin. 😄 Ich werde berichten. DANKE!

EDIT: Ergebnis ist schon vorhanden, wenn auch nicht sichtbar, da es nicht geholfen hat. Geschadet hat es aber auch nicht.

EDIT2: Wird im IOReg auch angezeigt ..

ATY,Part#	Data	<"113-4E353BU-U02">
ATY,SCLK	Number	0x5397
revision-id	Data	<cf 00 00 00>
CFG_NO_MSI	Boolean	False
ATY,MCLK	Number	0x7530
CFG_USE_CP2	Boolean	True

Beitrag von „modzilla“ vom 14. November 2017, 21:19

So klappt's auch! Naja gut, dann nutze ich jetzt den Futomaki Framebuffer, auch wenn dann kein SL-DVI mehr funktioniert...