

**Erledigt**

## **Problem die RX 480 Powercolour zu installieren**

**Beitrag von „pawelpipowich“ vom 21. April 2017, 15:29**

Guten Tag liebe gemeinde,

ich habe nun endlich geschafft HAckintosh zu Installieren. Mein System:  
i5 3570K, Asus H61M-K, Powercolour RX 480 8GB DDR5  
sierra 10.12.2

kann mir jemand schritt für schritt erklären was ich tun soll. ich habe viel gegoogelt und habe einiges gefunden aber ich scheiter daran schon /System/Libery/Extensions/ aufzurufen. die ist auch bei mir leer... Wie ich das verstanden habe muss ich da eine ATI4xxxx ktext finden und auf Desktop kopieren dann da einiges eintragen und dann sollte meine karte wenigstens als eine r9 fury oder ähnlich laufen. aber wie gesagt bei mir ist der Ordner ja leer. dann habe ich ziemlich gleiche Anleitungen im netz gefunden, da ist dann aber anschliesen rede von einem coverconfigurator. ich komme einfach nicht weiter.

wenn jemand die gleiche karte hatt wäre das ziemlich nett wenn er mir schritt für schritt erklärt was zu tun ist.  
dabei müsste ich auch eventuel grundlegendes erklärt bekommen.

---

**Beitrag von „ralf.“ vom 21. April 2017, 15:56**

[Zitat von pawelpipowich](#)

bei mir ist der Ordner ja leer.

Wenn das keine M.2-SSD ist, dauert es etwas bis die Kexte angezeigt werden.  
Sonst schauen ob die [SIP](#) deaktiviert ist. Evtl noch ShowAllfiels ausführen.

Die ID deiner Karte [hier](#) eintragen

---

### Beitrag von „pawelpipowich“ vom 21. April 2017, 16:01

Was ist [SIP](#). ? Und wie kann ich ShowAllFiles ausführen. ?  
edit: habe ich rausgefunden.

Und die blidanleitung verstehe ich. Nur wo wird die kext ulity ausgegührt? Mir fehlt da irgendwie der anfang

---

### Beitrag von „derHackfan“ vom 21. April 2017, 21:00

[Zitat von pawelpipowich](#)

Was ist [SIP](#). ?

Guckst du [hier](#) und [hier](#) ... [System Integrity Protection](#). 😊

Btw: Kannst du bitte deine Hardware Angaben aus dem ersten Beitrag im Benutzerprofil und oder in der Signatur eintragen, so weiss jeder Besucher im Thread gleich worum es geht.



---

### Beitrag von „DerJKM“ vom 21. April 2017, 21:51

Hier die Anleitung für RX480 Karten mit der die volle Performance erreicht wird:

[MSI Radeon RX 480 Gaming X unter Sierra 10.12.2 \(mit allen 36 Compute Units\)](#)

Im Prinzip ist die oben verlinkte Anleitung korrekt, jedoch werden die falschen Kexte editiert.

Für die RX480 müssen die AMD9510Controller.kext und die AMDX4100.kext editiert werden

---

**Beitrag von „ralf.“ vom 21. April 2017, 22:51**

### [Zitat von DerJKM](#)

Im Prinzip ist die oben verlinkte Anleitung korrekt, jedoch werden die falschen Kexte editiert. Für die RX480 müssen die AMD9510Controller.kext und die AMDX4100.kext editiert werden

In den ersten Sierra-Versionen war es der AMD9500Controller.kext

```
<string>com.apple.kext.AMD9510Controller</string>
<key>IOClass</key>
<string>AMD9510Controller</string>
<key>IOMatchCategory</key>
<string>IOFramebuffer</string>
<key>IOName</key>
<string>AMD9510Controller</string>
<key>IOPCIMatch</key>
<string>0x67DF1002</string>
```

AMDX4100.kext

hängt von der Sierra-Version ab. In den ersten Sierra-Versionen wars der AMDX4000.kext, wo Baffin enthalten war.

```
<string>com.apple.AMDRadeonX4000</string>
<key>GpuDebugPolicy</key>
<integer>0</integer>
<key>IOClass</key>
<string>AMDBaffinGraphicsAccelerator</string>
<key>IODVDBundleName</key>
<string>AMDRadeonVADriver</string>
<key>IOKitDebug</key>
<integer>0</integer>
<key>IOMatchCategory</key>
<string>IOAccelerator</string>
<key>IOPCIMatch</key>
<string>0x67DF1002 0x67FF1002 0x67EF1002</string>
```

[@pawelpipowich](#)

Den roten Teil ändern. Du musst den Part finden, wo es um die Baffin-GPU geht, siehe Bild

---

### Beitrag von „pawelpipowich“ vom 22. April 2017, 22:14

Vielen Dank alles Profis hier 😊 Tut mir leid das ihr mit mir euch rumschlägt das ist sehr nett.

jetzt muss ich das irgendwie in die tat umsetzen. also erste aufgabe erst diese ktexte finden.  
wenn ich das mit editor geändert bekomme einfach abspeichern... und dann sollte die karte  
einfach laufen?  
zurzeit läuft die als 7mb grafikkarte

---

## **Beitrag von „ralf.“ vom 22. April 2017, 23:17**

EDIT

[@pawelpipowich](#)

Mit dem Editor ändern. Direkt abspeichern geht nicht (es sei denn man verschiebt den Kext auf den Desktop). Aber die Option Duplicate geht, und abspeichern auf dem Desktop. Die muß dann genauso heißen. Als nächstes die Datei vom Desktop in den Kext ziehen und die alte ersetzen, mit Passwort bestätigen. Kext Utility laufen lassen.

Die IGP noch als initial-Grafik einstellen.

Und bitte die Hardware ins Profil oder die Signatur eintragen. Kein großer Aufwand.

---

## **Beitrag von „pawelpipowich“ vom 23. April 2017, 00:53**

Ich habe die kext datei gefunden. Wo auch Baffin Steht. Habe auch das rote geändert und die kextdatei in kext ulity per drag and drop reingeworfen. Es scheint alles funktioniert zu haben. Die datei hies AMDRadeonX4100.kext. Die Graka allerdings wird weiter als 7mb karte erkannt. Oder habe ich was da vergessen?

oder muss ich da noch was mit cloverconfigurator was machen?

so nun habe ich alles so hinbekommen. Nur jetzt ist das Problem das ich nicht hochfahren kann bleibt eingefrohren kurz vor dem Apfel Balken ende. Wenn ich auf pcie also auf rx480 schalte fährt osx hoch aber mit low performance grafik 7mb

---

## Beitrag von „ralf.“ vom 23. April 2017, 08:22

[Zitat von ralf.](#)

Die IGP noch als initial-Grafik einstellen.

Im Bios.

Bootet dann Blind bis zu Desktop. Oder 2.Monitor

---

## Beitrag von „pawelpipowich“ vom 23. April 2017, 14:26

Ne den boot und uefi clover sehe ich. Bleibt dann kurz vorm ende des apfel balken hängen. Habe ich da noch was vergessen? Bild habe ich. Angeschlossen auf dvi am mainboard

---

## Beitrag von „ralf.“ vom 23. April 2017, 14:36

Das ist dann aber falsch.

Noch mal: Die IGP noch als initial-Grafik einstellen.

Monitor nicht am Mainboar anschließen.

---

## Beitrag von „pawelpipowich“ vom 23. April 2017, 14:50

hab da nur die möglichkeit der priorität einzustellen als pcie oder igpu oder Auto. Wenn ich igpu einstelle habe ich nur auf mainboard bild. Wenn pcie dann nur auf der graka.

---

## Beitrag von „ralf.“ vom 23. April 2017, 14:58

[Zitat von pawelpipowich](#)

Wenn ich igpu einstelle habe ich nur auf mainboard bild.

So soll es ja auch sein. Damit [dies](#) funzt.

---

## Beitrag von „pawelpipowich“ vom 23. April 2017, 15:30

Also einfach dvi an pcie graka anschliessen. Und dann warten bis bild habe? Aber wenn ich dvi am mainboard angeschlossen habe, habe ich doch das boot bild. Es freezt halt kurz vom ende des apfel balken ein. Und das wird nicht mit dem graka ausgang passieren?

---

## Beitrag von „griven“ vom 23. April 2017, 15:51

Korrekt da sollte das nicht passieren. Der Hintergrund ist das der Rechner die Treiber für die AMD Karte erst innerhalb des Bootvorgangs lädt und dann auch auf diese Treiber umschaltet das Ergebnis ist ein schwarzes Bild wenn an der AMD nichts angeschlossen ist 😊

---

## Beitrag von „pawelpipowich“ vom 23. April 2017, 18:17

bildschirm bleibt schwarz leider.

Ich vermute langsam das es am fehlenden hdmi port am mainboard liegt. Es freezt bei dem satz. "Sound assertion in AppleHDADriver at line 3747 und 1098."

da ich einfach dvi und vga ausgang habe. Kein hdmi

darnach kommt drmstatus PPGTT is enabled und darnach busy time out ....

---

### **Beitrag von „ralf.“ vom 23. April 2017, 18:30**

Nach meinen Infos gibt es keine RX480 mit VGA.

Finde mal die ID der Karte raus. Geht über Apfelmenü, DPCIManager, Sysinfo, oder in Win: GPU-Z

---

### **Beitrag von „pawelpipowich“ vom 23. April 2017, 18:34**

Vga am board und dvi. Kein hdmi. Die rx480 hatt das nicht

nochmal zur verständnis.

Kext bearbeitet. Darnach bios auf igpu eingestellt. Ergebnis:

dvi angeschlossen am mainboard: habe bild und seeeeeeehhhr langer boot vorgang. Freezt ein.

Dvi anschluss am graka. Kein bild auch nicht nach 10min.

---

### **Beitrag von „griven“ vom 23. April 2017, 20:41**

Die AMD müsste laut Datenblatt 1XDVI, 2XDP und 1XHDMI besitzen als Ausgänge. Der genutzte Framebuffer (Baffin) definiert 4 Displayports demnach kaum ein Wunder das aus dem DVI Anschluss kein Signal kommt 😄 Entweder den Monitor per DP anbinden (es geht auch DP zu HDMI adapter) oder aber den Framebuffer patchen sonst wird das nix...

---



### **Beitrag von „pawelpipowich“ vom 23. April 2017, 21:15**

Aha mist. Und ich setze neu auf. Aber danke für die info. Also dieses hex umändern ist einfach portbelegung?

noch neben bei weis einer warum mit Asus h61m-k ab und zu maus nicht reagiert und tastatur auch. Manchmal wird boot ob install boot oder auch normal boot von der festplatte mit verbotszeichen gefreezt? Hdenke es liegt an einer einstellung im bios. Weil vorhin hatte dieses lange booten problem nicht. Aber leider Lästs sich nicht viel einstellen.

Und fetten dank gür die Geduld. Echt Komponente leute hier

---

### **Beitrag von „griven“ vom 23. April 2017, 23:28**

Sonst schau Dir bitte auch mal diesen Thread an hier ist einiges von Interesse für Dich drin:  
[MSI Radeon RX 480 Gaming X unter Sierra 10.12.2 \(mit allen 36 Compute Units\)](#)

---

### **Beitrag von „pawelpipowich“ vom 24. April 2017, 00:20**

Vielen lieben dank. Den thread kenne ich berreits. Bei mir scheitert es wenn ich auf interne graka einstelle. Freezt ein. Ich schätze es liegt am mainbord. Hat nähmlich kein hdmi.

---

### **Beitrag von „griven“ vom 24. April 2017, 00:40**

Das ist eigentlich vollkommen egal es geht hierbei nur und ausschließlich darum das die Interne GPU auf Bios Ebene vor der RX480 initialisiert wird sprich OS-X muss damit nicht umgehen können es reicht wenn die IGPU einfach initialisiert wird. Wobei Dein I5 sogar eigentlich auch eine voll taugliche HD4000 besitzt sprich die wäre sogar auch voll nutzbar

(muss nicht HDMI sein es geht auch DVI oder DP je nachdem was das Mainboard halt hat nur VGA nicht). Im Idealfall richtest Du alles so ein, dass der Rechner die iGPU als erstes initialisiert (im Bios einstellen) und so bootet (macht er dann ohne Bild solange kein Monitor angeschlossen ist). Clover wird so konfiguriert wie im verlinkten Thread beschrieben und ferner so eingestellt das OS-X direkt und ohne weitere Interaktion startet. Das Ergebnis ist zwar ein Blindflug bis zum Login Screen von OS-X dafür dann aber auch eine laufende RX480.

---

### **Beitrag von „pawelpipowich“ vom 24. April 2017, 09:31**

muss monitor an einem displayport hängen? Oder kann ich hdmi weiter nutzen?  
bis jetzt nicht weiter gekommen bildschirm bleibt schwarz

---

### **Beitrag von „ralf.“ vom 24. April 2017, 10:23**

textedit

Ein bisschen Tuning: [Per BIOS-Flash von der RX 480 zur RX 580](#)

---

### **Beitrag von „pawelpipowich“ vom 24. April 2017, 11:29**

Danke. Ich bin vezweifelt. Ich habe alles so wie beschrieben erstellt. Unt trotzdem bildschirm bleibt schwarz. Habe die Kext bearbeitet über Kext Uility laufenlassen, im Bios die interne iGPU akteviert, hdmi ist an der rx480 angeschlossen. Und trotzdem schwarz

edit: habe erfahren das mit sierra 10.12.3 die rx480 als eine R9 karte erkannt wird. Das reicht mit ja im grunde auch. Was mache den nun wenn ich ein usb unibeasg mit 10.12.4 habe. Kann man irgendwie downgraden? Oder kann ich irgendwie ein unibeast mit 10.12.3 bekommen? da ja im appstore die 10.12.4 aktuell ist?

---

### **Beitrag von „griven“ vom 24. April 2017, 12:25**

Ob da R9 oder RX480 steht ist an der Stelle Jacke wie Hose sprich es würde an Deinem aktuellen Problem nichts ändern. Entweder hast Du in Deinem Workflow an irgendeiner Stelle einen Bock eingebaut oder aber Deine Karte gibt über den HDMI Port kein Bild aus. Ich habe das mit dem Displayport nicht ganz ohne Hintergedanken erwähnt oder anders gesprochen ohne einen Framebufferpatch der den primären Ausgang als HDMI definiert bleibt der Schirm dunkel. Du hast jetzt 2 Möglichkeiten entweder kaufst Du Dir einen DP zu HDMI Adapter und hängst den Monitor an einen der Displayports (hier dann ruhig alle durchprobieren) oder aber Du beschäftigst Dich damit den Framebuffer entsprechend zu patchen (meine R9-270X gibt am HDMI auch nur dann ein Bild aus wenn am DP Port etwas angeschlossen ist. Ich arbeite mit einem DP zu HDMI Adapter und habe den 2. Monitor dann an HDMI hängen). Der Gedanke das mit 10.12.3 ein anderes und/oder besseres Ergebnis zu erzielen wäre kannst Du direkt vergessen.

---

### **Beitrag von „pawelpipowich“ vom 24. April 2017, 12:45**

Soll ich dir RX480 erst draussen lassen? Sprich ich installiere sierra neu mit laufender igpu. Bearbeite kext und dann cloverconfig. Dann rx480 einbauen. Nur igpu auswählen 128mb zuweisen. Oder ist das auch Jacke wie Hose?  
weil bis jetzt hatte ich mit der rx480 gearbeitet. Hab ja bild nur halt auf 7mb low performance

---

### **Beitrag von „ralf.“ vom 24. April 2017, 14:13**

Framebuffer - [ich glaub das hier ist eine gute Anleitung](#). Hab die jetzt nicht genauer studiert.

Oder einfach ein DP-HDMI-Kabel. Hatte bei Ebay eins für EUR 6,49 gekauft. Das kann angeblich auch 4K.

Oder einen Adapterstecker für 1,66

---

### **Beitrag von „pawelpipowich“ vom 24. April 2017, 16:56**

Habe den adapter bestellt. Aber habe wenig hoffnung darauf. Es ist nirgends die rede vom dp in den ganzen anleitung gewesen.

---

### **Beitrag von „griven“ vom 26. April 2017, 21:54**

Was aber vielleicht auch einfach daran liegen könnte das der DisplayPort inzwischen eine ziemlich verbreitete Schnittstelle ist und man es daher schlicht nicht weiter erwähnt?

---

### **Beitrag von „pawelpipowich“ vom 27. April 2017, 01:40**

Ich hoffe sehr das es der grund ist. Warte noch auf den Adapter.

Mir ist aber noch was aufgefallen, das wenn ich im Bios Grafik auf IGFX einstelle und am Hdmi angeschlossen bin in der rx480. Der pc nicht mehr bootet. Ich meine das keine aktivität der festplatte vorhanden ist. Pc lässt sich sogar einfach ausschalten. So als ob der nicht im Betriebssystem ist. Habe ich am mainboard hdmi angeschlossen b bootet er. Mittlerweile habe ich ein anderes Mainboard gekauft. Und das komische dran ist das wenn ich auf mainboard bleibe per hdmi mac os startet und ich kann moch anmelden. Das war mit meinem alten board nicht so. Jetzt habe ich ein ga-z77-d3

---

### **Beitrag von „griven“ vom 29. April 2017, 01:35**

Wie auch immer die IGPU wird nur und ausschließlich als Starthilfe benötigt und sonst für nichts weiter.

Ich handhabe es mit meiner R9-270X auch nicht anders. Im Bios habe ich eingestellt das die IGPU aktiv ist und als erste Initialisiert werden soll (an der IGPU ist nichts angeschlossen beide Monitore hängen an der R9). Ich verwende für die IGPU die folgende PlattfomID: 0x01660003

vielleicht ist das ja der passende Hint für Dich denn wenn die IGPU nicht korrekt initialisiert werden kann kommt OS-X natürlich auch nicht auf die Beine...

---

## **Beitrag von „Mork vom Ork“ vom 29. April 2017, 10:26**

Hallo Zusammen,

um hier wirklich mal effizient helfen zu können, fehlen von pawelpipowich einfach mal grundlegende Angaben:

- a) welches Mainboard nutzt er nun (wenn er doch oben sagt, er nutze nun ein neues Mainbaord): Hersteller und Modell des Mainboards, ggf. auch welche BIOS-Version.
- b) welche PowerColour RX480 hat er genau: das Referenzdesign, die Red Devil, die Red Dragon? Welche Anschlüsse hat die Karte genau?  
sprich wieviele HDMI-, wieviele Displayport- und wieviele DVI-Anschlüsse hat die Karte?
- c) ist es möglich den vorhandenen Monitor ersteinal nur an die Mainboard-eigenen Grafikanschlüsse anzuschliessen und zunächst ausschliesslich darüber zu betreiben?  
die RX480 Karte kann aber durchaus eingebaut bleiben.

Dann sollte zunächst mal das Mac OS X 10.12. mindestens auf die aktuelle 10.12.4 Version upgedated werden... damit wir eine Grundlage haben, welche Kexte in S/L/E genau wie geändert und angepasst werden müssen.

Wenn diese Angaben und Vorraussetzungen geschaffen sind, dann lässt sich wesentlich leichter eine detaillierte Schritt-für-Schritt-Anleitung erstellen.

[pawelpipowich](#): wenn Du diese Angaben nachreichst und uns Bescheid gibst, bin ich gerne bereit, Dir hierbei Schritt-für-Schritt zu helfen.

Regards... Mork vom Ork

---

### Beitrag von „pawelpipowich“ vom 29. April 2017, 23:58

Ich habe nun Gigabyte Ga-z77-d3 mainboard.

AMD RX 480 von Powercolor Red Dragon

mainboard hat nur HDMI.

Die RX480 4mal displayport 1hdmi 1 dvi

ja man kann über mainboard hdmi den rechner betreiben. Wenn ich alles auf igfx und interne graka aktiviere.

Sierra ist auf dem neuesten stand 10.12.14

wie erwähnt wenn ich die Grafik onboard aktiviere und als initialisierung IGFX bootet nichts.ich warte noch auf den Adapter dp zu hdmi, vielleicht tut sich dann was. da wenn ich auf mainboard per hdmi anschließe fängt os zu booten.

zurzeit aber habe ich das Problem gelöst indem ich mir eine oob Grafikkarte zugelegt habe die ATI HD 7850. die macht den Job auch sehr gut. Hoffe das meine RX 480 bald oob wird. Oder kann ich da lange warten?

---

### Beitrag von „griven“ vom 30. April 2017, 01:22

Mit dem [@Mork vom Ork](#) bist Du was die RX Karten angeht in den besten Händen die man sich wünschen kann. Es gibt meines Wissens nach niemanden der mehr Forschung auf dem Gebiet betrieben hat als er 😄

---

### Beitrag von „Mork vom Ork“ vom 30. April 2017, 09:02

So, da ich nun alle für mich relevanten Daten von Dir (pawelpipowich) habe, fange ich mal an.

Als erstes müssen wir dafür sorgen, daß Dein Mainboard via HDMI mit deinem Monitor verbunden ist. Sollte Dein Monitor nur einen Displayport- oder DVI-Eingang haben, benötigst Du dafür ein HDMI-auf-Displayport- oder HDMI-auf-DVI-Kabel. Bekommst Du günstig und schnell via AMAZON.DE.

Als nächstes musst Du dein Gigabyte-BIOS so konfigurieren, daß Deine IGPU (Mainboardinterne Grafik) als primary gesetzt ist. Da dieses Mainboard (und das dazu passende letzte BIOS-Update) bereits von 2013 ist, schau bitte mal, ob Du im BIOS auch eine Funktion namens "CSM" hast, die Du ggf. ein- bzw. ausschalten kannst. Schalte diese Funktion bitte ein. Lt. meinen GOOGLE-Recherchen heisst die Funktion in deinem BIOS "CSM Support" und befindet sich unter den Einstellungen "BIOS Features".

Wie ich bereits in meinem letzten Beitrag erwähnt habe, kann die PowerColor RX480 bereits im PCIe slot #1 eingebaut sein. In erster Linie geht es darum, das Du mittels der IGPU ins BIOS kommst und somit auch unter CLOVER ein Bild auf dem Monitor hast.

Gehen wir mal davon aus, daß dies nun der Fall ist, so kommen wir zum nächsten Schritt: wir benötigen für die RX480 einen für SIERRA (macOS 10.12.4) passenden Framebuffer, mit dem wir die 3 Displayport sowie den HDMI- als auch den DVI-Anschluss ansprechen können. Ich habe mir das VBIOS Deiner PowerColor RX480 bei Techpowerup.com besorgt und hoffe mal, daß ich das richtige erwischte habe.

Für den Framebuffer-Patch benötigen wir 2 Scripts: **radeon\_bios\_decode.sh** und **redsock\_bios\_decoder.sh** sowie die ".rom"-Datei Deines VBIOS. Beide Scripts findet Ihr als Anhang zu diesem Beitrag unten. Anwendung beider Scripts wäre wie folgt:

Unter OS X ein Terminalfenster aufmachen, das erste Script per drag-and-drop ins offene Terminalfenster ziehen, gefolgt von einem "<" und einem Leerzeichen und anschliessend das "xxx.rom"-File per drag-and-drop ins Terminalfenster ziehen. Dann sollte dort in etwa stehen (das erste ist jeweils der Pfadname, der bei Euch anders aussehen kann und derzeit meine Pfade zeigt, unter dem die Dateien bei mir liegen):

```
" /Volumes/Install\ macOS\ Sierra/APPS_and_FILES/Decoders/radeon_bios_decode <
```

/Users/mvo/Desktop/**Powercolor.RX480.8192.160727.rom** "

Jetzt ENTER drücken und man erhält daraufhin folgende Ausgabe:

Code

1. ATOM BIOS Rom:
2. SubsystemVendorID: 0x148c SubsystemID: 0x2372
3. IOBaseAddress: 0x0000
4. Filename: I1727OAD.SLC
- 5.
- 6.
7. BIOS Bootup Message:
8. D00901 Polaris10 XT A1 GDDR5 256Mx32 8GB I1727OAD.SLC 2016
- 9.
- 10.
11. PCI ID: 1002:67df
12. Connector at index 0
13. Type [@offset 40846]: DisplayPort (10)
14. Encoder [@offset 40850]: INTERNAL\_UNIPHY2 (0x21)
15. i2cid [@offset 40956]: 0x90, OSX senseid: 0x1
16. HotPlugID: 6
- 17.
- 18.
19. Connector at index 1
20. Type [@offset 40856]: DisplayPort (10)
21. Encoder [@offset 40860]: INTERNAL\_UNIPHY2 (0x21)
22. i2cid [@offset 40983]: 0x92, OSX senseid: 0x3
23. HotPlugID: 4
- 24.
- 25.
26. Connector at index 2
27. Type [@offset 40866]: DisplayPort (10)
28. Encoder [@offset 40870]: INTERNAL\_UNIPHY1 (0x20)
29. i2cid [@offset 41010]: 0x91, OSX senseid: 0x2
30. HotPlugID: 1
- 31.
- 32.
33. Connector at index 3
34. Type [@offset 40876]: HDMI-A (11)
35. Encoder [@offset 40880]: INTERNAL\_UNIPHY1 (0x20)
36. i2cid [@offset 41037]: 0x93, OSX senseid: 0x4



37. HotPlugID: 5
- 38.
- 39.
40. Connector at index 4
41. Type [@offset 40886]: DVI-D (3)
42. Encoder [@offset 40890]: INTERNAL\_UNIPHY (0x1e)
43. i2cid [@offset 41064]: 0x95, OSX senseid: 0x6
44. HotPlugID: 3

Alles anzeigen

Auf die selbe Art und weise holen wir uns mit dem zweiten Script nun noch die fehlenden Werte:

script plus "< " plus VBIOS.ROM-File ins Terminalfenster ziehen und ENTER drücken. Ergebnis sieht dann wie folgt aus:

Code

1. D00901 Polaris10 XT A1 GDDR5 256Mx32 8GB I1727OAD.SLC 2016
2. Subsystem Vendor ID: 148c
3. Subsystem ID: 2372
4. Object Header Structure Size: 340
5. Connector Object Table Offset: 48
6. Router Object Table Offset: 0
7. Encoder Object Table Offset: fb
8. Display Path Table Offset: 12
- 9.
- 10.
11. Connector Object Id [19] which is [DISPLAY\_PORT]
12. encoder obj id [0x21] which is [INTERNAL\_UNIPHY2 (osx txmit 0x12 [duallink 0x2] enc 0x4)] linkb: false
- 13.
- 14.
15. Connector Object Id [19] which is [DISPLAY\_PORT]
16. encoder obj id [0x21] which is [INTERNAL\_UNIPHY2 (osx txmit 0x22 [duallink 0x2] enc 0x5)] linkb: true
- 17.
- 18.
19. Connector Object Id [19] which is [DISPLAY\_PORT]
20. encoder obj id [0x20] which is [INTERNAL\_UNIPHY1 (osx txmit 0x11 [duallink 0x1] enc 0x2)] linkb: false
- 21.
- 22.

23. Connector Object Id [12] which is [HDMI\_TYPE\_A]
24. encoder obj id [0x20] which is [INTERNAL\_UNIPHY1 (osx txmit 0x21] [duallink 0x1] enc 0x3)] linkb: true
- 25.
- 26.
27. Connector Object Id [4] which is [DVI\_D]
28. encoder obj id [0x1e] which is [INTERNAL\_UNIPHY (osx txmit 0x10 [duallink 0x0] enc 0x0)] linkb: false

Alles anzeigen

Jetzt haben wir alle relevanten Infos, die wir zum Erstellen eines passenden Frambuffer-Patches brauchen.

Holen wir uns also die originalen Framebufferdaten aus SIERRA, um diese für uns passend zu patchen. Hierzu benötigen wir das PHP-file "ATI\_FrameBuffers\_Sierra\_Edition.php" und ebenfalls das Terminal unter OS X. Dort tippen wir den Befehl "php " ein, ziehen wieder die Datei "ATI\_FrameBuffers\_Sierra\_Edition.php" direkt dahinter ins Terminalfenster und drücken Enter (sollte macOS Euch jetzt auffordern XCODE zu laden, tut dies bitte, denn wir werden es später erneut benötigen).

Ergebnis sieht wie folgt aus (da für uns nur der AMD9150Controller wichtig ist, hier nur dessen Einträge):

Code

1. -----AMD9510Controller.kext-----
2. Exmoor (6) @ 0x107080
3. LVDS, LVDS, DP, DP, DP, DP
4. 020000000001000000010151000000002205020400000000
5. 020000000001000000010261010000001204010300000000
6. 000400000403000000010343000000001102030100000000
7. 000400000001000000010431000000002103050500000000
8. 000400000403000000010523000000001000040200000000
9. 000400000001000000010611000000002001050500000000
- 10.
- 11.
12. Berbice (5) @ 0x107110
13. LVDS, DP, DP, DP, DP
14. 020000000001000039050108000000002001010100000000
15. 000400000001000000010243000000001000020200000000
16. 000400000403000000010313000000002103030300000000
17. 000400000403000000010453000000001102040400000000
18. 000400000403000000010533000000001204050500000000

- 19.
- 20.
21. Baladi (6) @ 0x107300
22. DP, DP, DP, DP, DP, DP
23. 000400000403000000010300000000001204030300000000
24. 000400000403000000010100000000001102010100000000
25. 000400000403000000010200000000002103020200000000
26. 000400000403000000010400000000002205040400000000
27. 000400000403000000010500000000001000050500000000
28. 000400000403000000010600000000002001060600000000

Alles anzeigen

Die von uns genutzte PowerColor RX480 hat insgesamt 5 Anschlüsse, also ist der für uns am besten passende Framebuffer "BERBICE", da dieser ebenfalls über 5 Anschlüsse verfügt. Wir können aber erkennen, das BERBICE für einen internen Anschluss (LVDS) sowie 4 externe Displayport-Anschlüsse ausgelegt ist.

Nun heisst es diesen zu patchen. Dadurch wird dann aus:

Code

1. LVDS, DP, DP, DP, DP
2. 020000000001000039050108000000002001010100000000
3. 000400000001000000010243000000001000020200000000
4. 000400000403000000010313000000002103030300000000
5. 000400000403000000010453000000001102040400000000
6. 000400000403000000010533000000001204050500000000

dieser Code mit Anschlüssen für 3x Displayport, 1x HDMI und 1x DVI-D:

Code

1. DP, DP, DP, HDMI, DVI-D
2. 000400000403000000010243000000001204060100000000
3. 000400000403000000010313000000002205040300000000
4. 000400000403000000010453000000001102010200000000
5. 000800000402000000010533000000002103050400000000
6. 040000001402000000010300000000001000030600000000

Daraus ergibt sich dann für die spätere Verwendung unter CLOVER folgender Patchcode:

original

FB:

020000000001000039050108000000002001010100000000000400000001000000010243000000001000002

patched

FB:

00040000040300000001024300000000**12040601**

00000000000040000040300000001031300000000**22050403**

00000000000040000040300000001045300000000**11020102**

00000000000080000040200000001053300000000**21030504**

0000000004000000140200000001030000000000**10000306**00000000

Soweit, so gut. Jetzt müssen wir noch die entsprechenden AMD-Kexte patchen, damit unsere RX480 Karte überhaupt von den Apple Treibern erkannt wird. Dazu legen wir uns Kopien der nachfolgend genannten Kext-Dateien an:

### **AMD9510Controller.kext** und **AMDRadeonX4100.kext**

Bei beiden müssen wir die interne "Info.plist"-Datei anpassen. Dazu einen Rechtsklick auf die jeweilige Kext-Datei und "Paketinhalt zeigen" auswählen, den Ordner "Contents" öffnen und nun die "Info.plist" per doppelklick öffnen. Wer sich, wie ich gesagt hatte, XCode heruntergeladen und installiert hat, sollte daraufhin folgendes Bild bekommen (am Beispiel der **AMDRadeonX4100.kext** 😞)



Für uns relevant ist hier der Eintrag unter "IOPCIMatch", denn hier müssen wir nun die Device-ID unserer RX480 nachtragen: **0x67DF1002**. Einfach diesen Wert mit einem Leerzeichen davor hinter dem Wert "0x67EF1002" eintragen. Info.plist speichern - fertig.

Das selbe machen wir im Falle der **AMD9510Controller.kext**:



Auch hier tragen wir wieder unsere Device-ID nach: " **0x67DF1002**" direkt hinter "0x67EF1002". Speichern, Fertig.

Jetzt beide gepatchten Kexte mittels Kext Wizard wieder in den Ordner

/System/Library/Extensions laden, den KernelCache refreshen und weiter geht es.

Finaler Schritt: anpassen der CLOVER "**config.plist**" Datei.

Hierzu öffnen wir unsere CLOVER config.plist und suchen uns folgenden Bereich raus:

Code

1. <key>Graphics</key>
2. <dict>
3. <key>DualLink</key>
4. <integer>1</integer>
5. <key>FBName</key>
6. <string>Berbice</string>
7. <key>ig-platform-id</key>
8. <string>0x19120000</string>
9. <key>Inject</key>
10. <dict>
11. <key>ATI</key>
12. <true/>
13. <key>Intel</key>
14. <false/>
15. </dict>
16. <key>InjectEDID</key>
17. <false/>
18. </dict>

Alles anzeigen

Kann bei Euch ein bißchen anders aussehen, da der hier zu sehende Teil aus meiner config.plist stammt (welche bereits für eine RX480/RX580 genutzt wird). Wichtig sind hierbei die Einträge **<key>FBName</key>** und **<key>ATI</key>**. Bei FBNames tragt Ihr darunter den Namen des von Euch gepatchten Framebuffers ein (in unserem Fall eben BERBICE, und für ATI setzt Ihr darunter den Wert auf <true/>, da CLOVER ja das Injecten für AMD Grafikkarten übernehmen soll. Das ist alles, was in dieser Sektion eingestellt werden muss.

Nächster Part für die CLOVER config.plist: der Bereich "**<key>KernelAndKextPatches</key>**" - bei mir sieht dieser so aus:

Code

1. <key>KernelAndKextPatches</key>
2. <dict>
3. <key>ATISConnectorsController</key>
4. <string>9510</string>
5. <key>ATISConnectorsData</key>
6. <string>02000000000100003905010800000000200101010000000000400000001000000010243000
7. <key>ATISConnectorsPatch</key>
8. <string>0004000004030000000010100000000001204060100000000000400000403000000010200000

Hier sind nun folgende 4 Einträge durch die von uns früher ermittelten Werte zu ersetzen:

Code

1. <key>ATISConnectorsData</key>
2. <string></string>
3. <key>ATISConnectorsPatch</key>
4. <string></string>

Hier setzen wir zwischen die beiden "string"-Felder unseren ermittelten Wert des original BERBICE-Framebuffers, in unserem Fall also:

Code

1. <key>ATISConnectorsData</key>
2. <string>02000000000100003905010800000000200101010000000000400000001000000010243000

Das selbe machen wir für den gepatchten Framebuffereintrag, so daß dort dann folgendes steht:

Code

1. <key>ATISConnectorsPatch</key>
2. <string>0004000004030000000010243000000000120406010000000000400000403000000010313000

**FERTIG!** Eigentlich gar nicht so schwer, wenn man weiss, wie es geht 😊

Wenn wir nun unseren Hackintosh durchbooten, sollte SIERRA die RX480 vollständig erkennen und unterstützen.

**ABER:**

jedoch nur solange wir unsere IGPU als primary GFX im BIOS gesetzt haben! Als alleinige und primary Grafikkarte läuft die RX480 **nicht**, bzw wenn nur ohne METAL-Unterstützung, also **ohne** Beschleunigung. **Und bislang ist für dieses Problem noch keine Lösung in Sicht.**

Solange man aber eine Helperkarte hat, welche man als primary GFX deklarieren kann, kann man die RX-Karten in ihrer vollen Pracht auch unter macOS nutzen.

Anbei die von mir in diesem Tutorial erwähnten Scripte zum Auslesen der VBIOS-Daten und der Framebuffer unter SIERRA.

Ist leider ein bißchen länger als gedacht geworden, aber ich dachte mir: einfach nur die fertigen Patche hier reinklatschen kann ja jeder. Vielleicht interessiert es ja den Ein oder Anderen, wie ich auf die benötigten Patche komme - daher dieser Beitrag.

---

### Beitrag von „al6042“ vom 30. April 2017, 11:16

Holla die Waldfee... 🍷

Vielen Dank für diese gelungene und super detaillierte Anleitung.  
Darf ich diese mit deiner Erlaubnis in unser Wiki übertragen?

---

### Beitrag von „Mork vom Ork“ vom 30. April 2017, 11:46

na lego 😊

#### Ergänzung:

wer sich nun fragt, woraus sich das Ergebnis des gepackten Framebuffers ergibt, dem sei auch

dies hier nochmal kurz erklärt:

dazu müssen wir uns nochmal die beiden Ergebnisse der ersten beiden „radeon“-scripte im Detail ansehen. Wir schauen uns hierzu mal nur die Werte für den ersten Displayport aus vorherigem Beispiel an:

Das Script „**radeon\_bios\_decode.sh**“ liefert uns folgendes Ergebnis:

```
Connector at index 0
Type [@offset 40846]: DisplayPort (10)
Encoder [@offset 40850]: INTERNAL_UNIPHY2 (0x21)
i2cid [@offset 40956]: 0x90, OSX senseid: 0x1
HotPlugID: 6
```

Das Script „**redsock\_bios\_decoder.sh**“ liefert uns folgendes Ergebnis:

```
Connector Object Id [19] which is [DISPLAY_PORT]
encoder obj id [0x21] which is [INTERNAL_UNIPHY2 (osx txmit 0x12 [duallink 0x2] enc 0x4)]
linkb: false
```

Für den exakten Patch wichtig sind die Werte der folgenden Angaben (in der Reihenfolge, wie sie im Patch eingepasst werden):

**osx txmit** , **enc**, **HotPlug** und **senseid**

Schauen wir uns die erste Zeile des gepachten Frambuffers an:

**0004**0000**0403**000000001024300000000**12040601**00000000

Wir müssen hier also nur darauf achten, dass wir die einzelnen Werte, die uns beide Scripte liefern, korrekt setzen. Dabei muss darauf geachtet werden, dass der erste Anschluss auch die erste Reihe sein sollte, der zweite Anschluss die zweite Reihe usw.



Die beiden vorderen Werte (in unserem Beispiel „**0004**“ und „**0403**“) stehen für die Art des Anschlusses, sprich ob es sich dabei um einen Displayport-, HDMI-, DVI- oder internen LVDS Anschluss handelt. Dabei gilt folgende Regel:

Displayport ist gleich 0004 und 0403  
HDMI ist gleich 0008 und 0402  
DVI ist gleich 0400 und 1402  
LVDS (intern) ist gleich 0200 und 0001

Bastelt man das alles zusammen, ergibt sich daraus für unser Beispiel eben:

patched FB:

**0004**0000**0403**000000001014300000000**12040601**00000000 ---> Displayport  
**0004**0000**0403**000000001021300000000**22050403**00000000 ---> Displayport  
**0004**0000**0403**000000001035300000000**11020102**00000000 ---> Displayport  
**0008**0000**0402**000000001043300000000**21030504**00000000 ---> HDMI  
**0400**0000**1402**000000001050000000000**10000306**00000000 ---> DVI-D

und so wird daraus dann:

**0004**0000**0403**000000001014300000000**12040601**00000000**0004**0000**0403**  
000000001021300000000**22050403**00000000**0004**0000**0403**000000001035300000000**110201**  
**02**00000000**0008**0000**0402**000000001043300000000**21030504**00000000**0400**0000**1402**  
000000001050000000000**10000306**00000000

---

**Beitrag von „griven“ vom 1. Mai 2017, 22:10**

[@Mork vom Ork](#) danke für dieses ausführlichen und absolut lehrreichen Beitrag.

---

**Beitrag von „al6042“ vom 1. Mai 2017, 22:15**

Habe auch bereits krampfhaft versucht per GPU-Z auf einer Windows10 portable Installation das BIOS meiner R7 M360 des Ideapad 500 zu sichern, um ggf. die Anleitung einsetzen zu können, aber das vermaledeite VBios lässt sich nicht entnehmen...

---

### **Beitrag von „Mork vom Ork“ vom 1. Mai 2017, 22:31**

Das BIOS der R7 M360 sollte sich auch aus einem DOS-Prompt mit ATIFlash heraus sichern lassen. Habe den genauen Befehl nicht im Kopf, boote mal in WINDOWS, rufe ein DOS-Fenster mit "CMD" als Admin auf und wechsel in das Verzeichnis, in dem Du ATIFlash hast.

Die aktuelle Version davon findest Du [hier](#). Tippe dann einfach nur ATIFlash.exe ein und Du solltest eine Liste aller Befehle erhalten. Da ist auch einer dabei, mit dem Du das aktuell geflashte VBIOS sichern kannst. Dieses kannst Du dann wie in meiner Anleitung beschrieben nutzen.

Sach Bescheid, wenn es damit geklappt hat.

---

### **Beitrag von „al6042“ vom 1. Mai 2017, 22:47**

Erstaunlicherweise gibt mir atiflash -i einen "Adapter not found" aus.

EDIT:

Auch die "atiflash -i 0" bis "atiflash -i 9" werfen den Hinweis aus.

---

### **Beitrag von „Mork vom Ork“ vom 1. Mai 2017, 22:57**

Schau mal, ob Du im BIOS "CSM" einschalten kannst. Wenn ja, dann setze es auch auf "Legacy zuerst" - falls Du hier zusätzlich noch wählen kannst, ob er UEFI oder LEGACY zuerst laden soll. Hatte ich beim umflashen meiner RX480 auf das BIOS der RX580 neulich ebenfalls. Er war partout der Meinung, es sei keine AMD-Karte im Rechner. Nach der Umstellung hat er sie dann

brav erkannt

---

### Beitrag von „al6042“ vom 1. Mai 2017, 23:18

CSM habe ich im BIOS des Ideapad nicht,  
"Legacy Support" ist verfügbar, hilft aber nicht.  
Die USB-Platte ist als UEFI eingerichtet.

Nicht vergessen, es handelt sich um eine R7 M360 kombiniert mit der HD520 des i7-6500U.  
Der AtiFlash und ATiWinFlash zeigen nichts an, aber der GPU-Z gibt sie sauber aus.

Ich muss mich hier auch ausklinken... 😊  
Feierabend für heute.

---

### Beitrag von „pawelpipowich“ vom 5. Mai 2017, 17:31

Vielen Dank für die mühe. Ich muss erst mal lernen wie ich das alles umsetzen kann. Bin erstaunt wie gut man in der materie sich auskennen kann. Hut ab.

Der adapter ist erst heute angekommen. Wenn ich zeit finde, mach ich mich ans werk.

zurzeit habe ich ja eine hd7850oc drin und alles läuft fabelhaft. Hoffe das wenn ich [bios update](#) durchführe das auch so bleibt. Falls ich das mit der rx 480 nicht schaffe.

meint ihr die rx 480 wird auch mal oob sein? Oder ist die zu spezifisch?  
Mein gedanke ist so das wenn ich das doch nicht schaffen sollte , werde ich einfach warten bis sie oob im update wird . Wenn die zu spezifisch ist dann fang ich lieber jetzt an die rx 480 zu installieren.

Danke

---

### Beitrag von „griven“ vom 5. Mai 2017, 21:55

Eigentlich sollte die RX480/RX460, wenn man mal von der Notwendigkeit absieht die Device & VendorID in den Kext einzutragen, genau so OOB laufen wie die es die HD7850 macht und das macht sie auch wenn man sie zum Beispiel in einen MacPRO 5,1 steckt. Beim MAC gibt es eben dann erst ein Bild wenn das OS geladen wurde und vorher bleibt der Schirm schwarz. Wenn man mal ein wenig darüber nachdenkt warum das wohl so ist kommt man schnell auf die Idee das es mit dem Bios auf den Karten zusammenhängen muss denn wenn man eine Karte die kein MAC Bios besitzt in einen MAC einbaut kann der MAC auf Bios Ebene damit nichts anfangen wohl aber sobald das OS den Treiber geladen hat sprich hier liegt also der Schluss nahe das der MAC bzw. dessen Firmware die Karte nicht initialisieren kann (logisch denn das EFI eines MAC's unterscheidet sich von dem UEFI eines PCs) mit anderen Worten der MAC kann weder den klassischen vBIOS Teil gebrauchen noch den dahinter liegenden UEFI Teil trotzdem meldet sich die Karte als PCIe Gerät am Bus an. Lax gesprochen hat der MAC also auf Firmware Ebene mit einem DisplayController zu kämpfen für den er keinen Treiber hat (GOP) er bindet also das Gerät mit Device und VendorID in den Gerätebaum ein und sobald sich beim Start des OS ein Treiber an das Gerät bindet gibt es ein Bild 😄

Das Verhalten kann man auf einem PC rekonstruieren indem man der Karte ein modifiziertes Bios verpasst (DeviceID im UEFI Teil des Bios wird ausgetauscht) die Karte meldet sich also als PCIe Device an und auch in dem Fall findet das UEFI des Rechners keine passende Firmware dazu (logisch die ID's im Bios der Karte wurden verbogen) das Ergebnis ist identisch zu dem auf dem MAC sprich black Screen bis das OS den Treiber geladen hat. Demnach muss also das Problem irgendwo in der UEFI Implementation des vBIOS der Karte liegen oder anders ausgedrückt der geladene GOP EFI Treiber der dafür sorgt das die Karte auf UEFI Ebene ein Bild erzeugt verhindert das der OS-X Treiber die Kontrolle über das Gerät übernimmt das Ergebnis ist der bekannte black Screen.

Es gibt hier jetzt natürlich noch den "Trick" über die helper GPU und auch hier wird eigentlich nur vermieden das der GOP Treiber der AMD Karte geladen wird denn das ist ja nicht mehr nötig wenn schon ein Bild ausgegeben wird. Der Verbund IntelHD/AMD ist bei Apple im übrigen gar nicht so ungewöhnlich so gibt es ihn zum Beispiel beim iMAC 15.1 und auch bei diversen

MacBooks ist eine solche Lösung bekannt. Gerade der iMAC ist hier aber ein besonders interessantes Beispiel denn bei ihm ist die iGPU nur im Hintergrund präsent und tritt erst auf den Plan wenn Airplay oder QuickSync benötigt werden.

Ich denke hier ist einfach noch viel Hirnschmalz in der Community gefragt um zu ergründen warum der GOP Treiber nicht sauber entladen werden kann um die Kontrolle an das OS abzugeben letztlich denke ich ist hier Apple mal ausnahmsweise nicht der Sündenbock denn letztlich kann Apple nichts dafür wie das UEFI des Board mit der Karte interagiert oder eben auch nicht...

---

### Beitrag von „ralf.“ vom 6. Mai 2017, 08:35

Ein sehr guter Beitrag 👍 Hilft das ganze besser zu verstehen.

Im anderen Forum hatte jemand eine alte Grafikkarte vom alten VBios auf GOP-Bios umgeflasht. Vielleicht ist es ja möglich z.B. eine R9-380X auf R9-280X oder HD 7870 umzuflaschen (die soll ja angeblich kein GOP-Bios haben), um dann keine Helper-Grafik mehr zu brauchen.

---

### Beitrag von „Mork vom Ork“ vom 6. Mai 2017, 10:01

[Zitat von ralf.](#)

Im anderen Forum hatte jemand eine alte Grafikkarte vom alten VBios auf GOP-Bios umgeflasht. Vielleicht ist es ja möglich z.B. eine R9-380X auf R9-280X oder HD 7870 umzuflaschen (die soll ja angeblich kein GOP-Bios haben), um dann keine Helper-Grafik mehr zu brauchen.

Das waren Florian und ich im TMX86 Forum. Florian war so nett und hat mir ein GOP-freies BIOS für die RX480 gebastelt und ich habe es getestet. Leider brachte auch diese Modifikation keine Besserung.

Ich habe dieses GOP-freie VBIOS noch hier... falls es jemand testen möchte. Aber wie gesagt: bei mir brachte es keine Änderung im Verhalten der RX480.

---

## Beitrag von „Mork vom Ork“ vom 17. Juni 2017, 23:51

Ich habe mich heute nochmal genauer der "ati.c" Datei von CLOVER angenommen, in der die AMD Grafikkarten initialisiert werden. Und ich bin wieder einen Schritt weiter gekommen. Schaut Euch mal meine aktuelle CLOVER Bootlog an (AMD RX480 solo, keine iGPU und CSM disabled im BIOS):

Code

```
1. 40:846      0:000      ===      [      RestSetup      macOS      ]
   =====
2. 40:846 0:000 EdidDiscovered size=128
3. 40:846 0:000 Inject EDID
4. 40:846 0:000 ----- EDID Table
5. 40:846 0:000 000 | 00 FF FF FF FF FF FF 00 1E 6D
6. 40:846 0:000 010 | 09 5B 65 BC 05 00 09 1A 01 04
7. 40:846 0:000 020 | B5 3C 22 78 9F 30 35 A7 55 4E
8. 40:846 0:000 030 | A3 26 0F 50 54 21 08 00 71 40
9. 40:846 0:000 040 | 81 80 81 C0 A9 C0 D1 C0 81 00
10. 40:846 0:000 050 | 01 01 01 01 4D D0 00 A0 F0 70
11. 40:846 0:000 060 | 3E 80 30 20 65 0C 58 54 21 00
12. 40:846 0:000 070 | 00 1A 28 68 00 A0 F0 70 3E 80
13. 40:846 0:000 080 | 08 90 65 0C 58 54 21 00 00 1A
14. 40:846 0:000 090 | 00 00 00 FD 00 30 3D 87 87 38
15. 40:846 0:000 100 | 01 0A 20 20 20 20 20 20 00 00
16. 40:846 0:000 110 | 00 FC 00 4C 47 20 55 6C 74 72
17. 40:846 0:000 120 | 61 20 48 44 0A 20 01 0F
18. 40:846 0:000 VendorID = 0x1E6D
19. 40:846 0:000 ProductID = 0x5B09
20. 40:846 0:000 ----- New EDID Table
21. 40:846 0:000 000 | 00 FF FF FF FF FF FF 00 6D 1E
22. 40:846 0:000 010 | 09 5B 65 BC 05 00 09 1A 01 04
23. 40:846 0:000 020 | B5 3C 22 78 9F 30 35 A7 55 4E
24. 40:846 0:000 030 | A3 26 0F 50 54 21 08 00 71 40
25. 40:846 0:000 040 | 81 80 81 C0 A9 C0 D1 C0 81 00
```

```

26. 40:846 0:000 050 | 01 01 01 01 4D D0 00 A0 F0 70
27. 40:846 0:000 060 | 3E 80 30 20 65 0C 58 54 21 00
28. 40:846 0:000 070 | 00 1A 28 68 00 A0 F0 70 3E 80
29. 40:846 0:000 080 | 08 90 65 0C 58 54 21 00 00 1A
30. 40:846 0:000 090 | 00 00 00 FD 00 30 3D 87 87 38
31. 40:846 0:000 100 | 01 0A 20 20 20 20 20 20 00 00
32. 40:846 0:000 110 | 00 FC 00 4C 47 20 55 6C 74 72
33. 40:846 0:000 120 | 61 20 48 44 0A 20 01 0F
34. 40:846 0:000
35. 40:846 0:000 ..... Initialize RADEON card .....
36. 40:846 0:000
37. 40:846 0:000 Framebuffer @0xC0000000 • MMIO @0xD0000000 • I/O Port
    @0x0000E000 • ROM Addr @0xDFE40000
38. 40:846 0:000 PCI region 1 = 0x00000000, region3 = 0x00000000, region5 =
    0xDFE00000
39. 40:846 0:000 Use region5 as MMIO space
40. 40:846 0:000
41. 40:846 0:000 ... Begin BIOS_SCRATCH ...
42. 40:846 0:000 BIOS_0_SCRATCH=0x00000000, 1=0x00000000, 2=0x00000003,
    3=0x00000000, 4=0x00000000, 5=0x00000000, 6=0x00000000
43. 40:847 0:000 ... End BIOS_SCRATCH ...
44. 40:847 0:000
45. 40:847 0:000 ... Beginn CARD POSTED/NOT POSTED ...
46. 40:848 0:000 RADEON_CRTC2_GEN_CNTL == 0x00000000
47. 40:848 0:000 Card posted because CONFIG_MEMSIZE = 0x2000
48. 40:848 0:000 ATI card POSTed
49. 40:848 0:000 ... End CARD POSTED/NOT POSTED ...
50. 40:848 0:000
51. 40:848 0:000 ... Initializing VRAM ...
52. 40:848 0:000 ATI Chip Family is: Ellesmere
53. 40:848 0:000 R600: get_vram_size returned 0x2000
54. 40:848 0:000 Set VRAM for Ellesmere = 8192MB
55. 40:848 0:000 ATI: get_vram_size returned 0x2000
56. 40:848 0:000 ... VRAM successfully initialized ...
57. 40:848 0:000
58. 40:848 0:000 ... Load VBIOS File ...
59. 40:848 0:000 Found generic VBIOS ROM file (1002_67DF.rom)
60. 40:849 0:001 Loaded ROM len=262144
61. 40:849 0:000 Calculated ROM len=58880
62. 40:849 0:000 VideoBIOS read from file
63. 40:849 0:000 ... VBIOS File successfully loaded ...

```

64. 40:849 0:000  
65. 40:849 0:000 ••• ATI RADEON Chipfamily •••  
66. 40:849 0:000 ATI Radeon EVERGREEN+ family  
67. 40:849 0:000 ••• ATI RADEON Chipfamily •••  
68. 40:849 0:000  
69. 40:849 0:000 ••• Various CONFIG values •••  
70. 40:849 0:000 Device's default Framebuffer: Orinoco  
71. 40:849 0:000 Device's default Nr. of ports: 5  
72. 40:849 0:000 ATI returned Nr. of ports: 5  
73. 40:849 0:000 Value of name\_parent is set to: ATY,OrinocoParent  
74. 40:849 0:000 ••• Various CONFIG values •••  
75. 40:849 0:000  
76. 40:849 0:000 ••• ATI Custom Properties detected •••  
77. 40:849 0:000 Custom Properties Key: name • Custom Properties Value: <4154595F> •  
Custom Properties Value Length: 8  
78. 40:849 0:000 Custom Properties Key: device\_type • Custom Properties Value:  
<6770752D> • Custom Properties Value Length: 15  
79. 40:849 0:000 ••• ATI Custom Properties injected •••  
80. 40:849 0:000  
81. 40:849 0:000 ATI CHIP FAMILY: Ellesmere • CARD NAME: AMD Radeon RX480/580 •  
VRAM: 8192MB • FRAMEBUFFER: Orinoco • VENDOR-/DEVICE-ID: 1002:67DF •  
SUBSYSTEM-ID: 1DA2:E366 • PciRoot(0x0)\Pci(0x1,0x0)\Pci(0x0,0x0)  
82. 40:849 0:000  
83. 40:849 0:000 ••••• Initialize RADEON card •••••  
84. 40:849 0:000  
85. 40:849 0:000 RCBA disabled; cannot use it  
86. 40:850 0:000 stringlength = 3600  
87. 40:850 0:000 CurrentMode: Width=3840 Height=2160  
88. 40:850 0:000 Beginning FSInjection  
89. 40:850 0:000 Blocking kext caches

Alles anzeigen

Wie man sehen kann, wird der VRAM nun an allen Stellen korrekt ausgelesen und gesetzt! Ich muss meine Änderungen an der Datei jetzt nur noch dahingehend prüfen, dass die Änderungen das problemlose Auslesen des VRAMs bei älteren Karten nicht beeinflusst - ich teste das mal mit einer AMD R9 280X und melde mich dann nochmal.



### **Beitrag von „burzlbaum“ vom 3. Juli 2017, 00:06**

Ich hatte unter 10.12.1 die rx480 recht flott mit voller Unterstützung zum Laufen gebracht. Das will mir irgendwie nach dem Update auf 10.12.5 nicht gelingen.

Ich habe die ID der RX480 in den beiden Kexten 4100 und Controller hinzugefügt. Da ich aber wohl ein Framebuffer Problem habe (komisch, dass ich das bei 10.12.1 nicht hatte - habe ich auch den Patch von Mork vom Ork angewandt und kämpfe weiterhin mit dem Problem. Meine 2 Monitore sind via HDMI und DVI angeschlossen und leider habe ich gerade kein DP to HDMI Kabel hier.

Vielleicht hab ich aber auch einfach irgendwas übersehen, könnte mal jemand über meine config sehen?

//edit

Ohne die Framebuffereinträge klappt es dann doch - was ja auch logisch ist. Ich will ja per HDMI ansteuern... Aber wie kann ich jetzt überprüfen ob auch alle 36 Units angesteuert werden?

---

### **Beitrag von „DerJKM“ vom 4. Juli 2017, 21:47**

Es wird nur noch ein paar Tage bis 10.12.6 dauern - das kann aufgrund des neuen iMac 36CUs ohne Patch. Die Mühe mit 10.12.5 würde ich mir nicht mehr machen.