

Erledigt

USB-Mapping Problem USBPort.kext falsch eingebunden

Beitrag von „Wolfe“ vom 22. November 2020, 13:46

Nachtrag: Problem gelöst. USBPorts.kext (mit Hackintool erstellt) war falsch in Open Core 0.6.3. eingebunden, dann kein erfolgreicher Bootvorgang sondern Stillstand beim Apfellogo ohne Ladebalken.

Fehler: ExecutablePath gesetzt, obwohl keiner vorhanden sein durfte, da das item in kernel/add auf eine kext verwies. ExecutablePath-Eintrag gelöscht, Rechner bootet erfolgreich durch.

Liebes Forum,

Ich brauche Hilfe beim USB-Mapping für ein GA-H97M-HD3.

Versucht habe ich Dortania, jeweils intel und manual (zu kompliziert für mich), sowie das Hackintool. Mein Rechner startet mit den jeweiligen Ergebnissen meiner Bemühungen nicht und bleibt beim Apfellogo hängen. Mit USBInjectAll.kext startet der Rechner.

Entweder sind meine mappings fehlerhaft oder sie sind falsch in die config.plist von open core eingebunden, nehme ich an. Was muss ich bei der Einbindung alles beachten? Vorhandensein und Syntax der Items, Vorhandensein der kexte...Puuh.

Ich wäre dankbar, wenn sich jemand diesem Thema noch einmal widmen würde, auch wenn es gefühlt das tausendste Mal ist.

Gruß

Wolfe

Beitrag von „jboeren“ vom 22. November 2020, 13:51

Ich war bei meinem mainboard GA-Z97X-UD3H in sachen USB einfach faul und habe im bios usb3 ausgeschaltet. Es bleiben 14 usb2 anschlüsse die über USBInjectAll.kext alle

funktionieren. Ich habe keine usb3 geräte.

Beitrag von „Wolfe“ vom 22. November 2020, 14:37

jboeren Vielen Dank für den Hinweis, aber ich hätte schon gerne alle USB3-Ports. Noch wichtiger sind mir aber die internen, welche den Sleep-Modus sabotieren.

Beitrag von „jboeren“ vom 22. November 2020, 14:39

sleep-modus gibt es hier nicht... da ich den BigHack als server benutze.

Beitrag von „Wolfe“ vom 23. November 2020, 12:53

An welchen Stellen in der config.plist muss ich aufpassen?

- ACPI: add und patch?
- Kernel: xhciportlimit?
- NVRAM: bootargs?

Die entsprechenden Stellen sind überall in der config.plist verstreut

Könnte jemand von euch Lieben meine EFI anschauen und mir einen Fingerzeig geben? Das wäre super!

Edit: Teilerfolg. Der Rechner bootet nun mit einer USB-Lösung auf Basis einer SSDT durch, aber die USB-Ports sind noch nicht richtig gemapped. Ich zeige Hackintool, wie der Computer welche Ports behandeln soll, exportiere das Ergebnis in eine SSDT.aml, füge die Datei in den Ordner EFI/.../ACPI ein, aber Open Core scheint das nicht zu interessieren. Die Datei ist auch wohl richtig geadded.

Was mache ich falsch?

Beitrag von „LetsGo“ vom 23. November 2020, 16:42

[Wolfe](#)

Die USBPorts.kext hat keinen ExecutablePath. Somit musst du das in deiner config.plist richtig stellen.

Sollte, wie hier aussehen:

Comment	x String	
MacKernel	x String	
PlatPath	x String	ContentsInfo.plist
Enabled	x Boolean	True
MacKernel	x String	
ExecutablePath	x String	
BundlePath	x String	USBPorts.kext

Du wirst wahrscheinlich vorher auch folgende Punkte ausführen müssen: (siehe info.plist von der USBPorts.kext)

```
SMBIOS needing XHC1, EHC1 and EHC2 renames:  
• iMac16,x and older
```

<https://dortania.github.io/Ope...ing-what-renames-you-need>

Beitrag von „Wolfe“ vom 23. November 2020, 17:06

[LetsGo](#) Vielen Dank für den Hinweis!

Es kann sein, dass ich das ursprünglich übersehen hatte. Nach einigen weiteren Versuchen habe ich nun in der aktuellen config.plist nachgeschaut und einen vollständigen ExecutablePath für die USBPorts.kext gefunden.

Zwar funktioniert der Boot nun mit einer SSDT, aber die Ports sind noch nicht korrekt eingerichtet. Ich werde diese Kext also noch einmal ausprobieren!

Beitrag von „LetsGo“ vom 23. November 2020, 17:11

[Wolfe](#)

ich hänge mal meine meine USBPorts.kext an. Ist zwar von einem iMac19,1, aber du wirst erhebliche Unterschiede in der info.plist sehen. Ich habe nur einen XHC Eintrag und keine EHxx Einträge!

Beitrag von „Wolfe“ vom 23. November 2020, 17:53

[LetsGo](#) Die Einträge bei mir unter EHC kommen von meinem H97-Mainboard.

Was habe ich gemacht?

config.plist - XHC und EHC Renames deaktiviert, XHCPortlimit aktiviert, UIAC-SSDT deaktiviert
Neustart

Hackintool - neue USBPorts.kext erstellt - in EFI/.../Kexte kopiert

config.plist - Kernel/Add USBPorts.kext aktiviert (ExecutablePath korrekt)
Neustart

Hoffentlich ist meine Liste oben komplett. Eines noch: ~~nach-Start~~ vor dem Start von OC erscheinen Fehlermeldungen, die zu schnell sind, um sie zu entziffern. Irgendetwas mit "...calculate..."

Boot nur bis Apfellogo, kein Ladebalken sichtbar.

Beitrag von „LetsGo“ vom 23. November 2020, 18:04

Die USBInjectall wird nicht benötigt, wenn du die USBPorts.kext nutzt.

Beitrag von „Wolfe“ vom 23. November 2020, 18:47

[LetsGo](#) Hatte ich vergessen, den zu deaktivieren?

So, nun bin ich noch einmal das Tutorial von Dortania penibel durchgegangen und habe mit USBMap.command eine USBMap.kext erstellt. Sie sieht gut aus, finde ich. Eingebunden in kernel/add, aktiviert, usbinjectall.kext durch USBMap.kext in EFI/OC/Kexte ersetzt.

Die Renames von Xhc und Ehc habe ich noch aktiv, den XhciPortlimit(Entferner) aber wieder deaktiviert.

Bootet nur bis zum Apfellogo.
Ich werde noch verrückt.

Anbei diese vermaledeite EFI.

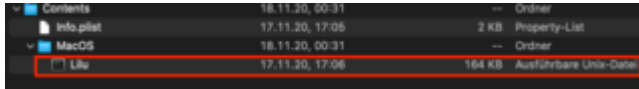
Beitrag von „LetsGo“ vom 23. November 2020, 19:45

[Wolfe](#)

Was mir auf den ersten Blick auffällt, aber ich gucke natürlich noch weiter.

Du hast schon wieder einen executablepath eingetragen. Es gibt aber keinen in deiner USBMap.kext. Öffne mal den Paketinhalt der Kext und du wirst nur eine info.plist finden.

Hier im Vergleich mal eine Kext mit ausführbarem Pfad anhand von Lilu.kext



Vielleicht hilft es auch einen NVRAM Reset zu machen.

Füge doch mal bitte die angehängte SSDT-USBX.aml hinzu. In deiner Kext (info.plist) fehlt nämlich der Teil für die Stromversorgung (USB Power Fix).

Bei mir sieht das so aus. Ich habe meine USBPorts.kext jedoch mit dem Hackintool erstellt. Dadurch wird dann die SSDT-USBX hinfällig.



Du hast außerdem keinen einzigen SSxx (sondern nur HSxx) Port aktiviert. Willst du keinen USB3 Port benutzen?

Beitrag von „Wolfe“ vom 23. November 2020, 20:41

[LetsGo](#) Danke für die Hinweise! Ich verstehe nur etwas nicht...

„schon wieder einen executablepath eingetragen...keiner in der USBMap.kext vorhanden“: ich habe die kext mit usbmap.commander erstellt, und keinen Hinweis bei Dortania gefunden, dass die kext nach der Erstellung noch weiter bearbeitet werden soll.

Nvram-Resets habe ich regelmäßig gemacht. Und ja: usb3-Ports brauche ich wohl, und bin davon ausgegangen, dass Dortania schon das richtige Programm dafür empfohlen hat. Aber wahrscheinlich habe ich da einen Fehler gemacht. Schau ich mir morgen noch einmal an.

Beitrag von „LetsGo“ vom 23. November 2020, 20:45

Damit meine ich, dass du diesen Eintrag (**rot durchgestrichen**) in deiner config.plist löschen musst.



~7	Dictionary	8 key/value pairs
Comment	String	
MaxKernel	String	
PlistPath	String	Contents/info.plist
Enabled	Boolean	True
MinKernel	String	
ExecutablePath	String	Contents/MacOS/USBMap.kext
Arch	String	Any
BundlePath	String	USBMap.kext

Jeder physische USB 3 Port besteht aus einem HS und SS Anteil. In deiner info.plist (die in deinem USBMap.kext enthalten ist) sind nur HS Ports definiert!

Beitrag von „bluebyte“ vom 23. November 2020, 21:08

jhisdgfsfdghkgfhm,ög.fsdajhjsfdjfgkldsa uhffdjfysk<ls,x<y hldfshgaikMDLFBVGÖ N

Beitrag von „Wolfe“ vom 23. November 2020, 21:12

[bluebyte](#) Wenn das schon alles sagt, dann kann ich endlich zugeben, dass ich eigentlich nur jemanden gesucht habe, um ihn tüchtig mit meiner vorgespielten Unwissenheit zu ärgern. Und als Oberesel gehe ich doch gerne über Brücken, meistens goldene. Übrigens noch ein Urlaubstipp: in England freuen sich die Leute sehr, wenn man über sie spricht und sie anwesend sind.

eins, zwei, drei, vier,

Beitrag von „LetsGo“ vom 23. November 2020, 21:16

Denke schon das er seinePorts Mappen muss, falls er eine custom USBPorts.kext nutzen möchte.

Beitrag von „bluebyte“ vom 23. November 2020, 21:17

Frage mich nur nach dem Sinn bei diesem alten Brett.

Beitrag von „Wolfe“ vom 23. November 2020, 21:19

es ist mein altes Brett, und deshalb entscheide ich über den Sinn dieses alten Brettes.

Und jetzt freue ich mich, wenn wir uns wieder mit der übliche Freundlichkeit begegnen, für die ich das hackintosh-forum liebe. Nichts für ungut.

Beitrag von „LetsGo“ vom 23. November 2020, 21:19

Du meinst, weil dies Brett nur insgesamt 12 USB Ports hat und damit das 15 Ports Limit nicht erreicht?

Beitrag von „bluebyte“ vom 23. November 2020, 21:21

Hier versteht wohl keiner mehr Spass, oder?

Beitrag von „Wolfe“ vom 24. November 2020, 01:26

[LetsGo](#) Erfolg! Dank deiner Hilfe funktioniert der Schlafmodus wunderbar. Alle USB-Ports sind jetzt nicht nur sauber gemapped, sondern auch korrekt in Open Core eingebunden. Dein Hinweis in [post#6](#) war entscheidend. Leider habe ich ihn nicht sofort verstanden, da ich mich erst seit wenigen Tagen mit Open Core befasse und mich über jeden Erfolg noch freue wie ein kleines Kind.

Danke! 😊

Beitrag von „LetsGo“ vom 24. November 2020, 01:29

[Wolfe](#)

Freut mich zu hören. Verwendest du noch immer die selbe USBMap.kext oder hast du dir eine Neue erstellt (wegen den HSxx und SSxx Ports)?

PS: Wie gesehen habe, hast du die Prebuild SSDT`s verwendet. Diese könntest du bei Gelegenheit auch noch ersetzen.

Beitrag von „Wolfe“ vom 24. November 2020, 01:47

[LetsGo](#) Ja, ich habe die Ports mit dem Hackintool neu gemapped. Die Prebuild-SSDTs sind noch die letzte Baustelle. Wenn das nicht zu schwierig ist, dann mache ich die auch noch fertig.

Beitrag von „LetsGo“ vom 24. November 2020, 02:08

[Wolfe](#)

Wollte dir ebenfalls Hackintool vorschlagen. Falls du eine WIFI/Bluetooth Karte verwendest (z.B. Fenvi) musst du halt den internen USB Connector, an welchem die Fenvi Karte angeschlossen ist auf internal (entspricht dem Type255) setzen.

Nein SSDT`s erstellen ist absolut nicht schwierig. Musst halt nur die DSDT aus WIN oder Linux dumpen (geht ganz einfach mit SSDTTime)!

Kannst du bei Gelegenheit nochmals deinen EFI Ordner hochladen. Würde mir den nochmals anschauen.

Beitrag von „Wolfe“ vom 26. November 2020, 18:42

[LetsGo](#) Ich brauche noch einmal deine Hilfe. Endlich habe ich ein funktionierendes linux live mit ubuntu, schaffe es aber nicht ssdtttime richtig zu starten. Das Programm scheint nicht ausgeführt zu werden, sondern zeigt mir nur seinen Code, egal ob ich x.py, x.command oder x.bat starte.

Wie geht das?

Beitrag von „LetsGo“ vom 26. November 2020, 18:45

[Wolfe](#)

Hast du das zip runtergeladen oder den SSDTTime Ordner mittels **git clone** erstellt?

Beitrag von „Wolfe“ vom 26. November 2020, 18:46

Ich habe den zip-ordner heruntergeladen

Beitrag von „LetsGo“ vom 26. November 2020, 18:48

[Wolfe](#)

Kann es sein, dass du noch Python installieren musst?

Beitrag von „Wolfe“ vom 26. November 2020, 18:51

Hm, ich habe die neueste Version von ubuntu benutzt, aber vergessen nachzuschauen, welche Version von python installiert ist.

Dasselbe Spiel hatte ich mit Linux MX, auf der python 2,7 und zerquetschte installiert war.

Welche Version wird denn von ssdttime verlangt?

Beitrag von „LetsGo“ vom 26. November 2020, 18:56

Ich habe es damals nur unter WIN benutzt um meine DSDT zu dumpen. Soweit ich mich

erinnern kann, hat es beim Starten von SSDTTime.command das fehlende Python Script von Selbst installiert. Bin mir aber auch nicht sicher.

In der Github Beschreibung steht auch nur etwas von "any recent version of python". Sehr dehnbarer Begriff.



The screenshot shows a GitHub-style 'Instructions' section. It has a heading 'Instructions:' followed by a horizontal line. Below that, it says 'Linux:' and then a bullet point: '• Launch SSDTTime.py with any somewhat recent version of Python from either a terminal window or by running the file normally.'

Code

1. python SSDTTime.py

Der Befehl funktioniert nicht? (Natürlich musst du dafür im SSDTTime Ordner sein)

Beitrag von „Wolfe“ vom 26. November 2020, 18:57

wie bekomme ich python in mein linux live ubuntu auf dem usb-stick?

Beitrag von „LetsGo“ vom 26. November 2020, 19:09

Ich denke, du musst einfach im Terminal in deinen SSDTTime Ordner wechseln und **python SSDTTime.py** ausführen!

Normalerweise sollte der Ubuntu Live Stick schon Python enthalten.

Ist jetzt zwar unter macOS, aber vielleicht hilft es dir ja weiter:

```
Last login: Thu Nov 26 19:01:11 on ttys000
[ ~ ]
cd Github/SSDTime

[ ~/Github/SSDTime on P master ?1 ]
python SSDTTime.py
#####
#                               SSDT Time                               #
#####

Current DSDT:  None

1. FixHPET      - Patch Out IRQ Conflicts
2. FakeEC       - OS-Aware Fake EC
3. FakeEC Laptop - Only Builds Fake EC - Leaves Existing Untouched
4. PluginType  - Sets plugin-type = 1 on First ProcessorObj
5. PMC         - Enables Native NVRAM on True 300-Series Boards
6. AWAC        - Context-Aware AWAC Disable and RTC Fake
7. USB Reset   - Reset USB controllers to allow hardware mapping

D. Select DSDT or origin folder
Q. Quit

Please make a selection: █
```

Beitrag von „Wolfe“ vom 27. November 2020, 18:53

[LetsGo](#) Habs eben mit Erfolg ausprobiert. Das aktuelle ubuntu hat Python 3.8 vorinstalliert und mit Deinem Tip konnte ich ssdtime starten. Danke!

Eben habe ich damit die SSDTs erstellt, in die config.plist eingebunden und erfolgreich getestet.

Heißa!

Du wolltest noch einmal in die EFI schauen? Ich füge sie bei.