

USB Geräte krepieren randomly, macOS läuft munter weiter. Hilft nur reboot.

Beitrag von „Kaffeebecher“ vom 27. September 2024, 16:51

Holerö!

Seit Januar einen neuen Hackintosh zusammengestellt, mit etwas Hilfe aus einem Post beim Nachbarforum von Tony.

Er läuft sehr gut, **abgesehen von einer Sache, die mich wahnsinnig macht!**

Hier ist mein Build ([PC Part Picker für die detaillierten Komponenten](#))

- i5-12600KF
- Gigabyte B760 X DDR4
- macOS Monterey 12.7.5
- OpenCore Version: 1.0

Ich habe alle USB-Ports gemappt und einen USB Kernel daraus gemacht. Ich hänge mal vorsorglich meinen OC Ordner an.

Alle Ports scheinen zu funktionieren, aber dann, **während ich dann in Ableton Live Musik produziere kackt irgendwann aus dem Nichts sämtliche USB Peripherie ab.**

Das erneute Einstecken bringt gar nichts.

Jedes Mal, wenn es passiert ist, war ich in Ableton Live und habe an Musik gearbeitet, bei regulären Browsen oder sonstigen ist das noch nie vorgekommen, soweit ich mich erinnere.

Das Ding ist, macOS friert nicht ein oder fährt herunter, die Musik spielt weiter, alles läuft weiter - nur alle USB-Peripheriegeräte sind tot.

Dies passiert sehr zufällig - manchmal tagelang nicht, manchmal innerhalb von Stunden. Halt nur immer wenn man in Ableton ist. Da ich ein USB MIDI Keyboard dran hab, seh ich da eine

Verbindung. Das sollte aber dann nicht dazu führen, dass sämtliche USB Geräte krepieren. Die einzige Lösung ist, den PC hart auszuschalten (oder von Remote sich per Fernsteuerung einzuloggen und die Maus dann da weiter steuern und runterfahren ect.)

Angeschlossene USB-Geräte:

- USB-Hub von meinem Display-Monitor, an dem eine Apple USB-Tastatur und eine Logitech MX518 USB-Maus angeschlossen sind.
- USB-MIDI-Keyboard (Roland PCR-300) direkt an der Rückseite des PCs angeschlossen.

Dieses Problem tritt unter Windows nicht auf. Das Problem tritt nur im Kontext Ableton Live (oder vllt hohe CPU Last?) auf.

Es ist wirklich frustrierend, und ich habe keine Ahnung, warum dieser Prozess verrückt spielt. Irgendwelche Ideen?

Als es vorhin erst wieder passiert ist, hab ich mich mit anderen Mac mit Bildschirmübertragung Verbunden, um die Systemprotokolle zu überprüfen.

Dann festgestellt, dass **IOUSBHostFamily** irgendwelche komischen Reportings macht, aus denen ich auch nicht wirklich schlauer werde.

Hier mal die Logs:

Code

```
1. 2024-09-27 14:21:26.236387+0200 localhost kernel[0]: (IOUSBHostFamily)
<IOUSBHostFamily`AppleUSBHostPort::terminateDevice(IOUSBHostDevice*,
AppleUSBHostPort::AppleUSBHostPortTerminateDeviceReason)> 018470.092428
HS04@14300000: AppleUSBHostPort::terminateDevice: destroying 0x0bda/5411/0117
(USB2.1 Hub): connect change interrupt
```

und dann ein wenig später gehts richtig ab:

Zitat

Code

1. 2024-09-27 14:21:27.867240+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:936][0x1000005ec]
CompleteInReport:0xe00002eb
- 2.
3. 2024-09-27 14:21:27.867248+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:727][0x1000005ec] Schedule retry
reason:0xe00002eb count:10 deadline:1ms
- 4.
5. 2024-09-27 14:21:27.868347+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:972][0x1000005ec] TimerOccurred retry:10
- 6.
7. 2024-09-27 14:21:27.868362+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:675][0x1000005ec] initInputReport:0xe00002d8
- 8.
9. 2024-09-27 14:21:27.868367+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:727][0x1000005ec] Schedule retry
reason:0xe00002d8 count:9 deadline:50ms
- 10.
11. 2024-09-27 14:21:27.868772+0200 localhost kernel[0]: tcp_timers: tcp_output()
returned 0 with retransmission timer disabled for 5900 > 54316 in state 4, reset
timer to 748
- 12.
13. 2024-09-27 14:21:27.883322+0200 localhost kernel[0]: (IOAudioFamily)
<IOAudioFamily`IOAudioEngineUserClient::performWatchdogOutput(IOAudioClientBufferSet*,
unsigned int)> +
IOAudioEngineUserClient[<private>]::performWatchdogOutput(<private>,
1504926) - (307fc,6cf)
- 14.
15. 2024-09-27 14:21:27.883341+0200 localhost kernel[0]: (IOAudioFamily)
<IOAudioFamily`IOAudioEngineUserClient::performWatchdogOutput(IOAudioClientBufferSet*,
unsigned int)> -
IOAudioEngineUserClient[<private>]::performWatchdogOutput(<private>,
1504926) - (307fc,acf)
- 16.
17. 2024-09-27 14:21:27.907212+0200 localhost kernel[0]: (IOAudioFamily)
<IOAudioFamily`IOAudioEngineUserClient::performWatchdogOutput(IOAudioClientBufferSet*,
unsigned int)> +

```

IOAudioEngineUserClient[<private>]::performWatchdogOutput(<private>,
1504928) - (307fc,acf)
18.
19. 2024-09-27 14:21:27.907233+0200 localhost kernel[0]: (IOAudioFamily)
<IOAudioFamily`IOAudioEngineUserClient::performWatchdogOutput(IOAudioClientBufferSet*,
unsigned int)> -
IOAudioEngineUserClient[<private>]::performWatchdogOutput(<private>,
1504928) - (307fc,ecf)
20.
21. 2024-09-27 14:21:27.976899+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:972][0x1000005ec] TimerOccurred retry:8
22.
23. 2024-09-27 14:21:27.976919+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:675][0x1000005ec] initInputReport:0xe00002d8
24.
25. 2024-09-27 14:21:27.976928+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:727][0x1000005ec] Schedule retry
reason:0xe00002d8 count:7 deadline:50ms
26.
27. 2024-09-27 14:21:27.979797+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:936][0x1000005e4]
CompleteInReport:0xe00002eb
28.
29. 2024-09-27 14:21:27.979803+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:727][0x1000005e4] Schedule retry
reason:0xe00002eb count:10 deadline:1ms
30.
31. 2024-09-27 14:21:27.980140+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:936][0x1000005e9]
CompleteInReport:0xe00002eb
32.
33. 2024-09-27 14:21:27.980146+0200 localhost kernel[0]:
(com.apple.AppleUserHIDDrivers.dext)
[IOUserUSBHostHIDDevice.cpp:727][0x1000005e9] Schedule retry
reason:0xe00002eb count:10 deadline:1ms

```

34.

35. 2024-09-27 14:21:27.981434+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserUSBHostHIDDevice.cpp:972][0x1000005e9] TimerOccurred retry:10

36.

37. 2024-09-27 14:21:27.981448+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserUSBHostHIDDevice.cpp:972][0x1000005e4] TimerOccurred retry:10

38.

39. 2024-09-27 14:21:27.981450+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserUSBHostHIDDevice.cpp:675][0x1000005e9] initInputReport:0xe00002d8

40.

41. 2024-09-27 14:21:27.981454+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserUSBHostHIDDevice.cpp:727][0x1000005e9] Schedule retry
 reason:0xe00002d8 count:9 deadline:50ms

42.

43. 2024-09-27 14:21:27.981460+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserUSBHostHIDDevice.cpp:675][0x1000005e4] initInputReport:0xe00002d8

44.

45. 2024-09-27 14:21:27.981464+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserUSBHostHIDDevice.cpp:727][0x1000005e4] Schedule retry
 reason:0xe00002d8 count:9 deadline:50ms

46.

47. (...)

48.

49. 2024-09-27 14:21:28.316289+0200 localhost kernel[0]: (IOHIDFamily)
 <IOHIDFamily`IOHIDDevice::handleClose(IOService*, unsigned int)> close by
 AppleHIDKeyboardEventDriver 0x10000064e (0x0)

50.

51. 2024-09-27 14:21:28.316301+0200 localhost kernel[0]:
 (com.apple.AppleUserHIDDrivers.dext)
 [IOUserHIDEventService.cpp:349][0x100000648] Close interface: 0x100000635
 0x0

52.

53. 2024-09-27 14:21:28.317799+0200 localhost kernel[0]: (IOUSBHostFamily) <IOUSBHostFamily`AppleUSBHostPort::terminateDevice(IOUSBHostDevice*, AppleUSBHostPort::AppleUSBHostPortTerminateDeviceReason)> 018472.173834 AppleUSB20HubPort@14312000: AppleUSBHostPort::terminateDevice: destroying 0x05ac/0250/0074 (Apple Keyboard): upstream hub is terminating
- 54.
55. 2024-09-27 14:21:28.318020+0200 localhost kernel[0]: (IOUSBHostFamily) <IOUSBHostFamily`AppleUSBHostPort::terminateDevice(IOUSBHostDevice*, AppleUSBHostPort::AppleUSBHostPortTerminateDeviceReason)> 018472.174062 AppleUSB20HubPort@14330000: AppleUSBHostPort::terminateDevice: destroying 0x043e/9a39/0427 (LG Monitor Controls): upstream hub is terminating
- 56.
57. 2024-09-27 14:21:28.318133+0200 localhost kernel[0]: (IOUSBHostFamily) <IOUSBHostFamily`AppleUSBHostPort::terminateDevice(IOUSBHostDevice*, AppleUSBHostPort::AppleUSBHostPortTerminateDeviceReason)> 018472.174175 AppleUSB20HubPort@14310000: AppleUSBHostPort::terminateDevice: destroying 0x05ac/1006/9615 (Keyboard Hub): upstream hub is terminating
- 58.
59. 2024-09-27 14:21:28.318197+0200 localhost kernel[0]: (IOUSBHostFamily) <IOUSBHostFamily`AppleUSBHostPort::terminateDevice(IOUSBHostDevice*, AppleUSBHostPort::AppleUSBHostPortTerminateDeviceReason)> 018472.174239 AppleUSB20HubPort@14320000: AppleUSBHostPort::terminateDevice: destroying 0x046d/c08e/4000 (MX518 Gaming Mouse): upstream hub is terminating

Alles anzeigen

[Weil das sehr viel Text ist, hier ein Pastebin mit dem Rest vom Log:](#)

Beitrag von „KungfuMarek“ vom 27. September 2024, 16:58

HSxx Ports sind USB 2 only. Bei dir sind es USB3 Ports, denke es liegt dadran.

Beitrag von „Kaffeebecher“ vom 27. September 2024, 17:01

Das verstehe ich nicht ganz, sind die USB Maps nicht abwärtskompatibel? Was wäre hier der nächste Schritt?

Beitrag von „KungfuMarek“ vom 27. September 2024, 17:03

Keine Ahnung, ich habe alle Ports folgendermaßen gemappt:

HSxx = USB2 oder Intern

SSxx= Typ C oder USB 3

Nutze auch Ableton, läuft ohne Probleme.

Thunderbolt Interface, USB-Midi Hub und Ableton Push.

Map einfach mal deine HSxx Ports als USB 2, Versuch ist wert.

Beitrag von „Arkturus“ vom 27. September 2024, 18:12

[Zitat von Kaffeebecher](#)

Das verstehe ich nicht ganz, sind die USB Maps nicht abwärtskompatibel? Was wäre hier der nächste Schritt?

Du kannst die config. Plist des USB. Kext bearbeiten und die Ports von 3auf 0 setzen.

Beitrag von „kaneske“ vom 27. September 2024, 18:14

Klemm mal HUB(s) ab, die machen oft Stress, ich persönlich verteufele die Teile

Beitrag von „genex“ vom 27. September 2024, 18:28

Ich sehe du hast die meisten Sachen Richtig und die TopCase 2 mal konfiguriert z.B. aber die mit H am Anfang sind USB2 also solltest du die auch so Deklarieren.

Siehe #2 von [KungfuMarek](#)

Ich würde die kext mit rechteckig Paketinhalte anzeigen öffnen und die .plist in contents mit ProperTree editieren.

Siehe #5 von [Arkturus](#) , ich ziehe aber 1 vor :

Alle HS** mit UsbConnector 1 und nicht 3 das sind alle SS***

Intern geht auch das ist dann 255 und USB C ist 9

Ganz wichtig zum Grundverständnis:

Jeder einzelne USB Port der USB3 kann auch für USB2 Konfiguriert werden wenn du den auch für USB2 Geräte nutzen willst, und hat dafür auch eine extra Adresse an nur einem physikalischen Port.

Wird der USB Anschluss nur für USB2 konfiguriert läuft auch alles daran nur mit USB2 Speed. (zu sehen im Systembericht)

Max 15 Ports zählt alles was Konfiguriert wurde also 7 x USB3 auch mit USB2 sind 14.

Jeder Hub daran zählt nicht, aber Mac ist da wählerisch und zickig.

Auf den Stromverbrauch der angeschlossenen Teile musst du auch achten und da ist ein Hub mit Power oft besser.

Wichtig ist evtl. auch wie in meinem Fall bei dem MSI Board den USB Port für die MainBoard Beleuchtung nicht zu konfigurieren.

Du kannst zumindestens wenn der *.kext für usb von [usbmap](#) erstellt wurde danach den kext mit Rechts Klick Paketinhalt anzeigen und editieren der *.plist schnell anpassen.

ich habe auch zu jedem Port einen "comment" eingefügt damit ich den identifizieren kann.

▼ HS08	‡ Dictionary	3 key/value pairs
Comment	‡ String	Front USB 2 Left
UsbConnector	‡ Number	1
port	‡ Data	<08000000>
> HS09	‡ Dictionary	3 key/value pairs
> HS010	‡ Dictionary	3 key/value pairs
> HS11	‡ Dictionary	3 key/value pairs
> SS01	‡ Dictionary	3 key/value pairs
> SS02	‡ Dictionary	3 key/value pairs
> SS03	‡ Dictionary	3 key/value pairs
> SS04	‡ Dictionary	3 key/value pairs
> SS05	‡ Dictionary	3 key/value pairs
▼ SS06	‡ Dictionary	3 key/value pairs
Comment	‡ String	Front Left USB 3
UsbConnector	‡ Number	3
port	‡ Data	<18000000>

Name	Location ID	Port	Connector	Dev Speed	Device	Comment
HS01	0x14100000	0x01	USB2	Unknown		Back USB 2 Blue Right
HS02	0x14200000	0x02	USB2	480 Mbps	WEB CAM	Back USB 2 Blue Left
HS03	0x14300000	0x03	USB2	Unknown		Back USB 2 Red A
HS04	0x14400000	0x04	USB2	Unknown		Back USB 2 Red C
HS07	0x14500000	0x07	USB2	12 Mbps	BCM920702 Bluetooth 4.0	Front USB 2 Right
HS08	0x14600000	0x08	USB2	Unknown		Front USB 2 Left
HS09	0x14700000	0x09	USB2	Unknown		USB 2 Back Left Top
HS010	0x14800000	0x0A	USB2	12 Mbps	CHERRY Keyboard	USB 2 Back Right Top
SS01	0x14A00000	0x11	USB3	Unknown		Back USB 3 Right
SS02	0x14B00000	0x12	USB3	Unknown		Back USB 3 Left
SS03	0x14C00000	0x13	TypeC+Sw	Unknown		Back USB 3.2 C
SS04	0x14D00000	0x14	USB3	Unknown		Red USB 3.1 B
SS05	0x14E00000	0x17	USB3	Unknown		Front Right USB 3
SS06	0x14F00000	0x18	USB3	Unknown		Front Left USB 3

Beitrag von „KungfuMarek“ vom 27. September 2024, 19:32

Das was [genex](#) schreibt! +1

Amen

Beitrag von „apfelnico“ vom 27. September 2024, 19:37

Einspruch!

Die USB3, die auch USB2 enthalten (HSxx/SSxx) werden GLEICH deklariert (zum Beispiel USB3, oder USB-C). Lediglich einzelne USB2 werden als solche deklariert, oder als intern.

Beitrag von „Kaffeebecher“ vom 28. September 2024, 00:40

Hab mal die Rückseiten Ports (SSxx), obwohl die natürlich alle USB3 können als USB2 deklariert und beobachte mal die Situation. Habe auch den internen USB Port für die RGB Steuerung (HS05) aus der Config rausgenommen (und damit hoffentlich deaktiviert).

Vielen Dank für vielen Nachrichten!

Beitrag von „bluebyte“ vom 28. September 2024, 08:33

[apfelnico](#) das klingt jetzt sehr verwirrend. Der HSS vom USB3 hat doch Connector-Type 0 und nicht 3.

Beitrag von „G.com“ vom 28. September 2024, 09:06

[bluebyte](#) Da muss ich [apfelnico](#) Recht geben. Die zu Grunde liegende Architektur ist USB, nur abwärtskompatibel. Deswegen werde nur reine USB2 Ports als solche deklariert.

Beitrag von „apfelnico“ vom 28. September 2024, 11:04

[bluebyte](#)

Ne, gar nicht verwirrend, ganz einfach. Ein USB3 ist mit seinem USB2-Anteil komplett USB3. Ebenso USB-C und USB-C Type Switch (unterschiedliche Kodierung).

Lediglich ein echter (einzelner) USB2 ist ein solcher, und wenn intern, dann gern so. Interne USB3 gibt es nicht.

Ist übrigens KEIN „Apple-Ding“, sondern allgemeiner ACPI-Standard. Es kann lediglich die „PC-Interpretation“ der Boardhersteller oft nicht zufriedenstellend von macOS gelesen werden (warum auch), daher dieser Zirkus. Die Deklaration dafür steht aber schon in der DSDT oder weiterführende SSDT in der ACPI des Mainboards. Andere Systeme, wofür es gebaut ist, lesen das sauber aus.

Beitrag von „KungfuMarek“ vom 28. September 2024, 13:01

Also wenn ich meine HSxx Ports, als USB3 Mappe, dann geht mein Midi-Hub nicht mehr, mein Elektron Overbridge Hub macht dann nur noch USB 3 und USB 2 Geräte funktionieren gar nicht mehr dran

Ich lass alles beim alten, funktioniert ohne Kopfschmerzen.

Beitrag von „genex“ vom 28. September 2024, 15:21

[Zitat von KungfuMarek](#)

Also wenn ich meine HSxx Ports, als USB3 Mappe, dann geht mein Midi-Hub nicht mehr, mein Elektron Overbridge Hub macht dann nur noch USB 3 und USB 2 Geräte funktionieren gar nicht mehr dran

Ich lass alles beim alten, funktioniert ohne Kopfschmerzen.

So hat das bei mir auch immer hervorragend mit diversen

Audio Interfaces

Midi Interfaces

NVME in USB 3.2x2

Guitar Rig / Cubase

usw. funktioniert also lasse auch ich das so deklariert.

Ich hatte Zero Probleme an USB. Auch einen Wake nach Sleep Kext für usb hatte ich nicht, auch kann ich immer noch den Hacky mit der BT Tastatur wecken.

Macht für mich auch keinen Sinn einen USB2 Stream von einem USB3 Anschluss der ja eine eigene Hardware Adresse hat als USB3 zu deklarieren.

Wie auch immer es funktioniert Perfekt.

Wie [KungfuMarek](#) richtig beschreibt ist immer gut an einem USB3 Hub zu sehen im Systembericht ob beide Streams durchgereicht werden. Dann werden 2 Hubs angezeigt ein USB3 und ein USB2.

In Zwischenzeit funktioniert USB an meinem MBA M2 fast so gut wie an dem Hackintosh.