Erledigt Wacom Intuos S funktioniert nicht nach dem Aufwachen

Beitrag von „mabam“ vom 10. Mai 2019, 00:41

Hallo dort!

Ich betreibe ein Wacom Intuos S an meinem Hack, aber nach längerer Zeit im Sleep-Modus wacht es manchmal nicht auf, das Kontrolllämpchen auf dem Gerät bleibt dann aus. Ich muss dann den USB-Stecker neu einstecken, damit es wieder läuft.

Es scheint ein Wacom-Problem zu sein. Für Windows gibt es Lösungen im Netz. Aber ich weiß nicht, wie ich auf macOS den USB-Port „neu starten“ (also deaktivieren und dann wieder aktivieren) kann. Ich habe die Infos zum Gerät aus der IO Registry gelesen, und würde am liebsten einen Skript schreiben, der nach dem Aufwachen per Launch Agent ausgeführt wird und den USB-Port (oder irgendwas anderes, wodurch das Gerät neu verbunden wird) neu startet.

Weiß jemand einen entsprechenden Terminal-Befehl, mit dem das geht? Oder hat wer einen anderen kreativen Vorschlag?

Hier die Daten aus der IO Registry (Seriennummern entfernt):

Code

```
$ ioreg -w 0 -rn "Intuos S@14900000" -c AppleUSBDevice
+-o Intuos S@14900000 <class AppleUSBDevice, id 0x100002970, registered, matched, active, busy 0 (2 ms), retain 14>
| {
| "sessionID" = 6108317963685
| "iManufacturer" = 1
| "bNumConfigurations" = 1
| "idProduct" = 884
| "bcdDevice" = 263
| "Bus Power Available" = 250
```

10. | "USB Address" = 4
11. | "bMaxPacketSize0" = 64
12. | "iProduct" = 2
13. | "iSerialNumber" = 3
14. | "bDeviceClass" = 0
15. | "Built-In" = No
16. | "locationID" = 344981504
17. | "bDeviceSubClass" = 0
18. | "bcdUSB" = 512
19. | "USB Product Name" = "Intuos S"
20. | "PortNum" = 9
21. | "non-removable" = "no"
22. | "IOCFPlugInTypes" = {"9dc7b780-9ec0-11d4-a54f-000a27052861"="IOUSBFamily.kext/Contents/PlugIns/IOUSBLib.bundle"}
23. | "bDeviceProtocol" = 0
24. | "IOUserClientClass" = "IOUSBDeviceUserClientV2"
25. | "IOPowerManagement" = {"DevicePowerState"=0,"CurrentPowerState"=3,"CapabilityFlags"=65536,"MaxPowerState"=4,"DriverI
26. | "kUSBCurrentConfiguration" = 1
27. | "Device Speed" = 1
28. | "USB Vendor Name" = "Wacom Co.,Ltd."
29. | "idVendor" = 1386
30. | "IOGeneralInterest" = "IOCommand is not serializable"
31. | "USB Serial Number" = "XXXXXXXXXXXXX"
32. | "IOClassNameOverride" = "IOUSBDevice"
33. | }
34. |
35. +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x100002974, !registered, !matched, active, busy 0, retain 5>
36. +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x10000297a, registered, matched, active, busy 0 (0 ms), retain 5>

Alles anzeigen
Code

1. $ ioreg -w 0 -rn "Intuos S@14900000" -c IOUSBHostDevice
2. +-o Intuos S@14900000 <class IOUSBHostDevice, id 0x10000296e, registered, matched, active, busy 0 (611 ms), retain 24>
3. | {
4. | "sessionID" = 6108317963685
5. | "USBSpeed" = 1
6. | "IOServiceLegacyMatchingRegistryID" = 4294977904

7. | "idProduct" = 884
8. | "IOReportLegendPublic" = Yes
9. | "IOPowerManagement" = {"PowerOverrideOn"=Yes,"CapabilityFlags"=32768,"MaxPowerState"=2,"DevicePowerState"=2,"Childr
10. | "bcdDevice" = 263
11. | "bDeviceClass" = 0
12. | "USB Product Name" = "Intuos S"
13. | "AppleUSBAlternateServiceRegistryID" = 4294977904
14. | "locationID" = 344981504
15. | "kUSBSerialNumberString" = "XXXXXXXXXXXXX"
16. | "bDeviceSubClass" = 0
17. | "kUSBCurrentConfiguration" = 1
18. | "IOCFPlugInTypes" = {"9dc7b780-9ec0-11d4-a54f-000a27052861"="IOUSBFamily.kext/Contents/PlugIns/IOUSBLib.bundle"}
19. | "bDeviceProtocol" = 0
20. | "USBPortType" = 0
21. | "USB Vendor Name" = "Wacom Co.,Ltd."
22. | "idVendor" = 1386
23. | "USB Serial Number" = "XXXXXXXXXXXXX"
24. | "IOGeneralInterest" = "IOCommand is not serializable"
25. | "IOReportLegend" = ({"IOReportGroupName"="Power","IOReportChannels"=((5795982523037122560,12886081538)),"IOR Idle Policy")),"IOReportChannelInfo"={"IOReportChannelUnit"=72058126613872640},"IOReportSubGroupN Policy"})
26. | "kUSBVendorString" = "Wacom Co.,Ltd."
27. | "IOClassNameOverride" = "IOUSBDevice"
28. | }
29. |
30. +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x100002971, !registered, !matched, active, busy 0, retain 8>
31. +-o AppleUSBHostCompositeDevice <class AppleUSBHostCompositeDevice, id 0x100002978, !registered, !matched, active, busy 0, retain 4>
32. +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x100002979, registered, matched, active, busy 0 (6 ms), retain 11>
33. +-o IOUSBHostHIDDevice@14900000,0 <class IOUSBHostHIDDevice, id 0x10000297c, registered, matched, active, busy 0 (1 ms), retain 10>
34. +-o IOHIDInterface <class IOHIDInterface, id 0x10000297e, registered, matched, active, busy 0 (1 ms), retain 6>
35. +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x10000297f, !registered, !matched, active, busy 0, retain 6>

36. +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x100002980, !registered, !matched, active, busy 0, retain 6>

Alles anzeigen

Beitrag von „kuckkuck“ vom 10. Mai 2019, 04:26

Könnte auch ein Hacky Problem sein, lädt AppleBusPowerController laut kextstat oder IOReg?

Beitrag von „mabam“ vom 10. Mai 2019, 13:30

Der lädt wohl:

Code

1. $ kextstat | grep AppleBusPowerController
2. 22 1 0xffffff7f80f1d000 0x8000 0x8000 com.apple.driver.AppleBusPowerController (1.0) 79B78C02-CBE8-3C9B-A19D-D8AB5875A429 <13 11 6 5 4 3>

Beitrag von „kuckkuck“ vom 10. Mai 2019, 17:08

Welches SMBios benutzt du und für welches Gerät lädt der Treiber laut IOReg?

Beitrag von „mabam“ vom 10. Mai 2019, 20:48

Ich benutze iMac18,1.

Über `ioreg -w 0 | grep AppleBusPowerController` bekomme ich gar keine Ausgabe. `ioreg -w 0 | grep USB` gibt mir das:

Code

1. $ ioreg -w 0 | grep USB
2. | | | +-o XHC@14000000 <class AppleIntelCNLUSBXHCI, id 0x1000002af, registered, matched, active, busy 0 (5258 ms), retain 136>
3. | | | +-o HS01@14100000 <class AppleUSB20XHCIPort, id 0x1000002cc, registered, matched, active, busy 0 (0 ms), retain 10>
4. | | | +-o HS02@14200000 <class AppleUSB20XHCIPort, id 0x1000002cd, registered, matched, active, busy 0 (0 ms), retain 10>
5. | | | +-o HS03@14300000 <class AppleUSB20XHCIPort, id 0x1000002ce, registered, matched, active, busy 0 (0 ms), retain 10>
6. | | | +-o HS04@14400000 <class AppleUSB20XHCIPort, id 0x1000002cf, registered, matched, active, busy 0 (0 ms), retain 10>
7. | | | +-o HS05@14500000 <class AppleUSB20XHCIPort, id 0x1000002d0, registered, matched, active, busy 0 (0 ms), retain 10>
8. | | | +-o HS06@14600000 <class AppleUSB20XHCIPort, id 0x1000002d1, registered, matched, active, busy 0 (0 ms), retain 10>
9. | | | +-o HS07@14700000 <class AppleUSB20XHCIPort, id 0x1000002d2, registered, matched, active, busy 0 (0 ms), retain 10>
10. | | | +-o HS08@14800000 <class AppleUSB20XHCIPort, id 0x1000002d3, registered, matched, active, busy 0 (0 ms), retain 10>
11. | | | +-o HS09@14900000 <class AppleUSB20XHCIPort, id 0x1000002d4, registered, matched, active, busy 0 (5187 ms), retain 14>
12. | | | | +-o Intuos S@14900000 <class IOUSBHostDevice, id 0x100000317, registered, matched, active, busy 0 (5187 ms), retain 24>
13. | | | | +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x10000031d, !registered, !matched, active, busy 0, retain 8>
14. | | | | +-o AppleUSBHostCompositeDevice <class AppleUSBHostCompositeDevice, id 0x100000321, !registered, !matched, active, busy 0, retain 4>
15. | | | +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x100000322, registered, matched, active, busy 0 (5176 ms), retain 11>
16. | | | | +-o IOUSBHostHIDDevice@14900000,0 <class IOUSBHostHIDDevice, id 0x100000367, registered, matched, active, busy 0 (5157 ms), retain 10>
17. | | | +-o HS10@14a00000 <class AppleUSB20XHCIPort, id 0x1000002d5, registered, matched, active, busy 0 (56 ms), retain 14>
18. | | | | +-o macally@14a00000 <class IOUSBHostDevice, id 0x100000325, registered, matched, active, busy 0 (56 ms), retain 30>

19. | | | | +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x100000328, !registered, !matched, active, busy 0, retain 8>
20. | | | | +-o AppleUSB20Hub@14a00000 <class AppleUSB20Hub, id 0x10000032b, registered, matched, active, busy 0 (44 ms), retain 19>
21. | | | | | +-o AppleUSB20HubPort@14a10000 <class AppleUSB20HubPort, id 0x10000032e, registered, matched, active, busy 0 (44 ms), retain 16>
22. | | | | | | +-o macally@14a10000 <class IOUSBHostDevice, id 0x100000334, registered, matched, active, busy 0 (44 ms), retain 24>
23. | | | | | | +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x100000337, !registered, !matched, active, busy 0, retain 8>
24. | | | | | | +-o AppleUSBHostCompositeDevice <class AppleUSBHostCompositeDevice, id 0x10000033b, !registered, !matched, active, busy 0, retain 4>
25. | | | | | | +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x10000033c, registered, matched, active, busy 0 (28 ms), retain 11>
26. | | | | | | +-o IOUSBHostHIDDevice@14a10000,0 <class IOUSBHostHIDDevice, id 0x10000033e, registered, matched, active, busy 0 (20 ms), retain 12>
27. | | | | | +-o AppleUSB20HubPort@14a20000 <class AppleUSB20HubPort, id 0x10000032f, registered, matched, active, busy 0 (0 ms), retain 12>
28. | | | | | +-o AppleUSB20HubPort@14a30000 <class AppleUSB20HubPort, id 0x100000330, registered, matched, active, busy 0 (0 ms), retain 12>
29. | | | | +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x10000032c, !registered, !matched, active, busy 0, retain 9>
30. | | | +-o HS11@14b00000 <class AppleUSB20XHCIPort, id 0x1000002d6, registered, matched, active, busy 0 (0 ms), retain 10>
31. | | | +-o SS01@14c00000 <class AppleUSB30XHCIPort, id 0x1000002e3, registered, matched, active, busy 0 (0 ms), retain 10>
32. | | | +-o SS02@14d00000 <class AppleUSB30XHCIPort, id 0x1000002e4, registered, matched, active, busy 0 (0 ms), retain 10>
33. | | | +-o SS03@14e00000 <class AppleUSB30XHCIPort, id 0x1000002e5, registered, matched, active, busy 0 (0 ms), retain 10>
34. | | | +-o SS04@14f00000 <class AppleUSB30XHCIPort, id 0x1000002e6, registered, matched, active, busy 0 (0 ms), retain 10>
35. | | | +-o SS05@15000000 <class AppleUSB30XHCIPort, id 0x1000002e7, registered, matched, active, busy 0 (0 ms), retain 10>
36. | | | +-o SS06@15100000 <class AppleUSB30XHCIPort, id 0x1000002e8, registered, matched, active, busy 0 (0 ms), retain 10>
37. | | | +-o SS07@15200000 <class AppleUSB30XHCIPort, id 0x1000002e9, registered, matched, active, busy 0 (0 ms), retain 10>
38. | | | +-o SS08@15300000 <class AppleUSB30XHCIPort, id 0x1000002ea, registered, matched, active, busy 0 (0 ms), retain 10>

39. +-o USBInjectAll_config <class USBInjectAll_config, id 0x100000122, registered, matched, active, busy 0 (0 ms), retain 5>
40. +-o AppleUSBHostResources <class AppleUSBHostResources, id 0x1000002a0, registered, matched, active, busy 0 (26 ms), retain 33>
41. | +-o AppleUSBLegacyRoot <class AppleUSBLegacyRoot, id 0x1000002c6, registered, matched, active, busy 0 (1 ms), retain 16>
42. | | +-o AppleUSBXHCI@14000000 <class AppleUSBController, id 0x100000319, registered, matched, active, busy 0 (1 ms), retain 13>
43. | | +-o AppleUSBXHCI Root Hub Simulation@14000000 <class AppleUSBRootHubDevice, id 0x10000031a, registered, matched, active, busy 0 (0 ms), retain 13>
44. | | | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x10000057a, !registered, !matched, active, busy 0, retain 5>
45. | | +-o Intuos S@14900000 <class AppleUSBDevice, id 0x10000031c, registered, matched, active, busy 0 (0 ms), retain 14>
46. | | | +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x100000323, registered, matched, active, busy 0 (0 ms), retain 5>
47. | | | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x100000596, !registered, !matched, active, busy 0, retain 5>
48. | | +-o macally@14a00000 <class AppleUSBDevice, id 0x100000327, registered, matched, active, busy 0 (0 ms), retain 15>
49. | | | +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x10000032d, !registered, !matched, active, busy 0 (0 ms), retain 4>
50. | | | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x100000597, !registered, !matched, active, busy 0, retain 5>
51. | | +-o macally@14a10000 <class AppleUSBDevice, id 0x100000336, registered, matched, active, busy 0 (0 ms), retain 14>
52. | | +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x10000033d, registered, matched, active, busy 0 (0 ms), retain 5>
53. | | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x100000598, !registered, !matched, active, busy 0, retain 5>
54. | +-o AppleUSBHostPacketFilterService <class AppleUSBHostPacketFilterService, id 0x1000002c8, !registered, !matched, active, busy 0, retain 4>
55. G3:~ markus$

Alles anzeigen

Wie heißt der AppleBusPowerController in IOReg? Wonach genau soll ich weiter filtern?

Beitrag von „kuckkuck“ vom 11. Mai 2019, 04:54

Genau so heißt der Treiber in IOReg

Hast du ein EC und USBX Device im IOReg?

Beitrag von „mabam“ vom 11. Mai 2019, 13:38

Mit

ioreg -w 0 | grep EC lande ich beim AppleIntelMEClientController, aber den wirst du wohl nicht meinen.

ioreg -w 0 | grep USBX wirft das aus:

Code

1. | | | +-o XHC@14000000 <class AppleIntelCNLUSBXHCI, id 0x1000002b4, registered, matched, active, busy 0 (100 ms), retain 136>
2. | | +-o AppleUSBXHCI@14000000 <class AppleUSBController, id 0x100000319, registered, matched, active, busy 0 (1 ms), retain 13>
3. | | +-o AppleUSBXHCI Root Hub Simulation@14000000 <class AppleUSBRootHubDevice, id 0x10000031a, registered, matched, active, busy 0 (0 ms), retain 13>

Das führt mich zu diesen anderen Eingaben:

1) `$ ioreg -w 0 -rc AppleIntelCNLUSBXHCI` (da ist der Intuos S unter der Zeile `+-o HS09@14900000 <class AppleUSB20XHCIPort, id 0x1000002d4, registered, matched, active, busy 0 (40 ms), retain 14>` zu finden):

Code

1. +-o XHC@14000000 <class AppleIntelCNLUSBXHCI, id 0x1000002b4, registered, matched, active, busy 0 (100 ms), retain 136>
2. | {
3. | "IOClass" = "AppleIntelCNLUSBXHCI"
4. | "CFBundleIdentifier" = "com.apple.driver.usb.AppleUSBXHCIPCI"

5. | "IOProviderClass" = "IOPCIDevice"
6. | "kUSBSleepSupported" = Yes
7. | "IOPowerManagement" = {"ChildrenPowerState"=3,"DevicePowerState"=0,"CurrentPowerState"=3,"CapabilityFlags"=32768,"Ma
8. | "IOPCITunnelCompatible" = Yes
9. | "Revision" = <0103>
10. | "IOProbeScore" = 1000
11. | "port-count" = <1a000000>
12. | "locationID" = 335544320
13. | "IOMatchCategory" = "IODefaultMatchCategory"
14. | "ports" = {"HS01"={"UsbConnector"=3,"port"=<01000000>},"HS05"={"UsbConnector"=3,"port"=<05000000>
15. | "IOGeneralInterest" = "IOCommand is not serializable"
16. | "controller-statistics" = {"kControllerStatIOCount"=75718,"kControllerStatPowerStateTime"={"kPowerStateOff"="0ms (0%)","kPowerStateSleep"="0ms (0%)","kPowerStateOn"="320716ms (98%)","kPowerStateSuspended"="4172ms (1%)"},"kControllerStatSpuriousInterruptCount"=0}
17. | "IOPCIPrimaryMatch" = "0x9ded8086 0xa36d8086"
18. | "device-properties" = {"AAPL,current-in-sleep"=<e803>,"AAPL,current-extra"=<bc02>,"acpi-path"="IOACPIPlane:/_SB/PCI0@0/XHC@140000","built-in"=<00>,"AAPL,clock-id"=<00>,"device_type"=<"XHCI">,"acpi-device"="IOACPIPlatformDevice is not serializable","AAPL,current-available"=<b004>}
19. | "RM,USBInjectAll" = Yes
20. | "64bit" = Yes
21. | "name" = <"XHC">
22. | }
23. |
24. +-o HS01@14100000 <class AppleUSB20XHCIPort, id 0x1000002cc, registered, matched, active, busy 0 (0 ms), retain 10>
25. +-o HS02@14200000 <class AppleUSB20XHCIPort, id 0x1000002cd, registered, matched, active, busy 0 (0 ms), retain 10>
26. +-o HS03@14300000 <class AppleUSB20XHCIPort, id 0x1000002ce, registered, matched, active, busy 0 (0 ms), retain 10>
27. +-o HS04@14400000 <class AppleUSB20XHCIPort, id 0x1000002cf, registered, matched, active, busy 0 (0 ms), retain 10>
28. +-o HS05@14500000 <class AppleUSB20XHCIPort, id 0x1000002d0, registered, matched, active, busy 0 (0 ms), retain 10>
29. +-o HS06@14600000 <class AppleUSB20XHCIPort, id 0x1000002d1, registered, matched, active, busy 0 (0 ms), retain 10>

30. +-o HS07@14700000 <class AppleUSB20XHCIPort, id 0x1000002d2, registered, matched, active, busy 0 (0 ms), retain 10>
31. +-o HS08@14800000 <class AppleUSB20XHCIPort, id 0x1000002d3, registered, matched, active, busy 0 (0 ms), retain 10>
32. +-o HS09@14900000 <class AppleUSB20XHCIPort, id 0x1000002d4, registered, matched, active, busy 0 (40 ms), retain 14>
33. | +-o Intuos S@14900000 <class IOUSBHostDevice, id 0x100000317, registered, matched, active, busy 0 (40 ms), retain 24>
34. | +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x10000031d, !registered, !matched, active, busy 0, retain 8>
35. | +-o AppleUSBHostCompositeDevice <class AppleUSBHostCompositeDevice, id 0x100000321, !registered, !matched, active, busy 0, retain 4>
36. | +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x100000322, registered, matched, active, busy 0 (28 ms), retain 11>
37. | +-o IOUSBHostHIDDevice@14900000,0 <class IOUSBHostHIDDevice, id 0x10000036c, registered, matched, active, busy 0 (10 ms), retain 10>
38. | +-o IOHIDInterface <class IOHIDInterface, id 0x10000037b, registered, matched, active, busy 0 (6 ms), retain 7>
39. | | +-o IOHIDEventDriver <class IOHIDEventDriver, id 0x100000387, registered, matched, active, busy 0 (0 ms), retain 8>
40. | | +-o IOHIDEventServiceUserClient <class IOHIDEventServiceUserClient, id 0x10000038a, !registered, !matched, active, busy 0, retain 6>
41. | +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x100000422, !registered, !matched, active, busy 0, retain 6>
42. | +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x10000059a, !registered, !matched, active, busy 0, retain 6>
43. +-o HS10@14a00000 <class AppleUSB20XHCIPort, id 0x1000002d5, registered, matched, active, busy 0 (38 ms), retain 14>
44. | +-o macally@14a00000 <class IOUSBHostDevice, id 0x100000325, registered, matched, active, busy 0 (38 ms), retain 30>
45. | +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x100000328, !registered, !matched, active, busy 0, retain 8>
46. | +-o AppleUSB20Hub@14a00000 <class AppleUSB20Hub, id 0x10000032b, registered, matched, active, busy 0 (26 ms), retain 19>
47. | | +-o AppleUSB20HubPort@14a10000 <class AppleUSB20HubPort, id 0x10000032e, registered, matched, active, busy 0 (26 ms), retain 16>
48. | | | +-o macally@14a10000 <class IOUSBHostDevice, id 0x100000334, registered, matched, active, busy 0 (26 ms), retain 24>
49. | | | +-o AppleUSBHostLegacyClient <class AppleUSBHostLegacyClient, id 0x100000337, !registered, !matched, active, busy 0, retain 8>

50. | | | +-o AppleUSBHostCompositeDevice <class AppleUSBHostCompositeDevice, id 0x10000033b, !registered, !matched, active, busy 0, retain 4>
51. | | | +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x10000033c, registered, matched, active, busy 0 (4 ms), retain 11>
52. | | | +-o IOUSBHostHIDDevice@14a10000,0 <class IOUSBHostHIDDevice, id 0x10000033e, registered, matched, active, busy 0 (1 ms), retain 12>
53. | | | +-o IOHIDInterface <class IOHIDInterface, id 0x100000340, registered, matched, active, busy 0 (1 ms), retain 7>
54. | | | | +-o IOHIDEventDriver <class IOHIDEventDriver, id 0x100000341, registered, matched, active, busy 0 (0 ms), retain 8>
55. | | | | +-o IOHIDEventServiceUserClient <class IOHIDEventServiceUserClient, id 0x10000035f, !registered, !matched, active, busy 0, retain 6>
56. | | | +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x100000358, !registered, !matched, active, busy 0, retain 6>
57. | | | +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x100000423, !registered, !matched, active, busy 0, retain 6>
58. | | | +-o IOHIDLibUserClient <class IOHIDLibUserClient, id 0x100000580, !registered, !matched, active, busy 0, retain 6>
59. | | +-o AppleUSB20HubPort@14a20000 <class AppleUSB20HubPort, id 0x10000032f, registered, matched, active, busy 0 (0 ms), retain 12>
60. | | +-o AppleUSB20HubPort@14a30000 <class AppleUSB20HubPort, id 0x100000330, registered, matched, active, busy 0 (0 ms), retain 12>
61. | +-o IOUSBHostInterface@0 <class IOUSBHostInterface, id 0x10000032c, !registered, !matched, active, busy 0, retain 9>
62. +-o HS11@14b00000 <class AppleUSB20XHCIPort, id 0x1000002d6, registered, matched, active, busy 0 (0 ms), retain 10>
63. +-o SS01@14c00000 <class AppleUSB30XHCIPort, id 0x1000002d7, registered, matched, active, busy 0 (0 ms), retain 10>
64. +-o SS02@14d00000 <class AppleUSB30XHCIPort, id 0x1000002d8, registered, matched, active, busy 0 (0 ms), retain 10>
65. +-o SS03@14e00000 <class AppleUSB30XHCIPort, id 0x1000002d9, registered, matched, active, busy 0 (0 ms), retain 10>
66. +-o SS04@14f00000 <class AppleUSB30XHCIPort, id 0x1000002da, registered, matched, active, busy 0 (0 ms), retain 10>
67. +-o SS05@15000000 <class AppleUSB30XHCIPort, id 0x1000002db, registered, matched, active, busy 0 (0 ms), retain 10>
68. +-o SS06@15100000 <class AppleUSB30XHCIPort, id 0x1000002dc, registered, matched, active, busy 0 (0 ms), retain 10>
69. +-o SS07@15200000 <class AppleUSB30XHCIPort, id 0x1000002dd, registered, matched, active, busy 0 (0 ms), retain 10>

70. +-o SS08@15300000 <class AppleUSB30XHCIPort, id 0x1000002de, registered, matched, active, busy 0 (0 ms), retain 10>

Alles anzeigen

...

~~EDIT:~~

~~Ich musste die Antwort splitten, weil sie mehr als 10.000 Zeichen hatte. Aber wenn ich den Rest in einer zweiten Antwort posten will, bekomme ich die Meldung:~~

~~„Die letzte Antwort in diesem Thema stammt bereits von dir, du kannst erst in 86.365 Sekunden erneut auf dieses Thema antworten.“~~

~~Kann da mal wer einen Post mit irgend einer kurzen Zeile dranhängen, damit ich den Rest noch posten kann? Danke!~~

EDIT2:

Hab’ das jetzt über einen zweiten Account selbst geregelt – siehe nächster Post. Ich hoffe, das ist für den Zweck okay.

Beitrag von „mabam2“ vom 11. Mai 2019, 16:06

...

2) `$ ioreg -w 0 -rc AppleUSBController` (da ist auch Verschiedenes zum Wacom bzw. Intuos S zu finden):

Code

1. +-o AppleUSBXHCI@14000000 <class AppleUSBController, id 0x100000319, registered, matched, active, busy 0 (1 ms), retain 13>
2. | {
3. | "locationID" = 335544320
4. | }

5. |
6. +-o AppleUSBXHCI Root Hub Simulation@14000000 <class AppleUSBRootHubDevice, id 0x10000031a, registered, matched, active, busy 0 (0 ms), retain 13>
7. | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x10000057f, !registered, !matched, active, busy 0, retain 5>
8. +-o Intuos S@14900000 <class AppleUSBDevice, id 0x10000031c, registered, matched, active, busy 0 (0 ms), retain 14>
9. | +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x100000323, registered, matched, active, busy 0 (0 ms), retain 5>
10. | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x10000058f, !registered, !matched, active, busy 0, retain 5>
11. +-o macally@14a00000 <class AppleUSBDevice, id 0x100000327, registered, matched, active, busy 0 (0 ms), retain 15>
12. | +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x10000032d, !registered, !matched, active, busy 0 (0 ms), retain 4>
13. | +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x100000590, !registered, !matched, active, busy 0, retain 5>
14. +-o macally@14a10000 <class AppleUSBDevice, id 0x100000336, registered, matched, active, busy 0 (0 ms), retain 14>
15. +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x10000033d, registered, matched, active, busy 0 (0 ms), retain 5>
16. +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x100000591, !registered, !matched, active, busy 0, retain 5>

Alles anzeigen

3) `$ ioreg -w 0 -rc AppleUSBRootHubDevice` (da wird das WacomTabletDrive ganz unten genannt):

Code

1. +-o AppleUSBXHCI Root Hub Simulation@14000000 <class AppleUSBRootHubDevice, id 0x10000031a, registered, matched, active, busy 0 (0 ms), retain 13>
2. | {
3. | "iManufacturer" = 0
4. | "bNumConfigurations" = 1
5. | "idProduct" = 32775
6. | "bMaxPacketSize0" = 8
7. | "Built-In" = Yes
8. | "iProduct" = 0
9. | "USB Product Name" = "AppleUSBXHCI Root Hub Simulation"

10. | "iSerialNumber" = 0
11. | "bDeviceClass" = 9
12. | "IOPowerManagement" = {"DevicePowerState"=0,"CurrentPowerState"=4,"CapabilityFlags"=32768,"MaxPowerState"=4,"DriverI
13. | "IOUserClientClass" = "IOUSBDeviceUserClientV2"
14. | "locationID" = 335544320
15. | "bDeviceSubClass" = 255
16. | "bcdUSB" = 256
17. | "non-removable" = "yes"
18. | "IOCFPlugInTypes" = {"9dc7b780-9ec0-11d4-a54f-000a27052861"="IOUSBFamily.kext/Contents/PlugIns/IOUSBLib.bundle"}
19. | "bDeviceProtocol" = 3
20. | "USB Vendor Name" = "Apple Inc."
21. | "Device Speed" = 3
22. | "idVendor" = 1452
23. | "IOClassNameOverride" = "IOUSBRootHubDevice"
24. | }
25. |
26. +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x10000057f, !registered, !matched, active, busy 0, retain 5>

Alles anzeigen

Könnte irgendwas davon zielführend sein?

Beitrag von „kuckkuck“ vom 14. Mai 2019, 16:29

Benutz doch lieber den IORegistry Explorer, das ist wesentlich übersichtlicher für dich.

Im IOReg kannst du dann ausschließlich nach dem Gerät EC suchen. Scheint mir jedoch nicht so, als würde ein Gerät mit dem Namen EC und eins als USBX existieren, dementsprechend solltest du die SSDT aus diesem WIKI Beitrag nutzen und danach das Resultat überprüfen: 2.07 USB Port Lösungen

---

Beitrag von „mabam“ vom 20. Mai 2019, 15:28

Danke für den Tipp, ich habe das SSDT-Add-on jetzt installiert.

Wenn ich nach USBX suche, bekomme ich auch nichts anderes als das, was ich am 11. Mai gepostet habe.

Aber ein Gerät „EC“ gibt es jetzt:

Code

1. +-o EC <class IOACPIPlatformDevice, id 0x1000001f8, registered, matched, active, busy 0 (12 ms), retain 9>
2. | {
3. | "name" = <"EC000000">
4. | "_STA" = 18446744073709551615
5. | }
6. |
7. +-o AppleBusPowerController <class AppleBusPowerController, id 0x100000292, registered, matched, active, busy 0 (0 ms), retain 6>

Code

1. $ ioreg -w 0 -rc AppleBusPowerController
2. +-o AppleBusPowerController <class AppleBusPowerController, id 0x100000292, registered, matched, active, busy 0 (0 ms), retain 6>
3. {
4. "IOClass" = "AppleBusPowerController"
5. "CFBundleIdentifier" = "com.apple.driver.AppleBusPowerController"
6. "IOProviderClass" = "IOACPIPlatformDevice"
7. "kUSBSleepPortCurrentLimit" = 2100
8. "kUSBWakePowerSupply" = 5100
9. "kUSBWakePortCurrentLimit" = 2100
10. "IOProbeScore" = 0
11. "IONameMatch" = "EC"
12. "IOGeneralInterest" = "IOCommand is not serializable"
13. "IOMatchCategory" = "AppleBusPowerController"
14. "kUSBSleepPowerSupply" = 5100
15. "UsbBusCurrentPoolID" = 4294967954
16. "IONameMatched" = "EC"
17. }

Alles anzeigen

Das heißt, ich lasse den Rechner jetzt nochmal länger in Sleep und schaue, ob das Tablet danach ohne Nachhelfen funktioniert?

Beitrag von „kuckkuck“ vom 21. Mai 2019, 08:18

Klingt gut, probier das mal 

Beitrag von „mabam“ vom 21. Mai 2019, 22:42

Nein, es hat sich leider nichts geändert: Nach längerem Sleep ist das Wacom aus und ich muss dessen USB-Stecker aus- und einstöpseln, damit es wieder funktioniert.

Kann ich den zugehörigen USB-Port (oder notfalls alle USB-Ports) irgendwie softwaremäßig neu starten? Denn das ließe sich über SleepWatcher automatisieren.

Beitrag von „kuckkuck“ vom 22. Mai 2019, 01:19

Es gibt Tools, die alle USB Ports vor dem Sleep auswerfen. Wenn die Ports danach korrekt renitialisiert werden sollte ja alles passen.

Beitrag von „mabam“ vom 24. Mai 2019, 16:00

> Zitat von kuckkuck
>
> Es gibt Tools, die alle USB Ports vor dem Sleep auswerfen. Wenn die Ports danach korrekt renitialisiert werden sollte ja alles passen.

Danke für den Tipp. Nachdem ich mich inzwischen dusselig gesucht und nur Tools gefunden habe, die vor dem Sleep *Laufwerke* auswerfen, die Frage zurück an dich: Weißt du ein Tool, das auch andere Geräte als nur Laufwerke auswirft?

Beitrag von „kuckkuck“ vom 24. Mai 2019, 16:39

Stimmt, garnicht dran gedacht Es sollte rein softwaretechnisch die Möglichkeit geben USB Ports softwaremäßig zu reinitialisieren, sollte eigentlich auch über das ACPI gehen, wenn man eine entsprechende Methode schreibt (Wirkt sich dann aber wahrscheinlich auf alle Ports aus, sprich alle Ports werden reinitialisiert. zB _PWR Methoden beziehen sich meist auf den USB Controller, nicht auf einzelne Ports). Ich habe mich nie mit dem Thema beschäftigt und kann mich da aktuell auch nicht wirklich reinarbeiten. Angesichts dessen, dass es jedoch wahrscheinlich einiges an Arbeit für dich ist, dich dort reinzuarbeiten (außer du findest jemanden der selbiges bereits gemacht hat) würde ich dir fast raten die Zeit zu investieren um das Problem zu beheben anstatt eine Notlösung zu finden. Frage ist hier, ob das wirklich ein HardwareProblem des Wacom ist, wenn ja siehts natürlich bisschen blöd aus und außerdem die

Frage was nötig ist um das Problem zu beheben. Ein Neustart des Controllers? Sprich Strom entfernen und neu verbinden? Dann würde mir als ziemlicher Dirty Hack einfallen allen USB Ports den Strom im Sleep komplett zu nehmen, dieser ist dann nach dem Sleep wieder verfügbar. In der _DSM des USBX Devices gibt es entsprechende Parameter, mit denen könntest du mal rumspielen. Schön ist aber was anderes, da müsstest du aktuell lieber mal wen anderes fragen

---

Beitrag von „mabam“ vom 24. Mai 2019, 21:31

Naja, wenn ein Dirty Hack funktioniert, darf es von mir aus auch ein Dirty Hack sein. Nur fliegt die Tastatur dann auch raus und kann ich den Rechner damit nicht mehr aufwecken, was ich eher doof fände.

Ich glaube, es liegt wirklich am Wacom. Andere berichten vom selben Problem, auch unter Windows.

EDIT:

Es liegt definitiv am Wacom. Wenn es nach dem Wake aus bleibt (also das Lämpchen am Wacom nicht leuchtet), funktioniert es zwar nicht, aber es hat Strom und wird im System erkannt:

Code

1. $ uhubctl
2. Current status for hub 20-4 [2109:2811 VIA Labs, Inc. USB2.0 Hub, USB 2.10, 4 ports]
3. Port 1: 0103 power enable connect [056a:0374 Wacom Co.,Ltd. Intuos S 8BH00R2024087]
4. Port 2: 0100 power
5. Port 3: 0100 power
6. Port 4: 0100 power

Ein Gedanke:

Wenn ich eine zusätzliche USB-Karte einbaue, die nur mit einem separaten Treiber funktioniert, dann müsste ich diesen doch per Skript unloaden und loaden können und damit das Problem beheben, oder?

Nur welche Karte funktioniert nur mit separatem Treiber? Hat da wer einen Tipp?

EDIT:

Mal von hinten nach vorne aufgerollt:

- Laut Systeminformationen benötigt das Tablet 500 mA.
- Ohne eingesteckten Netzadapter ist das genau das Maximum, das das „AmazonBasics USB Hub 3.0 mit 4 Ports“ liefern kann.
- Laut https://github.com/mvp/uhubctl ist das „AmazonBasics USB Hub 3.0 mit 4 Ports“ eines der wenigen USB-Hubs, bei dem sich USB Power sogar je Port softwaremäßig ein- und ausschalten lässt.
- uhubctl kann mithilfe von Homebrew oder MacPorts auf macOS installiert werden und dient als Command Line Tool, um den Strom der USB-Ports ein- und auszuschalten.

Ich glaube, ich habe einen bezahlbaren Workaround für mein Problem gefunden.

EDIT 2:

Laut https://github.com/mvp/uhubctl#usb-30-duality-note muss man mit uhubctl und USB 3.0 aufpassen. Das Tablet braucht sowieso nur USB 2.0, also nehme ich vielleicht doch besser das Belkin F5U701-BLK mit USB 2.0.

Beitrag von „mabam“ vom 2. Juni 2019, 11:31

Ich habe mir jetzt doch das AmazonBasics USB Hub gekauft. Es wird von uhubctl einwandfrei erkannt (als zwei Hubs, eins mit USB 2 und eins mit USB 3, wie in der Beschreibung zu uhubctl erwähnt):

Code

```
$ uhubctl
Current status for hub 20-6 [2109:2811 VIA Labs, Inc. USB2.0 Hub, USB 2.10, 4 ports]
Port 1: 0103 power enable connect [056a:0374 Wacom Co.,Ltd. Intuos S 8BH00R2024087]
Port 2: 0100 power
Port 3: 0100 power
Port 4: 0100 power
Current status for hub 21-22 [2109:8110 VIA Labs, Inc. USB3.0 Hub, USB 3.00, 4 ports]
Port 1: 02a0 power 5gbps Rx.Detect
Port 2: 02a0 power 5gbps Rx.Detect
Port 3: 02a0 power 5gbps Rx.Detect
Port 4: 02a0 power 5gbps Rx.Detect
```

Alles anzeigen

Das Wacom Tablet an Port 1 wird erkannt und ich kann es per `uhubctl -a off -p 1` aus- bzw. per `uhubctl -a on -p 1` wieder einschalten.

In der IO Registry sieht das Hub so aus:

Code

```
$ ioreg -w 0 -rc AppleUSBDevice
[...]

+-o USB3.0 Hub @15100000 <class AppleUSBDevice, id 0x1000005c2, registered, matched, active, busy 0 (2 ms), retain 14>
| {
| "sessionID" = 132686865729
| "iManufacturer" = 1
| "bNumConfigurations" = 1
| "idProduct" = 33040
| "bcdDevice" = 36980
| "Bus Power Available" = 450
| "USB Address" = 4
| "bMaxPacketSize0" = 9
| "iProduct" = 2
| "iSerialNumber" = 0
| "bDeviceClass" = 9
| "Built-In" = No
```

18. | "locationID" = 353370112
19. | "bDeviceSubClass" = 0
20. | "bcdUSB" = 768
21. | "USB Product Name" = "USB3.0 Hub "
22. | "PortNum" = 1
23. | "non-removable" = "no"
24. | "IOCFPlugInTypes" = {"9dc7b780-9ec0-11d4-a54f-000a27052861"="IOUSBFamily.kext/Contents/PlugIns/IOUSBLib.bundle"}
25. | "bDeviceProtocol" = 3
26. | "IOUserClientClass" = "IOUSBDeviceUserClientV2"
27. | "IOPowerManagement" = {"DevicePowerState"=0,"CurrentPowerState"=3,"CapabilityFlags"=65536,"MaxPowerState"=4,"DriverI
28. | "kUSBCurrentConfiguration" = 1
29. | "Device Speed" = 3
30. | "USB Vendor Name" = "VIA Labs, Inc. "
31. | "idVendor" = 8457
32. | "IOGeneralInterest" = "IOCommand is not serializable"
33. | "IOClassNameOverride" = "IOUSBDevice"
34. | }
35. |
36. +-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x1000005c6, !registered, !matched, active, busy 0, retain 5>
37. +-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x1000005cd, !registered, !matched, active, busy 0 (0 ms), retain 4>
38.
39. +-o USB2.0 Hub @14600000 <class AppleUSBDevice, id 0x1000005d9, registered, matched, active, busy 0 (2 ms), retain 15>
40. | {
41. | "sessionID" = 132721445830
42. | "iManufacturer" = 1
43. | "bNumConfigurations" = 1
44. | "idProduct" = 10257
45. | "bcdDevice" = 36976
46. | "Bus Power Available" = 250
47. | "USB Address" = 5
48. | "bMaxPacketSize0" = 64
49. | "iProduct" = 2
50. | "iSerialNumber" = 0
51. | "bDeviceClass" = 9
52. | "Built-In" = No
53. | "locationID" = 341835776

```
| "bDeviceSubClass" = 0
| "bcdUSB" = 528
| "USB Product Name" = "USB2.0 Hub "
| "PortNum" = 6
| "non-removable" = "no"
| "IOCFPlugInTypes" = {"9dc7b780-9ec0-11d4-a54f-000a27052861"="IOUSBFamily.kext/Contents/PlugIns/IOUSBLib.bundle"}
| "bDeviceProtocol" = 1
| "IOUserClientClass" = "IOUSBDeviceUserClientV2"
| "IOPowerManagement" = {"DevicePowerState"=0,"CurrentPowerState"=3,"CapabilityFlags"=65536,"MaxPowerState"=4,"DriverI
| "kUSBCurrentConfiguration" = 1
| "Device Speed" = 2
| "USB Vendor Name" = "VIA Labs, Inc. "
| "idVendor" = 8457
| "IOGeneralInterest" = "IOCommand is not serializable"
| "IOClassNameOverride" = "IOUSBDevice"
| }
|
+-o WacomTabletDrive <class IOUSBDeviceUserClientV2, id 0x1000005dd, !registered, !matched, active, busy 0, retain 5>
+-o AppleUSBInterface@0 <class AppleUSBInterface, id 0x1000005e3, !registered, !matched, active, busy 0 (0 ms), retain 4>

[...]
```

Alles anzeigen

Das Netzteil des Hubs habe ich nicht angeschlossen, die Stromversorgung reicht wie erhofft auch so für das Wacom Tablet.

Jetzt muss ich das Ganze noch über SleepWatcher konfigurieren, sodass das Tablet automatisch vor dem Sleep aus- und nach dem Wake eingeschaltet wird.

Außerdem möchte ich das Gehäuse des Amazon-Hubs entfernen und die Anschlüsse mit dem restlichen Innenleben an einem PCI-Slotblech befestigen, damit ich es im Rechner anschließen und die vier Ports praktisch wie interne USB-Anschlüsse nutzen kann.

EDIT:

Nach dem Wake per `uhubctl -a on -p 1` wieder einschalten ist nicht nötig, da macOS beim Aufwachen wohl automatisch alle Ports mit Strom versieht. Port 1 des 3.0-Hubs ist dann aber „disabled“. Vielleicht, weil das Wacom sich nur über USB 2.0 verbindet (?)

Ich kann es vor dem Sleep aber mit `uhubctl -a off -el 20-6 -p 1` ausschalten, dann wird nur Power für USB 2.0 und nicht für USB 3.0 ausgeschaltet.

Beitrag von „kuckkuck“ vom 2. Juni 2019, 17:14

Wow, das nenne ich mal einen ausgeklügelten Workaround, sehr schön die Idee und Ausführung!

Beitrag von „mabam“ vom 2. Juni 2019, 19:47

Danke!

Ich mag Workarounds: https://www.hackintosh-forum.de/forum/thread/42802

Beitrag von „kuckkuck“ vom 2. Juni 2019, 20:05

Man muss sich nur der Grenze zwischen "Workaround" und "Dirty Hack" bewusst sein

---

Beitrag von „mabam“ vom 2. Juni 2019, 20:24

Screen Wakener ist im Prinzip auch nur ein Workaround. Ich habe nur einen Sicherheitsmechanismus eingebaut, der vielleicht etwas weiter geht als ein Workaround. Das führt schlimmstenfalls aber schlicht dazu, dass die automatische Konfiguration von Screen Wakener nicht funktioniert, um den Monitor nicht mit Einstellungen zu versehen, die nicht zu ihm passen.

Wenn ich mir das Ganze aber recht anschaue, übernimmt der im Bundle inbegriffene displayplacer eigentlich den Sicherheitscheck (ich hatte zuerst vor, ein anderes Tool mitzuliefern, bevor ich displayplacer fand). Ich werde den Mechanismus also wohl in einer neuen Version rausschmeißen.

Aber das gehört eigentlich in den anderen Thread.

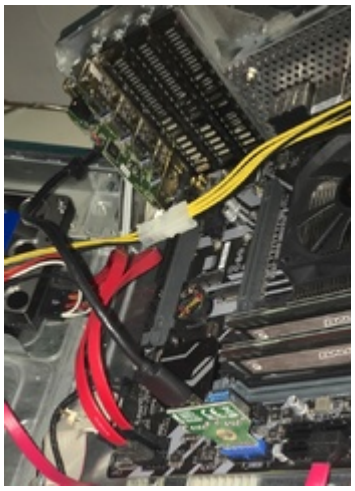
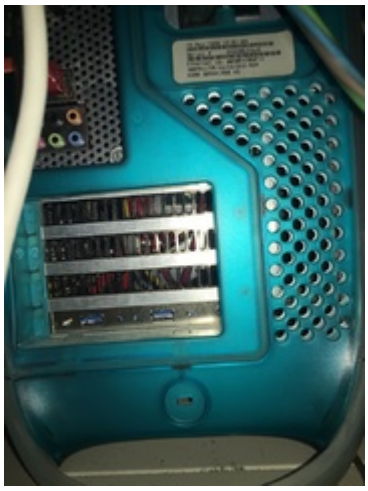
Beitrag von „kuckkuck“ vom 2. Juni 2019, 22:43

Ich meinte jetzt nicht unbedingt, dass Screen Wakener ein Dirty Hack ist, man muss bei Workarounds nur immer abwägen wie weit man geht, um Dirty Hacks zu verhindern. Im schlimmsten Fall ist ein Dirty Hack immernoch besser als die Nichtfunktion, das ist in der Hackintoshwelt common procedure und damit hatte ich selbst auch schon gut was am Hut

---

Beitrag von „mabam“ vom 13. Juni 2019, 01:03

So, nachdem ich das erste Hub beim Versuch, es mit einem PCI-Slotblech zu versehen verheizt habe (weil ich erst nach dem Durchbohren der Platine sah, dass es sich um eine Multilayer-Leiterplatte handelte und damit im Inneren wohl noch was Anderes als nur Masse erwischt habe), kam gestern das neue Hub und ist nun eingebaut:

- Auf dem ersten Bild sieht man ganz unten einen Adapter USB 3-Pinheader –> 2 × USB A-Buchse, in den das Hub eingesteckt ist.
- Auf dem zweiten Bild ist die Platine des USB-Hubs mit einen Slotblech verschraubt.
- Auf dem dritten Bild sieht man das Ganze dann von außen.
  Das Slotblech hatte ich noch rumliegen. Es hat nur zwei Öffnungen, aber ich habe sonst genügend USB-Anschlüsse und zwei schaltbare reichen mir. (Zusätzliche Löcher im Slotblech wären mit meinem Werkzeug wohl relativ hässlich geworden, also hab ich’s lieber gelassen.)

Vor dem Sleep Power ausschalten geht über:

Code

1. uhubctl=$(uhubctl); uhubctl -a off -el $(echo "$uhubctl" | grep 'USB 2' | cut -d ' ' -f 5) -p $(echo "$uhubctl" | grep Wacom | cut -d ' ' -f 4)

So funktioniert es auch noch, sollte ich das Tablet mal am anderen Anschluss einstecken.

Power per Befehl wieder einschalten ist nicht nötig, da macOS das nach dem Sleep automatisch macht.

Beitrag von „mabam“ vom 30. Juni 2019, 00:18

So, falls irgendwer mal ein ähnliches Problem hat, möchte ich meine letztendliche

Konfiguration hier dokumentieren:

Ich wollte mit sleepwatcher keinen externen Skript ausführen, obwohl das eigentlich so gedacht ist. Denn sonst muss ich neben dem Launch Agent unter /Library/LaunchAgents/org.mabam.wacom.sleepwatcher.plist noch irgendwo anders einen Skript unterbringen, der aber eigentlich zum Launch Agent gehört und ein recht einsames Dasein pflegen würde. Ich hab' lieber übersichtlich alles in einem, damit ich auch Jahre später noch auf einen Blick sehen kann, was wozu gehört.

Wenn ich im Launch Agent Variablen ins Argument hinter /usr/local/sbin/sleepwatcher schreibe, expandiert letzterer diese einmal und merkt sich den daraus resultierenden Wert für jegliche folgende Ausführung. Sollte ich in der Zwischenzeit das Intuos also in einen anderen USB-Port gesteckt haben, versagt sleepwatcher.

Gelöst habe ich das, indem der Launch Agent beim Laden den benötigten Skript nach /tmp schreibt, wo sleepwatcher ihn dann ausführt. So werden die Variablen bei jedem Ausführen neu expandiert und ich kann das Intuos nach Lust und Laune umstöpseln. Beim Runterfahren wird der Skript vom System gelöscht und beim nächsten Hochfahren vom Launch Agent wieder erstellt (ein ewiger Kampf also ...). Auf diese Weise ist alles Nötige einzig im Launch Agent enthalten.

Das Problem, dass Dateien in /tmp nicht ausführbar gemacht werden können, wird umgangen, indem das Argument hinter /usr/local/sbin/sleepwatcher mit /bin/bash beginnt.

Hier der Launch Agent:

XML

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
   "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3. <plist version="1.0">
4. <dict>
5. <key>Label</key>
6. <string>org.mabam.wacom.sleepwatcher</string>
7. <key>LimitLoadToSessionType</key>
8. <array>
```

```
9. <string>LoginWindow</string>
10. <string>Aqua</string>
11. </array>
12. <key>ProgramArguments</key>
13. <array>
14. <string>/bin/bash</string>
15. <string>-c</string>
16. <string>printf '# Get current status from uhubctl; parse it to only keep the ID\n# of the hub with the port no. the Intuos S is connected to.\nuhubctlHubPort=$(/usr/local/bin/uhubctl | awk -F '\''[: ]'\'' '\''/Current status for hub/ {u=$5} /Intuos S/ {print u; print $4}'\'')\n\n# Split the two lines from above into two separate variables.\nuhubctlHub=$(printf $uhubctlHubPort)\nuhubctlPort=$(printf "${uhubctlHubPort##*$'\''\n'\''}")\n\n# Unpower the Intuos S using the generated variables.\n/usr/local/bin/uhubctl -a off -el $uhubctlHub -p $uhubctlPort\n' > /tmp/org.mabam.wacom.sleepwatcher.sh; /usr/local/sbin/sleepwatcher -Vs "/bin/bash /tmp/org.mabam.wacom.sleepwatcher.sh"</string>
17. </array>
18. <key>RunAtLoad</key>
19. <true/>
20. <key>KeepAlive</key>
21. <true/>
22. </dict>
23. </plist>
```

Alles anzeigen

Ich habe inzwischen übrigens festgestellt, dass mein Dell 1704FPT-Monitor über ein per uhubctl schaltbares USB 2-Hub verfügt. Wenn ich das mit obiger Lösung verwende, wacht der Rechner jedoch sofort wieder auf, nachdem er im Sleep-Modus angelangt ist. Seltsam. Wenn ich das AmazonBasics-Hub (USB 3) verwende, funktioniert alles einwandfrei.