

[Sammelthread] MacOS Monterey 12.x DEV-Beta Erfahrungen

Beitrag von „LetsGo“ vom 9. Oktober 2021, 13:09

[apfelnico](#)

Mich würde es wirklich interessieren, wie man das per SSDT bewerkstelligen könnte. Bei meinem Dell habe ich so eine SSDT mit halt 26 definierten Ports. Gibt es irgendwo einen Guide, wie man eine SSDT für sauberes USB Mapping erstellt? Oder kann man die nicht benötigten Ports (um das 15 Port Limit einzuhalten) einfach aus der SSDT entfernen und damit eine Neue erstellen?

```
SSDT
L_SSDT_KHC_RHUB
L_SSDT_KHC_RHUB_H001
L_SSDT_KHC_RHUB_H002
L_SSDT_KHC_RHUB_H003
L_SSDT_KHC_RHUB_H004
L_SSDT_KHC_RHUB_H005
L_SSDT_KHC_RHUB_H006
L_SSDT_KHC_RHUB_H007
L_SSDT_KHC_RHUB_H008
L_SSDT_KHC_RHUB_H009
L_SSDT_KHC_RHUB_H010
L_SSDT_KHC_RHUB_H011
L_SSDT_KHC_RHUB_H012
L_SSDT_KHC_RHUB_H013
L_SSDT_KHC_RHUB_H014
L_SSDT_KHC_RHUB_L001
L_SSDT_KHC_RHUB_L002
L_SSDT_KHC_RHUB_S001
L_SSDT_KHC_RHUB_S002
L_SSDT_KHC_RHUB_S003
L_SSDT_KHC_RHUB_S004
L_SSDT_KHC_RHUB_S005
L_SSDT_KHC_RHUB_S006
L_SSDT_KHC_RHUB_S007
L_SSDT_KHC_RHUB_S008
L_SSDT_KHC_RHUB_S009
L_SSDT_KHC_RHUB_S010

// Intel ACPI Component Architecture
// AML/ASL Interpreter version 20190220 (2019-02-20)
// Copyright (c) 2009 - 2019 Intel Corporation
//
// Disassembling to symbolic ASL+ operators
//
// Disassembly of IASL\ASL\IACI.asl, Sat Oct  9 22:04:48 2021
//
// Original Table Header:
//
// Signature      "SSDT"
// Length        0x00000026 (38)
// Revision      0x00000001
// Checksum      0x00000000
// OEM ID        "DELL"
// OEM Table ID  "DELL_SSDT"
// OEM Revision  0x00000000 (0)
// Compiler ID   "AIDL"
// Compiler Revision 0x00000017 (23)
//
//
// DefinitionBlock ("SSDT", 1, "DELL", "DELL_SSDT", 0x00000000)
//
//
// External (_SB._GPIV.Methods) // 3 Arguments
// Internal (_SB._PCI0.MPC._MPCB.H001, Device)
// Internal (_SB._PCI0.MPC._MPCB.H002, Device)
// Internal (_SB._PCI0.MPC._MPCB.H003, Device)
// Internal (_SB._PCI0.MPC._MPCB.H004, Device)
// Internal (_SB._PCI0.MPC._MPCB.H005, Device)
// Internal (_SB._PCI0.MPC._MPCB.H006, Device)
// Internal (_SB._PCI0.MPC._MPCB.H007, Device)
// Internal (_SB._PCI0.MPC._MPCB.H008, Device)
// Internal (_SB._PCI0.MPC._MPCB.H009, Device)
// Internal (_SB._PCI0.MPC._MPCB.H010, Device)
// Internal (_SB._PCI0.MPC._MPCB.H011, Device)
// Internal (_SB._PCI0.MPC._MPCB.H012, Device)
// Internal (_SB._PCI0.MPC._MPCB.H013, Device)
// Internal (_SB._PCI0.MPC._MPCB.H014, Device)
// Internal (_SB._PCI0.MPC._MPCB.S001, Device)
// Internal (_SB._PCI0.MPC._MPCB.S002, Device)
// Internal (_SB._PCI0.MPC._MPCB.S003, Device)
// Internal (_SB._PCI0.MPC._MPCB.S004, Device)
// Internal (_SB._PCI0.MPC._MPCB.S005, Device)
// Internal (_SB._PCI0.MPC._MPCB.S006, Device)
// Internal (_SB._PCI0.MPC._MPCB.S007, Device)
// Internal (_SB._PCI0.MPC._MPCB.S008, Device)
// Internal (_SB._PCI0.MPC._MPCB.S009, Device)
// Internal (_SB._PCI0.MPC._MPCB.S010, Device)
// External (IOPIC, Processor)
//
```