



Dann noch das hier, das ist für mich echt verdammt viel Gold wert:

### Zitat von Brumbaer

Die `_LID` Routine gibt es nur einmal. Was mache ich denn, wenn es eine Routine mehrmals gibt, wie `_DSM` oder `_CRS` ?

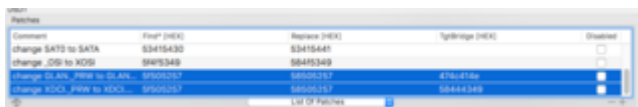
Es gibt in der Clover DSDT Patches Tabelle eine Spalte **TgtBridge**. Dort kann man eine Kennung eines Gerätes eintragen auf das der Patch begrenzt werden soll. Auch wieder als Hexzahlen. TgtBridge ähnelt dem Scope-Befehl. Dummerweise ist die Funktionalität eingeschränkt, da das nur funktioniert, wenn das zu ersetzende Etwas innerhalb des Device-Befehls der TgtBridge steht. Leider werden Scope-Befehle ignoriert.

Da ich die DSDT nicht patchen möchte und im Log öfter mal das hier gefunden habe:

### Zitat

```
2018-02-06 19:26:02.227393+0100 localhost kernel[0]: (AppleACPIPlatform) Wake reason: GLAN XDCI
```

... musste ich das direkt mal ausprobieren. ACPI-Patches, mit denen ich die Methode `_PRW` in den Geräten XDCI und GLAN in XPRW umbenannt habe:



Comment	Find HEX	Replace HEX	TgtBridge HEX	Disabled
change SATS to SATA	5B415430	5B415441		<input type="checkbox"/>
change _DS to XDS	5B4F5349	5B4F5349		<input type="checkbox"/>
change GLAN_PRW to GLAN_XPRW	5F022157	5F022157	476c416a	<input checked="" type="checkbox"/>
change XDCI_PRW to XDCI_XPRW	5F022157	5F022157	5B444349	<input checked="" type="checkbox"/>

Dabei kommt dann das hier raus:

```

Scope (_SB.PCI0)
{
  Name (LTMZ, Zero)
  Name (SBPZ, Zero)
  Name (LWSL, Zero)
  Name (LWSL, Zero)
  Device (GLAN)
  {
    Name (_ADR, 0x01F000) // _ADR: Address
    Method (_DGM, 4, Serialized) // _DGM: Device-Specific Method
    {
      IF (PCID (Arg0))
      {
        Return (PCID (Arg0, Arg1, Arg2, Arg3))
      }

      Return (Buffer (One)
      {
        0x00
      })
    }
  }

  OperationRegion (GLBA, PCI_Config, Zero, 0x0100)
  Field (GLBA, AnyAcc, NoLock, Preserve)
  {
    DVID, 16
  }

  Field (GLBA, ByteAcc, NoLock, Preserve)
  {
    Offset (0x0C),
    Offset (0x0D),
    PRW, 1,
    PRS, 1
  }

  Method (_PRW, 0, NotSerialized)
  {
    Return (GPRW (0x0D, 0x04))
  }
}

```

Und dazu diese SSDT:

Code

1. DefinitionBlock ("", "SSDT", 2, "HACK ", "GLANXDCI", 0x00001000)
2. {
3. External (\_SB\_.PCI0, DeviceObj) // (from opcode)
4. External (\_SB\_.PCI0.GLAN, DeviceObj) // (from opcode)
5. External (\_SB\_.PCI0.XDCI, DeviceObj) // (from opcode)
6. External (GPRW, MethodObj) // 2 Arguments (from opcode)
- 7.
- 8.
9. Scope (\\_SB.PCI0)
10. {
11. Scope (\\_SB.PCI0.GLAN)
12. {
13. Method (\_PRW, 0, NotSerialized) // \_PRW: Power Resources for Wake
14. {
15. Return (GPRW (0x0D, Zero))
16. }
17. }
- 18.
- 19.
20. Scope (\\_SB.PCI0.XDCI)
21. {
22. Method (\_PRW, 0, NotSerialized) // \_PRW: Power Resources for Wake
23. {
24. Return (GPRW (0x0D, Zero))

- 25. }
- 26. }
- 27. }
- 28. }

Alles anzeigen

Sollte funktionieren... ~~Wobei ich mich frage, ob ich noch eine Dummy-Methode namens "XPRW" anlegen sollte?~~ Nee, Unsinn. XPRW wird ja eh nicht aufgerufen.