

# USB mittels SSDT deklarieren

Beitrag von „apfelnico“ vom 14. Oktober 2021, 08:54

## [Zitat von ST33Z0](#)

Ich habe einen USB C port an meinem rechner. Ich nutze den zwar nicht, aber mich würde interessieren, was die Bezeichnung bedeutet: "USB 3.2 Type-C+Sw." Sw = switch? Was wird denn da geschwitcht? offline

hatte ich im Eingangsthread geschrieben, vielleicht nicht genügend erklärt.

## [Zitat von apfelnico](#)

0x00 - USB2 (ausschliesslich unabhängige USB2 werden so deklariert)

0x03 - USB3 (auch zugehörige USB2, das heißt gleiche Buchse, werden so deklariert)

0x09 - USB-C (wenn unabhängig von der Drehung des USB-C-Steckers der `_GLEICHE_` Port genutzt wird)

0x0A - USB-C (wenn je nach Drehrichtung des USB-C-Steckers ein weiterer Port genutzt wird)

0xFF - USB2 intern (zum Beispiel für Bluetooth)

Dein hinterer USB-C Port ist soweit ich mich erinnere, HS08/SS08. Die bekommen "0x09" (SW = Switch). Denn du kannst den Stecker verdrehsicher einstecken, völlig egal wie rum, es sind alle Kontakte doppelt zum eigentlichen Port gelegt.

Der interne USB-E (so heißt die Buchse), den du für ein Gehäuse-USB-C verwenden kannst, liegt auf SS01 und SS02 für USB3.2, beide Ports werden abwechselnd je nach Steckerrichtung genutzt, bekommen also beide "0x0A". Der passende usb2-Anteil dazu ist SS01. Auch der wird mit "0x0A" deklariert. Der wechselt nicht mit einem anderen Port. Denn SS02 soll nach deinem PDF ja ein interner USB2-Hub sein.

## [Zitat von ST33Z0](#)

Könnte mir jemand diese Konstruktion erklären?

Das "Package" (die vier Werte) werden übergeben in jeden Port in die Methode `_UPC`. Die ersten beiden Werte ("Zero" und "One") des Package sind variabel und werden mittels Argument "Arg0" und "Arg1" angetriggert.

In jeden Port hingegen hast du dann die eigentliche Methode "`_UPC`".

Beispielsweise:

Code

```
1. Method (_UPC, 0, NotSerialized) // _UPC: USB Port Capabilities
2. {
3. Return (GUPC (0xFF, 0x03))
4. }
```

Hier wird letztlich über "return" gesagt, packe mir den Inhalt des Package von GUPC mit `Arg0=0xFF`, `Arg1=0x03` rein. Mann kann das dann wieder differenzierter machen mit:

Code

```
1. Method (_UPC, 0, NotSerialized) // _UPC: USB Port Capabilities
2. {
3. If (_OSI ("Darwin"))
4. {
5. Return (GUPC (Zero, Zero))
6. }
7. Else
8. {
9. Return (GUPC (0xFF, 0x03))
10. }
11. }
```

Alles anzeigen

In diesem Falle würde bei "Darwin" (macOS) dieser Port deaktiviert, unter allen anderen Systemen allerdings als "USB3.x" (mit "normaler" USB-A Schnittstelle) deklariert werden.