

# USB mittels SSDT deklarieren

Beitrag von „N0b0dy“ vom 16. Oktober 2021, 20:43

[apfelnico](#) ST33Z0

Aus eure verlinkte Webseiten könnte ich die Method GPLD entziffern. zum Beispiel nehme ich sie aus Coban's SSDT

Code

1. Method (GPLD, 2, Serialized)
2. {
3. Name (PCKG, Package (0x01)
4. {
5. Buffer (0x10){}
6. }
7. CreateField (DerefOf (Index (PCKG, Zero)), Zero, 0x07, REV)
8. Store (0x02, REV)
9. CreateField (DerefOf (Index (PCKG, Zero)), 0x07, One, RGB)
10. Store (One, RGB)
11. CreateField (DerefOf (Index (PCKG, Zero)), 0x40, One, VISI)
12. Store (Arg0, VISI)
13. CreateField (DerefOf (Index (PCKG, Zero)), 0x57, 0x08, GPOS)
14. Store (Arg1, GPOS)
15. Return (PCKG)
16. }

Alles anzeigen

1-CreateField (DerefOf (Index (PCKG, Zero)), Zero, 0x07, REV) beschreibt "Revision", daher steht REV und hat 8 bits von null bis 7

Store (0x02, REV) dann gibt immer Revision 2

2-CreateField (DerefOf (Index (PCKG, Zero)), 0x07, One, RGB) beschreibt "Ignore Color", daher steht auch RGB und hat nur ein Bit.

Store (One, RGB) dann gibt immer eins, das heißt inaktive

CreateField (DerefOf (Index (PCKG, Zero)), 0x40, One, VISI) beschreibt "User Visible", daher steht VISI und ebenfalls hat ein Bit also aktive oder inaktive.

Bei "Store (Arg0, VISI)" steht erstes Argument für die Methode entweder 0 invisible oder 1 visible

CreateField (DerefOf (Index (PCKG, Zero)), 0x57, 0x08, GPOS)

Store (Arg1, GPOS) hier beschreibt "Group Position" an den Bit 87, daher steht auch GPOS

Bei "Store (Arg1, GPOS) ist das zweite Argument für die Methode, die eben falls hier ändern können

In diesem Beispiel

Code

1. Method (\_PLD, 0, NotSerialized) // \_PLD: Physical Location of Device
2. {
3. Return (GPLD (One, One))
4. }

gibt uns GPLD zurück (Revision=2, RGB=0, Visible=Yes, Position=1)

das in Buffer nicht anders als so sehen würde

Code

1. /\* 0000 \*/ 0x82, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
2. /\* 0008 \*/ 0x31, 0x1C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00