USBswitchCMD

Das Programm „USBswitchCMD“ erlaubt eine Steuerung der Cleware USB-Geräte mit Schaltfunktion, z.B. USB-Switch, USB-IO16, USB-Relais, USB-Contact,...
Auch USB-Ampeln lassen sich sehr einfach mit einem Farbbefehl steuern.
Der Aufruf des Programms kann in einem Batchprogrammerfolgen oder in Skripten oder auch direkt in einem Kommando-Fenster.

USBswitchCMD [option]

Folgende Optionen können, auch gemeinsam, verwendet werden

0	Ausschalten (0 ist die Zahl 0)
1	Einschalten (1 ist die Zahl 1)
R	Rote Ampelleuchte einschalten
Y	Gelbe Ampelleuchte einschalten
G	Grüne Ampelleuchte einschalten
O	Alle Ampelleuchten ausschalten (O ist der Buchstabe O)
-n device	verwende den USB-Switch mit dieser Seriennummer
-r	aktuelle Schalterstellung abfragen
-R	aktuelle Schalterstellung abfragen, keine Ausgabe, nur der Rückgabewert ist für die Auswertung in Skripten gesetzt
-t	Schalter neu initialisieren
-# switchnum	wähle Schalter bei Mehrfachschaltern, erstes=0
-i nnn	interval test, schalte endlos ein- uns aus, Zeitinterval nnn ms
-I nnn	interval test, schalte ein, warte nnn ms und schalte wieder aus
-p t1 .. tn	pulse mode, der Schalter wird mehrfach für o,5 sekunden eingeschaltet die Wartezeiten zwischen den Schaltvorgängen wird durch t1 – tn in Sekunden festgelegt. Nach n Schaltvorgängen ist das Programm zu Ende
-b	Binary mode, setze oder lese alle Kanäle bei Mehrfachschaltern oder Kontakten auf einmal, die Kanäle sind binär kodiert (1=erster,2=zweiter,4=dritter,8=vierter,16,....)
-v	zeige Version
-m x	x=1 löscht Power-On Verhalten x=2 schaltet Power-On Verhalten ein
-L	listet alle angeschlossenen Cleware-Geräte
-h	zeige diesen Hilfetext
-d	Debug Mode, ausführliche Test-Informationen werden ausgegeben

Die Option -b ist neu eingeführt worden, um mit einem Aufruf mehrere Kontakte gleichzeitig zu ermöglichen.

Neu ist auch die Option -R zum lesen des aktuellen Zustands. Anders wie die Option -r wird hier keine Ausgabe erzeugt, nur ein Wert zurückgegeben. Das ist hilfreich bei Skripten (siehe Beispiele)

Mit der Option **-m** kann bei einem normalen USB-Switch die Grundeinstellung nach dem Systemstart eingestellt werden. Die Option „-m 2“ schaltet den PowerOn Status ein, der das Gerät immer sofort einschaltet, wenn das System startet, also ohne Befehl. Die Option „-m 1“ schaltet diesen Modus wieder aus.

Beispiele

Soll beispielsweise bei einem USB-Switch 3 die dritte Steckdose eingeschaltet werden, lautet der Aufruf

```
USBswitchCmd 1 -# 2
```

Eine andere Anwendung ist die Steuerung einer Feuerwerksbatterie. Diese werden durch einen kurzen Schaltpulse Schritt für Schritt gezündet. Mit dem USBswitch3,5“ kann das sehr preisgünstig realisiert werden.

```
USBswitchCMD -p 1 4 5 7 12 2 7
```

Um bei einer USB-Ampel das rote Licht einzuschalten, hilft

```
USBswitchCMD R
```

oder Grün und Gelb zusammen einschalten

```
USBswitchCMD G Y
```

Der Zustand aller Kontakte eines Gerätes kann mit der Option -b gesteuert werden. Sind z.B. bei einem USB-OptoIn die Eingänge 1 und 3 aktiv, liefert der Aufruf

```
USBswitchCMD -r -b
```

den Wert 5, also Binär 0000 0101.

Bei der USB-TischAmpel4 können die 12 Leuchten auch mit der -b Option einzeln angesteuert werden, dann kann z.B. eine Seite die Rot anzeigen, die andere Grün.
Der Wert, der mit diesen Kommando geschickt wird, ist 16 bit lang und ist wie folgt aufgebaut:

```
bbbbggggyyyyrrrr
0b0000000000000000     - Binärzahl im C/C++/C# Format
```

für Rot, Gelb (y) und Grün sind 4 Bits notwendig, da hier alle 4 Lichter einzeln angesteuert werden. Um alle 4 roten Lichter einzuschalten, wird 0b0000000000001111 = 0x000f = 15 gesendet. Das Blinken ist in den oberen Bits geregelt. Dieser Wert ist die Blinkfrequenz in 0,5 Sekundeneinheiten. Alle 4 roten Leuchten blinken im Sekundentakt mit 0b0010000000001111 = 0x200f = 8207 gesendet wird, also

```
USBswitchCMD -b 8207
```

Sollen der Zustand in einem Skript verarbeitet werden, empfiehlt sich die Option -R. Hier ein kleines Perlskript als Beispiel:

```
my @args = ("USBswitchCMD", "-b", "-R") ;
system(@args) ;
printf "@args returns %d\n", $? >> 8 ;
```