

1. Command "USBswitchCmd"

The commandline tool "USBswitchCmd" to control Cleware USB switches and displays/traffic lights, e.g. USB-Switch, USB-IO16, USB-Relais, USB-Contact,....

This function is designed to be used inside batch programs or scripts. A typical usage is to call USBswitchCMD to turn a light on when the email client gets a new mail. Most email programs allow to call a program in this case.

When "USBswitchCmd" is called with the Argument "0", a connected USB-Switch is turned off. If the Argument is "1", it is turned on. The following arguments are supported:

USBswitchCmd [command options]

Command options are:

0	turns switch off (0 is an zero)
1	turns switch on
R	turns red light on (in case of traffic light device)
Y	turns yellow light on (in case of traffic light device)
G	turns green light on (in case of traffic light device)
O	turn all lights off (in case of traffic light device, O is a letter)
-n device	use USB-Switch with this serial number
-r	read the current setting
-R	read the current setting, no print, just the return value is set for scripts
-t	reseT the device
-T	Show internal Cutter/Connect device type
-# switchnum	select switch for multiple switch device, first=0
-i nnn	interval test, turn endless on/off and wait nnn ms between state change
-I nnn	interval test, turn once on/off and wait nnn ms between state change
-p t1 .. tn	pulse mode, turn the switch n times for a 0.5 second time period on, the time between the pulse is defined by t1 t2 t3 .. tn in seconds
-b	binary mode, read or write multiple channels of a USB device with one call, the return value is binary encoded (1=first, 2=second, 4=third, 8=forth, ...)
-v	print version
-m x	x=1 disable power on at boottime x=2 enable power on at boottime
-L	list all connected Cleware devices
-h	print command usage
-d	print debug infos

The **-m** option can be used to set the default setting for a normal USB switch after system startup. The "-m 2" option switches on the PowerOn status, which always switches the device on immediately when the system starts, without sending a command. The "-m 1" option switches this mode off again.

The option -T indicate the device type in case of USB-Cutter, USB-Connect...
1=Cutter, 2=Connect, 3=Multi2, 4=Multi2X, 17=Cutter3.0, 18=Connect3.0

Examples:

Turning the third socket of the USB-Switch 3 on

```
USBswitchCmd 1 -# 2
```

Using an USB-Switch 3,5" to control a fireworks battery. These devices needs a short key press to initiate the next shot.

```
USBswitchCmd -p 1 4 5 1 12 7 8 12
```

Turning the red light of an USB-Ampel on

```
USBswitchCmd R
```

Turning the yellow and green light of an USB-Ampel on

```
USBswitchCmd Y G
```

Read all channels of an USB-OptoIn device

```
USBswitchCMD -r -b
```

it might return 5, which is binary 0000 0101. This indicate the channels 1 and 3 are active

The USB-TrafficLight4 offers a feature to control all 12 lights individually with the -b option, e.g. one side can display red, the other side is green.

The value sent with this command is 16 bits long and is structured as follows:

```
bbbbgggggyyyyrrrrr
```

0b0000000000000000 - Binary number in C/C++/C# format

4 bits are required for red, yellow and green, because all 4 lights are controlled individually here. To switch on all 4 red lights, 0b0000000000001111 = 0x000f = 15 is sent. The flashing is controlled in the upper 4 bits. This value is the flashing frequency in 0.5 second units. All 4 red lights flash every second with 0010000000001111 = 0x200f = 8207 is sent, concrete

```
USBswitchCMD -b 8207
```

This is a sample using USBswitchCMD to get the current setting in a Perl script:

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

