

# USB-Missile Anleitung

## Voraussetzung

Das Paket “libudev-dev” muss zum Kompilieren von “main.c” (Frontend) installiert sein.

## Treibermodul einrichten

Den USB-Raketenwerfer bitte erst nach dem Einrichten anschließen.

### a) Automatisch

Zur automatischen Einrichtung folgenden Befehl eingeben:

```
make install
```

### b) Manuell

Zum Kompilieren des Treibers und des Frontends folgenden Befehl eingeben:

```
make all
```

Zum Verwenden des Treibers sind folgende Schritte notwendig:

1. Kopiere “10-missile.rules” nach “/etc/udev/rules.d/”

```
sudo cp 10-missile.rules /etc/udev/rules.d/
```

2. Lade das Module.

```
sudo insmod drivers/usb-missile.ko
```

## Starten des Frontend-Programms

Sobald die Einrichtung des Treibermoduls durchgeführt wurde, kann das Frontend-Programm mit folgendem Befehl gestartet werden:

```
make start
```

## Entladen des Treibermoduls

### a) Automatisch

Folgenden Befehl eingeben:

```
make remove
```

## **b) Manuel**

Das Modul kann wieder entladen werden mit:

```
sudo rmmode usb_missile
```

Die udev-Regel entfernen:

```
sudo rm /etc/udev/rules.d/10-missile.rules
```

## **Notizen zur Entwicklung**

### **Für das Frontend:**

Zum generischen Auffinden der Devices folgenden Pfad verwenden:

```
/sys/bus/usb/drivers/usb_missile/1-3:1.0/
```

1-3 ist abhängig davon, an welchem Anschluss das Gerät eingestöpselt wurde.  
Der Link insgesamt ist allerdings von der Gestalt \*:1.0.

### **Ermittelte Spezifikationen des Geräts**

Im wesentlichen einem Python-Frontend aus dem Netz, das mit dem Standard-Treiber funktioniert, entnommen:

- <https://github.com/nmilford/stormLauncher>

Zum Experimentieren mit USB-Control-Messages dieses Frontend umgeschrieben.  
Quellcode in “frontend-launcher.py” im ./research Verzeichnis.

### **Hardware-Device-IDs**

Vendor: 0x2123

Product: 0x1010

### **USB-Control-Message-Bytes**

Befehle für Bewegungen, Feuern und LED in Hex-Format:

Stop 0x02, 0x00

Down 0x02, 0x01

Up 0x02, 0x02

Left 0x02, 0x04  
Right 0x02, 0x08  
Kombinationen aus Bewegung über Verundung des letzten Hexwertes.  
Z.B. Down-Left 0x02, 0x05  
Fire 0x02, 0x10  
LED off 0x03, 0x00  
LED on 0x03, 0x01

## Bugs und Probleme:

Memory Leak im Frontend verursacht durch libudev.  
Zwei fehlende Speicherfreigaben ausgehend von “udev\_enumerate\_add\_match\_subsystem(enumerate,”usb“);”  
und “udev\_enumerate\_scan\_devices(enumerate);”.

## Quellen

### Allgemein

- sysfs-Devices: <http://www.linuxjournal.com/article/7353>
- USB Spezifikation: [http://sdphca.ucsd.edu/lab\\_equip\\_manuals/usb\\_20.pdf](http://sdphca.ucsd.edu/lab_equip_manuals/usb_20.pdf)
- Udev Dokumentation: <http://udev.sourceforge.com/documentation/161-1/files.html>
- API zu usb-control-msg: <https://www.kernel.org/doc/html/docs/usb/API-usb-control-msg.html>
- usb\_control\_msg: <https://github.com/walac/pyusb/blob/9094c9b1ec2ac761dddce3c7d050fc4cd02e063d/usb/core.py#L997>
- Udev: <http://free-electrons.com/doc/udev.pdf>
- Udev-Einführung und -Konfiguration: [https://doc.opensuse.org/documentation/html/openSUSE\\_121/opensuse-reference/cha.udev.html](https://doc.opensuse.org/documentation/html/openSUSE_121/opensuse-reference/cha.udev.html)
- libudev API: <https://www.kernel.org/pub/linux/utils/kernel/hotplug/libudev/ch01.html>
- libudev Tutorial: <http://www.signal11.us/oss/udev/>

### Treiber-Beispiele

- <http://matthias.vallentin.net/blog/2007/04/writing-a-linux-kernel-driver-for-an-unknown-usb-device/>
- <https://github.com/joergschiller/launcher>
- [http://alex.nisnevich.com/blog/2013/02/19/face\\_tracking\\_with\\_open\\_cv\\_and\\_a\\_usb\\_missile\\_launcher.html](http://alex.nisnevich.com/blog/2013/02/19/face_tracking_with_open_cv_and_a_usb_missile_launcher.html)
- <http://developer.plantronics.com/article/accessing-usb-devices-hid>

## Veraltet

- <http://linux-usb.sourceforge.net/USB-guide/book1.html>
- <http://tldp.org/LDP/lkmpg/2.6/html/lkmpg.html> für den Linux 2.6 Kernel