

# Kopieren eines YUNSIM Simulator Sticks mit Hilfe eines Raspberry Pi

## Was wir brauchen

### Hardware:

Raspberry Pi <https://www.raspberrypi.org/>

CC2531 ZigBee USB stick z.B. <https://www.amazon.de/dp/B07X8TZRJR>

### Software:

Wiring Pi <http://wiringpi.com/download-and-install/>

CC2531 Flash Tool [https://github.com/jmichault/flash\\_cc2531](https://github.com/jmichault/flash_cc2531)

UAV-Pilot <https://yuneecskins.com/downloads>

## Vorbereitung

### Update Raspberry Pi

```
sudo apt update && sudo apt upgrade -y
```

### WiringPi installieren

```
sudo apt install wiringpi -y
```

Prüfen, ob die Installation erfolgreich war:

```
gpio -v
```

Wir sollten die gpio Version und ein paar Informationen zum Raspberry Pi sehen.

```
pi@testlite:~ $ gpio -v
gpio version: 2.50
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
Type: Model B, Revision: 02, Memory: 512MB, Maker: Egoman
* Device tree is enabled.
*--> Raspberry Pi Model B Rev 2
* This Raspberry Pi supports user-level GPIO access.
pi@testlite:~ $ █
```

Damit wir uns vom GitHub Zeug holen können, git installieren

```
sudo apt install git -y
```

## CC2531 Flash Tool für LINUX installieren

```
git clone https://github.com/jmichault/flash_cc2531.git
```

Diese Aktion erzeugt ein Unterverzeichnis für das Flash-Tool. Wir wechseln in das Unterverzeichnis und schauen nach, ob alles da ist:

```
cd flash_cc2531
```

```
ls -al
```

```
pi@testlite:~ $ cd flash_cc2531/
pi@testlite:~/flash_cc2531 $ ls -al
insgesamt 364
drwxr-xr-x 3 pi pi 4096 Sep  6 18:08 .
drwxr-xr-x 9 pi pi 4096 Sep  6 17:20 ..
-rwxr-xr-x 1 pi pi 29992 Sep  6 17:20 cc_chipid
-rw-r--r-- 1 pi pi 2192 Sep  6 17:20 cc_chipid.c
-rw-r--r-- 1 pi pi 15385 Sep  6 17:20 CCDebugger.c
-rw-r--r-- 1 pi pi 2389 Sep  6 17:20 CCDebugger.h
-rwxr-xr-x 1 pi pi 30004 Sep  6 17:20 cc_erase
-rw-r--r-- 1 pi pi 2249 Sep  6 17:20 cc_erase.c
-rwxr-xr-x 1 pi pi 31576 Sep  6 17:20 cc_read
-rw-r--r-- 1 pi pi 4107 Sep  6 17:20 cc_read.c
-rwxr-xr-x 1 pi pi 38332 Sep  6 17:20 cc_write
-rw-r--r-- 1 pi pi 11137 Sep  6 17:20 cc_write.c
drwxr-xr-x 8 pi pi 4096 Sep  6 17:20 .git
-rw-r--r-- 1 pi pi 10 Sep  6 17:20 .gitignore
-rw-r--r-- 1 pi pi 35149 Sep  6 17:20 LICENSE
-rw-r--r-- 1 pi pi 510 Sep  6 17:20 Makefile
-rw-r--r-- 1 pi pi 3087 Sep  6 17:20 README.md
-rw-r--r-- 1 pi pi 60048 Sep  6 18:10 yunsiml.hex
-rw-r--r-- 1 pi pi 60048 Sep  6 18:05 yunsim.hex
pi@testlite:~/flash_cc2531 $
```

Die grün markierten Dateien sind unsere Kommandos, die wir zum Prüfen / Lesen / Löschen und Beschreiben eines CC2531 Chips benötigen.

**Achtung:** Vor den Flashen einer neuen Firmware muss der Speicher immer komplett gelöscht werden (dies ist unten in der Prozedur enthalten).

So, jetzt sind wir bereit.

Wir brauchen nur noch die Firmware Datei „yunsim.hex“. Diese gibt es hier:

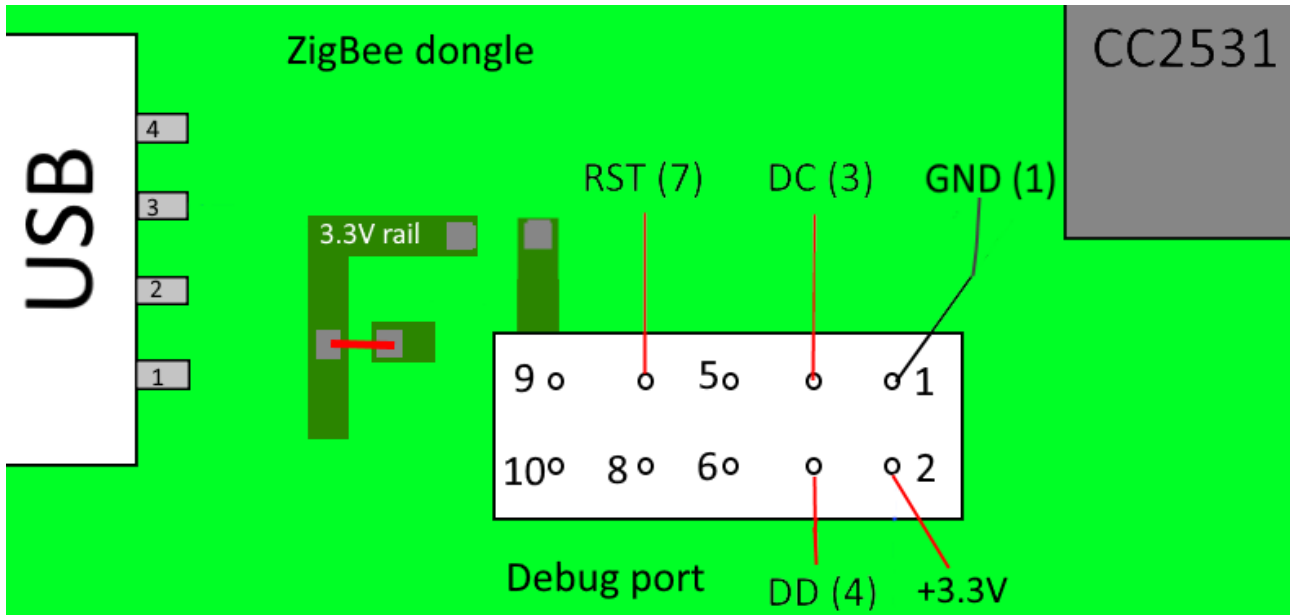
[https://1drv.ms/u/s!AtiK\\_DeOGL5shK1HofDMLfCx-C5VJg?e=dC7l2L](https://1drv.ms/u/s!AtiK_DeOGL5shK1HofDMLfCx-C5VJg?e=dC7l2L)

MD5 Summe ist: **92c2f74444819a76235d906daab2781d**.

Die Datei „yunsim.hex“ kopieren wir in das Toolverzeichnis /flash\_cc2531.

# ZigBee USB Stick mit dem Raspberry Pi verbinden

ZigBee USB Stick	Debug port pin	Raspi2 GPIO pins	Raspi3 GPIO pins
GND	1	9 Ground	39 Ground
DC	3	11 GPIO17	36 GPIO16
DD	4	13 GPIO27	38 GPIO20
RST	7	3 GPIO2	35 GPIO19



ZigBee Stick erst verbinden und dann in die USB-Buchse des Raspberry Pi stecken.

Raspberry Pi GPIO



Raspberry Pi B Rev 2 P1 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7

Key

Power +	UART
GND	SPI
IC	GPIO

Raspberry Pi B+ B+ J8 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Verbindung testen:

```
cd
flash_cc2531
./cc_chipid
```

Alternatives Kommando für Raspberry Pi 2 (mit kleinerem GPIO port Konnektor):

```
./cc_chipid -r 8 -c 0 -d 2
```

Wenn alles OK ist, wird die ID = b524 ausgegeben.

# CC2531 Firmware flashen

Optional kann man die alte Firmware speichern

```
./cc_read backup_fw.hex
```

Die alte Firmware muss zuerst gelöscht werden!

```
./cc_erase
```

Ausgabe:

```
ID = b524
```

```
erase result = 00a2
```

Dann spielen wir die YUNSIM Firmware auf. Das dauert eine Weile.

```
./cc_write yunsim.hex
```

```
pi@testlite:~ $ cd flash_cc2531/
pi@testlite:~/flash_cc2531 $ ls
CC2531_orig_fw.hex  CCDebugger.h  cc_write      README.md
CC2531ZNP-Prod.hex  cc_erase      cc_write.c    sniffer_fw_cc2531.hex
cc_chipid           cc_erase.c    firmware.txt  yunsim.hex
cc_chipid.c         cc_read      LICENSE       zboss_sniffer.hex
CCDebugger.c        cc_read.c    Makefile
pi@testlite:~/flash_cc2531 $ ./cc_chipid -r 8 -c 0 -d 2
ID = b524.
pi@testlite:~/flash_cc2531 $ ./cc_erase -r 8 -c 0 -d 2
ID = b524.
erase result = 00a6.
pi@testlite:~/flash_cc2531 $ ./cc_write -r 8 -c 0 -d 2 yunsim.hex
ID = b524.
reading line 1360.
file loaded (1368 lines read).
writing page 15/ 15.
verifying page 15/ 15.
flash OK.
pi@testlite:~/flash_cc2531 $ █
```

Alternative Kommandos für Raspberry Pi 2 (mit kleinerem GPIO port Konnektor):

```
./cc_read -r 8 -c 0 -d 2 backup_fw.hex
```

```
./cc_erase -r 8 -c 0 -d 2
```

```
./cc_write -r 8 -c 0 -d 2 yunsim.hex
```

USB Anschluß am Raspberry Pi testen:

```
ls -l /dev/ttyACM0
```

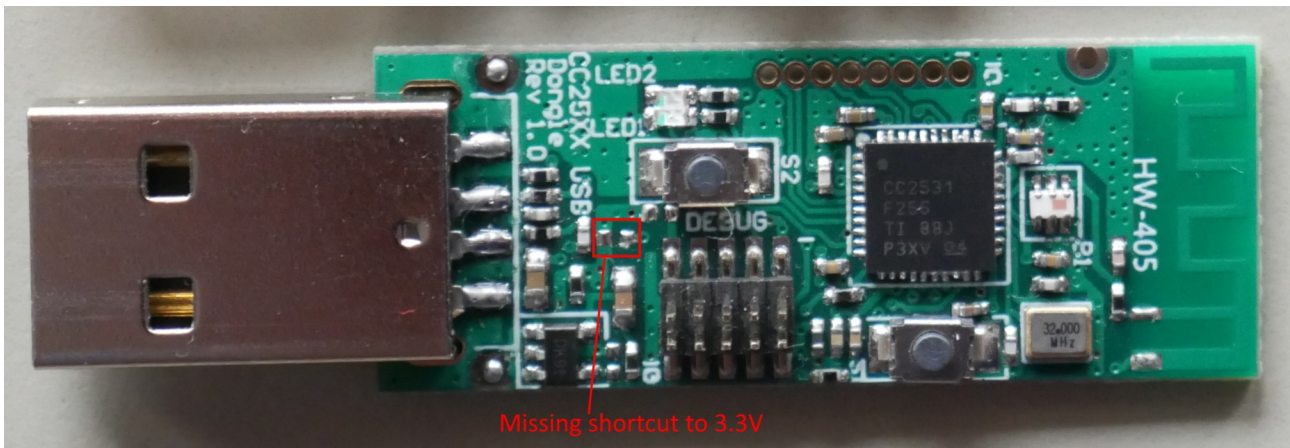
Ausgabe:

```
crw-rw---- 1 root dialout 166, 0 Nov 2 12:47 /dev/ttyACM0
```

## USB Verbindungsprobleme lösen

Die meisten der billigen ZigBee USB Sticks werden nach dem Flashen nicht als USB-Geräte erkannt. Der Grund ist, dass das USB Interface einen Pull-up Widerstand 1,5kOhm von USB D+ (3) nach 3,3V benötigt. Dies ist beim YUNSIM hart verdrahtet.

Auf der Leiterplatte der Sticks ist der Widerstand zwar auch vorhanden, es fehlt aber die Verbindung zur 3,3V Schiene.



Wir löten noch diese Verbindung wie oben gezeigt mit einem kleinen Drähtchen nach.

Siehe dazu:

<https://yuneecpilots.com/threads/looking-for-a-uav-pilot-simulator.20154/post-235812>

Nun ist der USB Stick bereit.

Wir prüfen nun ob der ZigBee USB-Stick von UAV-Pilot erkannt wird. Der Stick muss vor dem Starten des Simulators mit dem PC verbunden werden.

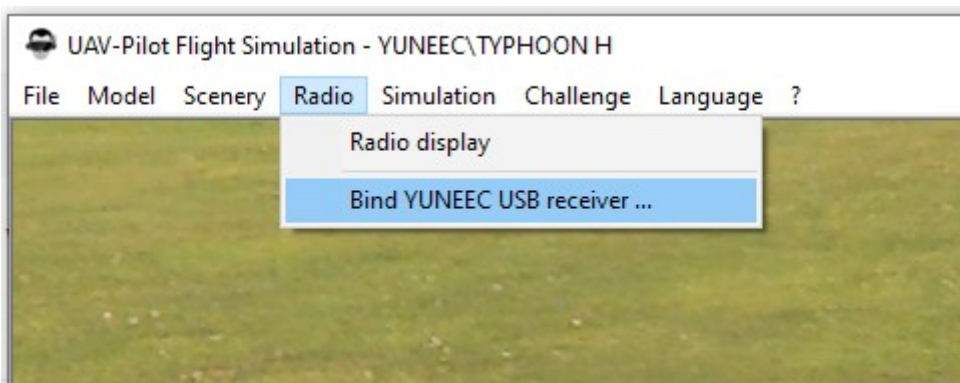
Wenn der Stick erkannt wurde, gibt es den Menüpunkt „Fernsteuerung > YUNEEC USB-Empfänger binden ...“.

# Simulatorstick mit Yuneec Fernsteuerung verbinden

Der Simulator Stick kann nun mit einer ST10(+) oder ST16 nach der Anleitung des Programms UAV-Pilot gebunden werden. Wir legen dazu ein neues Modell für den Simulator auf der Fernsteuerung an. Mit diesem Modell binden wir den ZigBee USB Stick.

Wenn die LED am ZigBee Stick rot blinkt, bedeutet das, erwartet auf eine Verbindung.

Menüpunkt „Fernsteuerung > YUNEEC USB-Empfänger binden ...“ aufrufen und der Anleitung folgen. LED am Stick aus bedeutet Bindestatus.

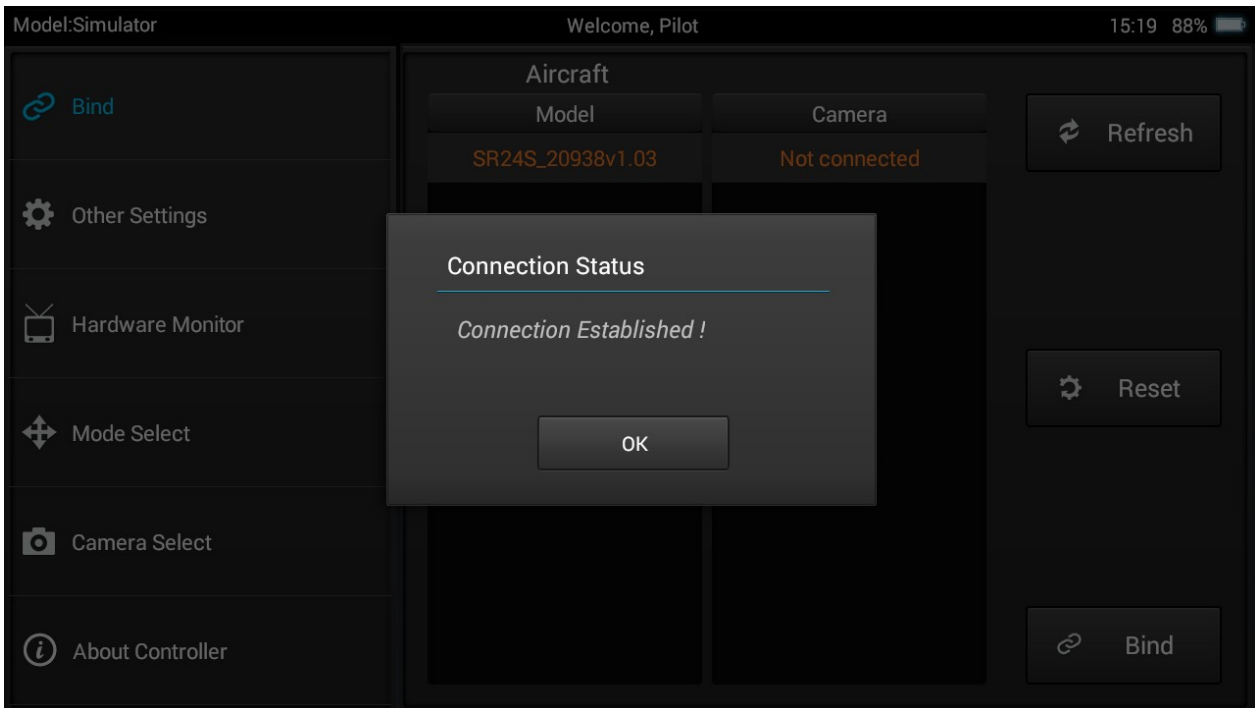


An der Fernsteuerung:  
System Settings > Bindestatus  
Refresh  
SR24S\_xxxx auswählen  
Bind

Wenn die Verbindung zustande gekommen ist, leuchtet die LED am Stick dauerhaft rot.

Nun sollten die Erfolgsmeldungen kommen.

Auf der Fernsteuerung:



Beim UAV-Simulator:

