

How clone a YUNSIM simulator dongle using Raspberry Pi

What we need

Hardware:

Raspberry Pi <https://www.raspberrypi.org/>
CC2531 ZigBee USB dongle i.e. <https://www.amazon.de/dp/B07X8TZRJR>

Software:

Wiring Pi <http://wiringpi.com/download-and-install/>
CC2531 flash tool https://github.com/jmichault/flash_cc2531

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

Preparation

Update Raspberry Pi

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

Installing WiringPi

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

Check if WiringPi was installed successful:

```
gpio -v
```

It should respond with gpio version and some information about the RaspberryPi.

```
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:~ $ █
```

Installing git if missing

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

Installing CC2531 flash tool for LINUX

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

This creates a subdirectory for the tool. Go into this directory and check if all is there:

```
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 $
```

The green marked executable are the commands we need for check / read / erase / write firmware to CC2531 chips.

Note: You must always erase the old firmware before you can flash a new one.

We are ready now to write firmware files to CC2531.

Download firmware file „yunsim.hex“

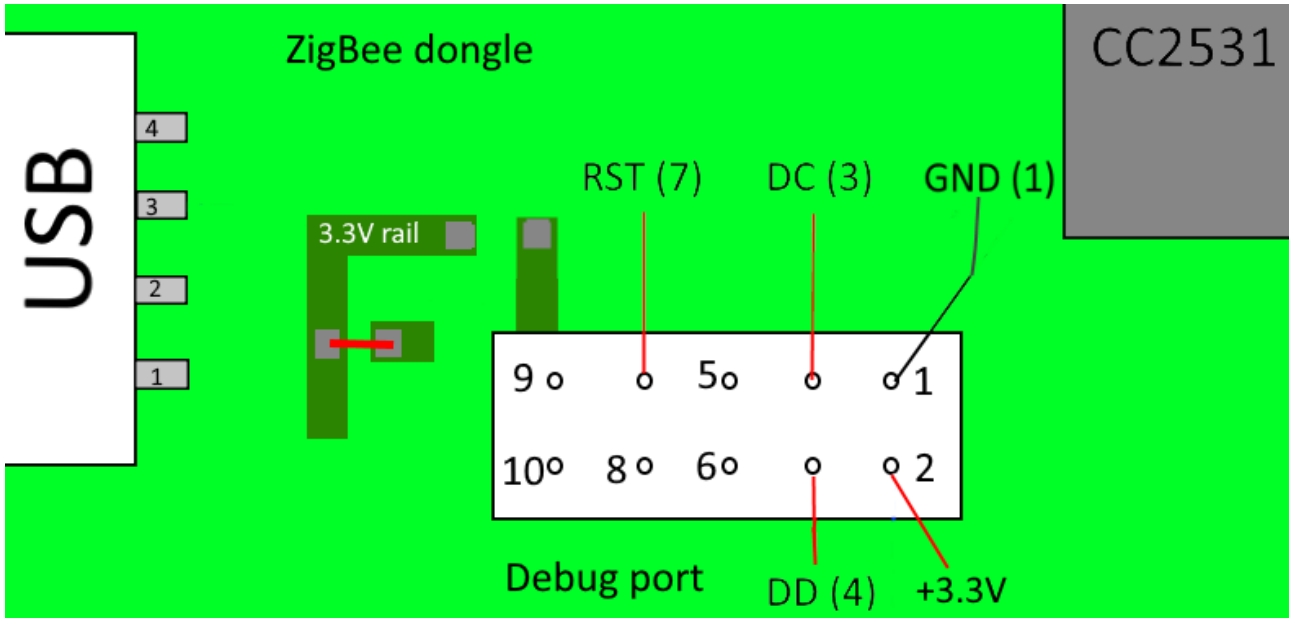
https://1drv.ms/u/s!AtiK_DeOGL5shK1HofDMLfCx-C5VJg?e=dC7l2L

MD5 sum should be **92c2f74444819a76235d906daab2781d**.

Copy file „yunsim.hex“ to tool folder on Raspberry Pi /**flash_cc2531**.

Connect ZigBee USB dongle to Raspberry Pi

ZigBee USB dongle	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



Raspberry Pi GPIO header



Raspberry Pi B Rev 2 P1 GPIO Header

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

Raspberry Pi B+ B+ J8 GPIO Header

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

Test connection:

```
cd flash_cc2531
./cc_chipid
```

Alternative command for Raspberry Pi 2 (with smaller GPIO port connector):

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

If all is OK test returns **ID = b524**

Flash firmware to CC2531

Optional save previous firmware

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

You must erase CC2531 firmware first!

```
./cc_erase
```

Flash YUNSIM firmware

```
./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 command for Raspberry Pi 2 (with smaller GPIO port connector):

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

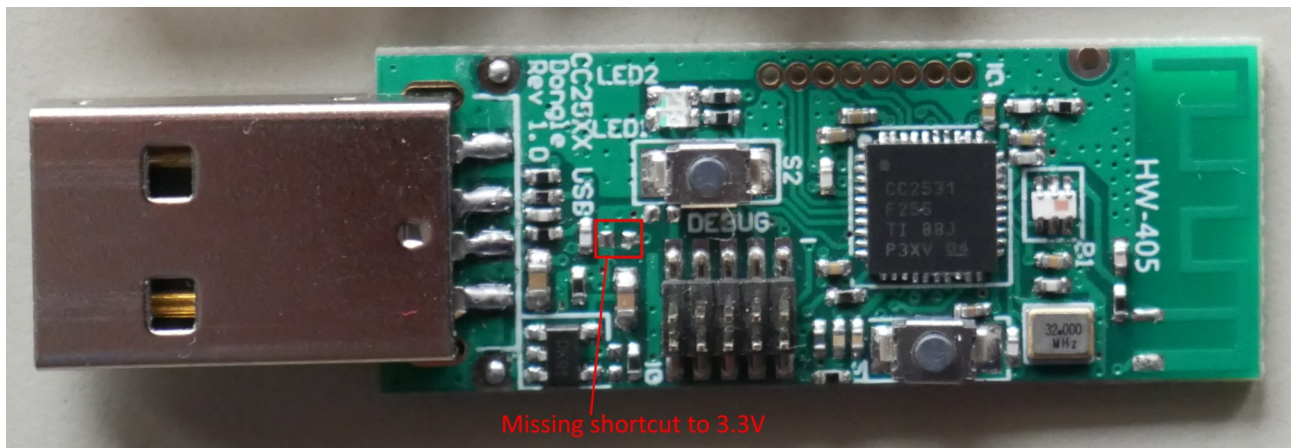
```
./cc_erase -r 8 -c 0 -d 2
```

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

Trouble shooting USB interface

Many of the cheap ZigBee dongles will not be recognized as USB device.

USB interface needs a pull-up resistor to +3.3V at USB D+ (3) wire. It is hard-wired at YUNSIM dongle. But some ZigBee dongles don't have it. A connection (0 Ohm "resistor" R92 is missing) between 3.3V and CC2531 p1_0 (pin 11) is not assembled. We need to solder a tiny wire as shown below.



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

Now the ZigBee dongle is ready to work.

Check if CC2531 dongle will be recognized by UAV-Pilot.

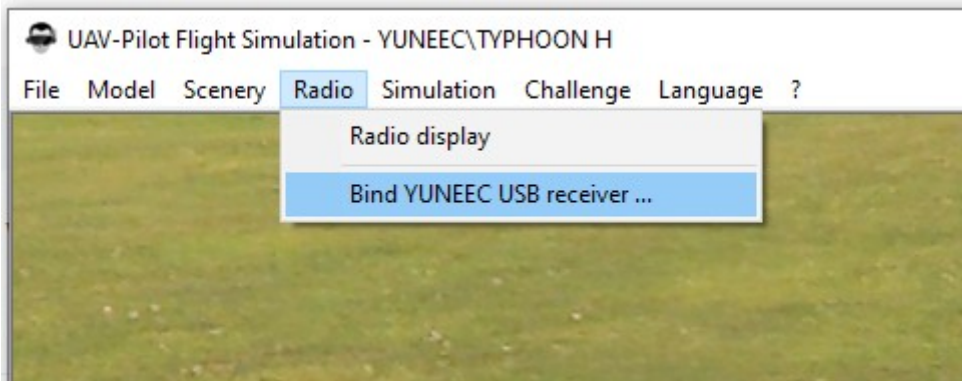
The Menu item „Bind YUNEEC USB receiver...“ at „Radio“ must appear.

Bind Simulator to Yuneec remote controller ST10 or ST16

Power on and boot up ST1x. Create a new model „Simulator“. Switch to this model. Enter ,Bind‘ menu.

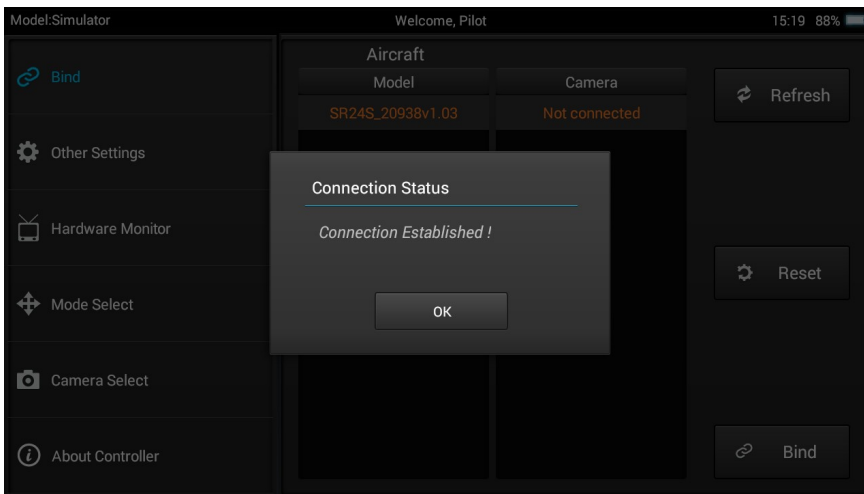
Connect ZigBee USB dongle to your PC. Start UAV-Pilot flight simulator. LED red blinking means waiting for connection.

Menu Radio > Bind YUNEEC USB receiver... and follow the instructions.



LED on ZigBee dongle off means binding mode.

ST1x:: System Settings > Bind
 Refresh
 Select SR24S_xxxxx > Bind



If we are connected the LED is solid red. Now the ZigBee dongle is bound and we are ready to fly.

