

```
1: mkdir tbs
2: cd tbs
3: wget http://www.tbsdtv.com/download/document/common/tbs-linux-drivers\_v151105.zip
4: unzip tbs-linux-drivers_v151105.zip
5: Install the CX3522+ firmware file "dvb-fe-cx24116.fw" to lib/firmware/

    mv dvb-fe-cx24116.fw /lib/firmware/
```

6: build, install and load S2API Linux drivers for TBS6920 and TBS8920 cards

6.1 (optional in case packages aren't already installed) install prerequisite packages using Ubuntu package manager:

```
Sudo apt-get update
sudo apt-get install linux-headers-`uname -r`
sudo apt-get install linux-kernel-devel
```

THIS LAST COMMAND RESULTS IN ERROR:

E: Unable to locate package linux-kernel-devel

6.2 extract linux-tbs-sources.tar.bz2 archive:

```
tar xjvf linux-tbs-sources.tar.bz2
```

THIS RESULTS IN ERROR:

```
tar (child): linux-tbs-sources.tar.bz2: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
```

THERE IS NO SUCH FILE SO INSTEAD IM GOING TO RUN:

```
tar xjvf linux-tbs-drivers.tar.bz2
```

6.3 go to driver source code directory:

```
cd linux-tbs-sources/ SUBSTITUTED TO linux-tbs-drivers/
```

6.4 build and install the driver:

```
make && make install
```

RESULTS IN THE FOLLOWING:

```
make -C /home/tv/tbs/linux-tbs-drivers/v4l
make[1]: Entering directory '/home/tv/tbs/linux-tbs-drivers/v4l'
No version yet, using 4.2.0-16-generic
scripts/make_makefile.pl
```

Updating/Creating .config
Preparing to compile for kernel version 4.2.0

WARNING: You do not have the full kernel sources installed.
This does not prevent you from building the v4l-dvb tree if you have the kernel headers, but the full kernel source may be required in order to use
make menuconfig / xconfig / qconfig.

If you are experiencing problems building the v4l-dvb tree, please try building against a vanilla kernel before reporting a bug.

Vanilla kernels are available at <http://kernel.org>.
On most distros, this will compile a newly downloaded kernel:

```
cp /boot/config-`uname -r` <your kernel dir>/.config  
cd <your kernel dir>  
make all modules_install install
```

Please see your distro's web site for instructions to build a new kernel.

```
Created default (all yes) .config file  
./scripts/make_myconfig.pl  
perl scripts/make_config_compat.pl /lib/modules/4.2.0-16-  
generic/build ./myconfig ./config-compat.h  
creating symbolic links...  
make -C firmware prep  
make[2]: Entering directory '/home/tv/tbs/linux-tbs-  
drivers/v4l/firmware'  
make[2]: Leaving directory '/home/tv/tbs/linux-tbs-  
drivers/v4l/firmware'  
make -C firmware  
make[2]: Entering directory '/home/tv/tbs/linux-tbs-  
drivers/v4l/firmware'  
CC ihex2fw  
make[2]: gcc: Command not found  
Makefile:35: recipe for target 'ihex2fw' failed  
make[2]: *** [ihex2fw] Error 127  
make[2]: Leaving directory '/home/tv/tbs/linux-tbs-  
drivers/v4l/firmware'  
Makefile:65: recipe for target 'firmware' failed  
make[1]: *** [firmware] Error 2  
make[1]: Leaving directory '/home/tv/tbs/linux-tbs-  
drivers/v4l'  
Makefile:26: recipe for target 'all' failed  
make: *** [all] Error 2
```

7 reboot in order to load the newly installed driver:

```
shutdown -r now
```

8 after reboot check that the newly installed driver is loaded correctly:

(for TBS 6920 only)

```
dmesg | grep cx23885
```

if everything is OK, the output from the above command should be similar to:

```
cx23885 driver version 0.0.1 loaded
cx23885 0000:04:00.0: PCI INT A -> GSI 32 (level, low) -> IRQ 32
CORE cx23885[0]: subsystem: 6920:8888, board: TurboSight TBS 6920
[card=14,autodetected]
cx23885_dvb_register() allocating 1 frontend(s)
cx23885[0]: cx23885 based dvb card
DVB: registering new adapter (cx23885[0])
cx23885_dev_checkrevision() Hardware revision = 0xb0
cx23885[0]/0: found at 0000:04:00.0, rev: 2, irq: 32, latency: 0, mmio:
0xdc000000
cx23885 0000:04:00.0: setting latency timer to 64
```

(for TBS 8920 only)

```
dmesg | grep cx88
```

III. here are some basic instructions how to test locking to transponder with "szap-s2":

III.1 extract "szap-s2.tar.bz2" archive:

```
# tar xjvf szap-s2.tar.bz2
```

III.2 go to szap-s2 source code directory:

```
# cd szap-s2
```

III.3 build szap-s2:

```
# make
```

III.4 szap-s2 uses configuration files (see configuration file "astra_szap-s2.conf" for example) - they are simple text files and each line of them contains information about a channel, for example:

```
BBC:11597:v:0:22000:163:92:10050
```

this means:

- channel name is BBC, you can choose whatever you want for channel name, it's just a text string

- channel frequency is 11597MHz

- transponder polarization is V(ertical)

- 0: use first LNB (if you have only one LNB, it's always 0, if you have DiSEqC switch and several LNBS connected to that switch it could be number greater than 0 and showing which LNB to use, i.e. to which DiSEqC switch port desired LNB is connected)

- 22000: symbol rate
- 163: Video PID (VPID)
- 92: Audio PID
- 10050: Service ID (SID)

the example is for "BBC World" channel on Astra 19.2E satellite. you can find all those numbers for example here:

<http://www.lyngsat.com/astra19.html>

III.5 write your szap-s2 configuration files (or create one using S2API compatible version of the "scan" tool)

III.6 use szap-s2 examples:

- to lock to DVB-S transponder: `./szap-s2 -c astra_szap-s2.conf -p -r -S 0 "BBC"`

- to lock to DVB-S2: `./szap-s2 -c sample.conf -p -r -S 1 -M 5 -C 23 "CHAN_NAME"`

where "-S 1" is DVB-S2, "-M 5" is 8PSK modulation, "-C 23" is FEC 2/3

IV. here are some basic instructions how to scan transponder with "scan-s2":

IV.1 extract "scan-s2.tar.bz2" archive:

```
# tar xjvf scan-s2.tar.bz2
```

IV.2 go to scan-s2 source code directory:

```
# cd scan-s2
```

IV.3 build scan-s2:

```
# make
```

IV.4 scan-s2 uses configuration files (see configuration file "astra_scan-s2.conf" for example) - they are simple text files and each line of them contains information about a transponder, for example:

```
S1 11778000 V 27500000 3/4 35 QPSK
S2 11362000 H 22000000 2/3 35 8PSK
```

where:

- "S1" or "S2" indicates the delivery system, i.e. either DVB-S or DVB-S2
- second parameter is the frequency, in our example respectively 11778000 kHz or 11362000 kHz
- "V" or "H" indicates that the polarization: V(ertical) or H(orizontal)
- 5th parameter is the FEC value, in our example respectively 3/4 or 2/3
- the next parameter is the Roll-off factor, "35" in our example
- and the last parameter is the modulation, in our example respectively QPSK or 8PSK

the Roll-off factor and the modulation are mandatory for DVB-S2 cards and without them the result is just unpredictable, most probably - no lock. the reason for that is that DVB-S2 supports several modulation like QPSK, 8PSK, etc and 3 Roll-off factors : 0.20, 0.25, 0.35 (respectively 20, 25, 35 in the configuration file). DVB-S supports only one Roll-off factor - 0.35 and so for DVB-S transponders Roll-off factor is always 35 in the configuration file

IV.5 write your scan-s2 configuration files

IV.6 use scan-s2 to scan a transponder