# The Silicon Labs Si470x FM Radio Receivers driver



## Information from Silicon Labs

Silicon Laboratories is the manufacturer of the radio ICs, that nowadays are the most often used radio receivers in cell phones. Usually they are connected with I2C. But SiLabs also provides a reference design, which integrates this IC, together with a small microcontroller C8051F321, to form a USB radio. Part of this reference design is also a radio application in binary and source code. The software also contains an automatic firmware upgrade to the most current version. Information on these can be downloaded here: http://www.silabs.com/usbradio

## Supported ICs

The following ICs have a very similar register set, so that they are or will be supported somewhen by the driver:

- Si4700: FM radio receiver
- Si4701: FM radio receiver, RDS Support
- Si4702: FM radio receiver
- Si4703: FM radio receiver, RDS Support
- Si4704: FM radio receiver, no external antenna required
- Si4705: FM radio receiver, no external antenna required, RDS support, Dig I/O
- Si4706: Enhanced FM RDS/TMC radio receiver, no external antenna required, RDS Support
- Si4707: Dedicated weather band radio receiver with SAME decoder, RDS Support
- Si4708: Smallest FM receivers
- Si4709: Smallest FM receivers, RDS Support

More information on these can be downloaded here: http://www.silabs.com/products/mcu/Pages/USBFMRadioRD.aspx

## Supported USB devices

Currently the following USB radios (vendor:product) with the Silicon Labs si470x chips are known to work:

- 10c4:818a: Silicon Labs USB FM Radio Reference Design
- 06e1:a155: ADS/Tech FM Radio Receiver (formerly Instant FM Music) (RDX-155-EF)
- 1b80:d700: KWorld USB FM Radio SnapMusic Mobile 700 (FM700)
- 10c5:819a: Sanei Electric, Inc. FM USB Radio (sold as DealExtreme.com PCear)

## Software

Testing is usually done with most application under Debian/testing:

- fmtools - Utility for managing FM tuner cards
- gnomeradio - FM-radio tuner for the GNOME desktop
- gradio - GTK FM radio tuner
- kradio - Comfortable Radio Application for KDE
- radio - ncurses-based radio application
- mplayer - The Ultimate Movie Player For Linux
- v4l2-ctl - Collection of command line video4linux utilities

For example, you can use:

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\[linux-master][Documentation][admin-guide][media]si470x.rst, line 73)

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

        v4l2-ctl -d /dev/radio0 --set-ctrl=volume=10,mute=0 --set-freq=95.21 --all
```

There is also a library libv4l, which can be used. It's going to have a function for frequency seeking, either by using hardware functionality as in radio-si470x or by implementing a function as we currently have in every of the mentioned programs. Somewhen

the radio programs should make use of libv4l.

For processing RDS information, there is a project ongoing at: http://rdsd.berlios.de/

There is currently no project for making TMC sentences human readable.

## Audio Listing

USB Audio is provided by the ALSA snd_usb_audio module. It is recommended to also select SND_USB_AUDIO, as this is required to get sound from the radio. For listing you have to redirect the sound, for example using one of the following commands. Please adjust the audio devices to your needs (/dev/dsp* and hw:x,x).

If you just want to test audio (very poor quality):

**System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\[linux-master][Documentation][admin-guide][media]si470x.rst, line 98)**

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

        cat /dev/dsp1 > /dev/dsp
```

If you use sox + OSS try:

**System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\[linux-master][Documentation][admin-guide][media]si470x.rst, line 104)**

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

        sox -2 --endian little -r 96000 -t oss /dev/dsp1 -t oss /dev/dsp
```

or using sox + alsa:

**System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\[linux-master][Documentation][admin-guide][media]si470x.rst, line 110)**

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

        sox --endian little -c 2 -S -r 96000 -t alsa hw:1 -t alsa -r 96000 hw:0
```

If you use arts try:

**System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\[linux-master][Documentation][admin-guide][media]si470x.rst, line 116)**

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

        arecord -D hw:1,0 -r96000 -c2 -f S16_LE | artsdsp aplay -B -
```

If you use mplayer try:

**System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\[linux-master][Documentation][admin-guide][media]si470x.rst, line 122)**

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none
```

```
mplayer -radio adevice=hw=1.0:arate=96000 \
        -rawaudio rate=96000 \
        radio://<frequency>/capture
```

## Module Parameters

After loading the module, you still have access to some of them in the sysfs mount under /sys/module/radio_si470x/parameters. The contents of read-only files (0444) are not updated, even if space, band and de are changed using private video controls. The others are runtime changeable.

## Errors

Increase tune_timeout, if you often get -EIO errors.

When timed out or band limit is reached, hw_freq_seek returns -EAGAIN.

If you get any errors from snd_usb_audio, please report them to the ALSA people.

## Open Issues

V4L minor device allocation and parameter setting is not perfect. A solution is currently under discussion.

There is an USB interface for downloading/uploading new firmware images. Support for it can be implemented using the request_firmware interface.

There is a RDS interrupt mode. The driver is already using the same interface for polling RDS information, but is currently not using the interrupt mode.

There is a LED interface, which can be used to override the LED control programmed in the firmware. This can be made available using the LED support functions in the kernel.

## Other useful information and links

http://www.silabs.com/usbradio