

osmo-fl2k

- **Table of contents**
- [osmo-fl2k](#)
 - [Introduction](#)
 - [History](#)
 - [Supported Hardware](#)
 - [USB 3.0 VGA Dongles](#)
 - [USB 3.0 to HDMI Dongles](#)
 - [USB Host Controller comparison](#)
 - [PCI Express bottleneck on older systems](#)
 - [Download](#)
 - [Source Code](#)
 - [Compiling](#)
 - [Using the Software](#)
 - [Realtime WBFM transmission](#)
 - [GSM transmission](#)
 - [Presentation Slides / Recording](#)
 - [Contact / Mailing List](#)
 - [Links](#)
 - [Legal Aspects / Keeping the RF Spectrum Clean](#)

Introduction

Since [rtl-sdr](#) has been released a couple of years ago, cheap SDR receivers are ubiquitous. SDRs with transmission capability have become cheaper as well, but are still more expensive. osmo-fl2k allows to use USB 3.0 to VGA adapters based on the Fresco Logic FL2000 chip, which are available for around \$5, as general purpose DACs and SDR transmitter generating a continuous stream of samples by avoiding the HSYNC and VSYNC blanking intervals.

History

The idea of using VGA adapters as a transmitter has a long history.

- In 2001 Erik Thiele released [Tempest for Eliza](#), which transmits a signal that can be received with an AM radio.
- In 2005 Fabrice Bellard used an ATI Radeon 9200SE to transmit [DVB-T, PAL and NTSC video signals](#).
- Later in 2009, Bartek Kania published [VGASIG](#), which allows to transmit wideband FM radio in realtime using SDL and a regular graphics card.
- In 2013, siro at das Labor used all three DAC outputs of a VGA card and [attached an I/Q modulator to transmit arbitrary signals](#).

However, a major downside with regular graphics cards is that the generated signal of the RAMDAC is not entirely user controllable, as VGA uses [horizontal and vertical blanking](#), thus interrupting the signal. In comparison to competitor devices from DisplayLink, which resemble a

classic graphics card with USB 2.0/3.0 interface, Fresco Logic took a different approach for their USB to VGA adapter, which they've also [patented](#). Instead of having the framebuffer for the VGA DAC in the device itself, they use the memory of the host computer and continuously stream the display content via USB. This makes such adapters very cheap, as it essentially reduces the adapter to a single chip without requiring framebuffer memory. The main drawback is that if the host CPU or USB bus is busy or congested, this results in dropouts and flickering of the screen.

After reverse-engineering the USB protocol of the FL2000 in 2016, Steve Markgraf discovered through experimentation that it is possible to operate the FL2000 in a way that both horizontal and vertical synchronization are disabled, thus creating a continuous stream of samples. This work resulted in osmo-fl2k, which so far was used to transmit low-power FM, DAB, DVB-T, GSM, UMTS and GPS signals.

Supported Hardware

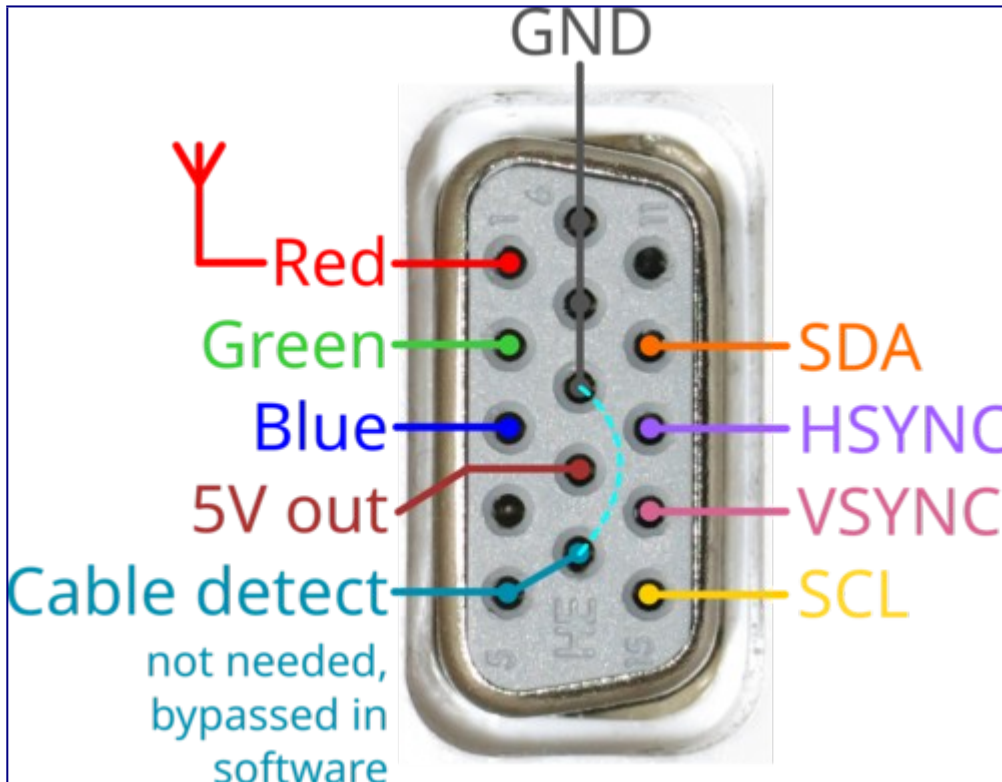
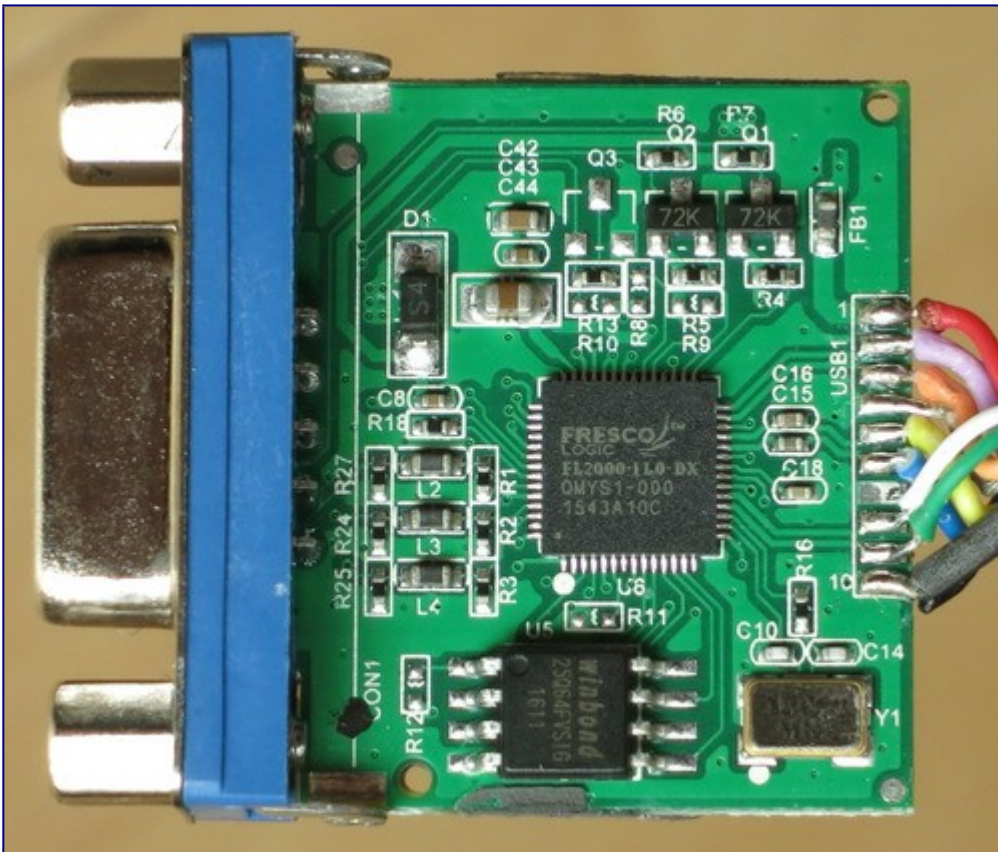
USB 3.0 VGA Dongles

Only devices based on the Fresco Logic FL2000 chip are supported. Typically they are branded as "USB 3.0 to VGA" adapters, and are advertised with a maximum resolution of 1920×1080 for USB 3.0 and 800×600 for USB 2.0. Sometimes also 1920×1200 is advertised as maximum resolution (this mode only works with few xHCI controllers and enabled RLE compression).

If they are advertised with 2048×1152 maximum resolution and support for Mac OS X, or only have a USB 2.0 interface, they contain a DisplayLink chipset and are **not** compatible with osmo-fl2k! The price range for the FL2000-based adapters is \$5-15, whereas the DisplayLink devices typically cost more than \$25. Also note that devices sold with USB type C connector contain a different chipset (e.g. Realtek RTD2166) and are just DisplayPort to VGA converters.

Sources include Aliexpress, eBay and Amazon. The devices are available in various sizes, sometimes also with additional DVI or HDMI support. Two of the most typical devices are shown below.



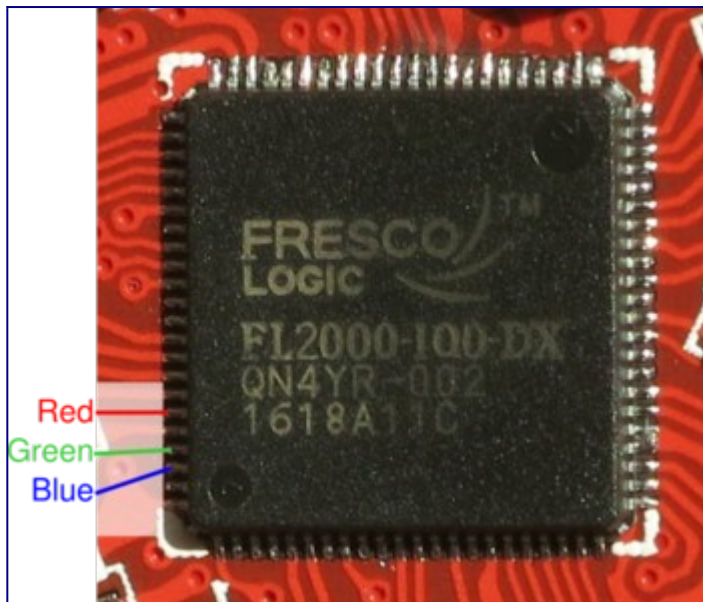


USB 3.0 to HDMI Dongles

If you only can get your hands on a USB 3.0 to HDMI adapter with FL2000-1Q0-DX (bigger 88-pin package with parallel RGB output), and you're not shy of soldering wires to QFN packages,

below is the pinout of the VGA DAC outputs.

All the FL2000-based USB 3.0 HDMI adapters seem to have the 1.2V regulator populated, so it's probably used for more than the DAC reference voltage.



USB Host Controller comparison

The SDR/DAC performance in Megasamples per second depends on the xHCI controller and the host computer. Below are some benchmarking results for your reference.

You can benchmark your USB host controller by running:

```
fl2k_test -s 162e6
```

Then you will get something like this:

```
real sample rate: 157023826 current PPM: -30717 cumulative PPM: -30717
```

In this example, the system cannot deliver more than 157 MS/s to the device (*3 DAC channels = 471 MB/s). You can then run fl2k_test again with a sample rate lower than what was determined as maximum to see if it works stable and to determine your clock error.

USB 3.0 Host controller	Achievable sample rate
Advanced Micro Devices, Inc. X470 chipset USB 3.1 Gen 2	157.86 MS/s
Intel Corporation 8 Series USB xHCI HC (rev 04)	157 MS/s
Intel Corporation 7 Series/C210 Series Chipset Family USB xHCI Host Controller (rev 04)	157 MS/s
Intel Corporation Wildcat Point-LP USB xHCI Controller (rev 03)	157 MS/s
Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)	155 MS/s
Samsung Galaxy S5 msm-dwc3 (based on Synopsys DesignWare USB3.0 IP)	148 MS/s
Odroid XU4 (based on Synopsys DesignWare USB3.0 IP)	147 MS/s
ASMedia Technology Inc. ASM1142 USB 3.1 Host Controller	140 MS/s
Fresco Logic FL1100 USB 3.0 Host Controller (rev 10)	140 MS/s

Fresco Logic FL1100 USB 3.0 Host Controller (rev 01)	136 MS/s
Fresco Logic FL1009 USB 3.0 Host Controller (rev 02)	136 MS/s
Etron Technology, Inc. EJ188/EJ198 USB 3.0 Host Controller	129 MS/s
VIA Technologies, Inc. VL805 USB 3.0 Host Controller (rev 01)	123 MS/s
Etron Technology, Inc. EJ168 USB 3.0 Host Controller (rev 01)	117 MS/s
Renesas Technology Corp. uPD720202 USB 3.0 Host Controller (rev 02)	115 MS/s
Intel USB 2.0 EHCI Controller	14 MS/s

PCI Express bottleneck on older systems

If you are using USB 3.0 add-on cards in older systems, especially in ExpressCard slots of notebooks, it often happens that the bottleneck is not the USB 3.0 host controller, but the PCI-E bus itself.

You can check this with the following command:

```
# lspci -vv
```

```
04:00.0 USB controller: Renesas Technology Corp. uPD720202 USB 3.0 Host
Controller (rev 02) (prog-if 30 [XHCI])
[...]
                LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L0s L1, Exit
Latency L0s <4us, L1 unlimited
                ClockPM+ Surprise- LLActRep- BwNot- ASPMOptComp-
                LnkSta: Speed 2.5GT/s, Width x1, TrErr- Train- SlotClk+
DLActive- BWMgmt- ABWMgmt-
                LnkCtl2: Target Link Speed: 5GT/s, EnterCompliance- SpeedDis-
```

The important part is the LnkSta register: in this case the PCI Express link is running only with 2.5 GT/s instead of the full 5 GT/s required for maximum USB 3.0 speed.

Download

Source Code

Source code is available from git.osmocom.org

- web access: <http://git.osmocom.org/osmo-fl2k>
- read-only clone access: `git clone git://git.osmocom.org/osmo-fl2k`

Compiling

In order to compile and install the code, you first need to install the libusb headers if not already present:

```
# On distributions based on debian
sudo apt-get install libusb-1.0-0-dev
```

Then you can continue with:

```
git clone git://git.osmocom.org/osmo-fl2k.git
mkdir osmo-fl2k/build
cd osmo-fl2k/build
cmake ../ -DINSTALL_UDEV_RULES=ON
make -j 3
```

```
sudo make install
sudo ldconfig
```

Before being able to use the device as a non-root user, the udev rules need to be reloaded:

```
sudo udevadm control -R
sudo udevadm trigger
```

Using the Software

Realtime WBFM transmission

With fl2k_fm it is possible to transmit wideband FM radio of your soundcard output.

You need to install the following additional tools:

```
sudo apt-get install sox pv
```

First, you need to know the name of the monitor source of your audio output:

```
# pacmd list-sources | grep name | grep monitor
name: <alsa_output.pci-0000_00_1b.0.analog-stereo.monitor>
```

To transmit the audio output as mono WBFM at 95 MHz, you can use the following command:

```
pacat -r -d alsa_output.pci-0000_00_1b.0.analog-stereo.monitor | \
pv -B 256k | \
sox -t raw -r 44100 -e signed-integer -L -b 16 -c 2 - -c 1 -e signed-integer -b
16 -t raw - \
biquad 4.76963 -2.98129 0 1 0.78833 0 sinc -15k loudness 5 | \
fl2k_fm - -s 130e6 -c 35e6 -i 44100
```

In this case, SoX is performing the 15 kHz lowpass filtering of the audio, stereo to mono conversion and FM pre-emphasis (with the biquad filter).

pv is providing some additional buffering, but is optional.

To transmit stereo WBFM with RDS, the following command can be used:

```
pacat -r -d alsa_output.pci-0000_00_1b.0.analog-stereo.monitor | \
pv -B 256k | \
sox -t raw -r 44100 -e signed-integer -L -b 16 -c 2 - -c 2 -e signed-integer -b
16 -t raw - \
biquad 4.76963 -2.98129 0 1 0.78833 0 sinc -15k rate -h 228000 loudness 5 | \
fl2k_fm - -s 130e6 -c 30e6 -i 228000 --stereo --rds
```

Now SoX also re-samples the audio to 228 kHz, which is required when using the RDS modulator. fl2k_fm accepts 16 bit signed raw PCM audio samples.

GSM transmission

The GNU Radio flowgraphs as well as baseband samples required to transmit GSM and UMTS can be found at <https://github.com/steve-m/fl2k-examples>.



Presentation Slides / Recording

On April 22nd, 2018, Steve Markgraf publicly released osmo-fl2k at OsmoDevCon.

- slides of the presentation available at http://people.osmocom.org/steve-m/fl2k_slides/osmo-fl2k.html
- video recording released at <https://media.ccc.de/v/HWVZPK>

A second presentation was held at OsmoCon 2018

- video recording released at <https://media.ccc.de/v/osmocon2018-83-osmo-fl2k-the-evil-transmit-side-brother-of-rtl-sdr>

Contact / Mailing List

All questions should be directed to the public osmocom-sdr@lists.osmocom.org mailing list, to which you can subscribe at <https://lists.osmocom.org/mailman/listinfo/osmocom-sdr>

Links

PCB pictures and reverse-engineered schematics: <https://gitlab.hamburg.ccc.de/MarBle/fl2k-liberation>

Analogue TV transmitter with osmo-fl2k support: <https://github.com/fsphil/hacktv>

Legal Aspects / Keeping the RF Spectrum Clean

If you are operating a radio transmitter of any sort, particularly a DIY solution or a SDR transmitter, it is assumed that you are familiar with both the legal aspects of radio transmissions, as well as the technical aspects of it. **Do not operate a radio transmitter unless you are clear of the legal and regulatory requirements.** In most jurisdictions the operation of homebrew / DIY transmitters requires at the very least an amateur radio license.

The raw, unfiltered DAC output contains lots of harmonics at multiples of the base frequency. This is what is creatively (ab)used if you use osmo-fl2k to generate a signal much higher than what you could normally achieve with a $\sim 165\text{MHz}$ DAC without upconversion. However, this means that the frequency spectrum will contain not only the one desired harmonic, but all the lower harmonics as well as the base frequency.

Before transmitting any signals with an FL2000 device, it is strongly suggested that you check the resulting spectrum with a spectrum analyzer, and apply proper filtering to suppress any but the desired transmit frequency.

Operating a transmitter with the unfiltered FL2000 DAC output attached to an antenna outside a RF shielding chamber is dangerous. Don't do it!

You have the following alternatives to broadcasting over the air:

- add proper output band pass filtering for your desired TX frequency, or
- operate transmitter + receiver in a shielded RF box / chamber, or
- connect transmitter + receiver over coaxial cable (with proper attenuators)