

Software to use ELAD devices with Linux/Gnuradio

Devices: S1, S2, DUO

Introduction:

S1 and S2 devices need an initial internal FPGA programming at power-up. This operation is made possible by the closed-source program *elad-firmware*.

The source package *gr-elad-all.tgz* allows the insertion of one or more ELAD S1,S2,DUO devices into a Gnuradio flowgraph.

Elad-firmware usage:

When you launch *elad-firmware* program without parameters, you obtain a message that reminds its usage:

```
Usage ./elad-firmware SERIAL SPEED
      SERIAL='+' or none => first available device
      ( SERIAL='+' must be used when S1 programming sample rate is needed )
      SPEED={1 .. 6} => {192kS/s .. 6.144 MS/s} for S1
      SPEED fixed to 192 kS/s for S2
```

Then, the program tries to program a device found according the following procedure:

Program scans the USB devices list, looking for idVendor 1721

The device is opened and its serial was read, verifying the it is equal to the SERIAL given as a program parameter (this control is successful for default if there is no SERIAL parameter, or if it is a simple “+”)

If the control is successful the device will be programmed according to the requested sample rate.

When the program is launched without parameters, it programs the first ELAD device it is able to find, programming the default sample rate (192kS/s). This is handy for people owning a single device.

People who need to connect more devices must use SERIAL to point to the right device. The *elad-firmware* program must be launched one time for every device to program.

If the serial of the device was forgotten it can be read launching the *elad-firmware* program with an “impossible” serial: this forces the program to try all the devices and to display their serial as in this example:

```
./elad-firmware unknown_serial
Operations init elad-firmware version 1.0
Usage ./elad-firmware SERIAL SPEED
      SERIAL='+' or none => first available device
      ( SERIAL='+' must be used when S1 programming sample rate is needed )
      SPEED={1 .. 6} => {192kS/s .. 6.144 MS/s} for S1
      SPEED fixed to 192 kS/s for S2
0 vid 05e3 pid 0732 0
1 vid 1d6b pid 0003 0
```

```

2 vid 1721 pid 061c 0
FDM Device Opened
Interface claimed
FDM S2 Serial: SF02QF
USB serial (SF02QF) does not match given serial (unknown_serial)
3 vid 16c0 pid 05dc 1
4 vid 0b05 pid 1825 1
5 vid 04d9 pid 1503 1
6 vid 06a2 pid 0033 1
7 vid 1721 pid 0610 1
FDM Device Opened
Interface claimed
FDM S1 Serial: SC0EOU_0003T
USB serial (SC0EOU_0003T) does not match given serial (unknown_serial)
8 vid 1d6b pid 0002 2
Cannot find FDM Device

```

Obviously the devices are not programmed so we need to launch the program with the needed serial:

```

./elad-firmware SF02QF
Operations init elad-firmware version 1.0
Usage ./elad-firmware SERIAL SPEED
    SERIAL='+' or none => first available device
    ( SERIAL='+' must be used when S1 programming sample rate is needed )
    SPEED={1 .. 6} => {192kS/s .. 6.144 MS/s} for S1
    SPEED fixed to 192 kS/s for S2
0 vid 05e3 pid 0732 0
1 vid 1d6b pid 0003 0
2 vid 1721 pid 061c 0
FDM Device Opened
Interface claimed
FDM S2 Serial: SF02QF
libusb devices list freed
S2 selected 192kS/s
USB Driver Version: 3.0
HW Version: 1.3
Serial: SF02QF
ADC set
External filter set
XYLINX reset
programming XYLINX: wait 1 second
programming XYLINX:.....
final programming XYLINX (802294)
XYLINX programming complete (802295)
XYLINX programming correct
read XYLINX status (02)

```

At that time the device is programmed.

Note: Some serial have “leading blanks”. In this case the SERIAL parameter must be enclosed into double quotes as in the example:

```
./elad-firmware "SE0COD "
```

Source package gr-elad-all.tgz usage:

The package must be expanded with the command `tar xvzf gr-elad-all.tgz` to obtain a `gr-elad` folder. In this folder it is a text file `INSTALL` with the instructions.

Remember that you need, at least, `gnuradio`, `gnuradio-dev`, `libusb1.0-dev` packages.

Once completed the installation you can use the `gnuradio-companion` program to build a flowgraph including your device.

In the right list you can see an “ELAD” tag with inside The “FDM Source” element. The element allows you to link any S1,S2,DUO device to the flowgraph.

In the example you can see a flowgraph with the “FDM Source” initial setup opened, showing how to setup Frequency, attenuator, filter, serial.

Please note that, in this case, the trailing spaces in a serial do not involve the double-quotes usage.

The screenshot shows the GNU Radio Companion interface. A dialog box titled "Properties: FDM Source" is open, showing the following settings:

- General tab selected
- ID: elad_fdm_source_c_3
- LO frequency: 14180000
- Filters: LP(S1/DUO)/From hv
- Input Attenuator: Bypass
- Serial: LP(S1/DUO)/From hwLO(S2)
- Resampling: 192000

The flowgraph shows the following components and connections:

- FDM Source** (LO frequency: 14.18M, Serial: SF020F) feeds into a **WX GUI FFT Sink** (Sample Rate: 192k, Baseband Freq: 0, Y per Div: 10 dB, Y Divs: 10, Ref Level (dB): -20, Ref Scale (p2p): 2, FFT Size: 1.024k, Refresh Rate: 15).
- The FDM Source also feeds into a **WX GUI Scope Sink** (Title: I & Q channels, Sample Rate: 192k, Y Scale: 5m, Window Size: 250, 100, Grid Position: 0, 0, 1, 1, Trigger Mode: Normal, Y Axis Label: Counts).
- The FDM Source feeds into a **WX GUI Constellation Sink** (Title: Constellation Plot, Sample Rate: 192k, Frame Rate: 1.5k, Constellation Size: 512, M: 4, Theta: 0, Loop Bandwidth: 3.2m, Max Freq: 60m, Mu: 500m, Gain Mu: 2, Symbol Rate: 3k, Omega Limit: 5m, Window Size: 250, 250, Grid Position: 1, 0, 2, 1).
- The FDM Source feeds into a **WX GUI FFT Sink** (Title: FFT Plot, Sample Rate: 192k, Baseband Freq: 0, Y per Div: 10 dB, Y Divs: 10, Ref Level (dB): 0, Ref Scale (p2p): 2, FFT Size: 1.024k, Refresh Rate: 15).
- The FDM Source feeds into a **Multiply** block, which then feeds into a **Complex To Float** block, which finally feeds into an **Audio Sink** (Sample Rate: 48KHz, Device Name: hw1.0).

The right sidebar shows a list of components, including "FDM Source". The bottom terminal shows the following output:

```
>>> Done
Showing: "/home/giovanni/Desktop/WORK/ELAD/DUO/fourphases/fourphases.grc"
Loading: "/home/giovanni/Desktop/WORK/ELAD/ALL/gr-elad-completo/examples/fourphases.grc"
>>> Done
Showing: "/home/giovanni/Desktop/WORK/ELAD/ALL/gr-elad-completo/examples/fourphases.grc"
```