<u>Other artic</u> Share: <u>Fac</u> By browsing further, you agree to the use of cookies and similar technologies by this site and its partners to personalize ads and analyze traffic. <u>More information</u> | <u>Preferences</u> | <u>OK</u>

LeanTRX: A web interface for embedded SDR platforms

*** WORK IN PROGRESS ***

LeanTRX adds a web interface to the LeanSDR toolkit. Its main purpose is to turn embedded SDR platforms into user-friendly, multi-mode RF modems which can be controlled with any web browser. Potential applications range from handheld transceivers to Internet-connected networks of ground stations.



Table of Contents

1. Introduction 2. Related work

> 2.1. Other web-based SDR projects 2.2. Other user interfaces for LeanSDR

3. Features

3.1. Multi-channel FM receiver 3.2. DVB-S demodulator 3.3. DVB-S beacon

4. Usage
5. Benefits
6. Known problems and limitations
7. Troubleshooting
8. WebRTC support
9. Implementation details
10. Development
11. PlutoSDR support

11.1. As a network-connected RF modem

11.1.1. Requirements 11.1.2. Installation

11.2. As a RF modem for a desktop PC or a laptop



<u>11.2.1. Requirements</u> 11.2.2. Installation

11.3. As a RF modem for a smartphone

<u>11.3.1. Requirements</u> <u>11.3.2. Installation</u> <u>11.3.3. Variant with USB OTG cable</u>

- 11.4. Permanent installation
- 12. Generic Linux support 13. Android support 14. Future work Bibliography

List of Figures

- 1. PlutoSDR, OTG hub, Ethernet
- 2. PlutoSDR as USB peripheral
- 3. PlutoSDR, OTG hub, USB Tethering
- 4. PlutoSDR, OTG cable, USB Tethering
- 5. PlutoSDR, OTG splitter cable, USB Tethering

List of Tables

- 1. Targeted platforms
- 2. WebRTC capabilities

1. Introduction

LeanTRX aims to provide a user-friendly way to perform SDR tasks ranging from traditional audio communications to digital modes, amateur TV, satellite telemetry and wide-scale collaborative experimentation.

The underlying SDR functions are provided by <u>LeanSDR</u>, a toolkit of command-line tools which requires proficiency with signal processing, UNIX shell tricks, and command pipelines. LeanTRX wraps them inside CGI scripts and adds sensible default parameters for popular activities.

Both LeanSDR and LeanTRX are suitable for embedded devices with limited resources. External software dependencies are minimized. Development focuses on the ADALM-PLUTO (PlutoSDR), but other platforms should be able to run it eventually.

Platform	CPU	Clock	RAM	SDR	OS
PlutoSDR	Zynq ARMv7 1(2?) cores	667 MHz	512 MiB (256 MiB CMA buffers)	Built-in AD9363+FPGA	Buildroot
Dreamcatcher 2	Allwinner ARMv7	1 GHz	512 MiB	Built-in RTL-SDR	Debian
Raspberry Pi	Broadcom ARMv6/v7/v8 1- 4 cores	700- 1200 MHz	512-1024 MiB	RTL-SDR, LimeSDR	Raspbian

Table 1. Targeted platforms

2. Related work

2.1. Other web-based SDR projects

WebSDR and OpenWebRX have already demonstrated Internet-connected radios with web-based interfaces. Among other benefits, they allow multiple concurrent users to remotely control high-performance receivers installed at favourable locations. For obvious reasons, the publicly-accessible devices do not allow transmission. By contrast, LeanTRX focuses on low-power SDR devices and on single-user applications, including two-way modes.

Current-generation web SDRs typically use WebSockets and the Web Audio API for audio streaming and playback. Instead, LeanTRX will use WebRTC, a newer framework which also supports video playback and capture.

2.2. Other user interfaces for LeanSDR

The <u>rpidatv</u> software by Evariste F5OEO includes a native user interface and wrapper scripts for running LeanDVB on Raspberry Pi platforms. LeanTRX supports LeanDVB in a similar way and adopts a web-based interface for portability and network transparency.

3. Features

3.1. Multi-channel FM receiver

FFT-based channelizer. Can demodulate 30..40 WFM mono audio channels simultaneously on a modified PlutoSDR. Can monitor the whole PMR446 band, the whole amateur 2 m band, etc.

3.2. DVB-S demodulator

Realtime constellation and spectrum display. Currently outputs TS files. Future version will stream TS over IP and hopefully decode video inside the user's browser.

Non-standard constellations for experimentation.

3.3. DVB-S beacon

Currently transmits a test pattern only (not valid MPEG). Future version will hopefully capture video from the user's smartphone or webcam.

4. Usage

For installation instructions, see the platform-specific sections of this document:

• <u>Section 11, " PlutoSDR support "</u>

Once the software is running:

- From the main page http://xxx.xxx.xxx/leantrx/index.html, select an activity.
- Review and customize the configuration parameters.
- Activate "ENABLE SDR", "ENABLE UI", "ENABLE AUDIO" as required.

After modifying configuration parameters, toggle "ENABLE SDR" OFF then ON to apply changes.

There is no provision for saving configuration data. Bookmarking URLs with modified parameters achieves the same effect.

5. Benefits

- **Network transparency**. The UI works over any IP connection with sufficient bandwidth. Typical use cases are:
 - Portable device controled by smartphone over USB OTG cable
 - Remote Radio Head with Ethernet or fiber connection.
 - Remote operation over an Internet VPN.
- **Portable user interface**. The software can be controlled from any desktop, laptop, tablet or smartphone with an IP connection to the device and a reasonably modern web browser. Besides, an Android smartphone can connect to the PlutoSDR directly with a USB OTG cable; this does not require rooting.
- Customizable user interface . The UI can be restyled simply by editing CSS and HTML files.
- **Modularity**. Third-party SDR functions provided as command-line tools can be inserted into processing pipelines by editing the CGI scripts. This can be used to implement new modulations or take advantage of hardware accelerators not supported by LeanSDR.
- **Scalability** . LeanTRX favours processing signals as close to the SDR hardware as possible, but CPU-intensive functions can be offloaded to remote servers by editing the CGI scripts.
- **Predictable performance** . A dedicated embedded platform with a stripped-down OS can provide consistent performance for real-time applications. By contrast, general-purpose computers tend to exhibit variable latency and processing capacity.

6. Known problems and limitations

- Tested with Firefox, Chrome and webkit-based browsers. Not tested with iOS.
- Audio playback has high latency and will hang after about 90 min. This will improve when the current implementation (based on HTTP polling) is replaced with multimedia streaming protocols.
- QPSK RF output is not properly bandpass-filtered. This will be fixed by optimizing the interpolator or by enabling filtering in the FPGA.

- LeanTRX must not be exposed to an untrusted network. The UI does not have access control, the CGI scripts have not been audited for security, and connections are not encrypted. The PlutoSDR has a hardcoded root password.
- By default the web-based user interface has a slow refresh rate. This preserves CPU capacity for signal processing. Adjust the "UI refresh rate" parameter for a smoother display.
- USB Tethering will fail with smartphones that don't use use 192.168.42.0/24 for RNDIS networking.

7. Troubleshooting

- Examine the process console at the bottom of the page.
- Examine the Javascript console of the browser.
- Log into the embedded device to monitor CPU load, memory usage and running processes.
- If the PlutoSDR randomly becomes unresponsive, try again with a better power supply and shorter USB cables.
- If the PlutoSDR does not load LeanTRX from USB Mass Storage (the LED does not flash), repair the partition with fsck or an equivalent Windows/Mac utility. Alternatively, install LeanTRX manually over ssh as in Section 11.2, "As a RF modem for a desktop PC or a laptop ".
- A smartphone may try to draw too much current from the PlutoSDR over USB OTG. Charge the phone fully before experimenting, or use a powered USB hub.
- Inexpensive SDR devices often have oscillators with poor frequency accuracy. This can affect demodulation of narrowband modes, including FM in the amateur bands and the PMR446 band.

The PlutoSDR will apply frequency corrections after the real frequency of the oscillator has been provided with this command:

```
iio_attr -d ad9361-phy xo_correction 40000000 # Nominal: 40 MHz
```

• Many embedded platforms are powered by USB supplies, which can be very noisy. Even with a linear lab power supply, reception can be affected by long wires acting as antennas and by noisy grounds. If in doubt, use a battery pack and a linear voltage regulator.

8. WebRTC support

Work in progress.

Table 2. WebRTC capabilities

Browser	Platform	Audio	Video
Chrome 44.0.2403.133	Android 6.0.1 ARM tablet	-	-
Chrome 62.0.3202.84	Android 7.0 ARM smartphone	UDP/TLS/RTP/SAVPF opus/48000/2 ISAC/16000 {G722,PCMU,PCMA}/8000 CN/{16000,8000} telephone- event/{48000,16000,8000}	UDP/TLS/RTP/SAVPF {VP8,VP9,H264}/90000 red,rtx,ulpfec
Chrome 63.0.3239.11	Android 5.0.1 x86 tablet	UDP/TLS/RTP/SAVPF opus/48000/2 ISAC/{16000,32000} {G722,PCMU,PCMA}/8000 CN/{32000,16000,8000} telephone- event/{48000,32000,16000,8000}	UDP/TLS/RTP/SAVPF {VP8,VP9}/90000 red,rtx,ulpfec
Chromium 63.0.3239.108	GNU/Linux x86_64	UDP/TLS/RTP/SAVPF opus/48000/2 ISAC/{16000,32000} {G722,PCMU,PCMA}/8000 CN/{32000,16000,8000} telephone- event/{48000,32000,16000,8000}	UDP/TLS/RTP/SAVPF {VP8,VP9}/90000 red,rtx,ulpfec
Firefox 49.0.2 GNU/Linux x8		UDP/TLS/RTP/SAVPF opus/48000/2 G722/8000/1 {PCMU,PCMA}/8000 telephone-event/8000/1	UDP/TLS/RTP/SAVPF VP8/90000
Firefox 57.0.4 GNU/Linux x86_64		UDP/TLS/RTP/SAVPF opus/48000/2 G722/8000/1 {PCMU,PCMA}/8000 telephone-event/8000/1	UDP/TLS/RTP/SAVPF {VP8,VP9}/90000
Firefox 58.0 GNU/Linux x86_64		UDP/TLS/RTP/SAVPF opus/48000/2 G722/8000/1 {PCMU,PCMA}/8000 telephone-event/8000/1	UDP/TLS/RTP/SAVPF {VP8,VP9}/90000

9. Implementation details

A SDR function in LeanTRX typically comprises a HTML page, a Javascript file, a CGI script, and command-line utilities from LeanSDR. For example, the DVB receiver is implemented as follows:

- leantrx/html/dvbrx.html defines the layout of the user interface.
- leantrx/cgi-bin/dvbrx.cgi spawns the command pipeline which performs the actual SDR work on the embedded device, from IQ capture to demodulation to network streaming.
- leantrx/html/dvbrx.js specifies configuration parameters, how to pass them to the CGI script, how to communicate with the processes running on the device, and how to render the UI.

The main web page, leantrx/html/index.html, contains several links to dvbrx.html, each with its own set of configuration values passed as URL search parameters.

leantrx/html/stdui.js contains helper functions for a simple use case where leansdrserv forwards status data to the browser in JSON format over HTTP.

10. Development

How to rebuild from sources and package for PlutoSDR:

```
mkdir build
cd build
git clone https://github.com/pabr/leansdr.git
git clone https://github.com/pabr/leantrx.git
cd leantrx
make ARCH=armv7l leansdr # On a Raspberry Pi, or with a cross-compiler.
make plutosdr
```

11. PlutoSDR support

11.1. As a network-connected RF modem

This configuration is the easiest way to test LeanTRX. It does not require any UNIX proficiency.

LeanTRX will reload automatically from USB storage whenever the PlutoSDR reboots. The UI can be first accessed over Ethernet from a desktop browser (easier to troubleshoot than a mobile browser), and later from a smartphone or tablet over WiFi and Ethernet. Simultaneously, one can log into the PlutoSDR over SSH for debugging.

Figure	1.]	Pluto	SDR,	ОТG	hub,	Ethe	rnet

11.1.1. Requirements

- PlutoSDR
- USB OTG adapter (Micro-B plug to Type-A receptacle)
- USB hub
- USB-Ethernet adapter (USB-WiFi may work; not tested)
- USB Mass Storage device (USB thumb drive or SD card in USB reader)
- USB power supply with Micro-B output.

An *integrated OTG/hub/Ethernet adapter* conveniently combines several of these functions into a single device.



11.1.2. Installation

- On a computer, download <u>leantrx-20180211.zip</u> and unzip it to the USB drive. Confirm that the root of the USB drive contains a file named runme-leantrx and a directory named leantrx.
- Connect the USB drive, network adapter and USB hub to the OTG port of the PlutoSDR.
- Apply power to the PlutoSDR. The blue LED should emit a rapid succession of flashes within 10 s.
- With a web browser, open http://192.168.2.1/leantrx/. The actual IP address is assigned by a DHCP server on your network. A ZeroConf browser may be able to detect it automatically. Alternatively, it might be possible to assign a static address by mounting the PlutoSDR as a USB Storage device and editing ipaddr_eth in config.txt (refer to PlutoSDR documentation).

11.2. As a RF modem for a desktop PC or a laptop

In this variant the PlutoSDR is connected as a USB device to a host computer. IP networking is encapsulated over USB. Some proficiency with SSH is required.

Figure 2. PlutoSDR as USB peripheral

11.2.1. Requirements

- PlutoSDR
- USB cable (Type-A plug to Micro-B plug)
- Linux computer with scp and ssh.

11.2.2. Installation

- Connect the PlutoSDR to the computer.
- On the computer, run:

```
$ cd /tmp
$ wget http://www.pabr.org/radio/leantrx/leantrx-20180211.zip
$ unzip leantrx-20180211.zip
$ scp -r leantrx root@192.168.2.1:/tmp
Password: analog
$ ssh root@192.168.2.1
Password: analog
# cd /tmp/leantrx/bsp/plutosdr
# ./start.sh
```

• On the computer, open <u>http://192.168.2.1/leantrx/</u> in a web brower.

Note: This procedure also works over Ethernet, as in <u>Section 11.1, "As a network-connected RF modem</u>" without a USB Mass Storage device.

11.3. As a RF modem for a smartphone

In theory a smartphone or tablet could act as a USB OTG Host and play the same role as the PC in <u>Section 11.2, "</u> <u>As a RF modem for a desktop PC or a laptop "</u>. Unfortunately mainstream mobile operating systems do not implement this use case: when acting as a USB OTG Host, a smartphone or tablet will typically support USB Mass Storage devices, but not USB networking.

As a workaround, in this variant, we configure the PlutoSDR as a USB OTG Host and the phone acts as a USB

OTG Device. This is done in order to trigger the USB Tethering functionality of the phone.

The normal purpose of USB Tethering would be to share the WWAN connection of the phone with the PlutoSDR. Here we only take advantage of the underlying USB networking to open HTTP connections from the phone to the PlutoSDR.

Note that USB Tethering is often disabled on carrier-locked phones and on tablets which only have a Wi-Fi connection. An unlocked Android smartphone is recommended.

Figure 3. PlutoSDR, OTG hub, USB Tethering

11.3.1. Requirements

- PlutoSDR
- USB OTG adapter (Micro-B plug to Type-A receptacle)
- USB hub
- USB Mass Storage device
- USB power supply with Micro-B output.
- Smartphone with USB Tethering functionality.

11.3.2. Installation

The procedure is similar to that in <u>Section 11.1, "As a network-connected RF modem</u>". The PlutoSDR sees the phone as a RNDIS network interface rather than as an Ethernet adapter.

On Android, USB Tethering must be started manually in Settings -> Connections -> Mobile hotspot and tethering.

Once tethering is active, open http://192.168.42.42/leantrx/. The address is currently hardcoded. This may need customization for some smartphones.

Note that in this configuration modern smartphones may attempt to draw significant current through a solidstate switch (ADP198) and a 500 mA fuse inside the PlutoSDR. Use a powered hub if the USB connection appears to shut down randomly or the USB drive is not detected.

11.3.3. Variant with USB OTG cable

After LeanTRX has loaded from the Mass Storage device, the phone can be plugged directly into the PlutoSDR with a USB OTG cable (Micro-B plug to Micro-B plug). The Host end of the cable must be plugged into the PlutoSDR; it is sometimes identified by a red plug.

Note: This procedure must be performed without connecting a hub with a USB-Ethernet interface, otherwise LeanTRX will run live from the Mass Storage device instead of installing to the ramdisk. Customize leantrx/bsp/plutosdr/start.sh if needed.

As in the previous section, some phones may draw excessive current through the PlutoSDR.

In the picture below the phone is powered by an <u>OTG splitter cable</u>. This eliminates concerns about drawing high current from the PlutoSDR. Actually, the PlutoSDR itself is also powered by the cable, even though its data port is configured as USB Host. Sheet #8 of plutosdr_hw_schematic_revb.pdf suggests that the hardware can tolerate this. Power should not be applied to the other USB port simultaneously. Use at your own risk.

Figure 5. PlutoSDR, OTG splitter cable, USB Tethering



Figure 4. PlutoSDR, OTG cable, USB Tethering

11.4. Permanent installation

 $Lean TRX \ could \ also \ be \ merged \ into \ the \ firmware \ that \ runs \ from \ the \ internal \ flash \ memory \ of \ the \ Pluto SDR. \ This \ is \ beyond \ the \ scope \ of \ this \ article.$

12. Generic Linux support

TBD. How to install on a Raspberry Pi or similar embedded Linux platform with a USB SDR device.

13. Android support

TBD. How to install on a rooted Android smartphone or tablet with a USB SDR device.

14. Future work

- Use WebRTC or WebSockets instead of HTTP polling for status data.
- Use WebRTC for audio/video streaming.
- Implement modulators and demodulators for legacy modes (AM/SSB/FM/PSK31/RTTY).
- Implement RDS, CTCSS.
- Implement APRS receiver and beacon.
- Implement modes with simultaneous TX and RX (cross-band repeaters).
- Implement user interfaces with TX and RX (single- or full-duplex).
- Improve performance with PlutoSDR FPGA.
- Improve performance with Raspberry Pi GPU.

Bibliography

[LEANDVB] *leandvb: A lightweight software DVB-S demodulator*. http://www.pabr.org/radio/leandvb/leandvb.en.html .

[LEANSDR] Error: no bibliography entry: LEANSDR found in biblio.xml

Revision History				
1.0	2018-02-11	Initial release.		

http://www.pabr.org/radio/leantrx/leantrx.en.html

Copyright © 2018 pabr@pabr.org

All rights reserved.

<u>Other articles | ToC | PDF | Feed (this article) | @G+ | @Youtube | About | Discuss: Google+ |</u> Share: <u>Facebook Twitter Reddit StumbleUpon Google+</u>