Stupid Pluto Tricks with the ADALM-PLUTO

FOSDEM 2018

ROBIN GETZ

MICHAEL HENNERICH

02/04/2018
Agenda

► Analog Devices and Education
► Introduction to the ADALM-PLUTO
► Software support
  ▪ Libiio
  ▪ Supported applications
► Building custom images
► Watching airplanes (via dump1090)
► Detecting cell phone jammers
Analog Devices Educational offering

Secondary Schools
- Static Voltage/Current

First Year University: General Technology
- Time varying signals

Second/Third Year University: Electrical Engineering
- Impedance Measurements

Fourth Year/MSc University: Electrical Engineering
- Frequency Domain

Fourth Year/MSc University: Electrical Engineering
- Mechatronics / Controls

PhD Students: Practicing Electrical Engineers
- Network Analysis
- High Frequency & Specialized

Tools requirements evolve as student understanding increases

Tools for Explorations & Understanding

Introductory Instrumentation

Advanced PC Based Instrumentation

Provided by Mouser Electronics
ADALM-PLUTO
AD9363 Software Defined Radio Active Learning Module

► Captures I/Q Samples
  ▪ 12-bits
  ▪ 65.1 kSPS to 61.44 MSPS
  ▪ 200kHz to 20 MHz signal bandwidth

► Sends them to PC for processing over USB2

► $149
  ▪ $99 introductory price

► Tuning range;
  ▪ 325 MHz to 3.8 GHz

AD9363 SDR Active Learning Module
Inside the ADALM-PLUTO

ADALM-PLUTO
AD9363 SDR Active Learning Module

Tx  Rx

USB

AD9363

ZYNQ

DDR3L

FLASH

USB Phy

ADI

FIR

LNA

Mixer

Filter

DAC

FIR

Interface

DMA

Drivers

Linux kernel

libiiio

USB 2.0
ADALM-PLUTO Design

- Design is open, just like all other ADI designs
  - Shows a minimal full system design
    - From antenna to USB
    - RF to bits
  - Only 72 parts on the BOM
    - All IC, R, C, L, connectors, etc
  - Schematics, Gerbers, BOM, Allegro Files posted
    - [https://wiki.analog.com/university/tools/pluto/hacking/hardware](https://wiki.analog.com/university/tools/pluto/hacking/hardware)
  - Passes FCC and CE tests
  - Achieves better RF than AD9363 datasheet specs
Regulation? (FCC is local, but most countries have similar organizations)

- ADALM-PLUTO is not a “Radio”.
  - WiFi, BLE, transmitters
    - Waveform, dwell time, LO frequency, bandwidth, etc.
  - These require type **certification**

- ADALM-PLUTO is nothing more than:
  - RF arbitrary waveform generator
  - RF capture device
  - These sorts of devices are FCC **verified**.
    - this device does not cause harmful interference.
    - this device must accept any interference received
  - We do this – we pass part 15 (Class A)
    - For use in business/industrial/commercial environments only.

- End users make it a radio.
  - End users may need certification
  - Highly encourage every user to get their HAM radio license

- The FCC allows a hobbyist to build up to five devices of a single design for personal use with no testing whatsoever.

- If you are contacted by the FCC (or anyone else) about a matter of spectrum interference, immediately stop using the device, don't use it again.

- Home-built transmitters, like all Part 15 transmitters, are not allowed to cause interference to licensed radio communications and must accept any interference that they receive.

- If the Commission determines that the operator of a transmitter has not attempted to ensure compliance by employing good engineering practices then that operator may be fined up to $10,000 for each violation and $75,000 for a repeat or continuing violation.
AD9363 Under the Hood

- **AD9361**: 2 Rx + 2 Tx
- **AD9364**: 1 Rx + 1 Tx
- **AD9363**: 2 Rx + 2 Tx

**Major sections:**
- RF input/output paths
- RF PLL/LO
- Clock generation
- ADC/DAC
- Digital filters
- Digital interface
- Enable state machine
- RX Gain (AGC)
- TX Attenuation
- Aux DAC/ADC and GPOs
- Analog and Digital Correction/Calibration

For more information:
- [http://www.analog.com/ad9361](http://www.analog.com/ad9361)
- [http://www.analog.com/ad9364](http://www.analog.com/ad9364)
- [http://www.analog.com/ad9363](http://www.analog.com/ad9363)
Performance Data (meets or exceeds AD9363 specs)

Tx:
- EVM (64 QAM, LTE10) of -46dB @ 800MHz
  - Waveform created with MathWorks LTE Toolbox, played out the ADALM-PLUTO, connected to Keysight PXA 9030A via SMA cable, and analyzed with Keysight Signal Studio.

Rx:
- EVM (64 QAM, LTE10) of -43 dB @ 800MHz
  - Waveform created with MathWorks LTE Toolbox, played out Keysight Arbitrary waveform generator connected to the ADALM-PLUTO via SMA cable, and then analyzed with Keysight Signal Studio.
It’s a learning tool, for educational settings
Just like the dwarf planet, ADALM-PLUTO is the dwarf SDR

- Temp range: 10°C to 40°C
  - Easier to correct for oscillator we used

- USB 2.0
  - 7 – 12 MSPS, depending on the host, without loosing samples.

- FPGA Size: tiny

- ARM: Single Core

- Oscillator
  - Rakon RXO3225M,
    - ±25ppm (uncorrected)
    - ±10ppm (factory calibrated)
    - ±1ppm (tuned for temperature)

- Tuning Range:
  - 300 – 3800 MHz (datasheet specs)
  - 70 – 6000 MHz (out of spec)

- RF Shielding
  - None

- RF Filtering
  - None

- Output power
  - 0dBm (CW), varies with frequency
ADALM-PLUTO software stack

- Runs Linux inside the device
- Uses Linux’s IIO framework to expose I/Q data and control
- Multi-Function Device
  - Native IIO over USB
  - Serial over USB
  - Ethernet over USB
  - Mass Storage
  - Device Firmware Update
- Host
  - USB dongles

- Cross Platform
  - Windows
  - Linux
  - MAC

- Cross framework
  - Stacked libraries based on libiio
ADALM-PLUTO possible use cases include IoT!

- Connect to host
- USB Thumb Drive
- USB LAN
- USB WiFi
- USB Audio

Connect to host
- PA
- Mixer
- Filter
- DAC
- LNA
- Filter

USB Thumb Drive
- PA
- Mixer
- Filter
- DAC
- LNA
- Filter

USB LAN
- PA
- Mixer
- Filter
- DAC
- LNA
- Filter

USB WiFi
- PA
- Mixer
- Filter
- DAC
- LNA
- Filter

USB Audio
- PA
- Mixer
- Filter
- DAC
- LNA
- Filter

Operating Systems:
- Linux
- Mac
- Windows

Libraries:
- libio
- USB 2.0

Note: USB WiFi and USB Audio are depicted with additional accessories such as a wall charger and microphone.
ADALM-PLUTO with IIO Oscilloscope
GNU Radio

- GNU Radio
  - Open-source software development toolkit that provides signal processing blocks to implement software radios.

- GR fosphor
  - GNU Radio block for RTSA-like spectrum visualization using OpenCL and OpenGL acceleration
Gqrx SDR

- **gqrx**
  - Gqrx is an open source software defined radio receiver (SDR) powered by the GNU Radio and the Qt graphical toolkit.
  - Change frequency, gain and apply various corrections (frequency, I/Q balance).
  - AM, SSB, CW, FM-N and FM-W (mono and stereo) demodulators.
  - Special FM mode for NOAA APT.
  - Variable band pass filter.
  - AGC, squelch and noise blankers.
  - FFT plot and waterfall.
  - Record and playback audio to / from WAV file.
  - Record and playback raw baseband data.
  - Spectrum analyzer mode where all signal processing is disabled.
SDRangel is an Open Source Qt5 / OpenGL 3.0+ SDR and signal analyzer frontend to various hardware (including Pluto)

- C, C++
- Decoders built in
- Linux, Windows
● Native MATLAB and Simulink support
  ▪ Hardware Support Package

● ADI's IIO system object
  ▪ On github
The most important thing in education: TextBooks and Labs

- Dr. Alex Wyglinski
  - WPI
- Dr. Di Pu
- Dr. Travis Collins

- Dr. Dennis Silage
  - Temple
ADALM-PLUTO runs embedded Linux!

- **U-Boot**
  - Linux 4.6.0 kernel
    - Root password: "analog"
- **buildroot**
  - Busybox
- ~2 second boot time

- 32Mbytes of Flash
- 512Mbytes of DDR3

**Boot process**

- U-Boot boots from SPI flash
  - Checks button,
    - if pressed DFU flash mode
  - Checks boot mode
    - Previous kernel can tell U-Boot to go into different modes
      - # device_reboot
        - Usage: /usr/sbin/device_reboot {ram|sf|reset|verbose|break}
        - sf : Reboot and enter Serial Flash DFU mode
        - ram : Reboot and enter RAM DFU mode
        - reset : Reboot
        - verbose: Reboot and start serial console Verbose
        - break : Reboot and HALT in u-boot
  - DFU ram mode – loads image into RAM and boots it – great for testing
- Default load U-Boot FIT image, and check CRC, then boot it

Interact with U-Boot via serial console with UART adapter ADALM-JTAGUART
Open Source Firmware

Build Instructions:

```bash
git clone --recursive https://github.com/analogdevicesinc/plutosdr-fw.git
cd plutosdr-fw
export CROSS_COMPILE=arm-xilinx-linux-gnueabi-
export PATH=$PATH:/opt/Xilinx/SDK/2016.2/gnu/arm/lin/bin
export VIVADO_SETTINGS=/opt/Xilinx/Vivado/2016.2/settings64.sh
make
```

Results in:

<table>
<thead>
<tr>
<th>File</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>pluto.frm</td>
<td>Main PlutoSDR firmware file used with the USB Mass Storage Device</td>
</tr>
<tr>
<td>pluto.dfu</td>
<td>Main PlutoSDR firmware file used in DFU mode</td>
</tr>
<tr>
<td>boot.frm</td>
<td>First and Second Stage Bootloader (u-boot + fsbl + uEnv) used with the USB Mass Storage Device</td>
</tr>
<tr>
<td>boot.dfu</td>
<td>First and Second Stage Bootloader (u-boot + fsbl) used in DFU mode</td>
</tr>
<tr>
<td>uboot-env.dfu</td>
<td>u-boot default environment used in DFU mode</td>
</tr>
<tr>
<td>plutosdr-fw-vX.XX.zip</td>
<td>ZIP archive containing all of the files above</td>
</tr>
<tr>
<td>plutosdr-jtag-bootstrap-vX.XX.zip</td>
<td>ZIP archive containing u-boot and Vivado TCL used for JATG bootstrapping</td>
</tr>
</tbody>
</table>
Building the firmware images

- Download and install Xilinx FPGA Tools
  - Vivado HLx **2016.4: WebPACK** and Editions - Linux Self Extracting Web Installer
  - During installation check under design tools **Software Development Kit (SDK)**
  - Under devices SoC make sure **Zynq-7000** is selected
  - Xilinx gcc tools are distributed as 32-bit binaries you may need to add 32-bit libs

```bash
michael@HAL9000:~/devel$ dpkg --add-architecture i386
michael@HAL9000:~/devel$ apt-get update
michael@HAL9000:~/devel$ sudo apt-get install libc6:i386 libstdc++6:i386
michael@HAL9000:~/devel$ sudo apt-get install git build-essential fakeroot libncurses5-dev libssl-dev ccache
michael@HAL9000:~/devel$ sudo apt-get install dfu-util u-boot-tools device-tree-compiler libssl1.0-dev mtools
```

- Install other build dependencies

```bash
michael@HAL9000:~/devel$ git clone --recursive https://github.com/analogdevicesinc/plutosdr-fw.git
michael@HAL9000:~/devel$ cd plutosdr-fw
michael@HAL9000:~/devel/plutosdr-fw$ export CROSS_COMPILE=arm-xilinx-linux-gnueabi-
michael@HAL9000:~/devel/plutosdr-fw$ export PATH=$PATH:/opt/Xilinx/SDK/2016.4/gnu/arm/lin/bin
michael@HAL9000:~/devel/plutosdr-fw$ export VIVADO_SETTINGS=/opt/Xilinx/Vivado/2016.4/settings64.sh
michael@HAL9000:~/devel/plutosdr-fw$ make
```

- Clone and build the Firmware image
Customizing the PlutoSDR filesystem

- Customize buildroot target packages

  ```bash
  michael@HAL9000:~devel/plutosdr-fw$ cd buildroot
  michael@HAL9000:~devel/plutosdr-fw/buildroot$ make menuconfig
  michael@HAL9000:~devel/plutosdr-fw/buildroot$ make savedefconfig
  michael@HAL9000:~devel/plutosdr-fw/buildroot$ cd ..
  michael@HAL9000:~devel/plutosdr-fw$ make
  ```

- Customize buildroot busybox tools

  ```bash
  michael@HAL9000:~devel/plutosdr-fw/buildroot$ make busybox-menuconfig
  michael@HAL9000:~devel/plutosdr-fw/buildroot$ cp output/build/busybox-*/.config board/pluto/busybox-*_.config
  michael@HAL9000:~devel/plutosdr-fw$ make
  ```
Customizing the PlutoSDR filesystem

Adding files

► For temporary modifications
  - Modify the target filesystem directly and then rebuild the image

► For permanent additions
  - **Post-build scripts**
    • Are shell scripts called after Buildroot builds all the selected software, but before the rootfs images are assembled.

```
michael@HAL9000:~/devel/plutosdr-fw$ cp ~/foobar.sh buildroot/output/target/sbin/
michael@HAL9000:~/devel/plutosdr-fw$ make
```

► **Filesystem overlays**
  - A tree of files that is copied directly over the target filesystem after it has been built.

```
michael@HAL9000:~/devel/plutosdr-fw$ cat buildroot/board/pluto/post-build.sh
[ - snip -]
${INSTALL} -D -m 0644 ${BOARD_DIR}/input-event-daemon.conf ${TARGET_DIR}/etc/
[- snip --]
```
Along with each PlutoSDR firmware release we also provide the buildroot generated sysroot.

- wget https://github.com/analogdevicesinc/plutosdr-fw/releases/download/v0.27/sysroot-v0.27.tar.gz
- tar xzvf sysroot-v0.27.tar.gz
- git clone https://github.com/PlutoSDR/dump1090.git
- cd dump1090

```
CC=arm-xilinx-linux-gnueabi-gcc CFLAGS=--sysroot=../staging LDFLAGS=--sysroot=../staging make
```

```
arm-xilinx-linux-gnueabi-gcc --sysroot=../staging -c dump1090.c
arm-xilinx-linux-gnueabi-gcc --sysroot=../staging -c anet.c
arm-xilinx-linux-gnueabi-gcc -g -o dump1090 dump1090.o anet.o --sysroot=../staging -liio -lpthread -lm -lad9361
```

- scp dump1090 root@192.168.2.1:/sbin/

This allows you to later compile dynamically linked applications that can be executed on the PlutoSDR.
Cross platform

- Runs on Windows
- Linux and OS-X
ADALM-PLUTO Docs – online now!
https://wiki.analog.com/university/tools/pluto

- Docs are on the wiki
- Made public mid Feb
- Needs more
  - if you want to help let us know

The ADALM-PLUTO Active Learning Module (PLUTO) is an easy to use tool available from Analog Devices Inc. (ADI) that can be used to introduce fundamentals of RF, Software Defined Radio (SDR) or Communications as advanced topics in electrical engineering in a self or instructor lead setting. The PLUTO provides a personal laboratory experience to the user. It provides a physical portable lab that, when used with a host, can augment the learning that takes place in the classroom. A variety of software packages such as VPAD or Simulink provide an innovative graphical user interface (GUI) allowing intuitive usage and minimizes the learning curve to enable students to learn faster, work smarter, and explore more.

The ADALM-PLUTO Active Learning Module is a learning tool for everyone.

Based on the AD9933, it offers one receive channel and one transmit channel that can be operated in full-duplex, capable of generating or measuring RF analog signals from 2.0 to 3.0GHz, at up to 61.44 Mega Samples per Second (MSPS) with a 20 MHz bandwidth. The PLUTO kit is completely self-contained, fits nicely in a shirt pocket or backpack, and is entirely USB powered with the default firmware. With support for Linux™, Windows™, and macOS™ it allows exploration and understanding of RF systems no matter where or when the user is.

Please note that at the moment - PLUTO/SDR is under development. You will need to have sufficient skills to perform basic PC operations such as uploading files, installing software, moving and copying files and have the motivation to learn new software. You will also need to be patient as we add more of the missing documentation. If it looks orange - the docs are on the to-do list.

As we all work together on things, we expect it to get better, and more user friendly. Thanks for your patience. If you have questions - or want to point something out - please ask on the PLUTO Classroom. If you want to update something in this wiki - login, and click the edit button.
ADALM-PLUTO Support Model

- Buy ADALM-PLUTO
- AD9363 Design Files
- AD9363 Datasheet

- Application and Drivers for Linux and No-OS
  - Linux IIO: Linux Abstraction for Data Converters
- No-OS drivers
- HDL
- U-Boot
- buildroot

- Documentation
- PCB Schematics, Gerbers, BOM

- Online support via EngineerZone
  - Virtual Classroom (for ADALM-PLUTO)
  - Wideband RF Transceiver Community
  - FPGA Reference Design Community
  - Linux and Microcontroller Devices Drivers Comm.

- buy.analog.com
- www.analog.com
- github.com/analogdevicesinc
- wiki.analog.com
- ez.analog.com
Support

- https://ez.analog.com/community/university-program
  - ADALM-PLUTO users

- https://ez.analog.com/community/fpga
  - FPGA Developers

  - libiio users and developers
  - Driver users and developers
Run scripts from USB drive (supported in default image)

- The Pluto will automount any USB mass storage device such as thumb drive or Hard Drives. The automounter will then look for some special file names:
  - runme[0-9].sh which it will run as a shell script
  - runme[0-9] which it will run as a binary file.

```bash
#!/bin/sh

# the default directory the script runs in is /dev, so change to the drive
cd /media/sda1/

# create a file
touch foobar

# change the RX_LO to 2.4GHz
iio_attr -a -c ad9361-phy RX_LO frequency 2400000000

ACTION=remove_all /lib/mdev/automounter.sh
```
Replace the input-event-daemon .conf file

Default file:

```
# /etc/input-event-daemon.conf
#

[Global]
listen = /dev/input/event0

[Keys]
BTN_0 = ACTION=remove_all /lib/mdev/automounter.sh
```

Replace it with one from USB drive, which plays back pre-recorded files, record waveforms, or runs custom application, and then restart input-event-daemon
Play a CW at 908,460,000 Hz

#!/bin/sh

# the default directory the script runs in is /dev, so change to the drive
cd /media/sda1/

# create a file
touch foobar.txt

echo default-on > /sys/class/leds/led0:green/trigger >> foobar.txt

# Set the LO up
/usr/bin/iio_attr -a -c ad9361-phy TX_LO frequency 908460000 >> foobar.txt

# Set the Sample frequency up, tone will appear at sampling_frequency/32
/usr/bin/iio_attr -a -c -o ad9361-phy voltage0 sampling_frequency 32000000 >> foobar.txt

# Turn the attenuation down
/usr/bin/iio_attr -a -c -o ad9361-phy voltage0 hardwaregain 0 >> foobar.txt

# https://wiki.analog.com/resources/tools-software/linux-drivers/iio-transceiver/ad9361#bist_tone
# Inject 0dBFS tone at Fsample/32 into TX (all channels enabled)
/usr/bin/iio_attr -a -D ad9361-phy bist_tone "1 0 0 0" >> foobar.txt

cd /root
ACTION=remove_all /lib/mdev/automounter.sh
Cell phone jammers

Company web site:

This Handheld Selectable 8 band All Cell Phone Signal Jammer & WiFi GPS L1 All in one Jammer High-capacity (USA Version) suit for USA, against 4G LTE networks 3G GSM cellphone signals, and blocking WiFi and GPS L1. And it great use for office, school, home to blocks internet browse, cellphone conversation and GPS signal, one device to coverage all 4G 3G GSM WiFi GPS frequencies, no need any other device to suppress wireless signal in your office, home, school.

FCC web site:

The use of "cell jammers" or similar devices designed to intentionally block, jam, or interfere with authorized radio communications (signal blockers, GPS jammers, or text stoppers, etc.) is a violation of federal law. Also, it is unlawful to advertise, sell, distribute, or otherwise market these devices to consumers in the United States. These devices pose serious risks to critical public safety communications, and can prevent you and others from making 9-1-1 and other emergency calls. Jammers can also interfere with law enforcement communications. Operation of a jammer in the United States may subject you to substantial monetary penalties, seizure of the unlawful equipment, and criminal sanctions including imprisonment.
This is a "No Cell Phone" Area. Signals are blocked.

JAMMING CELL PHONES AND GPS EQUIPMENT IS AGAINST THE LAW!
Record files in the cell phone bands to look for CW
Thanks

► Questions?