ECE 531: Software Defined Radio

Dr. Daniel Gallagher

Adjunct Assistant Professor, Dept. of Electrical and Computer Engineering

Lecture 7

Topics:

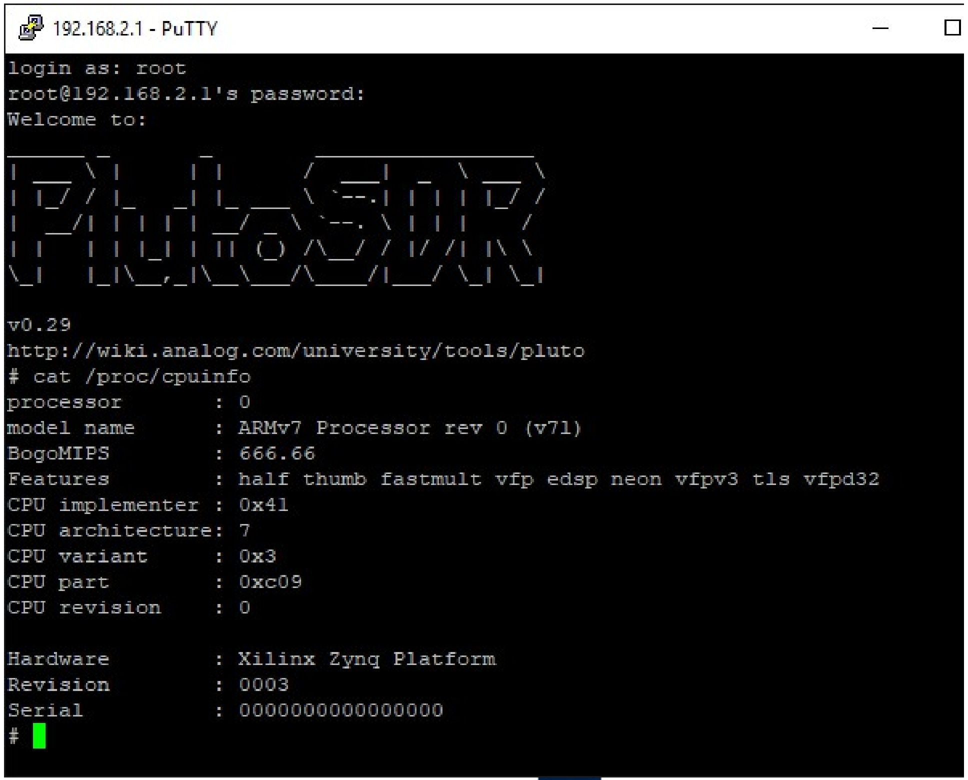
- Getting Started with Pluto SDR (SSH, sysfs, libiio, etc)

- IIO Oscilloscope

- Matlab with PlutoSDR

Getting Started with Pluto

- Pluto mounts as a USB drive
  - See info.html
  - Update firmware by Drag-n-drop into drive mount
- SSH into pluto
  - User: root
  - Pass: analog
- cat /proc/cpuinfo

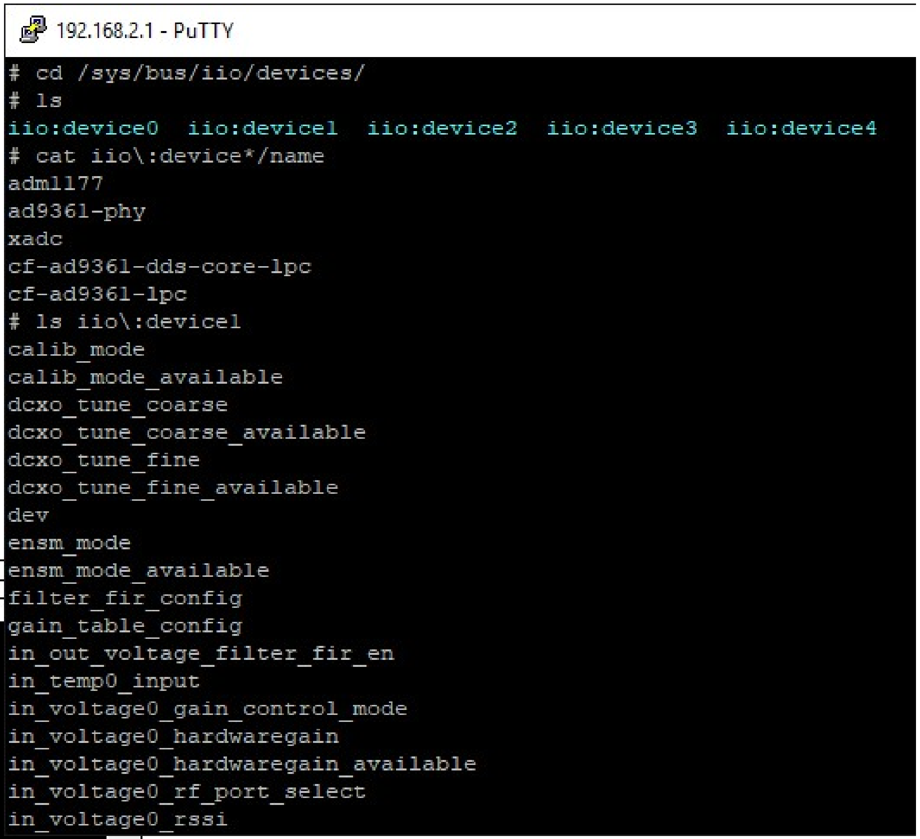


192.168.2.1 - PuTTY

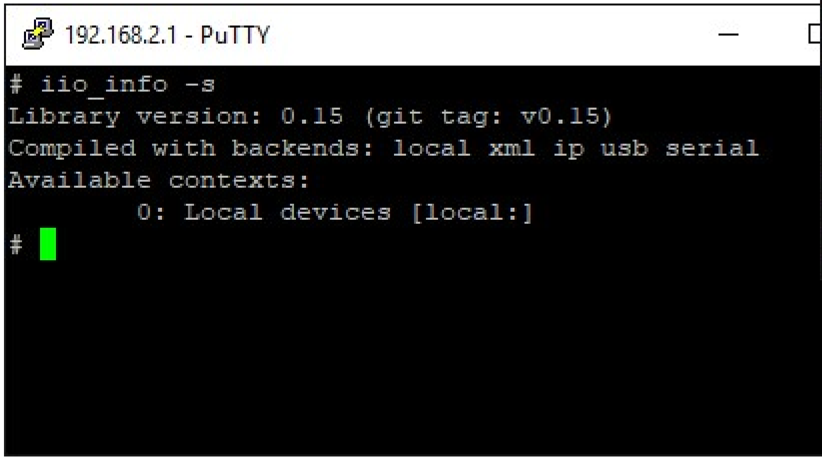
```
login as: root
root@192.168.2.1's password:
Welcome to:
______ _       _        _________________
| ___ \ |     | |      /  ___|  _  \ ___ \
| |_/ / |_   _| |_ ___ \ `--.| | | | |_/ /
|  __/| | | | | __/ _ \ `--. \ | | |    /
| |   | | |_| | || (_) /\__/ / |/ /| |\ \
\_|   |_|\__,_|\__\___/\____/|___/ \_| \_|

v0.29
http://wiki.analog.com/university/tools/pluto
# cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 0 (v7l)
BogoMIPS        : 666.66
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpd32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x3
CPU part        : 0xc09
CPU revision    : 0

Hardware        : Xilinx Zynq Platform
Revision        : 0003
Serial          : 0000000000000000
#
```

IIO Devices

- adm1177
  - Power monitor
- ad9361-phy
  - Controls transceiver
- xadc
  - Zynq AXI XADC IP core
- cf-ad9361-dds-core-lpc
  - DAC / TX output driver
  - Controls: TX DMA and HDL core
- cf-ad9361-lpc
  - ADC / RX capture driver
  - Controls: RX DMA and HDL core

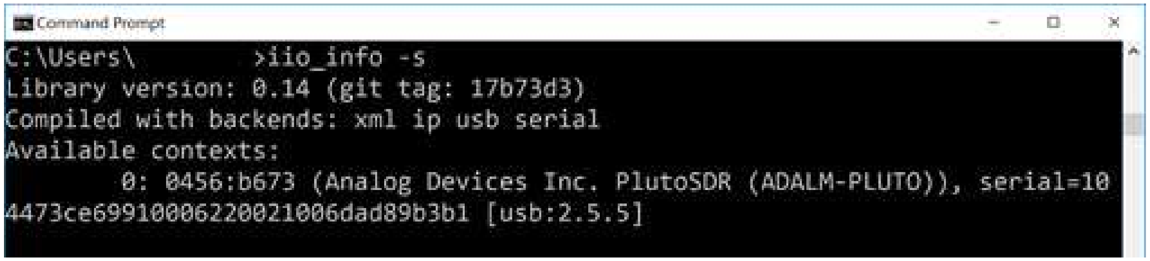


```
# cd /sys/bus/iio/devices/
# ls
iio:device0  iio:device1  iio:device2  iio:device3  iio:device4
# cat iio\:device*/name
adm1177
ad9361-phy
xadc
cf-ad9361-dds-core-lpc
cf-ad9361-lpc
# ls iio\:device1
calib_mode
calib_mode_available
dcxo_tune_coarse
dcxo_tune_coarse_available
dcxo_tune_fine
dcxo_tune_fine_available
dev
ensm_mode
ensm_mode_available
filter_fir_config
gain_table_config
in_out_voltage_filter_fir_en
in_temp0_input
in_voltage0_gain_control_mode
in_voltage0_hardwaregain
in_voltage0_hardwaregain_available
in_voltage0_rf_port_select
in_voltage0_rssi
```



```
# iio_info -s
Library version: 0.15 (git tag: v0.15)
Compiled with backends: local xml ip usb serial
Available contexts:
        0: Local devices [local:]
# 
```

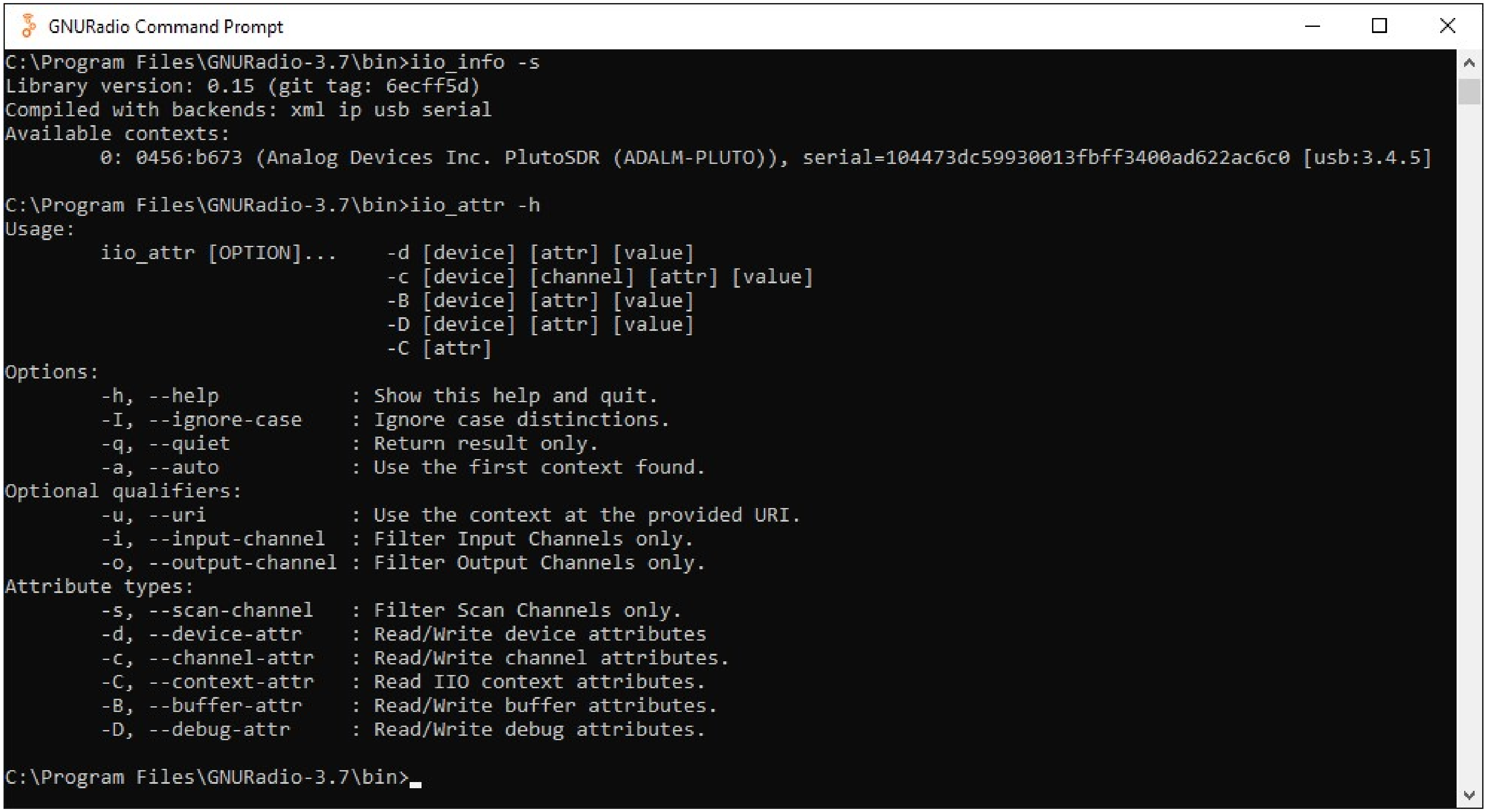
libIIO Command Line Tools



```
C:\Users\        >iio_info -s
Library version: 0.14 (git tag: 17b73d3)
Compiled with backends: xml ip usb serial
Available contexts:
        0: 0456:b673 (Analog Devices Inc. PlutoSDR (ADALM-PLUTO)), serial=10
4473ce69910006220021006dad89b3b1 [usb:2.5.5]
```

- iio_adi_xflow_check
  - Overflow/underflow testing
- **iio_attr**
  - Attribute reading and writing
- iio_genxml
  - Generate xml from context tree
- **iio_info**
  - Find devices and list attributes
- iio_readdev
  - Read from stream devices
- iio_reg
  - Read and write to registers
- iio_writedev
  - Write to stream devices

libIIO : Controlling Pluto SDR from Host

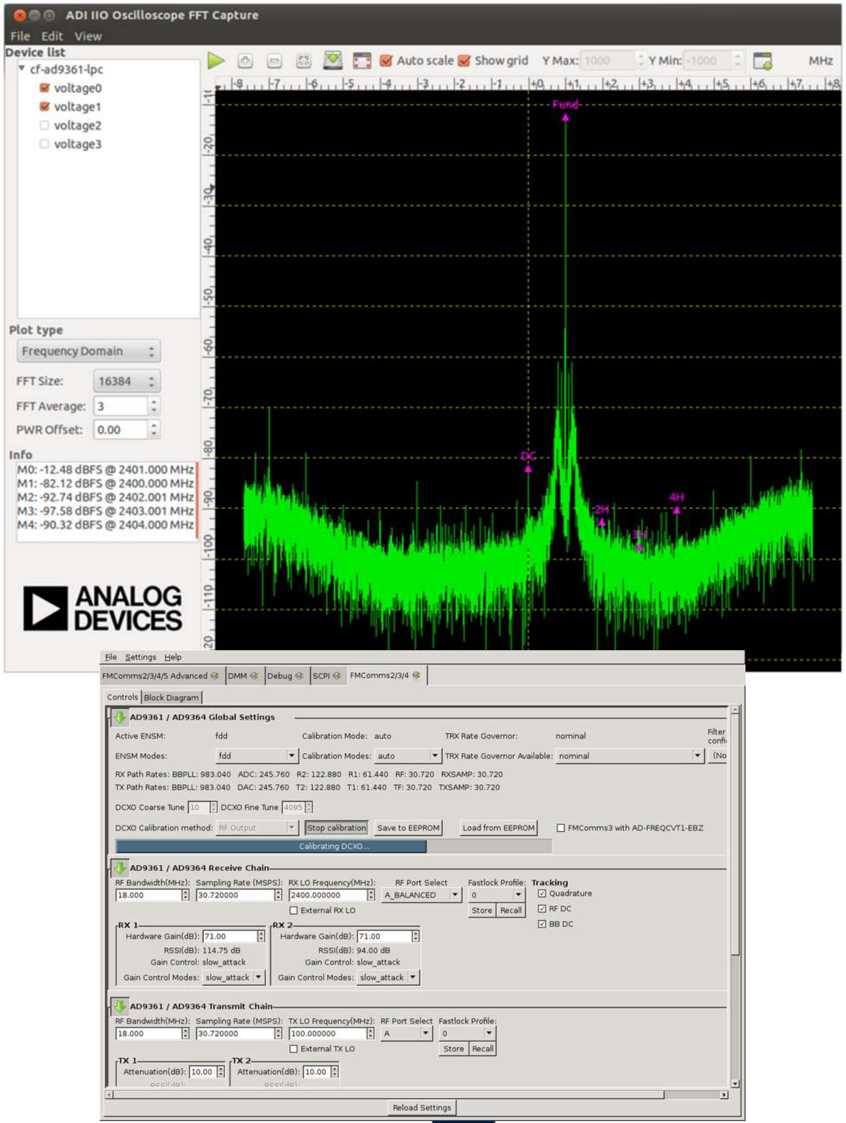


```
C:\Program Files\GNURadio-3.7\bin>iio_info -s
Library version: 0.15 (git tag: 6ecff5d)
Compiled with backends: xml ip usb serial
Available contexts:
        0: 0456:b673 (Analog Devices Inc. PlutoSDR (ADALM-PLUTO)), serial=104473dc59930013fbff3400ad622ac6c0 [usb:3.4.5]

C:\Program Files\GNURadio-3.7\bin>iio_attr -h
Usage:
        iio_attr [OPTION]...    -d [device] [attr] [value]
                                -c [device] [channel] [attr] [value]
                                -B [device] [attr] [value]
                                -D [device] [attr] [value]
                                -C [attr]
Options:
        -h, --help           : Show this help and quit.
        -I, --ignore-case    : Ignore case distinctions.
        -q, --quiet          : Return result only.
        -a, --auto           : Use the first context found.
Optional qualifiers:
        -u, --uri            : Use the context at the provided URI.
        -i, --input-channel  : Filter Input Channels only.
        -o, --output-channel : Filter Output Channels only.
Attribute types:
        -s, --scan-channel   : Filter Scan Channels only.
        -d, --device-attr    : Read/Write device attributes
        -c, --channel-attr   : Read/Write channel attributes.
        -C, --context-attr   : Read IIO context attributes.
        -B, --buffer-attr    : Read/Write buffer attributes.
        -D, --debug-attr     : Read/Write debug attributes.

C:\Program Files\GNURadio-3.7\bin>_
```

IIO Oscilloscope

- Open source C program
- Capture and display data
  - Time domain
  - Frequency domain
  - Constellation plot
- Plugins for IIO devices
  - Set device configuration
  - Read attributes
- Very useful for debugging and troubleshooting

- Included on Linux VM released to class
- Windows installer uploaded



Device list
- cf-ad9361-lpc
  - ☑ voltage0
  - ☑ voltage1
  - □ voltage2
  - □ voltage3

Plot type: Frequency Domain
FFT Size: 16384
FFT Average: 3
PWR Offset: 0.00

Info
M0: -12.48 dBFS @ 2401.000 MHz
M1: -82.12 dBFS @ 2400.000 MHz
M2: -92.74 dBFS @ 2402.001 MHz
M3: -97.58 dBFS @ 2403.001 MHz
M4: -90.32 dBFS @ 2404.000 MHz

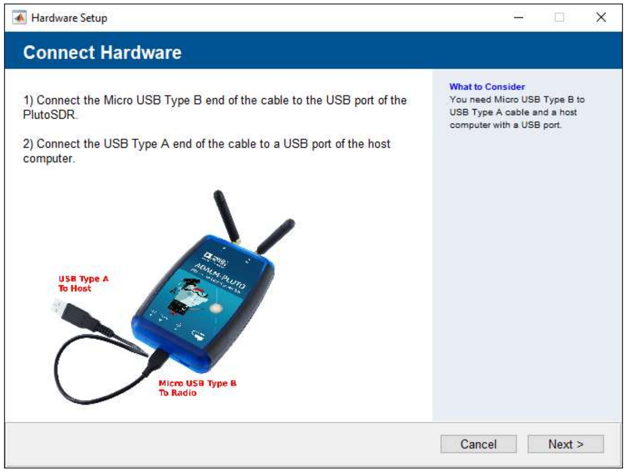
AD9361 / AD9364 Global Settings
Active ENSM: fdd Calibration Mode: auto TRX Rate Governor: nominal
ENSM Modes: fdd Calibration Modes: auto TRX Rate Governor Available: nominal
RX Path Rates: BBPLL: 983.040 ADC: 245.760 R2: 122.880 R1: 61.440 RF: 30.720 RXSAMP: 30.720
TX Path Rates: BBPLL: 983.040 DAC: 245.760 T2: 122.880 T1: 61.440 TF: 30.720 TXSAMP: 30.720
DCXO Coarse Tune: 10 DCXO Fine Tune: 4095
DCXO Calibration method: RF Output
□ FMComms3 with AD-FREQCVT1-EBZ
Calibrating DCXO...

AD9361 / AD9364 Receive Chain
RF Bandwidth(MHz): 18.000 Sampling Rate (MSPS): 30.720000 RX LO Frequency(MHz): 2400.000000 RF Port Select: A_BALANCED Fastlock Profile: 0
□ External RX LO
Tracking: ☑ Quadrature ☑ RF DC ☑ BB DC
RX 1: Hardware Gain(dB): 71.00 RSSI(dB): 114.75 dB Gain Control: slow_attack Gain Control Modes: slow_attack
RX 2: Hardware Gain(dB): 71.00 RSSI(dB): 94.00 dB Gain Control: slow_attack Gain Control Modes: slow_attack

AD9361 / AD9364 Transmit Chain
RF Bandwidth(MHz): 18.000 Sampling Rate (MSPS): 30.720000 TX LO Frequency(MHz): 100.000000 RF Port Select: A Fastlock Profile: 0
□ External TX LO
TX 1: Attenuation(dB): 10.00
TX 2: Attenuation(dB): 10.00

Pluto SDR with MATLAB

- Must install Mathworks Hardware Support Package (HSP)
  - – Provides two Pluto system objects
    - comm.SDRRxPluto and comm.SDRTxPluto
    - Act as an IIO client
    - See Appendix B.3 for more on system objects



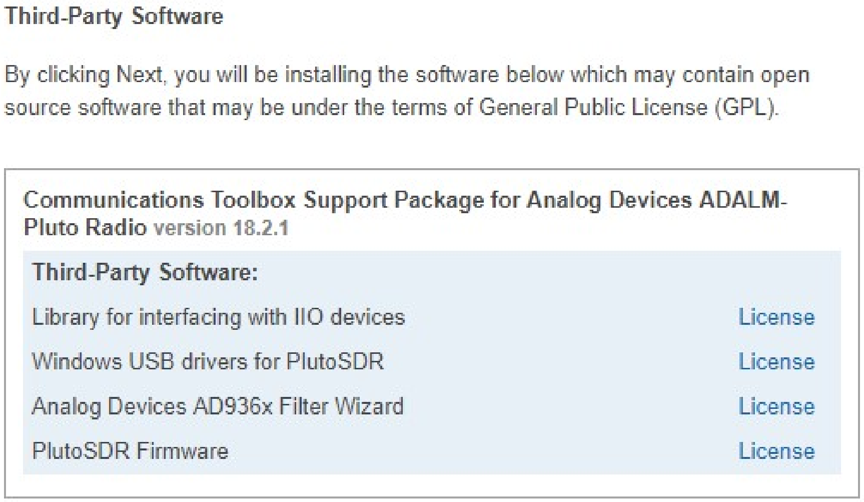
Hardware Setup

Connect Hardware

1) Connect the Micro USB Type B end of the cable to the USB port of the PlutoSDR.

2) Connect the USB Type A end of the cable to a USB port of the host computer.

What to Consider
You need Micro USB Type B to USB Type A cable and a host computer with a USB port.



Third-Party Software

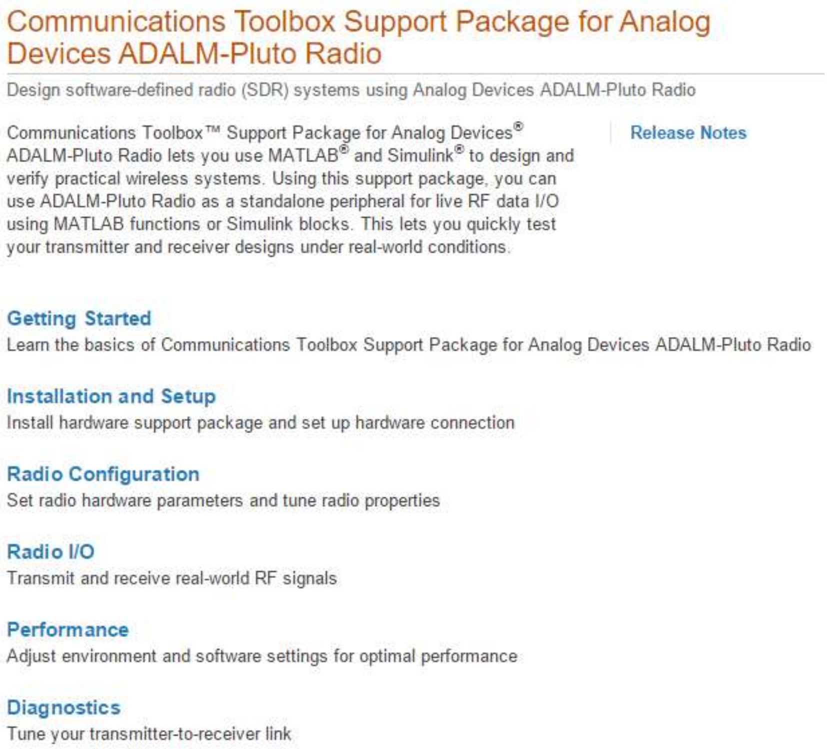
By clicking Next, you will be installing the software below which may contain open source software that may be under the terms of General Public License (GPL).

Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio version 18.2.1

Third-Party Software:	
Library for interfacing with IIO devices	License
Windows USB drivers for PlutoSDR	License
Analog Devices AD936x Filter Wizard	License
PlutoSDR Firmware	License

PlutoSDR MATLAB Documentation

- Some things to try with your hardware after adding PlutoSDR HSP
- `>> plutoradiodoc`
  - Getting Started documentation



Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio

Design software-defined radio (SDR) systems using Analog Devices ADALM-Pluto Radio

Communications Toolbox™ Support Package for Analog Devices® ADALM-Pluto Radio lets you use MATLAB® and Simulink® to design and verify practical wireless systems. Using this support package, you can use ADALM-Pluto Radio as a standalone peripheral for live RF data I/O using MATLAB functions or Simulink blocks. This lets you quickly test your transmitter and receiver designs under real-world conditions.

Release Notes

Getting Started
Learn the basics of Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio

Installation and Setup
Install hardware support package and set up hardware connection

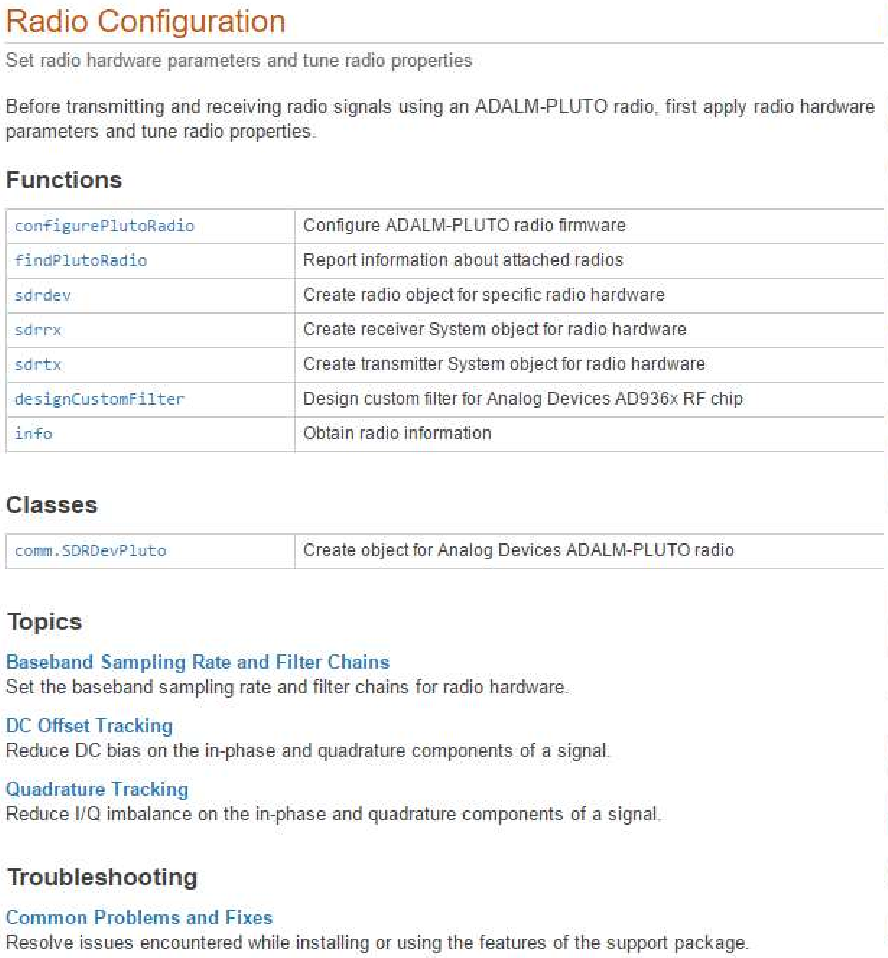
Radio Configuration
Set radio hardware parameters and tune radio properties

Radio I/O
Transmit and receive real-world RF signals

Performance
Adjust environment and software settings for optimal performance

Diagnostics
Tune your transmitter-to-receiver link

PlutoSDR Matlab Documentation



Radio Configuration

Set radio hardware parameters and tune radio properties

Before transmitting and receiving radio signals using an ADALM-PLUTO radio, first apply radio hardware parameters and tune radio properties.

Functions

`configurePlutoRadio`	Configure ADALM-PLUTO radio firmware
`findPlutoRadio`	Report information about attached radios
`sdrdev`	Create radio object for specific radio hardware
`sdrrx`	Create receiver System object for radio hardware
`sdrtx`	Create transmitter System object for radio hardware
`designCustomFilter`	Design custom filter for Analog Devices AD936x RF chip
`info`	Obtain radio information

Classes

`comm.SDRDevPluto`	Create object for Analog Devices ADALM-PLUTO radio

Topics

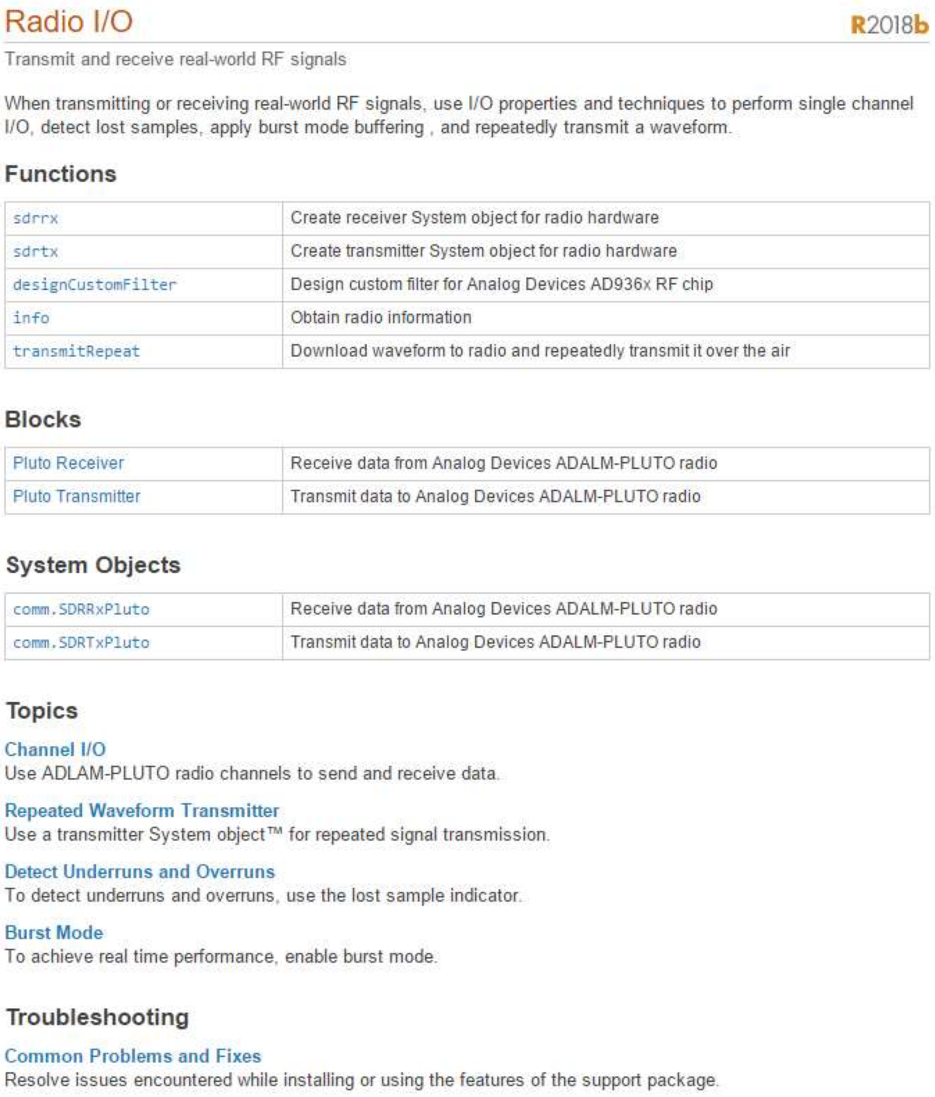
Baseband Sampling Rate and Filter Chains
Set the baseband sampling rate and filter chains for radio hardware.

DC Offset Tracking
Reduce DC bias on the in-phase and quadrature components of a signal.

Quadrature Tracking
Reduce I/Q imbalance on the in-phase and quadrature components of a signal.

Troubleshooting

Common Problems and Fixes
Resolve issues encountered while installing or using the features of the support package.



Radio I/O

R2018b

Transmit and receive real-world RF signals

When transmitting or receiving real-world RF signals, use I/O properties and techniques to perform single channel I/O, detect lost samples, apply burst mode buffering , and repeatedly transmit a waveform.

Functions

`sdrrx`	Create receiver System object for radio hardware
`sdrtx`	Create transmitter System object for radio hardware
`designCustomFilter`	Design custom filter for Analog Devices AD936x RF chip
`info`	Obtain radio information
`transmitRepeat`	Download waveform to radio and repeatedly transmit it over the air

Blocks

Pluto Receiver	Receive data from Analog Devices ADALM-PLUTO radio
Pluto Transmitter	Transmit data to Analog Devices ADALM-PLUTO radio

System Objects

`comm.SDRRxPluto`	Receive data from Analog Devices ADALM-PLUTO radio
`comm.SDRTxPluto`	Transmit data to Analog Devices ADALM-PLUTO radio

Topics

Channel I/O
Use ADLAM-PLUTO radio channels to send and receive data.

Repeated Waveform Transmitter
Use a transmitter System object™ for repeated signal transmission.

Detect Underruns and Overruns
To detect underruns and overruns, use the lost sample indicator.

Burst Mode
To achieve real time performance, enable burst mode.

Troubleshooting

Common Problems and Fixes
Resolve issues encountered while installing or using the features of the support package.

MATLAB PlutoSDR Examples

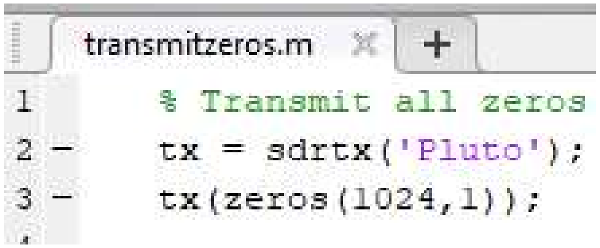
- `>> plutoradioexamples`

- Matlab and Simulink hardware examples
  - `plutoradioADSBSimulinkExample`
  - `plutoradioWLANTransmitReceiveExample`
  - `plutoradioRBDSExample`

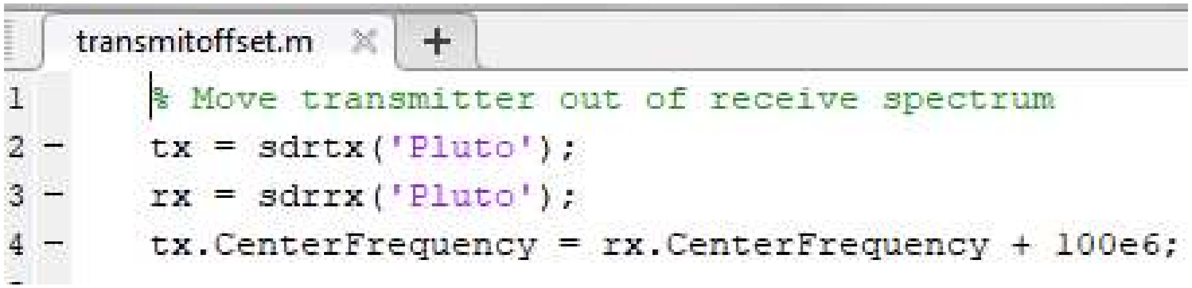
Pluto SDR: Continuous Transmit

- Anytime the Pluto SDR is powered on, the transmitter activates and will transmit data
- This occurs when user does not intend
  - (i.e. when using just the receiver)

- Possible fixes:
  - Write a vector of zeros to transmitter object
    - LO leakage still possible (May be self jamming)
  - Shift TX LO out of the receive band



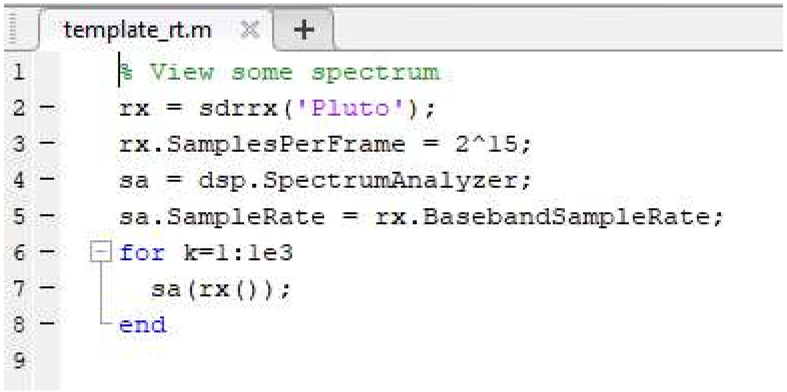
```
% Transmit all zeros
tx = sdrtx('Pluto');
tx(zeros(1024,1));
```



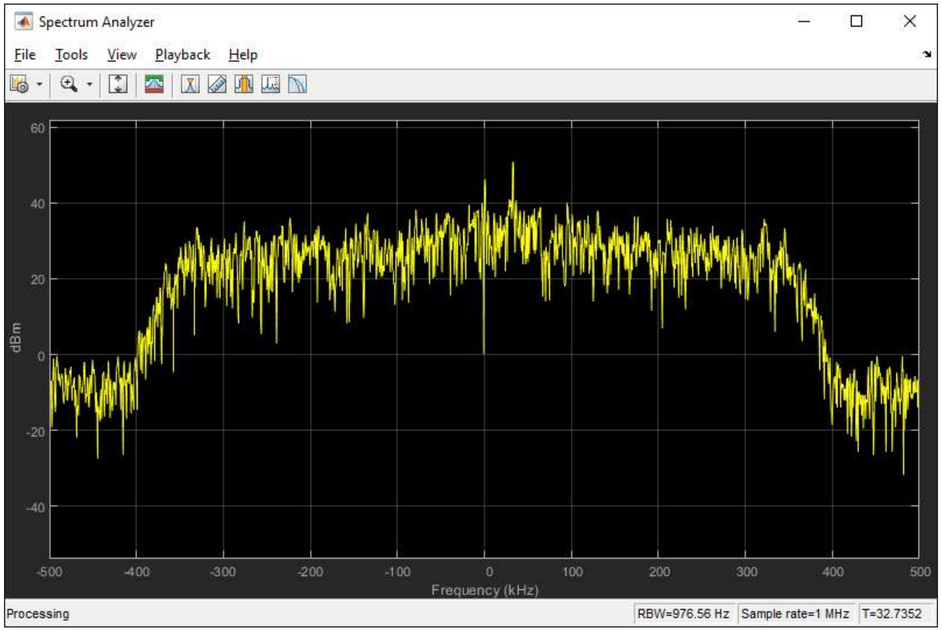
```
% Move transmitter out of receive spectrum
tx = sdrtx('Pluto');
rx = sdrrx('Pluto');
tx.CenterFrequency = rx.CenterFrequency + 100e6;
```

MATLAB RX Spectrum Analysis

- Spectrum Analyzer is available in the DSP Toolbox



```
% View some spectrum
rx = sdrrx('Pluto');
rx.SamplesPerFrame = 2^15;
sa = dsp.SpectrumAnalyzer;
sa.SampleRate = rx.BasebandSampleRate;
for k=1:1e3
  sa(rx());
end
```



Processing RBW=976.56 Hz Sample rate=1 MHz T=32.7352

Pluto SDR on GNU Radio

- PlutoSDR and general IIO support in GNU Radio using gr-iio

Pluto Buffer Size

- Buffer Size
  - Small buffer
    - Less latency, more overhead
  - Large buffer
    - More latency, less overhead