



AHEAD OF WHAT'S POSSIBLE™

Intro to SDR and Pluto GNU Radio Edition

JON KRAFT, FAE

FEB 2019

WWW.ANALOG.COM/RADIOVERSE

FIND THIS PRESENTATION AND WORKSHOP FILES AT:

[HTTPS://GITHUB.COM/JONKRAFT](https://github.com/jonkraft)

©2018 Analog Devices, Inc. All rights reserved.

The FAE Training logo, featuring a stylized white swoosh icon to the left of the text "FAE TRAINING" in a bold, white, sans-serif font. Below this, the text "ANALOG DEVICES, INC." is written in a smaller, white, sans-serif font.

FAE TRAINING
ANALOG DEVICES, INC.

Agenda

- ▶ Challenges in Getting Started with SDR
- ▶ Zero intermediate frequency (ZIF) radios vs. Super-Heterodyne radios
- ▶ Catalina (AD9361), Mykonos (AD9371), and Talise (ADRV9009) high-level overviews
- ▶ AD9361
- ▶ AD9371
- ▶ ADRV9009
- ▶ Getting Started with Pluto (aka AD936x)
 - Installing GNU Radio and Pluto Drivers
 - Lab Exercises
 - Lab 1: Play an AM Station on an FM Radio
 - Lab 2: Hack an RF Outlet

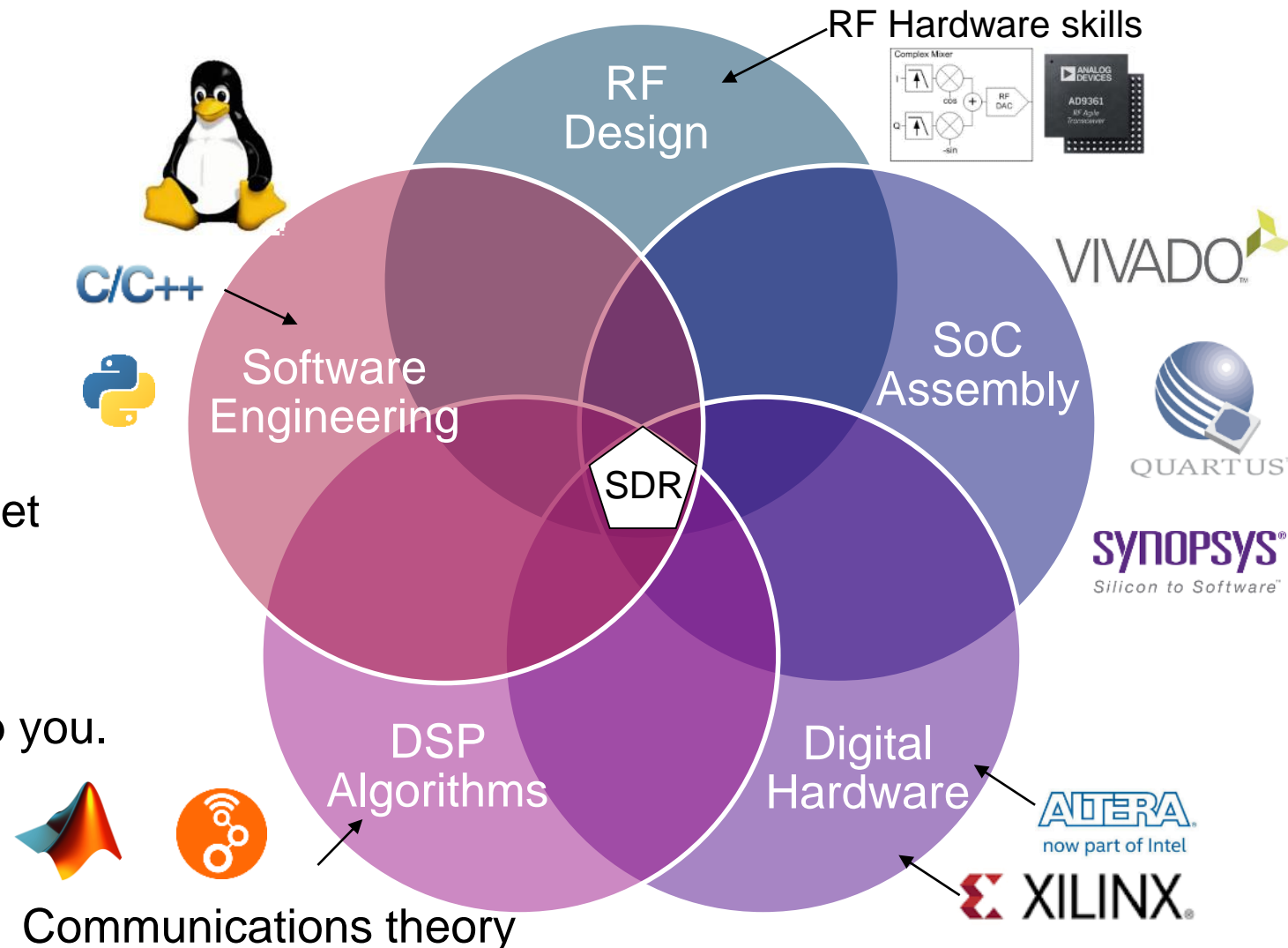
Challenges in Getting Started with SDR

► Software Defined Radio requires:

- Hardware Engineers
- Software Engineers
- Communications Engineers
- HDL Engineers
- Systems Engineers

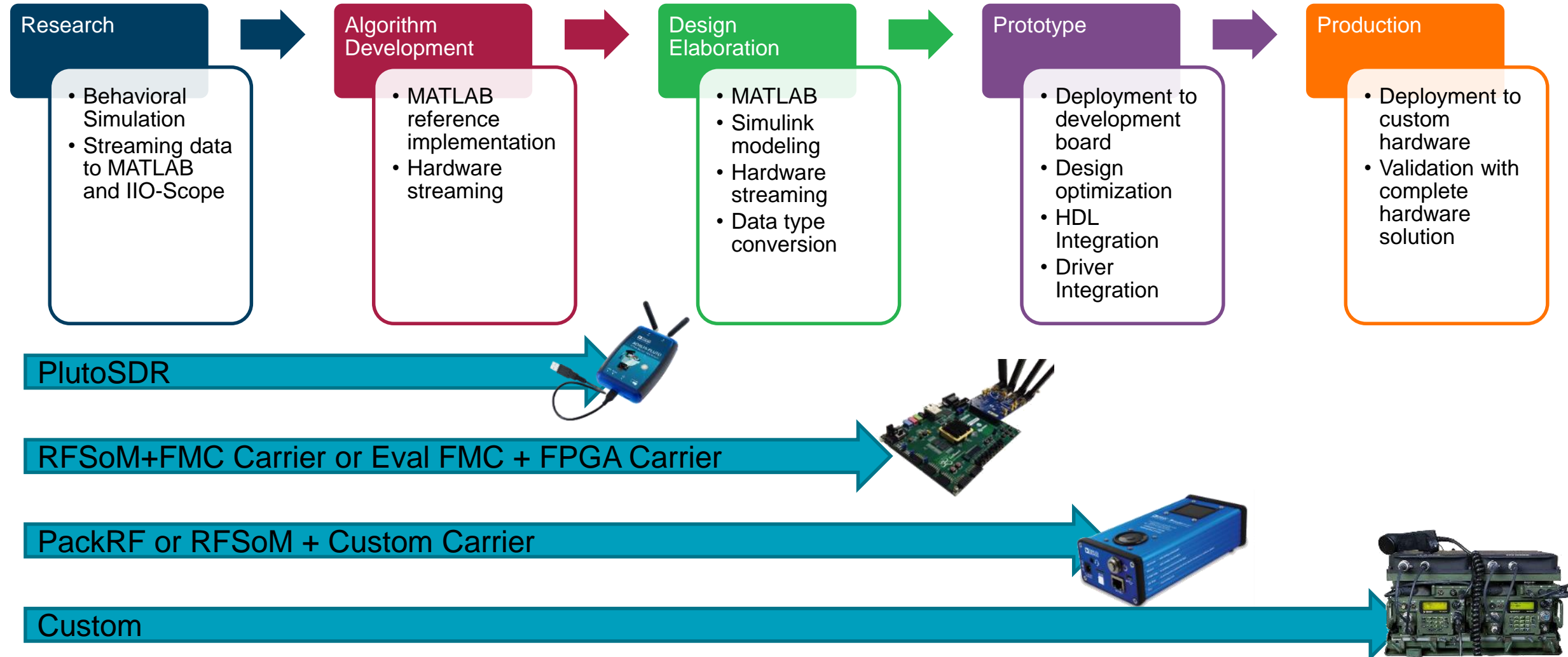
► So with so much entailed, how can we get started with SDR?

► ADI Has Hardware and Software to help you.



$$s[2\ell N + n] = \frac{1}{2N} \sum_{k=0}^{2N-1} p_k[\ell] e^{j(2\pi nk/2N)},$$

Design Flow



ADI's Hardware Prototyping Environment

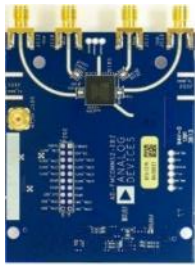
ADALM-PLUTO

- AD9363
- 1 x Rx, 1 x Tx
- 325 MHz – 3.8GHz
- 200kHz – 20 MHz channel bandwidth



AD-FMCOMMS2 AD-FMCOMMS3

- AD9361
- 2 x Rx, 2 x Tx
- **tuning range**
 - 2.2 GHz – 2.6GHz
 - 70 MHz – 6GHz
- 200kHz - 56 MHz channel bandwidth



AD-FMCOMMS4

- **AD9364**
- 1 x Rx, 1 x Tx
- 70 MHz – 6GHz tuning range
- 200kHz - 56 MHz channel bandwidth
- Shipping Now



ARRADIO

- AD9361
- HSMC, not FMC
- 2 x Rx, 2 x Tx
- **2.2 GHz – 2.6GHz tuning range**
- 200kHz - 56 MHz channel bandwidth
- Shipping Now!



AD-FMCOMMS5

- 2 x **AD9361**
- 4 x Rx, 4 x Tx
- **Synchronized RF**
- 70 MHz – 6GHz tuning range
- 200kHz - 56 MHz channel bandwidth
- Shipping Now!



ADRV9371-N/PCBZ ADRV9371-W/PCBZ

- **AD9371**
- 2 x Rx, 2 x Tx, 2 x Obs, 1x **Sniffer**
- tuning range
 - 1.8GHz – 2.6GHz
 - 300MHz – 6GHz
- Tx synthesis bandwidth 250 MHz
- Rx BW: 8 MHz to 100 MHz



ADRV9375-N/PCBZ ADRV9375-W/PCBZ

- **AD9375**
- 2 x Rx, 2 x Tx, 2 x Obs, 1x **Sniffer**
- tuning range
 - 1.8GHz – 2.6GHz
 - 300MHz – 6GHz
- **DPD actuator and adaptation engine for PA linearization**



ADRV9008-1W/PCBZ (Rx) ADRV9008-2W/PCBZ (Tx/Obs)

ADRV9009-W/PCBZ (TDD)

- **ADRV9008-1, ADRV9008-2, ADRV9009**
- 2 x Rx, 2 x Tx, 2 x Obs, 1x **Sniffer**
- 75MHz - 6GHz tuning range
- Tx synthesis bandwidth 450 MHz
- Rx BW to 200 MHz



ADRV9364-Z7020 ADRV9361-Z7035

- **AD9364 + Zynq 7020**
- **AD9361 + Zynq 7035**
- 70 MHz – 6GHz tuning range
- 200kHz - 56 MHz channel bandwidth
- 1GB DDR + 32MB FLASH
- Ethernet + USB Phy



PACKRF

- **ADRV9361 reference design**
- Battery, PoE, Screen, Audio, GPS, IMU



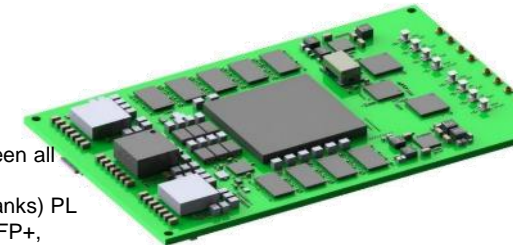
ADRV-DPD1

- **AD9375 + 250 mW PA**
- 2 Rx, 2 Rx
- LTE Band 7
- 2500 to 2570 Uplink
- 2620 to 2690 MHz Downlink
- 2 PAs, 2 LNAs, duplex filters



ADRV9009-ZU11EG

- 2 x **ADRV9009 + Zynq Ultrascale**
- 75MHz to 6GHz tuning range
- Rx BW 200MHz
- Tx synthesis bandwidth 450 MHz
- Integrated LO and Phase synch between all channels and Modules
- 4G x64 w/ECC PS; 4G (2Gb x32 x2Banks) PL
- USB3, USB2, PCIe 3.0 x8, QSFP+, SFP+, 1Gb Ethernet x2, and CPRI



ADI's Software Prototyping Environment: 4 Main Tools

► IIO Oscilloscope

- Built on the IIO LIB Linux Drivers
- Data Visualization Application
- Graphical Configuration Application

► TES GUI

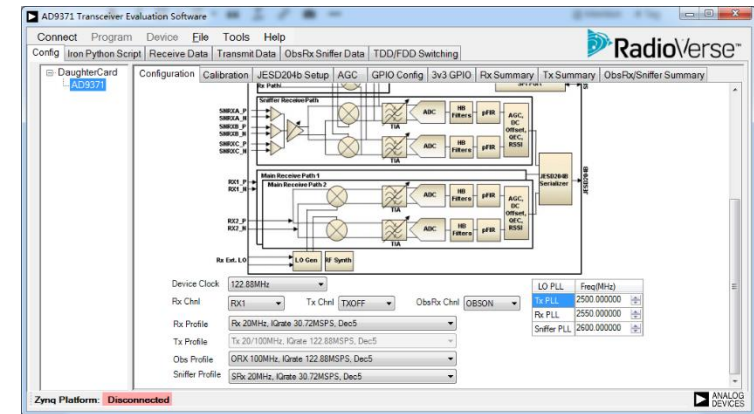
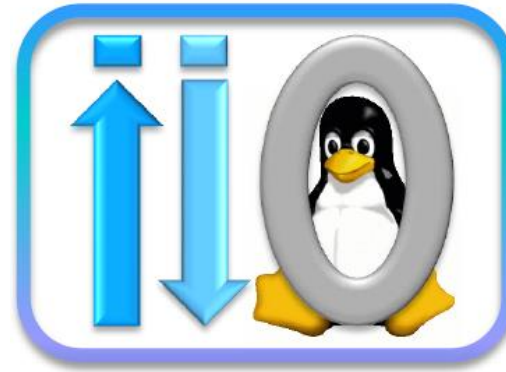
- For the AD937x and ADRV9008/9 products
- Evaluation and Python Scripting

► GNU Radio

- Free Linux Based Graphical Communications Toolbox
- Great intro to concepts and algorithm development
- Pluto and AD936x products supported, not AD937x or ADRV9008/9
- But for prototype and production, better to move to Matlab

► Matlab / Simulink

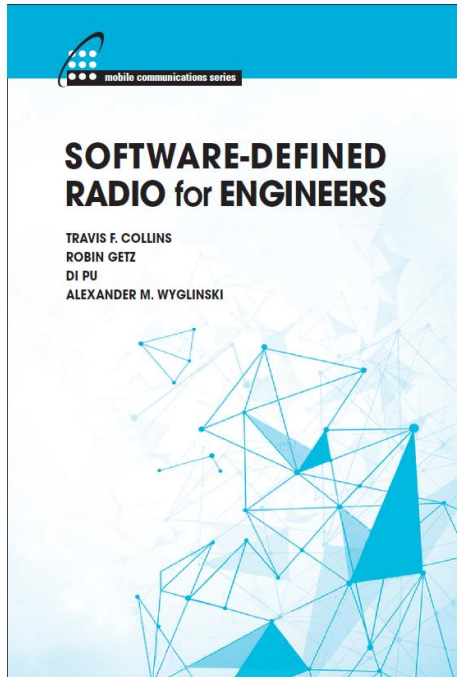
- All ADI Transceivers Supported in Matlab
- Evaluation → Verification → Detailed Design → Prototype
- See Next Slide for Seminar Details



Rapid Prototyping of RF Systems with Software Defined Radio

**THIS WORKSHOP IS COMING
TO A CITY NEAR YOU SOON!**

Hard cover text



ADALM-PLUTO SDR



MATLAB Trial



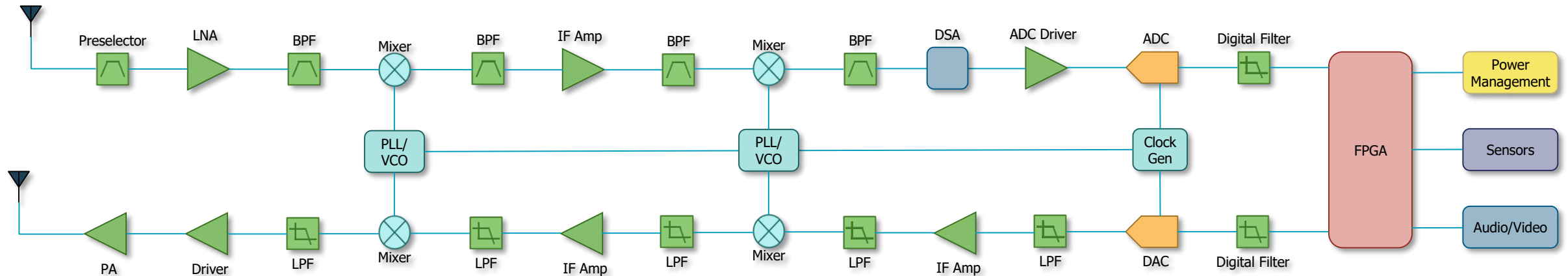
Time	Presenter	Topic
8:00 – 8:30	Richardson	<ul style="list-style-type: none">• Check In / Coffee• Pick up hardware and Textbook
8:30 – 9:00	Analog Devices MathWorks	<ul style="list-style-type: none">• Debug Software Installation Issues• Check Connectivity to hardware
9:00 – 9:10	Richardson	Introductions
9:10 – 9:45	Analog Devices	<ul style="list-style-type: none">• ADI Prototyping Ecosystem (RadioVerse)• AD9361 (chip)• Pluto SDR (system)• Instructor lead demo of dump1090
9:45 – 10:00	MathWorks	Model Based Design
10:00 – 10:45	Analog Devices	<ul style="list-style-type: none">• IIO infrastructure and Software Tools• capture and control radio with IIO command line tools, and the IIO Oscilloscope• Hands on labs (Coffee)
10:45 – 11:15	MathWorks	<ul style="list-style-type: none">• MATLAB and Simulink and system objects• Hands on labs
11:15 – 11:30	Analog Devices	Data Flow and Transfers
11:30 – 12:30	Analog Devices	<ul style="list-style-type: none">• Basic communications theory• Hands on labs
12:00 – 12:30	Lunch	<ul style="list-style-type: none">• Hands on labs (Working lunch)
12:30 – 12:45	MathWorks	Advanced Workflows
12:45 – 1:00	Analog Devices	Moving to Custom Hardware
1:00 – 3:00	Analog Devices	<ul style="list-style-type: none">• Care and Feeding of your AD9361 / Pluto SDR• Future Transceivers• Hands on Lab (Coffee)
3:00 – 4:00	Richardson	<ul style="list-style-type: none">• Signal Chain• LNA, PA, PLL, Filters, Up/Down converters & Power for RF Applications
4:00 – 4:15	All	Questions and Answers

ZIF Transceivers

COMPARISON TO SUPERHETERODYNE

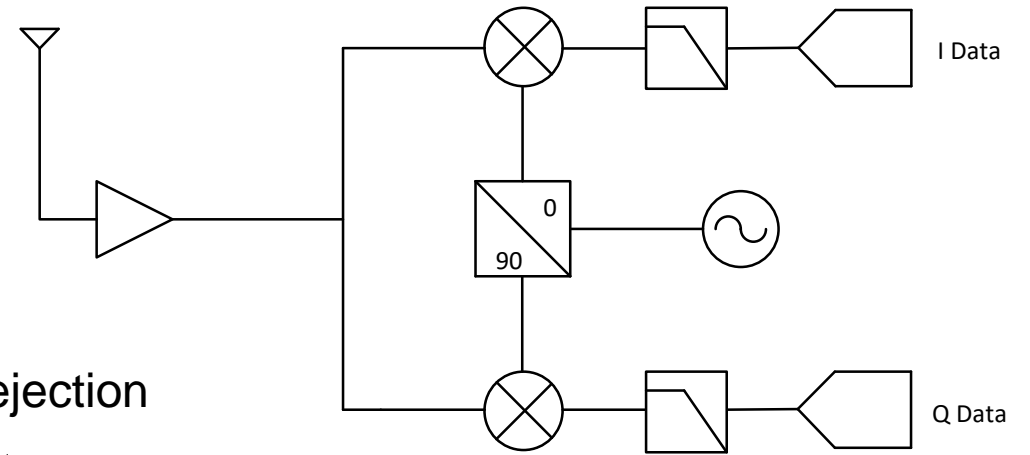
Super-Heterodyne Overview

- ▶ Introduced ~100 years ago
 - Still dominates most designs, in radar and MILCOM/EW/SIGINT
 - Trusted technique, allows for incremental improvements across devices
- ▶ High performance at the sacrifice of large size and power
 - Filters, especially in the IF strips, drive this



Direct Conversion Overview

- ▶ Direct conversion attempts to simplify the super-heterodyne
- ▶ One mixing stage
 - LO = RF
 - Rx goes directly to baseband
 - Tx goes directly to the desired RF frequency
- ▶ Removes filtering complexity in the design
 - Filtering takes place at baseband
 - But dependent on mixer performance for LO and image rejection
- ▶ Has numerous advantages over the superhet architecture
 - Filtering requirements less stringent
 - Significant reduction in power consumption
 - Resulting from reduced component count and reduced sample rate
 - Significant cost reduction
 - Resulting from reduced component count and filtering requirements



ZIF Architecture Problems

- Disadvantages over Superhet architecture
 - Inherent performance issues
 - I/Q phase/amplitude imbalance degrades image rejection
 - Poor LO isolation passes LO to RF (LO Leakage) and baseband (DC Offset)
 - Performance depends heavily on calibrations

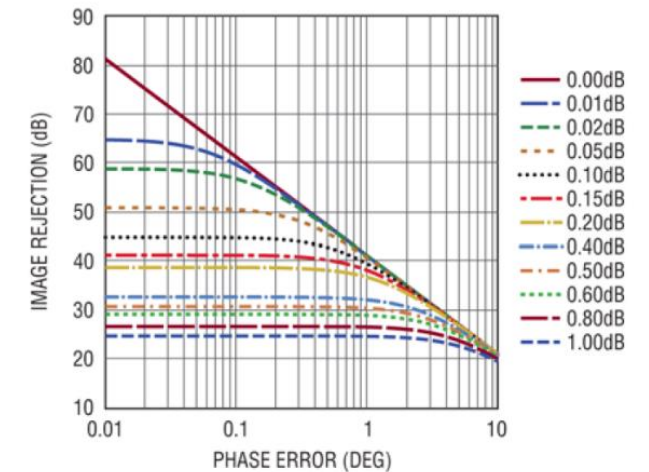
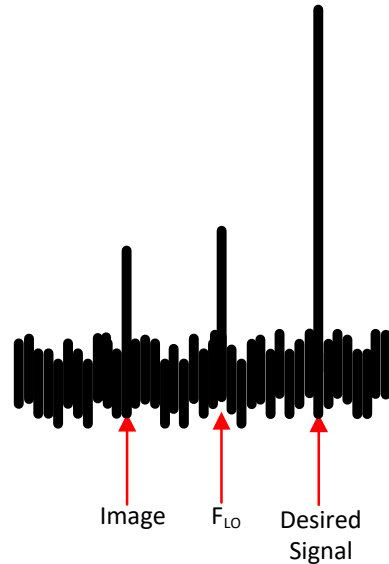
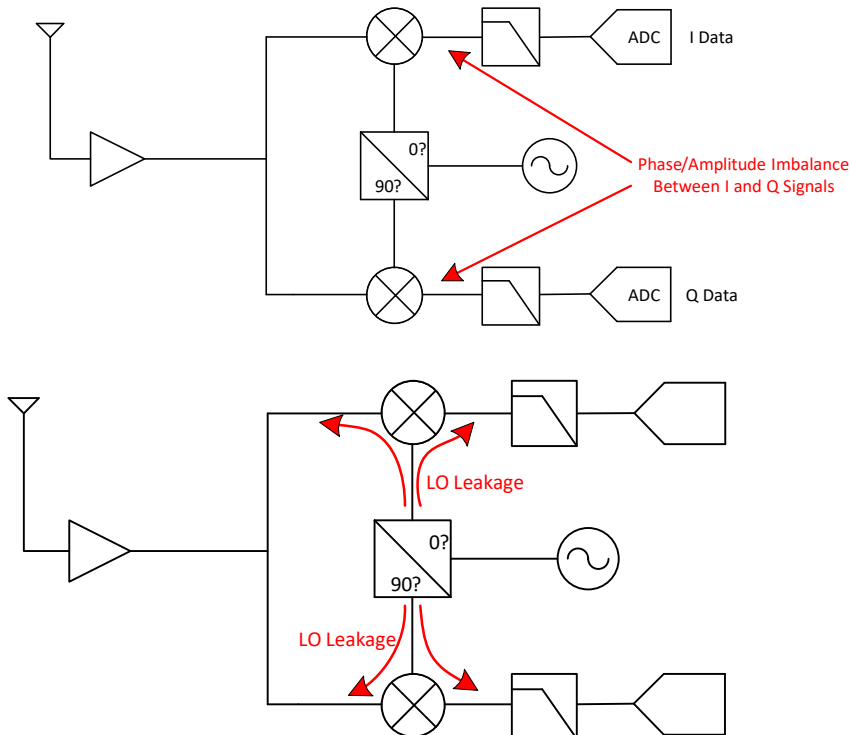
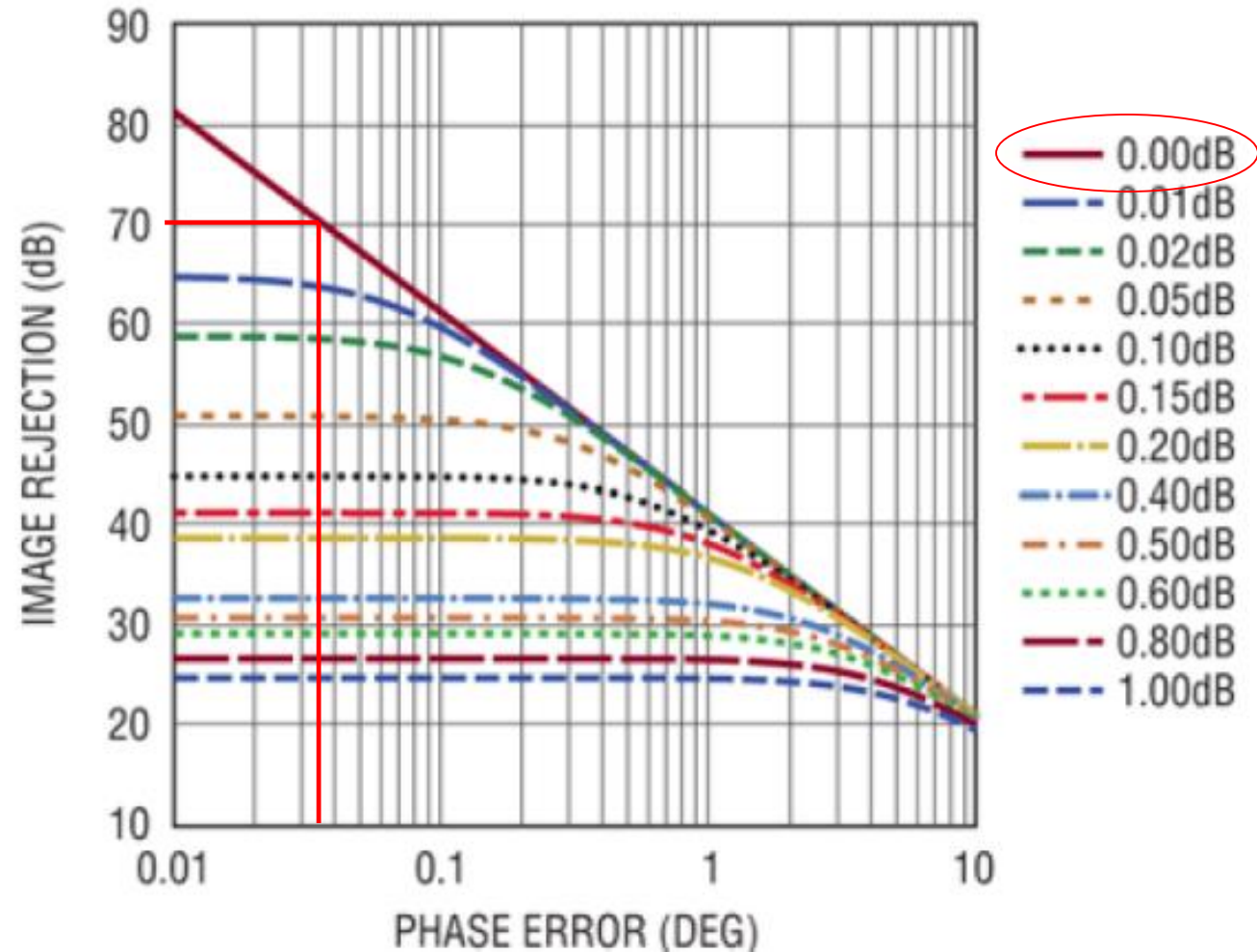
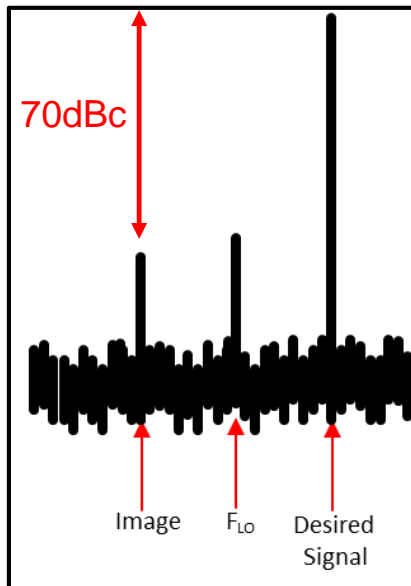


Figure C. Image Rejection vs Phase Error for Different I/Q Gain Mismatch

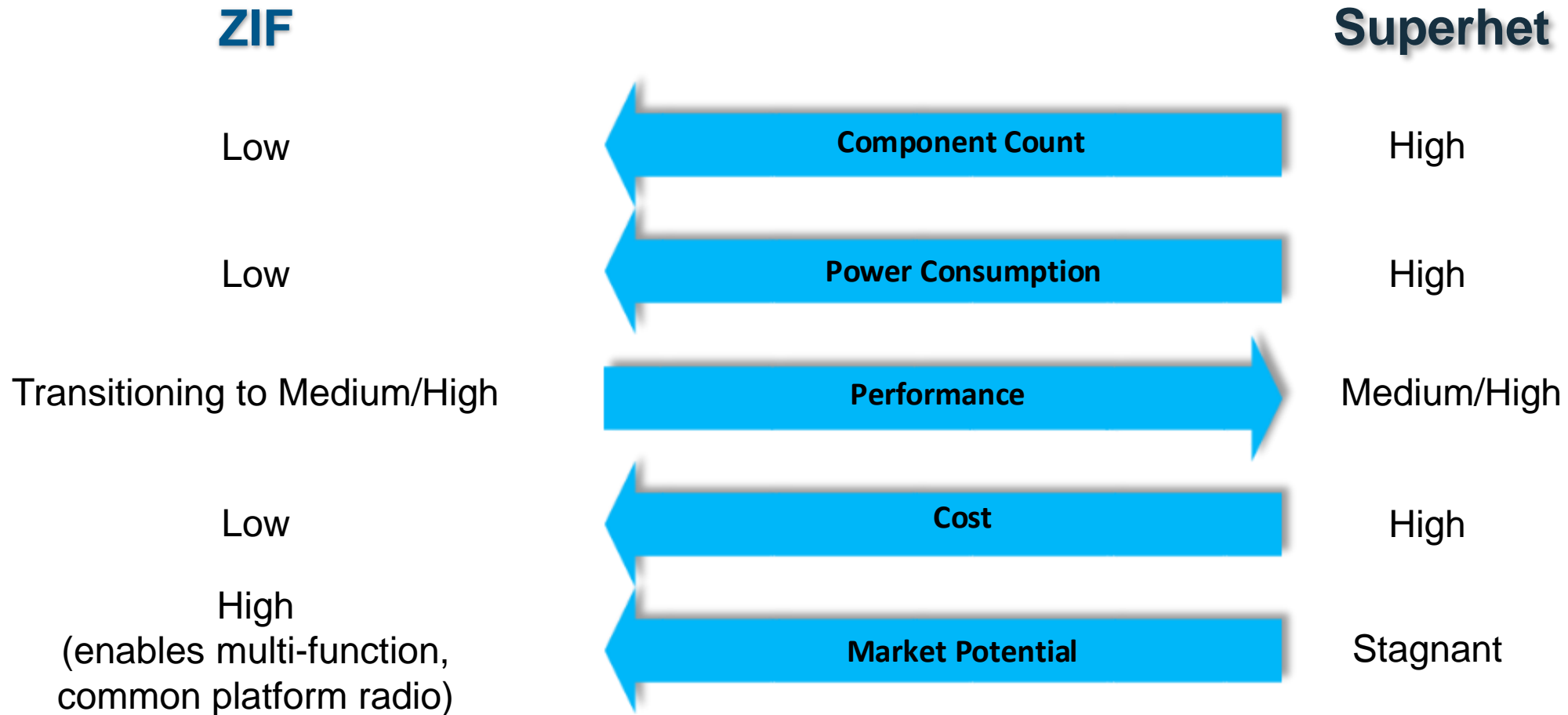
Image Rejection Challenges

► How hard is 70dB image rejection?

- 0.005dB I/Q gain error (11b matching)
- 0.035 deg phase error
 - ~ 30fs LO delay mismatch 4GHz
 - ~1ps LPF group delay matching



Summary of ZIF Architecture Comparison To Superhet



Zero IF SDR Transceivers

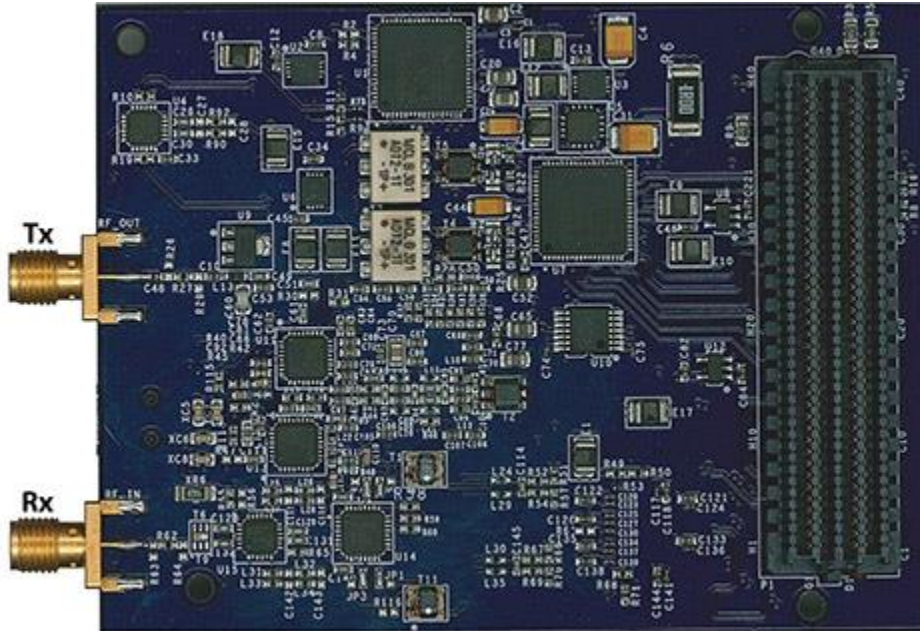
Catalina (AD936x)

Mykonos (AD9371)

Talise (ADRV9008/9)

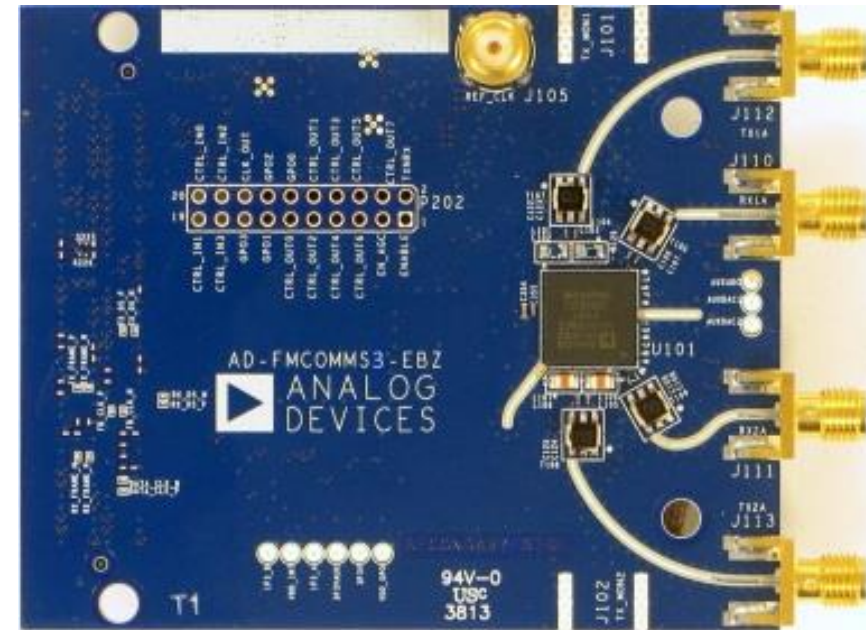
HIGH-LEVEL OVERVIEWS

Size and Power of Two ZIF Solutions



AD-FMCOMMS1

- Discrete ZIF, direct RF
- 1Rx, 1Tx
- 400 MHz – 4GHz tuning range
- 200+ MHz channel bandwidth
- Power, Clocks, ADC (AD9625), DAC (AD9162), PLL, DVGA
- Radio Power Consumption=4.5W



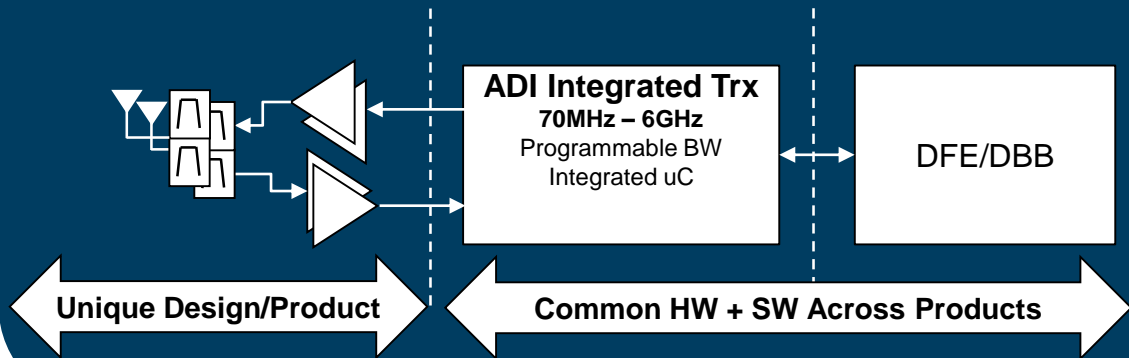
AD-FMCOMMS3

- AD9361 Integrated
- 2 x Rx, 2 x Tx
- 70 MHz – 6GHz tuning range
- 200kHz - 56 MHz channel bandwidth
- Power, AD9361
- Radio Power Consumption=0.7W

Wideband RF Transceiver Benefits

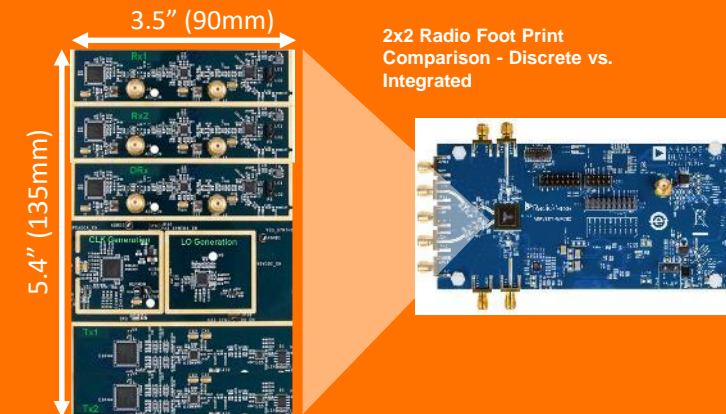
Highly Reconfigurable

Enables reduced time to market through common HW & SW
Small Signal Radio Platform



Highest Level of Integration

Enables higher density radio architectures e.g. M-MIMO

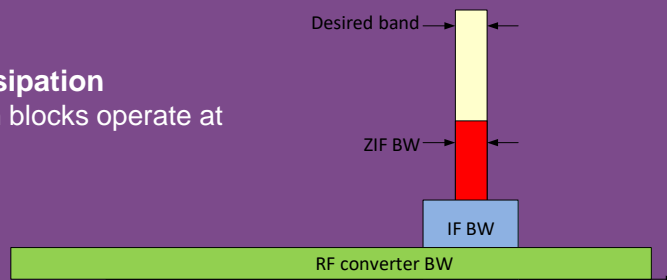


Lowest Power Consumption

Reduce thermal density, enable lower SWAP radios

Lowest possible power dissipation

- Highest power consumption blocks operate at minimum bandwidth



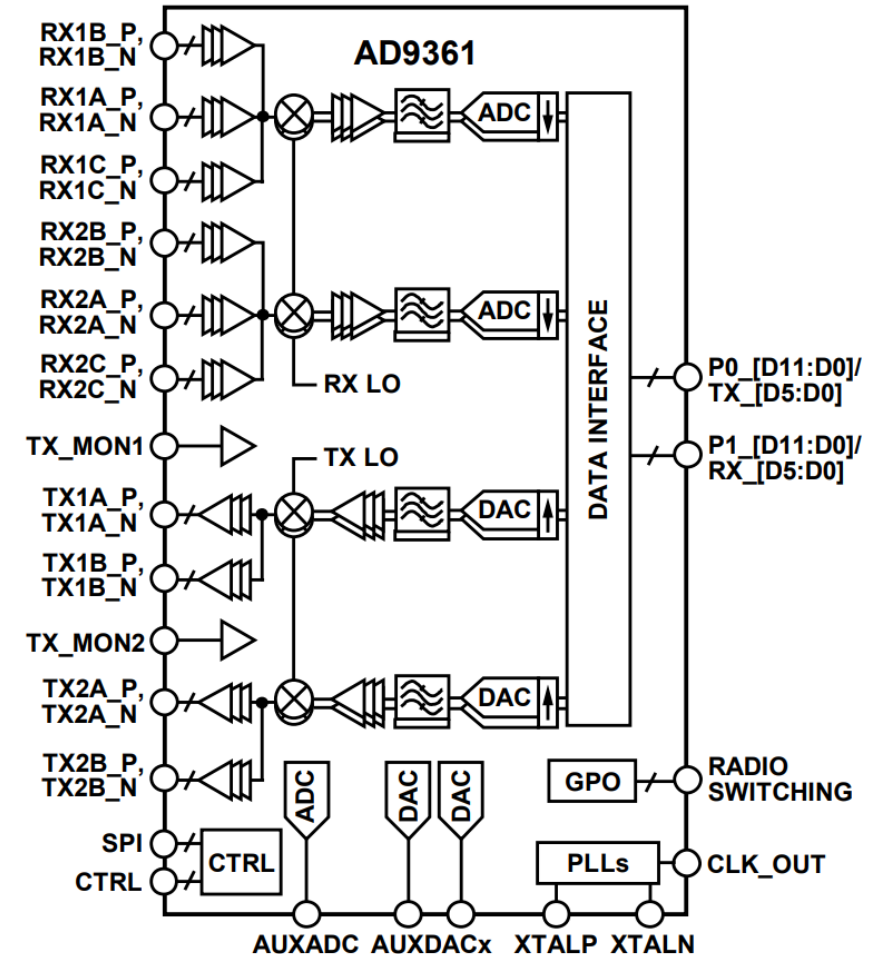
Lowest System Cost

Re-use of architecture used in handsets

- Components such as IF filters are eliminated
- RF filters are simplified enabled by the elimination of out-of-band images or aliases

AD9361: 2Rx/2Tx Integrated RF Transceiver (Catalina)

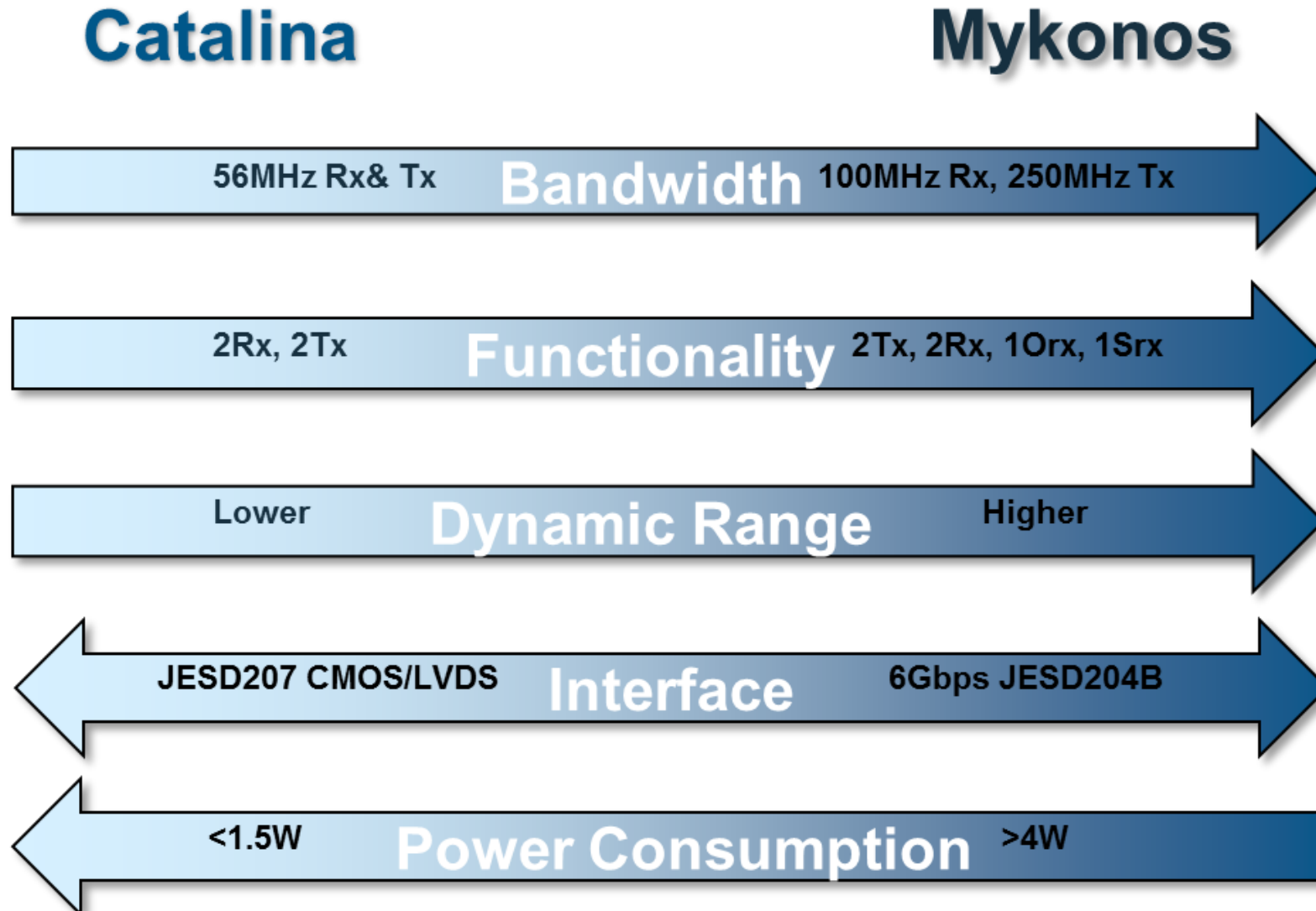
- ▶ 2Rx / 2Tx integrated RF transceiver
 - Tuning range: 70 MHz to 6GHz
 - Tunable channel BW: 200KHz to 56MHz
 - FDD/TDD operation
- ▶ Performance and power
 - Rx: 2.5dB NF
 - Tx: < -42dB Tx EVM
 - Tx Noise < -157dBm/Hz noise @ 70 MHz offset
 - Tx monitor: > 66dB dynamic range with 1dB accuracy
 - 12 bit ADCs/DACs
 - Phase noise: 0.25° @ 2.5 GHz
- ▶ Digital features
 - Rx: DC offset correction, quadrature calibration, AGC, programmable FIR filters
 - Tx: quadrature calibration, programmable FIR filters
- ▶ Typical power: **800-1100 mW**
 - 2Rx & 2Tx, 20 MHz BW, 0 dBm Tx power



Temp
-40°C – +85°C

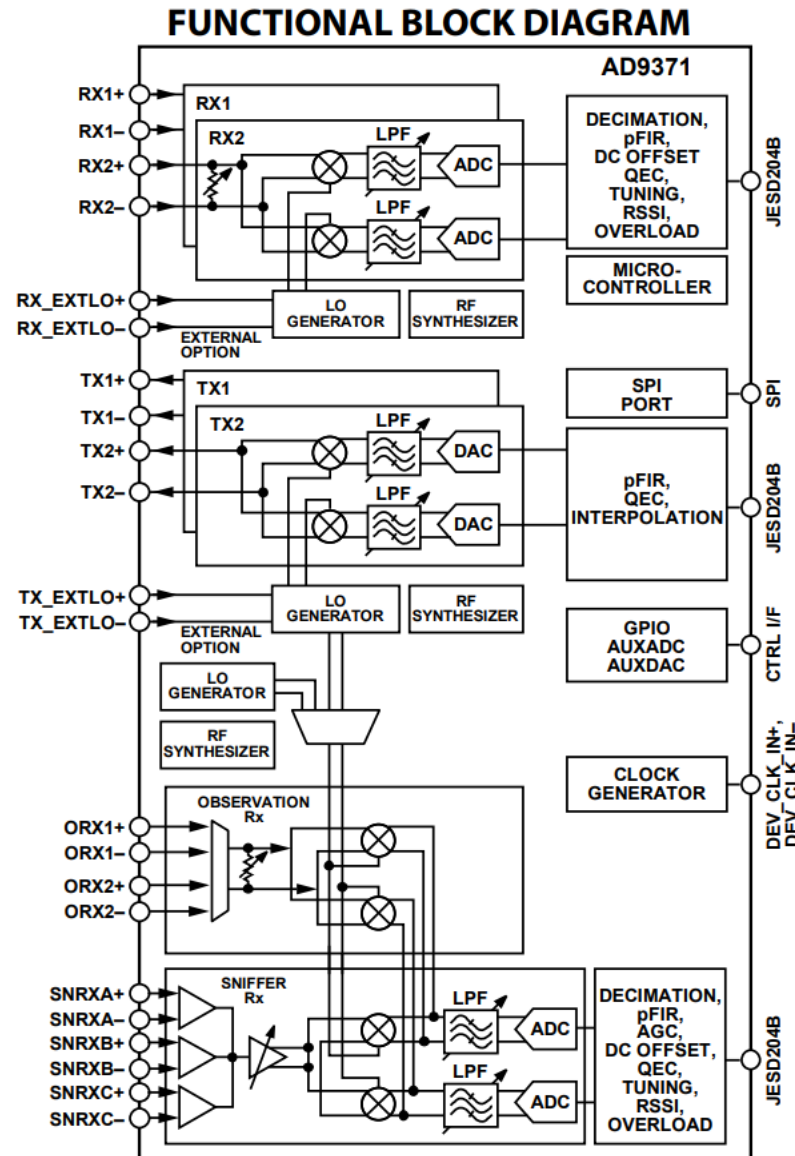
Package
144- CSP_BGA (10x10mm) Pb-Free

AD9361 (Catalina) vs AD9371 (Mykonos)



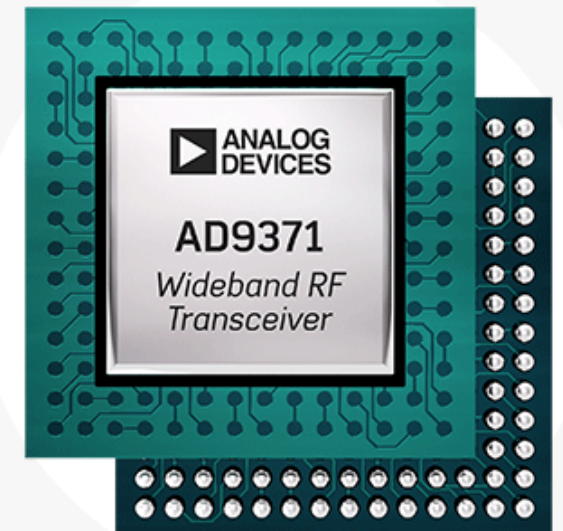
AD9371 Integrated, Dual RF Transceiver With Observation Path (Mykonos)

- ▶ Integrated dual-traffic Rx and Tx
 - Tuning range: $300\text{MHz} < F_c < 6\text{GHz}$
 - FDD/TDD operation
- ▶ Receiver
 - Max Rx BW = 100MHz
 - NF: 12dB
 - IIP_3 : 22dBm
 - IIP_2 : 65dBm
 - Gain range/step (dB): 30/0.5
- ▶ Transmitter
 - Max Tx BW = 250MHz
 - -64dB ACLR (4 UMTS Carriers)
 - OIP3: 27dBm (5dB atten)
 - Gain range/step (dB): 42/0.05
- ▶ Integrated observation and sniffer Rx
 - Max ORx BW = 250MHz
 - 2 inputs
 - Max SRx BW = 20MHz
 - Contains LNA
 - Dedicated LO
 - 3 inputs
- ▶ Total power (@ max bandwidth)
 - 2x Rx = 2.7W
 - 2x Tx = 3.7W
 - 2x Rx, 2x Tx, ORx = 4.86W
- ▶ Digital features
 - Tx/Rx QEC, DC offset, LO leakage
 - 6GSPS JESD204-B interface

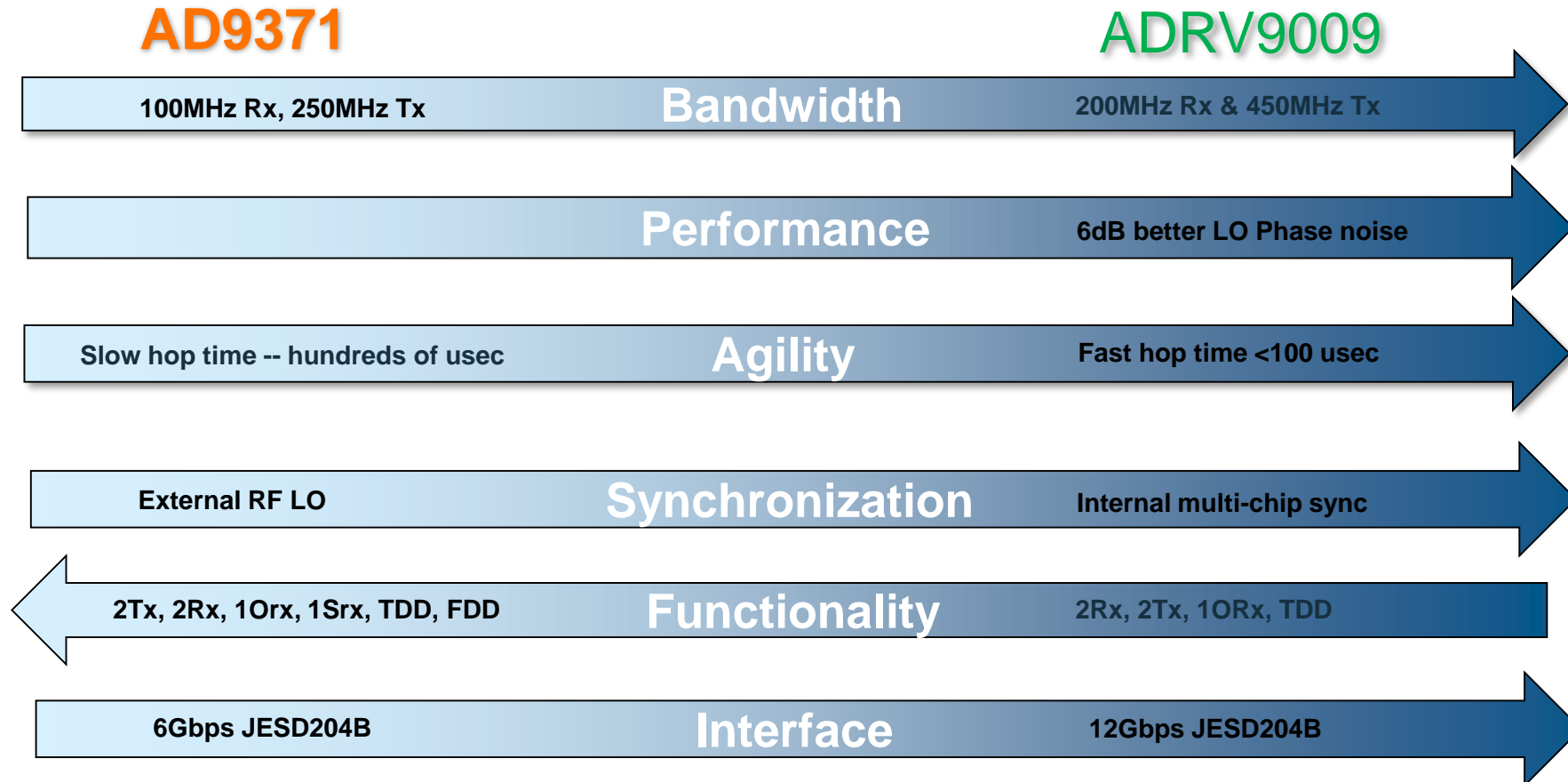


Package

196- CSP_BGA
(12x12mm) Pb-Free



AD9371 vs ADRV9009

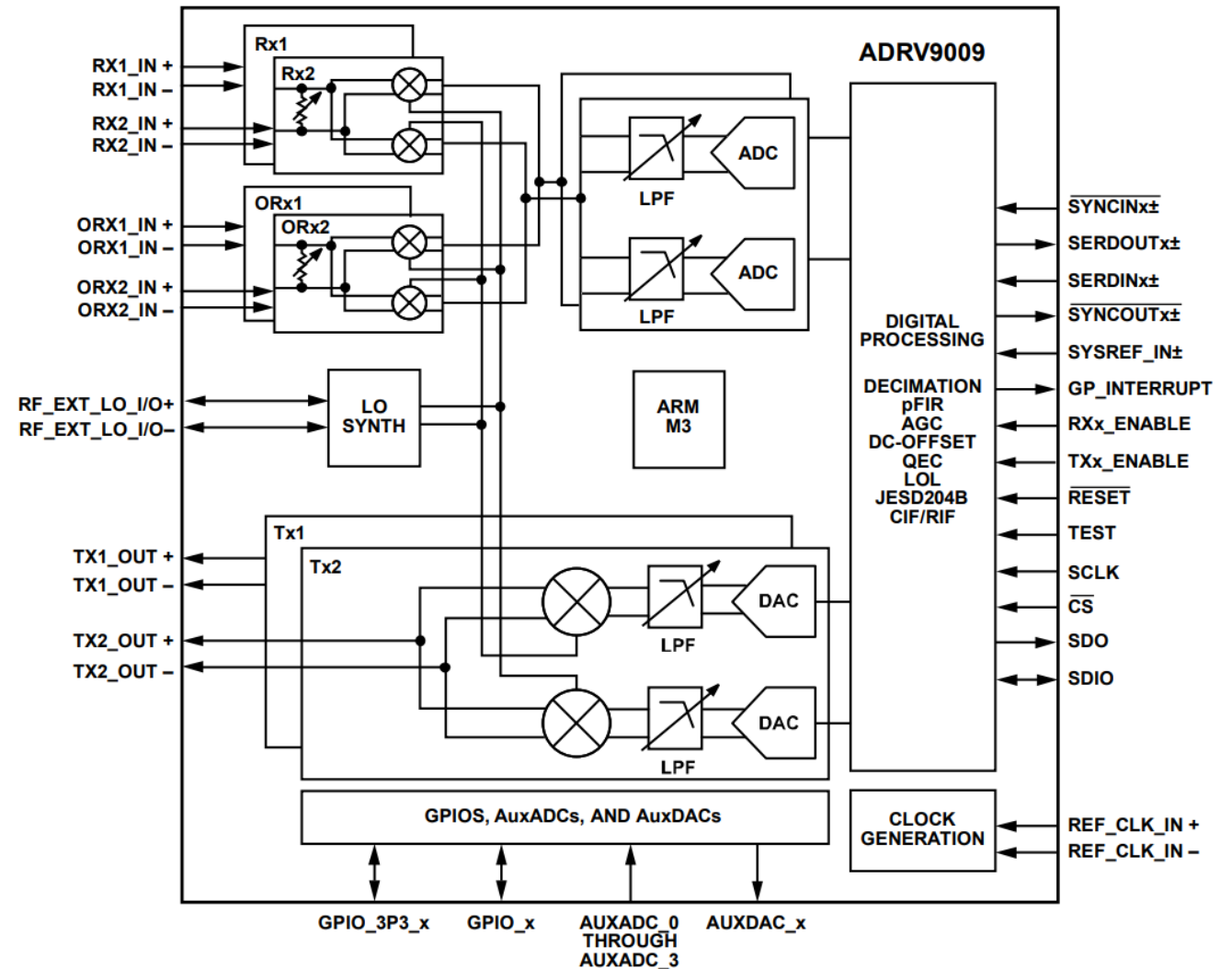


- ▶ Widest bandwidth, highest performance integrated radio solution from 75MHz to 6GHz
- ▶ Common platform design for 2G/3G/4G/5G base stations
- ▶ Supports multi-chip LO phase synchronization
- ▶ Enhanced frequency agility with fast frequency hopping and precalibration profiles

ADRV9009 Functionality & Block Diagram

- ▶ **TDD operation**
- ▶ **Bandwidth:** 200 MHz receiver, 450 MHz transmitter and observation receiver
- ▶ **Integration:** dual transmitters, dual receivers and observation receivers with shared input
- ▶ **Tuning Range:** 75MHz to 6GHz
- ▶ **Interface:** 12 Gbps JESD204B
- ▶ **Power Consumption:** 4.6W*
- ▶ **Multi-chip LO phase synchronization**
- ▶ **Package:** 12x12 BGA

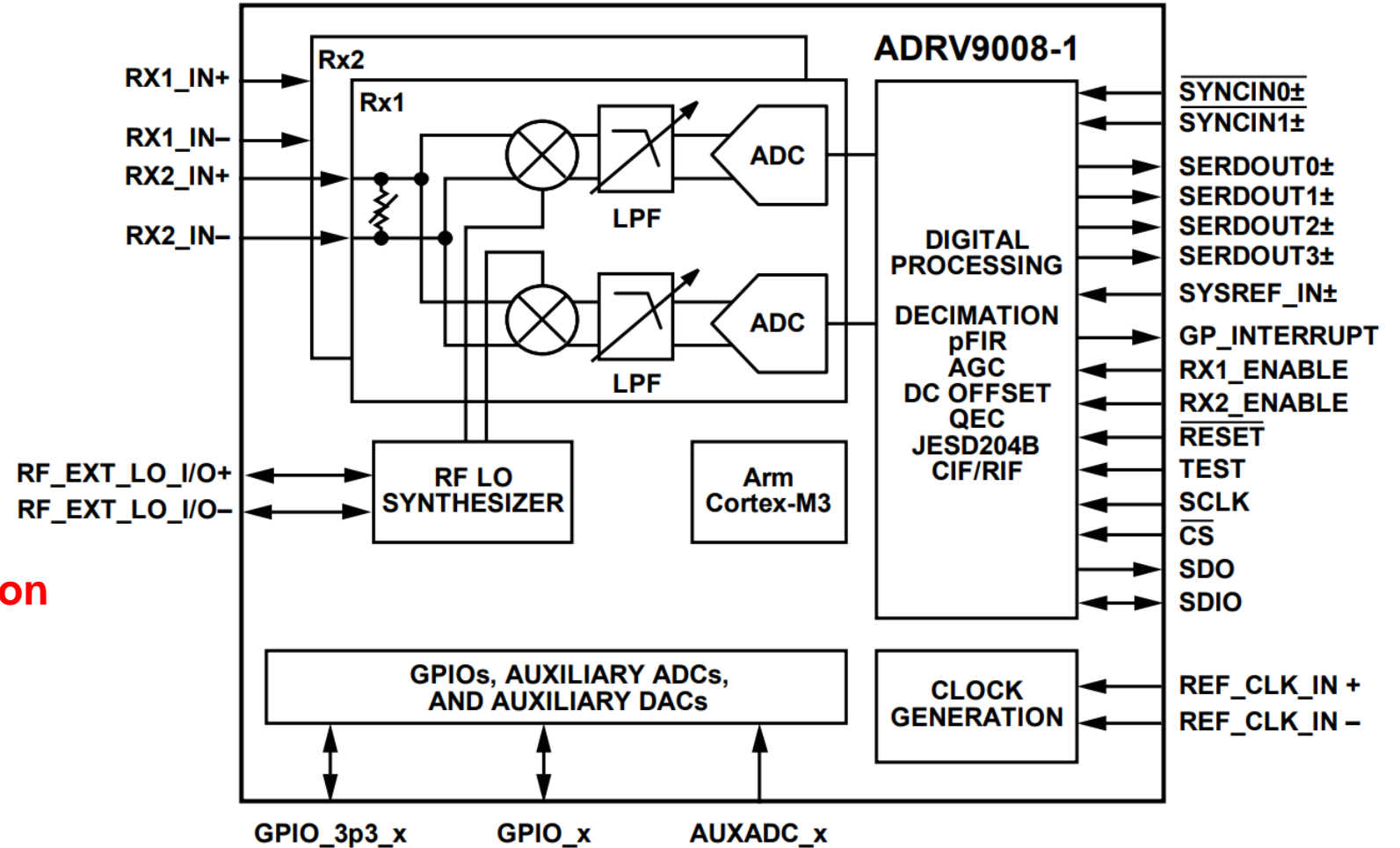
*For 50% Rx/Tx Duty Cycle, Orx on, 200MHz/450MHz BW, 0dB attenuation



ADRV9008-1 Functionality & Block Diagram

- ▶ **FDD Rx operation**
- ▶ **Bandwidth:** 200 MHz receiver
- ▶ **Integration:** dual receivers
- ▶ **Tuning Range:** 75MHz to 6GHz
- ▶ **Interface:** 12 Gbps JESD204B
- ▶ **Power Consumption:** 2.48W*
- ▶ **Multi-chip LO phase synchronization**
- ▶ **Package:** 12x12 BGA
- ▶ Pin compatible with ADRV9009

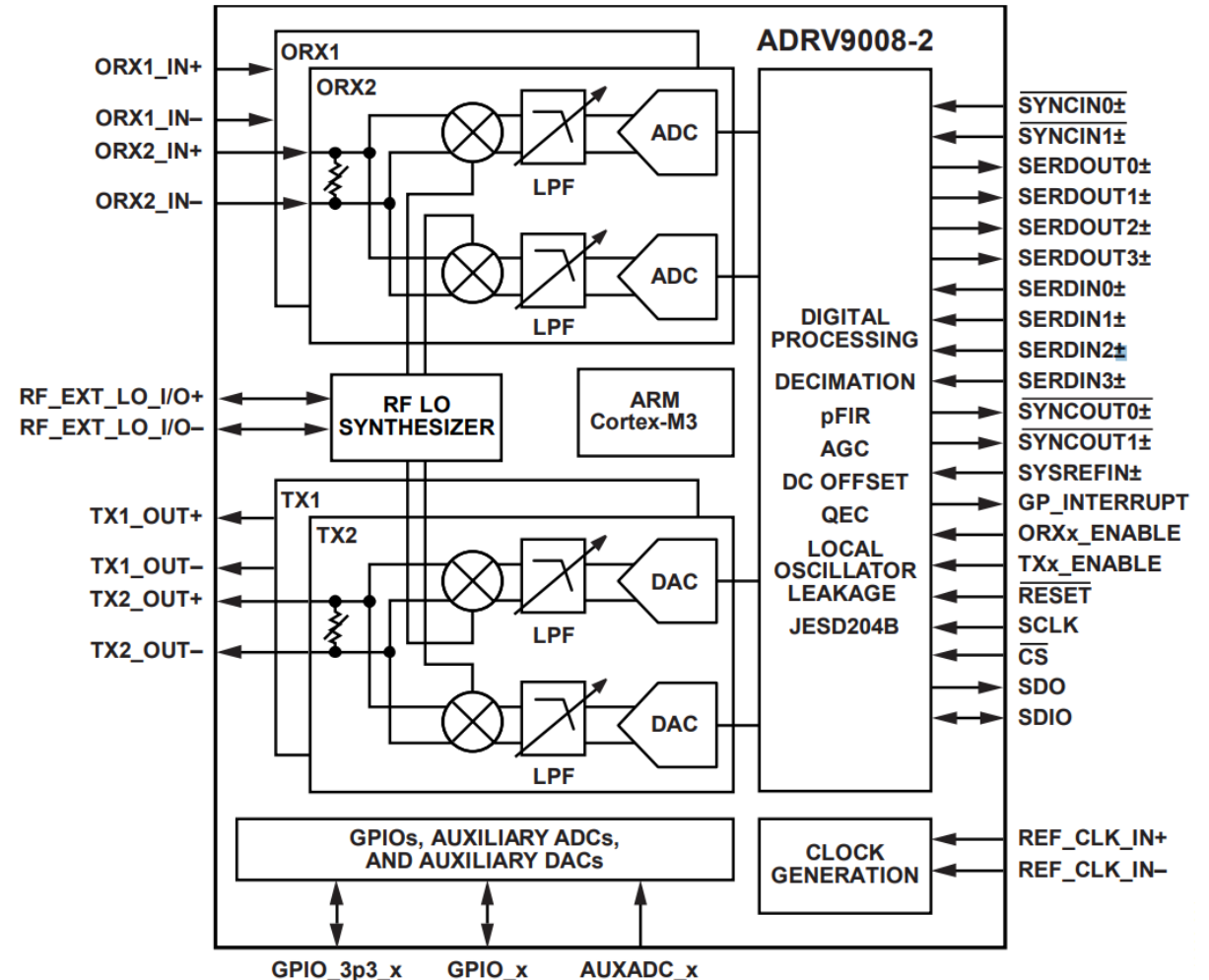
*For 2Rx, 200MHz BW



ADRV9008-2 Functionality & Block Diagram

- ▶ **FDD Tx/Orx operation**
- ▶ **Bandwidth:** 450 MHz transmitter and observation receiver
- ▶ **Integration:** dual transmitters, observation receiver with dual inputs
- ▶ **Tuning Range:** 75MHz to 6GHz
- ▶ **Interface:** 12 Gbps JESD204B
- ▶ **Power Consumption:** 4.34W Tx, 1.25W Orx *
- ▶ **Multi-chip LO phase synchronization**
- ▶ **Package:** 12x12 BGA
- ▶ Pin compatible with ADRV9009

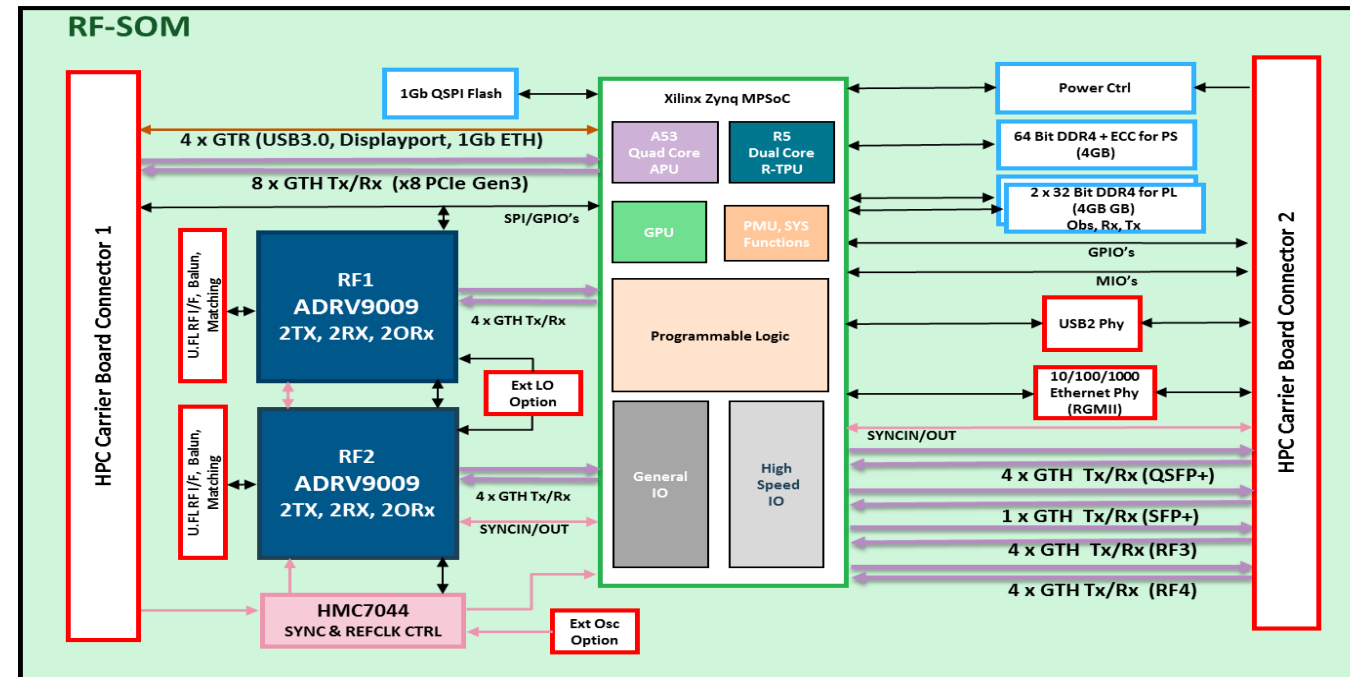
*For 450MHz BW, 0dB attenuation



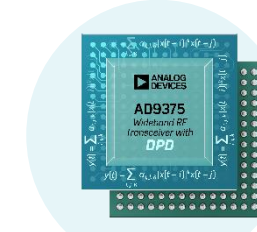
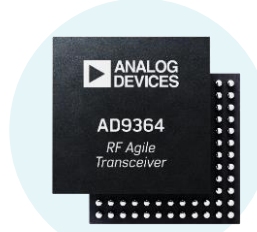
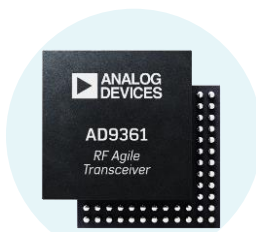
Introduce ADRV9009 System-on-Module (RF-SOM)

- ▶ Supports up to 4x ADRV9009 that can be synced in Freq & Phase
- ▶ Scalable with multiple RF-SOM's synced together
- ▶ I/O connector: USB 3.0, 10Gb Ethernet, PCIe x8
- ▶ 96mm x 160mm
- ▶ Comes with open source code support package hosted on GitHub
- ▶ Qualified 'production ready' module to speed up prototyping and integration into final production.
- ▶ Allows customers to focus on their own areas of differentiation
- ▶ Broad range of applications in cellular infrastructure, radar, portable defence and instrumentation

Engineering Sample
Available in 2018Q4



Wideband RF Transceiver Portfolio Released on RadioVerse™



Part #	Applications	Bandwidth	Functionality	RF Tuning Range	Rx Image Rejection*	Rx NF/IIP3**	Tx OIP3*	EVM	Package Size	Data Interface	Price
AD9361	3G/4G Picocell, SDR, Pt-Pt, Satcom, IoT Aggregator	56 MHz	2 Rx, 2 Tx	70 MHz to 6 GHz	50B	3dB/-14dBm	+19dBm	-40 dB	10 mm x 10 mm	CMOS/LVDS	\$175
AD9364	3G/4G Picocell, SDR	56 MHz	1 Rx, 1 Tx	70 MHz to 6 GHz	50dB	3dB/-14dBm	+19dBm	-40 dB	10 mm x 10 mm	CMOS/LVDS	\$130
AD9363	3G/4G Femtocell, UAV, Wireless Surveillance	20 MHz	2 Rx, 2 Tx	325 MHz to 3.8 GHz	50dB	3dB/-14dBm	+19dBm	-34 dB	10 mm x 10 mm	CMOS/LVDS	\$80
AD9371	3G/4G Macro BTS, Massive MIMO, SDR	100MHz Rx, 250MHz Tx	2Tx, 2Rx Orx & SnRx	300 MHz to 6GHz	75dB	1.6dB/+2dBm	+27dBm	-40 dB	12 mm x 12 mm	6GHz JESD204B	\$245
AD9375	3G/4G Small Cell, 3G/4G Massive MIMO	100MHz Rx, 250MHz Tx	2Tx, 2Rx Orx & SnRx	300 MHz to 6GHz	75dB	1.6dB/+2dBm	+27dBm	-40 dB	12 mm x 12 mm	6GHz JESD204B	\$325
ADRV9009	3G/4G/5G TDD macro cell, Massive MIMO, Phased array radar	200MHz Rx, 450MHz Tx	2Tx, 2Rx Orx	75 MHz to 6GHz	75dB	1.6dB/+2dBm	+27dBm	-43 dB	12 mm x 12 mm	12GHz JESD204B	\$319

* typical performance @ 2.6GHz

** AD9371 cascaded analysis with external LNA NF = 1.1dB, Gain = 19.5dB, IIP3 = 33dB (HMC8175A broadband LNA). Typical Performance @ 2.6GHz

** AD9361 assumes internal LNA, typical performance @ 2.6GHz

Intro to ADALM-Pluto

LEARN SDR CONCEPTS AND TOOLS WITH PLUTO!
GNU RADIO EDITION!



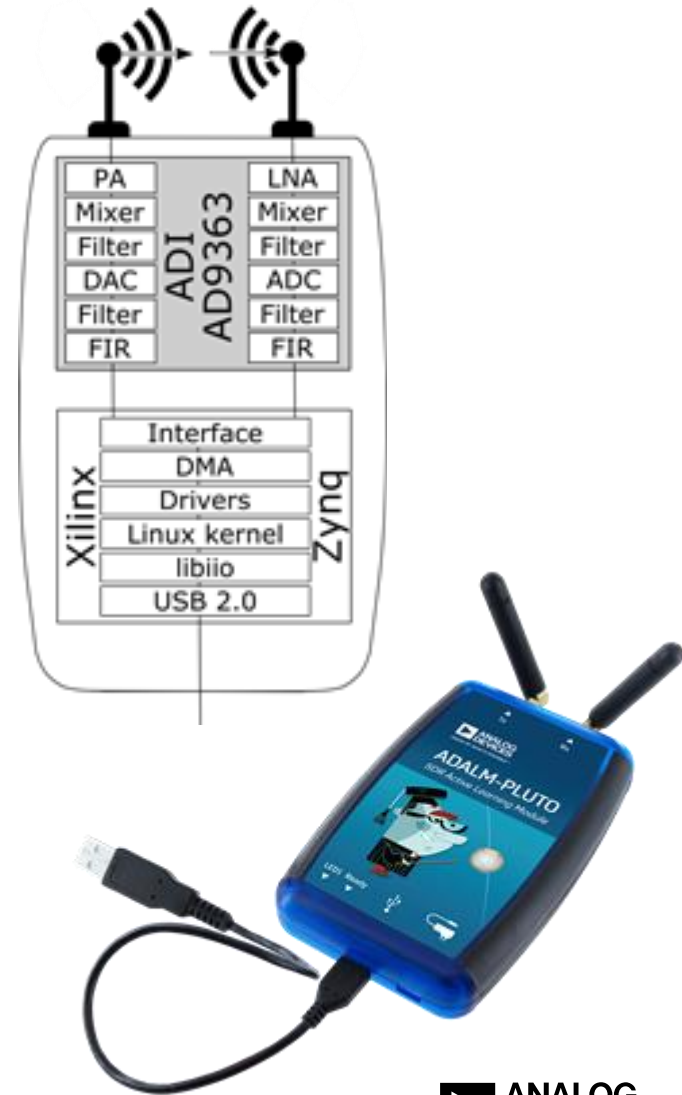
Agenda for Pluto Labs using GNU Radio

- ▶ Pluto Introduction
- ▶ Pluto Installation
 - IIO Oscilloscope
 - Verify Installation
 - Update Drivers (if necessary)
 - Install GNU Radio and Pluto Drivers
- ▶ Lab Exercises
 - Lab 1: Play an AM Station on an FM Radio
 - Lab 2: Hack an RF Outlet



Pluto is a Great Way to Get Started with Software Defined Radio!

- ▶ Pluto is a great way to get started with ADI's software defined radio products.
 - Pluto is a full AD9363 Transceiver with Xilinx Zync 7010 FPGA
 - The AD936x eval software and Design tools work with Pluto
 - Pluto is a low cost (\$150) solution to eval and learn the tools, vs. the full FPGA development boards (\$4k)
- ▶ www.analog.com/adalm-pluto
- ▶ Use with:
 - ADI's IIO-Scope (same program that AD936x eval boards use)
 - GNURadio, SDRangel, Matlab, etc.
- ▶ Free companion textbook here:
 - <https://www.analog.com/sdrforengineers>
- Online lectures here:
 - <https://www.youtube.com/playlist?list=PLBfTSoOqoRnOTBTLahXBIxaDUNWdZ3FdS>



What is GNU Radio?

- ▶ GNU Radio is a free, open source, toolkit that provides signal processing blocks to implement radios
 - <https://www.gnuradio.org/>
 - It is similar in many ways to Matlab Simulink
- ▶ GNU Radio was born in Linux, and it wants to live in Linux!
 - Windows is difficult for it..... We strongly recommend that you install onto a Linux based computer (Ubuntu LTS, etc.)
 - There is a Windows version, but YMMV.



Install Drivers and IIO Scope

► Download the USB drivers:

▪ Windows:

- <https://wiki.analog.com/university/tools/pluto/drivers/windows>

▪ Mac and Linux:

- <https://wiki.analog.com/university/tools/pluto/drivers/osx>

► Download the LIBIIO drivers:

- <https://github.com/analogdevicesinc/libiio>
- Scroll down to find the installation package for your OS:

► Download IIO-Oscilloscope:

- https://wiki.analog.com/resources/tools-software/linux-software/iio_oscilloscope

► More info on Pluto:

- <https://wiki.analog.com/university/tools/pluto/users>

Operating System	GitHub master status	Version	Primary Installer Package	Alternative Package, tarball or zip
Windows		Windows 10 Windows 8.1 Windows 8 Windows 7		
OS X		OS X High Sierra (v 10.13)		
		macOS Sierra (v 10.12)		
		OS X El Capitan (v 10.11)		
Linux		Ubuntu Bionic Beaver (v 18.04) ¹		
		Ubuntu Xenial Xerus (v 16.04) ¹		
		Ubuntu Trusty Tahr (v 14.04) ¹		
		CentOS 7		
		CentOS 6		

Install Option 1: GNU Radio Installation in Linux

► Installing GNU Radio in Linux:

- <https://wiki.analog.com/resources/tools-software/linux-software/gnuradio>

► Download the USB drivers:








- <https://wiki.analog.com/university/tools/pluto/drivers/osx>

► Download the LIBIIO drivers here:

- <https://github.com/analogdevicesinc/libiio>
- Scroll down to find the installation package for Linux

► Download IIO-Oscilloscope here:

- https://wiki.analog.com/resources/tools-software/linux-software/iio_oscilloscope

Operating System	GitHub master status	Version	Primary Installer Package	Alternative Package, tarball or zip
Windows		Windows 10 Windows 8.1 Windows 8 Windows 7		
OS X		OS X High Sierra (v 10.13)		
		macOS Sierra (v 10.12)		
		OS X El Capitan (v 10.11)		
Linux		Ubuntu Bionic Beaver (v 18.04) ¹		 
		Ubuntu Xenial Xerus (v 16.04) ¹		 
		Ubuntu Trusty Tahr (v 14.04) ¹		 
		CentOS 7		 
		CentOS 6		 

► Because we are using Linux with actual hardware, it is not recommended to use a Linux “Virtual Machine.” A dedicated Linux computer or a dual boot computer will give the best results.

► More info on Pluto:

- <https://wiki.analog.com/university/tools/pluto/users>

Install Option 2: GNU Radio Installation in Windows

- ▶ GNU Radio was born in Linux, and it wants to live in Linux!
 - Windows is difficult..... We strongly recommend that you install onto a Linux based computer (Ubuntu LTS, etc.)
 - There is a windows version, but YMMV. Use with many cautions:
 - https://ci.appveyor.com/api/buildjobs/3cigr6q3sb6tb7li/artifacts/gnuradio_3_7_11_iiosupport.msi
- ▶ Download the USB drivers:
 - <https://wiki.analog.com/university/tools/pluto/drivers/windows>
- ▶ Download the LIBIIO drivers:
 - <https://github.com/analogdevicesinc/libiio>
 - Scroll down to find the installation package for Windows:
- ▶ Download IIO-Oscilloscope:
 - https://wiki.analog.com/resources/tools-software/linux-software/iio_oscilloscope
- ▶ More info on Pluto:
 - <https://wiki.analog.com/university/tools/pluto/users>

Operating System	GitHub master status	Version	Primary Installer Package	Alternative Package, tarball or zip
Windows		Windows 10		
		Windows 8.1		
		Windows 8		
		Windows 7		
OS X		OS X High Sierra (v 10.13)		
		macOS Sierra (v 10.12)		
		OS X El Capitan (v 10.11)		
Linux		Ubuntu Bionic Beaver (v 18.04) ¹		 
		Ubuntu Xenial Xerus (v 16.04) ¹		 
		Ubuntu Trusty Tahr (v 14.04) ¹		 
		CentOS 7		  
		CentOS 6		  

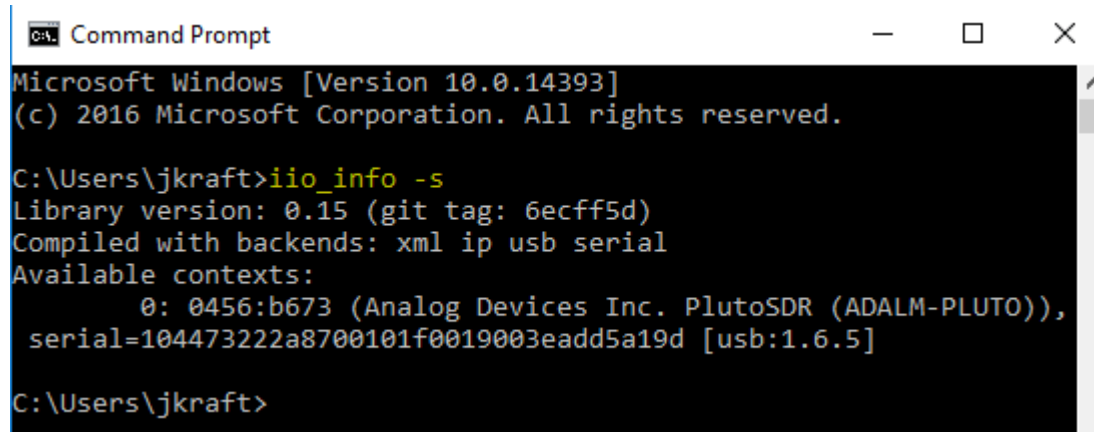
Now Check Your Installation:

► Plug Pluto into USB:

- You should see the blue “Ready” LED is on, and the blue “LED1” is blinking
- Only one USB cable is required. Plug this into the middle port with the USB symbol.
 - The other USB port, with the power plug icon, is for power only. So if you wanted to run an automated script on Pluto, with no computer connected, then you could power Pluto from this port. But that’s a topic for another day....

► Then verify that the LIB IIO drivers have been installed

- Open the command prompt
- Type “iio_info -s”
- You should see something like this:



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\jkraft>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=104473222a8700101f0019003eadd5a19d [usb:1.6.5]

C:\Users\jkraft>
```

- Library version refers to the IIO scope driver library. If you don’t see it, then you didn’t install IIO Scope drivers properly
- The serial number and USB information (i.e. 1.6.5) will be unique to your Pluto and computer. But if you see it, then it means that your computer can see the Pluto device on USB

Update Pluto (if necessary)

- ▶ Update Pluto Firmware here:

- <https://wiki.analog.com/university/tools/pluto/users/firmware>

- ▶ Widen the RF interface of Pluto to 70M to 6 GHz, and BW to 56MHz. Go to “Updating to the AD9364” on this page:

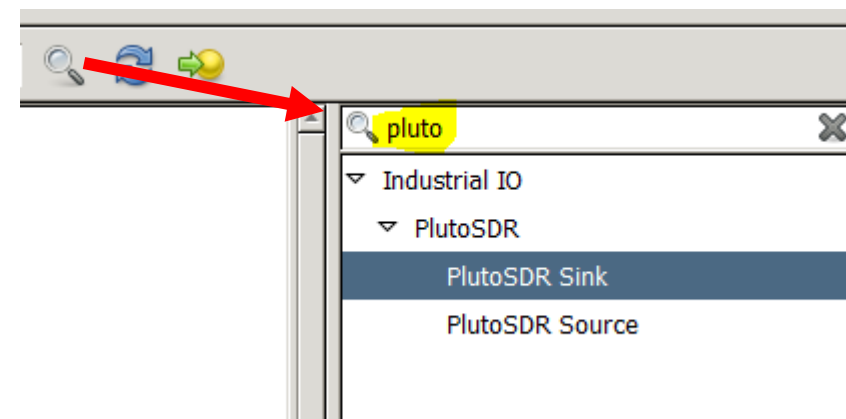
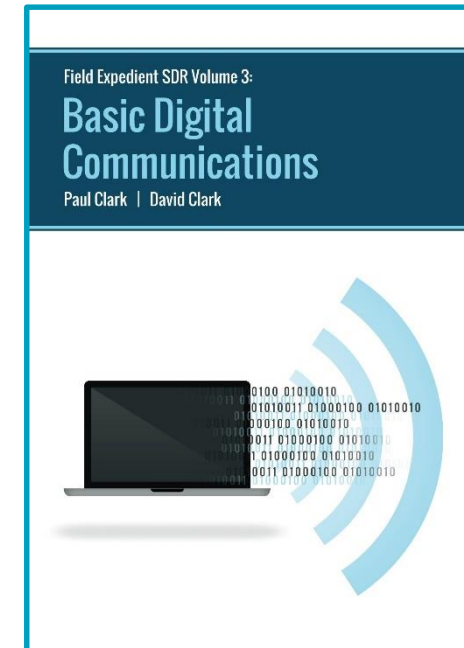
- <https://wiki.analog.com/university/tools/pluto/users/customizing>

- ▶ More info on Pluto here:

- <https://wiki.analog.com/university/tools/pluto/users>

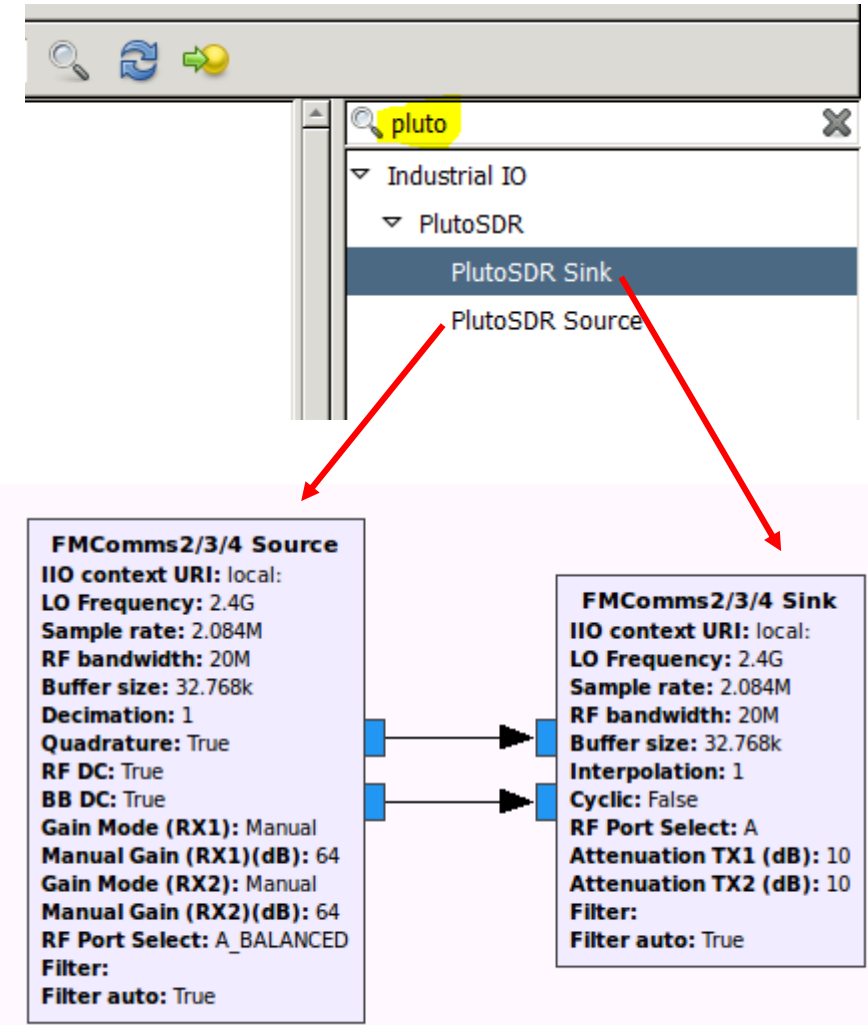
Getting Started with GNU Radio

- ▶ Check out the Field Expedient SDR series
 - It is a GREAT resource for SDR and GNU Radio:
 - <http://fieldxp.com/>
 - You can run all of their examples with the Pluto SDR
 - They also have great instructions on installing GNU Radio:
 - <http://fieldxp.com/install/>
- ▶ Also check out Great Scott Gadgets online tutorials:
 - <https://greatscottgadgets.com/sdr/>
- ▶ Check that Pluto is in GNU Radio:
 - Click find and type “pluto”
 - If you don’t see PlutoSDR Sink and Source, then you need to check you followed these instructions:
 - <https://wiki.analog.com/resources/tools-software/linux-software/gnuradio>



Using GNU Radio with Pluto

- ▶ The Pluto “Source” and “Sink” Blocks are well described here:
 - <https://wiki.analog.com/resources/tools-software/linux-software/gnuradio>
- ▶ For “IIO Context URI”, just leave blank. Works most of time...
 - If you get a “No Pluto Device Found” error, then enter the USB #
 - USB number is obtained with the `iio_info -s` command prompt (shown a few slides above)
 - It was USB: 1.6.5 for that example
 - Then the URI field is: USB: 1.6.5
 - Also if you are connecting multiple Plutos to USB, then you’ll need to use the USB port number also
- ▶ “Buffer Size”
 - Buffer size is the number of samples are output from Pluto during one USB burst.
 - Generally leave this at the default of 0x8000 (32768)
 - You can change the buffer, but it must be in increments of 1024
 - The max buffer size is 2^{23} (8388608, or 0x800000) samples.
 - When streaming, we want our buffers to be contiguous, but as we increase the same rate, the USB port or the computer may not be able to keep up. So if you are doing a high sample rate (i.e. 61.44 MSPS) then you may want to increase the buffer size. You still won’t be streaming data. But each buffer of data you get back will have contiguous samples in it. However, from buffer to buffer, you wouldn’t get contiguous samples (at high sample rates).



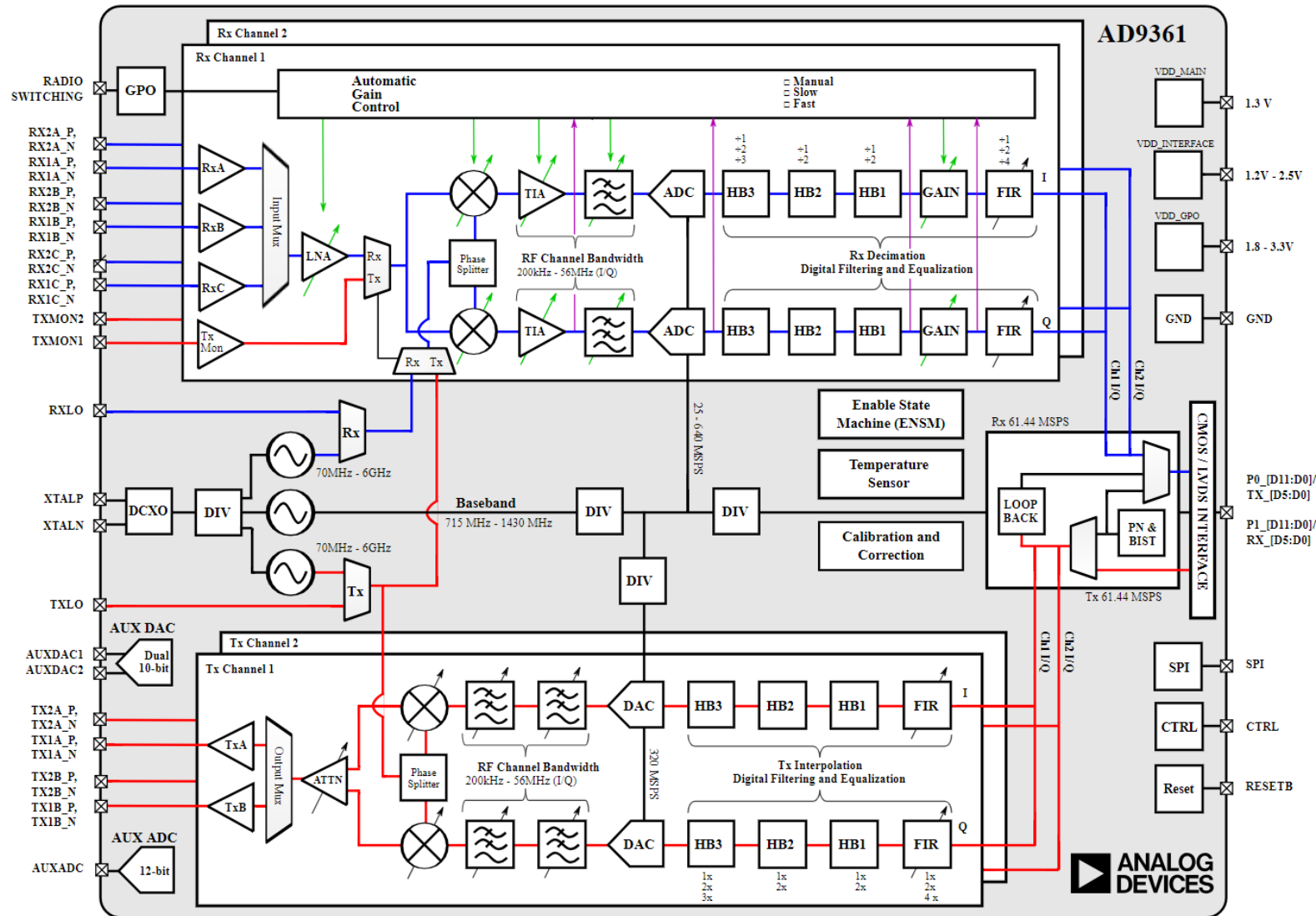
Some Background Info Before We Start Our Labs

Warning About Transmitting!

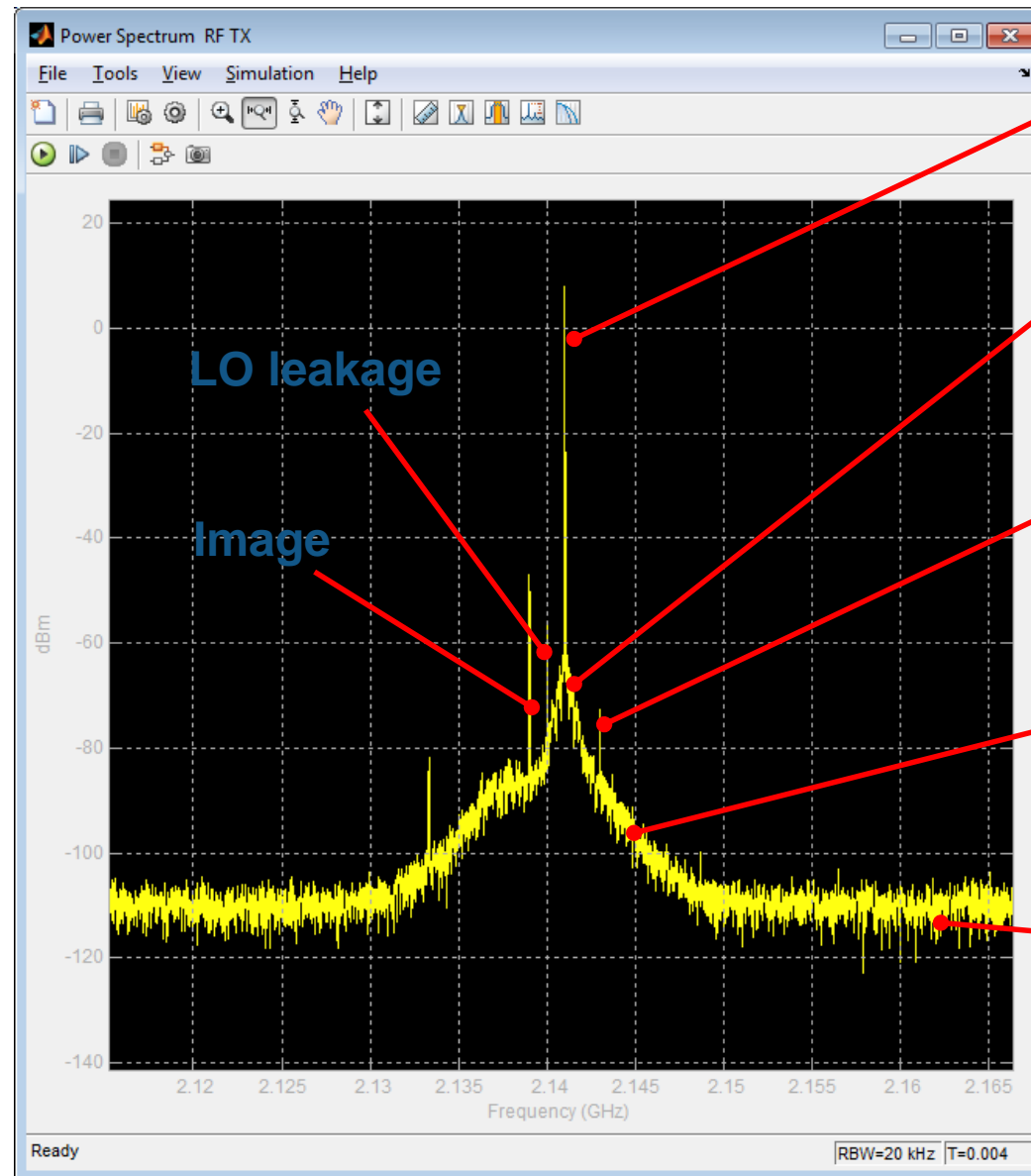
- ▶ Transmit only when and where you are legally allowed to!
- ▶ This is your responsibility to know.
- ▶ There are many emergency radio bands
 - https://en.wikipedia.org/wiki/International_distress_frequency
 - Note the civilian aircraft emergency band at 121.5 MHz. Stay far away from this!!!!
- ▶ DO NOT transmit ANYTHING on any of these bands
 - Either intentionally or unintentionally (via harmonics)
 - These bands are always being watched and authorities will triangulate your “distress” call!
- ▶ Use the ISM bands!
 - 864 to 870 MHz, 2.4 to 2.5 GHz, and 5.725 to 5.875 GHz.
- ▶ So be careful about your settings!
 - Use Pluto’s programmable bandwidth filter to filter your transmissions to the MINIMUM bandwidth
 - Transmit at the MINIMUM power level (put your Tx antenna close to your Rx antenna)
 - Choose your LO frequency so that you are away from these bands.
- ▶ I am not a lawyer! Nor am I offering legal advice! Learn what is permitted in your country!
 - Here’s the FCC (USA) guidance for low power transmission in the FM Bands:
 - <https://www.fcc.gov/media/radio/low-power-radio-general-information>



Details of the AD936x (Refer back to this slide during the labs)



Typical TX Spectrum for a Direct Conversion Receiver



Typical RX Spectrum for a Direct Conversion Receiver

Typical RX Spectrum:

In-band noise

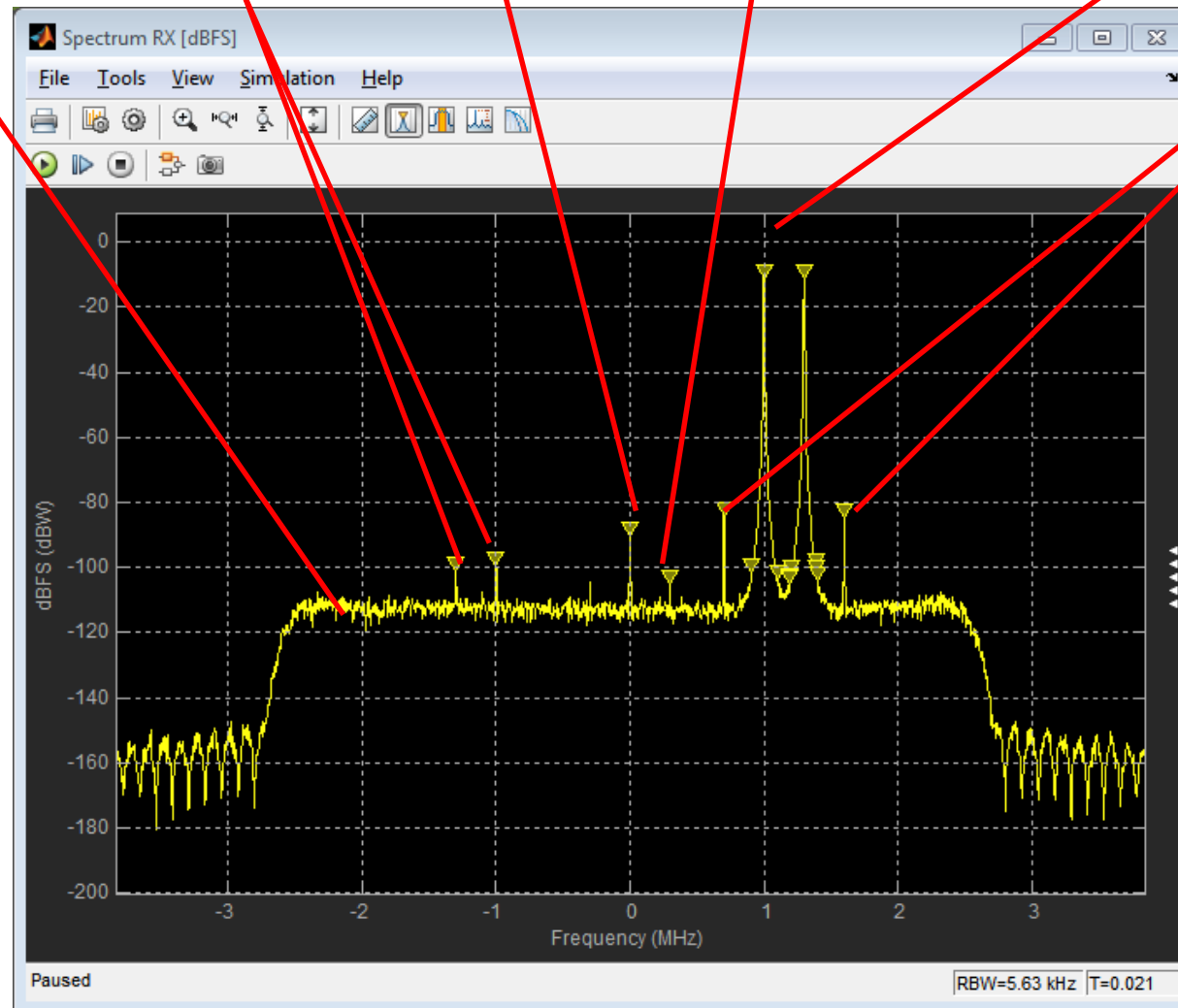
Images

DC Offset

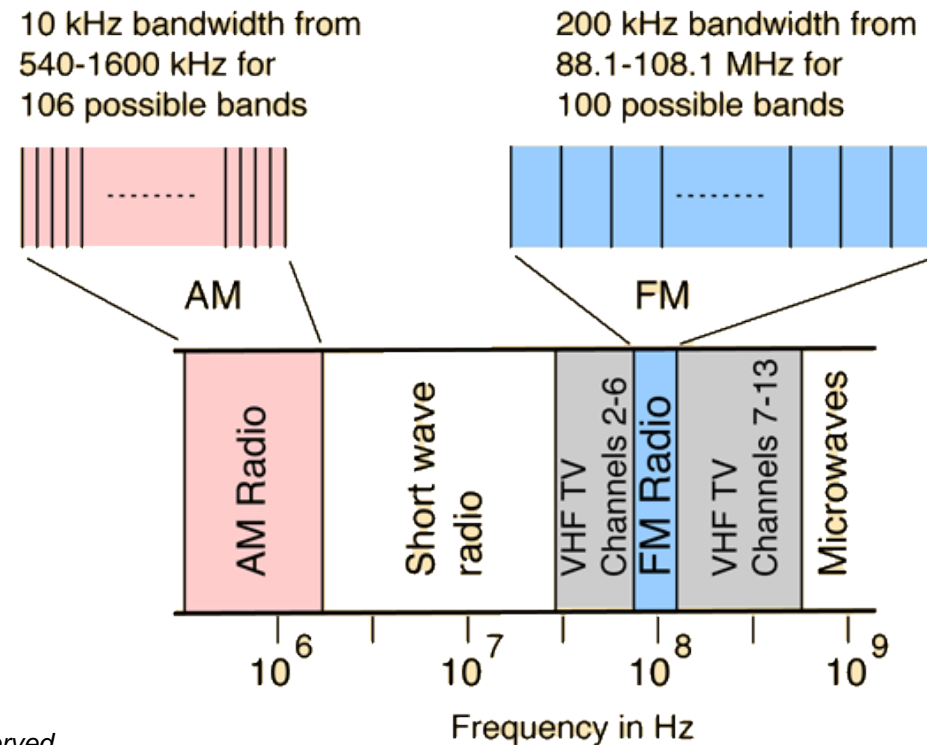
IM2

Fundamental

IM3

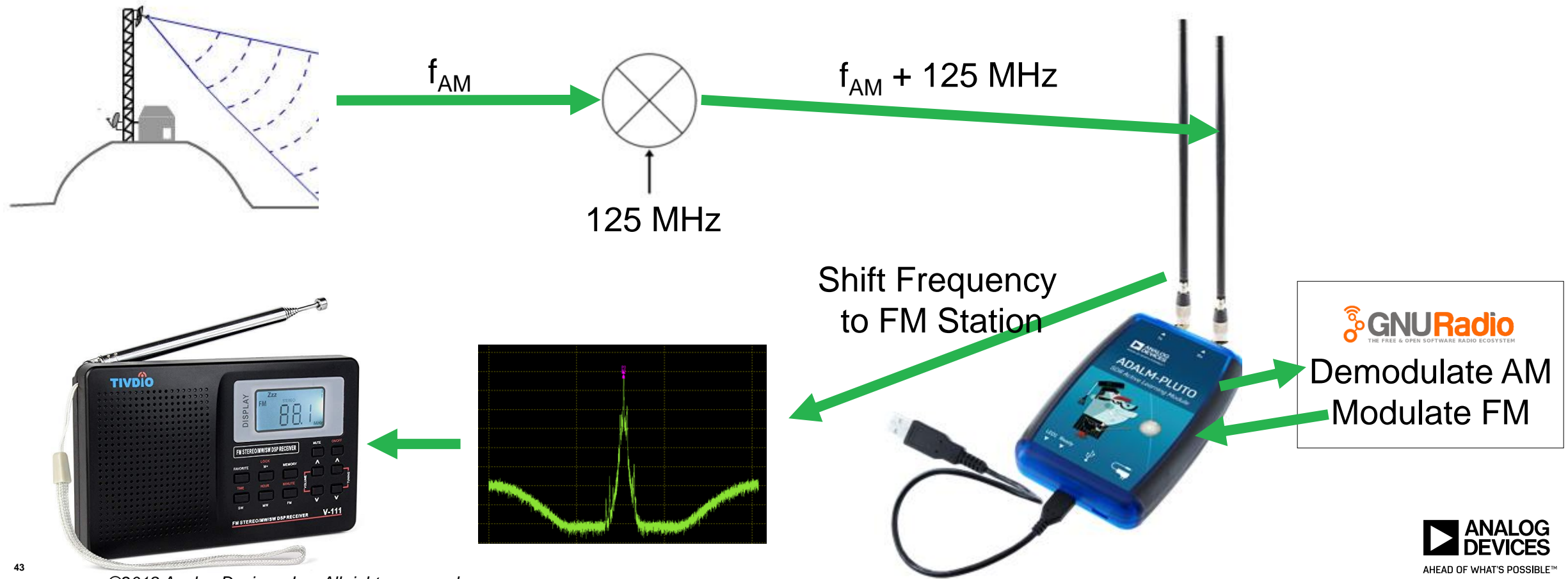


Lab 1: Play an AM Station on an FM Radio



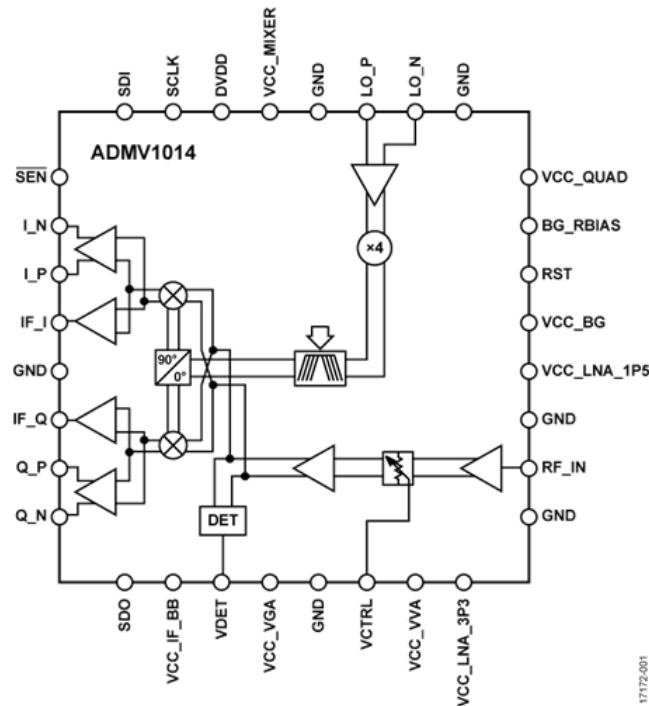
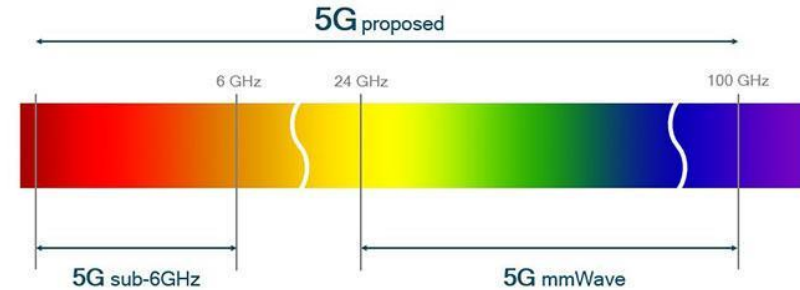
Pluto Lab 1: Play an AM Station on an FM Radio

- ▶ For this lab, we'll convert an AM station to FM and send it to an FM station that we choose
- ▶ AM is around 1000kHz. That's too low for Pluto, so we use a mixer to shift up by 125MHz

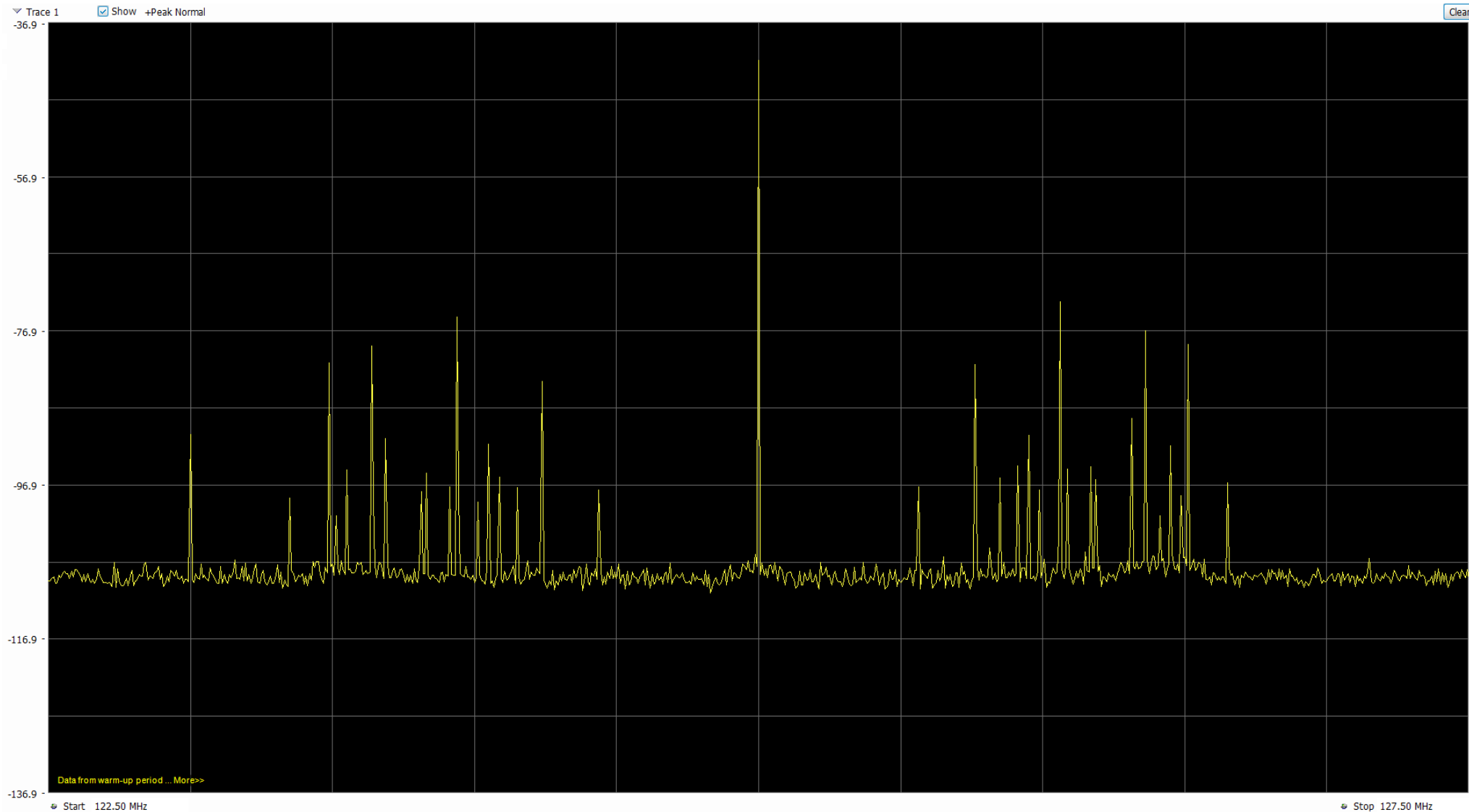


Pluto Lab 1: Play an AM Station on an FM Radio

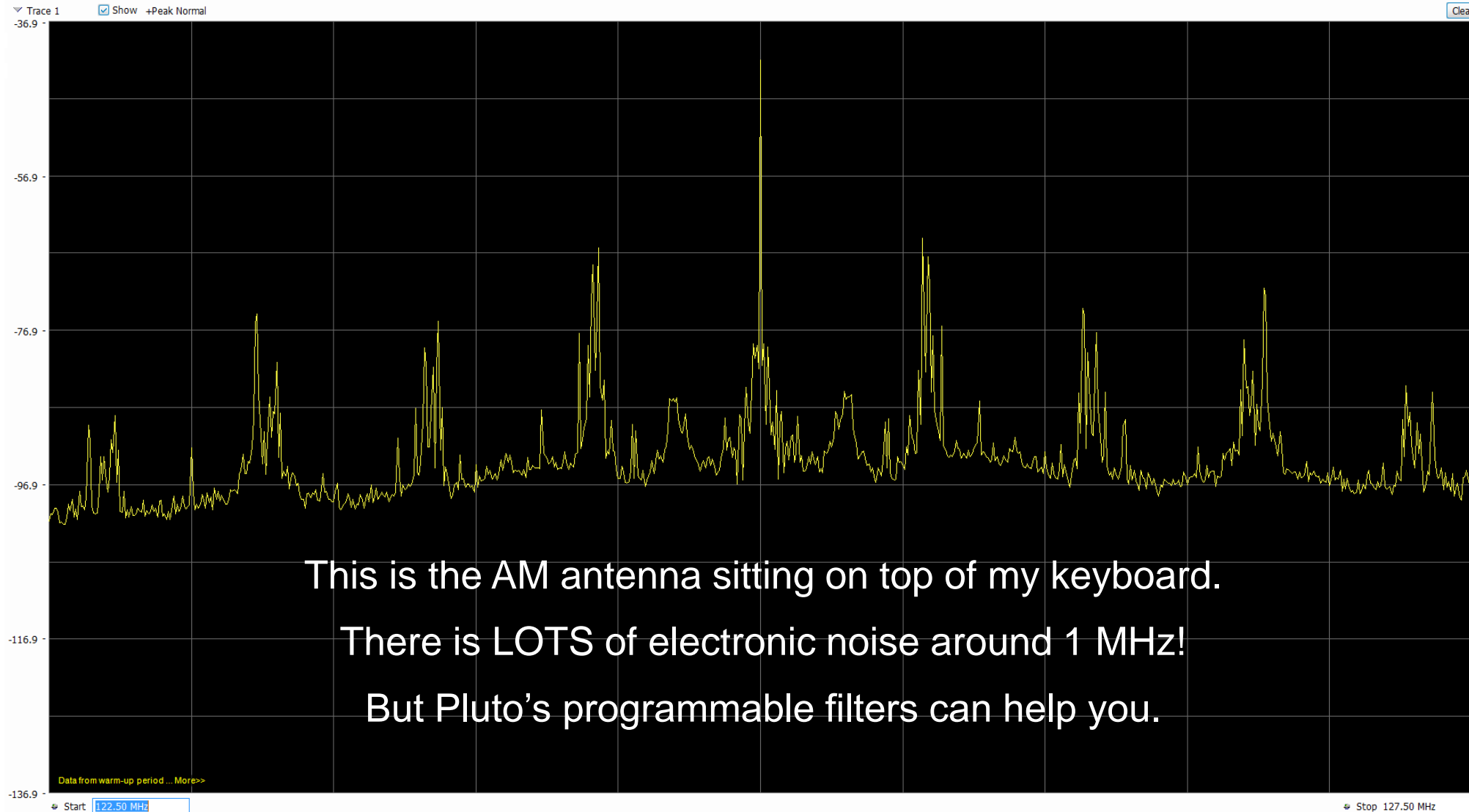
- ▶ Most of our TRx products go from 70 MHz to 6 GHz
- ▶ We don't often mix up from the 1 MHz AM freq.....
- ▶ BUT we will often mix down from X band or mmW Bands (like the 5G 28 and 39 GHz bands)



AM Bands After Mixing Up by 125 MHz



The AM Antenna Can Pick Up a Lot of Noise! Use the Pluto Filters!



AD9361 Programmable Filters

► Programmable Analog Filter

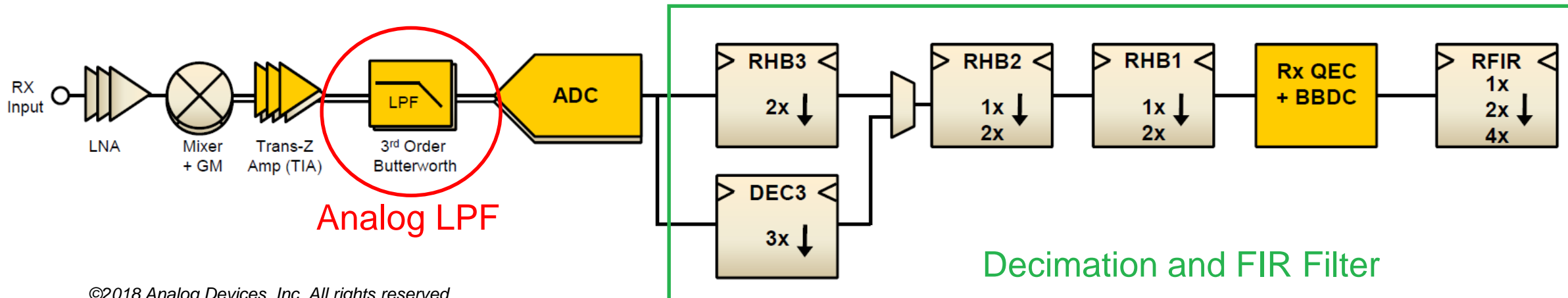
- Set the 3dB corner with “RF Bandwidth”
- It is a 3rd Order Butterworth Low Pass Filter

► Programmable FIR Filter

- ADC is a sigma delta, so it is a highly oversampled ADC. Sampling around 1 GHz, then DDC through a series of HB and digital filters to get the data rate
- All of the decimating filters are configurable
- FIR filter is 128 tap and it is programmable.
- You can add your own FIR coefficients.
- Recommend to use the “Filter Wizard” to program

FMComms2/3/4 Source
IIO context URI: local:
LO Frequency: 2.4G
Sample rate: 2.084M
RF bandwidth: 20M
Buffer size: 32.768k
Decimation: 1
Quadrature: True
RF DC: True
BB DC: True
Gain Mode (RX1): Manual
Manual Gain (RX1)(dB): 64
Gain Mode (RX2): Manual
Manual Gain (RX2)(dB): 64
RF Port Select: A_BALANCED
Filter:
Filter auto: True

FMComms2/3/4 Sink
IIO context URI: local:
LO Frequency: 2.4G
Sample rate: 2.084M
RF bandwidth: 20M
Buffer size: 32.768k
Interpolation: 1
Cyclic: False
RF Port Select: A
Attenuation TX1 (dB): 10
Attenuation TX2 (dB): 10
Filter:
Filter auto: True

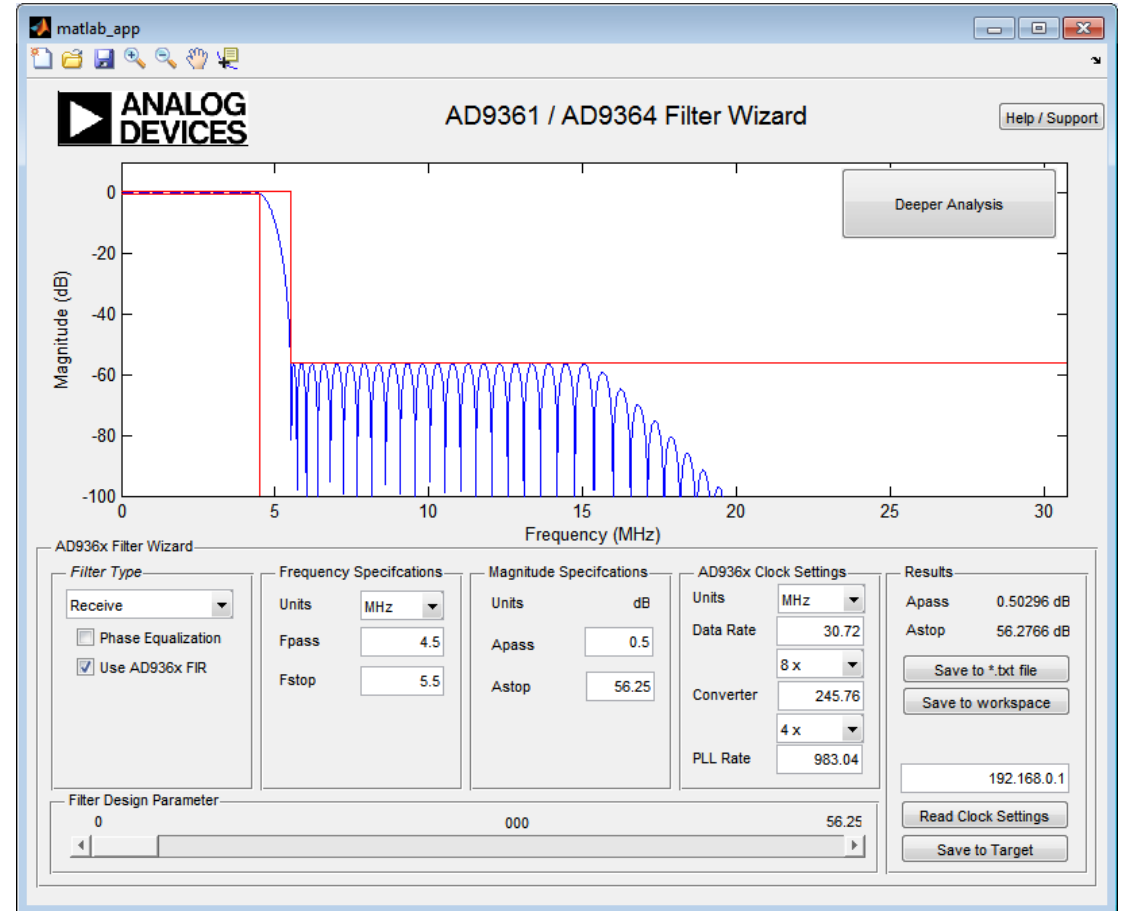


AD9361 Filter Wizard

Enables users to:

- ▶ Choose correct halfband digital filters for receive and transmit
- ▶ Design the programmable FIR filters for custom applications
- ▶ Examine the independent and composite response of the filters

Available within Matlab, or C code



Pluto Lab 1: Play an AM Station on an FM Radio

► For this lab, we'll need:

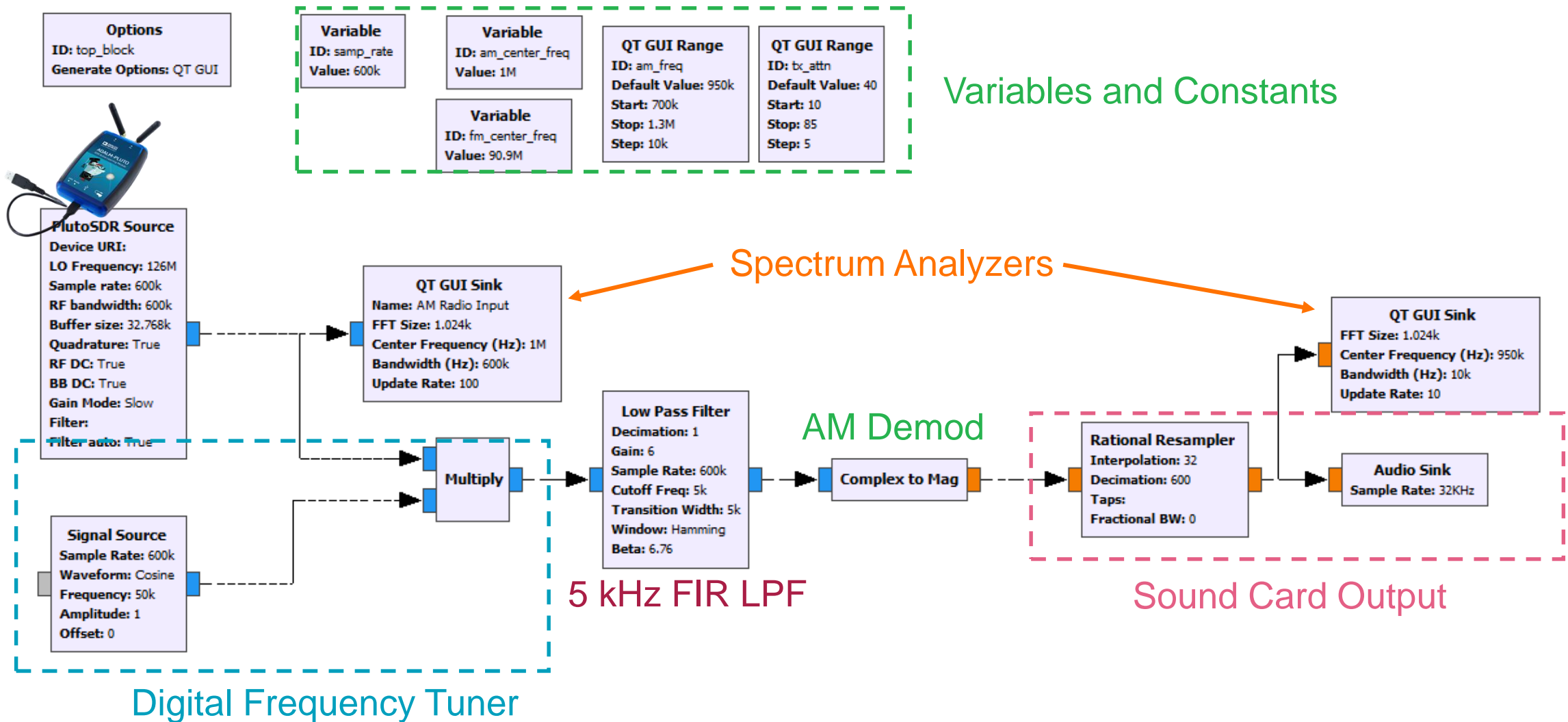
- Pluto SDR
- GNU Radio
- One AM antenna (or a big ol spool of wire)
- One FM antenna (the antennas included with your Pluto kit aren't great from FM freqs)
- A mixer to mix AM up to something >70 MHz (the lower limit of Pluto)
- An FM radio

► For the mixer, try “Ham It Up”

- <https://www.nooelec.com/store/sdr/sdr-addons/ham-it-up-plus.html>
- Now Pluto can Tx or Rx from **300 Hz** to 6 GHz!!!
- Just give add 125 MHz to Pluto's LO
 - i.e. a 760kHz AM station means Pluto's Rx LO needs to be 125.760 MHz

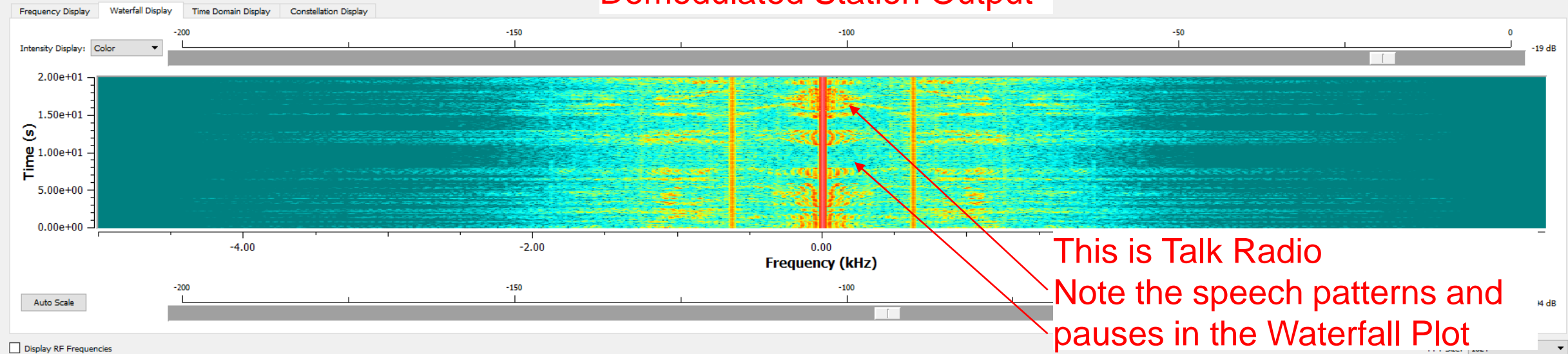


Lab1: GNU Radio Flow Chart

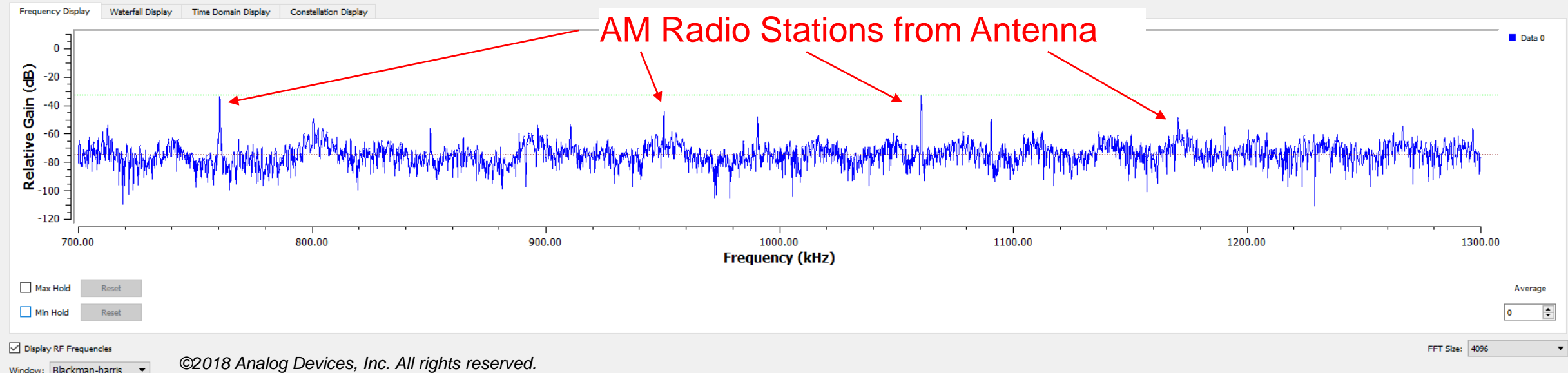


Lab 1: AM Radio Spectrums:

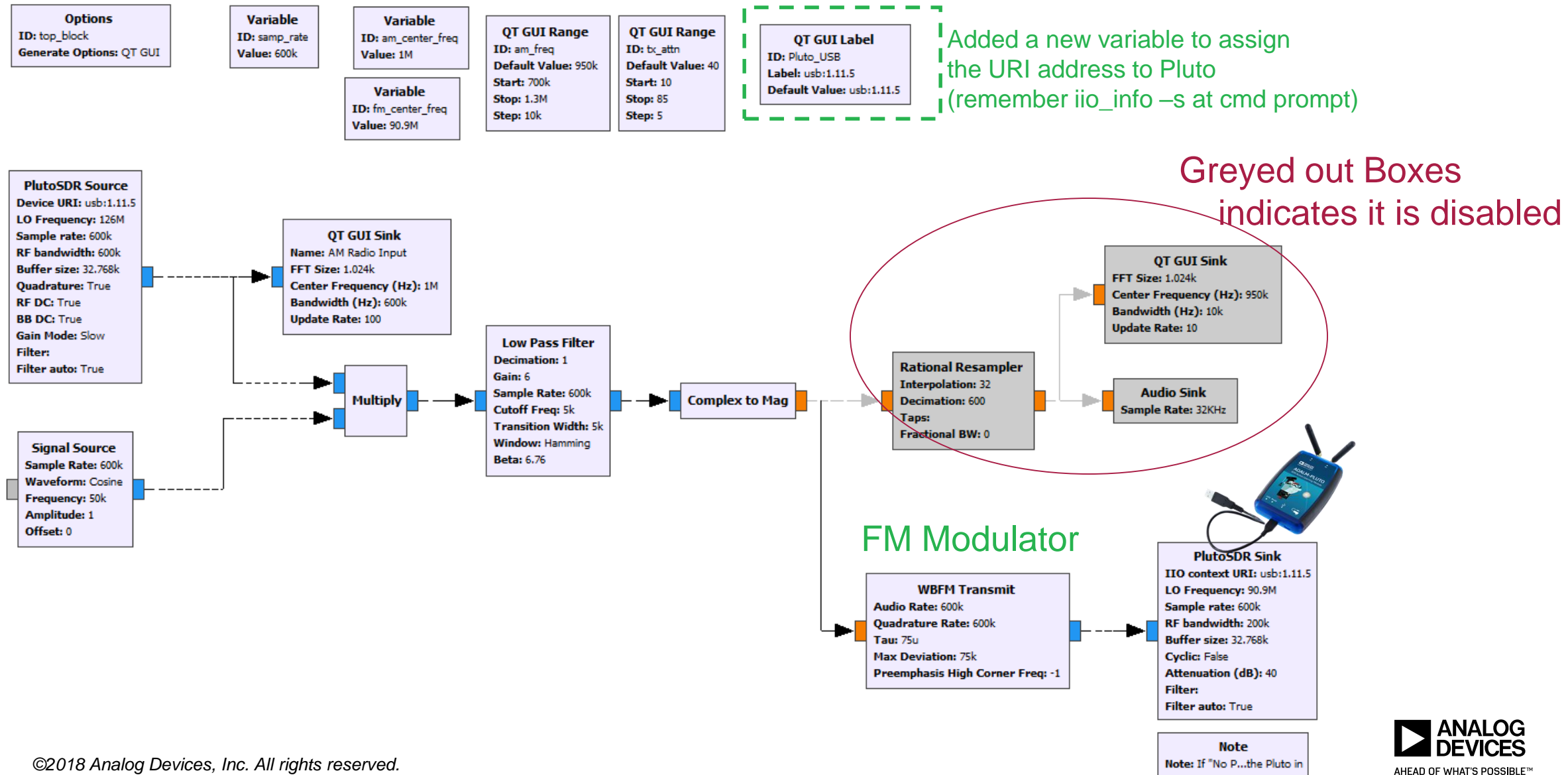
Demodulated Station Output



AM Radio Stations from Antenna



Lab1: GNU Radio Flow Chart



Lab 1: The End

The GNU Radio File is available at:

<https://github.com/jonkraft>

Lab 2: Hack an RF Outlet



Pluto Lab 2: Hack an RF Outlet

Order of attack for hacking an unknown RF signal:

1. Obtain the RF Spectrum

- We can often get the Tx freq from the transmitter. Or the FCC Website.
- Our transmitter is listed as 315 MHz. We center our LO around here.
- Then we save the transmit spectrum to a file for further analysis

2. Decode the RF information

- We want to find modulation type (i.e. OOK, PWM, FSK, ASK, etc.)
- We want to identify the preamble, encoding scheme, and bit function
- We want to get bit rate

3. Filter and Transmit

- After understanding the signal, we can extract just what we need
- Filter so that we are not polluting the spectrum!
- Then retransmit (being very careful to limit energy and obey the rules!!)

FCC ID PAGTR-009N

FCC ID PAGTR-009N External-Photos, PAG TR009N, PAGTR-009N, PAGTR-009N
KAB Enterprise Co., Ltd. Remote Controller TR-009N

FCC ID > / KAB Enterprise Co., Ltd. > / TR-009N



*Please Check Individual Documents for more images

Application: Remote Controller

Equipment Class: DSC - Part 15 Security/Remote Control Transmitter

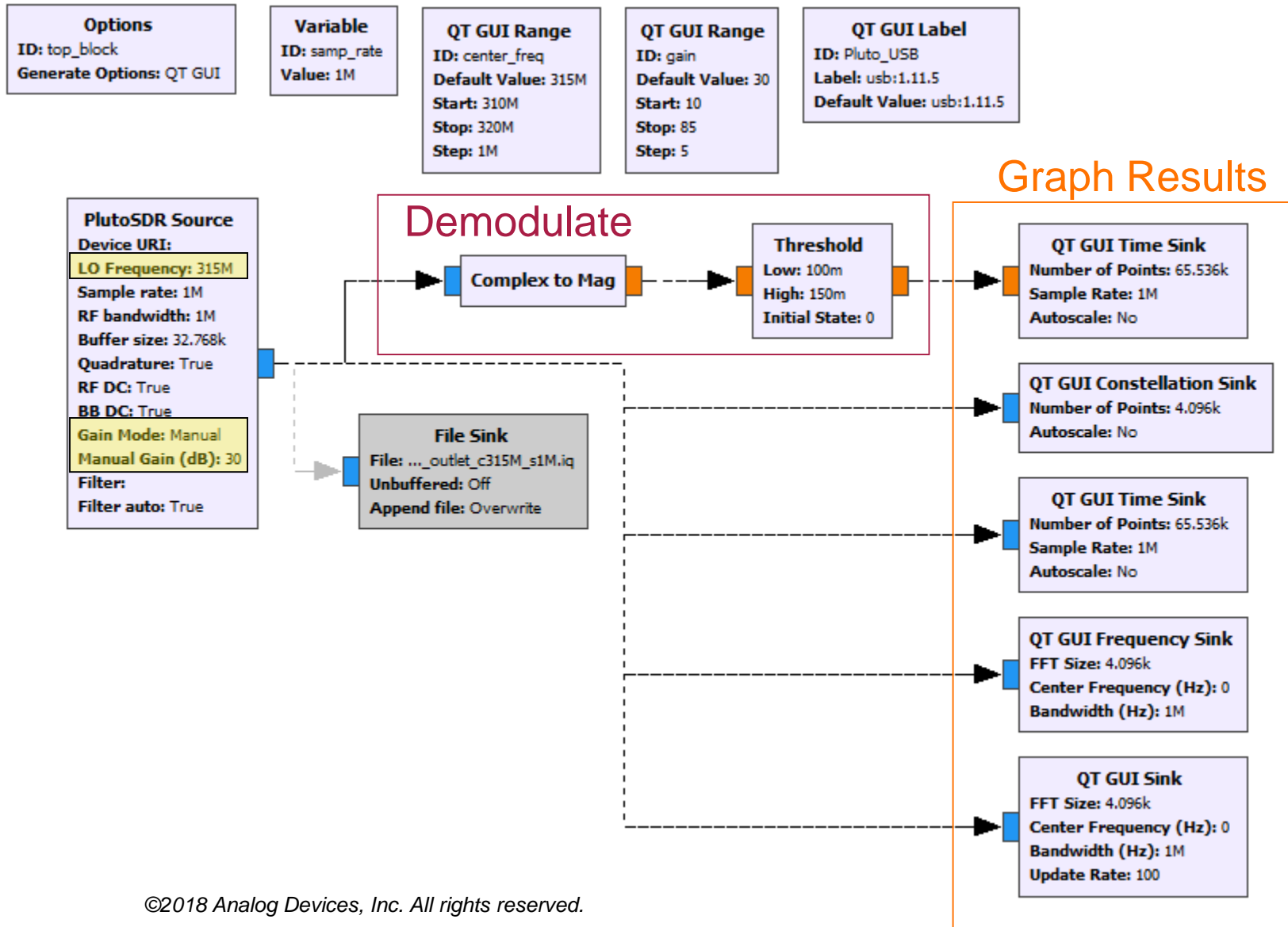
Alternate Sources: FCC.gov | FCC report

Registered By: KAB Enterprise Co., Ltd. - PAG (Taiwan)
you@youremail.com [Subscribe](#)

App #	Purpose	Date	Unique ID
1	Original Equipment	2018-02-29	KdM/22Tj/+w2Ccd4uYcokler==

Operating Frequencies		
Frequency Range	Rule Parts	Line Entry
315-315 MHz	15.231	1

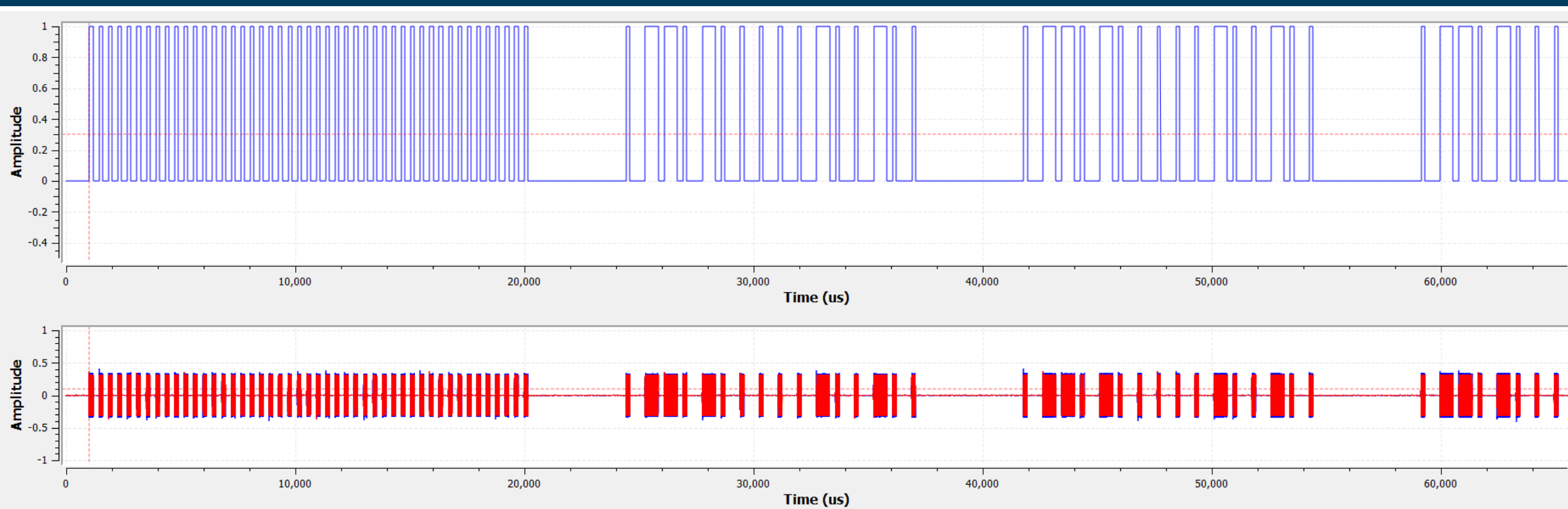
Pluto Lab 2: Hack an RF Outlet



Receive GNU Radio Flow:

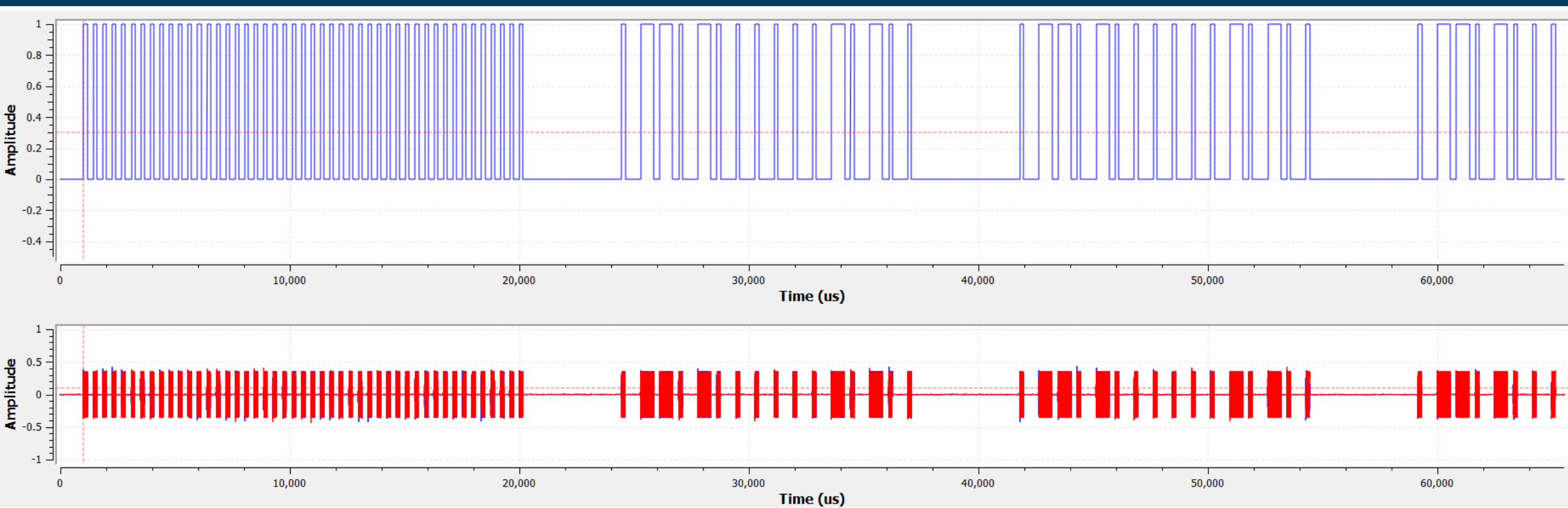
1. Change LO freq to match Tx
 - a. Or offset it a little, but still within BW
2. Set Gain Mode to Manual
 - a. Gives flexibility to properly scale intermittent signals
3. Graph Results
 - a. Each graphs help identify unique attributes of the transmitted signal
 - b. Spectrum analyzer
 - c. Time based oscilloscope
 - d. Waterfall Plot
 - e. Constellation
4. Try demodulation schemes to see what fits.
 - a. If you suspect OOK/ASK/PWM, then convert magnitude.
 - b. For FSK and PSK use those demod blocks

Pluto Lab 2: Hack an RF Outlet



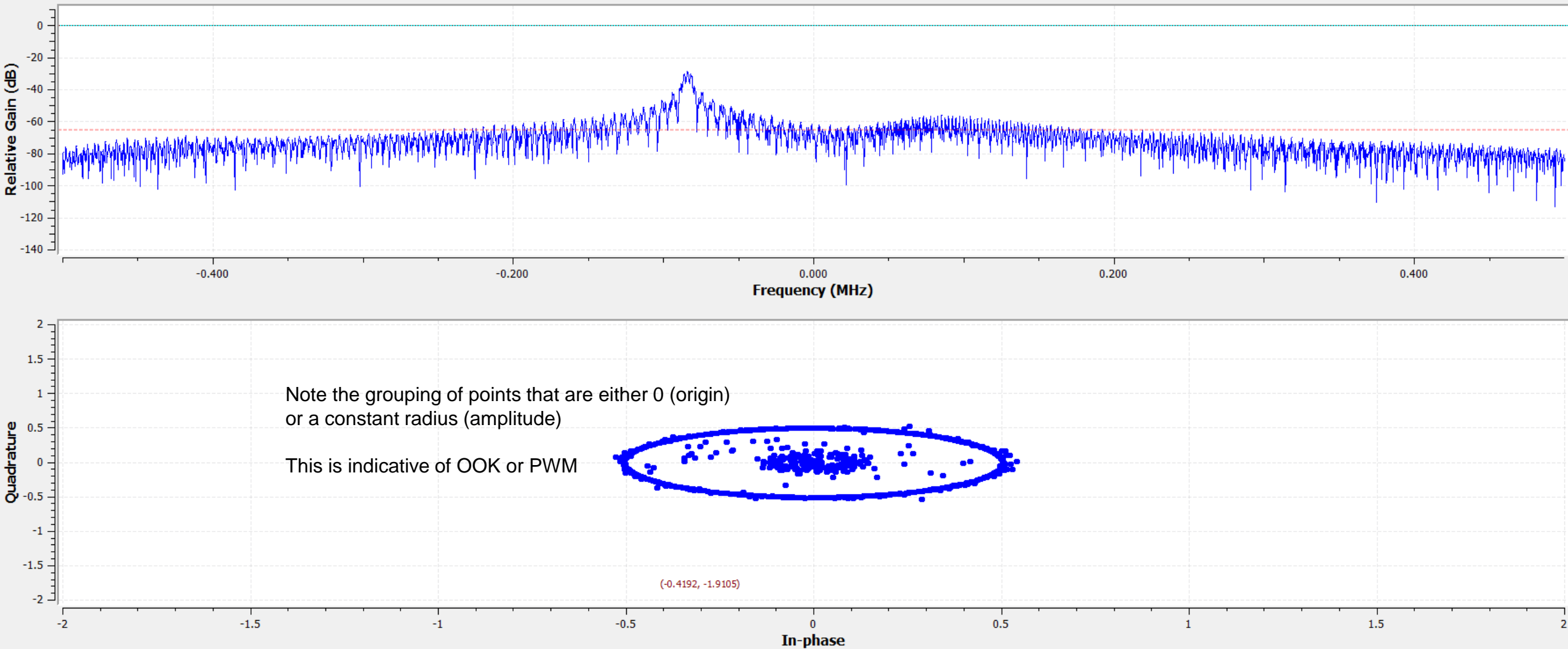
ON

Pluto Lab 2: Hack an RF Outlet

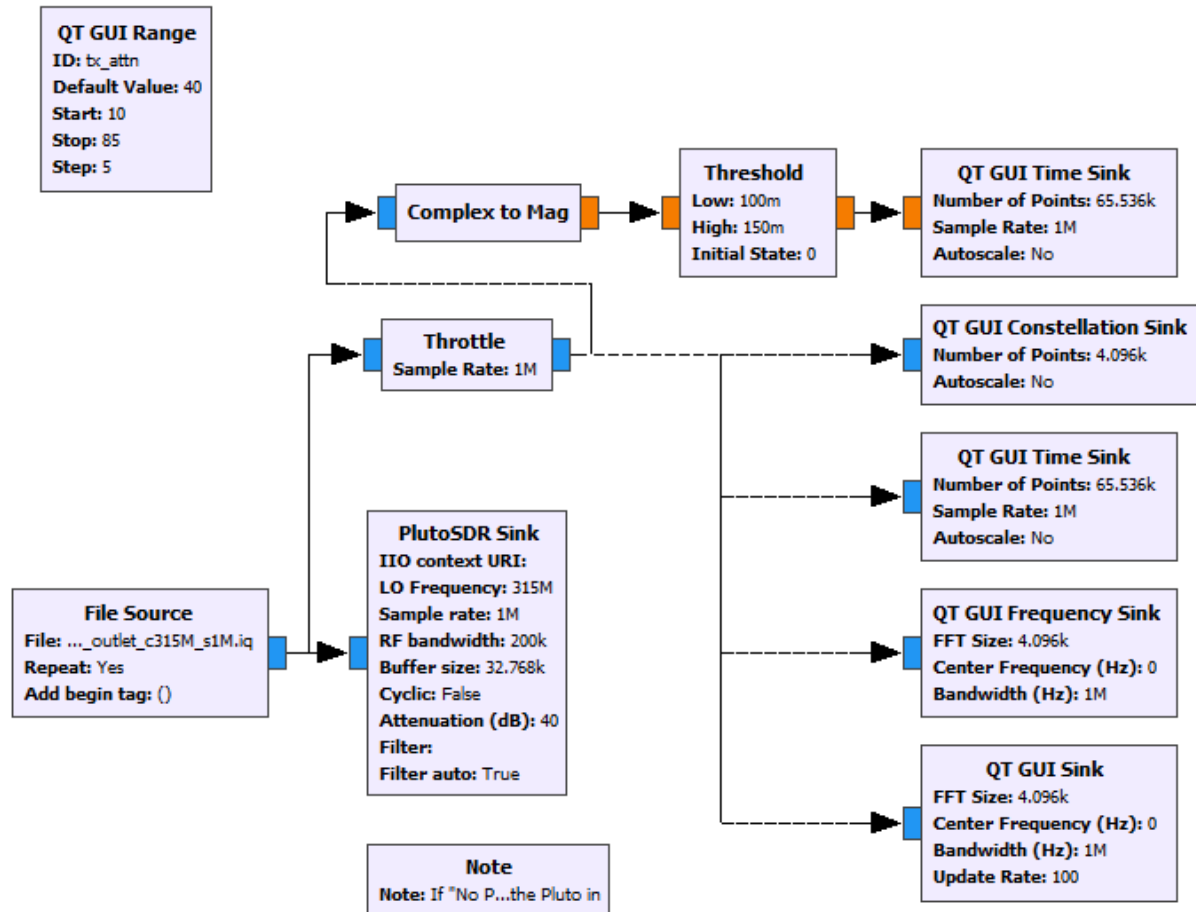


OFF

Pluto Lab 2: Hack an RF Outlet



Pluto Lab 2: Hack an RF Outlet



Transmit GNU Radio Flow:

1. Do not transmit anything if you do not have the legal permission to do so!!!!
2. Set the Tx attenuation to maximum
 - a. Then reduce to only the value required to reach the target
3. We simply “replay” the spectrum we capture
 - a. Filter the transmission (either digitally or with Pluto) so that you are not polluting other frequencies
 - b. Be sure to use the same center freq and sample rate
 - c. The GUI sinks (for plotting the signal) are not necessary. They are just to inform about signal source we are transmitting.

Lab 2: The End

The GNU Radio File is available at:

<https://github.com/jonkraft>