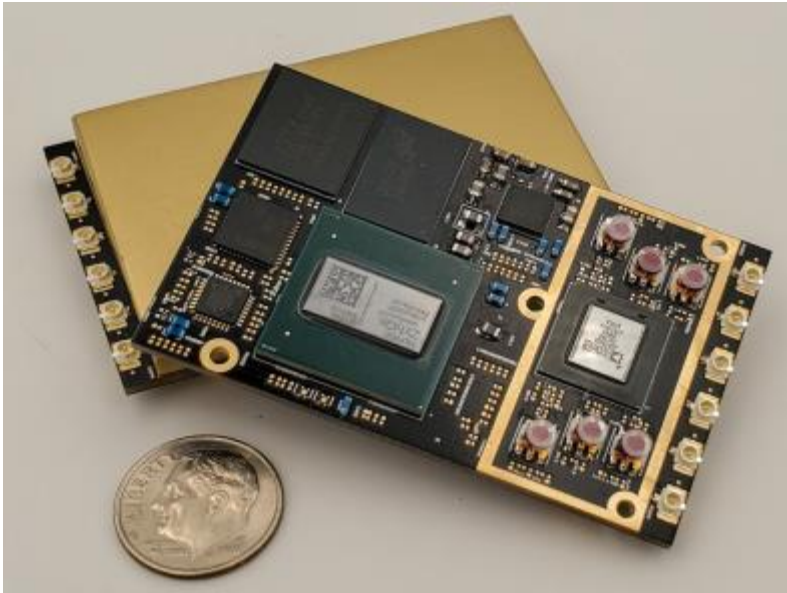
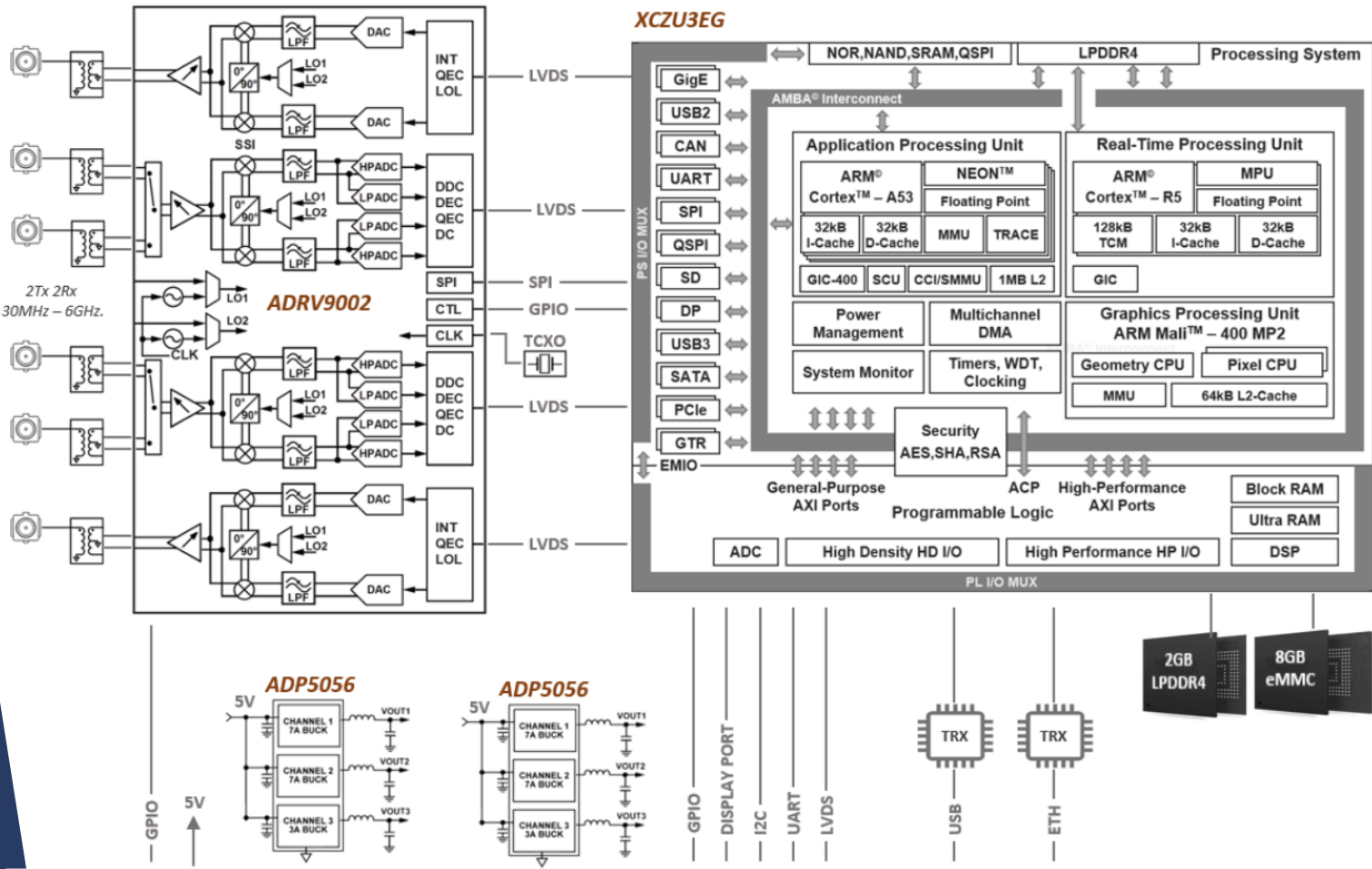


BytePipe Toolbox for MATLAB and Simulink

BytePipe: 0.03-6 GHz Transceiver with FPGA SoM (ADI ADRV9002 / Xilinx XCU3EG/ XCZU2CG)

HARDWARE BLOCK DIAGRAM



Design
Environment,
Libraries

External Interface / Breakout Board

BytePipe SOM

ADRV9002



Xilinx
XCU3EG/
XCZU2CG

LNA
microSD Card
Display Port
SATA Connector
Gigabit Ethernet
USB 3.0
Serial Port
JTAG Connector

Navassa Family



ADRV9002 – Full Featured Device

- **2T 2R** includes DPD



ADRV9003 – Single Channel Device

- **1T 2R** or **1T 1R** Configurations
- Part does not include DPD

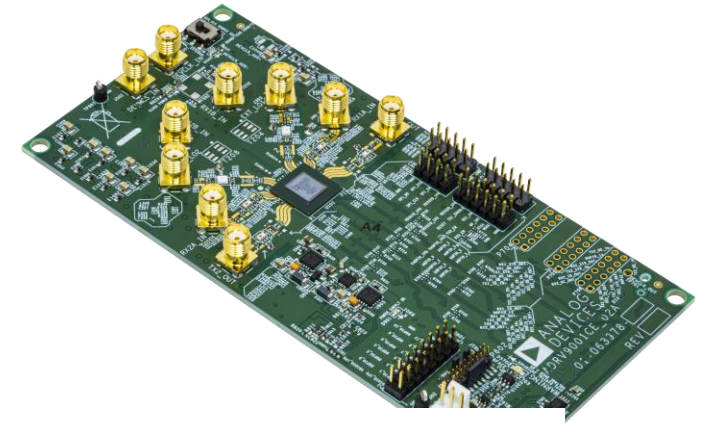


ADRV9004 – Dual Channel Device

- **2T 2R** Configurable
- Part does not include DPD

► ADRV9001 Family

- All parts supported by **one TES SW, EVAL board and User Guide** found on Analog.com product pages
 - analog.com/en/products/adrv9003.html
- All parts support **30MHz to 6GHz** operation and **12KHz to 40MHz** radio bandwidths
- All parts offer the same RF performance specifications
- All parts in the same package **12 x 12 and pin configuration**
- Any additional future devices will support the same Frequency/Bandwidths and RF performance but may differ in system features

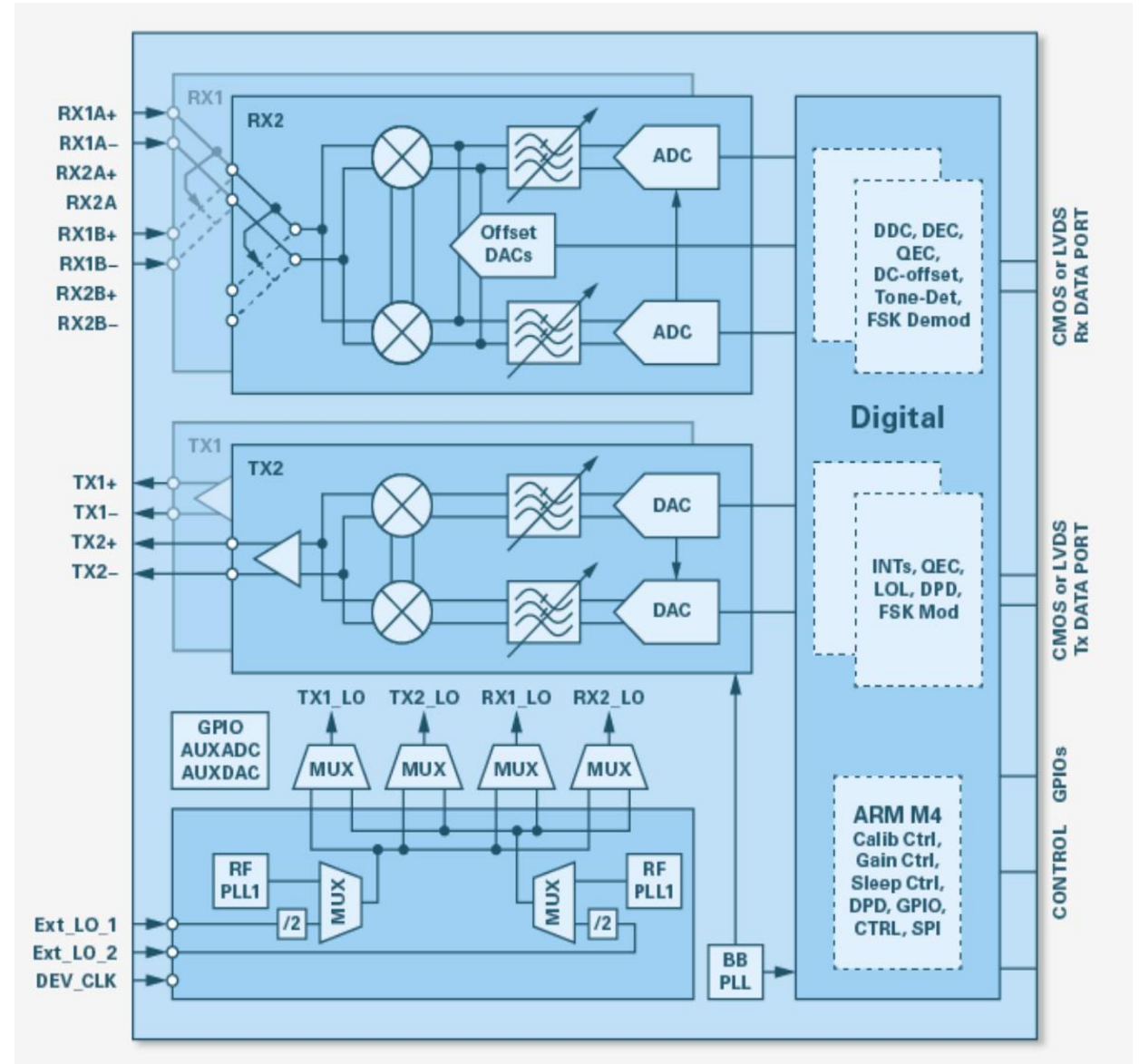


Part Numbers for Eval Boards

- **ADRV9002NP/W1/PCBZ** (30MHz to 3000MHz)
- **ADRV9002NP/W2/PCBZ** (3GHz to 6GHz)

Introduction to the ADRV9002/3/4 (AKA Navassa)

- **Highly Integrated Transceiver (RF to Digital)**
 - **Dual channel Transceiver (2Tx 2Rx shown ADRV9002)**
 - Dual flexible fully integrated RF LOs on chip
 - Direct conversion, IF Architecture, Tx direct modulation modes supported
- **Operating Frequency Bands**
 - 30MHz to 6GHz Operating frequency bandwidth
 - 12.5KHz to 40MHz Programmable radio signal bandwidths
- **Digital Functions**
 - Integrated digital downconverter (DDC) for NB IF operation
 - 128 taps Rx/Tx programmable FIR filters
 - Automatic and manual gain control
 - Digital Pre-Distortion (DPD)
 - Correction algorithms for RX QEC, Tx QEC/LOL, DC offset
- **Interface Modes**
 - Flexible multi-protocol data interface, LVDS / CMOS SSI digital interface
 - SPI, AuxDAC, AuxADC, GPIO, Tx/Rx enable control
- **System functions**
 - Auxiliary monitor functions
 - Fast profile switching
 - Frequency hopping
 - PA ramp control and PA protection
 - Monitor mode (autonomous RSSI monitoring)
 - Flexible power vs. performance modes



ADRV9002/3/4 RF System Value / Features

*High performance RF Agile SDR Transceivers provide user configurability
More than Silicon and Beyond Silicon*

- ❖ First broad market transceiver design for **mission critical communications NB / WB operation**
- ❖ **Best in class RF performance**
 - ❖ Highest Receiver dynamic range 150dBc/Hz
 - ❖ Highest Transmitter offers highest output linearity
- ❖ Offers exceptional dynamic range to absorb the signal and blocker **minimizes the need for analog IF and base band filtering**
- ❖ Latest technology for **optimum power Vs performance** with integrated radio signal processing
- ❖ Unique flexible solution agile, cognitive, **fully programable SDR**
- ❖ High level of Integration, **System Features, Digital Features, Calibrations**
- ❖ Advance system features **Monitor mode, MCS, DPS, FH, PA DPD**
- ❖ **Small packaged 12 x 12 mm BGA package**

IC / RF Performance

- Highest Linearity
- Highest TX output Power
- Lowest Integrated Phase Noise
- Best RF Sensitivity
- Flexible Digital Interface
- Integrated Digital Channel Filters
- Lowest NF / Noise Floor

RF System Calibrations

- Tx QEC Tracking
- Tx LOL leakage
- Tx Control Loops
- Rx DC Offset
- Rx QEC
- Rx Distortion correction

System Features

- Power Saving & Monitor Mode
- Fast Freq Hopping (FFH)
- Manual and Automatic Gain Control (AGC & MGC)
- Programmable FIR Filters (PFIR)
- Dynamic Profile Switching (DPS)
- Multi Chip Synchronization (MCS)
- Digital Pre-Distortion (DPD)



RF Agile Software-Defined Radio (SDR) Transceivers

Narrow-band to Wideband, High Dynamic Range, Low Power

▶ Narrow band to Wideband Integrated RF Agile Transceiver

- Freq of operation 70MHz to 6000MHz
- Independent Wideband / Narrow band 12KHz to 40MHz RX and TX Channels

▶ Direct conversion, offset IF, direct modulation operation

- High dynamic range and integrated digital baseband filtering
- Removes the need for RF / IF filtering with advance sigma-delta, $\Sigma\Delta$ Converters

▶ Power Vs Performance, High dynamic TRX for battery powered applications

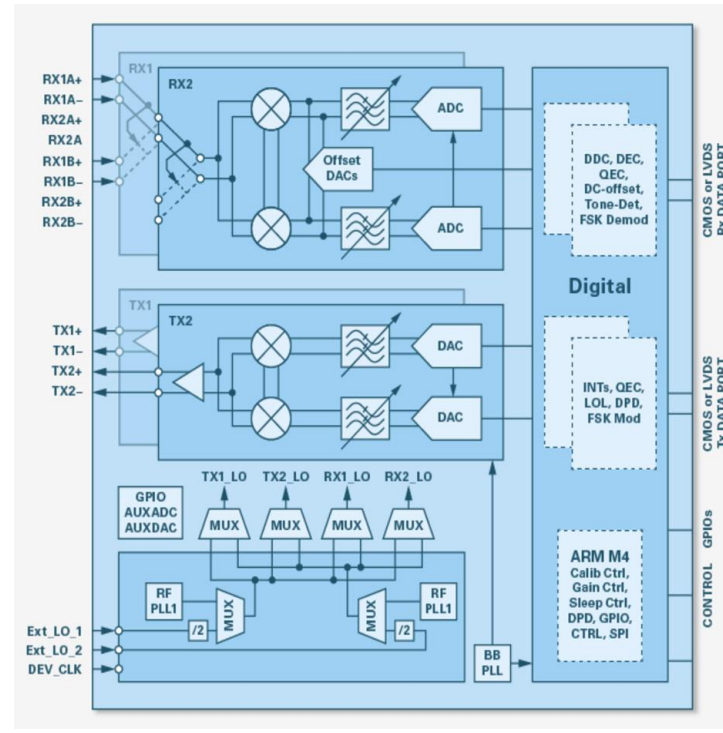
- Each RX and TX chain can be switched power up /down as needed
- Advance system power savings with Monitor Mode

▶ Agility of operation with integrated independent PLL/VCO circuits

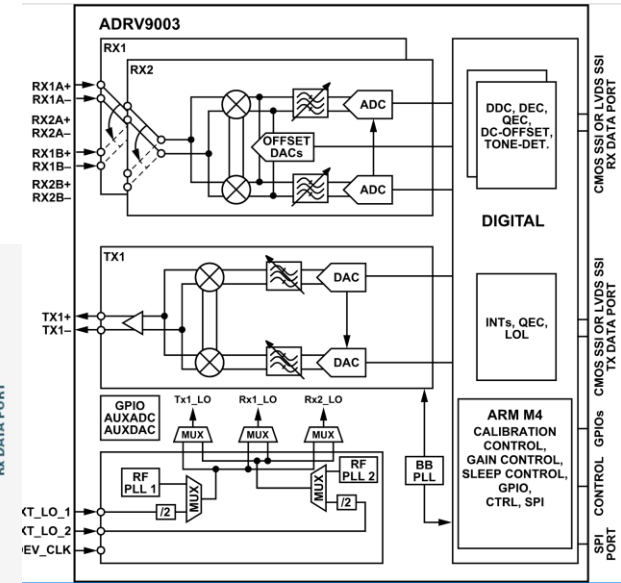
- FDD or TDD operation
- Operation with external VCO / LO
- Flexibility routed to any RX or TX

▶ Integrated digital signal processing / filtering to baseband

- filtering for each channel
- FSK modulation and demodulation
- Correction algorithms



ADRV9002 2Tx 2Rx

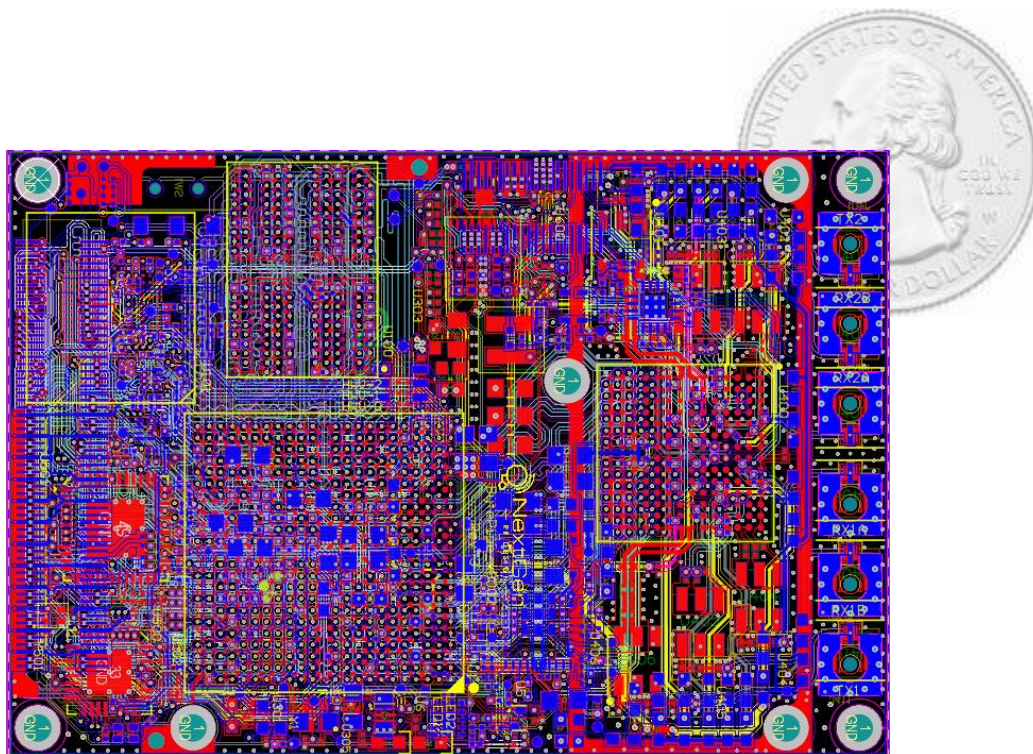


ADRV9003 1Tx 2Rx

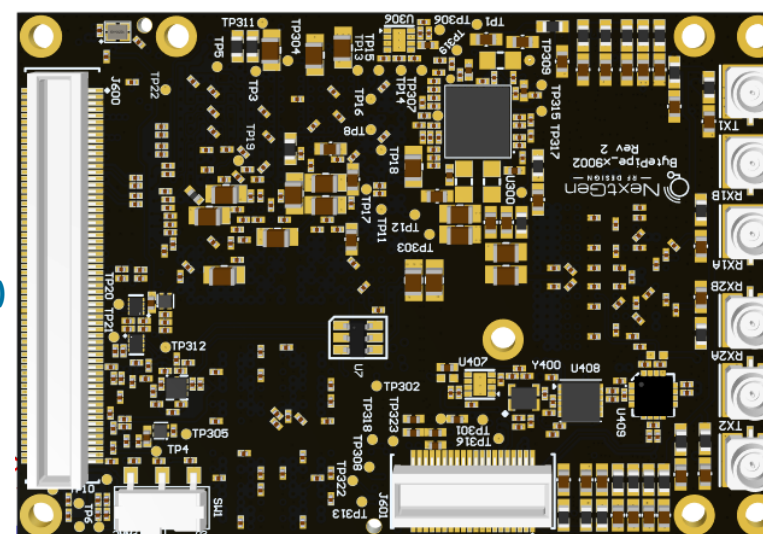
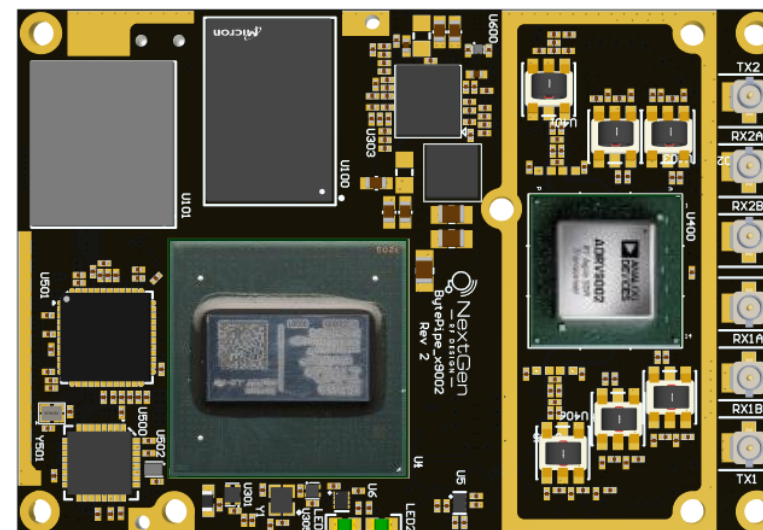


Hardware Design

Facilitating lower cost and shorter time to market.



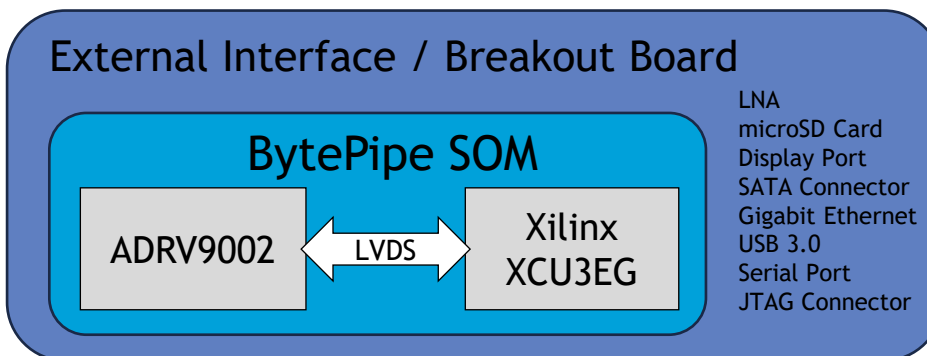
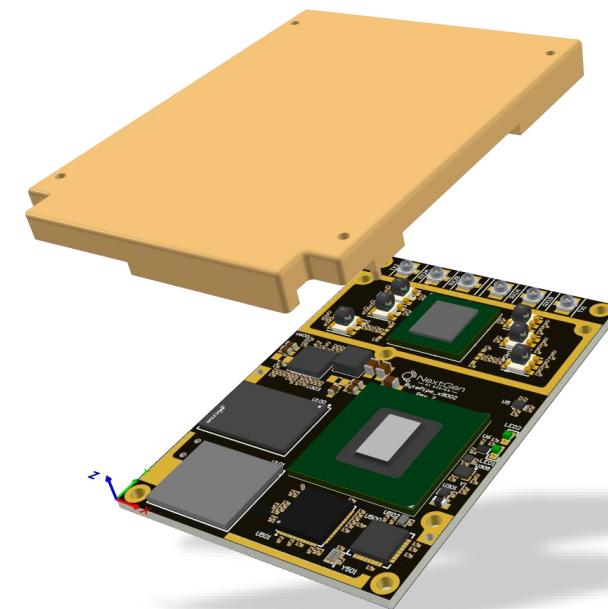
1.66"
(42mm)



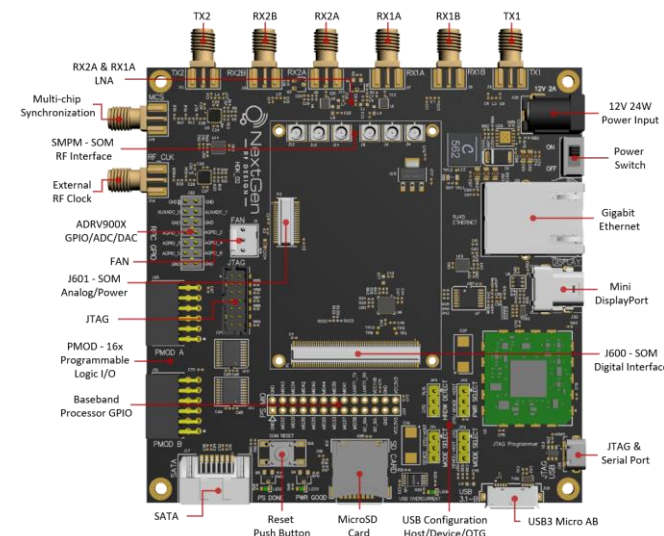
2.41"
(61mm)

BytePipe: 0.03-6 GHz Transceiver with FPGA SoM (ADI ADRV9002 / Xilinx XCU3EG/ XCZU2CG)

- Quad Core Cortex A53 Application Processor
- Dual Core Cortex R5 Realtime Processor
- Optional ARM Mali-400 Graphics Processor
- Programmable Logic with up to 360 DSP slices and 154,000 logic cells
- 8GB eMMC / 2GB LPDDR4
- Synchronizable
- 36.8mm x 61mm Dimensions
- Production ready: design time cut by 6 months
- Market specific RFFE's available
- Support for Third Party Tools
 - MATLAB, Simulink, Gnu Radio
- Breakout board available with
 - LNA for RX, 4x GTR, USB3, Gigabit Ethernet, PCIe, DisplayPort, SGMII, GPIO, SPI, UART, I2C, CAN, SD
- NextGen RF Customization supported



Design
Environment,
Libraries

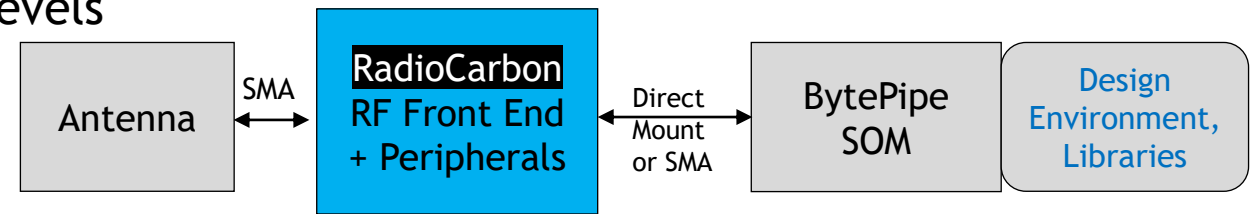
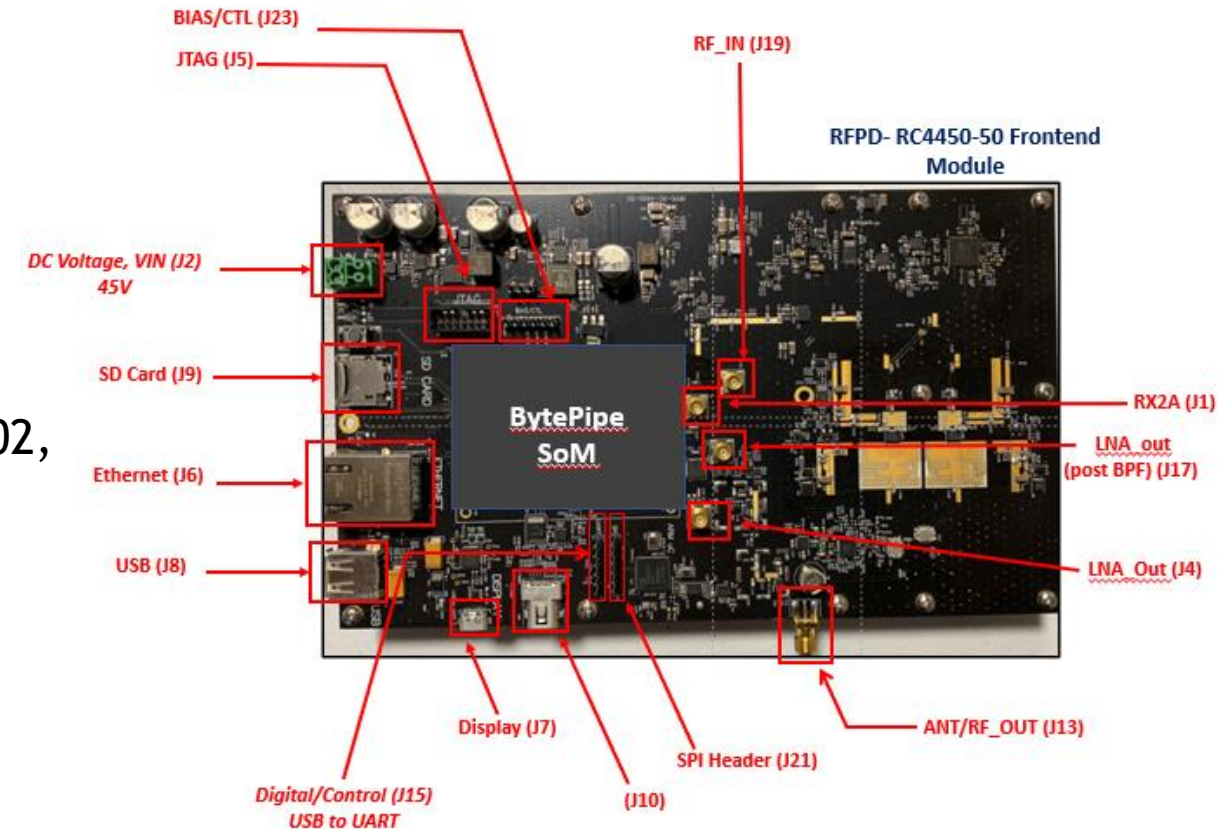


Software Defined Radio with RF Front End

Radiocarbon Design accelerator platform with design services

Features

- Single Channel Transmitter and Receiver
- Prototyping Platform for BytePipe SoM
 - Alternative to Eval kit
 - BytePipe mounts directly to carrier board
- Standalone software/GUI
 - Use with any ADI Transceiver with DPD (ADRV9002, AD9375, etc)
 - RF Board connections (SMA, MCX)
- Three integrated RF Front End (RFFE) options
 - DC power generation and PA bias
 - Small Footprint
 - Reference design for Military comms, Satellite, Troposcatter, NLOS backhaul, 5G
 - Modifiable to support other bands and power levels
 - BOM, schematic, and layout available



RFPD-RC-1327-50: 1.35-2.7 GHz, 2W Linear Front-End

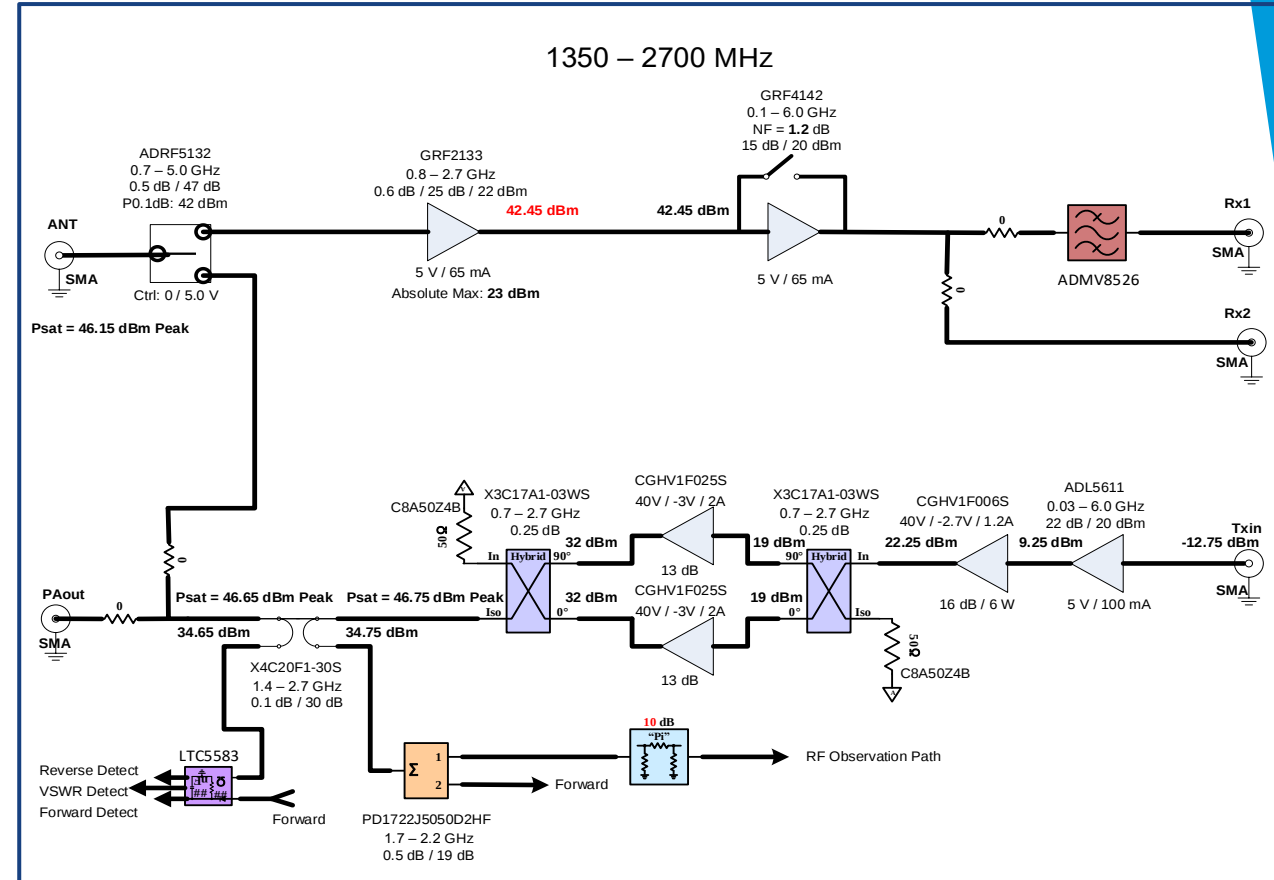
Radiocarbon Design accelerator platform with design services



Release 2H22

Technical overview

- TX
 - 33 dBm avg at antenna, 44.5 dBm PSAT
 - -50 dBc ACPR (9 dB PAR, BytePipe DPD)
 - 42 dB Gain
 - TX 27% PAE (PSAT)
- RX
 - 1.9 dB NF
 - 23 / 6 dB Gain
- 42-60V Vin



Paper Design

- 42-60V V_{in}





Agenda

- NextGen RF Design
- BytePipe Overview
- Transceiver Evaluation Software
- RadioCarbon DPD Demo
- BytePipe Toolbox for MATLAB
- HDL Coder Example

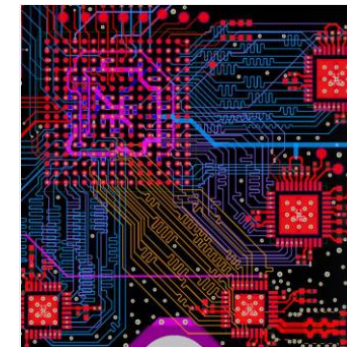
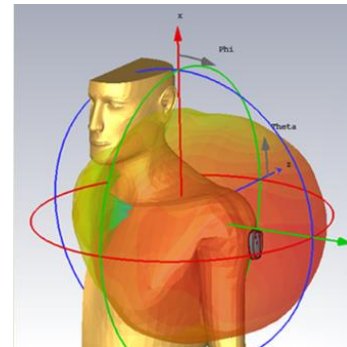
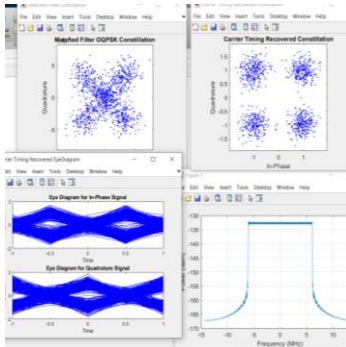
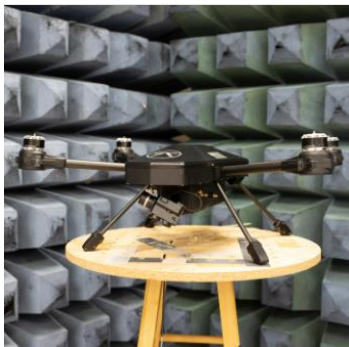
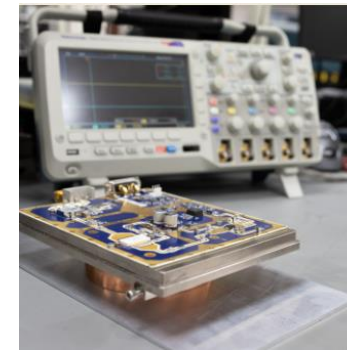
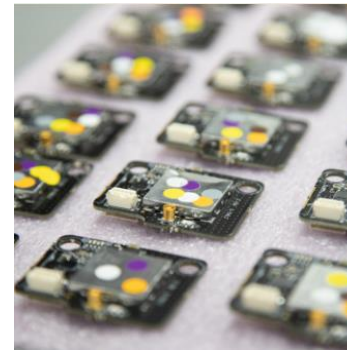
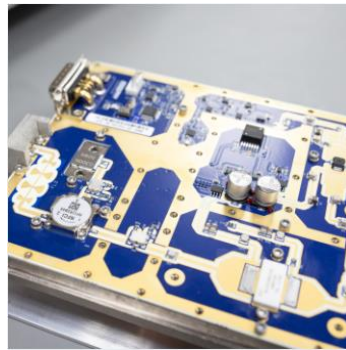


We know design.

Proven expertise for tough
challenges from start to finish.



About NextGen RF Design



RF/WIRELESS

- Discrete Design, Chip Down or Module/Modem
- High Power Amplifiers & Linearization
- Narrowband, High Performance or Wideband, High Data Rate Designs
- RF Range Optimization

EMBEDDED DESIGN

- Software Defined Radio
- Embedded Control Systems Using Microcontrollers/ Microprocessors, FPGAs, SoCs
- RTOS & Baseband Stack Development
- Custom Modulation Schemes

ANTENNAS

- Design Simulation (CST)
- Custom PCB Antenna Designs – Monopole, Dipole, Patch, Inverted F
- MIMO & Diversity System Design
- 3-Axis Pattern Measurement
- Matching and Optimization
- Pattern Steering

DIGITAL & ANALOG

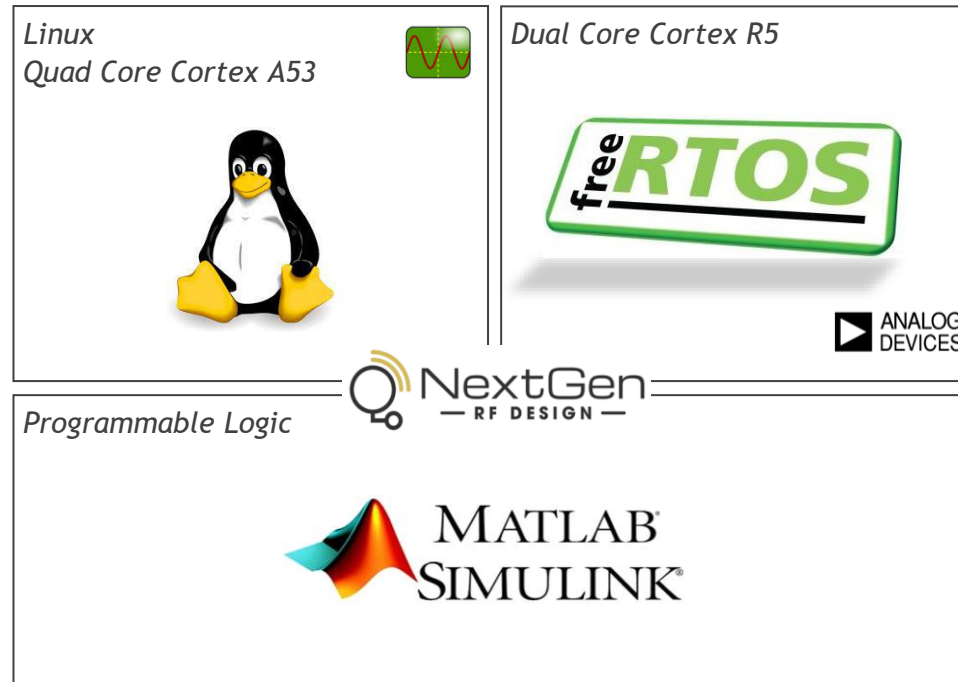
- Communications Interfaces – SPI, I2C, UART, Ethernet, USB and Parallel Address/Data Busses
- Power Supply Designs and Charging Systems
- ADC/DAC
- High Speed Memory
- Voltage & Waveform Generation
- Motor Control Circuit Design

PCB DESIGN

- Specialists for RF Design Layouts, RF Grounding and Shielding
- High Speed Digital, High Power, EMI/EMC Expertise
- 3D Modeling of PCBs and Components
- Altium, Mentor Graphics, OrCAD, Solidworks

TEST FIXTURES & ATS

- Specialized in RF/Wireless Functional Test Fixtures
- Single-Up & Multi-Up Designs
- Custom Bed-of-Nails and RF Interfaces
- Automated Test Systems
- LabVIEW Certified Programmers



Xilinx SoC Baseband Processor





Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

[NextGenRF-Design-Inc](#) / [bytepipe_sdk](#) Public

[Code](#) [Issues 8](#) [Pull requests](#) [Discussions](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[main](#) 1 branch 9 tags

[Go to file](#)

[Add file](#)

[Code](#)

[About](#)



BytePipe Software Development Kit

[Readme](#)

[GPL-3.0 License](#)

0 stars

3 watching

3 forks

[Releases 9](#)

[v02.16.22](#) Latest
on Feb 26

[+ 8 releases](#)

[Packages](#)

No packages published
[Publish your first package](#)

[Contributors 3](#)

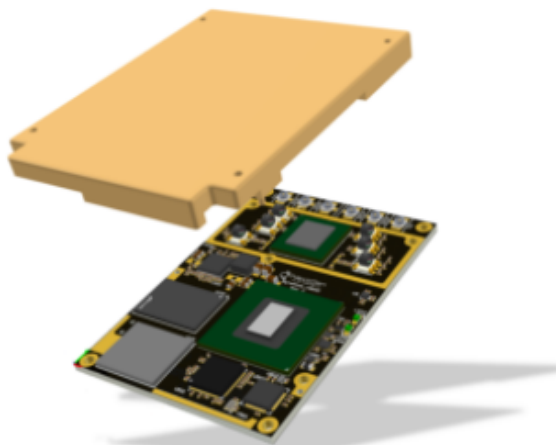
enelson83

g-anderson

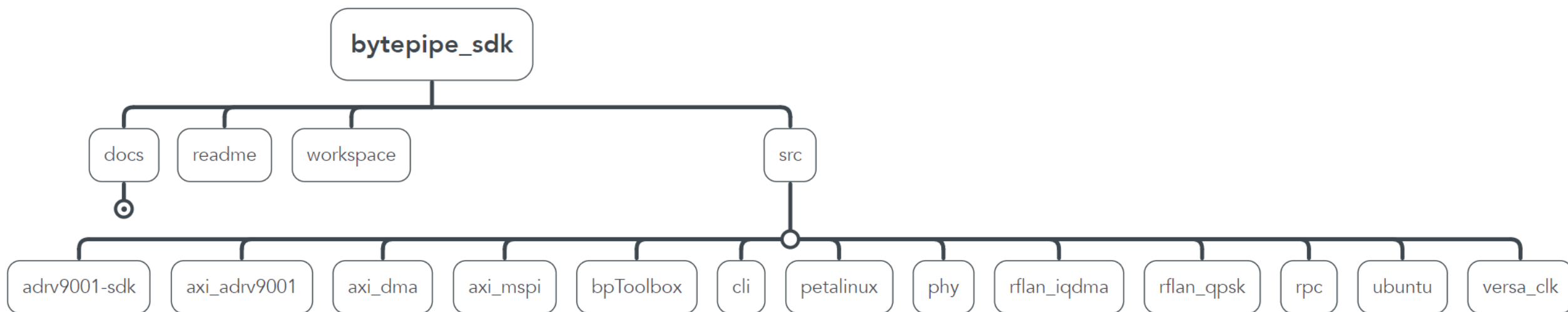
CharlesViel Charles Viel

	enelson83 updating rflan dynamic profile support to include single json file an...	d0ac085 on Mar 4	148 commits
	docs	adding docs	9 days ago
	src	latest	1 minute ago
	.gitignore	latest	1 minute ago
	DISCLAIMER.md	starting over	8 days ago
	LICENSE	starting over	8 days ago
	README.md	starting over	8 days ago

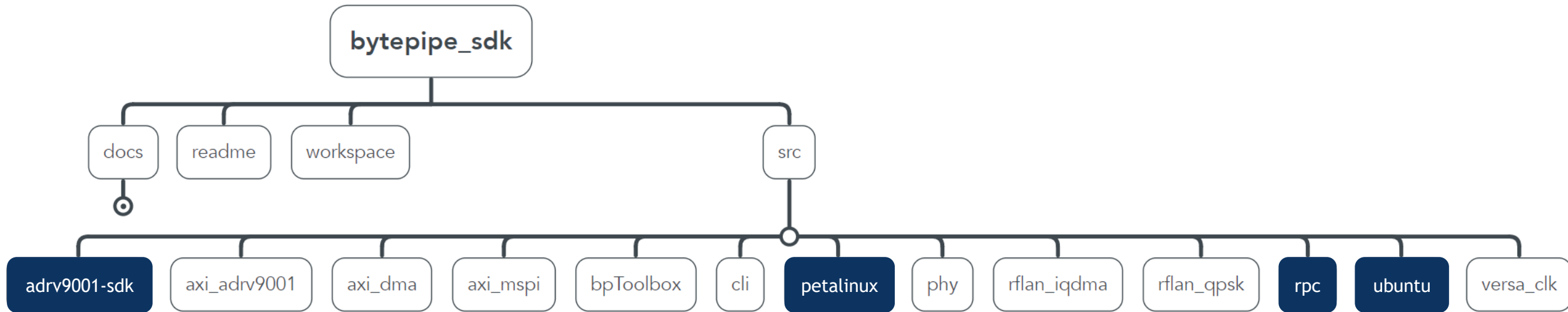
README.md



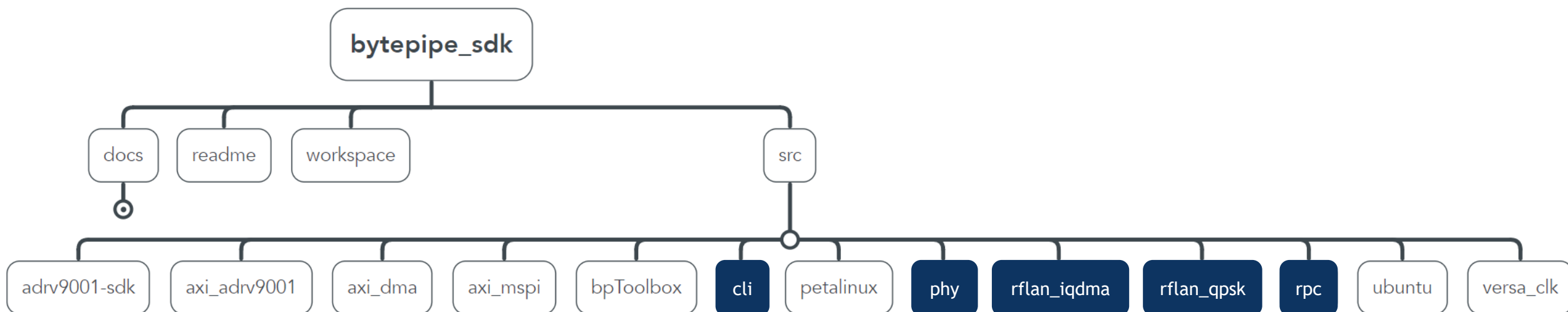
BytePipe SDK



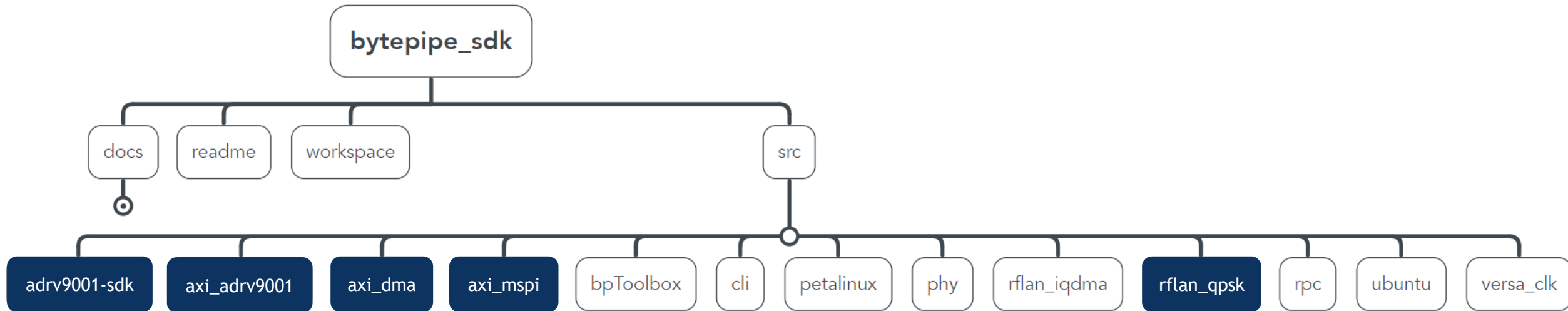
Application Processing Unit



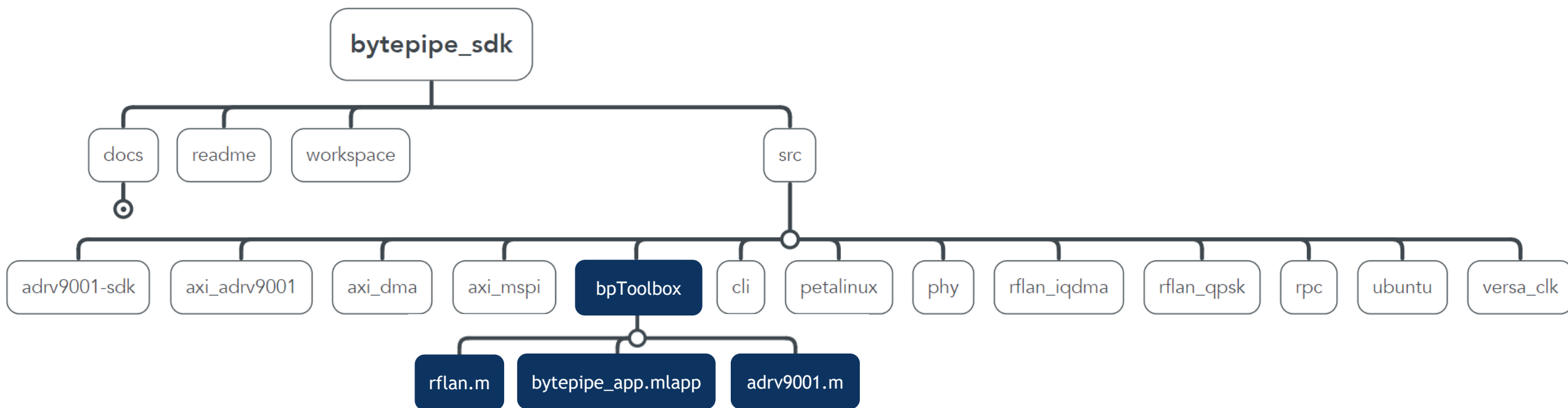
Real-time Processing Unit



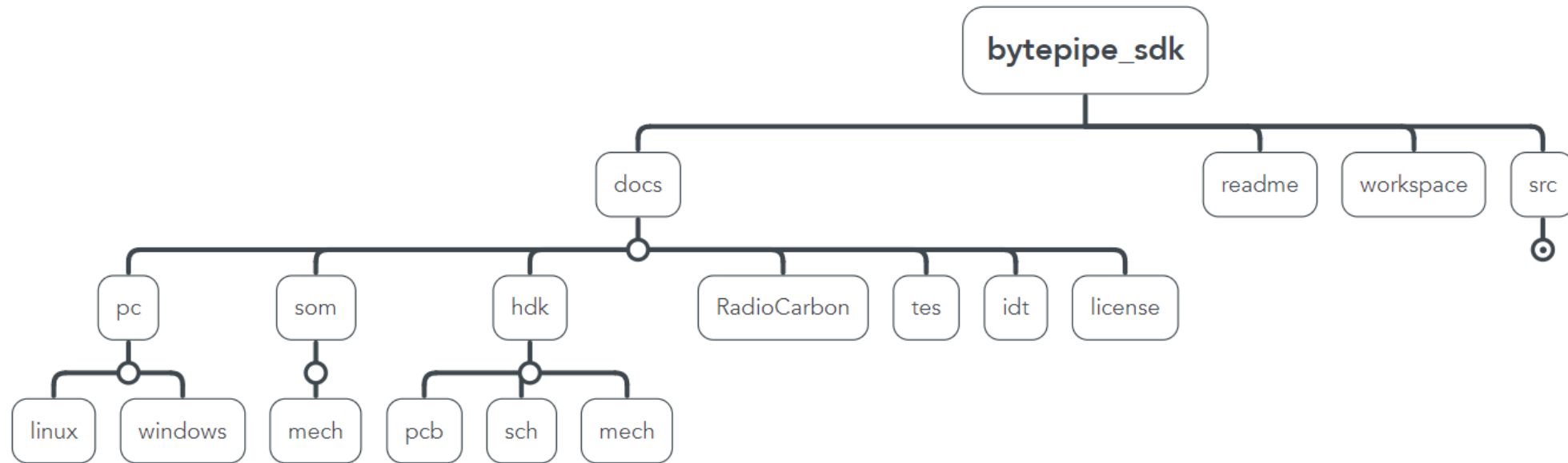
Programmable Logic



BytePipe Toolbox

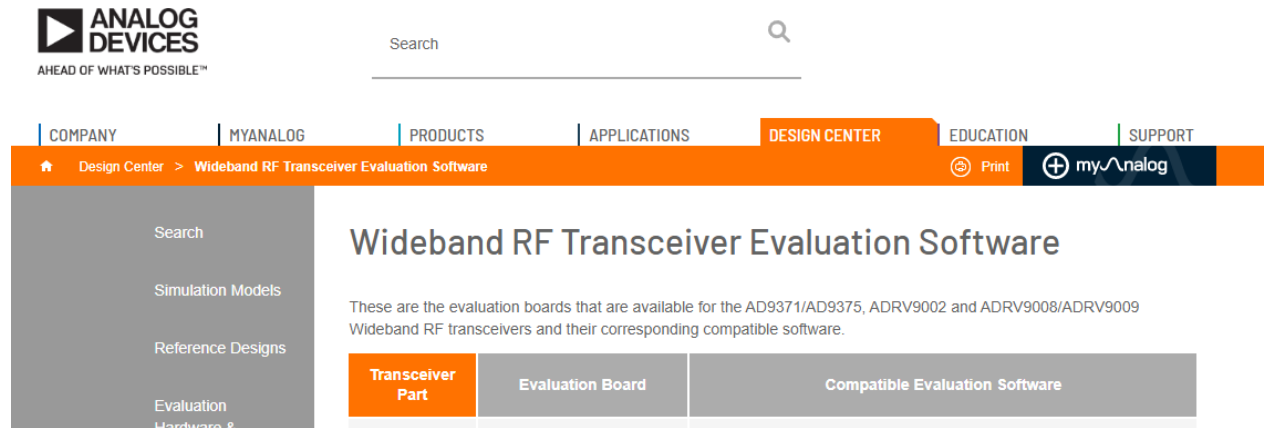


Documentation

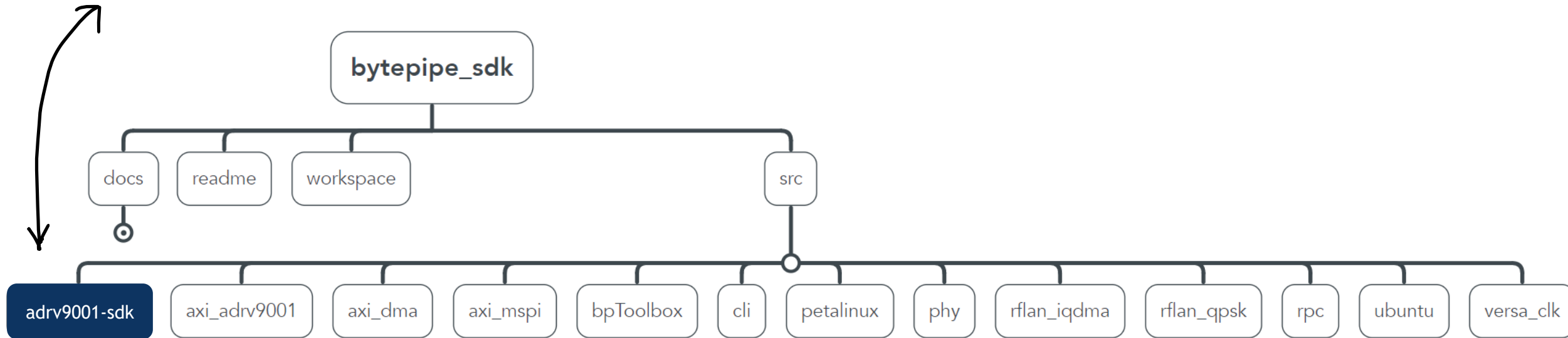


BytePipe Transceiver Evaluation Software

Analog Devices RF Transceiver Evaluation Software



<https://www.analog.com/media/en/evaluation-boards-kits/evaluation-software/adrv9001-sdk-eval-software.zip>



Live Demo - Building Software

The screenshot displays a Linux desktop environment with a terminal window, a file manager window, and a file selection dialog.

Terminal Window: The terminal shows the execution of a `make` command in the directory `~/adv9001-sdk20/platform/projects/bytepipe/sw`. The command is `make -C projects/bytepipe/sw sd_card`. The output shows the directory being entered and the execution of various `cp` commands to copy files from the `adv9001-sdk20` directory to the `media/ngrf/BOOT/` directory. The files being copied include `BOOT.bin`, `adv9001_zcu102_cmos.bin`, `adv9001_zcu102_lvds.bin`, `axirw.pl`, `axirw.regs`, `boot.bif`, `BOOT.bin`, `system.dtb`, `adv9001_zcu102_cmos.bin`, `adv9001_zcu102_lvds.8.2.21.bin`, and `adv9001_zcu102.h`. The terminal also shows the execution of `sudo cp` commands to copy files to the `media/ngrf/rootfs/home/analog/adv9001_server/resources/Adi.Adrv9001.FpgaBinaries/` directory. The terminal ends with the message `make: Leaving directory '/home/ngrf/adv9001-sdk20/platform/projects/bytepipe/sw'`.

File Manager Window: The file manager shows the directory `adv9001-sdk20` with the following files and folders: `xilinx-uboot-v2018.2`, `adv9001_zcu102.h`, `adv9001_zcu102_boot.tar.gz`, `adv9001_zcu102_cmos.bin`, `adv9001_zcu102_lvds.bin`, `atf.elf`, `atf.log`, `axirw.pl`, `axirw.regs`, `boot.bif`, `BOOT.bin`, `devicetree.log`, `Image`, `Image.log`, `linux.mk`, and `Makefile`.

File Selection Dialog: The dialog shows the file `xilinx_zynqmp_zcu102_rev1_0_defconfig` selected (518 bytes).

Live Demo - Using TES

Argo Navis - 0.20.0

File View Sample Code Disconnect Program Log File Help

Configure Transmit Receive Observe GPIO Gain Control Tx Front End Power Savings and Monitor Mode TDD Enablement Delays Automated TDD Tracking Cals Digital Pre-Distortion Auxiliary

Connection

Device Configuration

Board Configuration

Clocks

Carriers

Radio

Advanced Features

Initial Calibrations

Rx Filters

Tx Filters

Rx Overview

Tx Overview

Device connected.

TCP IP Connection

TCP IP Address 192 . 168 . 1 . 10

Port Number 55557

Switch to Hardware Connection

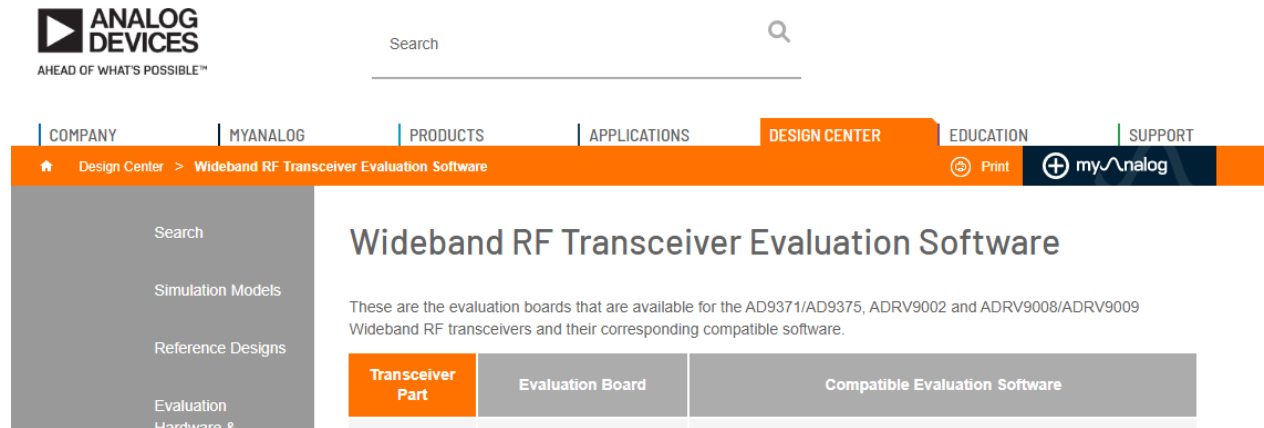
System Info

Transceiver Name	
Silicon Revision	Presumed C0
Evaluation Board Revision	
Tx / Rx optimal carrier frequencies	30 MHz to 6 GHz
External LO optimal frequencies	60 MHz to 12 GHz
FPGA	v0.0.0
Device Driver API	v0.0.0
Device Driver Client	v48.49.2
Firmware	v0.20.0.10
Profile Generator	v0.50.5.0
Stream Generator Assembly	v0.7.9.0
Transceiver Evaluation Software	v0.20.0
ADRV9001 Plugin	v0.20.0

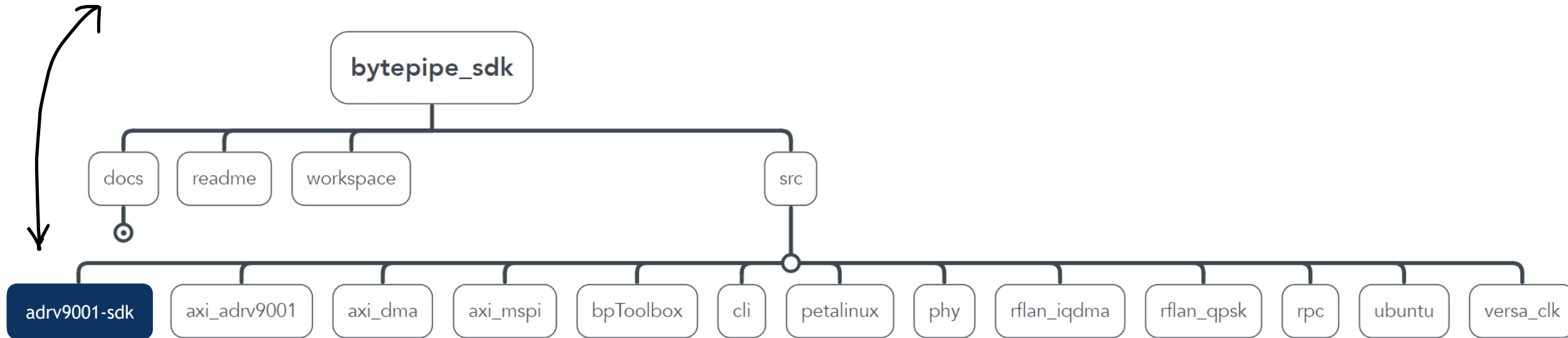
Connected

Radio Carbon DPD

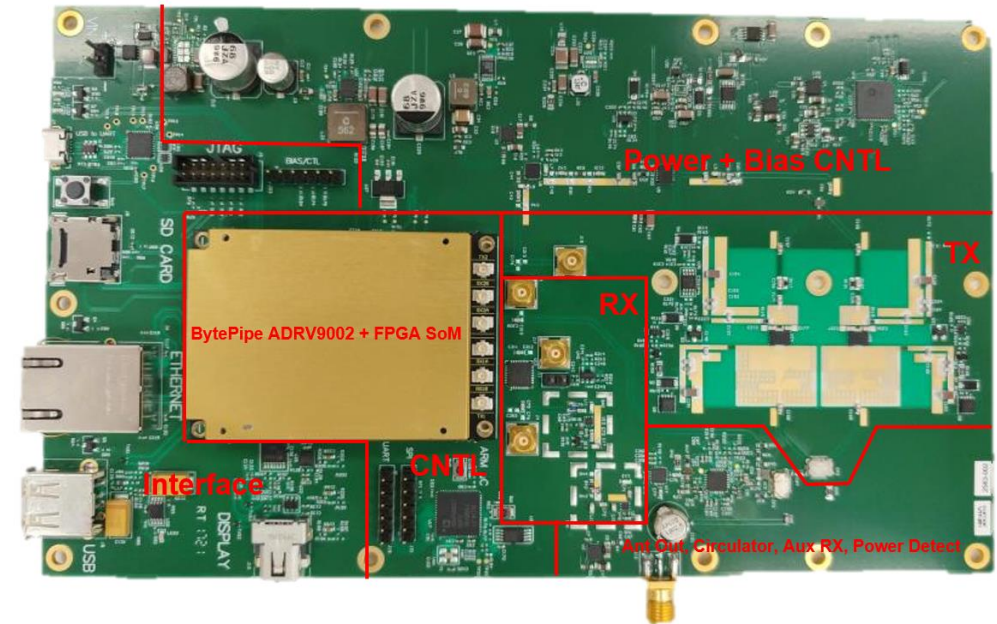
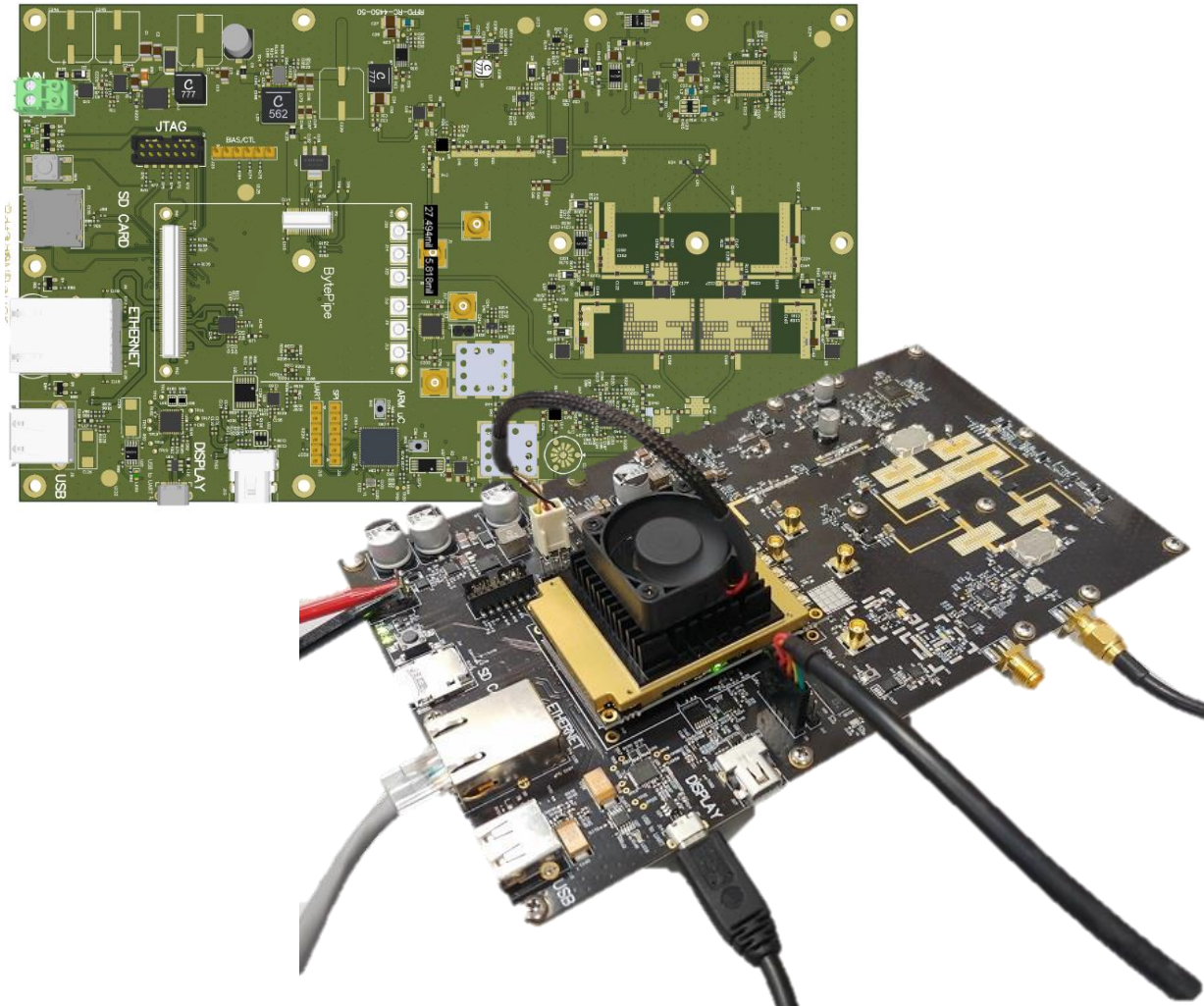
Analog Devices RF Transceiver Evaluation Software



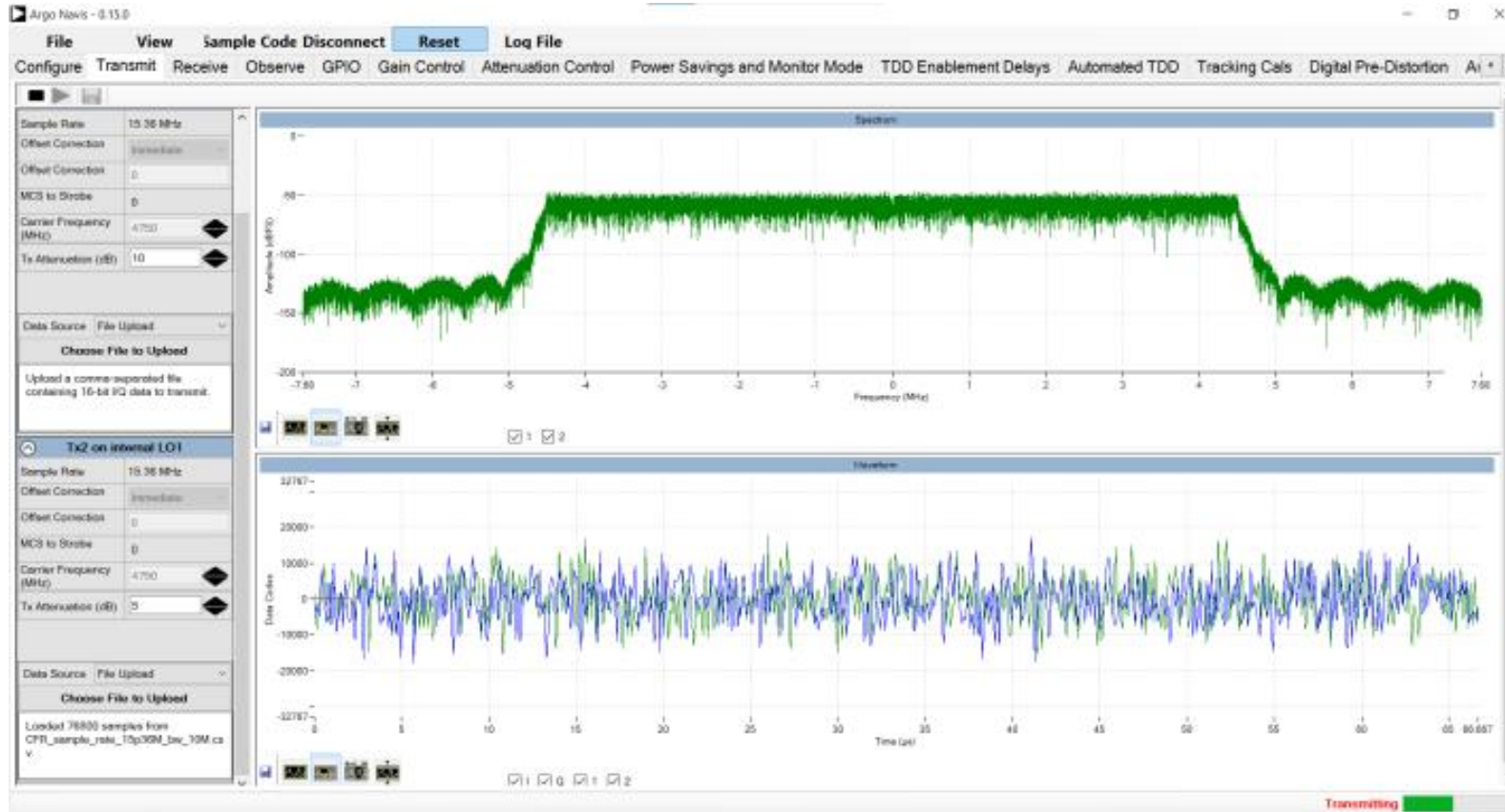
<https://www.analog.com/media/en/evaluation-boards-kits/evaluation-software/adrv9001-sdk-eval-software.zip>



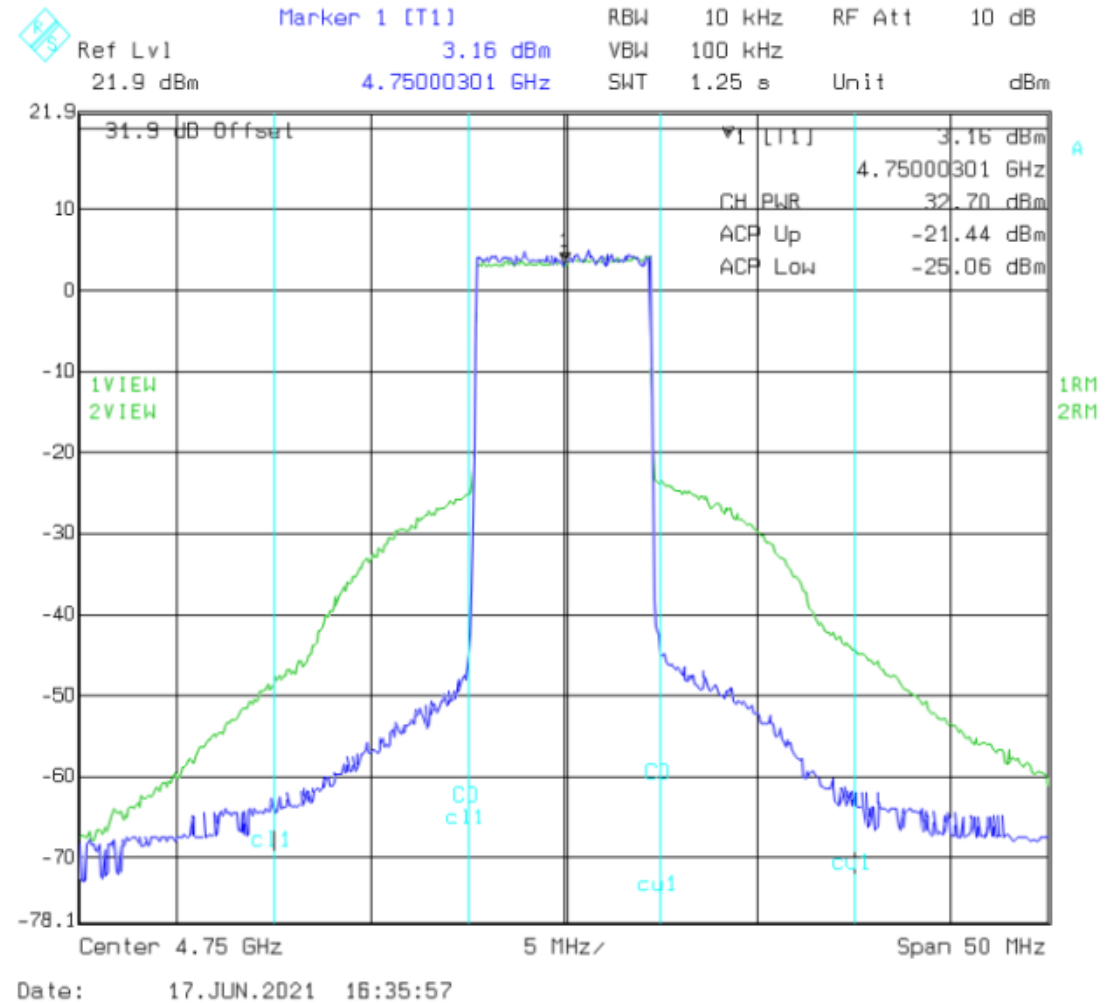
Live Demo - Hardware Setup



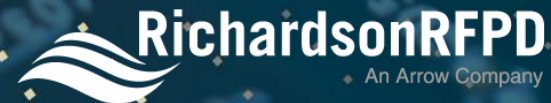
Live Demo - Using TES



Live Demo - Testing

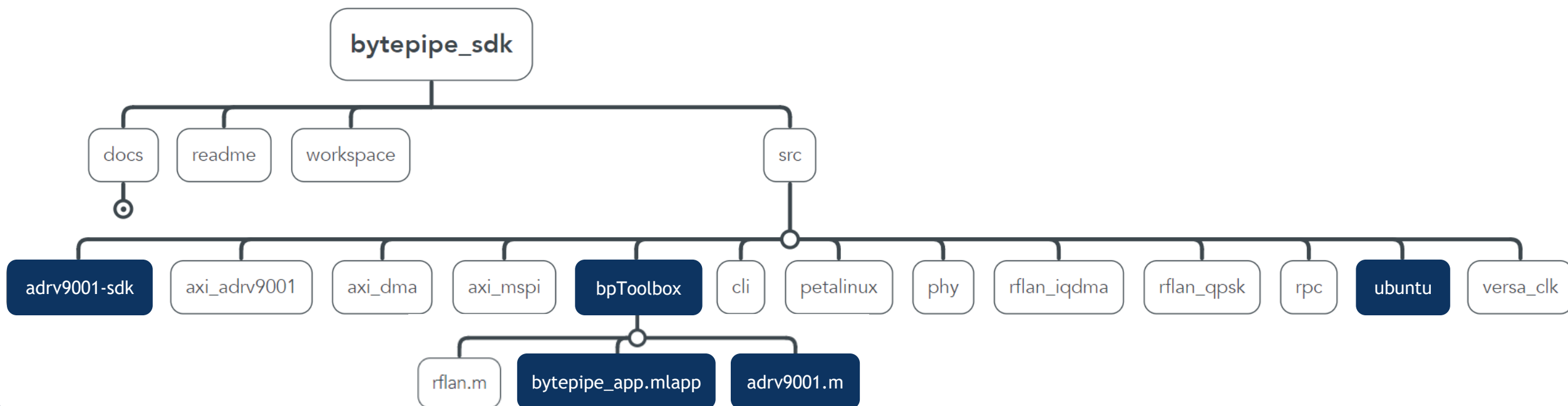


BytePipe Toolbox



Connecting RF Engineers to the Right Technology

BytePipe Toolbox - Adrv9001 Server

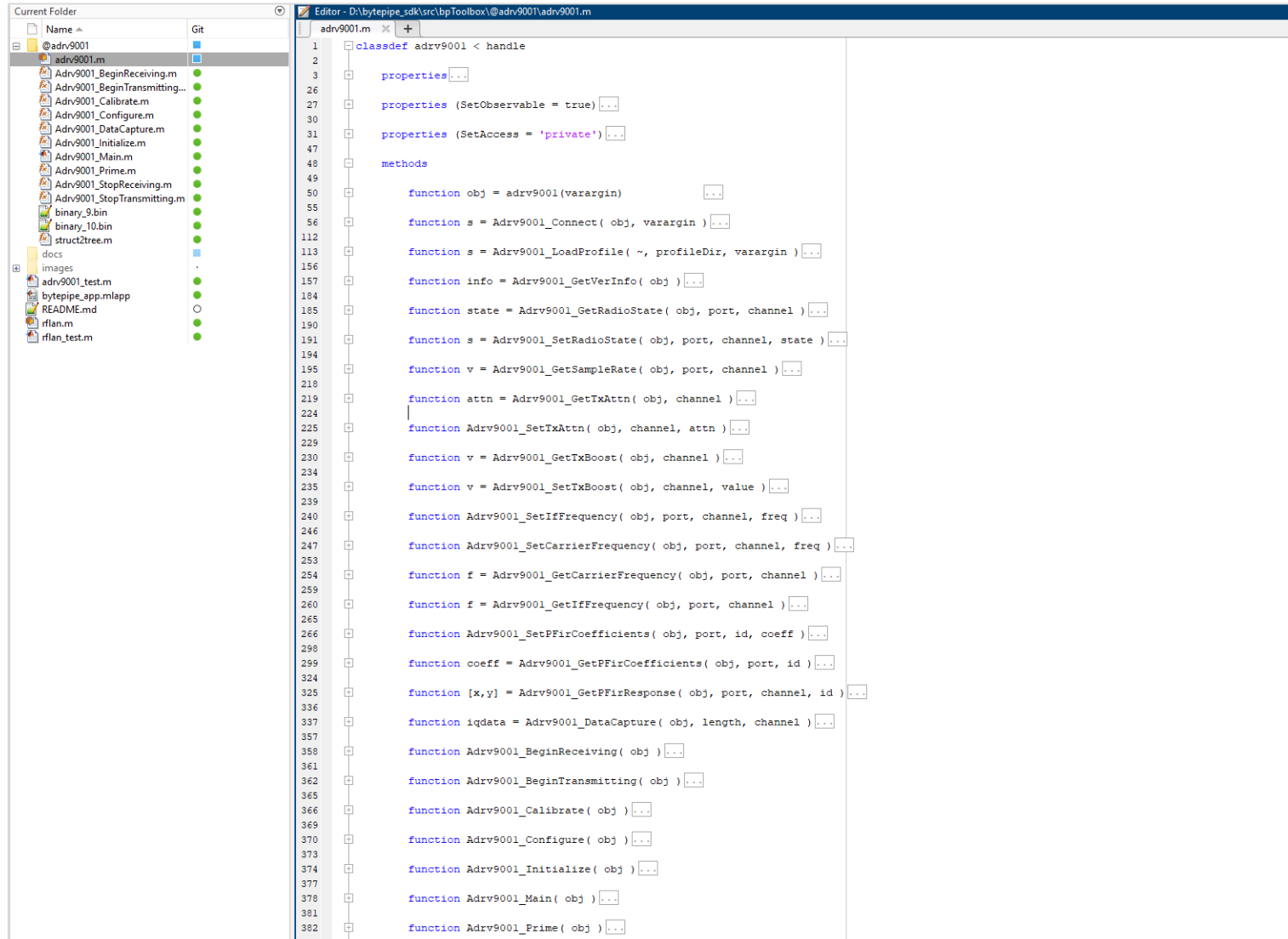


Live Demo - Building Software

The screenshot displays the Vitis IDE interface for a project named 'qpsk_b_vivado'. The top bar shows standard menu options like File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. Below this, the 'PROJECT MANAGER' pane on the left lists various settings and catalogs. The central 'SOURCES' pane shows a list of source files, including 'constraints' and 'utility sources'. The right-hand 'Project Summary' pane provides an overview of the project, including its name, location, product family, and part number. The main editor window shows the 'phy_adrv9001.c' file, which contains C code for the PHY driver. The code includes headers for stdio, string, and stdlib, and defines various macros and functions for the ADRV9001 device. The Explorer pane on the left shows the project's directory structure, including folders for 'hwp', 'platform', 'src', and 'debug'. The code editor shows the following C code:

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include "phy_adrv9001.h"
5 #include "config.h"
6 #include "axi_io.h"
7
8
9 #include "adi_adrv9001_radio.h"
10 #include "adi_adrv9001_profileutil.h"
11 #include "adi_adrv9001_gpio.h"
12 #include "adi_adrv9001.h"
13 #include "adi_adrv9001_tx.h"
14 #include "adi_adrv9001_rx.h"
15 #include "adi_adrv9001_arm.h"
16 #include "adi_adrv9001_auxdac.h"
17 #include "adi_adrv9001_ssi.h"
18
19 #define ADI_CHANNEL_RX1_INIT_MASK (0x40)
20 #define ADI_CHANNEL_RX2_INIT_MASK (0x80)
21 #define ADI_CHANNEL_TX1_INIT_MASK (0x04)
22 #define ADI_CHANNEL_TX2_INIT_MASK (0x08)
23
24 /**
25  ** PHY ADRV9001 Instance
26  **/
27 typedef struct{
28     adi_adrv9001_Device_t  AdiAdrv;    ///< ADRV9001 device
29     adrv9001_hal_t         Hal;        ///< ADRV9001 Instance
30     phy_adrv9001_init_t    *Init;      ///< Reference to initialization data
31     uint32_t               CtrlBase;   ///< Base Address of control
32 }phy_adrv9001_t;
33
34 phy_adrv9001_status_t PhyAdrv9001_GetSsiDelay( void *Instance, phy_adrv9001_ssi_delay_t *Delay ){}
35
36 phy_adrv9001_status_t PhyAdrv9001_SetSsiDelay( void *Instance, phy_adrv9001_ssi_delay_t *Delay ){}
37
38 phy_adrv9001_status_t PhyAdrv9001_ResetSsiPort( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port ){}
39
40 phy_adrv9001_status_t PhyAdrv9001_SetEnablePin( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port, bool Enabled ){}
41
42 phy_adrv9001_status_t PhyAdrv9001_LnaEnable( void *Instance, phy_adrv9001_channel_t Channel, bool Enable ){}
43
44 phy_adrv9001_status_t PhyAdrv9001_GetRssi( void *Instance, phy_adrv9001_channel_t Channel, uint32_t *Rssi_md8 ){}
45
46 phy_adrv9001_status_t PhyAdrv9001_EnableVcTcxo( void *Instance, bool Enable ){}
47
48 phy_adrv9001_status_t PhyAdrv9001_SetVcTcxo( void *Instance, float Voltage ){}
49
50 phy_adrv9001_status_t PhyAdrv9001_GetVcTcxo( void *Instance, float *Voltage ){}
51
52 phy_adrv9001_status_t PhyAdrv9001_IsRfEnabled( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port, bool *Enabled ){}
53
54 phy_adrv9001_status_t PhyAdrv9001_TorCalibrated( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port ){}
55
```

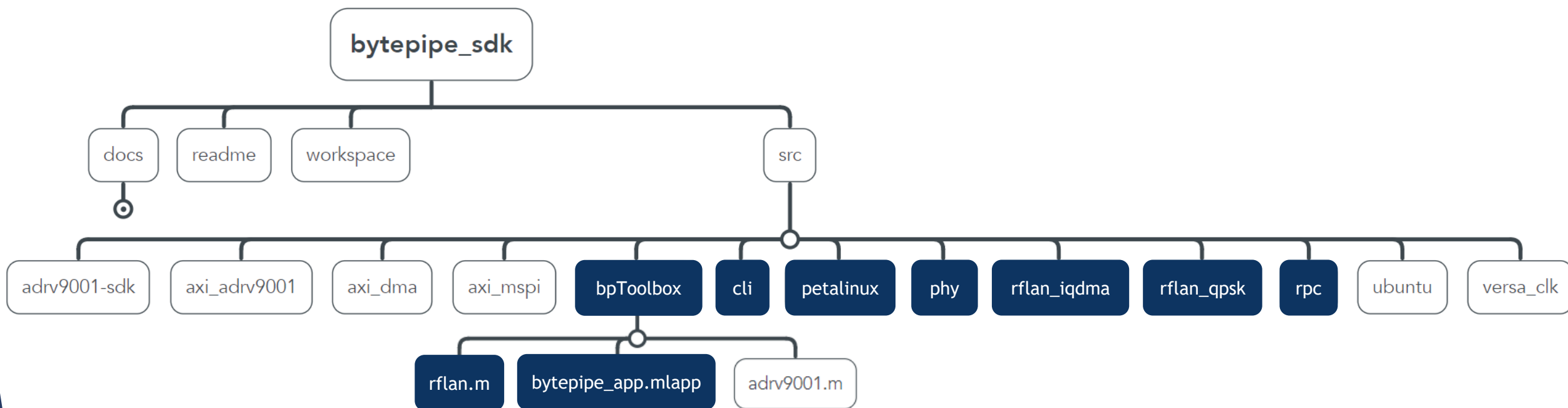
Live Demo - adrv9001.m



The image shows a MATLAB IDE window with two panes. The left pane, titled 'Current Folder', displays a file explorer view of the '@adrv9001' directory. It lists various MATLAB files including 'Adrv9001_BeginReceiving.m', 'Adrv9001_BeginTransmitting.m', 'Adrv9001_Calibrate.m', 'Adrv9001_Configure.m', 'Adrv9001_DataCapture.m', 'Adrv9001_Initialize.m', 'Adrv9001_Main.m', 'Adrv9001_Prime.m', 'Adrv9001_StopReceiving.m', 'Adrv9001_StopTransmitting.m', 'binary_9.bin', 'binary_10.bin', 'struct2tree.m', 'docs', 'images', 'adrv9001_test.m', 'bytepipe_app.m', 'README.md', 'rflan.m', and 'rflan_test.m'. The right pane, titled 'Editor', shows the code for 'adrv9001.m'. The code is a MATLAB class definition for 'adrv9001' which inherits from 'handle'. It includes properties for 'SetObservable' and 'SetAccess', and a series of methods for controlling the radio hardware, such as 'Connect', 'LoadProfile', 'GetVerInfo', 'GetRadioState', 'SetRadioState', 'GetSampleRate', 'GetTxAttn', 'SetTxAttn', 'GetTxBoost', 'SetTxBoost', 'SetIfFrequency', 'SetCarrierFrequency', 'GetCarrierFrequency', 'GetIfFrequency', 'SetPFirCoefficients', 'GetPFirCoefficients', 'GetPFirResponse', 'DataCapture', 'BeginReceiving', 'BeginTransmitting', 'Calibrate', 'Configure', 'Initialize', 'Main', and 'Prime'.

```
1 classdef adrv9001 < handle
2
3     properties ...
4
5     properties (SetObservable = true) ...
6
7     properties (SetAccess = 'private') ...
8
9     methods
10
11         function obj = adrv9001(varargin) ...
12
13         function s = Adrv9001_Connect( obj, varargin ) ...
14
15         function s = Adrv9001_LoadProfile( ~, profileDir, varargin ) ...
16
17         function info = Adrv9001_GetVerInfo( obj ) ...
18
19         function state = Adrv9001_GetRadioState( obj, port, channel ) ...
20
21         function s = Adrv9001_SetRadioState( obj, port, channel, state ) ...
22
23         function v = Adrv9001_GetSampleRate( obj, port, channel ) ...
24
25         function attn = Adrv9001_GetTxAttn( obj, channel ) ...
26         function Adrv9001_SetTxAttn( obj, channel, attn ) ...
27
28         function v = Adrv9001_GetTxBoost( obj, channel ) ...
29         function v = Adrv9001_SetTxBoost( obj, channel, value ) ...
30
31         function Adrv9001_SetIfFrequency( obj, port, channel, freq ) ...
32
33         function Adrv9001_SetCarrierFrequency( obj, port, channel, freq ) ...
34
35         function f = Adrv9001_GetCarrierFrequency( obj, port, channel ) ...
36         function f = Adrv9001_GetIfFrequency( obj, port, channel ) ...
37
38         function Adrv9001_SetPFirCoefficients( obj, port, id, coeff ) ...
39         function coeff = Adrv9001_GetPFirCoefficients( obj, port, id ) ...
40
41         function [x,y] = Adrv9001_GetPFirResponse( obj, port, channel, id ) ...
42
43         function iqdata = Adrv9001_DataCapture( obj, length, channel ) ...
44
45         function Adrv9001_BeginReceiving( obj ) ...
46         function Adrv9001_BeginTransmitting( obj ) ...
47
48         function Adrv9001_Calibrate( obj ) ...
49         function Adrv9001_Configure( obj ) ...
50
51         function Adrv9001_Initialize( obj ) ...
52
53         function Adrv9001_Main( obj ) ...
54
55         function Adrv9001_Prime( obj ) ...
56
57     end
58 end
```


BytePipe Toolbox - Rflan Server

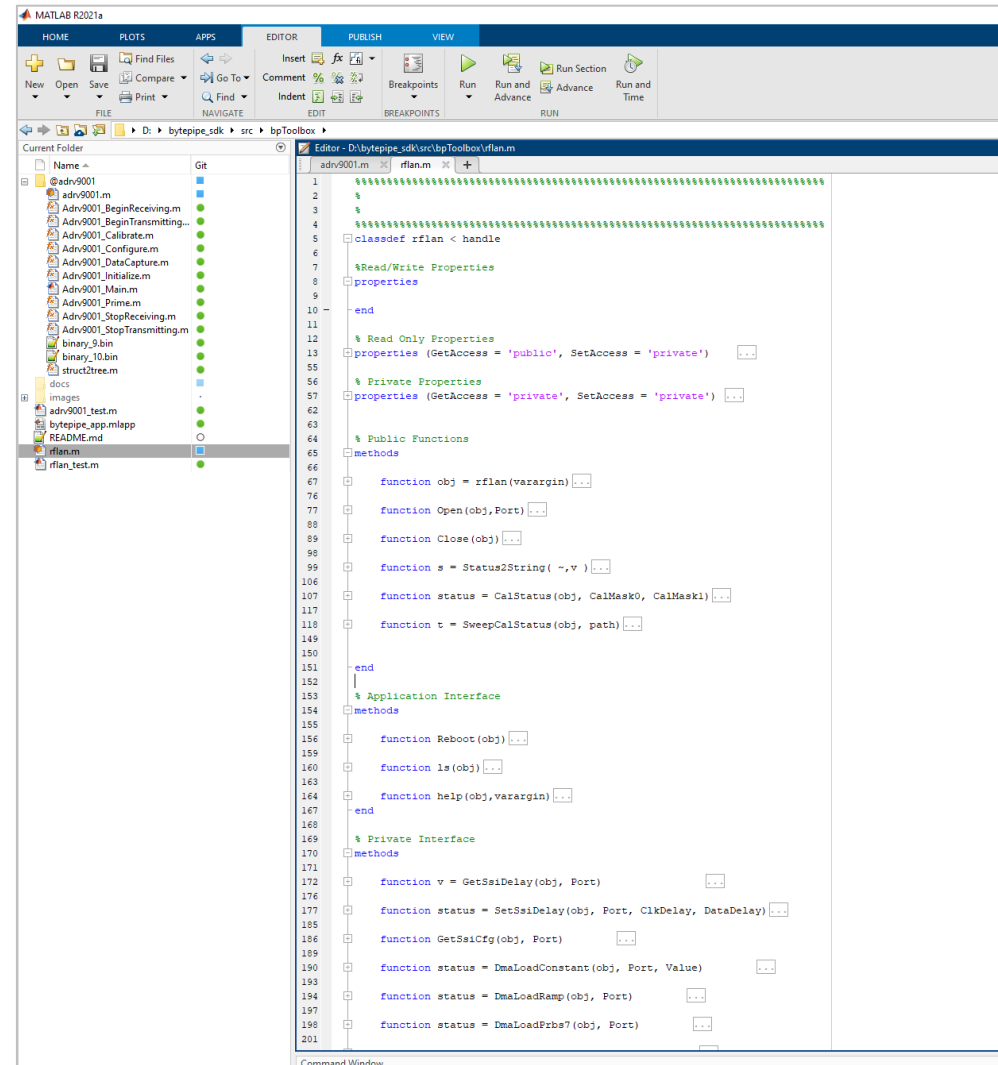


Live Demo - Building Software

The screenshot displays the Vitis IDE interface for a project named 'qpsk_b_vivado'. The top bar shows standard menu options like File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. Below this, the 'PROJECT MANAGER' pane on the left lists various settings and catalogs. The central 'SOURCES' pane shows a list of source files, including 'constraints' and 'utility sources'. The right-hand 'Project Summary' pane provides an overview of the project, including its name, location, product family, and part number. The main editor window shows the 'phy_adrv9001.c' file, which contains C code for the PHY driver. The code includes headers for stdio, string, and stdlib, and defines various macros and functions for the ADRV9001 device. The Explorer pane on the left shows the project's directory structure, including folders for 'hwp', 'platform', 'src', and 'debug'. The bottom status bar indicates the current file is 'phy_adrv9001.c'.

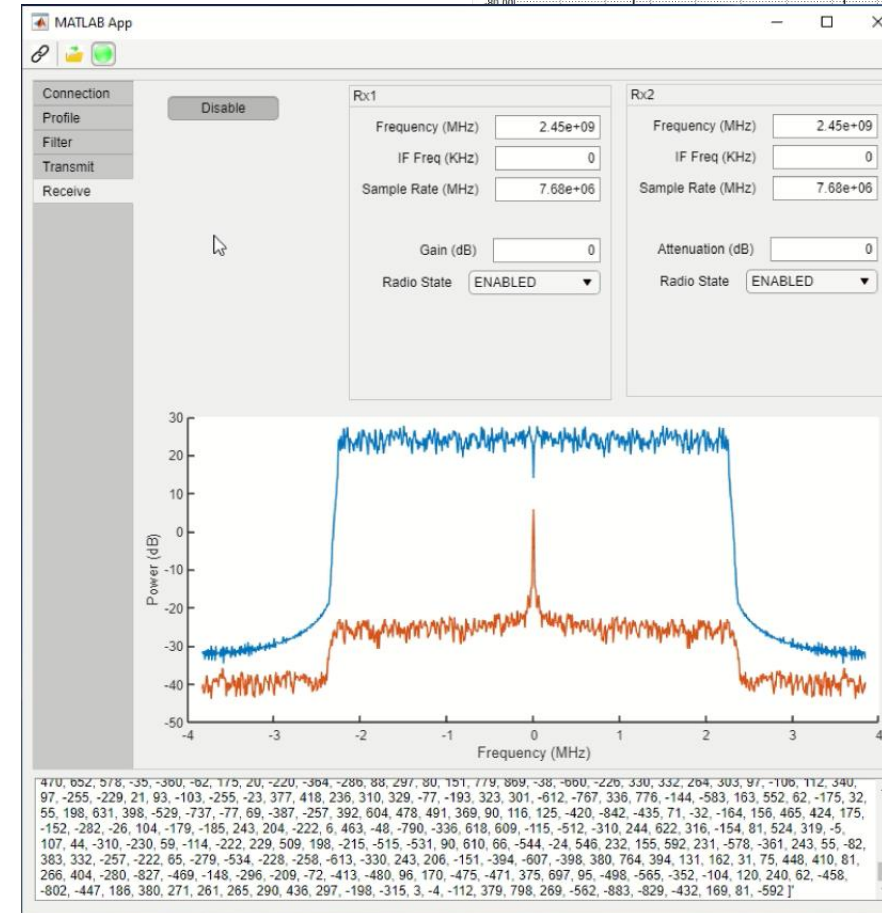
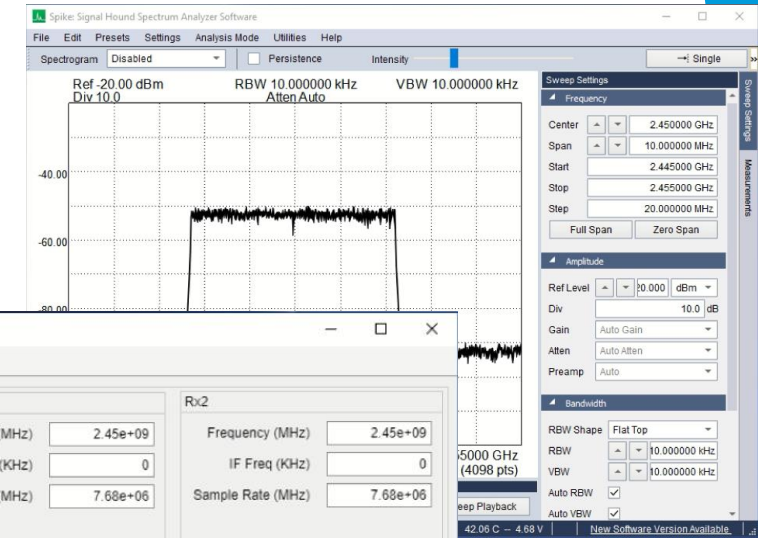
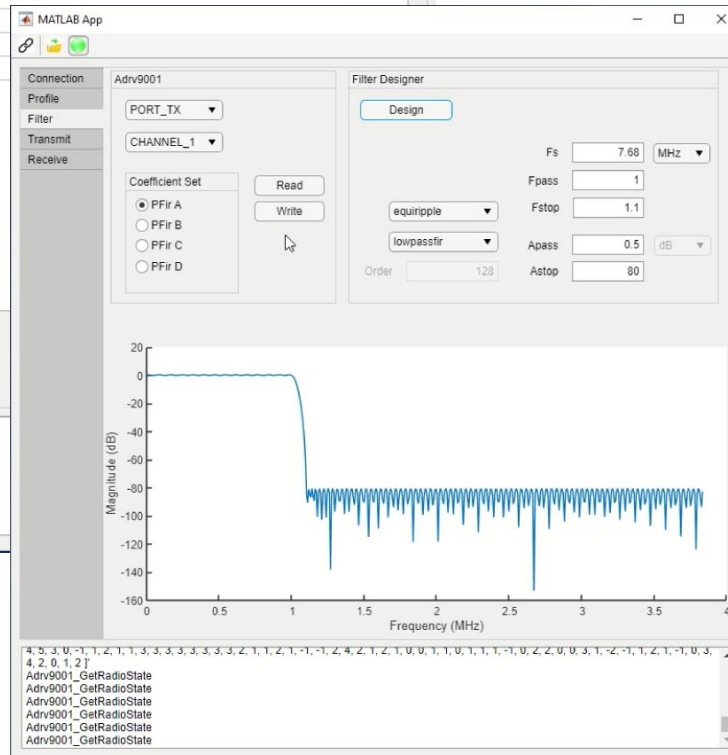
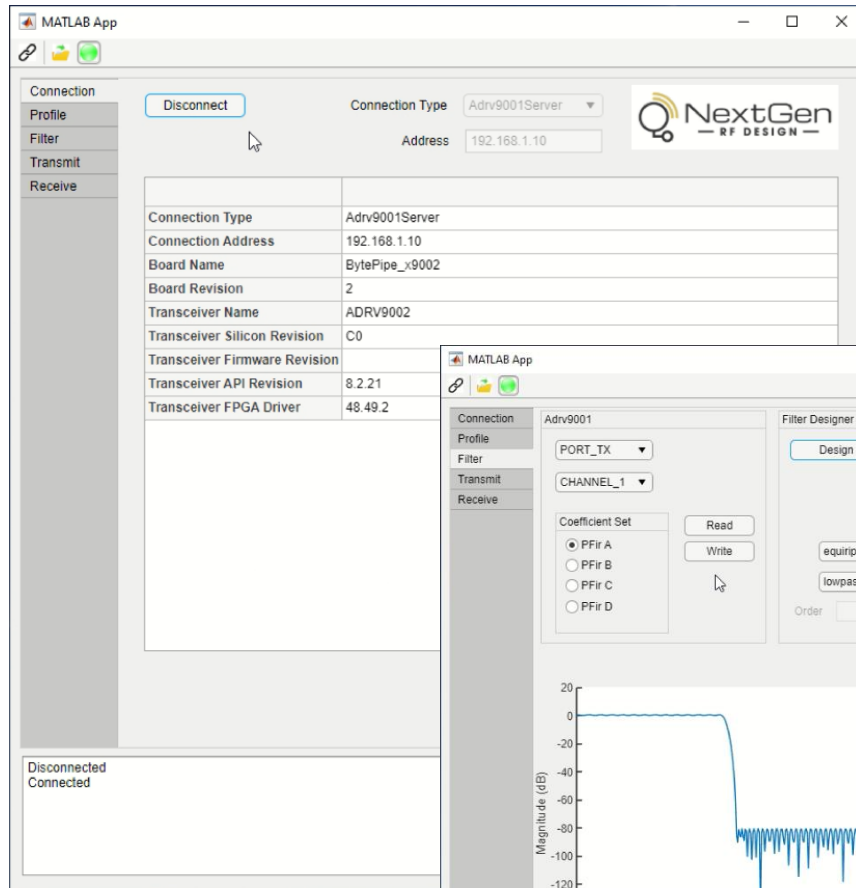
```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include "phy_adrv9001.h"
5 #include "config.h"
6 #include "axi_io.h"
7
8
9 #include "adi_adrv9001_radio.h"
10 #include "adi_adrv9001_profileutil.h"
11 #include "adi_adrv9001_gpio.h"
12 #include "adi_adrv9001.h"
13 #include "adi_adrv9001_tx.h"
14 #include "adi_adrv9001_rx.h"
15 #include "adi_adrv9001_arm.h"
16 #include "adi_adrv9001_auxdac.h"
17 #include "adi_adrv9001_ssi.h"
18
19 #define ADI_CHANNEL_RX1_INIT_MASK (0x40)
20 #define ADI_CHANNEL_RX2_INIT_MASK (0x80)
21 #define ADI_CHANNEL_TX1_INIT_MASK (0x04)
22 #define ADI_CHANNEL_TX2_INIT_MASK (0x08)
23
24 /**
25  ** PHY ADRV9001 Instance
26  **/
27 typedef struct{
28     adi_adrv9001_Device_t  AdiAdr;    ///< ADRV9001 device
29     adrv9001_hal_t         Hal;       ///< ADRV9001 Instance
30     phy_adrv9001_init_t    *Init;    ///< Reference to initialization data
31     uint32_t               CtrlBase;  ///< Base Address of control
32 }phy_adrv9001_t;
33
34 @phy_adrv9001_status_t PhyAdr9001_GetSsiDelay( void *Instance, phy_adrv9001_ssi_delay_t *Delay ){}
35
36 @phy_adrv9001_status_t PhyAdr9001_SetSsiDelay( void *Instance, phy_adrv9001_ssi_delay_t *Delay ){}
37
38 @phy_adrv9001_status_t PhyAdr9001_ResetSsiPort( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port ){}
39
40 @phy_adrv9001_status_t PhyAdr9001_SetEnablePin( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port, bool Enabled ){}
41
42 @phy_adrv9001_status_t PhyAdr9001_LnaEnable( void *Instance, phy_adrv9001_channel_t Channel, bool Enable ){}
43
44 @phy_adrv9001_status_t PhyAdr9001_GetRssi( void *Instance, phy_adrv9001_channel_t Channel, uint32_t *Rssi_md8 ){}
45
46 @phy_adrv9001_status_t PhyAdr9001_EnableVcTcxo( void *Instance, bool Enable ){}
47
48 @phy_adrv9001_status_t PhyAdr9001_SetVcTcxo( void *Instance, float Voltage ){}
49
50 @phy_adrv9001_status_t PhyAdr9001_GetVcTcxo( void *Instance, float *Voltage ){}
51
52 @phy_adrv9001_status_t PhyAdr9001_IsRfEnabled( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port, bool *Enabled ){}
53
54 @phy_adrv9001_status_t PhyAdr9001_TorFCalibrated( void *Instance, phy_adrv9001_channel_t Channel, phy_adrv9001_port_t Port ){}
55
```

Live Demo - rflan.m

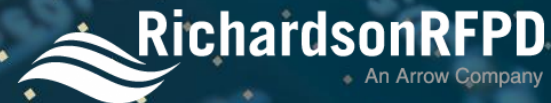


```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
2 %  
3 %  
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
5 classdef rflan < handle  
6 %Read/Write Properties  
7 properties  
8  
9  
10 end  
11  
12 % Read Only Properties  
13 properties (GetAccess = 'public', SetAccess = 'private')  
14  
15  
16 % Private Properties  
17 properties (GetAccess = 'private', SetAccess = 'private')  
18  
19  
20 % Public Functions  
21 methods  
22  
23 function obj = rflan(varargin)  
24  
25  
26 function Open(obj, Port)  
27  
28  
29 function Close(obj)  
30  
31  
32 function s = Status2String(., v)  
33  
34  
35 function status = CalStatus(obj, CalMask0, CalMask1)  
36  
37  
38 function t = SweepCalStatus(obj, path)  
39  
40  
41 end  
42  
43 % Application Interface  
44 methods  
45  
46 function Reboot(obj)  
47  
48  
49 function ls(obj)  
50  
51  
52 function help(obj, varargin)  
53  
54 end  
55  
56 % Private Interface  
57 methods  
58  
59 function v = GetSsiDelay(obj, Port)  
60  
61  
62 function status = SetSsiDelay(obj, Port, ClkDelay, DataDelay)  
63  
64  
65 function GetSsiCfg(obj, Port)  
66  
67  
68 function status = DmaLoadConstant(obj, Port, Value)  
69  
70  
71 function status = DmaLoadRamp(obj, Port)  
72  
73  
74 function status = DmaLoadPrbs(obj, Port)  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201
```

 Live Demo - byteword_app

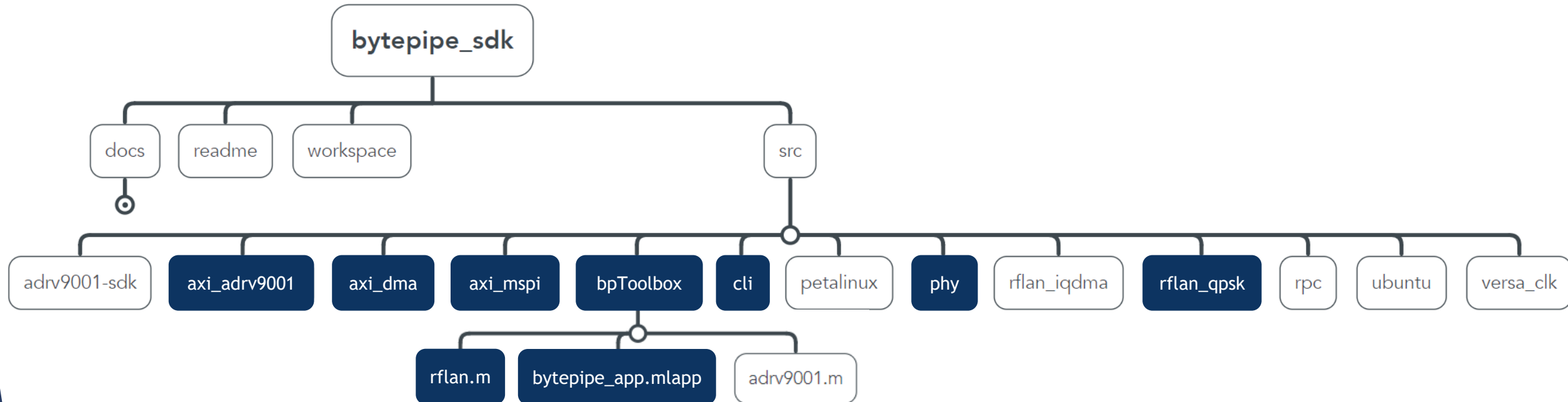


BytePipe HDL Coder

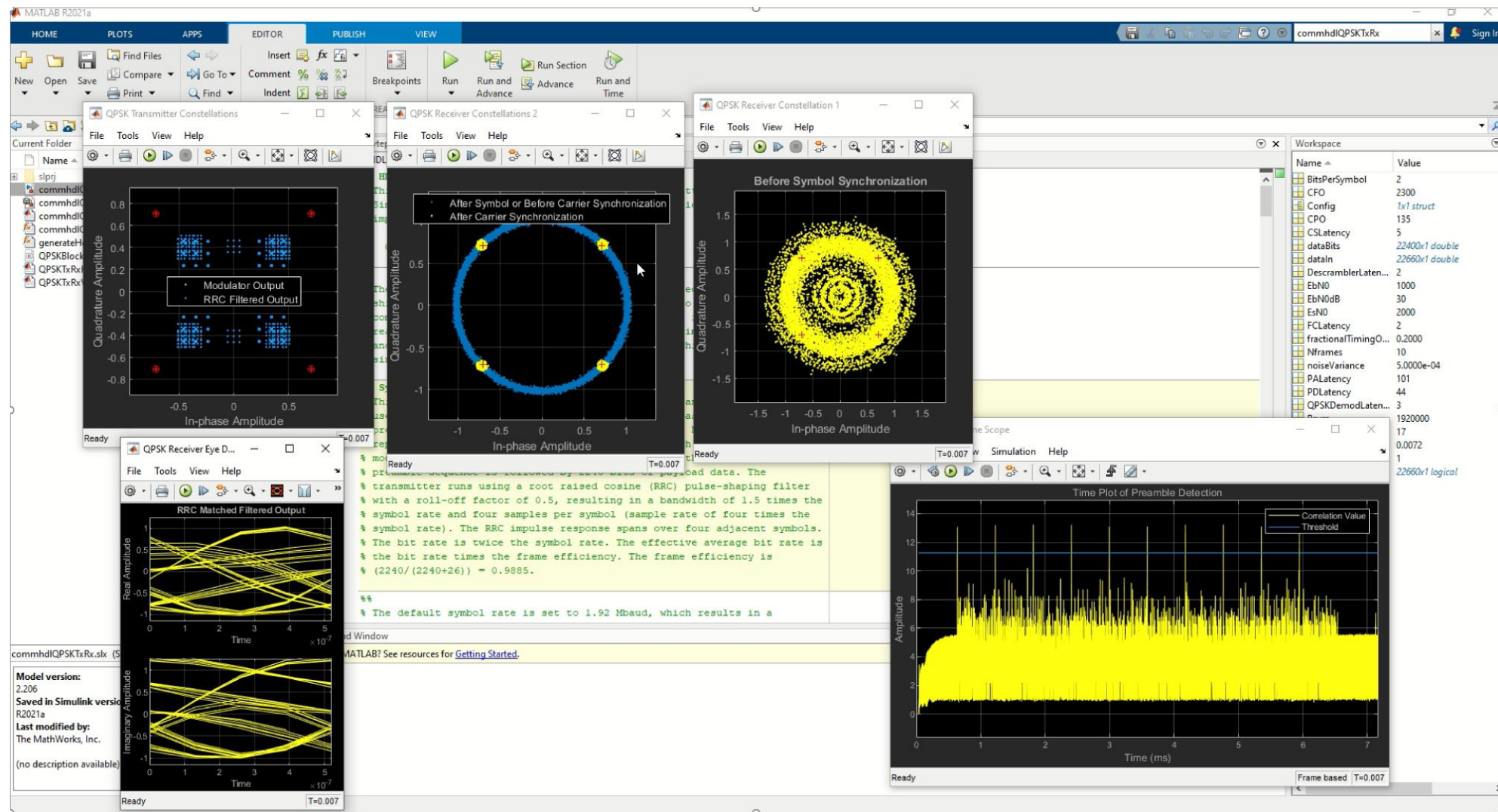


Connecting RF Engineers to the Right Technology

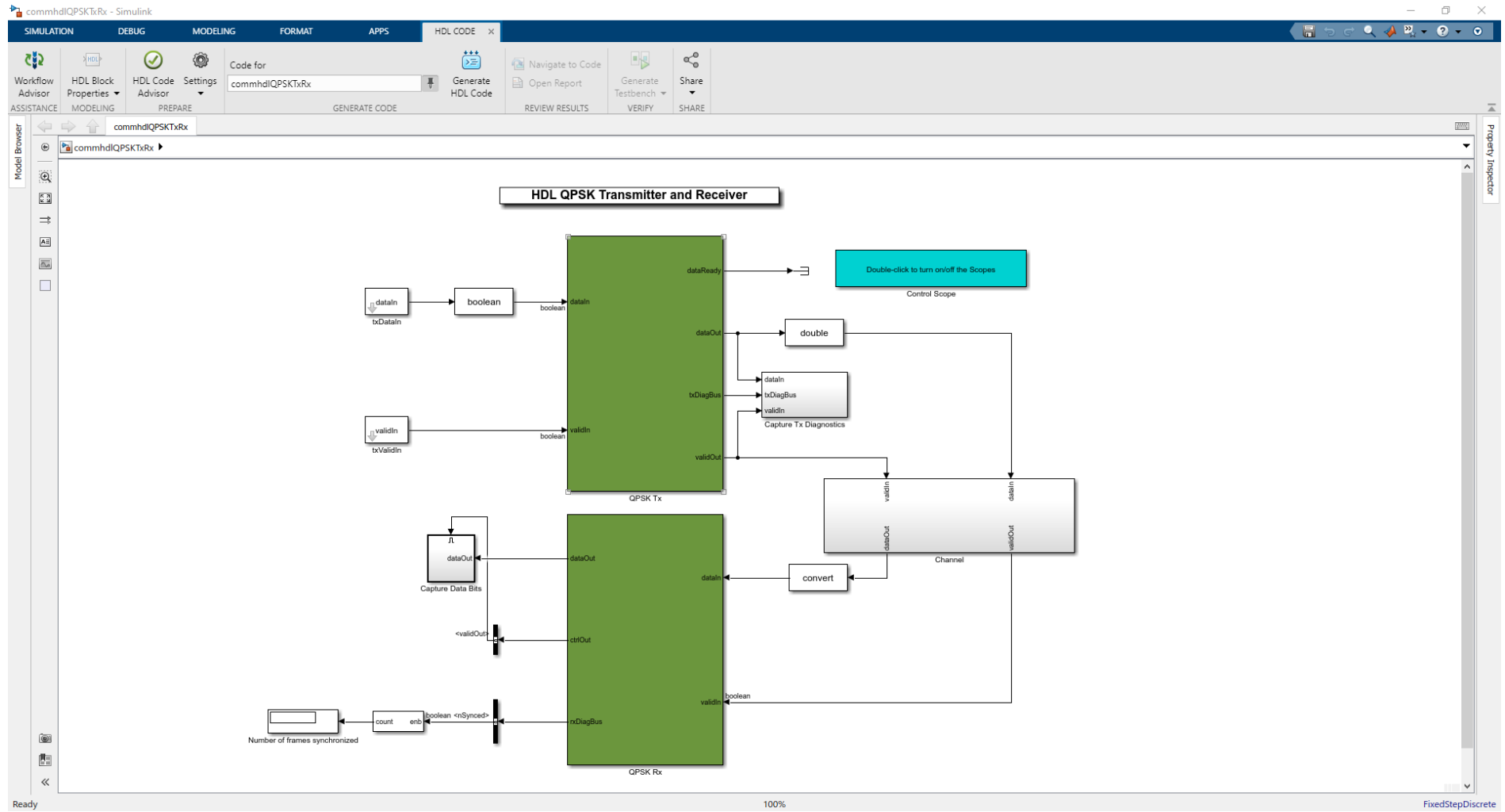
BytePipe Toolbox - rflan_qpsk

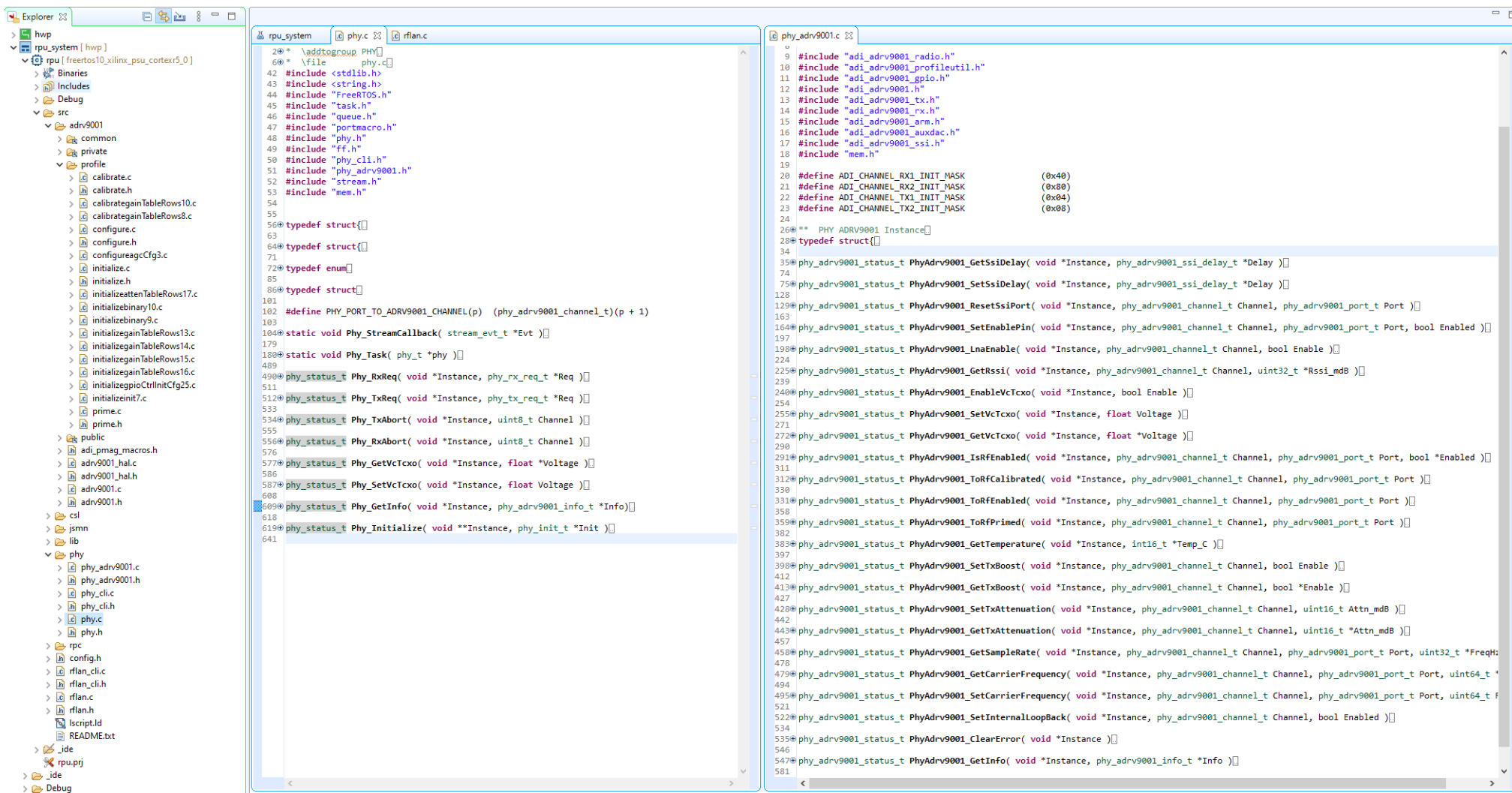


Live Demo - Simulating Model

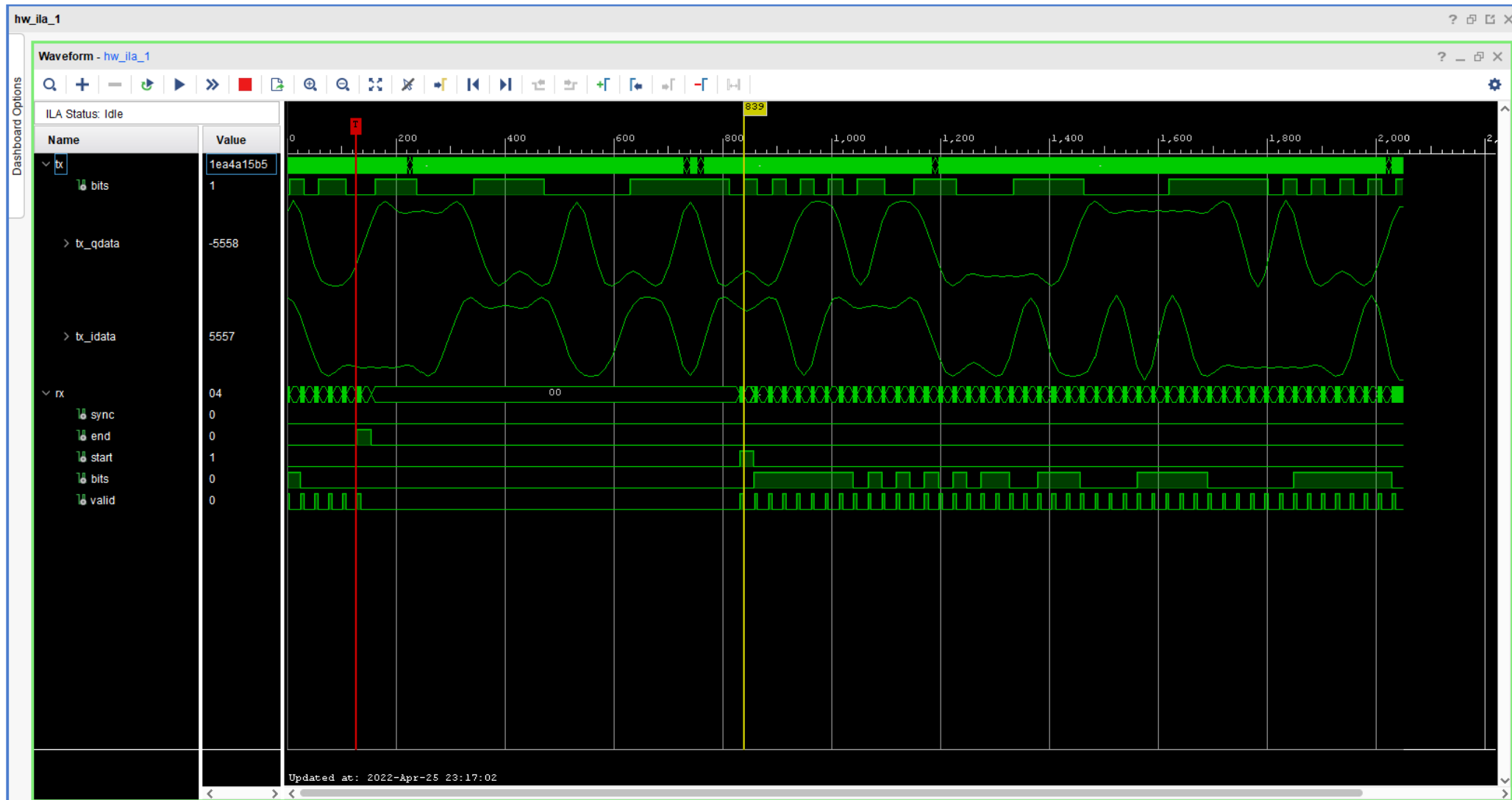


Live Demo - Building HDL





Live Demo - Simulating HDL



Live Demo - Testing

MATLAB App

Connection Type: Adrv9001Server
Address: 192.168.1.10

NextGen
— RF DESIGN —

Connection Type	Adrv9001Server
Connection Address	192.168.1.10
Board Name	BytePipe_x9002
Board Revision	2
Transceiver Name	ADRV9002
Transceiver Silicon Revision	C0
Transceiver Firmware Revision	
Transceiver API Revision	8.2.21
Transceiver FPGA Driver	48.49.2

Disconnected
Connected

```
Term VT
Control Window Help

the registered commands.
sorting string with first letters of command(s)
optional) >
Task Info

able Transmission from file.
anel (0 or 1), SampleCnt (-1=cyclic, 0 = file size)
abort active transmit request
channel (0 or 1) >

able receiver and save data to file or print to s
anel (0 or 1), SampleCnt, filename (optional) >
abort active receive request
channel (0 or 1) >

e: Load PHY Profile
< Index >
et PHY parameters.
PHY parameter.
>
Get UCTICK0 voltage
>
Set UCTICK0 voltage
voltage (0 to 1.8) >
```

