XRM2-RF-ATD Reference Design for ADM-XRC-7Z1

Introduction

The **XRM2-RF-ATD Reference Design for ADM-XRC-7Z1** is a set of resources for FPGA designers and software engineers working with Alpha Data's ADM-XRC-7Z1 and XRM2-RF-ATD boards that provides an example design to utilize the Analog Devices AD9361 RF Agile Transceiver.

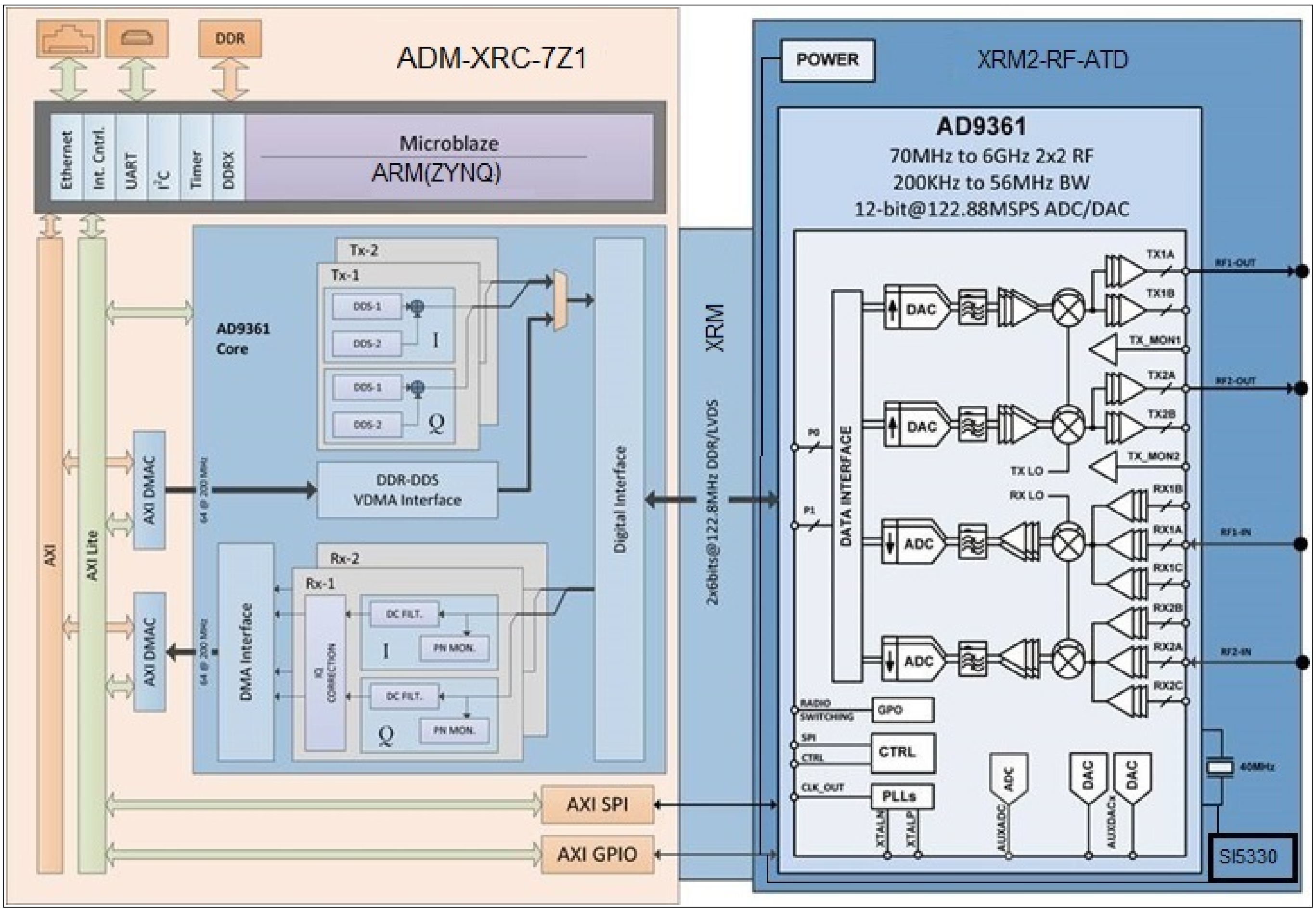


Figure 1 : XRM2-RF-ATD Reference Design for ADM-XRC-7Z1 Block Diagram

Note
The sources for the hdl and software projects have been using:
https://github.com/analogdevicesinc sources and been modified to support Alpha-Data boards.

The resources of the XRM2-RF-ATD Reference Design for ADM-XRC-7Z1 include:

- FPGA Vivado designs for Alpha-Data ADM-XRC-7Z1 board.
- no-OS Vitis software projects for Alpha-Data ADM-XRC-7Z1 board.
- Linux software projects for Alpha-Data ADM-XRC-7Z1 board.
- Documentation of how to build and use the XRM2-RF-ATD Reference Design for ADM-XRC-7Z1.
- Analog Devices libiio platform with example test batch files to run on Alpha-Data ADM-XRC-7Z1 board.
- Python example test codes to run on Alpha-Data ADM-XRC-7Z1 board.
- IIO Oscilloscope GTK+ application example test ini files to run on Alpha-Data ADM-XRC-7Z1 board.
- Prebuilt files to run on Alpha-Data ADM-XRC-7Z1 board.

Structure of this package

The directories making up the XRM2-RF-ATD Reference Design for ADM-XRC-7Z1 are organised as in Figure 2 below:

```
(ad_adi_7z1_'version') ........ The root of this Example Design
 ├─doc ........................ Contains this document
 ├─hdl
 │  └─projects ................ Contains the Alpha-Data board source files to create the FPGA project
 ├─no-OS
 │  └─projects ................ Contains the Alpha-Data board source files to create the no-OS Vitis project
 ├─linux
 │  └─meta-adi ................ Contains the Alpha-Data board source files to create the Petalinux project
 ├─libiio-0.25
 │  └─examples ................ Contains batch files to run example tests
 ├─pyadi-iio
 │  └─examples ................ Contains python files to run example tests
 ├─iio-oscilloscope
 │  └─profiles ................ Contains ini files to run example tests
 └─prebuilt
    ├─xsa ..................... Contains prebuilt .xsa files
    └─sd ...................... Contains prebuilt SD card images
```

Figure 2 : Structure of the Example Design

Recommended Vivado, Vitis and Petalinux versions

Recommended Vivado versions for the example FPGA designs is Vivado 2023.1 onwards.

Recommended Vitis version for the example designs is Vitis 2022.2 .

Recommended Petalinux version for the example designs is Petalinux 2023.1 .

Git-Bash version

Recommended Git-Bash versions to create Vivado and Vitis projects is Git-Bash 2.43.0 onwards.

Building HDL projects

1. Open Git-Bash
2. Set up Windows enviroment for Vivado and Vitis e.g:
   "export PATH=/d/Vitis/Vitis/2022.2/bin:/d/Vitis/Vitis/2022.2/gnu/aarch32/nt/gcc-arm-none-eabi/bin/:$PATH"
   "export PATH=/d/Vitis/Vivado/2023.2/bin:$PATH"
3. cd to the project and run "make" e.g:
   "cd ad_adi_7z1_'version'/hdl/projects/ad_rf_atd/7z1/"
   "make"
4. Wait until the project is been built and run. At the end you can find a "system_top.xsa" on the sdk folder of the built project that you can use for your Vitis or Linux project.
5. To clean the project use "make clean" command.

> **Note**
> More informations about the HDL projects and how to build them you can find on Analog Devices Github: **https://github.com/analogdevicesinc/hdl**.

Building no-OS project

1. Copy the "system_top.xsa" from the hdl project to the no-OS project folder e.g: "ad_adi_7z1_'version'/no-OS/projects/ad9361_7z1".
2. Open Git-Bash
3. Set up Windows enviroment for Vivado and Vitis e.g:
   "export PATH=/d/Vitis/Vitis/2022.2/bin:/d/Vitis/Vitis/2022.2/gnu/aarch64/nt/aarch64-none/bin/:$PATH"
   "export PATH=/d/Vitis/Vivado/2023.2/bin:$PATH"
4. cd to the project and run "make" e.g:
   "cd ad_adi_7z1_'version'/no-OS/projects/ad9361_7z1/"
   "make"
5. Wait until the project is been built.
6. Run "make sdkopen" to open the project in Vitis
7. To clean the project use "make reset" command.
8. Serial settings for no-OS are: Baudrate: 460800, Data Size: 8 bits, Parity: None , Stop bits: 1

> **Note**
> More informations about the no-OS projects and how to build them you can find on Analog Devices Github: **https://github.com/analogdevicesinc/no-OS**.

Here is an image with serial output running the no-OS on Alpha-Data's AD01253-ADM-XRC-7Z1 fpga board and AD01289-XRM2-RF-ATD IO adapter board that has Analog Devices AD9361 IC:

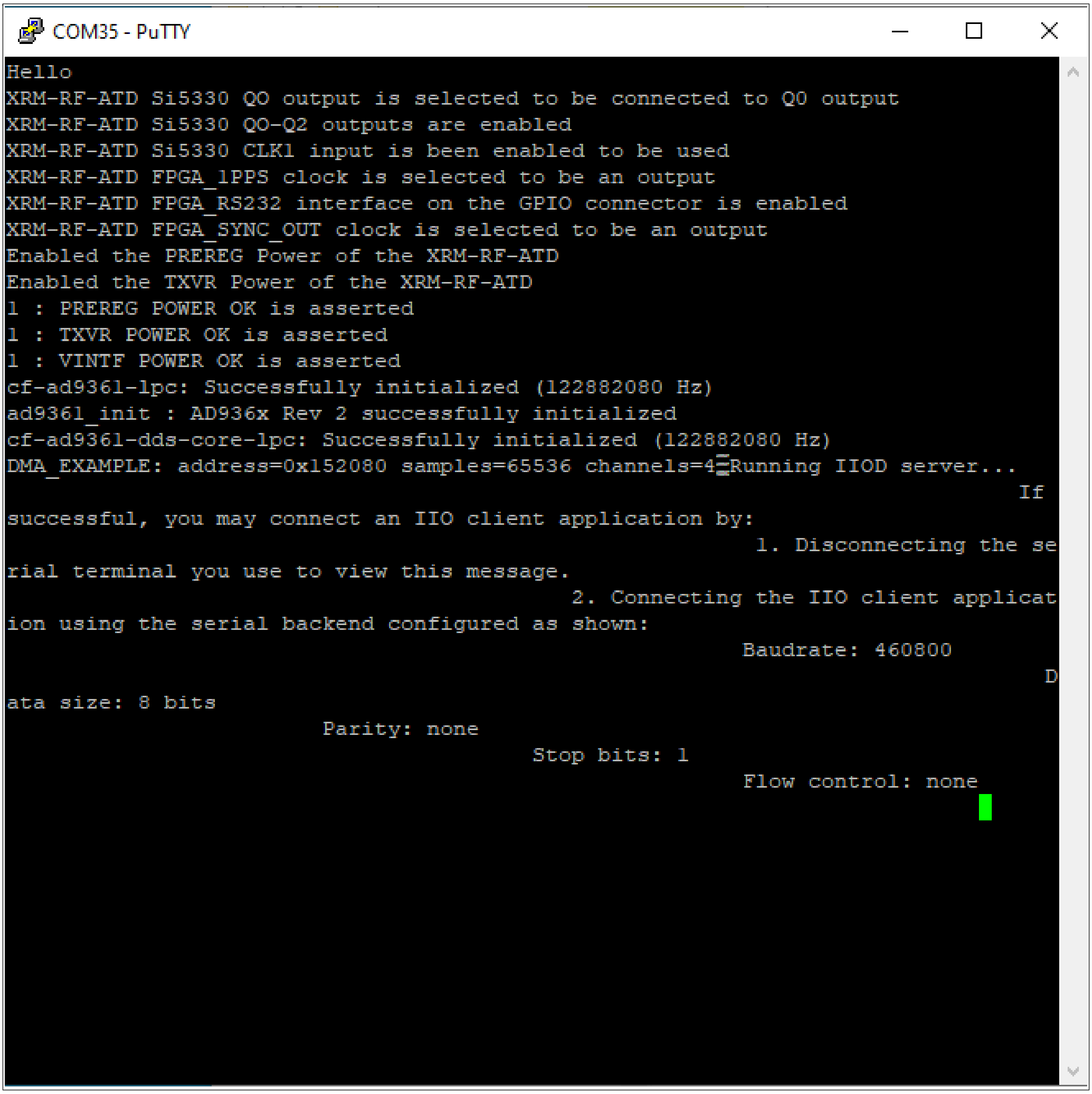


Figure 3 : ADM-XRC-7Z1:XRM2-RF-ATD no-OS project successful configuration

Petalinux OS and Building Linux Applications

1. Copy "met-adi" to a location accessible in Linux.
2. Source the PetaLinux configuration script e.g:
   "source "PetaLinux_install_dir"/settings.sh"
3. Copy the .xsa file to a location accessible in Linux, if it is not already
4. Create the PetaLinux project and change into the project directory, then configure the project to use the xsa file for the ADM-XRC-7Z1 e.g:
   "petalinux-create -t project --template zynqMP --name 7z1"
   "cd 7z1"
5. Load the hardware description file .xsa created by Vivado e.g:
   "petalinux-config --get-hw-description /path/to/hardware/description/file
6. When running the petalinux-config --get-hw-description="path to xsa file", a configuration menu will come up. Go to Yocto Settings->User layers and add the meta-adi-xilinx layer e.g:
   "/home/user/meta-adi/meta-adi-xilinx"
7. After you save the configuration menu select the devicetree that fits the project being built e.g:
   "echo "KERNEL_DTB=\"7z1\"" >> project-spec/meta-user/conf/petalinuxbsp.conf"
   'device tree files directory: "home\user\meta-adi\meta-adi-xilinx\recipes-bsp\device-tree\files"'
8. Change the directory to the build folder e.g:
   "cd build"
9. To build the Linux image, run e.g:
   "petalinux-build"
10. Once Linux is built, a BOOT.bin file can be created e.g:
    "petalinux-package --boot --fsbl --fpga --u-boot --force"
11. This will generate the following files can be found in the "plnx_proj"/images/linux directory:
    "BOOT.bin Bootloader, containing FSBL, u-boot and bitstream"
    "image.ub Linux kernel, root filesystem, device-tree."
    "boot.scr Boot script (2019.2 or later)"
12. Serial settings for Linux are: Baudrate: 115200, Data Size: 8 bits, Parity: None , Stop bits: 1
13. When you boot from the SD card successfully a username and password will be asked e.g:
    "username: root"
    "password: analog"

Note
More informations about the meta-adi projects and how to build them you can find on Analog Devices Github: **https://github.com/analogdevicesinc/meta-adi/tree/main/meta-adi-xilinx**.

libiio

libiio is a library that allows to intreface with IIO Subsystems

> **Note**
> More informations about libiio you can find on Analog Devices Github: **https://github.com/analogdevicesinc/libiio**.

1. To run the test example batch files, open cmd and cd to the examples folders e.g:
   "Drive:\> cd ad_adi_7z1_'version'\libiio-0.25\examples"
2. Run the batch file example test e.g:
   "Drive:\ad_adi_7z1_'version'\libiio-0.25\examples>xrm_ad_rf_atd_test.bat"

Here are images from the example test run on Alpha-Data's AD01253-ADM-XRC-7Z1 fpga board and AD01289-XRM2-RF-ATD IO adapter board that has Analog Devices AD9361 IC:

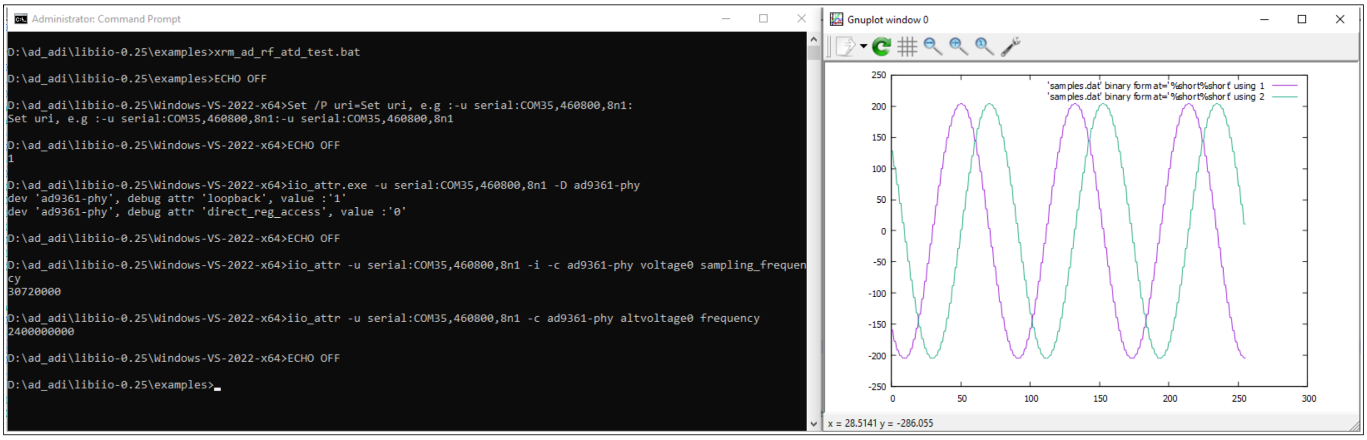


Figure 4 : ADM-XRC-7Z1:XRM2-RF-ATD TX-RX loopback test

pyadi-iio

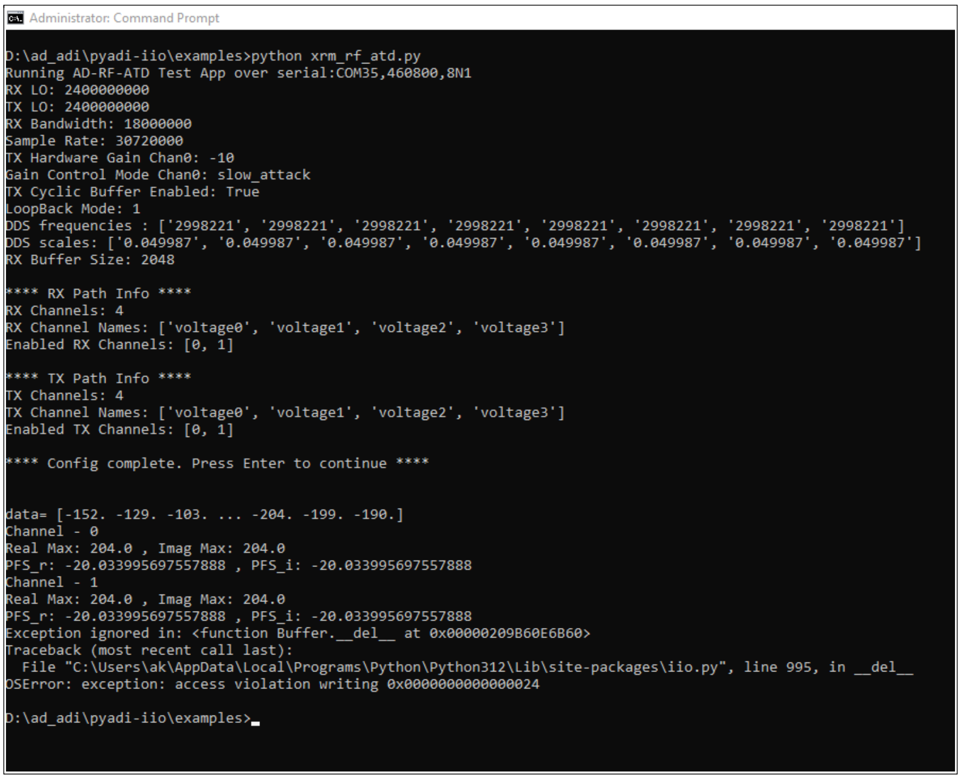
pyadi-iio is python scripts that are using libiio to communicate with the IIO devices

> **Note**
> More informations about pyadi-iio you can find on Analog Devices Github:**https://github.com/analogdevicesinc/pyadi-iio**.

Python should be installed before running the example test.

1. To run the test example files, open cmd and cd to the examples folders e.g:
   "Drive:\> cd ad_adi_7z1_'version'\pyadi-iio\examples"
2. Run the python file example test e.g:
   For no:OS: "Drive:\ad_adi_7z1_'version'\pyadi-iio\examples>python xrm_rf_atd.py"
   For Linux: "Drive:\ad_adi_7z1_'version'\pyadi-iio\examples>python xrm_rf_atd_linux.py"

Here are images from the example test run on Alpha-Data's AD01253-ADM-XRC-7Z1 fpga board and AD01289-XRM2-RF-ATD IO adapter board that has Analog Devices AD9361 IC:



```
Administrator: Command Prompt

D:\ad_adi\pyadi-iio\examples>python xrm_rf_atd.py
Running AD-RF-ATD Test App over serial:COM35,460800,8N1
RX LO: 2400000000
TX LO: 2400000000
RX Bandwidth: 18000000
Sample Rate: 30720000
TX Hardware Gain Chan0: -10
Gain Control Mode Chan0: slow_attack
TX Cyclic Buffer Enabled: True
LoopBack Mode: 1
DDS frequencies : ['2998221', '2998221', '2998221', '2998221', '2998221', '2998221', '2998221', '2998221']
DDS scales: ['0.049987', '0.049987', '0.049987', '0.049987', '0.049987', '0.049987', '0.049987', '0.049987']
RX Buffer Size: 2048

**** RX Path Info ****
RX Channels: 4
RX Channel Names: ['voltage0', 'voltage1', 'voltage2', 'voltage3']
Enabled RX Channels: [0, 1]

**** TX Path Info ****
TX Channels: 4
TX Channel Names: ['voltage0', 'voltage1', 'voltage2', 'voltage3']
Enabled TX Channels: [0, 1]

**** Config complete. Press Enter to continue ****


data= [-152. -129. -103. ... -204. -199. -190.]
Channel - 0
Real Max: 204.0 , Imag Max: 204.0
PFS_r: -20.033995697557888 , PFS_i: -20.033995697557888
Channel - 1
Real Max: 204.0 , Imag Max: 204.0
PFS_r: -20.033995697557888 , PFS_i: -20.033995697557888
Exception ignored in: <function Buffer.__del__ at 0x00000209B60E6B60>
Traceback (most recent call last):
  File "C:\Users\ak\AppData\Local\Programs\Python\Python312\Lib\site-packages\iio.py", line 995, in __del__
OSError: exception: access violation writing 0x0000000000000024

D:\ad_adi\pyadi-iio\examples>_
```

Figure 5 : ADM-XRC-7Z1:XRM2-RF-ATD TX-RX loopback test, cmd outputs

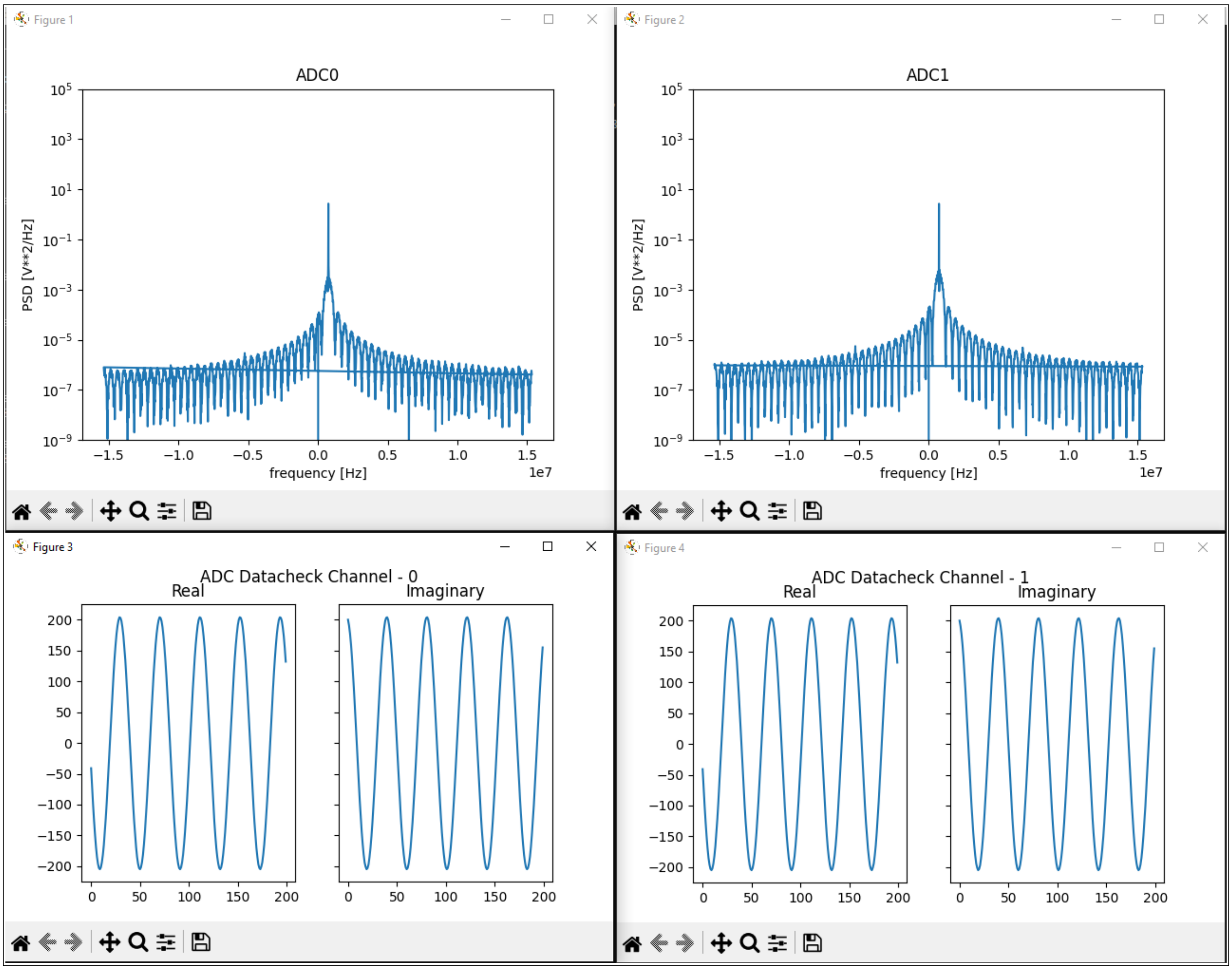


Figure 6 : ADM-XRC-7Z1:XRM2-RF-ATD TX-RX loopback test, frequency and time plots

iio-oscilloscope

iio-oscilloscope is a GTK+ application that allows to communicate with the IIO devices

> **Note**
> More informations about iio-oscilloscope you can find on Analog Devices Github:**https://github.com/analogdevicesinc/iio-oscilloscope**.

iio-oscilloscope application should be installed before running the example test.

1. To run the test example ini files, run iio-oscilloscope application and go to File->>Load/Save Profile
2. After go to ad_adi_7z1_'version'\iio-oscilloscope\profiles and find the test e.g no-OS: "xrm_rf_atd.ini" or Linux: "xrm_rf_atd_linux.ini"
3. Connect to libiio context after you load the profile ini file
4. Follow any messages that are prompt from the example test

Here are images from the example test run on Alpha-Data's AD01253-ADM-XRC-7Z1 fpga board and AD01289-XRM2-RF-ATD IO adapter board that has Analog Devices AD9361 IC:

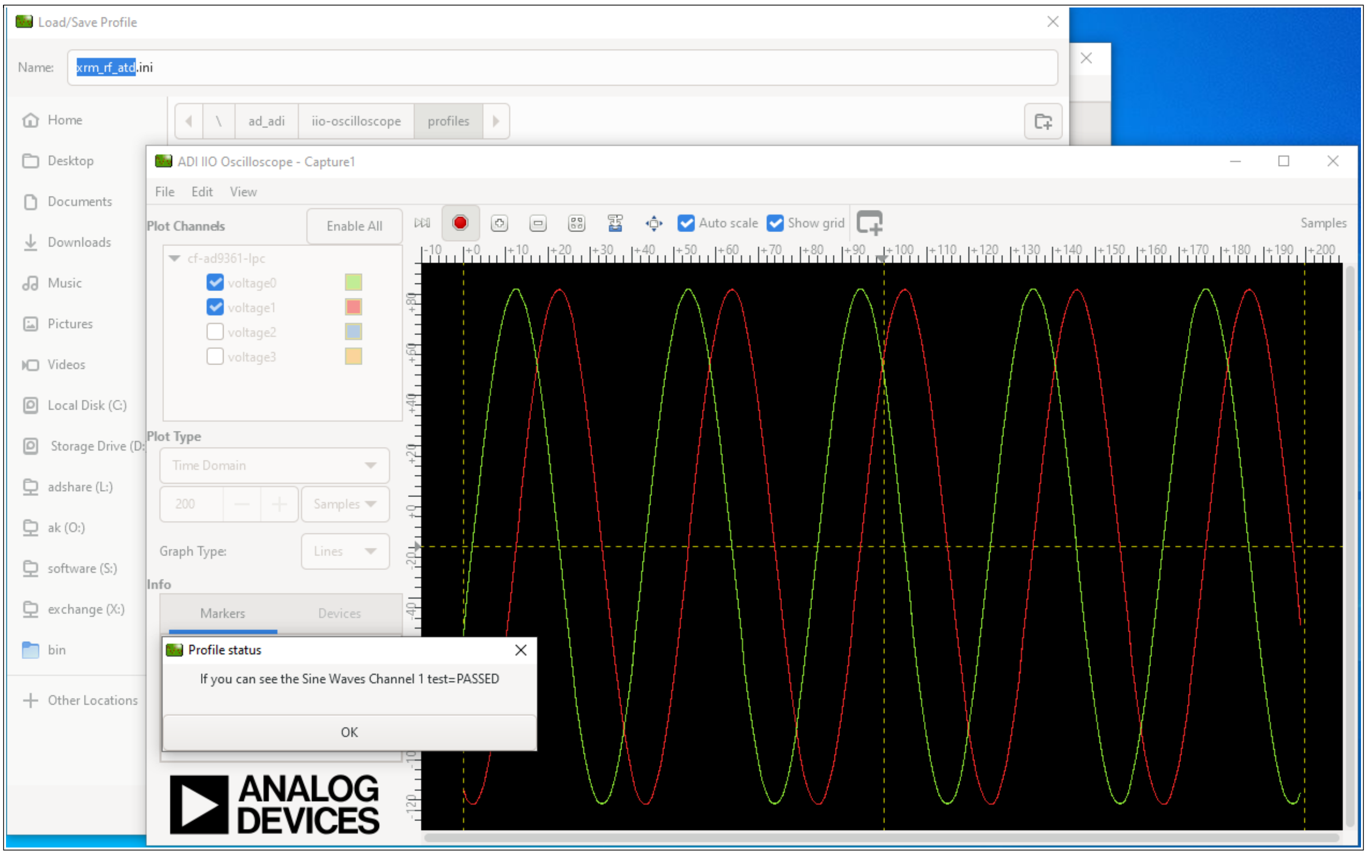


Figure 7 : AADM-XRC-7Z1:XRM2-RF-ATD TX1-RX1 cable loopback test, time plots

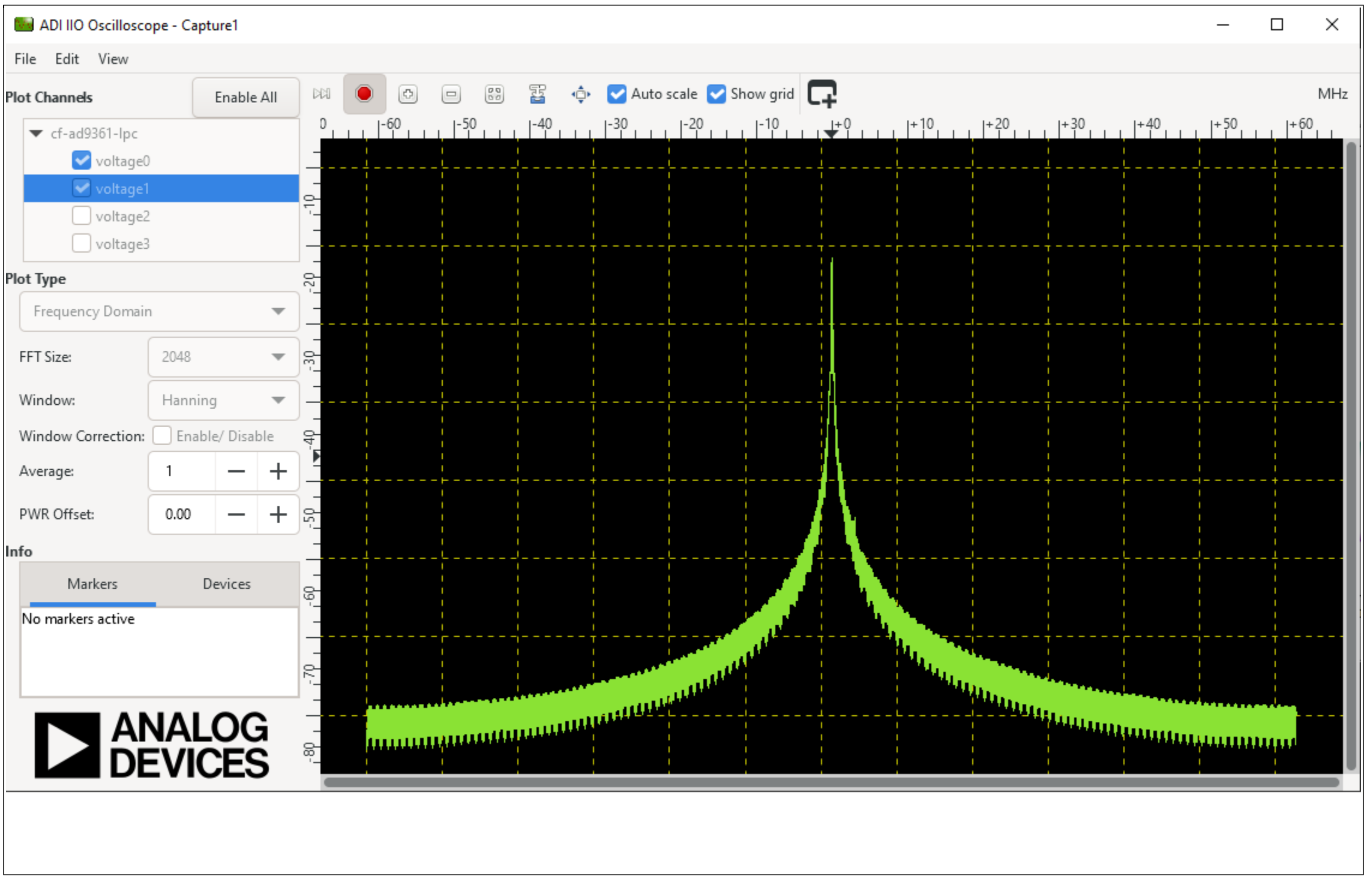


Figure 8 : ADM-XRC-7Z1:XRM2-RF-ATD TX-RX cable loopback test, frequency plots

Spectrum Analyzer Test

Here is the image of the board set up that includes the ADM-XRC-7Z1 fpga board, XMR2-RF-AT IO adaptor board and the ADC-XMC-BREAKOUT carrier board.

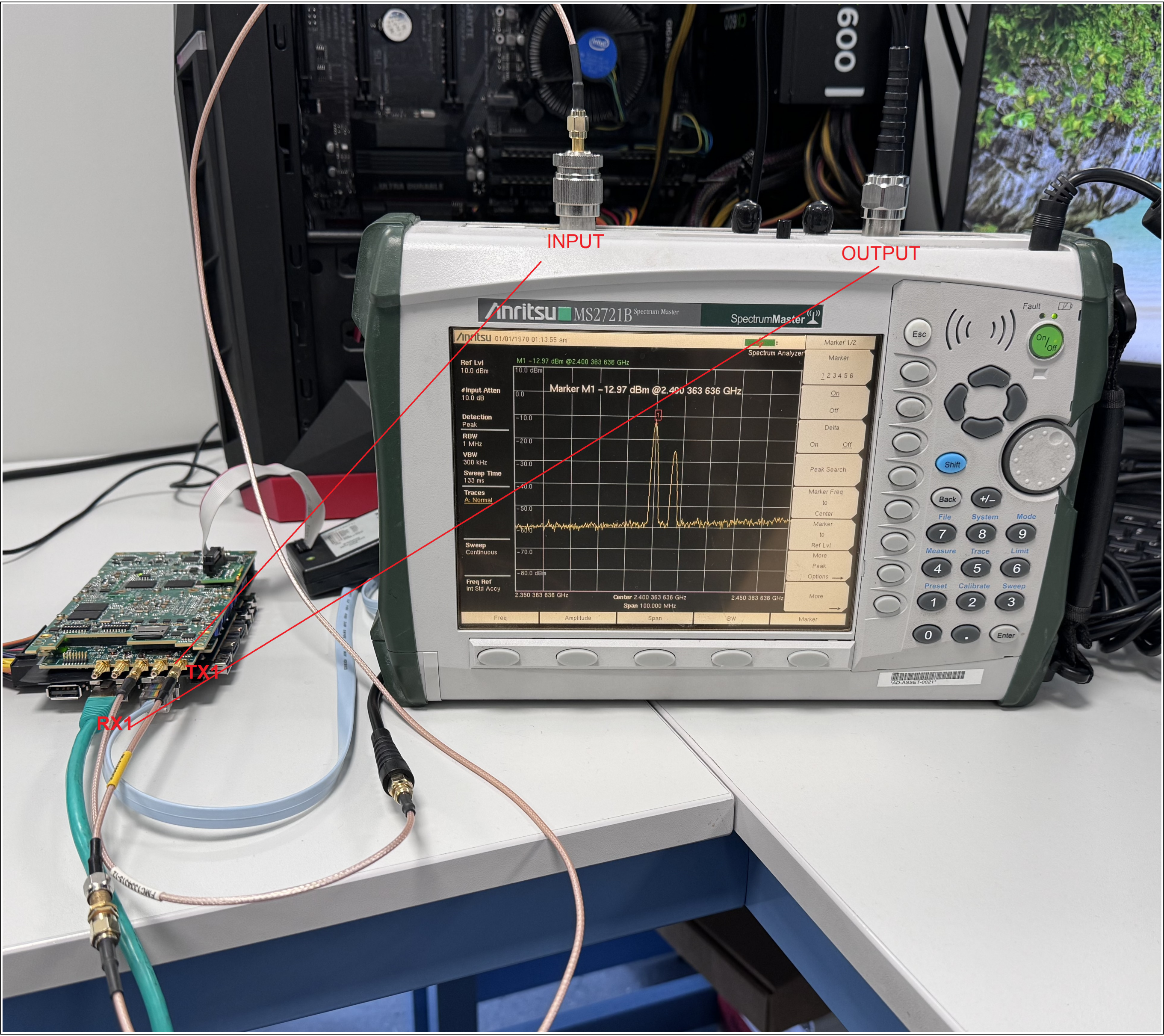


Figure 9 : Board Set Up

TX Test

XRM2-RF-ATD TX1 connector is connected to RF In 50 Ohms of the spectrum analyzer.

The properties of this test are:
>>> # Read properties
RX LO: 2400000000
TX LO: 2400000000
RX Bandwidth: 18000000
Sample Rate: 30720000
TX Hardware Gain Chan0: -10
Gain Control Mode Chan0: slow_attack
TX Cyclic Buffer Enabled: True
LoopBack Mode: 0
DDS frequencies : ['999407', '29999090', '999407', '29999090', '2998221', '2998221', '2998221', '2998221']
DDS scales: ['0.500000', '0.125000', '0.500000', '0.125000', '0.000000', '0.000000', '0.000000', '0.000000']
RX Buffer Size: 2048

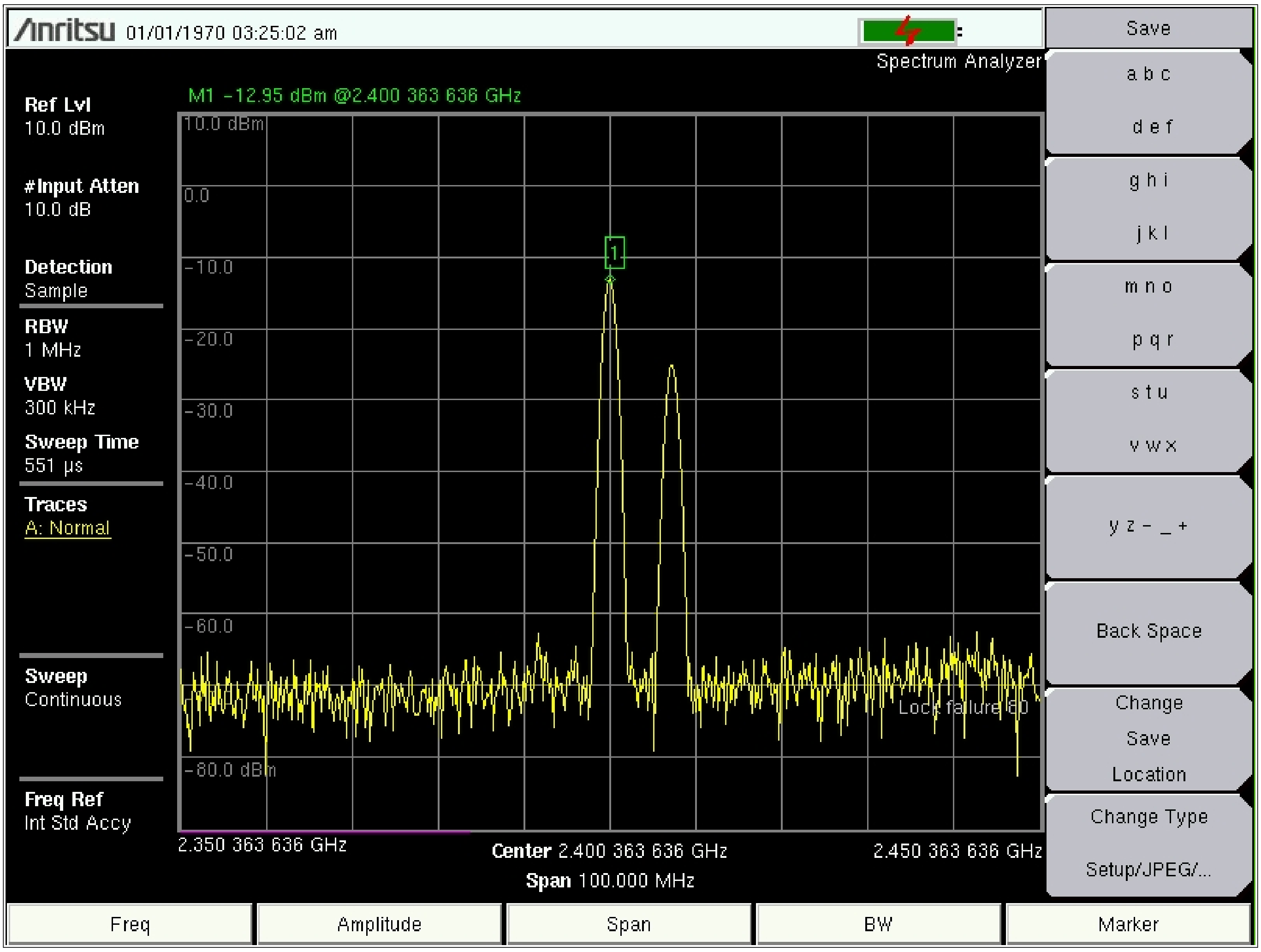


Figure 10 : 1MHz and 30MHz Tones, LO 2.4GHz Test

RX Test

XRM2-RF-ATD RX1 connector is connected to Gen Output 50 Ohms and XRM2-RF-ATD TX1 connector is connected to RF In 50 Ohms of the spectrum analyzer.

For this Test is selected a loopback mode that is passing RX1 to TX1 internally in the FPGA.

The properties for test 1 are:
>>> # Read properties
RX LO: 2400000000
TX LO: 2400000000
RX Bandwidth: 18000000
Sample Rate: 30720000
TX Hardware Gain Chan0: -10
Gain Control Mode Chan0: slow_attack
TX Cyclic Buffer Enabled: True
LoopBack Mode: 2 "Internally loopback in the FPGA RX-->TX"

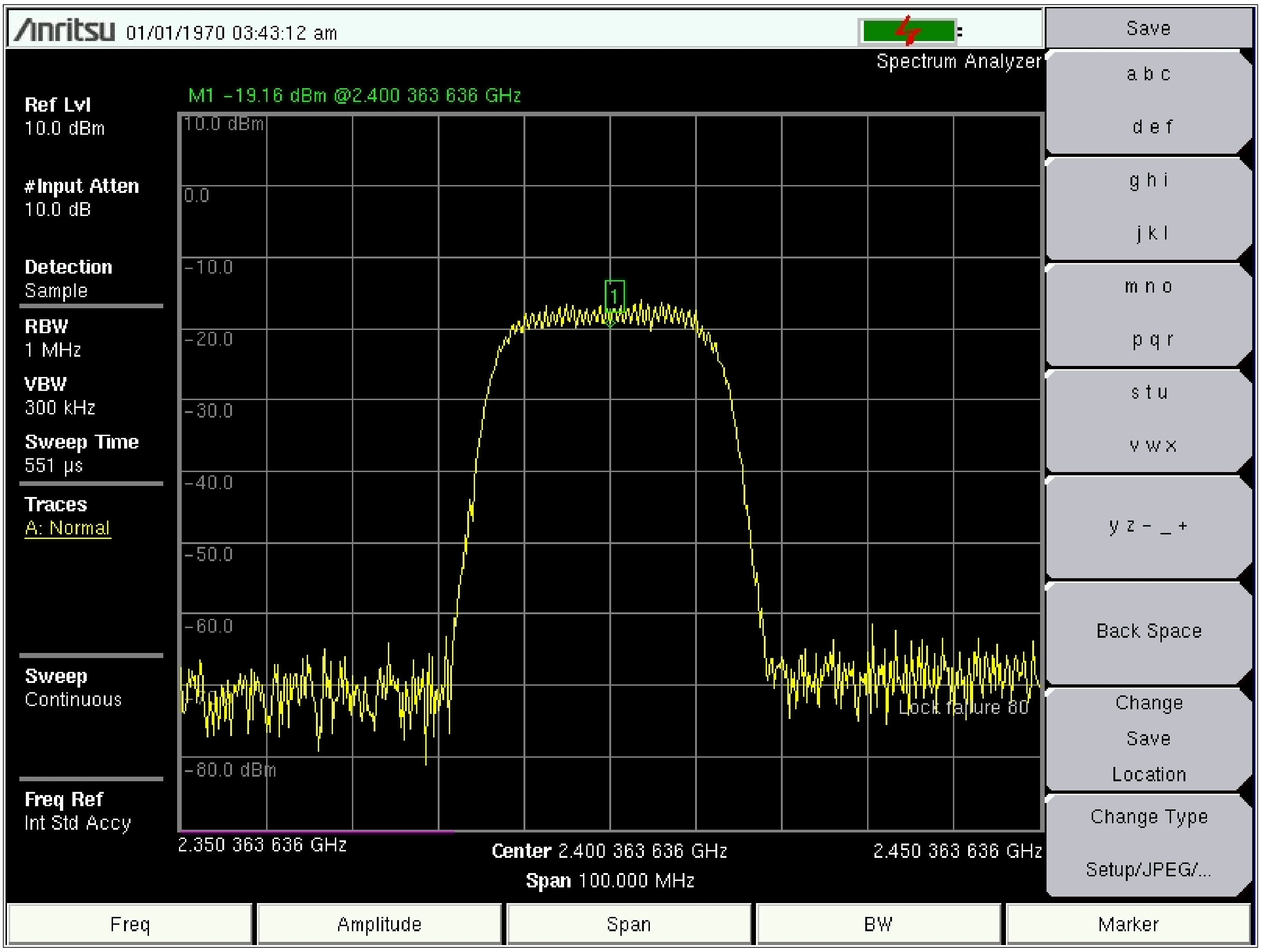


Figure 11 : RX1-->TX1 loopback, LO 2.4GHz, 18MHz Bandwidth

The properties for test 2 are:
>>> # Read properties
RX LO: 2400000000
TX LO: 2400000000
RX Bandwidth: 5000000
Sample Rate: 30720000
TX Hardware Gain Chan0: -10
Gain Control Mode Chan0: slow_attack
TX Cyclic Buffer Enabled: True
LoopBack Mode: 2 "Internally loopback in the FPGA RX-->TX"

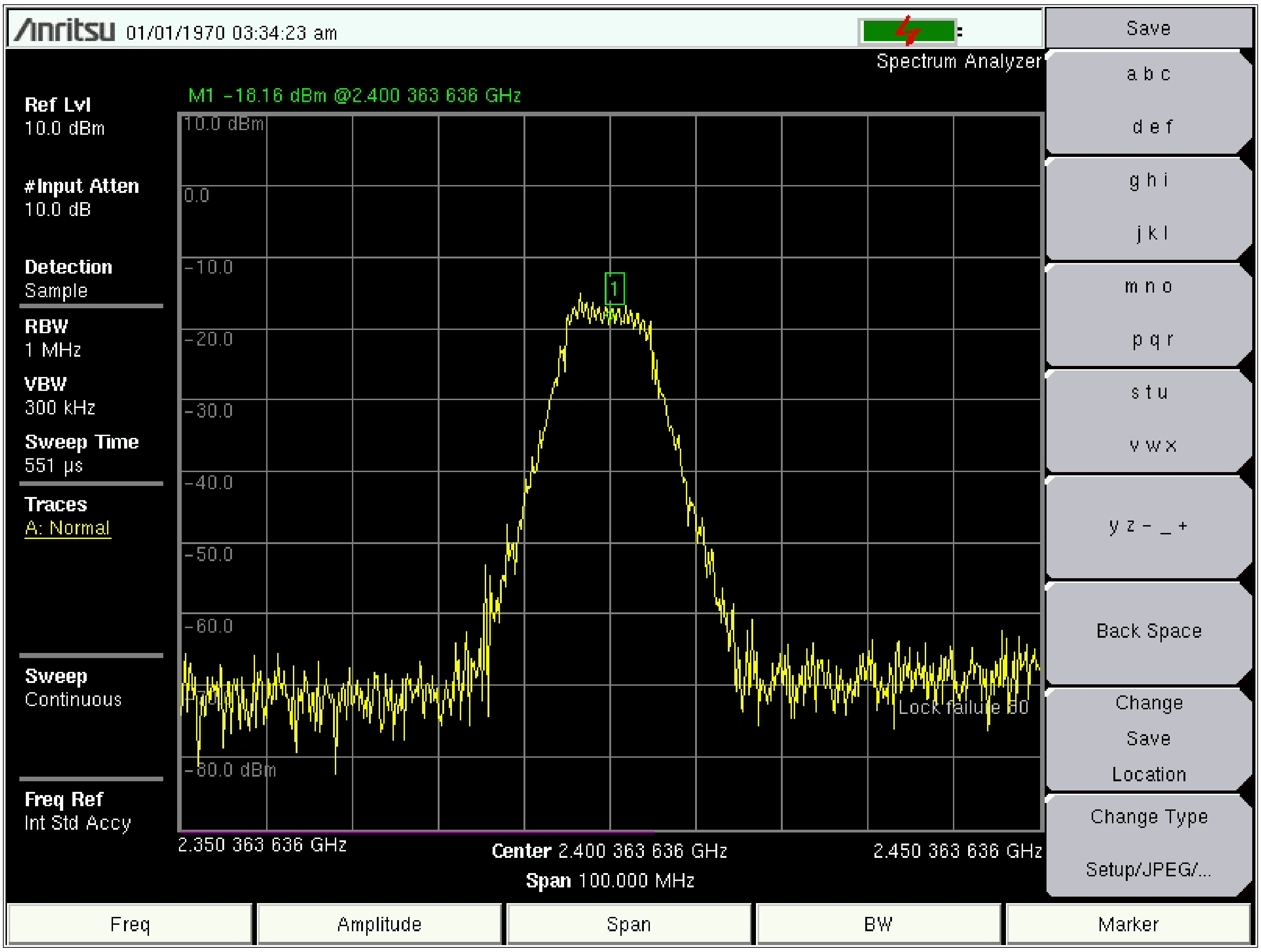


Figure 12 : RX1-->TX1 loopback, LO 2.4GHz, 5MHz Bandwidth

Revision History

Date	Revision	Nature of change
16th May 2024	1.0	Initial release.

Address: Suite L4A, 160 Dundee Street,
Edinburgh, EH11 1DQ, UK
Telephone: +44 131 558 2600
Fax: +44 131 558 2700
email: sales@alpha-data.com
website: http://www.alpha-data.com

Address: 10822 West Toller Drive, Suite 250
Littleton, CO 80127
Telephone: (303) 954 8768
Fax: (866) 820 9956 - toll free
email: sales@alpha-data.com
website: http://www.alpha-data.com