FMC5030 SERIES USER MANUAL

70 MHz to 18 GHz FMC Transceiver

Amtery corporation

2023 December

Contents

1. Introduction

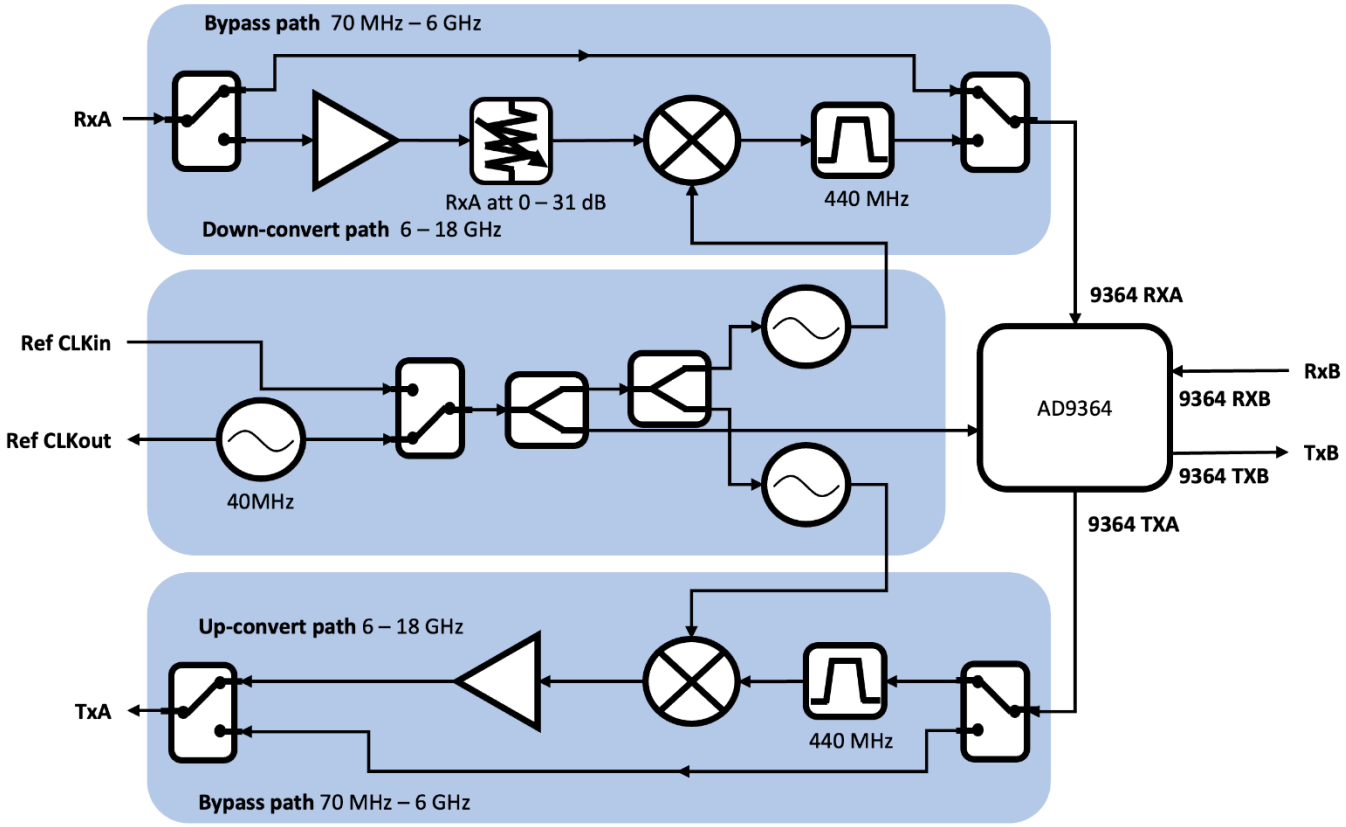
1.1. Overview

Amtery FMC5030 is a 70 MHz – 18 GHz transceiver FMC board. It is equipped with:

- Analog Devices AD9364 transceiver, which covers 70 MHz – 6 GHz.
- Upconverter and downconverter extending the frequency to cover 6 – 18 GHz.
- Internal or external reference clock.

The block diagram is shown below.

Figure 1. Block diagram



1.2. Compatible FPGA Carrier Board

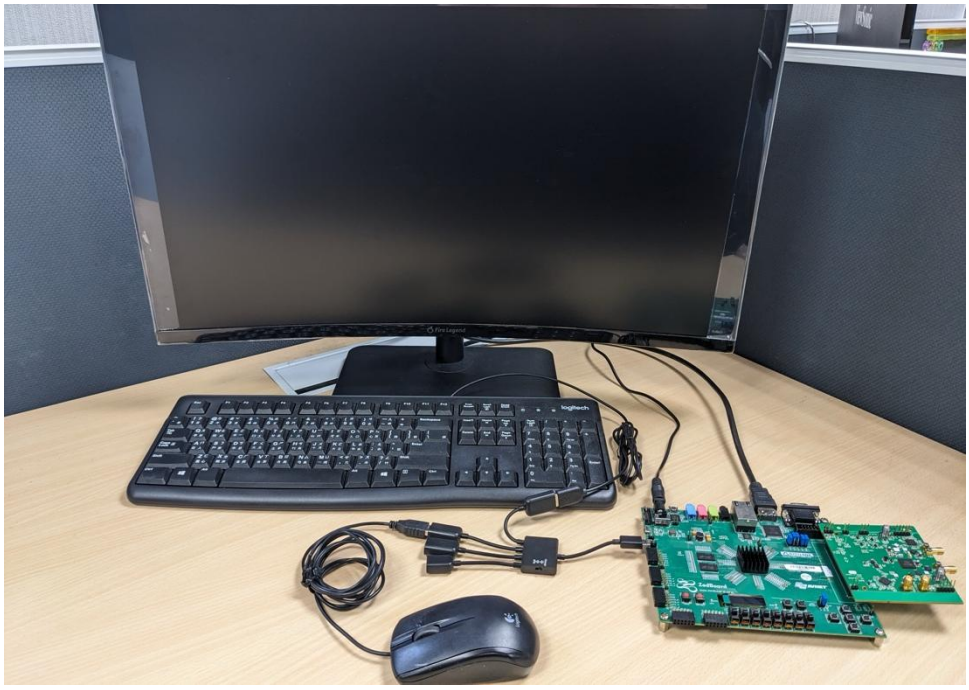
The boot file in SD card supports Digilent ZedBoard. Please contact us if another FPGA carrier board's boot file is required.

1.3. Shipped Content

a. Transceiver FMC board
b. SD card with:
   - Analog Devices Kuiper Linux
   - Python and LabVIEW examples

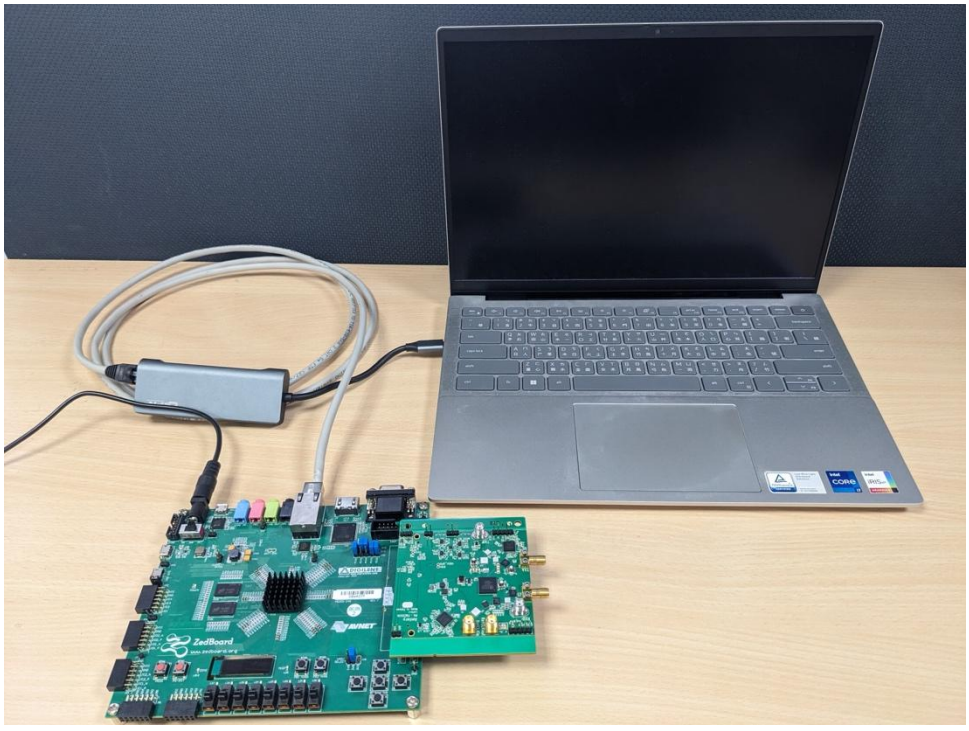
1.4. Installation and Getting Started

Use Zedboard as host to control FMC5030



a. Connect keyboard, mouse, monitor and power supply to ZedBoard.
b. Insert SD card into ZedBoard and switch on power.
c. In Kuiper Linux, execute “Application menu → Other → ADI IIO Oscilloscope” to control AD9364 onboard.
d. To control RF front end, execute “fmc5030 rst” to reset FMC5030.
e. Refer to 2.1 for detail usage.

Use external host (PC or notebook) to control Zedboard and FMC5030



a. Download and install LibIIO: Go to analogdevicesinc/libiio or Amtery website.

   Windows: Download and execute “libiio-0.25.gb6028fd-setup.exe”.

   Linux: Download and install “libiio-0.25-gb6028fd-(corresponding OS).deb (or .rpm)” according to the operating system. Some necessary packages may need to be installed to successfully install LibIIO. If none of the versions suit the OS, follow the build instruction to build LibIIO.

b. Install Python 3.6 or newer versions.

c. Install the amtFMC5030 through pip or pip3. This will install the necessary libraries and examples. After installation, the examples are in …\site-packages\amtFMC5030\examples. Or download from Amtery website or copy from SD card. To run the examples, packages matplotlib, scipy, and thread6 are required.

```
pip install amtFMC5030
```

d. Install “openSSH server”.

   Windows: Use commands below on PowerShell as administrator. Refer to Microsoft’s page for detail.

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0

Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

Start-Service sshd

Set-Service -Name sshd -StartupType 'Automatic'

if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction
SilentlyContinue | Select-Object Name, Enabled)) { Write-Output
"Firewall Rule 'OpenSSH-Server-In-TCP' does not exist, creating it..."
New-NetFirewallRule -Name 'OpenSSH-Server-In-TCP' -DisplayName 'OpenSSH
Server (sshd)' -Enabled True -Direction Inbound -Protocol TCP -Action
Allow -LocalPort 22 } else { Write-Output "Firewall rule 'OpenSSH-
Server-In-TCP' has been created and exists." }
```

   Linux: Install and start openssh-server:

```
sudo dnf/apt install openssh-server

sudo systemctl start ssh

service ssh status #check ssh status
```

e. Make sure the ethernet connection between Zedboard and PC.

f. To control RF front end, execute “amtFmcRfReset()“ to remotely reset FMC5030.

g. Refer to 2.2 for detail usage.

2. Use FMC5030

2.1. Control FMC5030 by Commands in ZedBoard Kuiper Terminal

FMC5030 can be controlled in Analog Devices Linux Kuiper terminal by using commands. The software architecture is in Figure 2. By this method, the tuner and AD9364 are controlled separately. The terminal commands control the tuner and the IIO's RX LO and TX LO frequency. The IIO controls the AD9364. When the RF frequency is over 6 GHz, the IIO's RX LO or TX LO freuquency will be set to corresponding intermediate frequency, 440 MHz. Refer to Table 1 for the commands list.

Figure 2 Software architecture of controlling FMC5030 by terminal commands in Kuiper

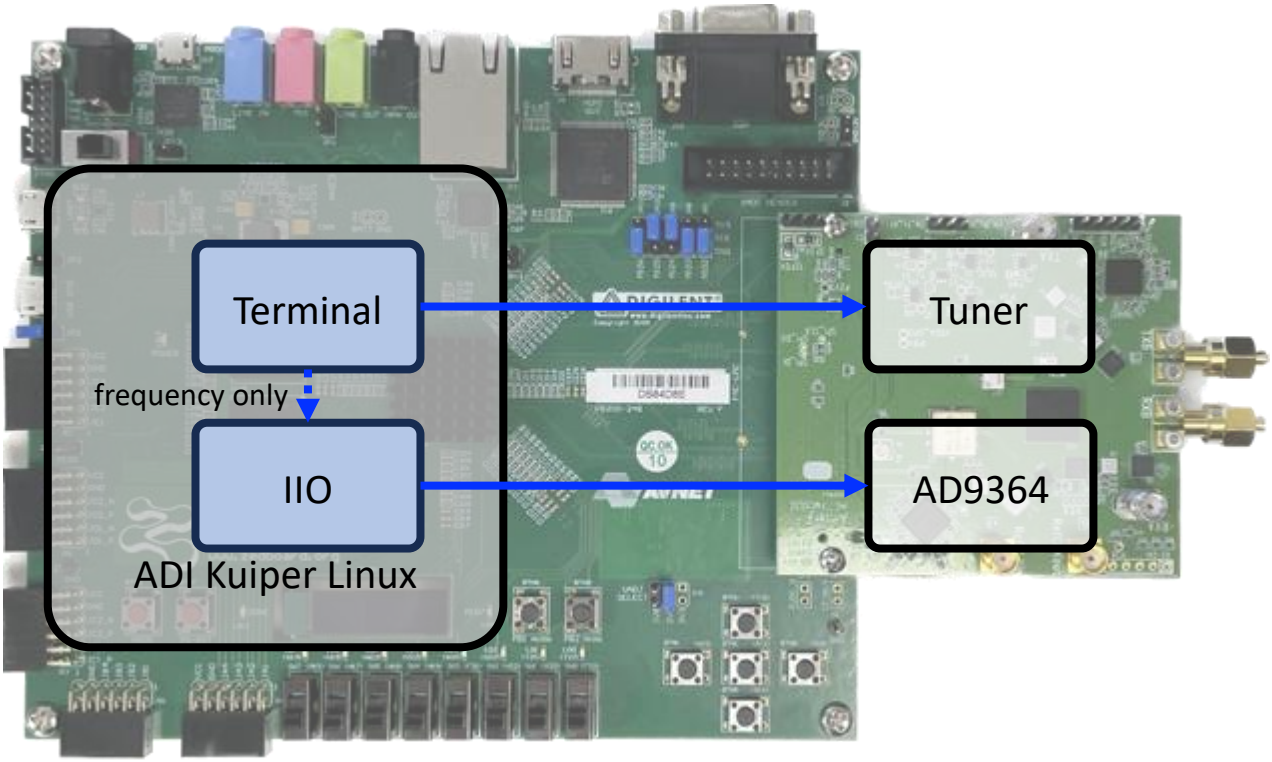


Table 1 Terminal Commands

These commands are case-insensitive.

Command	**Description**	**Parameters**	**Example**
fmc5030	Show help for these commands.		
fmc5030 rst	Reset and initialize FMC5030. The default settings are: Frequency: 9000 MHz RxA att: 0		fmc5030 rst (Reset FMC5030.)
fmc5030 freq [T/R] [frequency]	Set the TxA or RxA frequency. AD9364's ports will be set to corresponding IF frequency. (TxB and RxB are directly connected to AD9364 and controlled in IIO.)	[port] TR: Set both TxA and RxA. T: Set TxA, ignore RxA. R: Set RxA, ignore TxA. [frequency] Frequency in MHz, 1 Hz resolution.	fmc5030 freq T 10100 (Set TxA frequency to 10.1 GHz.)
fmc5030 att [value]	Set RxA attenuation.	[value] Attenuation in dB. 1 dB resolution.	fmc5030 att -15 (Set RxA attenuation to -15 dB.)
fmc5030 fv	Display firmware version.		fmc5030 fv

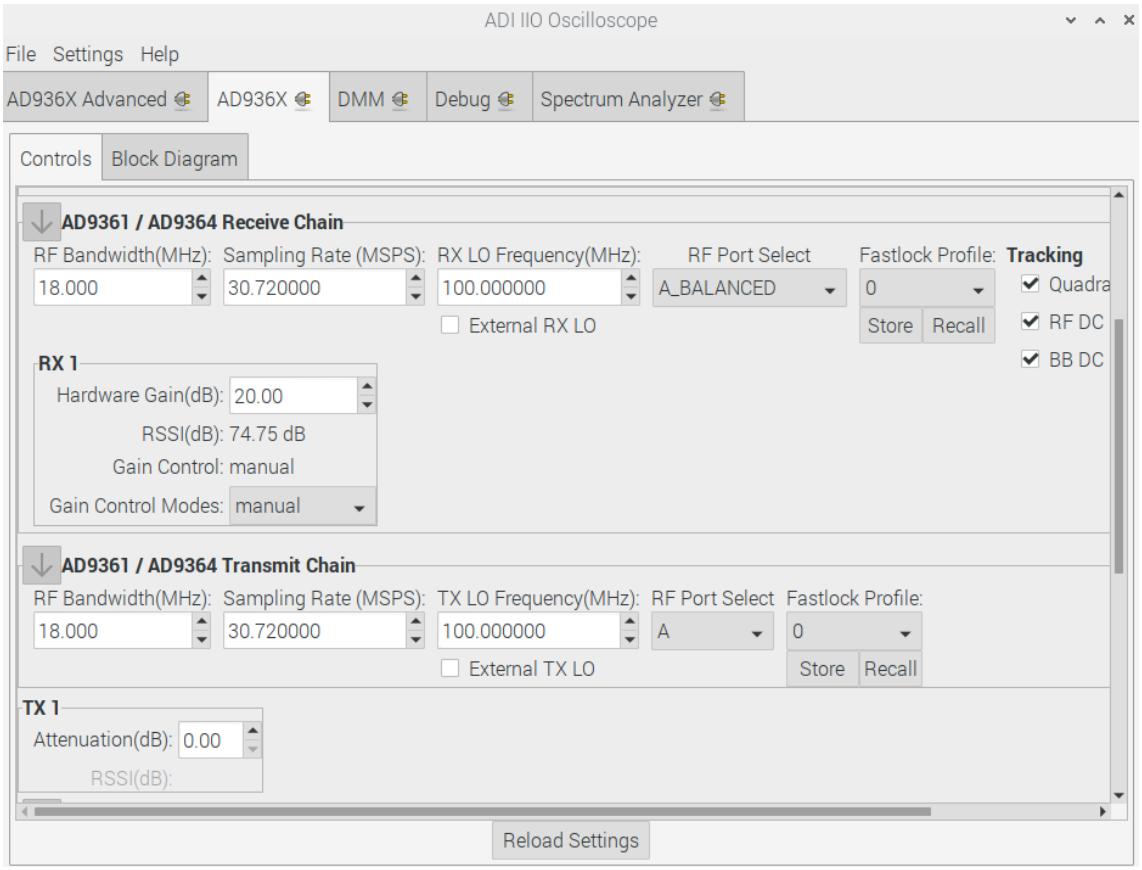
Terminal commands examples:

a. Acquire a signal at RxA, 0.1 GHz.

Terminal:

```
analog@analog:~/Public $ fmc5030 rst
Initialize channels.
Reset FMC5030. TxA and RxA frequency = 9000 MHz. RxA attenuation = 0 dB.
analog@analog:~/Public $ fmc5030 freq R 100
Use RxA bypass path.
```

IIO Oscilloscope: The RX LO Frequency(MHz) is set to 100 by terminal command. Press “Reload Settings” to update the GUI.

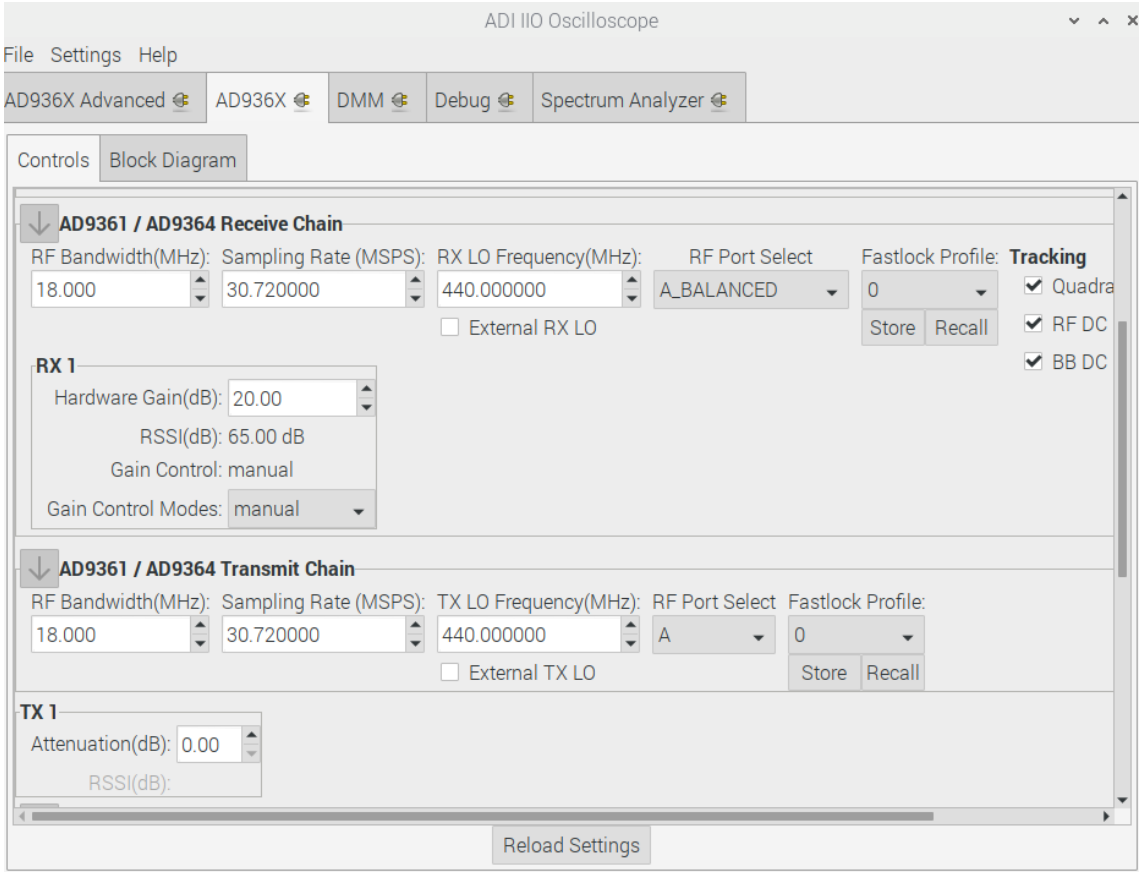


b. Acquire a signal at RxA, 12 GHz with attenuation -10 dB.

Terminal:

```
analog@analog:~/Public $ fmc5030 rst
Initialize channels.
Reset FMC5030. TxA and RxA frequency = 9000 MHz. RxA attenuation = 0 dB.
analog@analog:~/Public $ fmc5030 freq R 12000
Use RxA convert path.
RxA frequency = 12000 MHz.
analog@analog:~/Public $ fmc5030 att -10
RxA attenuation = -10 dB.
```

IIO Oscilloscope: The RX LO Frequency(MHz) is set to the corresponding intermediate frequency, 440, by terminal command. Press “Reload Settings” to update the GUI.

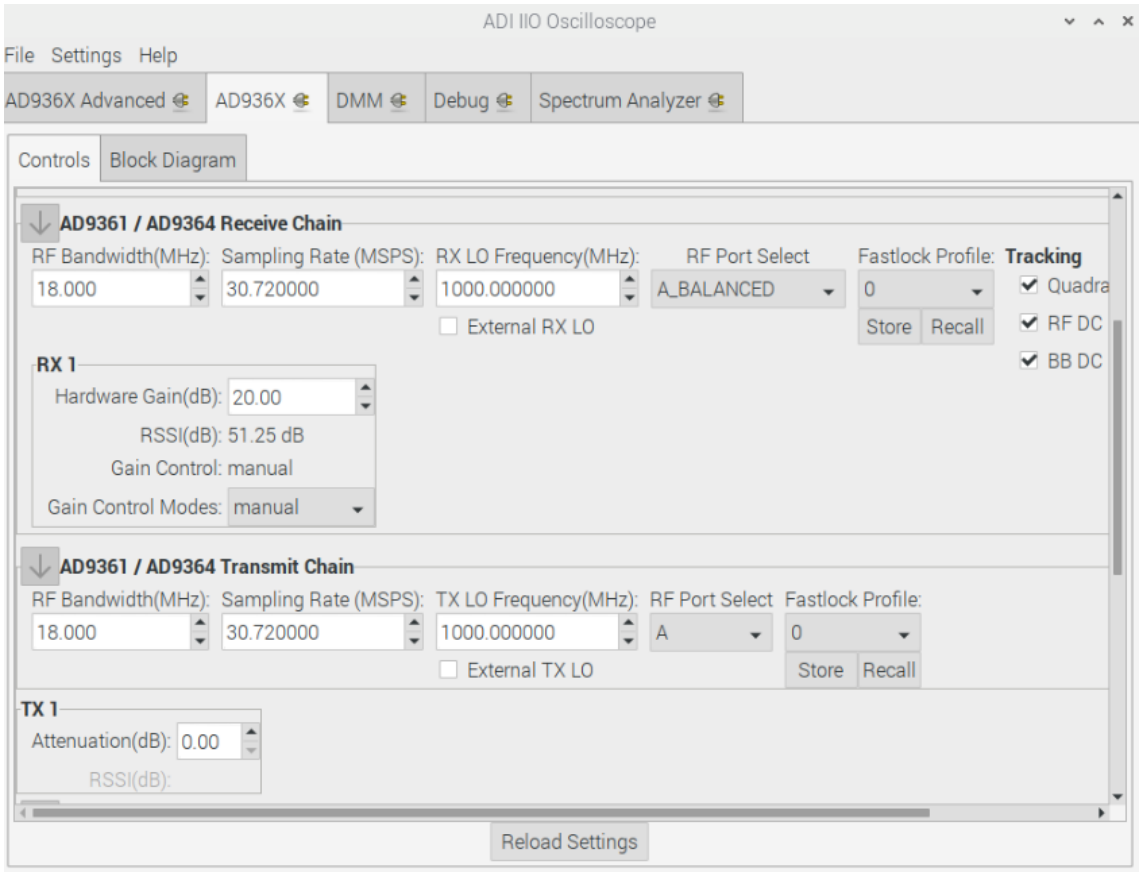


c. Transmit a signal at TxA, 1 GHz

Terminal:

```
analog@analog:~/Public $ fmc5030 rst
Initialize channels.
Reset FMC5030. TxA and RxA frequency = 9000 MHz. RxA attenuation = 0 dB.
analog@analog:~/Public $ fmc5030 freq T 1000
Use TxA bypass path.
```

IIO Oscilloscope: The TX LO Frequency(MHz) is set to 1000 by terminal command. Press “Reload Settings” to update the GUI.

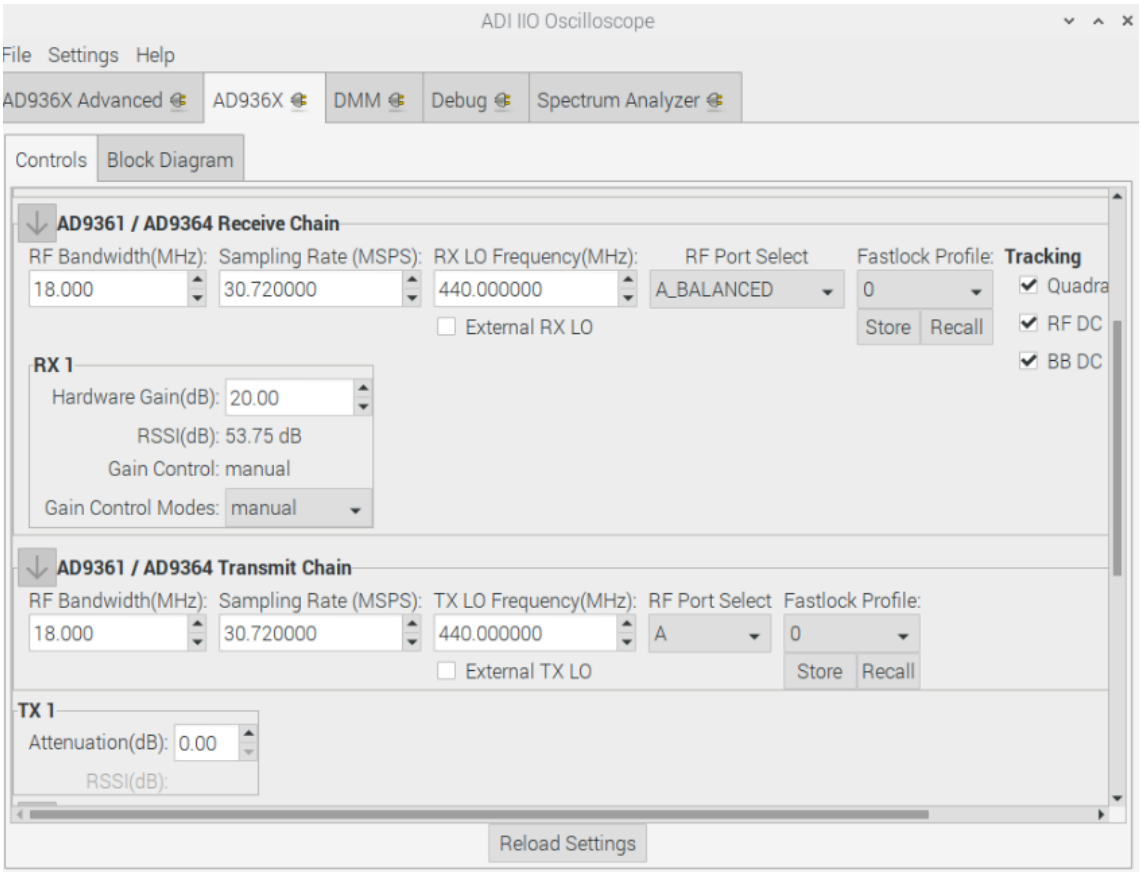


d. Transmit a signal at TxA, 18 GHz

Terminal:

```
analog@analog:~/Public $ fmc5030 rst
Initialize channels.
Reset FMC5030. TxA and RxA frequency = 9000 MHz. RxA attenuation = 0 dB.
analog@analog:~/Public $ fmc5030 freq T 18000
Use TxA convert path.
TxA frequency = 18000 MHz.
```

IIO Oscilloscope: The TX LO Frequency(MHz) is set to the corresponding intermediate frequency, 440, by terminal command. Press “Reload Settings” to update the GUI.



2.2. Remotely Control ZedBoard and FMC5030 From External Host

Python libraries are provided to control FMC5030. The program is executed in an external host computer. By this method, python or LabVIEW programs can control the tuner and AD9364 simultaneously. Refer to 1.4 for examples positions.

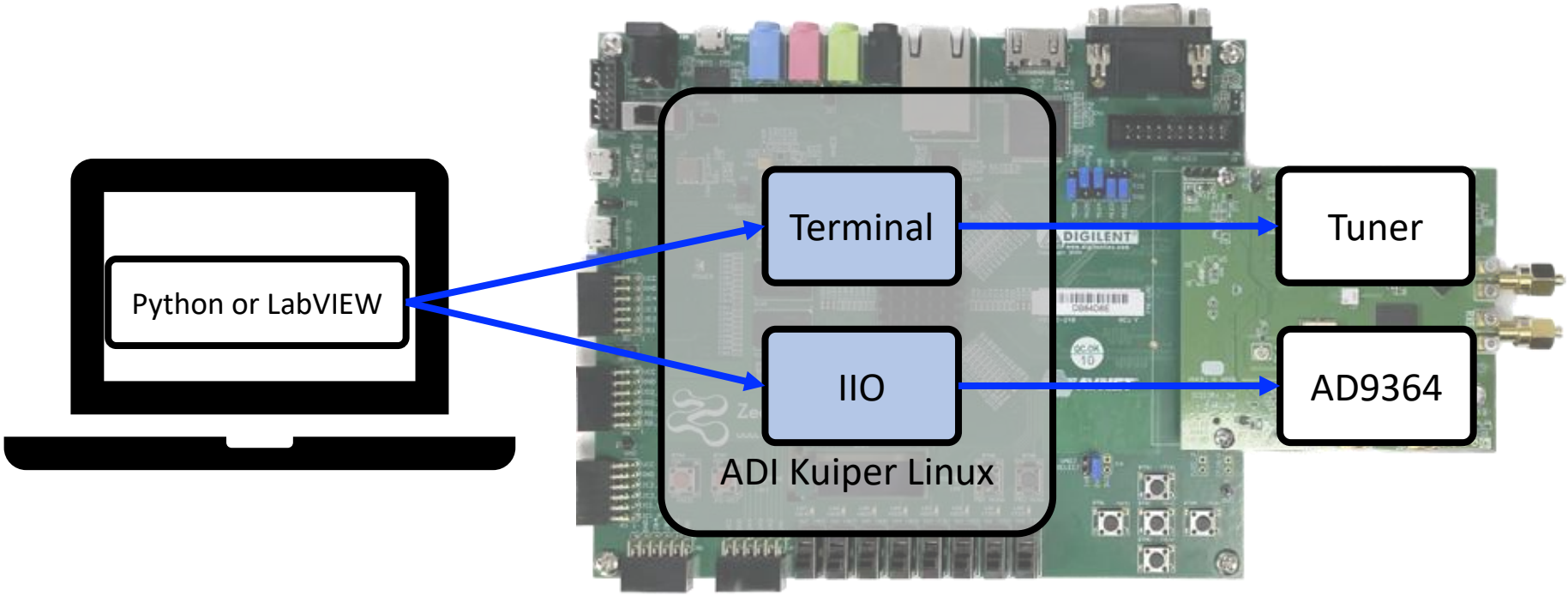


Table 2 Python and LabVIEW API

Function	Description	Parameters
amtFmcRfReset(ip: str, port: int, username: str, password: str) -> reference: list	Reset and initialize FMC5030. Return a reference, which denotes the connection between host and device. The default settings are: Frequency: 9000 MHz RxA att: 0 dB	ip: ZedBoard's IP. Defualt: 169.254.92.202 port: Zedboard's port for SSH. Defualt: 22 username: Zedboard's username. Defualt: analog password: Zedboard's password to the user. Defualt: analog reference: The connection between host and device.

def amtFmcTxConfig(rfPort: str, frequency: int, rate: int, bw: int, txAtt: int, cyclic: bool)	Config Tx parameters for transmission.	rfPort: Select transmission port, A or B. TxA's frequency range is 70 MHz to 18 GHz, TxB is 70 MHz to 6 GHz. frequency: Frequency in MHz. rate (optional): IQ rate in MSPS. Default value: 30.72 MSPS. AD9364's Tx and Rx share the same IQ rate. This value sets Rx rate simultaneously. bw (optional): Set AD9364's internal analog filter bandwith in MHz. Default value: 18 MHz. txAtt (optional): AD9364's Tx attenuation in dB. Range: 0 to -89 dB. Default value: 0 dB. cyclic (optional): True or false. If true, the data in buffer will be transmitted repeatedly. If false, the data in buffer will be transmitted only once.
amtFmcRfTxStart (data: complex)	Start transmitting data.	data: Transmitted data.
amtFmcRfTxStop ()	Stop transmitting.	
amtFmcRxConfig(rfPort: str, frequency: int, rate: int, bw: int, numOfSamples: int, rxAAtt: int, rxGain: int)	Config Rx parameters for receiving. Currently FMC5030 Rx doesn't support cyclic buffer. There are data gaps between each section of receiving data.	rfPort: Select receiving port, A or B. RxA's frequency range is 70 MHz to 18 GHz, RxB is 70 MHz to 6 GHz. frequency: Frequency in MHz. rate (optional): IQ rate in MSPS. Default value: 30.72 MSPS. AD9364's Tx and Rx share the same IQ rate. This value sets Tx rate simultaneously. bw (optional): Set AD9364's internal analog filter bandwith in MHz. Default value: 18 MHz. numOfSamples (optional): Number of IQ samples. Default value: 16384. rxAAtt (optional): RxA attenuation value in dB. Range: 0 to -31 dB. Default value: 0 dB. rxGain (optional): AD9364's Rx gain in dB. Range: 0 to 70 dB. Default value: 0 dB.
amtFmcRfRxRead () -> data: float	Receive one section of data. Currently FMC5030 Rx doesn't support	data: Received data. Row 0 stores the real part, and row 1 stores the imaginary part.

	cyclic buffer. There are data gaps between each section of receiving data.	
amtFmcRfFv ()	Display FMC5030 firmware version.	
amtFmcRfRef (ip: str, refClk: str, clkFreq: float)	Set internal or external reference clock and frequency. Refer to 2.3 for detail usage.	ip: Zedboard's IP. refClk: Select internal or external reference clock. "int" to use internal reference clock and "ext" to use external reference clock. clkFreq: External reference clock frequency. Range: 10 to 100 MHz. Ignored if internal reference clock is used.

2.3. Use external or internal reference clock

Once the reference clock is set to external or internal, it sustains until further change. Please note that the reference clock can't be set in programs dynamically due to reboot is necessary.

2.3.1. Use external reference clock

The external reference clock can be sine or square wave with frequency range 10 MHz to 100 MHz. It can be set in Kupier Linux terminal or by Python command in external PC.

Zedboard Kupier Linux Terminal

1. Execute "amtFmcRfRef ext (clock frequency in MHz)" as root.

```
analog@analog:~ $ su
Password:
root@analog:/home/analog# cd Public
root@analog:/home/analog/Public# amtFmcRfRef ext 10
```

Output:

```
Rebuild devicetree...
CURRENT BOARD CONFIG:
DONE
Reference clock = external. Clock frequency = 10 MHz.
```

2. Reboot the device.

Python in external PC

1. Use function "amtFmcRfRef (ip,refClk,clkFreq)".

```
# Zedboard's IP
ip = "192.168.1.193"
# Input "int" to use internal clock and "ext" to use external reference clock.
refClk = "ext"
# clkFreq: Reference clock frequency in MHz, range: 10 to 100 MHz.
clkFreq = 10
# Note that 10 MHz reference clock is required to be a sinusoidal wave.
fmc.amtFmcRfRef(ip,refClk,clkFreq)
```

Output:

```
Linux analog 5.10.0-98248-g1bbe32fa5182 #1142 SMP PREEMPT Wed Aug 3

The programs included with the Debian GNU/Linux system are free soft
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 23 08:45:44 2023 from 192.168.1.86
su
analog@analog:~$ su
Password:

root@analog:/home/analog#
cd /home/analog/Public
root@analog:/home/analog/Public#
./amtFmcRfRef.sh ext 10
Rebuild devicetree...
CURRENT BOARD CONFIG:
DONE
Reference clock = external. Clock frequency = 10 MHz.
```

2. Reboot the device.

2.3.2. Use internal reference clock

The internal reference clock can also be set in Kupier Linux terminal or by Python command in external PC.

Zedboard Kupier Linux Terminal

1. Execute “amtFmcRfRef int” as root.

```
analog@analog:~ $ su
Password:
root@analog:/home/analog# amtFmcRfRef int
```

Output:

```
Rebuild devicetree...
CURRENT BOARD CONFIG:
DONE
Reference clock = internal. Clock frequency = 80 MHz.
```

2. Reboot the device.

Python in external host

1. Use function “amtFmcRfRef (ip,refClk,clkFreq)”. Note that clkFreq is ignored.

```
# Zedboard's IP
ip = "192.168.1.193"
# Input “int” to use internal clock and “ext” to use external reference clock.
refClk = "int"
# Note that 10 MHz reference clock is required to be a sinusoidal wave.
fmc.amtFmcRfRef(ip,refClk)
```

Output:

```
Linux analog 5.10.0-98248-g1bbe32fa5182 #1142 SMP PREEMPT Wed Aug 3

The programs included with the Debian GNU/Linux system are free soft
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 23 09:07:48 2023
su
analog@analog:~$ su
Password:

root@analog:/home/analog#
cd /home/analog/Public
root@analog:/home/analog/Public#
./amtFmcRfRef.sh int 80
Rebuild devicetree...
CURRENT BOARD CONFIG:
DONE
Reference clock = internal. Clock frequency = 80 MHz.
```

2. Reboot the device.