آرتمیس کاریا

www.artemiskaria.ir

آرتمیس کاریا (آلینکس) - فناوری FPGA

AK-SDR 201/3 Series User Manual Pluto Porting Tutorial

Version Records：

Data	Version	Description
2023.2.16	V1.0	initial version

This tutorial will continue to revise, optimize and increase based on the actual Experience, that is to provide you with more and better Demos.

If you find some errors or any suggestion, contact with us.

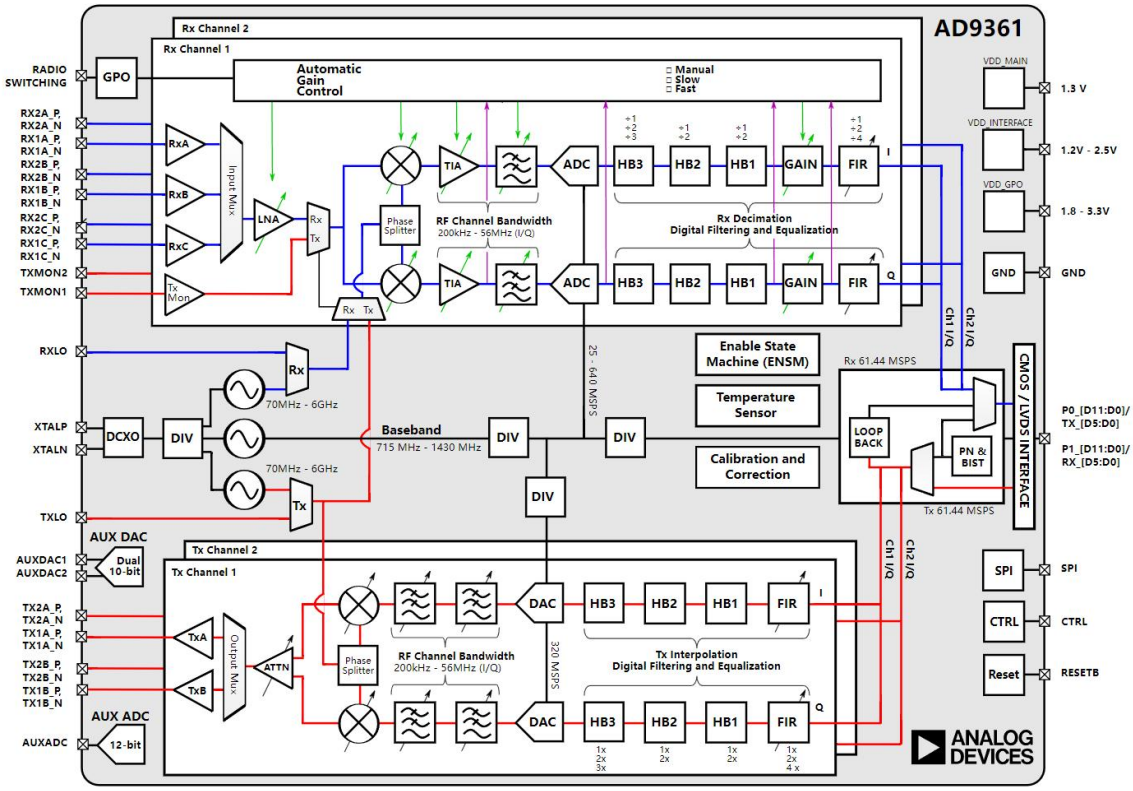
Content

Part 1: ADI-AD936X Series Introduction

The AD9361 , AD9364 and AD9363 are high-performance, highly integrated RF Agile Transceivers™. Their programmability and broadband capability make them ideal for a wide range of transceiver applications. Combining an RF front-end with a flexible mixed-signal baseband section and integrated frequency synthesizer, these devices simplify design by providing a configurable digital interface to the processor.

The AD9361 and AD9364 both operate in the 70 MHz to 6.0 GHz range, while the AD9363 operates in the 380 MHz to 3.8 GHz range, covering most licensed and unlicensed bands.

The AD9361 and AD9364 have channel bandwidths from 200 kHz to 56 MHz, while the AD9363 has a frequency range of 200 kHz to 20 MHz. Both the AD9361 and AD9363 are 2 Rx, 2 Tx devices, while the AD9364 is a 1 Rx, 1 Tx device.



The AD9361 and AD9364 are highly integrated radio frequency (RF) transceivers that can be configured for a wide range of applications. These devices integrate all the RF, mixed-signal, and digital modules needed to provide all transceiver functionality in a single device. Programmability makes this broadband transceiver suitable for a wide range of communication standards, including frequency division duplex (FDD) and time division duplex (TDD) systems. This programmability also allows the device to connect to a variety of baseband processors (BBPs) using a single 12-bit parallel data port, dual 12-bit parallel data ports, or a 12-bit low-voltage differential signaling (LVDS) interface.

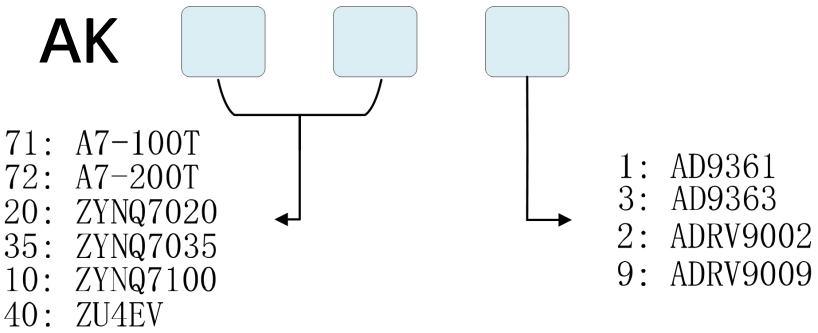
The AD9361 and AD9364 also offer a self-calibration and automatic gain control (AGC)

system that maintains high performance levels under changing temperature and input signal conditions. In addition, the devices include multiple test modes that allow system designers to insert test signals and create internal loopback patterns that designers can use to debug their designs during prototyping and optimize their radio configurations for specific applications.

Part 2: AK201 and AK203 of AK-SDR

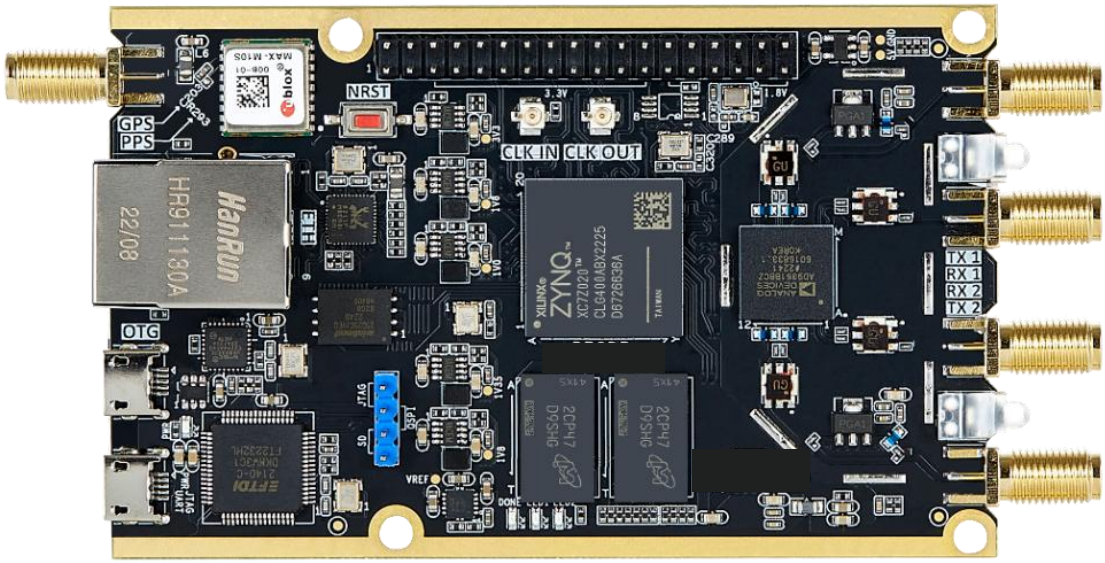
Part 2.1: AK-SDR Series Products Introduction

The AK-SDR series includes a number of RF front ends products based on xilinx FPGA chip and ADI's AD9363/AD9361/ADVR9002/ADVR9009. For example, the FPGA Chips are Artix7's A7-100T and A7-200T, ZYNQ7000 series 7020 and 7100, and ZYNQ UltraScale series 4EV, etc.There is a wide range of products, and to facilitate the selection process, we have defined the model specifications for the entire product family as shown in the following diagram, which details the differences between the models. This document is based on the P201, i.e. ZYNQ7020+AD9361 architecture.



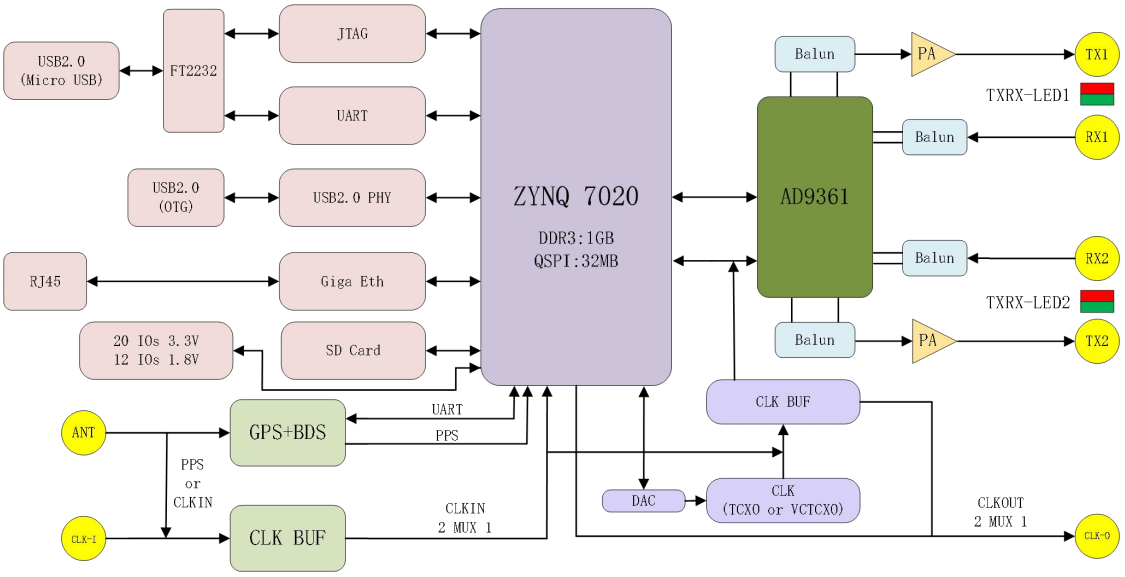
AK-SDR Module Definition

The AK201 and AK203 platforms are upgraded on the basis of ADI official pluto, the RF port adopts 2T2R mode, which is more powerful, and on this basis, it also integrates GPS, clock input and output and many other functions.



Part 2.2: Product Function Block Diagram

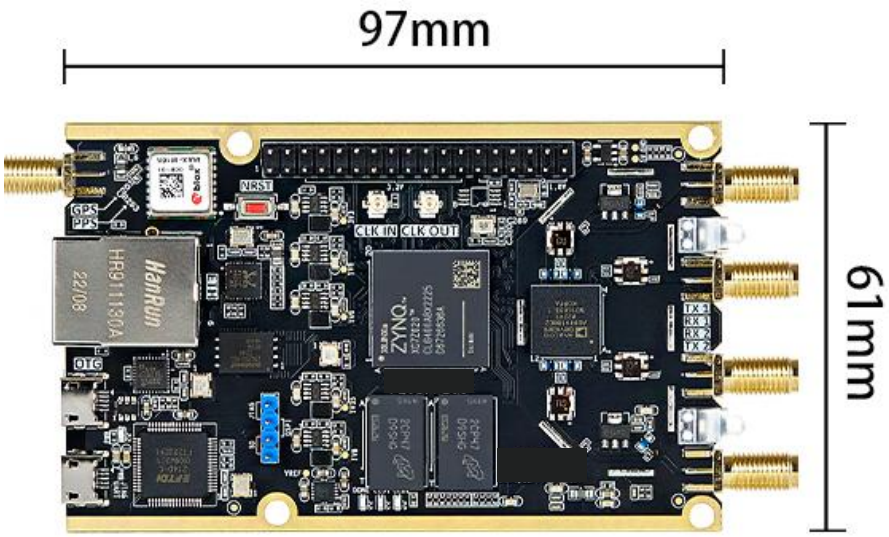
In this section, we will show the configuration details of the product in detail through the product block diagram and product specifications, as shown below.

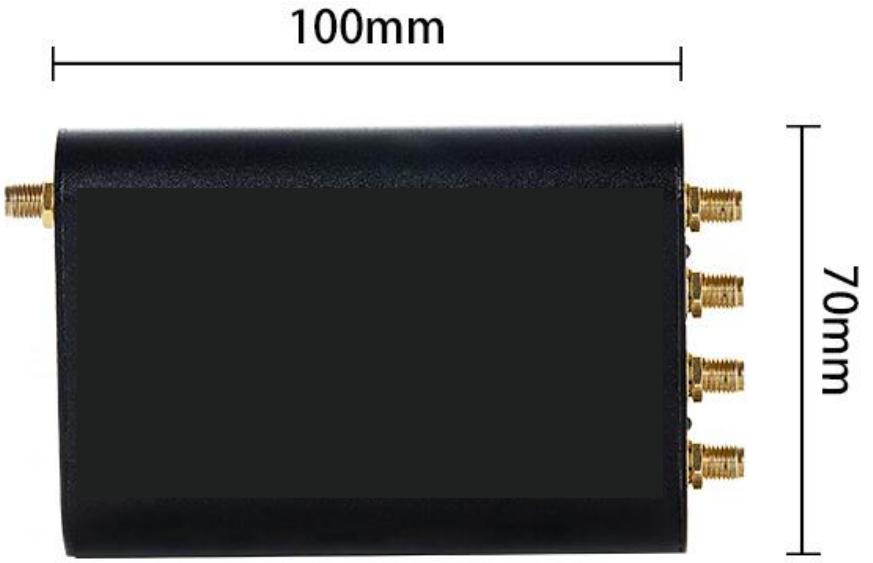


	AK201	AK203
RF Chip	AD9361	AD9363
Frequency range	70M-6Ghz	325M-3.8Ghz
Signal Bandwidths	200K-56Mhz	200K-20Mhz
Power Amplifier	2 way TX	
FPGA Chip	XC7Z020-2CLG400I	
Processor Core	Dual Core Cortex-A9, 766Mhz	
Logic Cells	85K	
DDR3	1GB	
QSPI FLASH	32MB	
RF Clock	Default TCXO/0.5PPM, VCTCXO is adjustable	
Clock Input/Output	1 input, 1 output	
Gigabit Ethernet	1	
USB OTG	1	
UART	1	
JTAG	1	
SD Card Slot	1	
GPS/PPS	Default GPS, PPS is adjustable	
Extended IO	20 x 3.3V IO, 12 x 1.8V IO	
Form Factors	Board: 97 x 61mm Shell: 100 x 70mm	
Power supply	5V/1A (USB powered)	

Part 2.3: Form Factors

Bare Board form Form Factors is 97mm * 61mm, and the shell box form factor is 100mm * 70mm



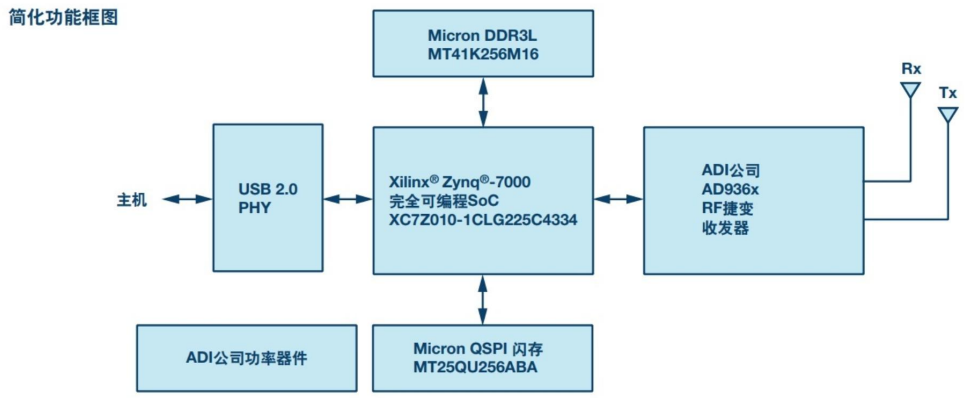


Part 3: Pluto Porting

Part 3.1: Pluto Introduction and Experiment Description

AK-SDR Series products is an upgrade from ADI-Pluto. Before we talk about AK-SDR product, let's introduce ADI-Pluto.

ADI-Pluto is an FPGA-based SDR architecture consisting of 5 main parts, as shown in the following figure:



The radio chip AD9363 within ADI-PLUTO is a high-performance, highly integrated RF agile transceiver based on a direct conversion receiver. The receiver subsystem includes a low-noise amplifier (LNA), direct conversion mixer, configurable analog filter, high-speed analog-to-digital converter (ADC), digital extraction filter, and 128-tap finite impulse response (FIR) filter that generates a 12-bit output signal at an appropriate sampling rate.

The receive chain is expanded with configurable automatic gain control (AGC) or manual gain modes, DC bias correction, and quadrature correction. The received I and Q signals are passed to the digital baseband processor, in this case the Xilinx Zynq SoC.

The transmission subsystem also uses a direct conversion architecture. The 12-bit I/Q samples are received from the baseband processor (in this case also a Xilinx Zynq SoC) and output to the antenna via a 128-tap finite impulse response (FIR) filter, digital interpolation filter, high-speed digital-to-analog converter (DAC), analog filter, direct conversion mixer, and small power amplifier (PA).

The phase lock loop (PLL) integrated in the AD9363 provides the clock and local oscillator for the receive and transmit channels, and the clock and sampling frequency for the ADC and DAC. The AKSDR uses the AD9361, which is an upgraded version of the AD9363, and the FPGA chip we used is the Zynq XC7Z020-2CLG400I. Because of the hardware differences, we needed to port the features implemented in the ADI-Pluto to the AKSDR product.

Experimental note: Since the file provided by ADI-Pluto-github does not have a file for our our FPGA development board (AKSDR Series Board), we modified and added it based on

the file provided by ADI-Pluto-github.

We need to compile the firmware for AKSDR board based on the modified ADI-Pluto-github file and be able to run the corresponding functions.

You can check the source files we provide and github to learn.

Github: https://github.com/analogdevicesinc/plutosdr-fw

Part 3.2: Pluto Porting Guideline

Note before porting: Pluto porting is done under Ubuntu system, users need to prepare Ubuntu system host or virtual machine (my Ubuntu version is 18.04), and install Vivado2021.1 version and Vivado2019.1 version.

Because we are using ADI's latest V0.35 version, we need both Vivado2021.1 and Vivado2019.1 environment when compiling, otherwise there will be errors.

If you use the source files provided by us, please refer to our steps.

Users can also refer to the steps of the official website, but the official source file did not add the files adapted to our FPGA development board (AK-SDR Series Board), users can modify to add their own, the official website link is as follows:

https://github.com/analogdevicesinc/plutosdr-fw

In the Ubuntu system terminal, start by executing the following command:

```
sudo apt-get install git build-essential fakeroot libncurses5-dev  libssl-dev ccache
sudo apt-get install dfu-util u-boot-tools device-tree-compiler  mtools
sudo apt-get install bc python cpio zip unzip rsync file wget
sudo apt-get install libtinfo5 device-tree-compiler bison flex  u-boot-tools
sudo apt-get purge gcc-arm-linux-gnueabihf
sudo apt-get remove libfdt-de
```

We have added the files for the user to adapt to our FPGA development board (AK-SDR Series Board), so the user can go to the pluto-fw directory, execute the following command, and wait for it to finish.

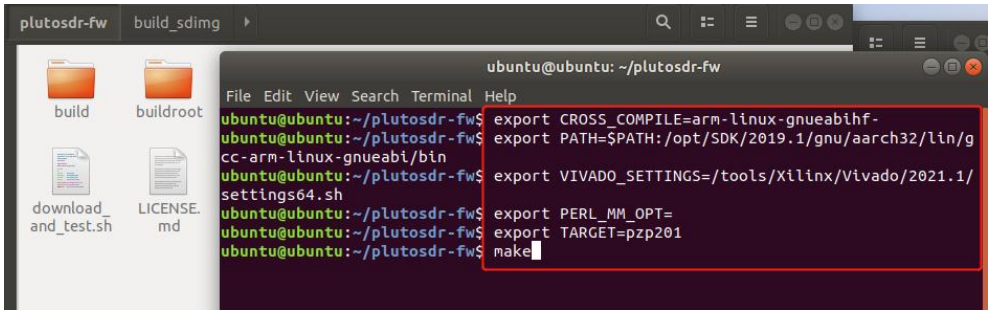
Note: The compilation process needs to be connected to the Internet, and the user needs to be able to access the Google website under Ubuntu, otherwise the compilation may go wrong.

The command is as follows (user path: user to modify according to their actual path):

```
export CROSS_COMPILE=arm-linux-gnueabihf-
export PATH=$PATH:\
/user path/SDK/2019.1/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin
export VIVADO_SETTINGS=/user path/Vivado/2021.1/settings64.sh
export PERL_MM_OPT=
export TARGET=pzp201
```

make (if you have run make before, you can execute make clean to clear it first, then execute make)

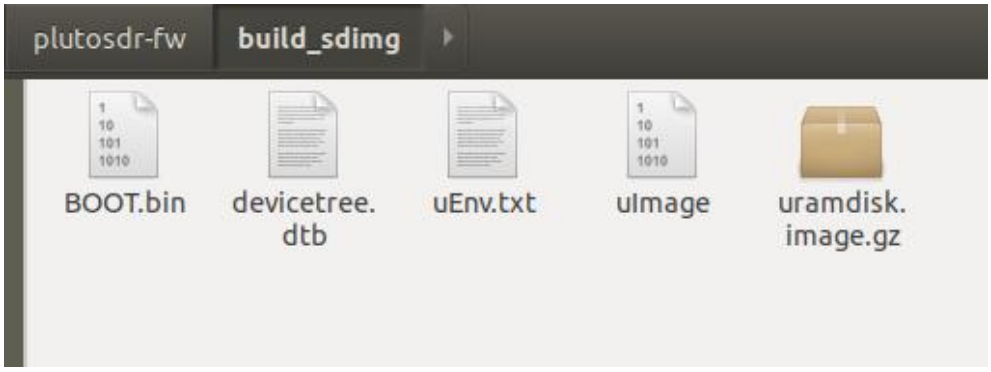
The author ran the command as follows:



After the previous command is executed, the following command is executed to generate the SD card firmware:

make sdimg

The generated firmware is as follows:

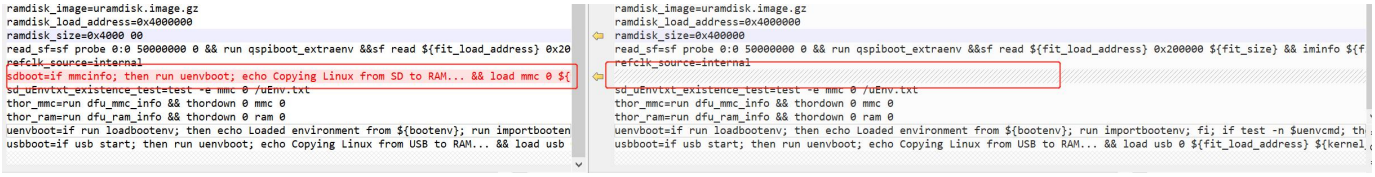


The uEnv.txt file we need to make the following changes in order to support SD card boot.

Take the content of the left red box and add the right red box location.

The left side reads as follows:

sdboot=if mmcinfo; then run uenvboot; echo Copying Linux from SD to RAM... && load mmc 0 ${fit_load_address} ${kernel_image} && load mmc 0 ${devicetree_load_address} ${devicetree_image} && load mmc 0 ${ramdisk_load_address} ${ramdisk_image} && bootm ${fit_load_address} ${ramdisk_load_address} ${devicetree_load_address}; fi



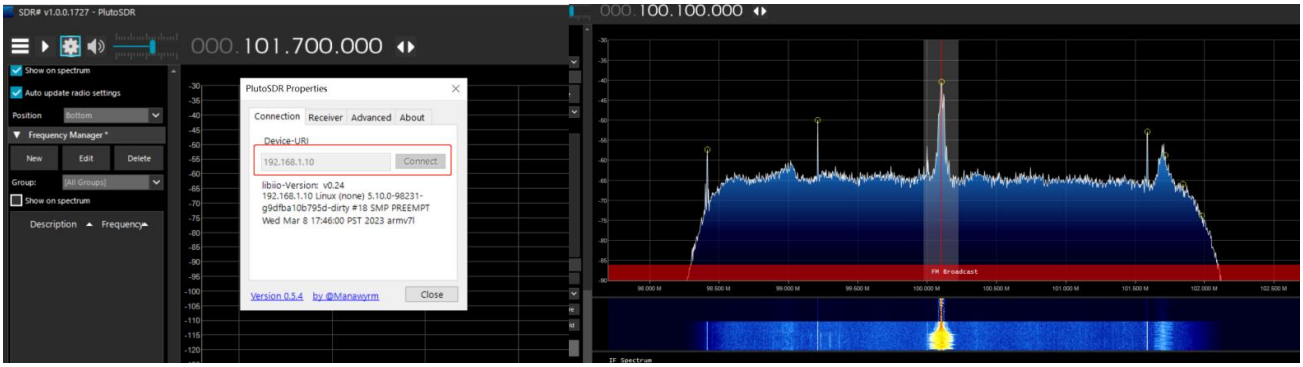
At this point, we can take the firmware generated above, copy it to the SD card, power it up and run it.

Login account is: **root**; Password: **artemis**

```
random: crng init done
random: 1 urandom warning(s) missed due to ratelimiting
Starting network: macb e000b000.ethernet eth0: PHY [e000b000.ethernet-ffffffff:01] driver [RTL8211F Gigabit Ethernet] (irq=POLL)
macb e000b000.ethernet eth0: configuring for phy/rgmii-id link mode
OK
Starting dhcpd Daemon & httpd Server: OK
Starting MSD Daemon: OK
Starting dropbear sshd: OK
Starting input-event-daemon: input-event-daemon: open(/dev/input/event0): No such file or directory
done

Welcome to AK
AK login: root
Password:
Welcome to:
 _____ _ _ _____ _____ _____
/  _  \/ |/ //  ___\/  _  \/  __\
| / \||   / |  \    \| | \||  \/|
| |-|||   \ |  \___ || |_/||    /
\_/ \|\_|\_|\_____/\____/\_/\_\

v0.37-1-g1077-dirty
https://artemiskaria.ir/

#
```

We can open the radio software to test the function

The FPGA development board is connected to the antenna, the network cable is connected to the computer, and the software is connected as follows to listen to the radio



At this point, the Pluto experiment is complete.