

DC-Coupled 10GSPS Digitizer Evaluation Board

FEATURES

- ▶ Dual-path, low noise design for broad coverage from DC to 5GHz
 - ▶ High-speed path with AD9213, a low noise 12-bit ADC running at 10GSPS
 - ▶ ADL5580, a fully differential 10GHz ADC driver
 - ▶ Precision path with AD4080, a 20-bit, 40MSPS, differential SAR ADC reduces 1/f noise
- ▶ ADC driver input, biased with DAC output voltage, adjusts DC offset of unipolar signal
 - ▶ Single-ended unipolar signal with DC offset to differential signal conversion by ADL5580
 - ▶ ADL5580 biased by DAC LTC2664 output for maximizing AD9213 input dynamic range
- ▶ FMC+ HSPC connector for interface with Xilinx VCU118 FPGA board

PACKAGE CONTENT

- ▶ ADMX6001-EBZ board
- ▶ AC-DC power supply, SDI65-12-UC-P6 (12V/65W)

ADDITIONAL EQUIPMENT NEEDED

- ▶ Xilinx VCU118 FPGA evaluation board
- ▶ A benchtop function generator

APPLICATION TOOLS AND DRIVER NEEDED

- ▶ Vivado Lab Edition, 2024.2 or newer (Xilinx Vivado Lab Edition downloads)
- ▶ Silicon Labs USB to UART Virtual COM Port Driver (Silicon Labs VCP Windows Driver)
- ▶ Terminal Emulator (Tera Term 5.4.0)
- ▶ IIO Oscilloscope (optional IIO Oscilloscope download, the ADMX6001-EBZ board can be fully evaluated with IIO Oscilloscope or Python scripts)
- ▶ Python 3.11 and the Python for ADI I/O devices (PyADI-IIO) library (optional, the ADMX6001-EBZ board can be fully evaluated with IIO Oscilloscope or Python scripts)
- ▶ Xilinx VCU118 programming files (admx6001_v1_boot.zip)

GENERAL DESCRIPTION

The ADMX6001-EBZ is a reference design of a DC-coupled single channel 10GSPS digitizer featuring the [AD9213](#), a low noise, 12-bit high-speed analog-to-digital converter (ADC) and the [AD4080](#), a 20-bit precision ADC. The dual-path design achieves true low noise digitization in the broadband from DC to 5GHz. By biasing the ADC driver [ADL5580](#) with the precision digital-to-analog converter (DAC) [LTC2664](#), the ADMX6001-EBZ is capable of handling unipolar and bipolar signals at various DC levels, maximizing utility of the input dynamic range of AD9213. This design is ideal for high performance time-domain instruments such as time-of-flight mass spectrometry (TOF MS), distributed fiber optic sensing (DFOS), and digital oscilloscope.

This document guides the setup and evaluation of the ADMX6001-EBZ board in conjunction with the Xilinx VCU118 field programmable gate array (FPGA) board.

EVALUATION BOARD PHOTO

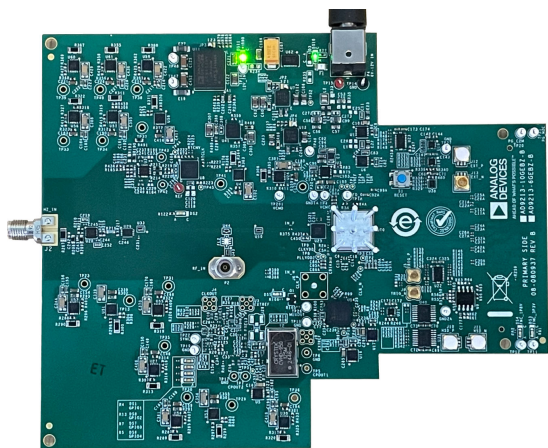


Figure 1. ADMX6001-EBZ Evaluation Board

TABLE OF CONTENTS

Features.....	1	ADMX6001-EBZ Data Acquisition With IIO	
Package Content.....	1	Oscilloscope.....	15
Additional Equipment Needed.....	1	IIO-Oscilloscope Plot Window.....	15
Application Tools And Driver Needed.....	1	Input Signal Offset Shift for AD9213.....	15
General Description.....	1	Configure and Control ADMX6001-EBZ With	
Evaluation Board Photo.....	1	Python Scripts.....	17
Hardware Setup.....	3	Install Python and Git for Windows.....	17
Software Setup.....	4	Download the PyADI-IIO Library.....	17
Program the Xilinx VCU118 Board.....	5	Create and Configure Python Venv for the	
Configure and Control the ADMX6001-EBZ		ADMX6001-EBZ.....	17
With IIO Oscilloscope.....	9	Install Additional Python Libraries.....	19
Connect to Xilinx VCU118 + ADMX6001-		Run Example Python Scripts for the	
EBZ.....	9	ADMX6001-EBZ.....	20
Configure the AD9213 High-Speed Path.....	11	Notes.....	21
Configure the AD4080 Precision Path.....	13		
AD4080 Data Interface Configuration			
Registers.....	14		

REVISION HISTORY**9/2025—Rev. 0 to Rev. A**

Changes to Hardware Setup Section.....	3
Changes to Program the Xilinx VCU118 Board Section.....	5
Changes to Configure the AD9213 High-Speed Path Section.....	11
Added Figure 21; Renumbered Sequentially.....	12

6/2025—Revision 0: Initial Version

HARDWARE SETUP

The ADMX6001-EBZ board works with a Xilinx VCU118 FPGA board that serves as the controller for configuring the ADMX6001-EBZ board and streams captured data to a PC for post processing.

The hardware setup includes the following components:

- ▶ ADMX6001-EBZ board (the power adapter is included in the ADMX6001-EBZ package)
- ▶ Xilinx VCU118 FPGA board (the power adapter is included in the Xilinx VCU118 package)
- ▶ 2x Micro-USB cables
- ▶ Ethernet cable
- ▶ Signal generator
- ▶ PC running Windows 10 or 11

The 12V power adapters for the ADMX6001-EBZ board and the Xilinx VCU118 FPGA board must be unplugged from the wall outlets before inserting into the barrel connector (P4) on the ADMX6001-EBZ board and the power connector on the Xilinx VCU118 FPGA board with the VCU118 power switch in the OFF position.

Figure 2 shows the connection of the ADMX6001-EBZ board to the Xilinx VCU118 FPGA board via an FMC+ connector and other cables for the connection of power supplies, USB, Ethernet, and the test signal from the signal generator. The Micro-USB cables and Ethernet cable are connected to the PC via a USB adapter if needed. An example of the complete setup is shown in Figure 3. A small bench-top fan is required for heat dissipation of the ADMX6001-EBZ board (refer to Figure 3).

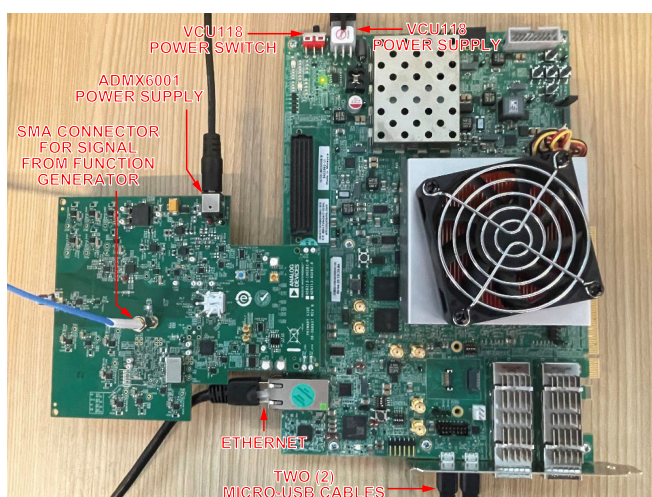


Figure 2. ADMX6001-EBZ Board with the VCU118 FPGA Board

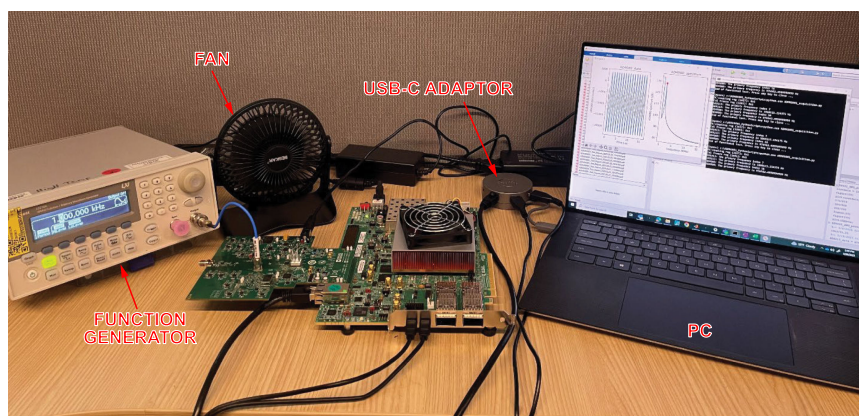


Figure 3. Example of the Complete Setup for the ADMX6001-EBZ Evaluation Board

SOFTWARE SETUP

Download and install the following software to configure and evaluate the ADMX6001-EBZ board:

- ▶ [Xilinx Vivado or Vivado Lab Edition](#)
- ▶ [UART Terminal Tera Term](#)
- ▶ [IIO Oscilloscope download](#)¹

Unless Xilinx Vivado has already been installed on your computer, it is recommended that you install the Vivado Lab Edition. Click [Downloads](#) and scroll down to the Vivado Lab Edition as highlight-

ed in [Figure 4](#) to download and install the Vivado Lab Edition. The default installation path for the Vivado Lab Edition is **C:\Xilinx\Vivado_Lab\2024.2\bin** where **2024.2** indicates the version. The path name needs to be revised accordingly if a different version of the Vivado Lab Edition is installed.

The UART Terminal Tera Term is for monitoring the booting process and then logging in to the Xilinx VCU118 system.

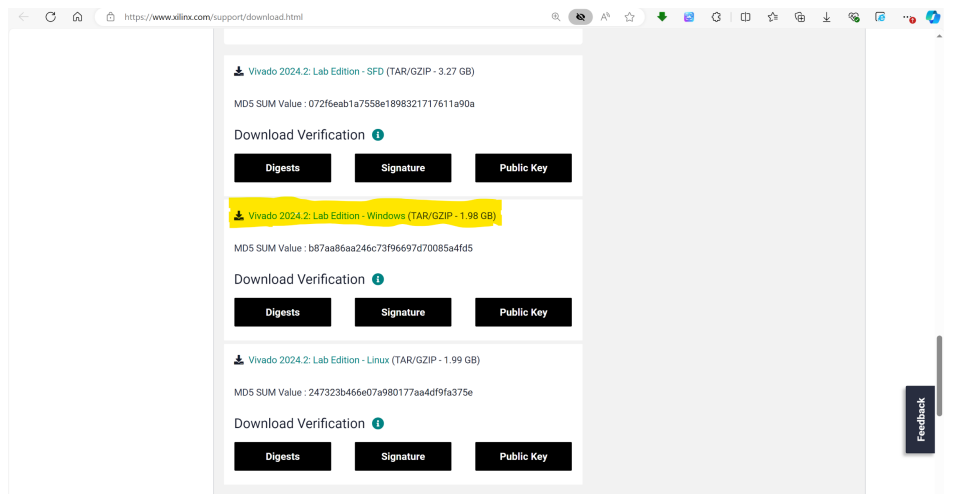


Figure 4. Download the Vivado Lab Edition for Windows

¹ Optional. IIO Oscilloscope is for configuring the ADMX6001-EBZ board and visualizing and/or saving the captured data. Alternately, you can use Python scripts to configure the ADMX6001-EBZ board and plot and/or save the captured data.

PROGRAM THE XILINX VCU118 BOARD

The programming files for Xilinx VCU118 FPGA are provided as a .zip file. Unzip the files to a working folder for all of the required files.

Refer to the [Hardware Setup](#) section. The 12V power adapters to both the ADMX6001-EBZ board and the Xilinx VCU118 FPGA board must be unplugged from the wall outlets before turning on the boards.

Turn on the boards sequentially. First, make sure that the connection of the 12V power adapter output to the barrel connector (P4) on the ADMX6001-EBZ board is securely in place, then turn on the ADMX6001-EBZ board by plugging in the 12V power supply to the wall outlet. The DS10 and DS11 on the same edge as the barrel connector on the ADMX6001-EBZ board must be lit. Otherwise, unplug the 12V power adapter from the wall outlet and wait for about 10 seconds before plugging the 12V power supply back to the wall outlet. Second, make sure that the power switch on the Xilinx VCU118 board is in the OFF position and the output of the 12V power supply is securely in place. Plug in the 12V power adapter for the Xilinx VCU118 board to the wall outlet, then turn on the Xilinx board by turning the power switch to the ON position. Refer to [Figure 2](#) for the Xilinx VCU118 board power supply connector and switch.

Before programming the Xilinx VCU118 board, it is important to configure Ethernet to an IP 192.168.2.xx where xx can be any number between 2 and 255. If a USB adapter with Ethernet port is used, plug in the USB adapter before reviewing and/or configuring the IP address.

Follow these steps to configure the Ethernet IP address:

1. **Control Panel > Network and Sharing Center > click Ethernet** to open the **Ethernet Status** window.
2. Click **Properties** to open the **Ethernet Properties** window.
3. Select **Internet Protocol Version 4 (TCP/IP)**.
4. Click **Properties** to open the **Internet Protocol Version 4 (TCP/IPv4) Properties** window.
5. Review/change IP address.

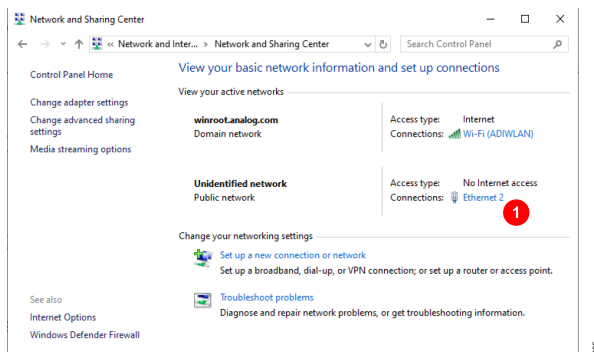


Figure 5. Review and Change Ethernet IP Address—Network and Sharing Center

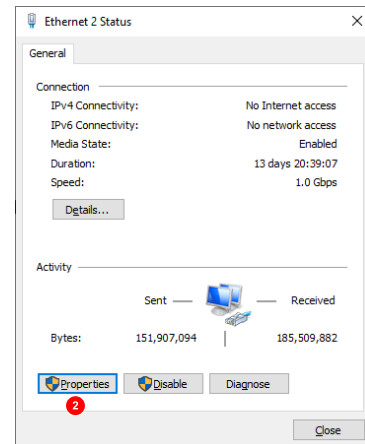


Figure 6. Review and Change Ethernet IP Address—Ethernet Properties

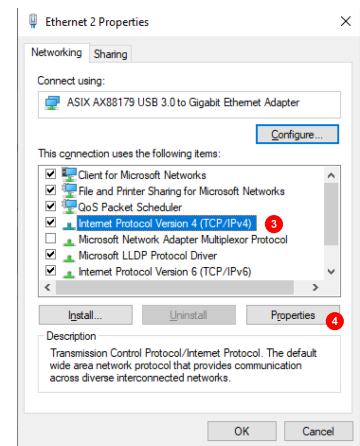


Figure 7. Review and Change Ethernet IP Address—TCP/IPv4

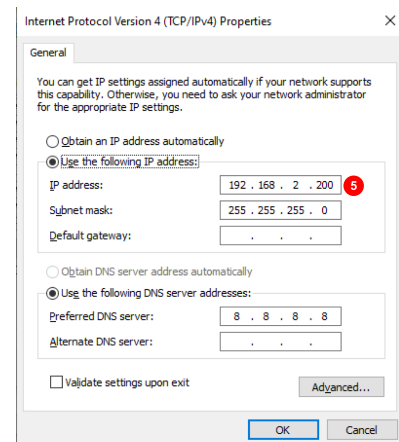


Figure 8. Review and Change Ethernet IP Address—Internet Protocol Version 4 (TCP/IPv4) Properties

Find the UART COM port of the Xilinx VCU118 board by opening **Device Manager** and expanding **Ports (COM & LPT)**. Locate **Silicon Labs Dual CP2105 USB to UART Bridge Standard COM Port (COMx)** to get the COM port number for the Tera Term serial

PROGRAM THE XILINX VCU118 BOARD

terminal connection, as shown in [Figure 9](#). The user may need to update the CP210x USB to UART Bridge VCP drivers.

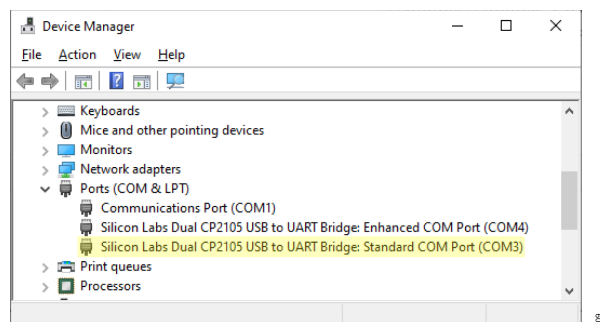


Figure 9. Find UART COM Port Number

PROGRAM THE XILINX VCU118 BOARD

Program the board using Vivado Lab tool with xsdb, as shown in [Figure 10](#), by completing the following:

1. Open a Command Prompt window and go to the working folder where the programming files for Xilinx VCU118 FPGA are located.
2. Start **xsdb.bat** (for Vivado Lab Edition). The prompt should change to **xsdb%**.
C:\Xilinx\Vivado_Lab\2022.2\bin\xsdb.bat
3. Program the VCU118 FPGA board.
xsdb% source run.tcl

```

C:\WINDOWS\system32\cmd.exe
C:\VADMX6001_Programming>cd C:\VADMX6001_Programming\adm6001_v1_boot
C:\VADMX6001_Programming\adm6001_v1_boot>C:\Xilinx\Vivado_Lab\2022.2\bin\xsdb.bat
***** Xilinx System Debugger (XSD) v2022.2
**** Build date : Oct 14 2022 05:18:10
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

xsdb% source run.tcl
attempting to launch hw_server
***** Xilinx hw_server v2022.2
**** Build date : Oct 14 2022 at 05:18:10
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

INFO: hw_server application started
INFO: Use Ctrl-C to exit hw_server application

***** Xilinx hw_server v2022.2
  
```

Figure 10. Xilinx VCU118 Programming with Vivado Lab

```

section, _kaytab_strings: 0x80c4032c - 0x80c62835
section, _param: 0x80c62838 - 0x80c62f3f
section, _bdevr: 0x80c62f40 - 0x80c62fcf
section, _ex_table: 0x80c62fd0 - 0x80c643df
section, .notes: 0x80c643e0 - 0x80c64433
section, .data2: 0x80c65000 - 0x80c64fff
section, .data: 0x80c65000 - 0x80c64fff
section, .data..percpu: 0x80c6fb00 - 0x80c6ffff
section, .init.text: 0x80c6fb00 - 0x80d2b9e3
section, .init.data: 0x80d2b9e4 - 0x80d2ded7
section, .init.lvt: 0x80d2ded8 - 0x80d2edaf
section, .init.setup: 0x80d2ed90 - 0x80d2f25f
section, .initcall.init: 0x80d2f260 - 0x80d2f2ef
section, .con.initcall.init: 0x80d2f2f0 - 0x80d2f7f3
section, .bss: 0x80d30000 - 0x80d5722f
section, .init.ramfs: 0x80e58000 - 0x81131723
1790 0.2m/s 01:38
Setting PC to Program Start Address 0x80000000
Successfully downloaded C:\VADMX6001_Programming\adm6001_v1_boot\simpleimage.vcu118_ad9213.ad4
000_eob_strip
Info: MicroBlaze #0 (target 3) Running
xsdb% Info: tcfchan#0 closed
xsdb%
  
```

Figure 11. Xilinx VCU118 Programming with Vivado Lab: xsdb%

While programming the Xilinx VCU118 is in progress, open and start the serial terminal Tera Term session to observe the boot process and log in to the Xilinx VCU118 as shown in [Figure 12](#).

1. Start and connect the serial terminal Tera Term session using the COM port found previously.
2. In Tera Term, click **Setup > Serial port ...** to set the speed to 115200.
3. Continue monitoring the Xilinx VCU118 boot process. Once the process is complete, log in with the following information:
Login: root
Password: analog

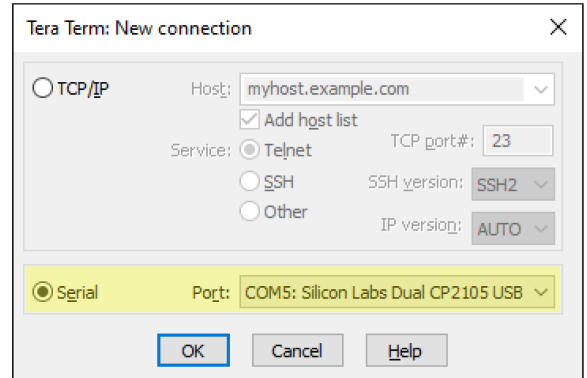


Figure 12. Serial Terminal Tera Term Session for Monitoring Xilinx VCU118 Booting and Login—Tera Term: New Connection

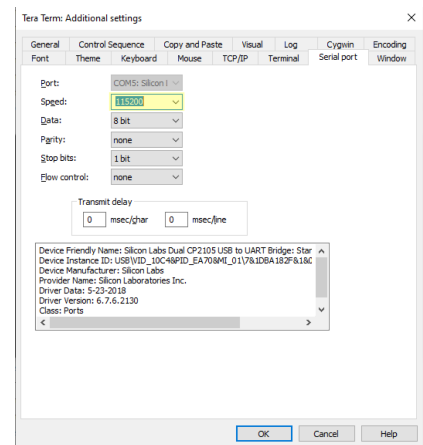


Figure 13. Serial Terminal Tera Term Session for Monitoring Xilinx VCU118 Booting and Login—Tera Term: Change Speed

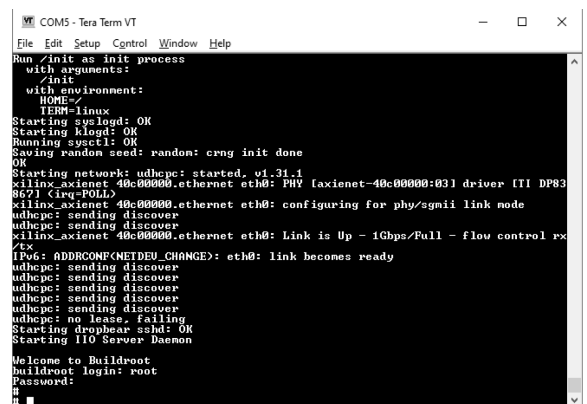


Figure 14. Booting and Login—Tera Term: Login

PROGRAM THE XILINX VCU118 BOARD

Note the following:

- ▶ For the Xilinx Vivado Lab version, only **xsdb.bat** is available (**C:\Xilinx\Vivado_Lab\2024\bin\xsdb.bat**).
- ▶ It is required to program the Xilinx VCU118 board twice in the Vivado Lab window.
 1. Run **source run.tcl** and wait for the login message to appear in the Tera Term window.
 2. Run **source run.tcl** a second time. Observe the boot process in the Tera Term window and log in.

If using Python scripts is preferred, skip the [Configure and Control the ADMX6001-EBZ With IIO Oscilloscope](#) section and the [ADMX6001-EBZ Data Acquisition With IIO Oscilloscope](#) section, and go directly to the [Configure and Control ADMX6001-EBZ With Python Scripts](#) section.

CONFIGURE AND CONTROL THE ADMX6001-EBZ WITH IIO OSCILLOSCOPE

Refer to the [IIO Oscilloscope wiki page](#) for guidance.

CONNECT TO XILINX VCU118 + ADMX6001-EBZ

The Xilinx VCU118 board uses a static IP address, **192.168.2.1**. Launch the IIO-Oscilloscope application and go to **Settings >**

Connect. In the popup window, connect to Xilinx VCU118 with manual URI **ip:192.168.2.1**, as shown in [Figure 15](#).

Once the IIO device is connected, the user can read/write registers and view the plot of captured data, as shown in [Figure 16](#) and [Figure 17](#).

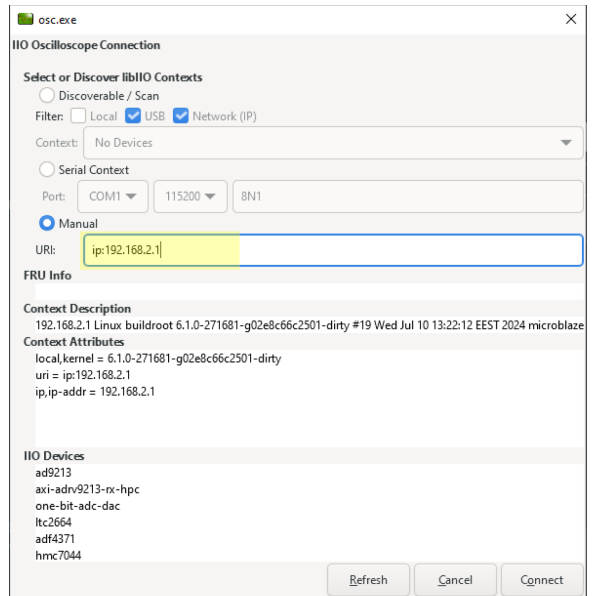


Figure 15. Launch IIO Oscilloscope and Connect to Xilinx VCU118

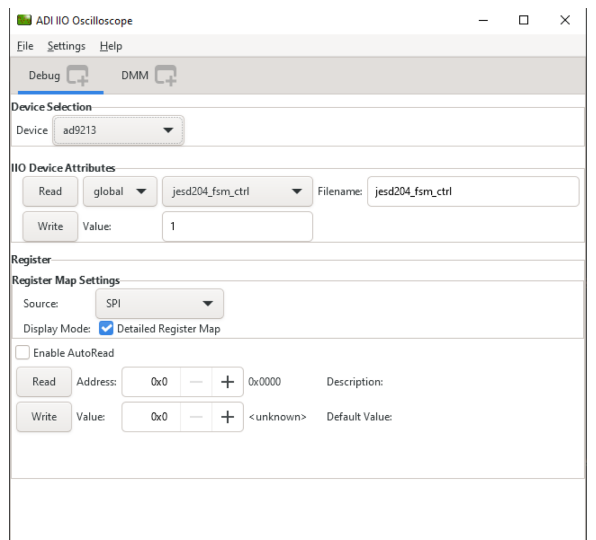


Figure 16. IIO Oscilloscope Window for Register Write/Read

CONFIGURE AND CONTROL THE ADMX6001-EBZ WITH IIO OSCILLOSCOPE

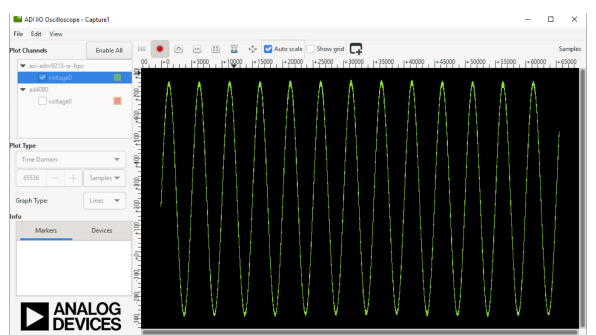


Figure 17. IIO Oscilloscope Plot Window

CONFIGURE AND CONTROL THE ADMX6001-EBZ WITH IIO OSCILLOSCOPE

CONFIGURE THE AD9213 HIGH-SPEED PATH

Set the following registers for the [AD9213](#) high-speed path:

- **ad9213** Register 0x1617 set to 0x01, as shown in [Figure 18](#).
- **ad9213** Register 0x1601 set to 0x01, as shown in [Figure 19](#).

- **ltc2664_clr** set to 1, as shown in [Figure 20](#).
- **adi5580_en** set to 1, as shown in [Figure 21](#).

The AD9213 high-speed path is ready for capturing data.

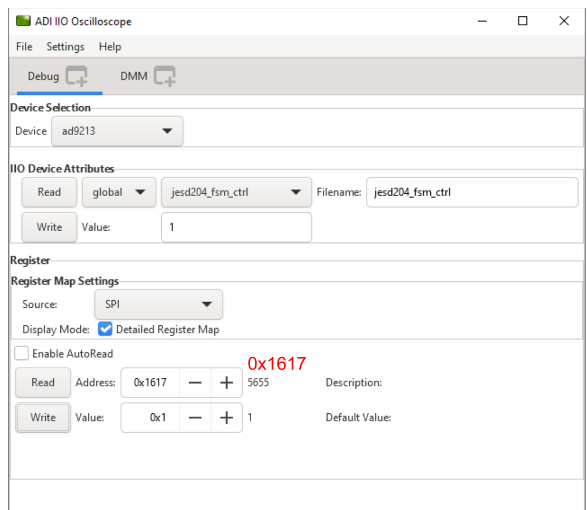


Figure 18. Register Settings for the AD9213 High-Speed Path Register 0x1617

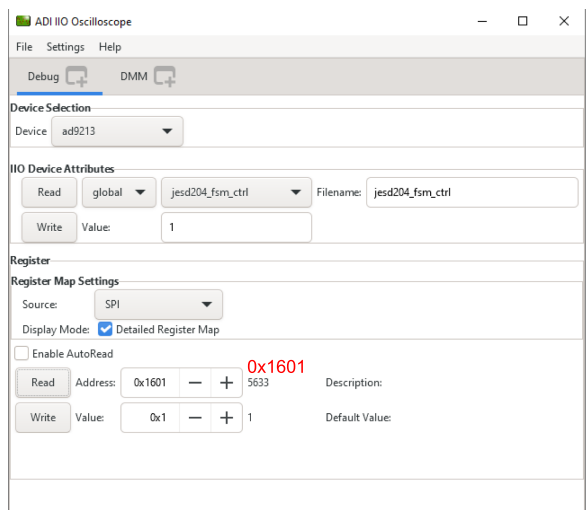


Figure 19. Register Settings for the AD9213 High-Speed Path Register 0x1601

CONFIGURE AND CONTROL THE ADMX6001-EBZ WITH IIO OSCILLOSCOPE

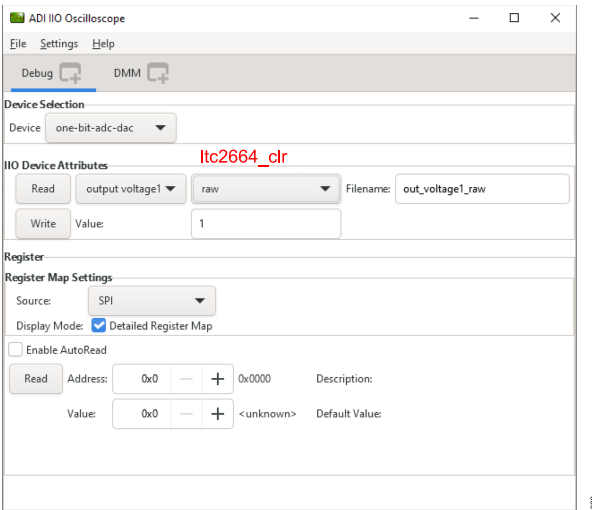


Figure 20. Register Settings for the AD9213 High-Speed Path ltc2664_clr

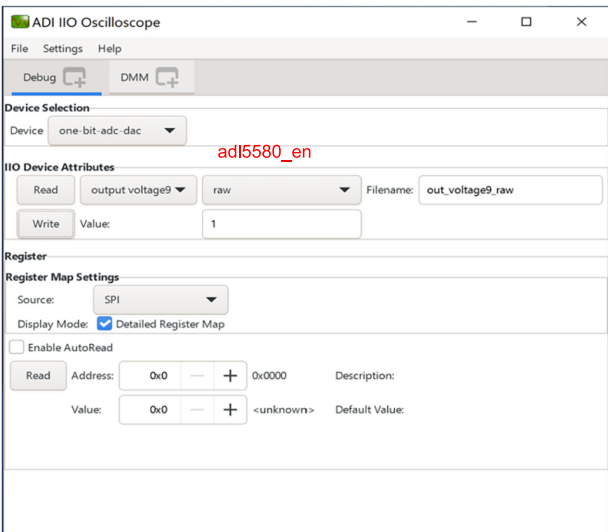


Figure 21. Register Settings for the AD9213 High-Speed Path adl5580_en

CONFIGURE AND CONTROL THE ADMX6001-EBZ WITH IIO OSCILLOSCOPE

CONFIGURE THE AD4080 PRECISION PATH

Before capturing the [AD4080](#) precision path data with the IIO Oscilloscope, a few registers need to be set, as shown in [Figure 22](#) to [Figure 25](#).

- ▶ **ltc2664_clr** set to 1 (also required for the AD9213 high-speed path)
- ▶ **ltc2664_output_voltage1** raw value set to 49152
- ▶ **ltc2664_output_voltage2** raw value set to 36045
- ▶ **ada4945_disable** set to 1
- ▶ **adg5419_ctrl** set to 1

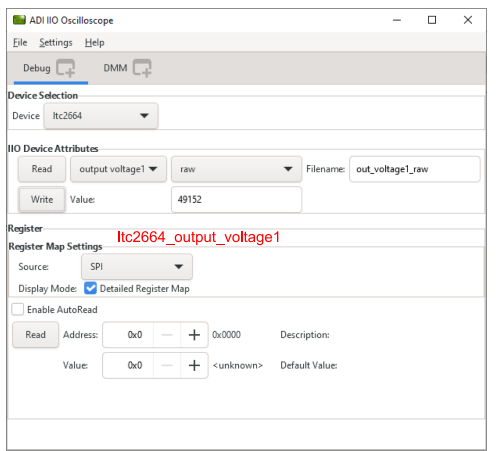


Figure 22. Register Settings for the AD4080 Precision Path: **ltc2664_output_voltage1**

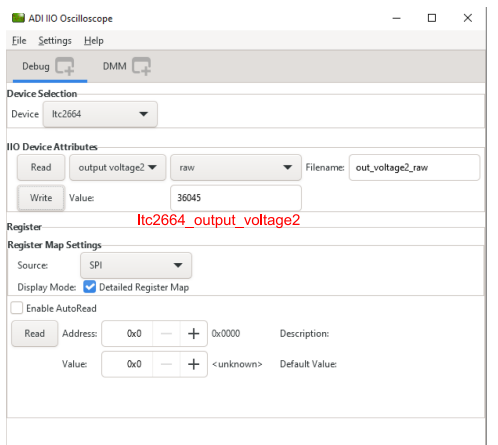


Figure 23. Register Settings for the AD4080 Precision Path: **ltc2664_output_voltage2**

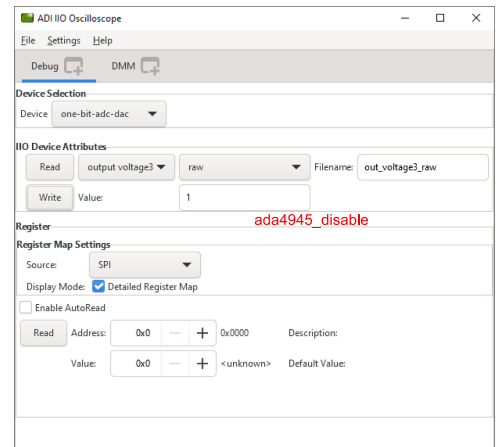


Figure 24. Register Settings for the AD4080 Precision Path: **ada4945_disable**

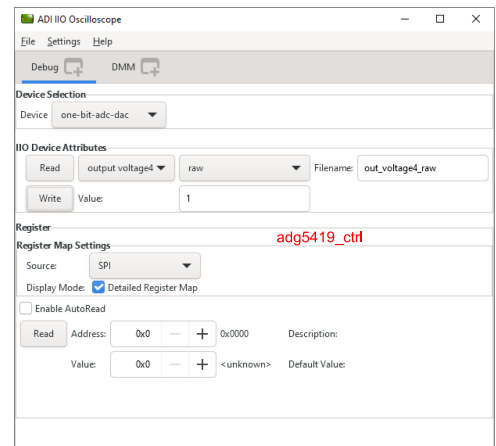


Figure 25. Register Settings for the AD4080 Precision Path: **adg5419_ctrl**

CONFIGURE AND CONTROL THE ADMX6001-EBZ WITH IIO OSCILLOSCOPE

AD4080 DATA INTERFACE CONFIGURATION REGISTERS

The AD4080 low voltage differential signaling (LVDS) data interface is highly configurable. In the ADMX6001-EBZ design, the AD4080 is configured to output the result data on single data lane. The AD4080 data interface configuration Register 0x15 and Register 0x16 need to be configured so that the Xilinx VCU118 controls are aligned with AD4080 conversion data for correct data output.

Complete the following steps for setting the AD4080 data interface configuration registers:

- ▶ AD4080 Register 0x15 set to 0x50 to enable the fixed pattern test, as shown in [Figure 26](#). This step sets AD4080 in test mode.
- ▶ In test mode, the 20-bit AD4080 data is expected to be binary 1010 1100 0101 1101 0110, decimal -342570, or hexadecimal 0xAC5D6. Save the AD4080 data to a .csv file in IIO Oscilloscope and check the value. Refer to the [ADMX6001-EBZ Data Acquisition With IIO Oscilloscope](#) section for saving data in IIO Oscilloscope.
- ▶ If the AD4080 data is different from the expected decimal -342570, set the AD4080 Register 0x15 to one of the following values: 0x51, 0x61, 0x71, 0x81, 0x91, 0x01, 0x11, 0x21, 0x31, or 0x41, as shown in [Figure 27](#). Save the data to a .csv file and check the value. Repeat this step with a value in the list until the AD4080 data is exactly -342570. The AD4080 register setting is completed.
- ▶ Set AD4080 Register 0x15 back to 0x40 to disable the fixed pattern test and enable AD4080 normal mode.

The AD4080 data interface configuration registers are set and the AD4080 is ready for data acquisition.

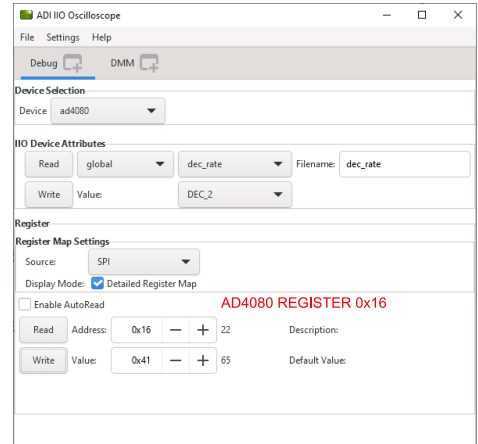


Figure 27. AD4080 Register 0x16 Setting

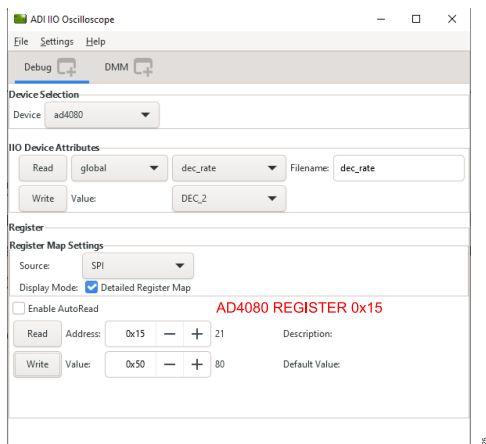


Figure 26. AD4080 Register 0x15 Setting—Value 0x50 for Fixed Test Pattern and 0x40 for Normal Data Acquisition Mode

ADMX6001-EBZ DATA ACQUISITION WITH IIO OSCILLOSCOPE

IIO-OSCILLOSCOPE PLOT WINDOW

Data captured by the the [AD9213](#) high-speed path and the [AD4080](#) precision path can be visualized and saved to a .csv file in the IIO Oscilloscope plot window, as shown in [Figure 28](#).

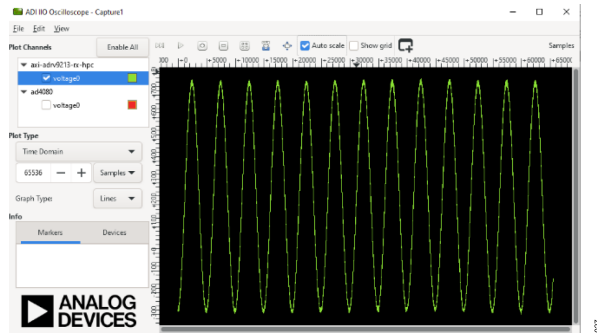


Figure 28. IIO Oscilloscope Plot Window—Time Domain: axi-adv9213-rx-hpc

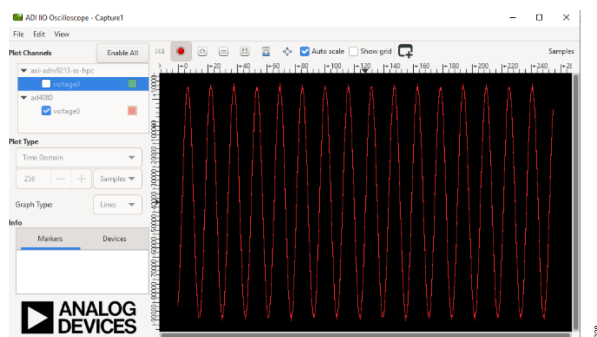


Figure 29. IIO Oscilloscope Plot Window—Time Domain: ad4080

The two data paths are listed in the top-left corner. Check one or both and then click the **Capture/Stop** button to view the data stream. For the IIO Oscilloscope to work smoothly, it is recommended that the number of samples is set to power of 2, such as 2^{10} (1024) or 2^{16} (65536). The IIO Oscilloscope supports only one scale for each axis so that it is not practical to view the AD9213 high-speed path and the AD4080 precision path simultaneously because the AD9213 high-speed path is 320 times faster than the AD4080 precision path.

The IIO Oscilloscope plot window can also view the captured data in the frequency domain. An example of a 2MHz sinewave captured by AD4080 is shown in [Figure 30](#).

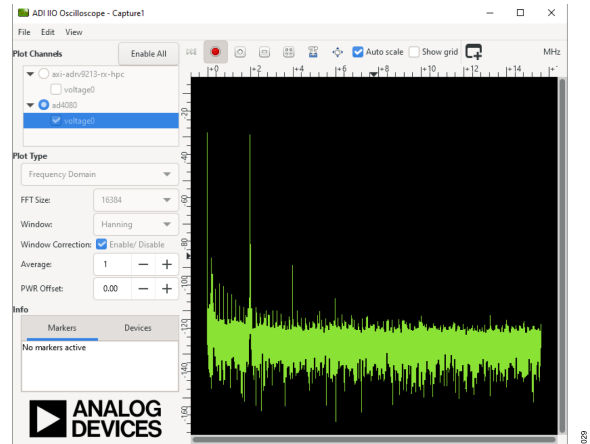


Figure 30. IIO Oscilloscope Plot Window—Frequency Domain

The data can be saved in a .csv file for post processing and analysis, as shown in [Figure 31](#). From the **File** menu, select **Save As** to open the **Save As** window. Make sure to select the path from the dropdown list and select the correct checkbox before clicking the **Save** button on bottom-right.

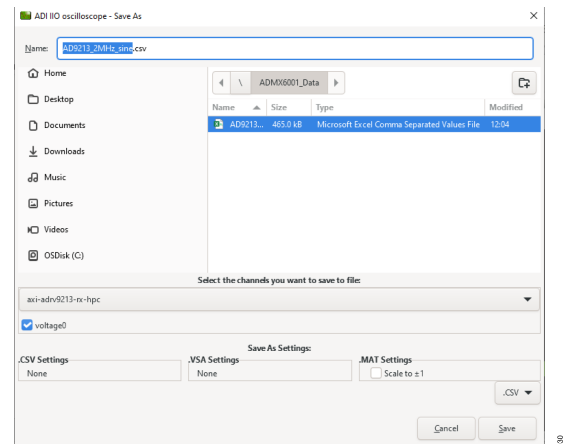


Figure 31. Save Data as a .CSV File in the IIO Oscilloscope Plot Window

INPUT SIGNAL OFFSET SHIFT FOR AD9213

The input signal is often from transducers or sensors as a unipolar signal with DC offset, whereas the AD9213 works best with differential input. By biasing the [ADL5580](#) input terminal with the output of the precision DAC [LTC2664](#), the ADL5580 converts the unipolar signal to a differential signal with an adjustable DC offset to maximize the utility of the AD9213 input dynamic range. The actual use case may vary depending on the specific application.

The DAC value of 32768 corresponds to 0V and is used as the default biasing to ADL5580. To achieve a targeted DC offset shift x mV, the DAC value can be calculated as

$$DAC\ value = 32768 \left(1 + \frac{x}{5000} \right) \quad (1)$$

where x is the DC offset in mV.

ADMX6001-EBZ DATA ACQUISITION WITH IIO OSCILLOSCOPE

Round the DAC value to the nearest integer as the DAC LTC2664 input. The DAC output range is -5V to +5V.

To change the LTC2664 DAC value for ADL5580 biasing, write the DAC value as raw data to **output_voltage0**, as shown in Figure 32. To illustrate the effect of offset shift, clear the **Auto scale** checkbox, set the Y max to 2048, and set the Y min to -2048. Figure 33 shows the AD9213 high-speed path data with the same sinewave input but different DAC settings for offset shift.

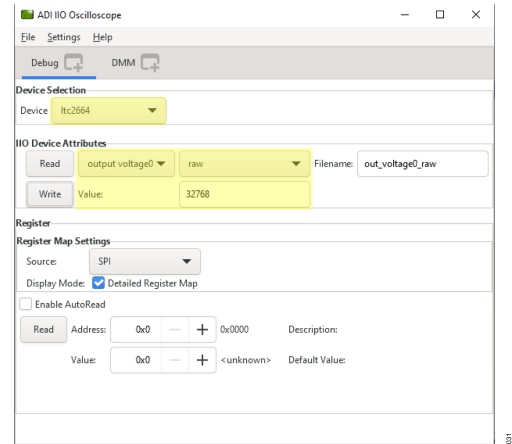


Figure 32. Write DAC Value of Targeted Voltage to LTC266

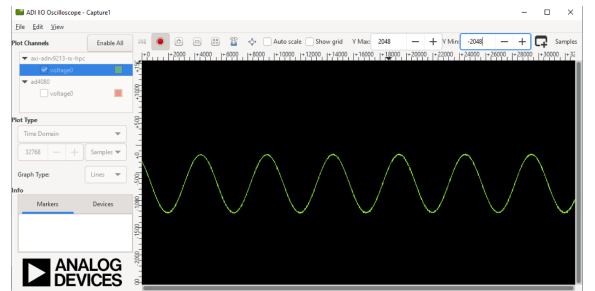


Figure 33. AD9213 Data with DC Offset at 0mV (32768 LSBs)

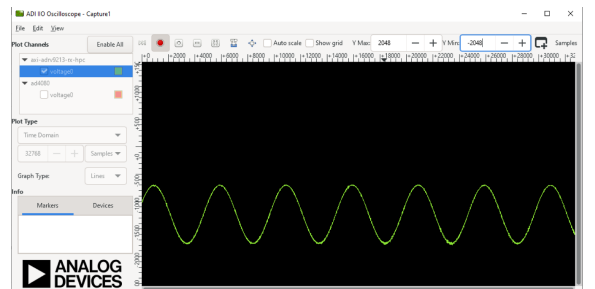


Figure 34. AD9213 Data with DC Offset at +100mV (33423 LSBs)

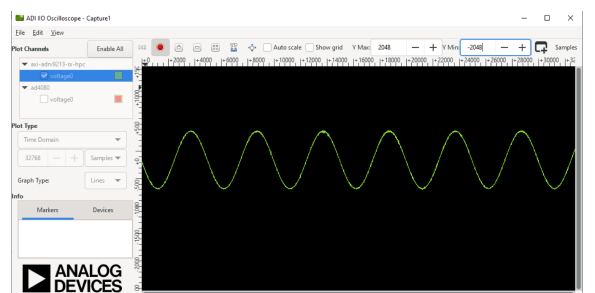


Figure 35. AD9213 Data with DC Offset at -100mv (32112 LSBs)

CONFIGURE AND CONTROL ADMX6001-EBZ WITH PYTHON SCRIPTS

Utilizing Python scripts offers advantages for configuring and controlling the ADMX6001-EBZ board and enables data visualization and processing. Python scripts are more efficient than the manual register read and write with IIO Oscilloscope. The Python virtual environment (venv) for running ADMX6001-EBZ Python scripts can be set with or without an IDE such as PyCharm. It is required that the user installs Python 3.11.5 or a newer version and downloads the PyADI-IIO library before proceeding to the venv setting in the following section.

INSTALL PYTHON AND GIT FOR WINDOWS

- Download **Windows Installer (64-bit)**, as shown in [Figure 36](#). Install Python 3.11.5. It is recommended to install Python 3.11.5 as an administrator by right clicking on the installer, as shown in [Figure 37](#).
- Download the **64-bit Git for Windows Setup** as shown in [Figure 38](#) and install Git.

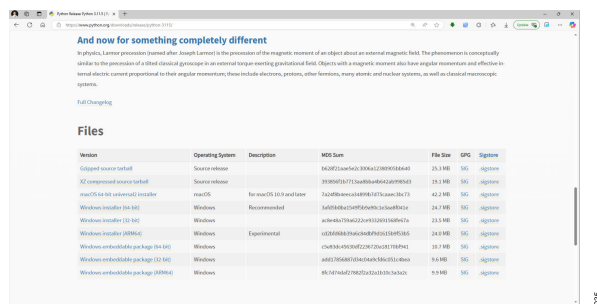


Figure 36. Download Python Windows Installer

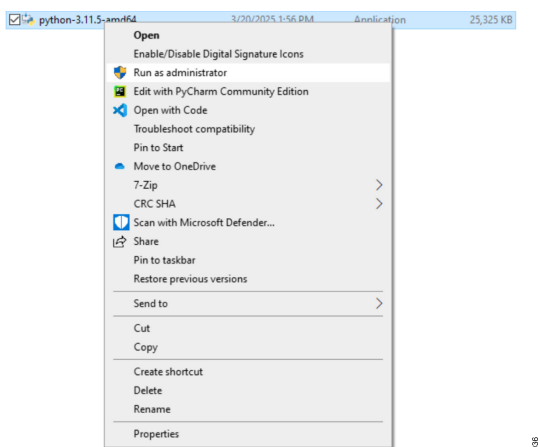


Figure 37. Install Python 3.11.5 as an Administrator

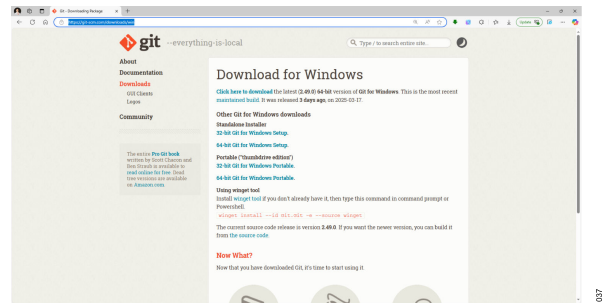


Figure 38. Download 64-bit Git for Windows Setup

DOWNLOAD THE PYADI-IIO LIBRARY

Start a Command Prompt window and clone the PyADI-IIO library for ADMX6001-EBZ, as shown in [Figure 39](#).

- Create and navigate to the folder for saving the PyADI-IIO library, `mkdir temp1 > cd temp1 > mkdir pyadi > cd pyadi`
- Clone the required PyADI-IIO library git clone from the GitHub website, https://github.com/analogdevicesinc/pyadi-iio/tree/admx6001_v1

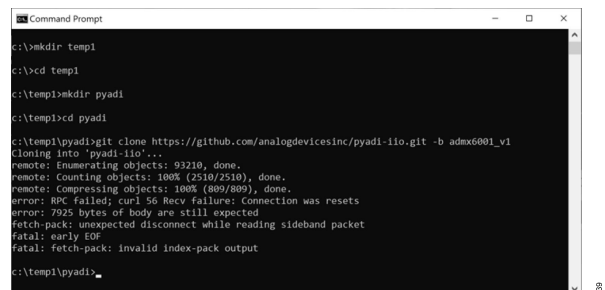


Figure 39. Clone PyADI-IIO Library for the ADMX6001-EBZ Evaluation Board

CREATE AND CONFIGURE PYTHON VENV FOR THE ADMX6001-EBZ

The venv module supports creating lightweight virtual environments, each with their own independent set of Python packages installed in the site directories of the virtual environments. The venv contains a specific Python interpreter, software libraries, and binaries that are needed to support a project (library or application). Complete the following steps to create and activate the venv for ADMX6001-EBZ and install the libraries:

- Create a folder and copy the Python scripts to this folder, in the example that follows **C:\ADMX6001_PythonScripts**.
- Open a Command Prompt window and go to the Python 3.11 folder (**C:\Python311** in the following example).
- To create the venv in the Command Prompt window run the following script: **python -m venv c:\ADMX6001_PythonScripts\myenv**.
- Activate the venv. Go to **C:\ADMX6001_PythonScripts\myenv\Scripts>activate**. Notice the prompt changes to (myenv), indicating the venv is activated.

CONFIGURE AND CONTROL ADMX6001-EBZ WITH PYTHON SCRIPTS

- In the (myenv) command window, install the PyADI-IIO library. Go to the PyADI-IIO library folder (C:\temp1\pyadi\pyadi-iio in the example) and install **pyadi-iio** library: **pip install**.

The venv is now created and activated, and the PyADI-IIO library is installed, as shown in [Figure 40](#).

```

Microsoft Windows [Version 10.0.19041.540]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ghuocd> cd C:\Python311
C:\Python311> python -m venv c:\adm6001_PythonScripts\myenv
C:\Python311> cd c:\adm6001_PythonScripts\myenv\Scripts
C:\adm6001_PythonScripts\myenv\Scripts> activate
(myenv) C:\adm6001_PythonScripts\myenv\Scripts> cd C:\temp1\pyadi\pyadi-iio
(myenv) C:\temp1\pyadi\pyadi-iio> pip install
Processing c:\temp1\pyadi\pyadi-iio
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: numpy<1.20 in c:\adm6001_PythonScripts\myenv\lib\site-packages (from pyadi-iio==0.17)
Requirement already satisfied: pylibiio==0.25 in c:\adm6001_PythonScripts\myenv\lib\site-packages (from pyadi-iio==0.17)
Building wheels for collected packages: pyadi-iio
Building wheel for pyadi-iio (pyproject.toml) ... done
Created wheel for pyadi-iio: filename=pyadi_iio-0.17-py3-none-any.whl size=184261 sha256=3bd60723458d47af911859d7b5
dc65c819b8f977b3cfd8701b1b8e6dd
Stored in directory: c:\users\ghu\appdata\local\pip\cache\wheels\0f\12\d7\6a31385047b5bf654297bcd496efc4d2b1299dac86
99f726
Successfully built pyadi-iio
Installing collected packages: pyadi-iio
Attempting uninstall: pyadi-iio
Found existing installation: pyadi-iio 0.8.17
Uninstalling pyadi-iio-0.8.17:
Successfully uninstalled pyadi-iio-0.8.17
Successfully installed pyadi-iio-0.17

[notice] A new release of pip is available: 21.2.1 -> 21.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(myenv) C:\temp1\pyadi\pyadi-iio>
  
```

Figure 40. Create and Activate the Venv and Install the PyADI-IIO library for the ADMX6001-EBZ Evaluation Board

The user can also setup the venv for ADMX6001-EBZ with an integrated development environment (IDE) such as PyCharm. To create and activate the venv, complete the following:

- Launch PyCharm and open the **ADMX6001_Pythonscripts** project, as shown in [Figure 41](#).
- Navigate to **File > Settings > Project > Python interpreter**, click **Add Interpreter > Add Local Interpreter** to create the venv, as shown in [Figure 44](#).
- The user can add additional Python libraries in a Command Prompt window (see the [Install Additional Python Libraries](#) section) or in PyCharm, as shown in [Figure 45](#).
- Launch a Command Prompt window to activate venv and install the PyADI-IIO, as shown in [Figure 47](#).

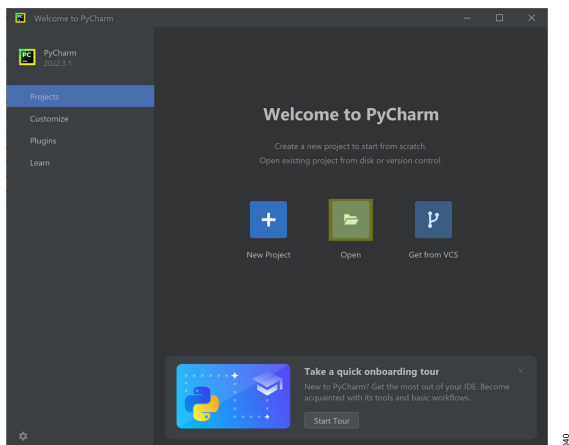


Figure 41. Launch PyCharm and Open the Python Scripts Project

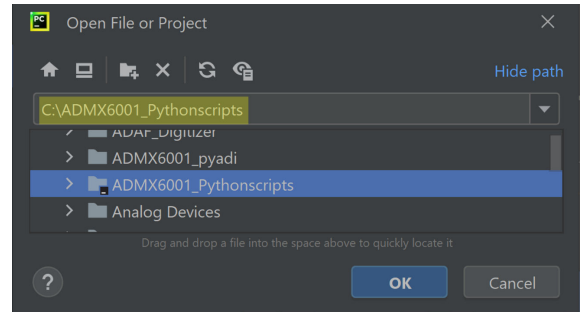


Figure 42. Open File or Project

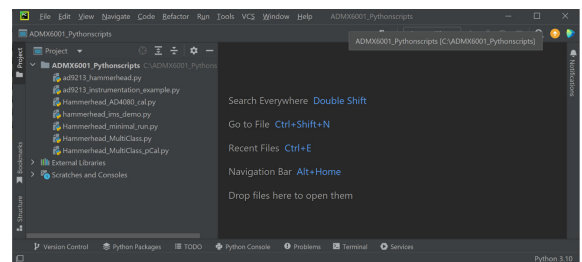


Figure 43. ADMX6001-EBZ Python Scripts

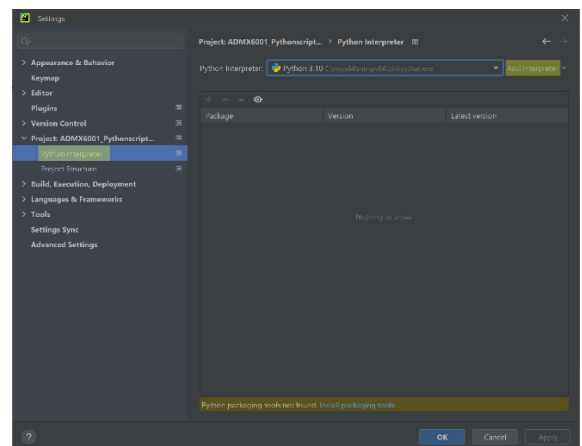


Figure 44. Create Venv by Adding a Local Interpreter

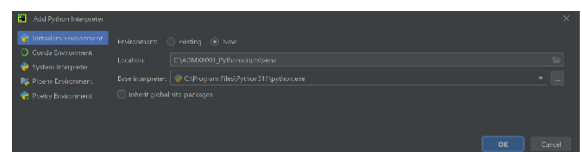


Figure 45. Install Additional Python Libraries in PyCharm

CONFIGURE AND CONTROL ADMX6001-EBZ WITH PYTHON SCRIPTS

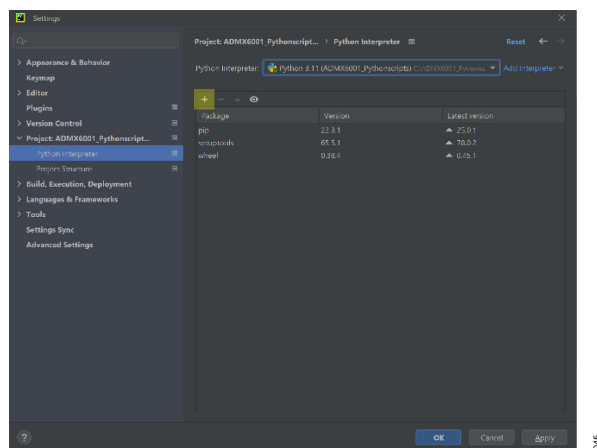


Figure 46. Python Interpreter Settings

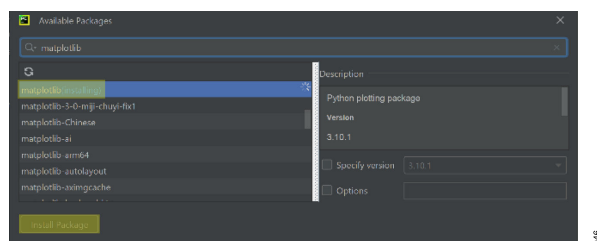


Figure 47. Activate Venv and Install PyADI-IIO

This completes the Python venv for running Python scripts for the ADMX6001-EBZ board

INSTALL ADDITIONAL PYTHON LIBRARIES

The Python scripts for the ADMX6001-EBZ board utilize modules from various libraries. Install these libraries, such as scipy and matplotlib, using **pip install <library name>** within venv before running the Python scripts, as shown in Figure 48.

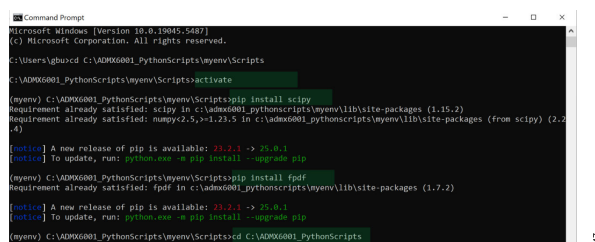


Figure 48. Install Additional Libraries and Run Python Scripts for ADMX6001-EBZ Evaluation Board

The Python scripts call modules in the PyADI-IIO library to initialize the ADMX6001-EBZ board by setting required registers, JSED and LVDS data lanes, and other control signals. Examples of Python scripts can be found at the product page for the ADMX6001-EBZ board.

RUN EXAMPLE PYTHON SCRIPTS FOR THE ADMX6001-EBZ

Examples of Python scripts are provided on the ADMX6001-EBZ board product page for board evaluation. These Python scripts call modules in the PyADI-IIO library to initialize the ADMX6001-EBZ board by setting required registers, JSED and LVDS data lanes, and other control signals. The user is encouraged to check the product page of the ADMX6001-EBZ board for updated and/or additional examples of Python scripts.

The example script **ADMX6001_MultiClass_pCal.py** defines the classes to setup, initiate, configure, and operate the ADMX6001-EBZ for data acquisition and visualization. A list of useful methods defined in this script for operating the setup of the ADMX6001-EBZ board and the Xilinx VCU118 board follows.

- ▶ **AD4080_CAL(self)**: configure the AD4080 LVDS data interface, Register 0x15 and Register 0x16, for correct data output.
- ▶ **set_dac_offset(self, voltage)**: set the DC offset in mV for the **AD9213** path. By setting the appropriate DC offset, the signal can be moved up or down for maximizing the dynamic range of the AD9213.
- ▶ **capture_data_ad9213(self, nsamples)**: the AD9213 captures data of nsamples at 10GSPS.
- ▶ **capture_data_ad4080(self, nsamples)**: the **AD4080** captures data of nsamples at 31.25MSPS.
- ▶ **plot_data_ad9213(self, data)**: plot data captured by the AD9213.
- ▶ **plot_data_ad4080(self, data)**: plot data captured by by AD4080.

```

Command Prompt - python.exe ADMX6001_acquisition.py
(mysenv) c:\ADMX6001_PythonScripts_JUN02\ExampleForRelease>python.exe ADMX6001_acquisition.py
DC-coupling reg (1617): 0x1
Cal freeze reg (1681): 0x1
ADC cal code
starting value=
[-342570 -342570 -342570 ... -342570 -342570 -342570]
0x21
AD4080 Already Calibrated
Press any key to continue ...
DC-coupling reg (1617): 0x1
Cal freeze reg (1681): 0x1
This plot has been saved as AD9213_plot.pdf in the current path.
This plot of AD9213 data and spectrum has been saved as AD9213DataSpectrum.png
AD9213: The primary frequency index 13
AD9213: The primary frequency is 1983642.578125 Hz
This plot has been saved as AD9213_plot.pdf in the current path.
This plot of AD9213 data with different offsets has been saved as AD9213_offsets.png
AD4080: The primary frequency index 33
AD4080: The primary frequency is 2014160.1562499998 Hz
This plot has been saved as AD4080_plot.pdf in the current path.
This plot of AD4080 data and spectrum has been saved as AD4080DataSpectrum.png
End of this example script. Press any key to close ...
  
```

Figure 49. Run the Python Script **ADMX6001_acquisitionI.py** to Acquire and Visualize Data

The example script **ADMX6001_acquisitionI.py** creates instance of classes defined in **ADMX6001_multiClass_pCal.py** and calls the methods to initialize the ADMX6001-EBZ board, calibrate the AD4080 LVDS data interface, and perform data acquisition.

Figure 50 shows the Python script **ADMX6001_acquisitionI.py**.

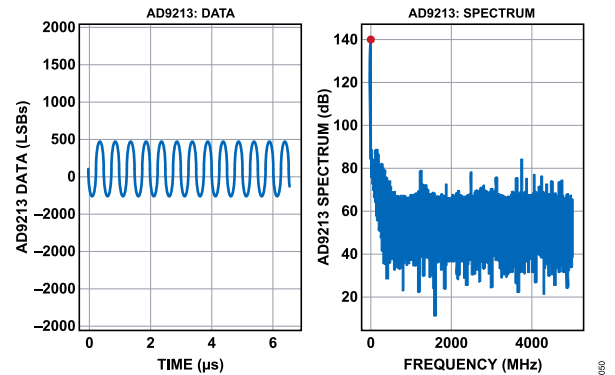


Figure 50. Plots Generated by Python Script **ADMX6001_acquisitionI.py**

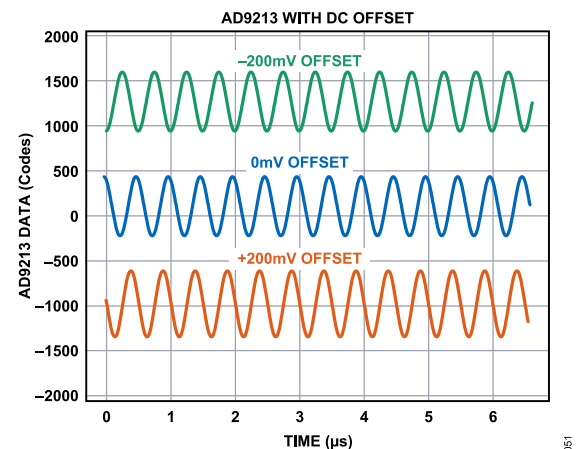


Figure 51. AD9213 Data with Different DC Offsets

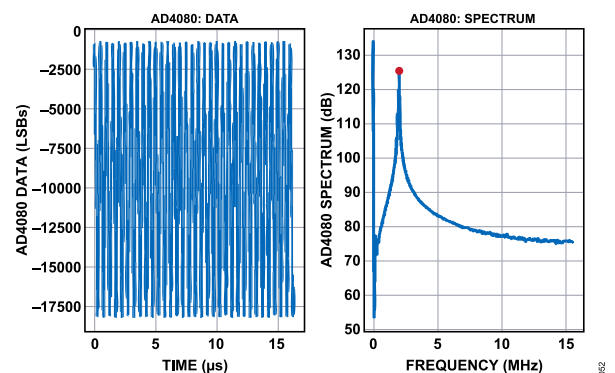


Figure 52. AD4080: Data and AD4080: Spectrum

NOTES

**ESD Caution**

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

Legal Terms and Conditions

By using the evaluation board discussed herein (together with any tools, components documentation or support materials, the "Evaluation Board"), you are agreeing to be bound by the terms and conditions set forth below ("Agreement") unless you have purchased the Evaluation Board, in which case the Analog Devices Standard Terms and Conditions of Sale shall govern. Do not use the Evaluation Board until you have read and agreed to the Agreement. Your use of the Evaluation Board shall signify your acceptance of the Agreement. This Agreement is made by and between you ("Customer") and Analog Devices, Inc. ("ADI"), with its principal place of business at One Analog Way, Wilmington, MA 01887-2356, U.S.A. Subject to the terms and conditions of the Agreement, ADI hereby grants to Customer a free, limited, personal, temporary, non-exclusive, non-sublicensable, non-transferable license to use the Evaluation Board FOR EVALUATION PURPOSES ONLY. Customer understands and agrees that the Evaluation Board is provided for the sole and exclusive purpose referenced above, and agrees not to use the Evaluation Board for any other purpose. Furthermore, the license granted is expressly made subject to the following additional limitations: Customer shall not (i) rent, lease, display, sell, transfer, assign, sublicense, or distribute the Evaluation Board; and (ii) permit any Third Party to access the Evaluation Board. As used herein, the term "Third Party" includes any entity other than ADI, Customer, their employees, affiliates and in-house consultants. The Evaluation Board is NOT sold to Customer; all rights not expressly granted herein, including ownership of the Evaluation Board, are reserved by ADI. CONFIDENTIALITY. This Agreement and the Evaluation Board shall all be considered the confidential and proprietary information of ADI. Customer may not disclose or transfer any portion of the Evaluation Board to any other party for any reason. Upon discontinuation of use of the Evaluation Board or termination of this Agreement, Customer agrees to promptly return the Evaluation Board to ADI. ADDITIONAL RESTRICTIONS. Customer may not disassemble, decompile or reverse engineer chips on the Evaluation Board. Customer shall inform ADI of any occurred damages or any modifications or alterations it makes to the Evaluation Board, including but not limited to soldering or any other activity that affects the material content of the Evaluation Board. Modifications to the Evaluation Board must comply with applicable law, including but not limited to the RoHS Directive. TERMINATION. ADI may terminate this Agreement at any time upon giving written notice to Customer. Customer agrees to return to ADI the Evaluation Board at that time. LIMITATION OF LIABILITY. THE EVALUATION BOARD PROVIDED HEREUNDER IS PROVIDED "AS IS" AND ADI MAKES NO WARRANTIES OR REPRESENTATIONS OF ANY KIND WITH RESPECT TO IT. ADI SPECIFICALLY DISCLAIMS ANY REPRESENTATIONS, ENDORSEMENTS, GUARANTEES, OR WARRANTIES, EXPRESS OR IMPLIED, RELATED TO THE EVALUATION BOARD INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, TITLE, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. IN NO EVENT WILL ADI AND ITS LICENSORS BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES RESULTING FROM CUSTOMER'S POSSESSION OR USE OF THE EVALUATION BOARD, INCLUDING BUT NOT LIMITED TO LOST PROFITS, DELAY COSTS, LABOR COSTS OR LOSS OF GOODWILL. ADI'S TOTAL LIABILITY FROM ANY AND ALL CAUSES SHALL BE LIMITED TO THE AMOUNT OF ONE HUNDRED US DOLLARS (\$100.00). EXPORT. Customer agrees that it will not directly or indirectly export the Evaluation Board to another country, and that it will comply with all applicable United States federal laws and regulations relating to exports. GOVERNING LAW. This Agreement shall be governed by and construed in accordance with the substantive laws of the Commonwealth of Massachusetts (excluding conflict of law rules). Any legal action regarding this Agreement will be heard in the state or federal courts having jurisdiction in Suffolk County, Massachusetts, and Customer hereby submits to the personal jurisdiction and venue of such courts. The United Nations Convention on Contracts for the International Sale of Goods shall not apply to this Agreement and is expressly disclaimed. All Analog Devices products contained herein are subject to release and availability.

