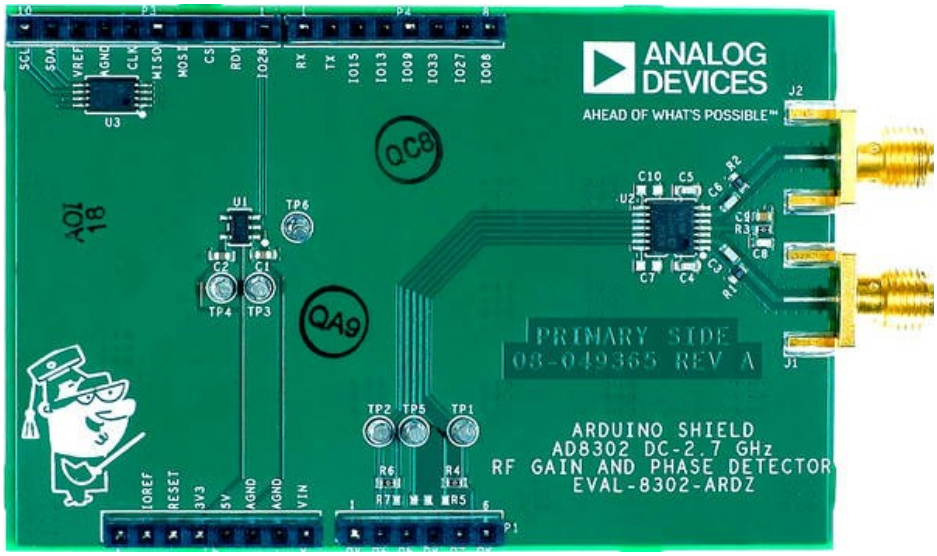
This version (17 Sep 2018 08:12) was ***approved*** by adrianmtolentino.
The Previously approved version (08 Sep 2018 20:23) is available.

EVAL-AD8302-ARDZ



The EVAL-AD8302-ARDZ shield illustrates the functionality of the AD8302, a gain and phase detector which operates from low frequency up to 2.7 GHz. The voltage outputs of the AD8302 are routed to the ANALOG IN connector of the Arduino base board. This allows the RF power detector's output voltage to be easily digitized and processed by the Arduino base board's integrated six-channel ADC.

The power supply for the board comes from the Arduino base board through the POWER connector (5V). So there is no need to connect an external power supply.

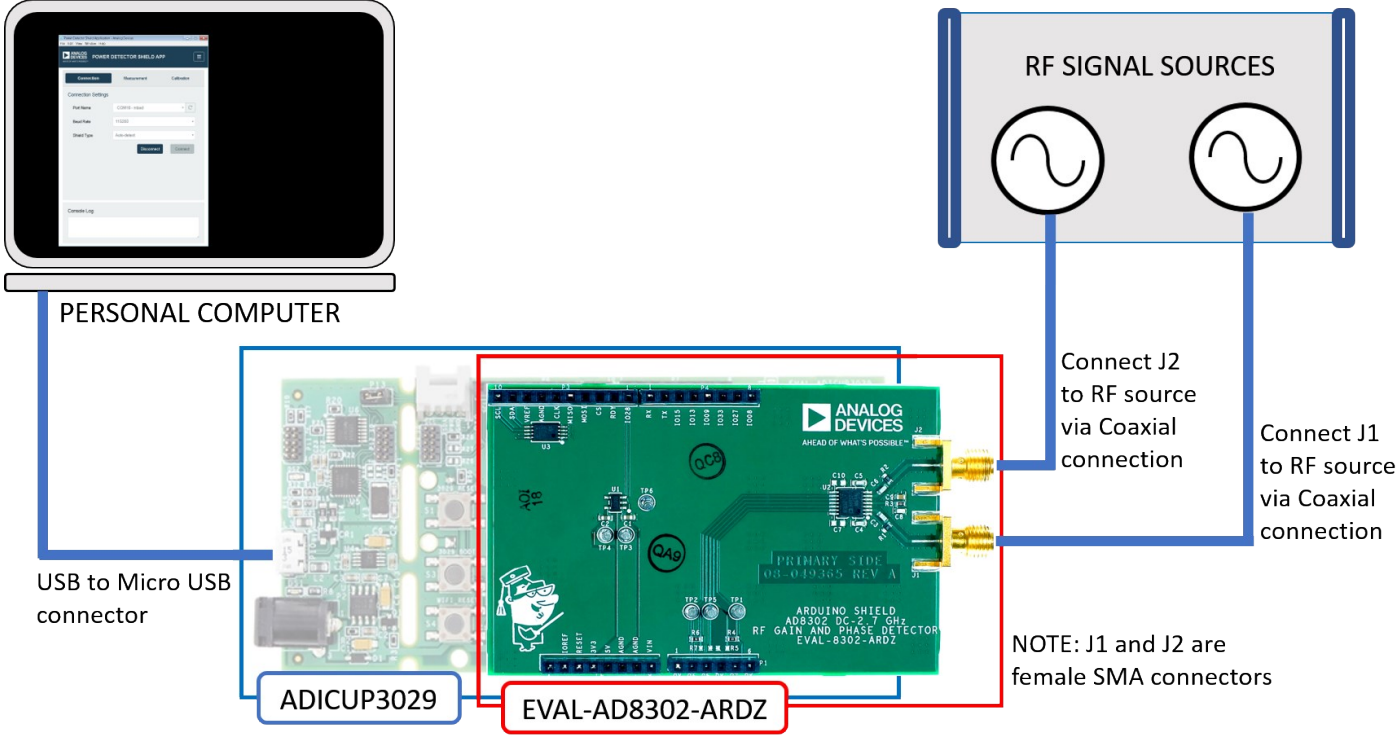
The EVAL-AD8302-ARDZ was designed to work with EVAL-ADICUP3029 and Linduino. For both platforms, a PC software GUI and device drivers are available.

Shield Specifications

- Supply:
    1. Voltage: 5V
    2. Operates at around 35mA
- Input RF Power Range: -60dBm to 0dBm
- Input RF Frequency Range: DC to 2.7GHz
- Has power down interface

Setting Up the Hardware

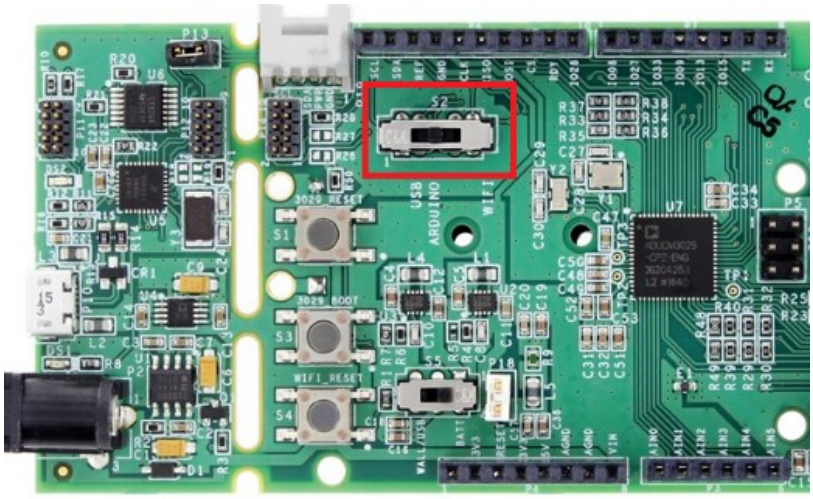
Table of Contents



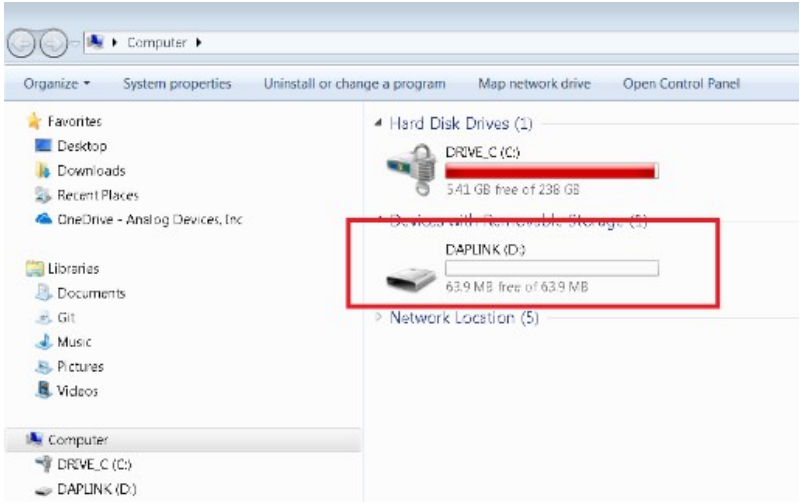
Software GUI for ADICUP3029

Software Installation

1. Download the Software GUI file here.
2. Extract the Software GUI file to your computer.
3. Connect the Eval-ADICUP3029 board using micro USB cable.
4. Set the S2 switch to USB.



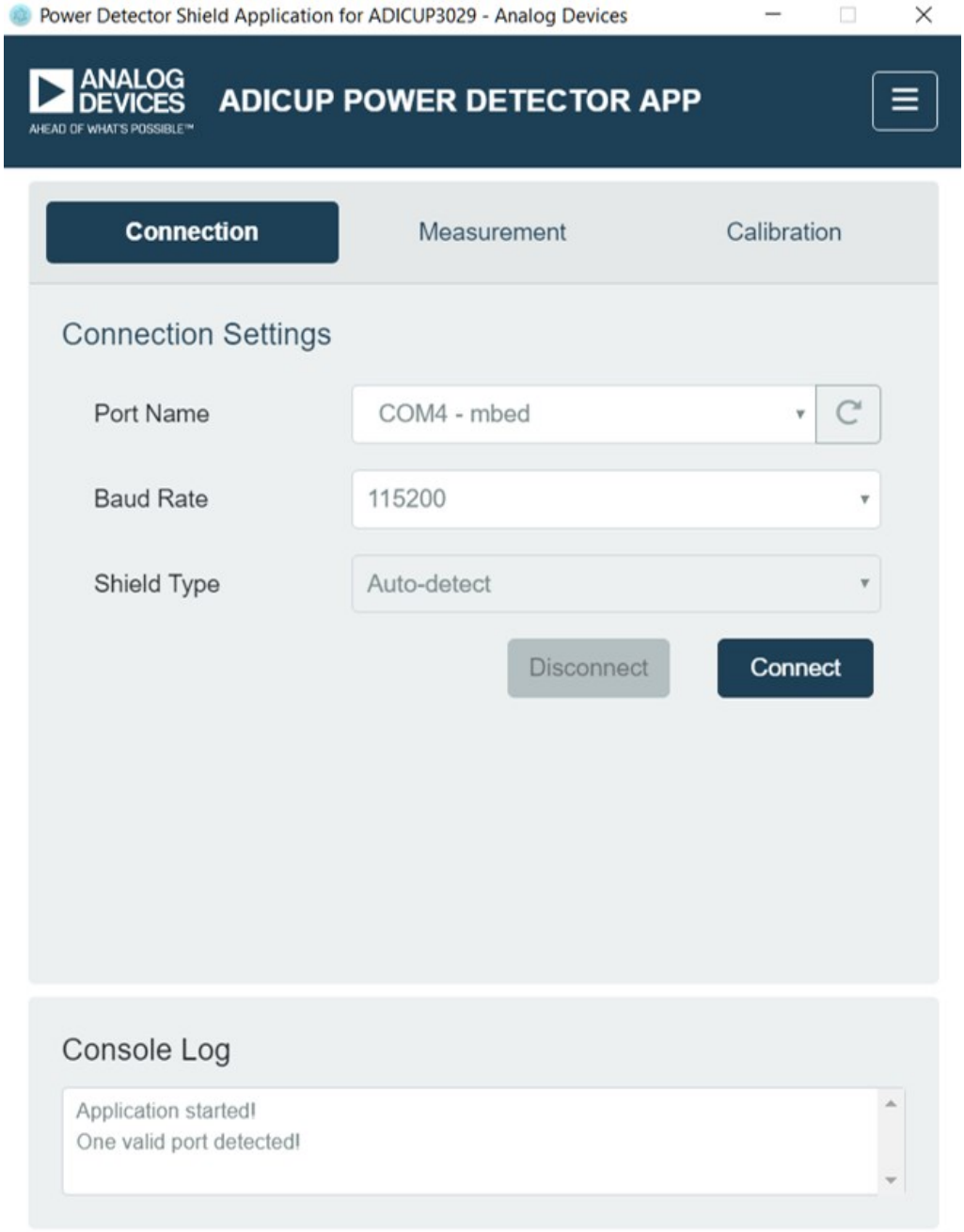
1. In the extracted files look for “power_detector-firmware.hex” then copy the hex file to Computer»DAPLINK drive



After loading the hex file to the DAPLINK drive the window explorer must automatically

close or else you need to load the hex file to the drive again.

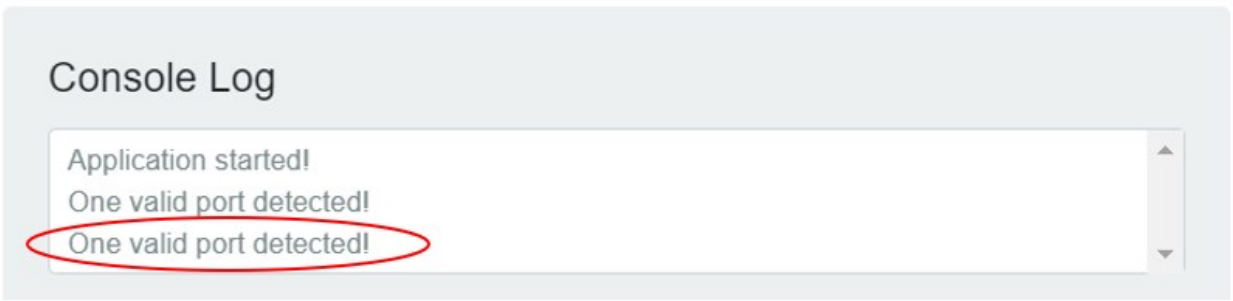
1. After the windows explorer automatically closes, reset the Eval-ADICUP3029 board by pressing the S1 (reset) button on the board.
2. Go to extracted files and look for “power_detector.exe” file and double click to run the software. The Connection Window will open.



Software Operation

Connection Window

1. Mount EVAL-AD8302-ARDZ to the ADICUP3029 and connect ADICUP3029 to computer as in Typical Hardware Setup for Measurement
2. Click the refresh button on Port Name to Identify the port where an ADICUP3029 is installed

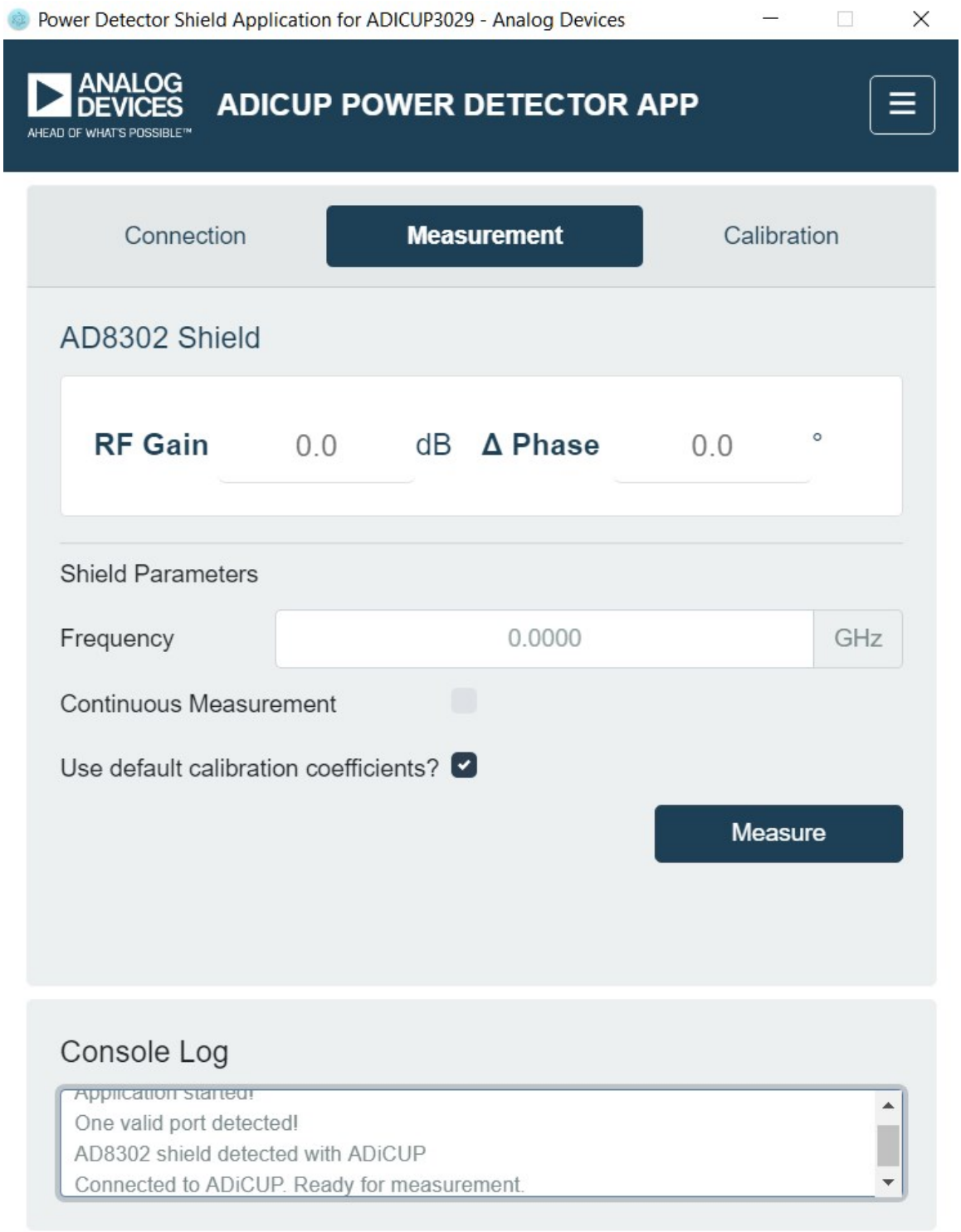


If there are many ADICUP3029 installed, select the port where ADICUP3029 and EVAL-AD8302-ARDZ connected

3. Set Baudrate to 115200
4. Select Auto-detect on Shield type.
5. Click Connect. The Measurement Window should Open.

Console Log must indicate “AD8302shield detected with ADiCUP”

Measurement Window



The shield measures Gain and Phase Difference based on a 2-point calibrated linear response characterized for a specific frequency. By using default calibration coefficients, the 2-point linear response corresponds to the datasheet specifications of AD8302. By using the user calibration coefficients, the frequency dependent 2-point linear response corresponds to the calibration made by the user.

The user calibration coefficients and default calibration coefficients are INITIALLY the same. Therefore any unchanged calibration at specific frequencies in the user calibration coefficients retains the default values

Related topic: Calibration of EVAL-AD8302-ARDZ

Select Calibration Coefficients:

- Check the box to use default calibration coefficients
- Uncheck to use user calibration coefficients

To make single measurement:

1. Enter the frequency of the input RF signal
2. Uncheck Continuous Measurement
3. Click Measure Button

Not entering the correct frequency may result to less accurate measurements.

To continuously make measurements:

1. Enter the frequency of the input RF signal
2. Check Continuous Measurement
3. Click Measure Button
4. Click Stop to stop measuring at the last measurement

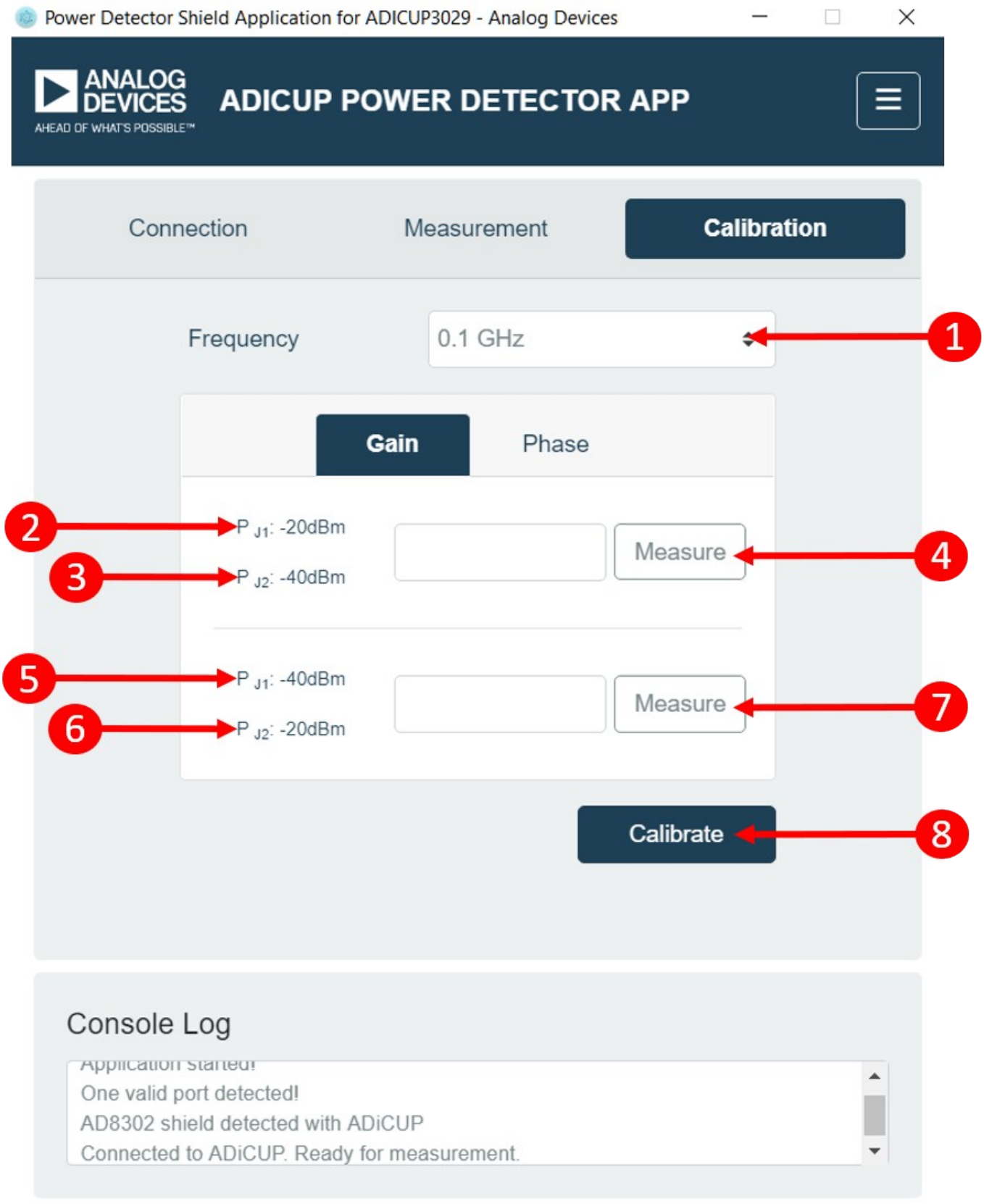
Not entering the correct frequency may result to less accurate measurements.

To switch windows:

Click “Connection” or “Calibration” to switch to respective window.

Calibration Window

- Gain Calibration



1. Select the frequency
2. Input to J1 an RF signal of-20dBm
3. Input to J2 an RF signal of-40dBm
4. Click “Measure”
5. Input to J1 an RF signal of-40dBm
6. Input to J2 an RF signal of-20dBm
7. Click “Measure”
8. Click “Calibrate”

- Phase Calibration

1. Select frequency
2. Set the signal power of inputs to -30dBm
3. Set phase between inputs to be 45º
4. Click measure
5. Set phase between inputs to be 135º
6. Click measure
7. Click “Calibrate”

User must be able to synchronize the phase of input signals perhaps, by using external devices/equipment, to do this calibration

If desired frequency of calibration or measurement is not on the list, calibrate on the immediate higher frequency available and on the immediate lower frequency available. If desired frequency is higher/lower than the available frequency selection, calibrate only on the highest/lowest frequency selection

Note on Calibration

Calibration can be implemented using 2, 3, or 4-point calibration techniques which can be used to approximate nearly linear response characteristics such as in AD8302. A typical characteristic of the AD8302 at 2.14GHz input is shown in Figure 1. This is Figure 50 from the AD8302 datasheet.

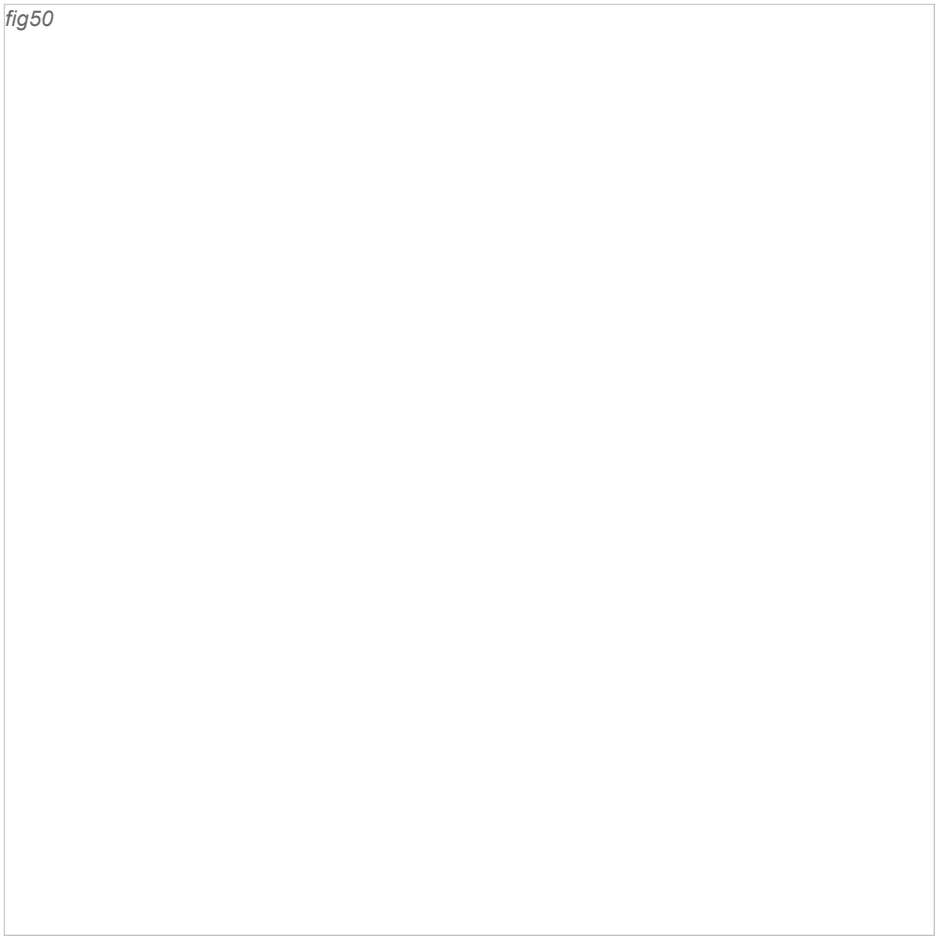


Figure 1. AD8302 Characteristic Response at 2.14GHz

Two-point calibration creates an approximated response characteristic utilizing two points on the typical characteristic line. By choosing two points, (VOUT1,INPUT1) and (VOUT2,INPUT2), from the typical response characteristic, a line using two point form can be obtained and is given by:

SLOPE1 = (VOUT_1 – VOUT_2)/(INPUT_1 − INPUT_2)

From this equation, the point intercept form of the approximated response characteristic is given by:

INTERCEPT1= VOUT_1/(SLOPE1 × INPUT_1)

This derives the INTERCEPT1. Given the SLOPE1 and INTERCEPT1, any point (INPUT,VOUT) along the approximated line is defined in the equation:

VOUT = SLOPE1 × (INPUT − INTERCEPT1)

The range of INPUT is the device's dynamic range. SLOPE1 is in mV/dB and INTERCEPT1 is in dBm.

To implement three-point calibration, suppose three points on the typical response characteristic, (INPUT1,VOUT1),(INPUT2,VOUT2), and (INPUT3,VOUT3), such that INPUT1

This is also applicable by using ADC codes instead of Vout.

Development on ADICUP3029

Development packages are available for C and Python. Other development environments may be used but this development guided is focused on software development on CrossCore Embedded Studio (for C) and on Pycharm(for Python).

C Development Guide

Installations

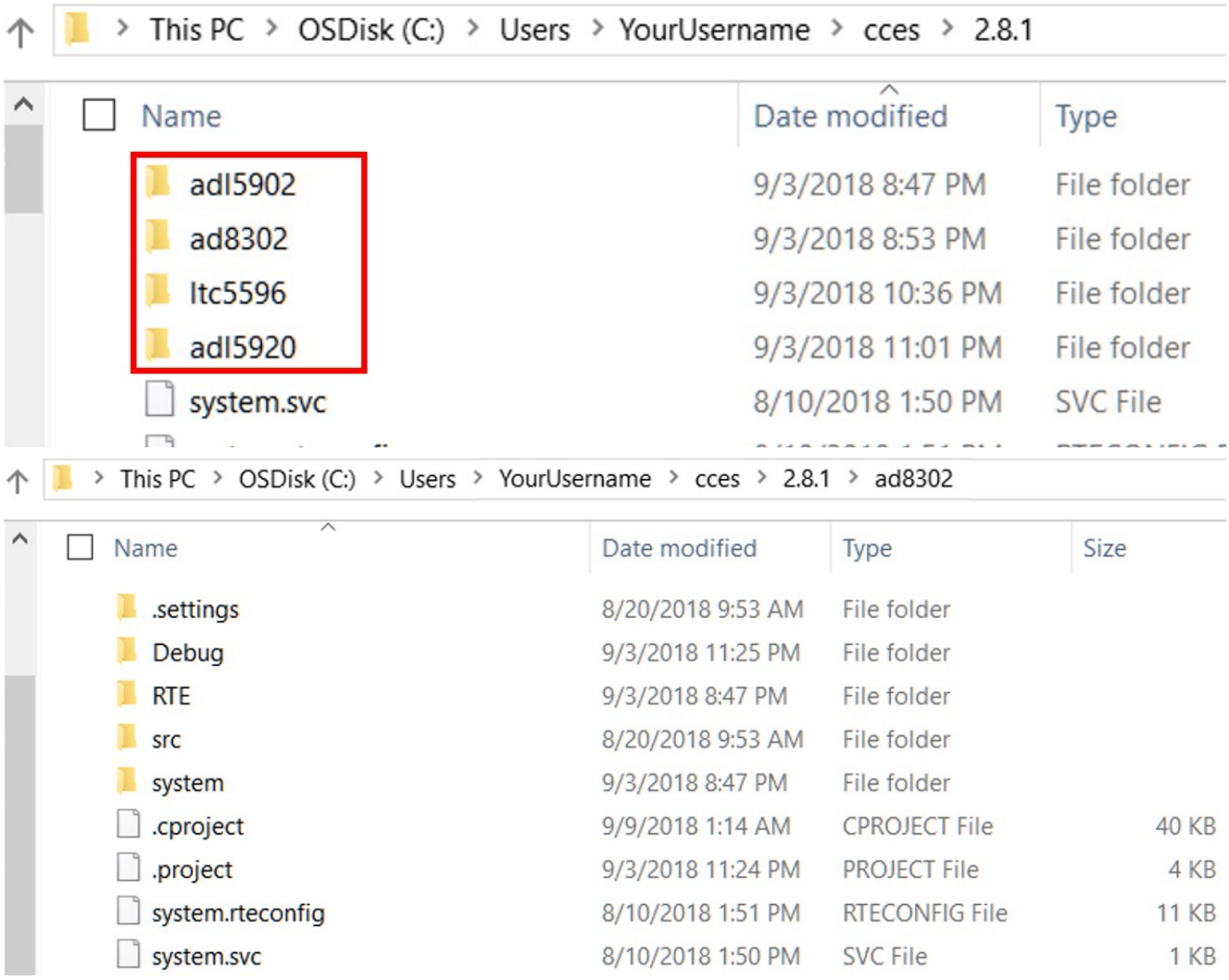
1. Download and install **CrossCore Embedded Studio (CCES) 2.8.1**
2. Download and install **mBed windows serial driver**

> Assumes a fresh installation of all required software

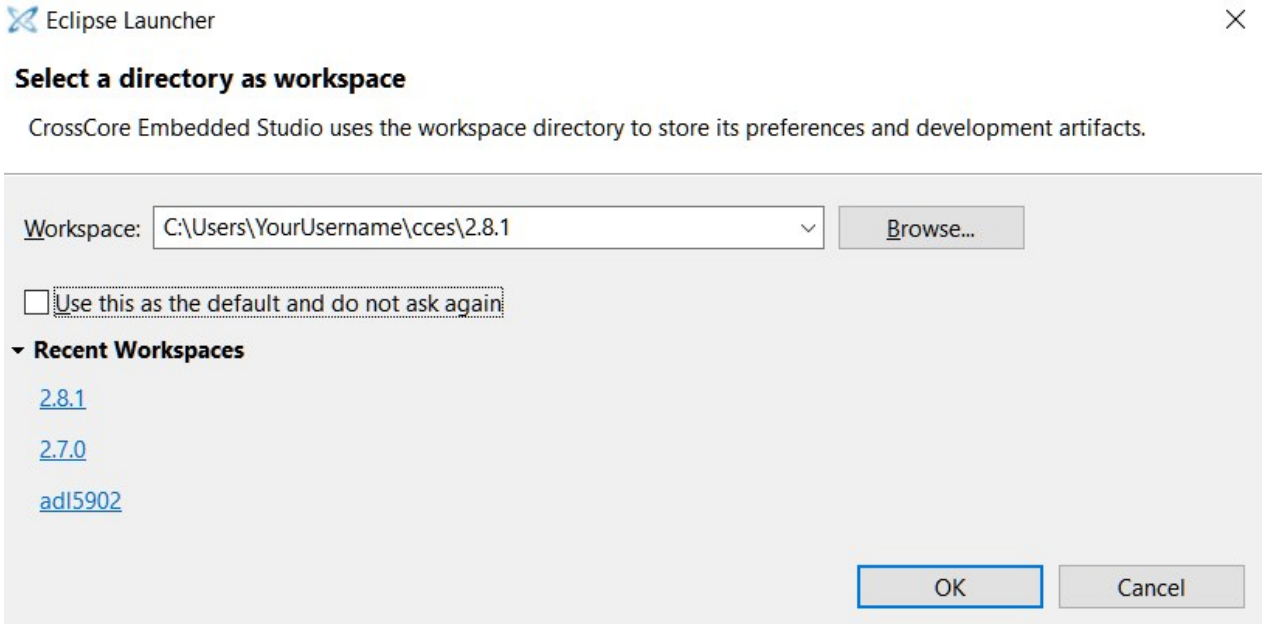
Setting Up CrossCore Embedded Studio

1. Install the following packs by following the **How to install or upgrade Packs for CCES** guide:
   - **ARM.CMSIS.5.4.0**
   - **AnalogDevices.ADuCM302x_DFP.3.1.2**
2. Switch back to **C/C + + window** 

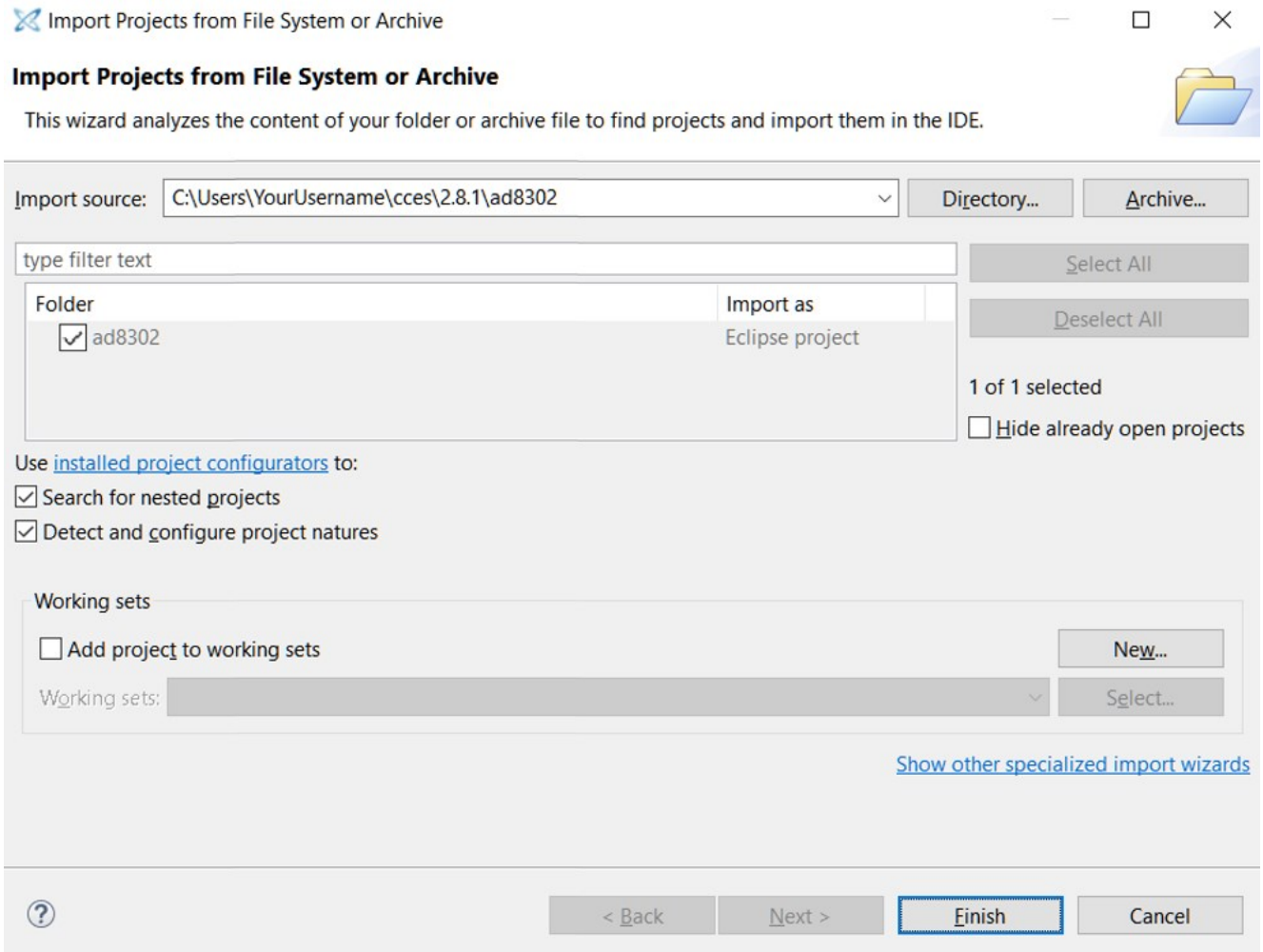
and close CCES 2.8.1
3. Download Dev Codes for Release.rar and unzip it.
4. Unzip ad8302.rar file to C:\Users\YourUsername\cces\2.8.1\ad8302. The contents of your unzipped folder should match the ones below.



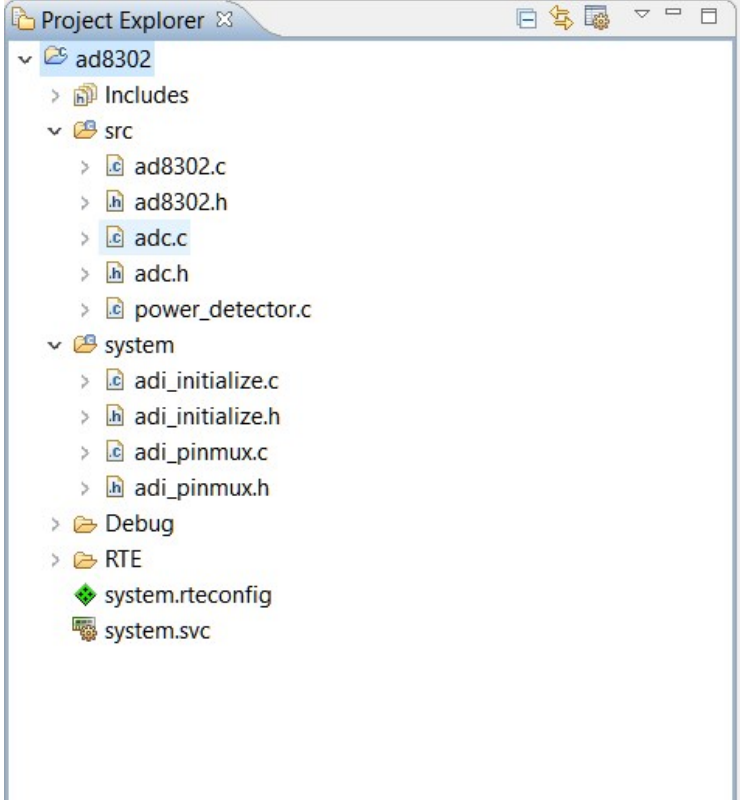
5. Launch CCES 2.8.1 and select workspace C:\Users\YourUsername\cces\2.8.1. If the ad8302.rar has been extracted elsewhere, choose that location as workspace. Switch to **C/C++ window** if it's not the current window.



6. To open the unzipped folder in the workspace, click **File** → **Open Projects from File System**. A new window will pop up and ask you to select the project or folder that you want to open. Select the proper directory then click **Finish**.

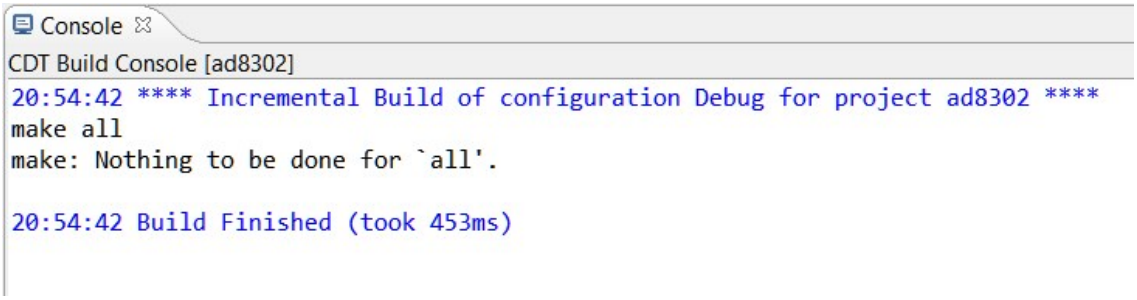


On the left side of the window, the structure of the loaded sample code should match the structure in the image shown below.



Development on CrossCore Embedded Studio

1. Setup Crosscore as in Setting Up CrossCore Embedded Studio
2. Connect your ADICUP3029 and power up the RF power detector shield then click Build .



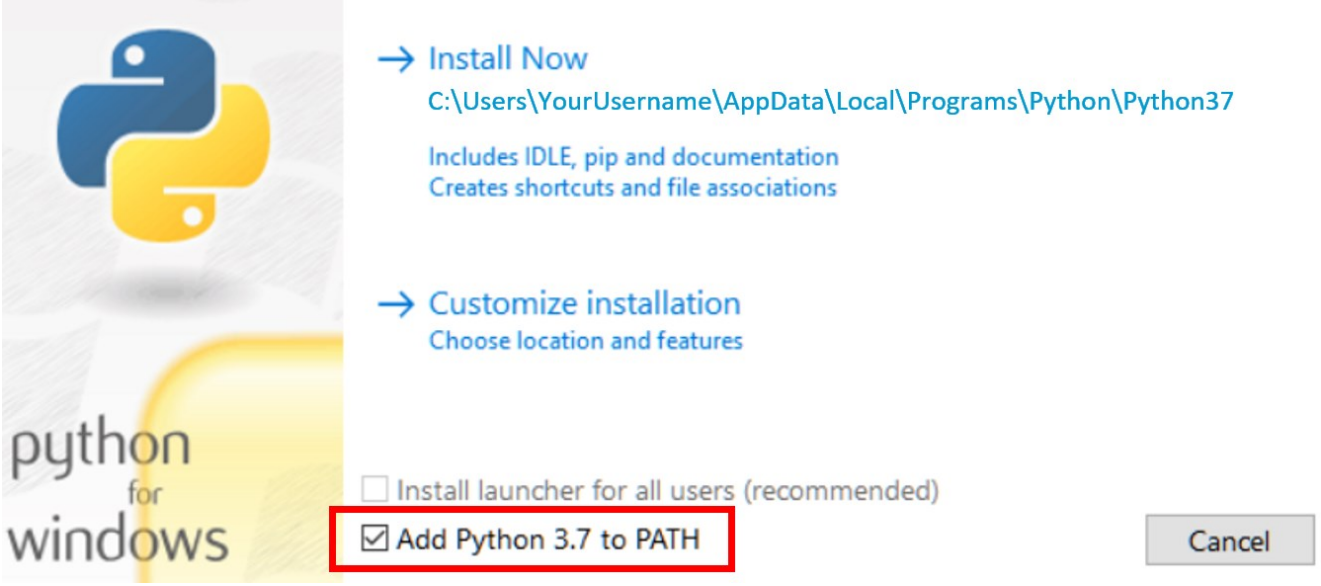
3. After it finishes building, click Debug and click Application with GDB and OpenOCD (Emulator). Copy the following Debug configurations on the new window that will appear then click the Debug button.
4. On the Debug window, click the Resume to run and display the results on the Console window.

Python Development Guide

Installations

Assumes a fresh installation of all required software

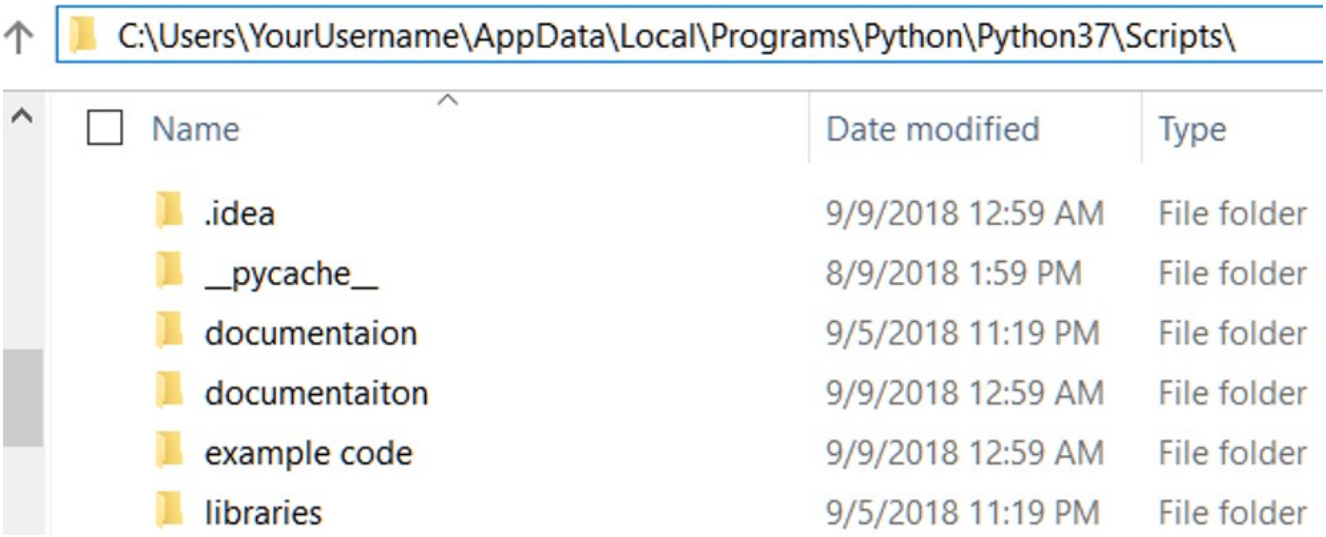
1. Download **python 3.7.0** version. Choose the right version depending on operating system. For windows, choose **Windows x86-64 executable installer**. (Do not run installer yet)
2. Run installer as Administrator. During installation, **check “Add Python 3.7 to PATH” before clicking “Install Now”**



3. Install **pyserial**. For windows, enter **pip3.7 install pyserial** on command prompt.
4. Download and install **PyCharm community version**
5. Download and install **mBed windows serial driver**

Setting Up PyCharm

1. Download **power detector.exe**
2. Install **power detector.exe** inside the "Scripts" directory where the python3.7 is located. For windows, the location path is similar to **C:\Users\MyUsername\AppData\Local\Programs\Python\Python37\Scripts**



3. Launch PyCharm and set up PyCharm interpreter by clicking file»settings»Project»Project Interpreter choose python 3.7 then click "Ok".

Development on PyCharm

1. Connect the Eval-ADICUP3029 board using micro USB cable.
2. In the Eval-ADICUP3029, set the S2 switch to USB.
3. Download **power_detector-firmware.hex**, then copy it to the DAPLINK directory. Wait for the window to exit automatically. Else, repeat the Development on PyCharm guide.
4. Press S1 (reset) button on the Eval-ADICUP3029 and mount the EVAL-AD8302-ARDZ to the Eval-ADICUP3029
5. On pyCharm, go to File»Open and browse for the **\PycharmProjects\example** code directory.
6. Click Project Tab located at left side of IDE and go to **ad8302** folder and double click **ad8302-getShieldReadings.py**
7. Change the default Port number ("COM10") in the example code. On your computer go to Control Panel»Device Manager look for Ports (COM & LPT) find the port number of "mbed Serial Port".
8. Right click on any point in the working space and click **Run ltc5596-getShieldReadings**

Software GUI for Linduino

Software Installation

Software Operation

Development on Linduino

Hardware Reference Information

Downloadable files contain the hardware reference information of EVAL-AD8302-ARDZ:

- Schematic Diagram of EVAL-AD8302-ARDZ
- Layout Design of EVAL-AD8302-ARDZ
- Bill of Materials of EVAL-AD8302-ARDZ
- Archive of .art files of EVAL-AD8302-ARDZ

resources/eval/user-guides/eval-ad8302-ardz.txt · Last modified: 17 Sep 2018 08:12 by adrianmtolentino