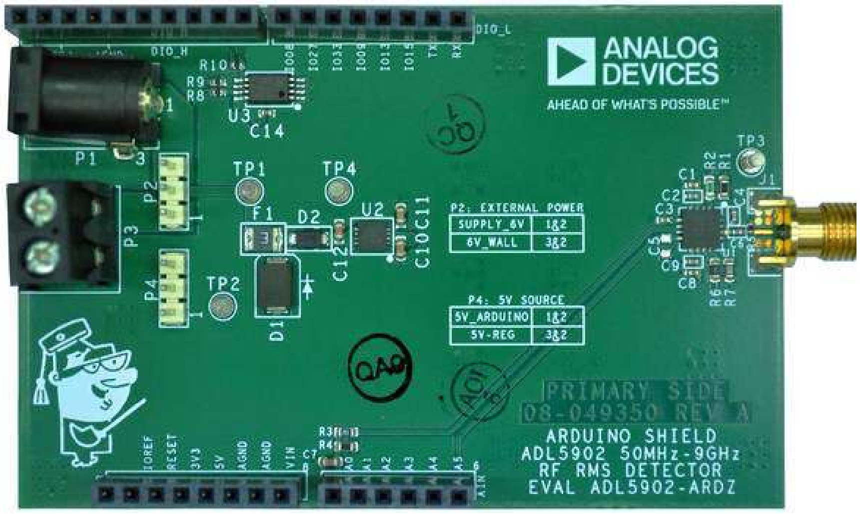
Wiki

Approvals: 0/1The Previously approved version (17 Aug 2018 14:29) is available.

EVAL-ADL5902-ARDZ

–Table of Contents



The EVAL-ADL5902-ARDZ shield illustrates the functionality of the **ADL5902, a 50 MHz to 9 GHz 65 dB TruPwr™ RMS responding RF power detector**. The voltage outputs of the ADL5902 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 output of the ADL5902's on-board temperature sensor is also routed to one of the ANALOG IN pins.

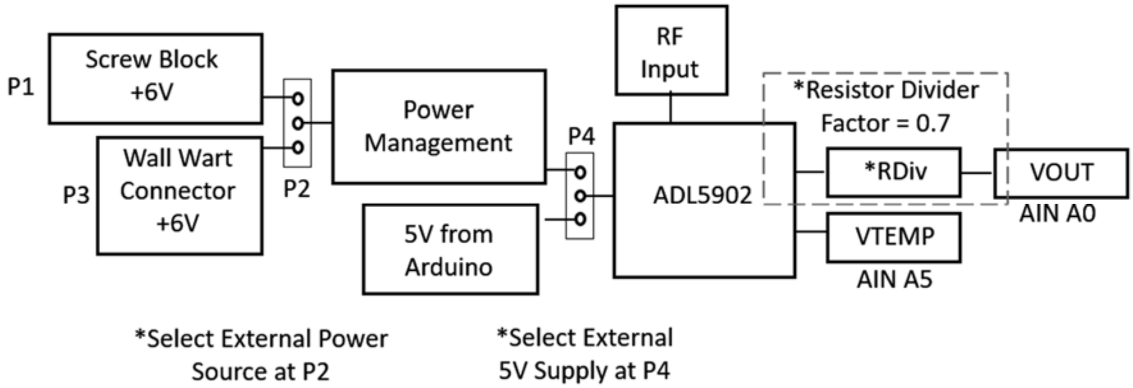
The **power supply** for the board comes from the Arduino base board through the POWER connector (5V). So there is **no need to connect an external power supply**.

The EVAL-ADL5902-ARDZ is compatible with **EVAL-ADICUP3029** and **Linduino**. For both platforms, **PC software GUI applications** (ADICUP3029, Linduino) are available using which, the user can make RF power measurements and also calibrate the device to decrease measurement error. **Device drivers** for ADICUP3029 and for Linduino Uno are also available, which the user may use to **develop their own code for RF measurement**, device calibration, and more.

Shield Specifications

- Supply:
    1. 5V Internal (short pin1 and pin2 of P4) – via microcontroller
    2. For operation without Arduino base board:
        1. 6V External supply (short pin1 and pin2 of P2; short pin2 and pin3 of P4)
        2. 6V Wall wart supply (short pin2 and pin3 of P2; short pin2 and pin3 of P4)
- Operates below 100mA
- Input Signal Maximum Power: 21dBm
- Input Dynamic Range: 65dB
- Linear only on approximately: -62dBm to 3dBm
- Input Frequency Range: 50MHz to 9GHz
- Input signal characteristic: Carrier (AC coupled), large crest factors (GSM, CDMA, W-CDMA, TD-SCDMA, WiMAX, and LTE modulated signals)
- Employs 3-point calibration
- Voltage Output Range:
    1. at VOUT: ~0.175V to ~2.45V
    2. at VTEMP: 1.1V to 1.8V

Functional Block Diagram

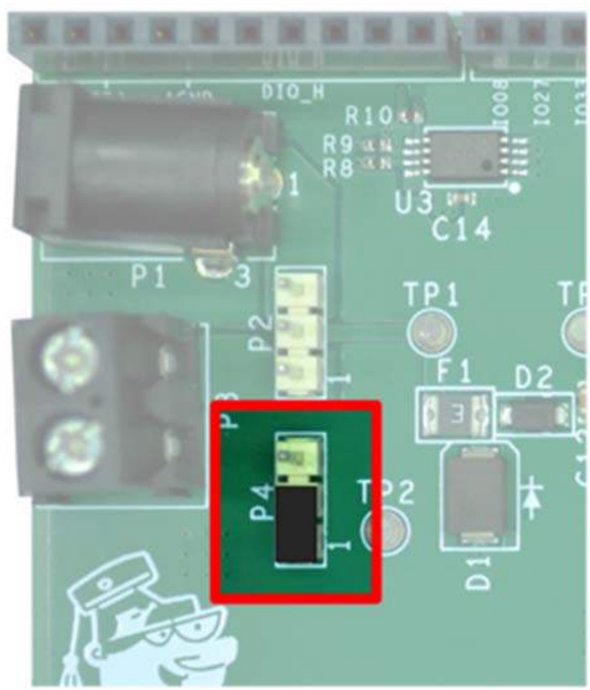


Setting Up the Hardware

Power Options Jumper Setting

Power up the EVAL-ADL5902-ARDZ using **any of the options** by shorting the correct pins using the provided shorting jumper caps.

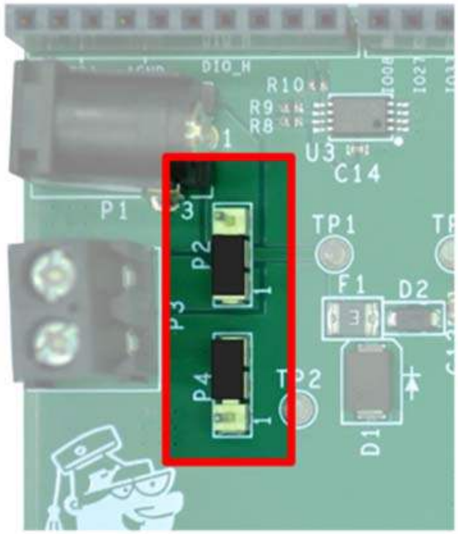
Option 1: 5V of ADICUP3029 or Linduino Uno



1. Connect **pin1 and pin2** of pin header **P4**.
2. Mount EVAL-ADL5902-ARDZ to ADICUP3029 or Linduino Uno.

This works regardless of the connections on pin header P2

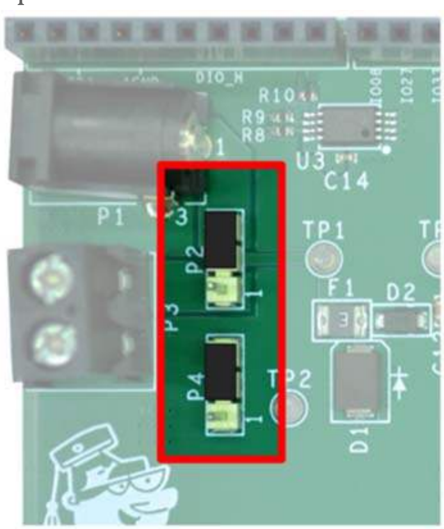
Option 2: 6V DC supply



1. Connect **pin2 and pin3** of pin header **P4**
2. Connect **pin1 and pin2** of pin header **P2**
3. Connect 6V to the EVAL-ADL5902-ARDZ via the **Screw terminal block**

EVAL-ADL5902-ARDZ is already functional using this option, even without ADICUP3029 or Linduino Uno

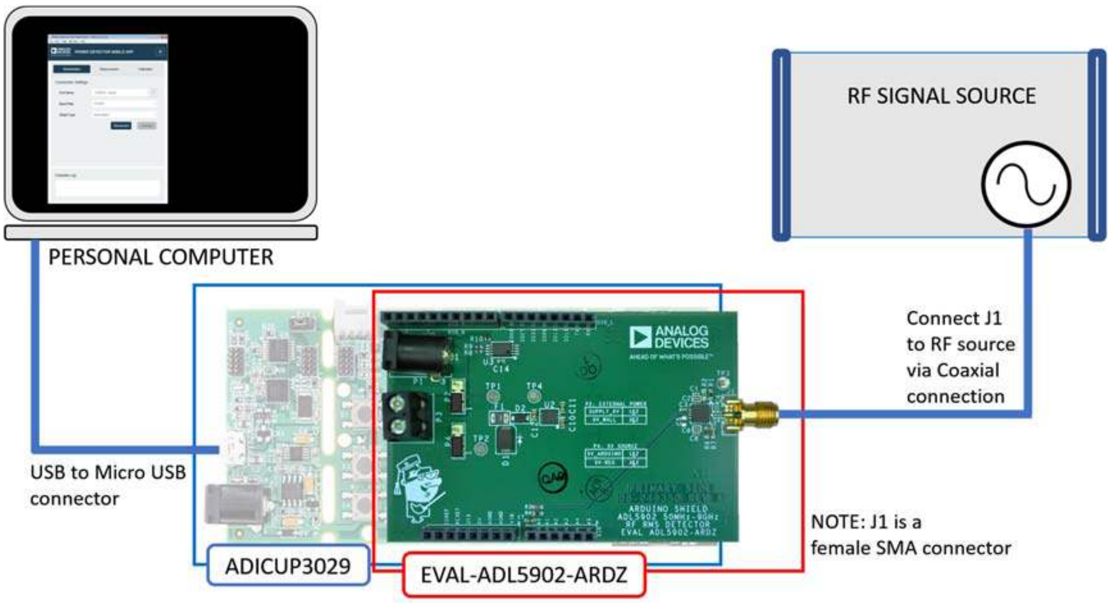
Option 3: 6V Wall wart



1. Connect **pin2 and pin3** of pin header **P4**
2. Connect **pin2 and pin3** of pin header **P2**
3. Connect 6V wall wart to the EVAL-ADL5902-ARDZ via the **DC Jack**

EVAL-ADL5902-ARDZ is already functional using this option, even without ADICUP3029 or Linduino Uno

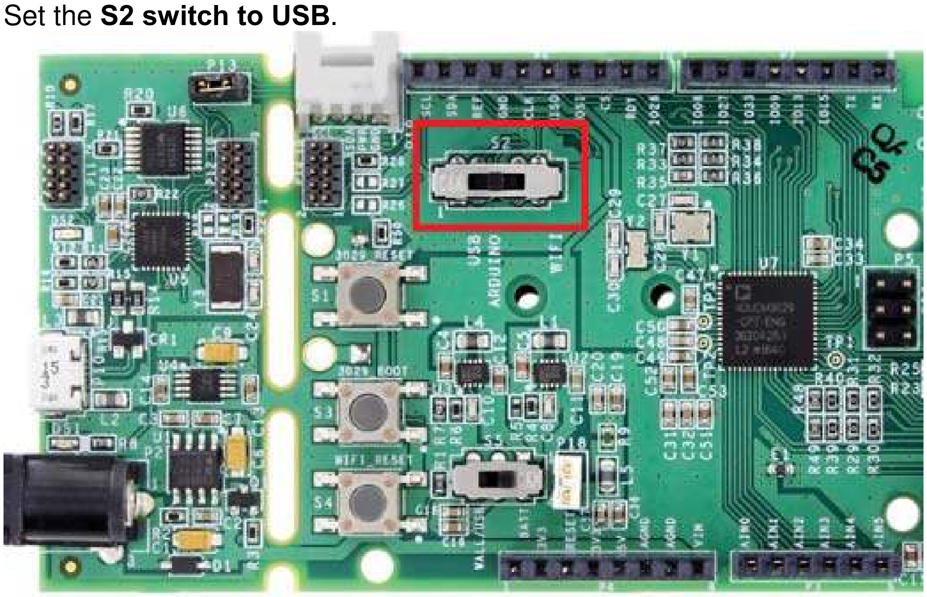
Typical Hardware Setup for measurement



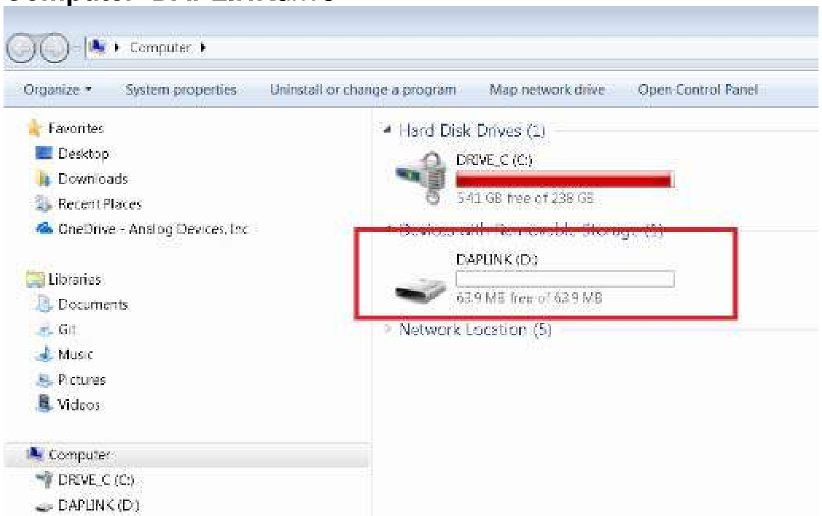
Software GUI for ADICUP3029

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**.

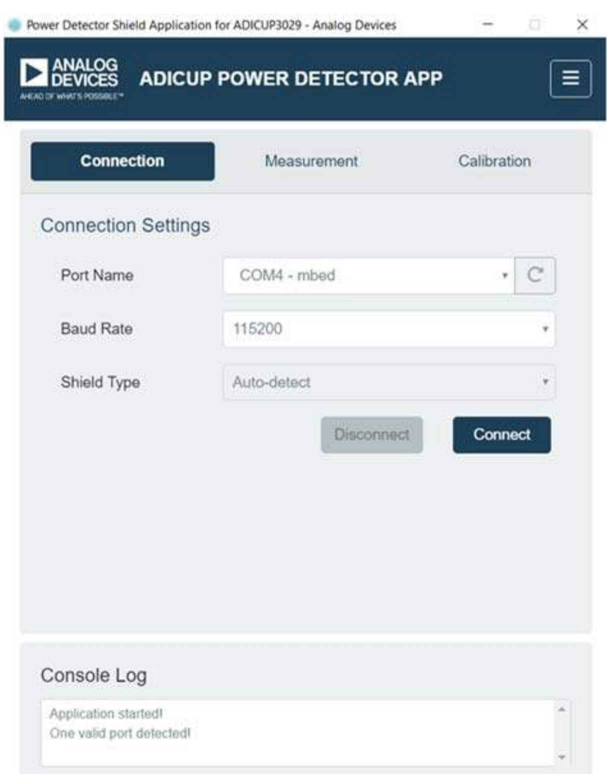


5. In the extracted files look for **power_detector-firmware.hex** then copy the hex file to **Computer»DAPLINK**drive



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.

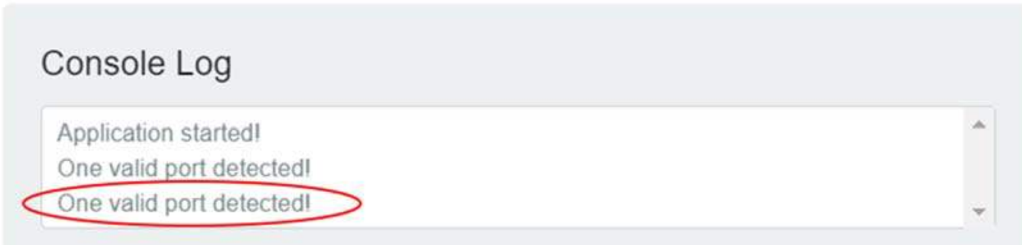
6. After the **windows explorer automatically closes**, reset the Eval-ADICUP3029 board by pressing the S1 (reset) button on the board.
7. Go to extracted files and look for **power_detector.exe** file and double click to run the software. The Connection Window will open.



Software Operation

Connection Window

1. Mount EVAL-ADL5902-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

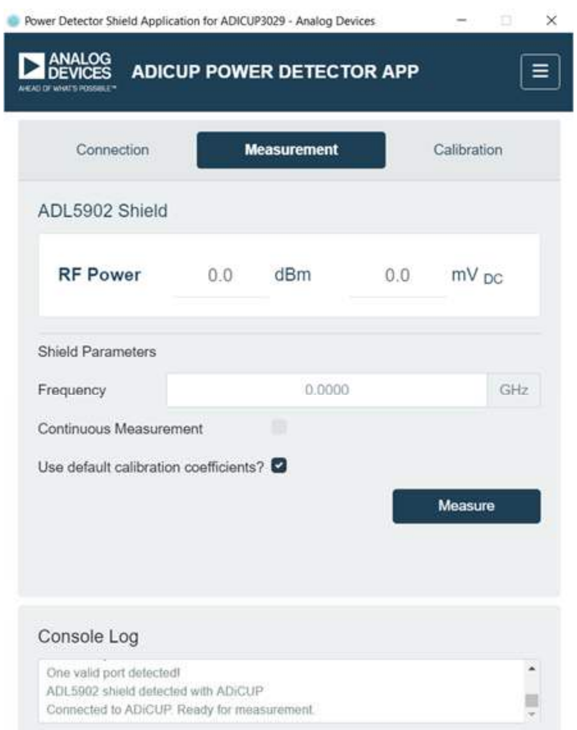


   If there are many ADICUP3029 installed, select the port where ADICUP3029 and EVAL-ADL5902-ARDZ connected

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

Console Log must indicate "ADL5902 shield detected with ADiCUP"

Measurement Window



The shield makes **RF power measurements based on a 3-point calibrated linear response** characterized by input RF frequency and power. By using **default calibration coefficients**, the 3-point linear response corresponds to the datasheet specifications of ADL5902. By using the user calibration coefficients, the 3-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-ADL5902-ARDZ

To 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.

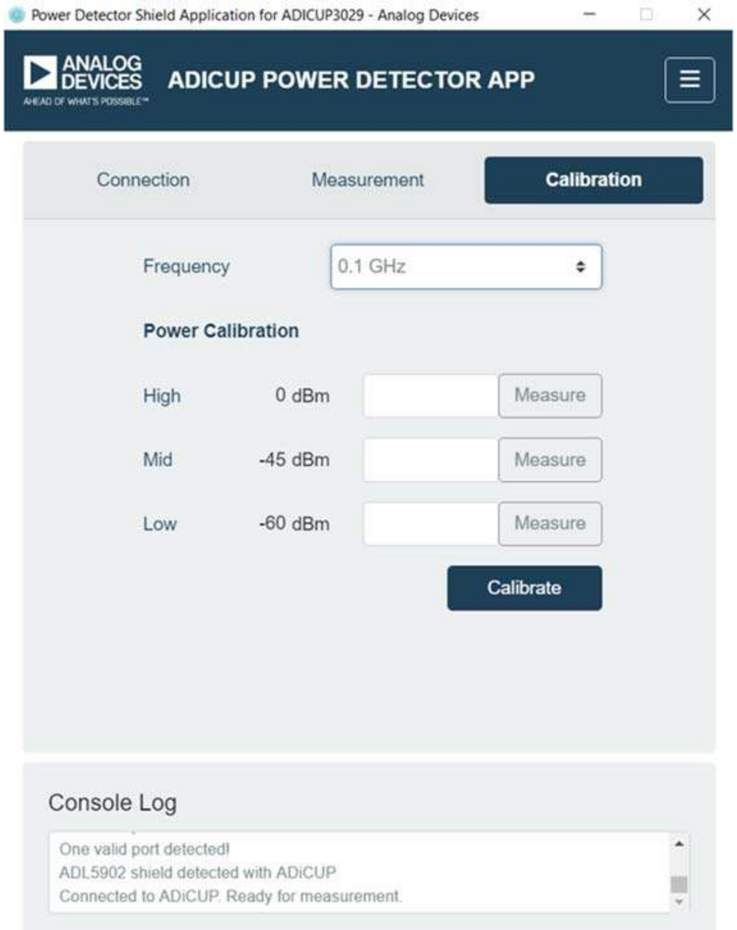
To continuously make measurements:

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



To calibrate at a specific frequency, to the following steps

1. Select the frequency
2. Input an RF signal of 0dBm power and of the selected frequency. Click the Measure Button across 0dBm.
3. Input an RF signal of -45dBm power and of the selected frequency. Click the Measure Button across -45dBm.
4. Input an RF signal of -60dBm power and of the selected frequency. Click the Measure Button across -60dBm.
5. Click Calibrate button. Console Log will indicate “User calibration coefficient for (frequency used) is updated.”

Follow steps strictly. User calibration coefficients will not update if the Calibrate Button is not clicked.

If making measurements or calibration at a frequency 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 ADL5902. A typical characteristic of the ADL5902 at 2.14GHz input is shown in Figure 1. This is Figure 50 from the ADL5902 datasheet.

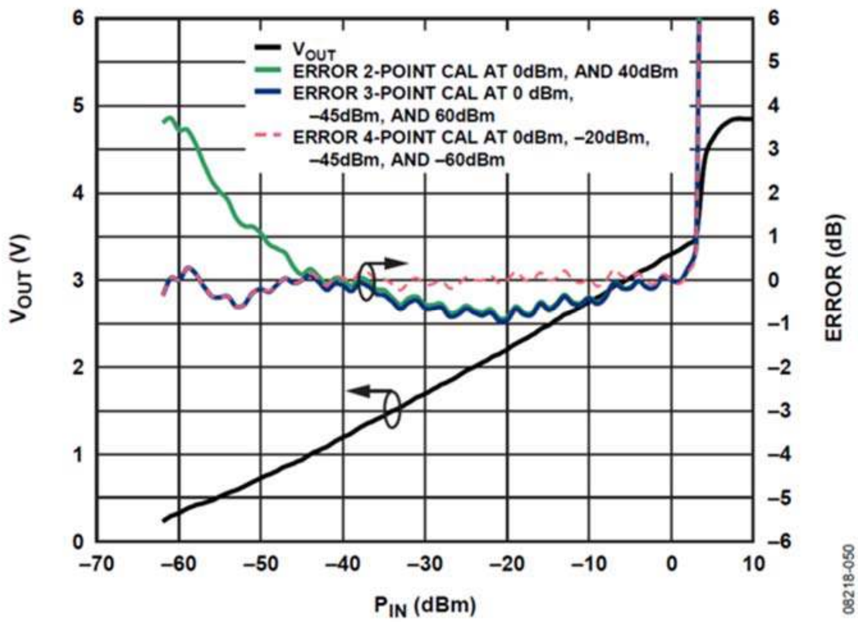


Figure 1. ADL5902 Characteristic Response at 2.14GHz

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** equation can be obtained and is given by:

```
SLOPE1 = (VOUT1 – VOUT2)/(INPUT1 – INPUT2)
```

This derives **SLOPE1**From this equation, the **point intercept form** of the approximated response characteristic is given by:

```
INTERCEPT1= VOUT1/(SLOPE1 × INPUT1)**
```

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<INPUT2<INPUT3**. Three-point calibration can be implemented by **applying the two-point calibration concept to (INPUT1,VOUT1) and (INPUT2,VOUT2), and applying it again to (INPUT2,VOUT2) and (INPUT3,VOUT3)**. For (INPUT1,VOUT1) and (INPUT2,VOUT2), **SLOPE1** and **INTERCEPT1** are derived to define a line, while for (INPUT2,VOUT2) and (INPUT3,VOUT3), **SLOPE2** and

INTERCEPT2 are derived to define another line. This makes a **piecewise approximation using the two lines** derived; the first is valid for INPUT of -62dBm to INPUT2, and the other is valid for INPUT2 to 3dBm. This technique can further be expanded to four point to implement four-point calibration.

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

Development on ADICUP3029

Development drivers 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

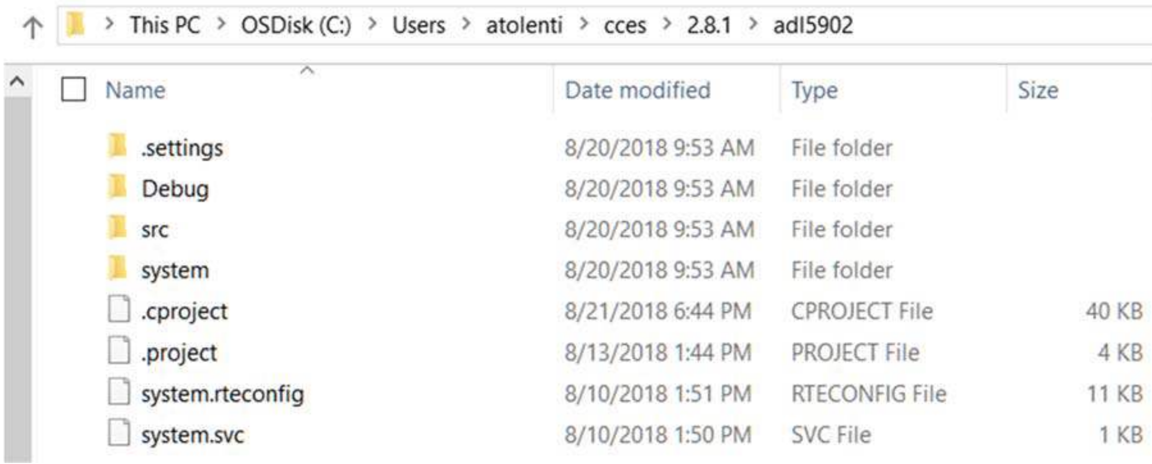
1. Download and install **CrossCore Embedded Studio (CCES) 2.8.1**
2. Download and install **mBed windows serial driver**

Assumes a fresh installation of all required software

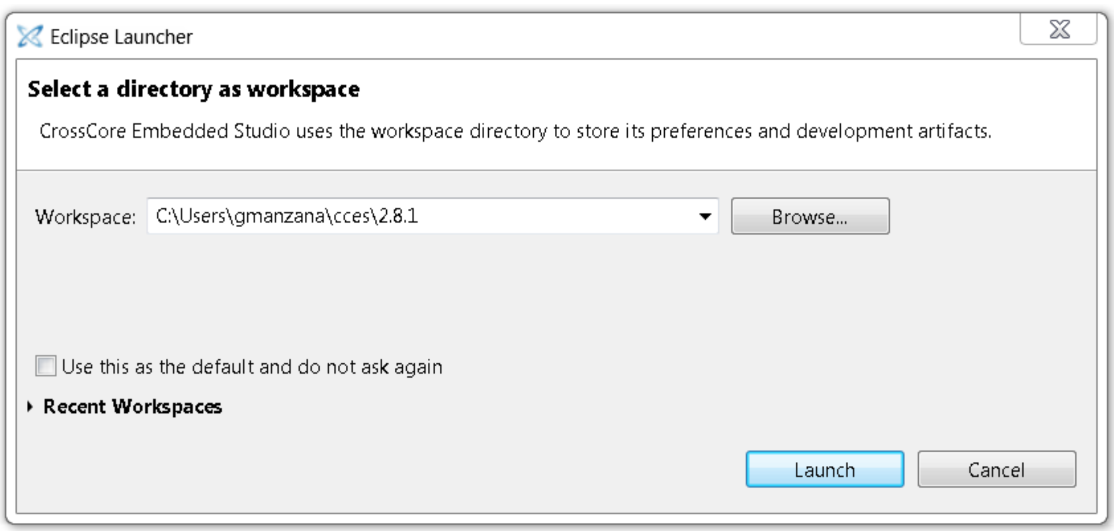
Setting Up CrossCore Embedded Studio

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 adl5902.rar file to C:\Users\YourUsername\cces\2.8.1\adl5902. The contents of your unzipped folder should match the ones below.

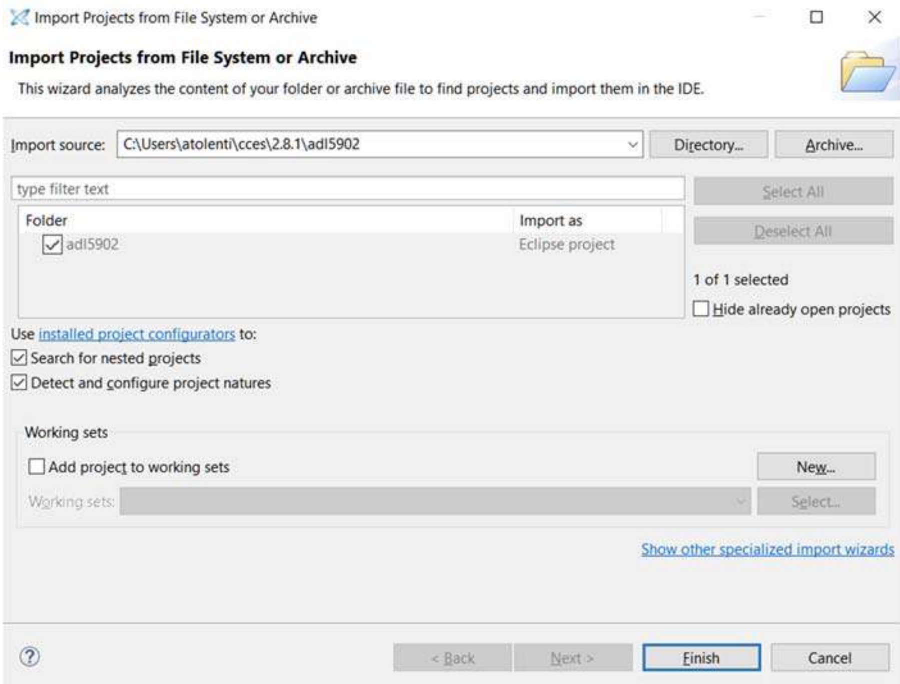


5. Launch CCES 2.8.1 and select workspace C:\Users\YourUsername\cces\2.8.1. If the adl5902.rar has been extracted elsewhere, choose that location as workspace. Switch to **C/C++ window** if it's not the current window.



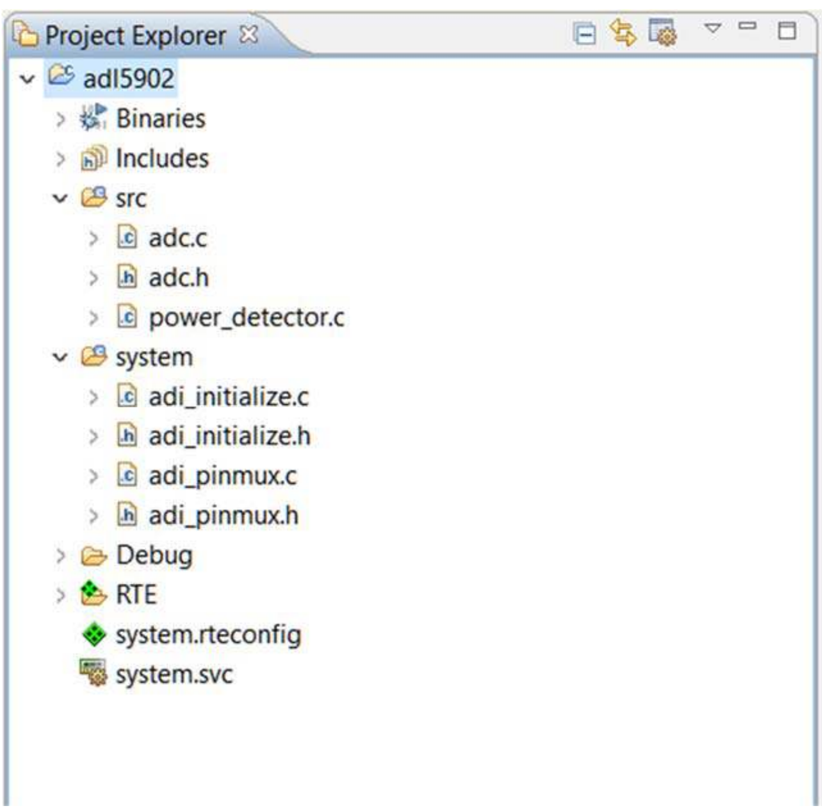
6. To open the unzipped folder in the workspace, click **File** → **Open Projects from File System**. A new window will pop up and ask you to select the project or folder that you want to open. Select the proper directory then click **Finish**.

7.



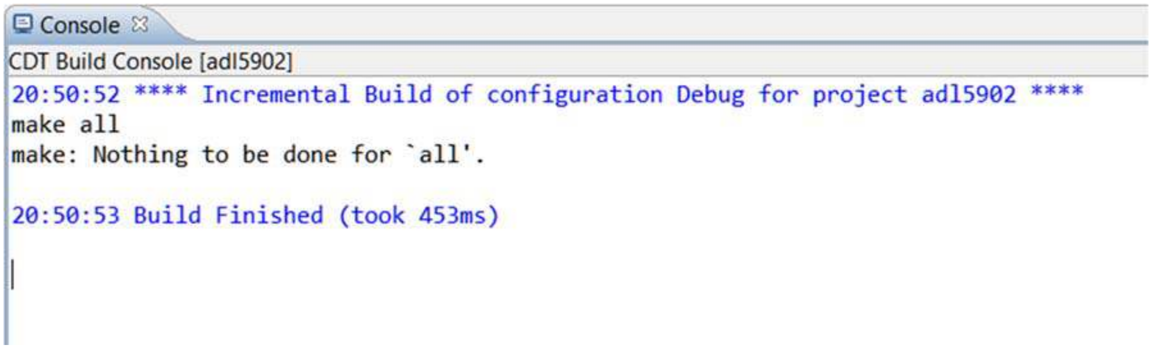
8.

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  .



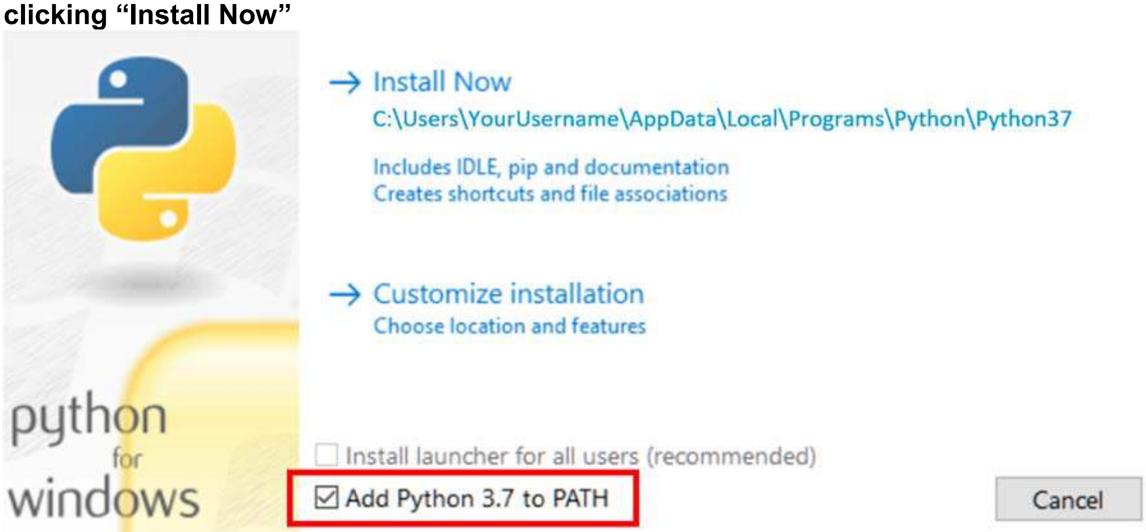
3. After it finishes building, click Debug and click Application with GDB and OpenOCD (Emulator). Copy the following Debug configurations on the new window that will appear then click the Debug button.
4. On the Debug window, click the Resume to run and display the results on the Console window.

Python Development Guide

Installations

Assumes a fresh installation of all required software

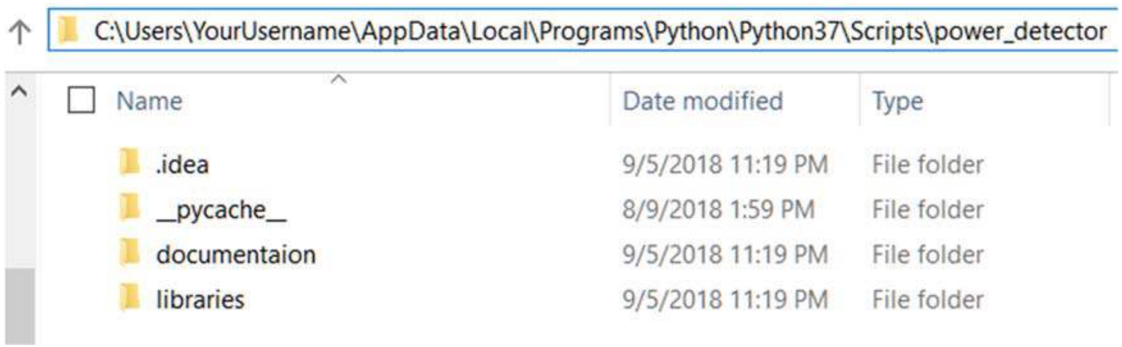
1. Download **python 3.7.0** version. Choose the right version depending on operating system. For windows, choose **Windows x86-64 executable installer**. (Do not run installer yet)
2. Run installer as Administrator. During installation, **check "Add Python 3.7 to PATH" before clicking "Install Now"**



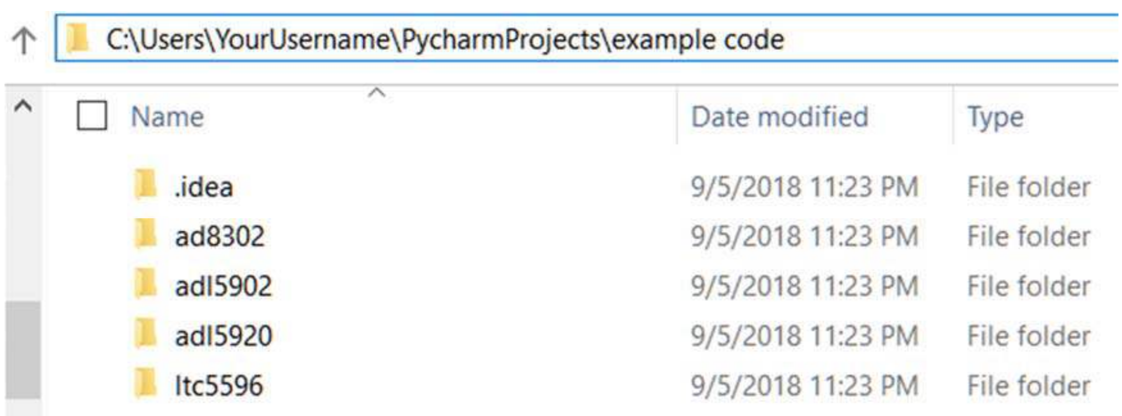
3. Install **pyserial**. For windows, enter **pip3.7 install pyserial** on command prompt.
4. Download and install **PyCharm community version**
5. Download and install **mBed windows serial driver**

Setting Up PyCharm

1. Download **power detector development code (alpha).zip** and extract the file on your PC.
2. In the extracted files, copy **power_detector** directory to inside the "Scripts" folder where the python3.7 is located. For windows, the default location is similar to **C:\Users\MyUsername\AppData\Local\Programs\Python\Python37\Scripts** path



3. In the extracted files, copy **example code** directory to **C:\Users\MyUsername\PycharmProjects**. Create \PycharmProjects directory if it does not exist yet.



4. Launch pyCharm. 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. In the extracted files look for **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-ADL5902-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 **adl5902** folder and double click **adl5902-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

EVAL-ADL5902-ARDZ Design Files

- Schematic Diagram of EVAL-ADL5902-ARDZ
- Layout Design of EVAL-ADL5902-ARDZ
- Fab Files of EVAL-ADL5902-ARDZ
- Assembly Files of EVAL-ADL5902-ARDZ