ADE9178 Metrology Example User's Guide

ADE9178 Metrology Example User's Guide
Version 1.1.0, July 2025

http://www.analog.com
energy.support@analog.com

1 Table of Contents

Copyright Information

Disclaimer

Analog Devices, Inc. reserves the right to change this product without prior notice. Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of Analog Devices, Inc.

Trademark and Service Mark Notice

2 Introduction

2.1 Purpose

The ADE9178 Metrology Example demonstrates how to configure the ADE9178 for energy metering applications and display metrology parameters. This document explains how to communicate with the evaluation kit (EVK), describes the commands supported by the application in the repository, and provides details about the GUI.

3 Communicating with the EVK

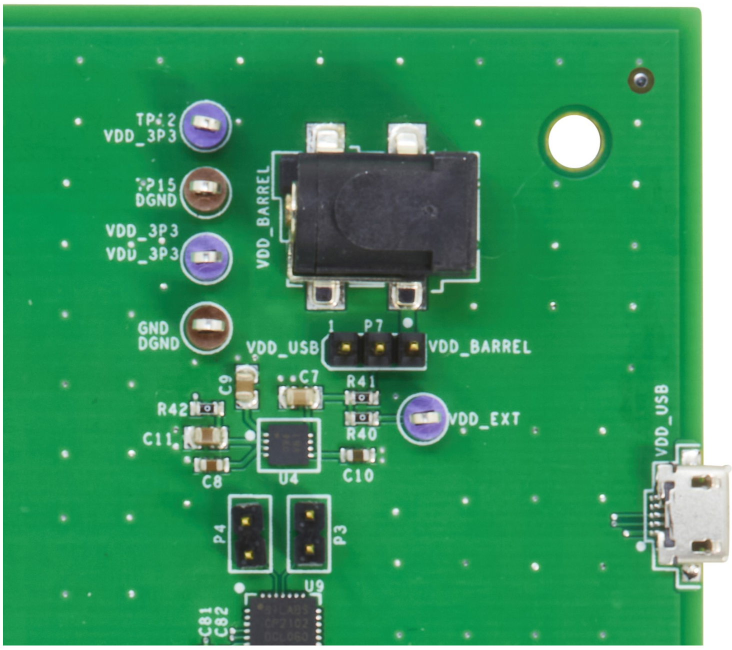
The EV Kit has an onboard MAX32670 microcontroller that comes pre-programmed with some useful evaluation software. UART communications are used to connect the board to a PC.

3.1 Establishing Connection

There are 3 ways to establish a UART communications connection with the MAX32670 Host MCU on the evaluation kit:

1. Via the Micro-USB port (VDD_USB) with on-board USB UART IC (Note: Temperature range of CP2102 is -40°C to 85°C)
   a. Populate P3 and P4 Jumpers (pictured below),
   b. A USB-UART Driver is required which can be downloaded here, follow installation instructions on the website,
   c. Connect other end of micro-USB cable to PC
2. MAX32625PICO connected to the SWD header on the board
   a. Remove P3 and P4 Jumpers, this disconnects the USB to UART IC from the Host MCU,
   b. Connect MAX32625PICO ribbon cable to SWD header,
   c. Connect USB side of MAX32625PICO to PC
3. An external USB to UART dongle connected to SER header on the board
   a. Remove P3 and P4 jumpers, this disconnects the USB to UART IC from the Host MCU,
   b. Connect TX, RX and GND lines of USB to UART dongle to relevant pins on the SER header,
   c. Connect USB side to PC

The default recommended way to connect is using the Micro-USB port.



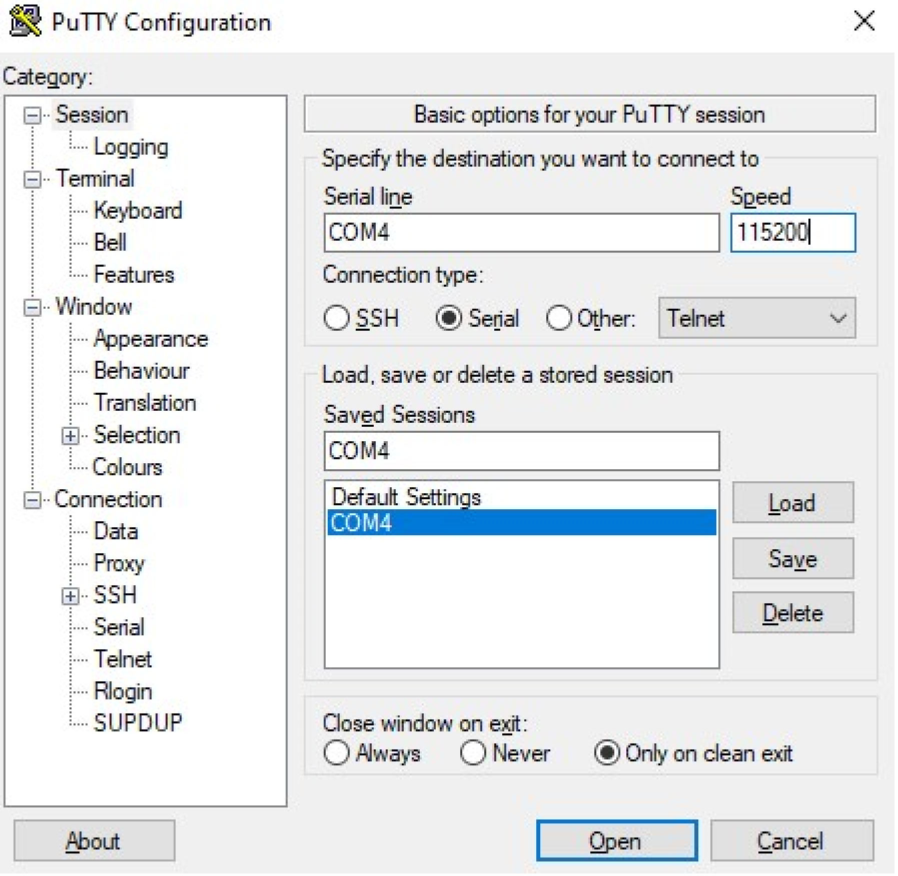
4 Getting Started with the CLI

The command line interface (CLI) allows us to communicate with the EVK board easily. This section contains detailed information on how to establish connection to the EVK and what each CLI command does. Some of this information is covered in the main EVK user guide.

4.1 Establishing Serial Connection

After connecting the board to your PC as detailed in the previous section, follow these steps to establish a serial connection with the evaluation board and access the command line interface (CLI):

1. Open a serial communications program such as Putty, Tera Term or equivalent
2. Under 'connection type' select Serial
3. Under 'serial line' select the COM port that is being used on your PC to establish communications with the evaluation board, you can find this by opening 'device manager' on Windows.
4. Under 'speed' enter 115200
5. Click 'open' to establish the connection.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM4

Speed: 115200

Connection type: ○ SSH ● Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: COM4

Default Settings
COM4

Close window on exit: ○ Always ○ Never ● Only on clean exit

Once connection has been establish, a blank terminal screen will be shown. Press the 'enter' key to get the following CLI prompt.

```
ADE9178>
```

Note: You may have to reset the MAX32670 MCU by pressing the HOST_RESET button.

Enter the 'help' command to verify the CLI is responding correctly. The following command line test will be shown:

```
ADE9178> help

        COMMANDS:           PARAMETERS:
          start               <num_cycles>
          version
          setdisplay          <options> <on|off>
          help                <command>
          reset
          manual              <on|off>
          setreg              <device> <register address> <register value>
          getreg              <device> <register address> <number of registers
to read>
          loadreg
          savereg
          setconfig           <config choices> <config values>
          getconfig           <config choices>
          capturewfrm         <wf src> <delay_num_cycles>
          displaywfrm

Command specific help is displayed with help <command>
```

These are all the commands available within the CLI.

4.2 Explanation of Commands

4.2.1 Start Command

```
ADE9178> start <num_cycles>
```

This prints the latest metrology parameters of the ADE9178 based on analog inputs passing through the ADE9113 isolated ADCs. The < `num_cycles>` parameter is the number of cycles for which the ADE9178 displays metrology results. By default line cycle frequency is 50Hz which gives a period of 20ms. Example output of typing `'start 1'` with 10V and 500mA input to main 3 phases, with power factor 1 is shown below. Nothing is applied to neutral phase. Main 3 phases are calibrated, neutral phase is uncalibrated. You should see something similar to this:

```
ADE9178> start 1
1, AWATT           = 5.000495
1, AVA             = 4.997583
```

```
1, APF             = 1.000000
1, BWATT           = 5.000131
1, BVA             = 4.997947
1, BPF             = 1.000000
1, CWATT           = 4.994308
1, CVA             = 4.992124
1, CPF             = 1.000000
1, AVRMS           = 10.040573
1, AVRMSONE        = 10.041113
1, AVRMSHALF       = 10.099980
1, BVRMS           = 10.044072
1, BVRMSONE        = 10.045040
1, BVRMSHALF       = 8.491629
1, CVRMS           = 10.036239
1, CVRMSONE        = 10.037213
1, CVRMSHALF       = 9.160602
1, AIRMS           = 0.497876
1, AIRMSONE        = 0.498094
1, AIRMSHALF       = 0.498188
1, BIRMS           = 0.498641
1, BIRMSONE        = 0.498903
1, BIRMSHALF       = 0.498799
1, CIRMS           = 0.498079
1, CIRMSONE        = 0.498210
1, CIRMSHALF       = 0.497818
1, AUX0RMS         = 0.008321
1, AUX0RMSONE      = 0.008697
1, AUX0RMSHALF     = 0.007682
1, AUX1RMS         = 0.008710
1, AUX1RMSONE      = 0.009145
1, AUX1RMSHALF     = 0.010205
1, AUX2RMS         = 0.008894
1, AUX2RMSONE      = 0.008901
1, AUX2RMSHALF     = 0.006714
1, AUX3RMS         = 0.024001
1, AUX3RMSONE      = 0.025062
1, AUX3RMSHALF     = 0.025306
1, AUX4RMS         = 0.008380
1, AUX4RMSONE      = 0.007175
1, AUX4RMSHALF     = 0.007201
1, AUX5RMS         = 0.008413
1, AUX5RMSONE      = 0.007741
1, AUX5RMSHALF     = 0.006549
1, AWATTHR_POS     = 62.149727
1, AWATTHR_NEG     = -0.001833
1, AWATTHR_SIGNED  = 62.147892
1, AVAHR           = 62.060295
1, BWATTHR_POS     = 62.111225
1, BWATTHR_NEG     = 0.000000
1, BWATTHR_SIGNED  = 62.111225
1, BVAHR           = 62.022549
1, CWATTHR_POS     = 62.148685
```

```
1, CWATTHR_NEG     = 0.000000
1, CWATTHR_SIGNED  = 62.148685
1, CVAHR           = 62.061146
1, APERIOD         = 0.019999
1, BPERIOD         = 0.019999
1, CPERIOD         = 0.019999
1, COM_PERIOD      = 0.019999
1, ANGL_AV_BV      = 4.187917
1, ANGL_BV_CV      = 4.189418
1, ANGL_AV_CV      = 2.094251
1, ANGL_AV_AI      = 0.000614
1, ANGL_BV_BI      = 0.000307
1, ANGL_CV_CI      = 0.000000
1, ANGL_AI_BI      = 4.187610
1, ANGL_BI_CI      = 4.188804
1, ANGL_AI_CI      = 2.093637
1, STATUS0         = 0xf7f80400
1, STATUS1         = 0xffff874f
1, STATUS2         = 0x0
1, STATUS3         = 0x0
1, ERROR_STATUS    = 0x0
```

Values have been converted to real world units. In the case of angle register values the unit used is radians. Status registers have been given in hexadecimal format since that is most useful. The number on the very left of each row represents a "count" of the IRQ0 that read the outputs. User can therefore identify if any samples are missing which can occur if too much data is streamed across the comms. If samples are missed a message will be printed at the bottom of the start command output:

```
Warn : missed IRQ0 count is 35
Try enabling single output parameters to reduce missed IRQ0. Refer to
setdisplay command for list of output parameters to select.
```

Reducing the number of output parameters to avoid missed samples will be discussed later in the guide. The user can also use a lower number of <num_cycles> to avoid skipped samples.

4.2.2 Version Command

```
ADE9178> version
```

This outputs the ADE9178 firmware version and build ID. The returned text should be as follows:

```
ADE9178> version
ADE9178 Firmware Version:1.0.0
ADE9178 Firmware Chip ID: 0xc6bfd62
ADE9178 Example Build ID: 0xa54ba7f
```

4.2.3 Set Display Command

```
ADE9178> setdisplay <option> <on|off>
```

This command allows for enabling/ disabling the display of various ADE9178 parameters normally shown by the `'start <num_cycles>'` command. The options are:

1. `all:` All display parameters.
2. `rms:` filtered RMS output parameters, on by default. This is all the current and voltage RMS values from the isolated ADCs.
3. `rmsone:` RMSONE output parameters, on by default. This is all the current and voltage RMSONE values from the isolated ADCs.
4. `rmshalf:` RMSHALF output parameters, on by default. This is all the current and voltage RMSHALF values from the isolated ADCs.
5. `eventrmsone:` All dip and swell RMSONE output parameters, off by default.
6. `eventrmshalf:` All dip and well RMSHALF output parameters, off by default.
7. `power:` All power output parameters, on by default. These are calculated from the current and voltage values.
8. `energy:` All energy output parameters, on by default.
9. `period:` All period output parameters, on by default.
10. `angle:` All angle output parameters, on by default.
11. `status:` All status output parameters, on by default.
12. `errorstatus`: Display of repeated error status messages, off by default. This is not related to the start command. If this setting is enabled error status messages will continuously be displayed in the CLI as they occur instead of just being displayed the first time the error occurs. This is a debug feature, turning it on is not recommended.

For example, to display only filtered RMS parameters, first type: `'setdisplay all off'`, and subsequently: `'setdisplay rms on'`.

As briefly mentioned earlier, turning unused display parameters off can allow for more data to be streamed across comms and prevent missed samples when using start command with a large <num_cycles> parameter.

4.2.4 Help Command

```
ADE9178> help <command>
```

Get a detailed help menu for a specific command.

4.2.5 Reset Command

```
ADE9178> reset
```

This command can be used to reset the ADE8167 and ADCs on the EVK board.

4.2.6 Manual Command

```
ADE9178> manual <on|off>
```

Enables or disables manual mode. When manual mode is turned off, no text is printed out in the CLI. On by default.

4.2.7 Set Register Command

```
ADE9178> setreg <device> <register address> <register value>
```

Writes to a register on the ADE9178 or one/all of the isolated ADCs. Possible `<device>` parameters are:

1. `ADE9178:` Write to a register on the EVK metrology part
2. `ADC0` : Write to a register on Neutral Phase isolated ADC
3. `ADC1:` Write to a register on Phase C isolated ADC
4. `ADC2:` Write to a register on Phase B isolated ADC
5. `ADC3:` Write to a register on Phase A isolated ADC
6. `ALL_ADC:` Write to a register on all isolated ADCs simultaneously. This can be used to write the same value to a particular register for all the ADCs.

The `<register address>` and `<register value>` parameters should be in hexadecimal. For example: ' `setreg ADE9178 b 3289'` . Please consult the datasheets for the ADE9178 and ADE9113 parts for a full list of register addresses and how to configure them.

4.2.8 Get Register Command

```
ADE9178> getreg <device> <register address> <number of registers to read>
```

Reads a register on the ADE9178 or one/all of the isolated ADCs. Possible <device> parameters are the same as the `'setreg'` command. The <register address> parameter should be in hexadecimal. The possible values for <number of registers to read> parameter is dependent on the device selected:

1. When the device is set to `ADE9178` , `<number of registers to read>` can only be greater than 1 for burst access mode if the address range lies from:
   a. `AVGAIN (0x0) to CONFIG_LOCK (0xDD)`
   b. `AVRMS (0x200) to VACSWELLONE (0x294)`
   c. `STATUS0 (0x400) to VERSION2 (0x40B)`
   d. `AVRMS_1 (0x600) to AUX5RMSHALF_1 (0x630)`
   e. `AVRMS_2 (0x631) to VACRMSONE_2 (0x661)`
2. If the number of registers requested in burst access mode goes outside these ranges, it will loop back to the start.
3. When the device is set to any of the isolated ADCs or `ALL_ADC` , `<number of registers to read>` can only be 1.

Format of the output of this command is:

```
ADE9178> getreg ADE9178 e 1
Address = 0xe    Value = 0x2d79
```

Here you can see the register values returned in hexadecimal. Similarly for ADC's we can use the command format:

```
ADE9178> getreg ADC1 e 1
Address = 0xe    Value = 0xbb
```

This returns the register value 0xe for ADC1 which corresponds to Phase C isolated ADC .

Using ALL_ADC gives the following command readout:

```
ADE9178> getreg ALL_ADC e 1
Address = 0xe    Value = 0xbb
Address = 0xe    Value = 0xbb
Address = 0xe    Value = 0xbb
Address = 0xe    Value = 0xbb
```

The value of all four ADCs will be printed out in order like shown above. They print in the order ADC1, ADC2, ADC3, ADC4. Register maps for ADE9113 and ADE9178 can be consulted to determine what register each value corresponds to.

4.2.9 Load Registers Command

```
ADE9178> loadreg
```

Load currently saved configuration registers to the ADE9178 metrology part. Registers were saved to flash memory using `'savereg'` command. `'loadreg'` returns a list of all the register values that are loaded in from flash memory. Useful after resetting the part.

4.2.10 Save Registers Command

```
ADE9178> savereg
```

Save modified ADE9178 configuration registers to the flash memory. These saved registers can be loaded from flash with the `'loadreg'` command. The `'savereg'` command returns a list of all register values saved to the flash memory. The registers are as follows:

- Voltage Registers: `xVGAIN, xVOS, xVPHCAL, xVRMSOS, xVRMSHALFOS, xVRMSONEOS`
- Current Registers: `xIGAIN, xIOS, xIPHCAL, xIRMSOS, xIRMSHALFOS, xIRMSONEOS`

Where `x` can be equal to `A, B, C, AUX0, AUX1, AUX2, AUX3, AUX4` and `AUX5.`

- Power Registers: `xPGAIN, xPWATTOS`

Where `x` can be equal to `A, B` and `C`.

- Configuration Registers: `CONFIG0, CF1_CONFIG, CF2_CONFIG, VNOM, USER_PERIOD, CF1_THR, CF2_THR, ZX_LP_SEL, ZXTHRSH, ACT_NL_LVL, APP_NL_LVL, EGY_TIME, EP_CFG, ADC_REDIRECT0, ADC_REDIRECT1, ADC_CONFIG, CF_LTMR, MASK0, MASK1, MASK2, MASK3, ERROR_MASK.`

4.2.11 Set Config Command

```
ADE9178> setconfig <config choices> <config value>
```

Allows for configuration of voltage, current and auxiliary channel scaling. This is used to convert values to real units for the start command.

Possible `<config choices>` parameters are:

1. `1: for Voltage Channel Scale`
2. `2: for Current Channel Scale`
3. `3: for Auxiliary Channel Scale`

The `<config value>` parameter is just the scaling value we want to be applied to the channel values in the start command.

Default values are `707` for voltage channel scale, `44.188` for current channel scale and `707` for auxiliary channel scale. These correspond to the full scale voltage and current values allowing users to configure the example code to work with different full scale values if they have swapped out shunts or resistors in the potential divider with different value components on the EVK.

4.2.12 Get Config Command

```
ADE9178> getconfig <config choices>
```

Displays the current scaling value applied to values on certain channels. The `<config choices>` parameter has the same options as the `'setconfig'` command. Example output from the command is shown below:

```
ADE9178> getconfig 2
current scale: 44.188000
```

4.2.13 Capture Waveform Command

```
ADE9178> capturewfrm <wf src> <delay_num_cycles>
```

This command captures waveform streaming data from ADE9178. The ADE9178 supports streaming of either raw ADC samples or samples after the phase compensation filter (PCF). It captures data on all the following channels: `AV, AI, BV, BI, CV, CI, AUX0, AUX1, AUX2, AUX3, AUX4 and AUX5.`

Possible <wf src> parameters are:

1. `0: for raw ADC streaming`
2. `1: for ADC streaming after PCF`

The `<delay_num_cycles>` parameter is used to set a delay in number of cycles before the waveform streaming data capture begins. For example, example the following command:

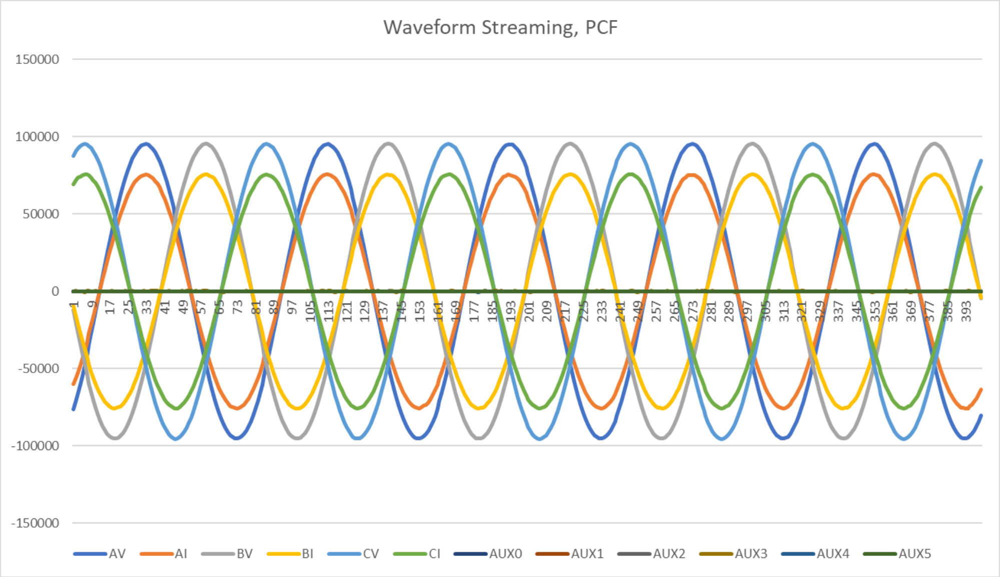
```
ADE9178> capturewfrm 1 50
Wfs Configuration is successful
```

This will capture waveform streaming data after the PCF filter, and it will begin capturing data after `50` cycles. With a frequency of `50Hz`, period `20ms`, this represents a delay of `1 second`.

4.2.14 Display Waveform Command

```
ADE9178> displaywfrm
```

This will display the data captured from the most recent `'capturewfrm'` configuration. Plotting the data produces a graph that will be similar to the one shown below. The `'y axis'` values represent codes and can be converted to real units by consulting the ADE9178 datasheet.



4.2.15 Get Error Count Command

```
ADE9178> geterrorcount
```

This fetches the amount of times errors have occurred in the part recently. Once this command is run the count is reset to 0.

4.3 ADE9178 Further Behavior and Interfaces

Refer to ADE9178 datasheet for the detailed description. You can also refer to example project README file located at `<METEX_root>/example/project/keil/max32670` for more information about how to setup the EVK and use the CLI commands.

4.4 Technical or Customer Support

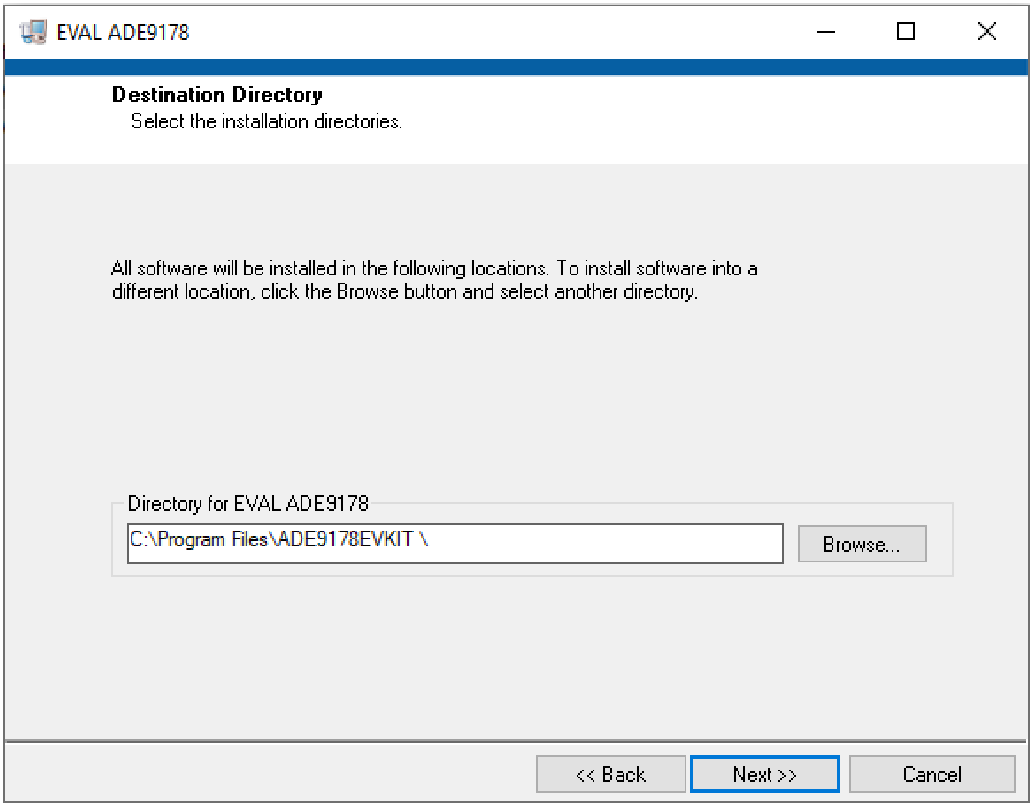
E-mail your ADE9178 Metrology Example questions to energy.support@analog.com

5 Getting Started with the GUI

For users who would prefer a graphical user interface (GUI) tool over the CLI there is one available. Firstly, download the GUI tool from the ADE9178EVKIT# EV kit product page on the Analog Devices website.

5.1 Installing Dependencies and the GUI Tool

1. Install the LabVIEW Runtime 2019 SP1 f3 (64-bit)
2. Install the NI-VISA Runtime 23.5
3. From the downloaded folder with the GUI tool, run the ADE9178EVKIT# SW installer executable (.exe) file that's available in EVAL-ADE9178 Evaluation Board | Analog Devices
4. Follow the on-screen instructions to complete the installation



EVAL ADE9178

Destination Directory

Select the installation directories.

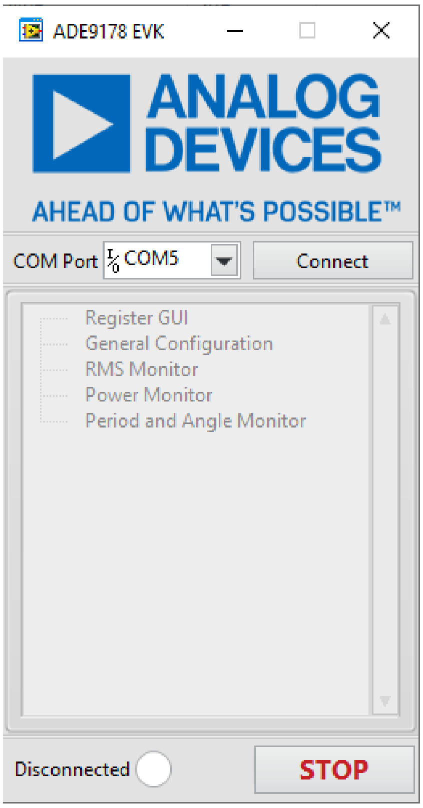
All software will be installed in the following locations. To install software into a different location, click the Browse button and select another directory.

Directory for EVAL ADE9178

C:\Program Files\ADE9178EVKIT \

5.2 Running the Software

1. Open the software by navigating to C:\Program Files\ADE9178EVKIT and launch the program ADE9178 EVK. This is the default installation directory.
2. The connection dashboard will appear. Select the COM port that is being used on your PC to connect to the board. You can find this through the 'device manager' on Windows.
3. Click 'connect' button to establish connection with the ADE9178EVKIT#. If connection is successful the LED indicator on the dashboard will turn green.
   GUI Connection Dashboard



For more details on how to use the GUI including how to read/write registers and monitor results the full GUI user guide can be consulted which is located within the folder downloaded from the Analog Devices website.