

DC Meter Source Lab 0:

ALICE 1.3 Desktop is a multi-purpose and useful set of tools for measuring electronic circuits in lab projects and experiments. But with all its capabilities it can be a little intimidating at first to get comfortable using all of the tools. In this Tutorial we are going to go over one of those features (The DC Meter Source) so that you can get the most out of the software.

1.0 Background:

The ALM1000 contains two 16 bit 100KSPS analog to digital converters (ADC) and two 16 bit digital to analog converters. It also contains a means to source both voltage and current from two 16 bit 100KSPS digital to analog converters (DAC). The system can simultaneously measure the current and voltage. These inputs and outputs are often used to measure time varying waveforms as in the ALICE oscilloscope interface. However, there are often times when simple, accurate, DC measurements are required such as in a resistor network. The following simple user interface is provided for that purpose. It provides the means to accurately calibrate the measurements of voltages outside the built in 0 to 5 Volt range of the ADALM1000 through the use of resistor voltage dividers.

1.1 Required files:

The DC Meter-Source Tool Windows executable program is included with the ALICE 1.3 desktop installer package. Instructions for installing the software on Windows can be found at the following link:

https://wiki.analog.com/_media/university/tools/m1k/alice/alice-install-procedure.pdf

The Python source code is also available from the Analog Devices GitHub repository and is generally compatible with version 2.7 of Python. Other than the pysmu library bindings the program only imports modules generally included with standard Python installation packages.

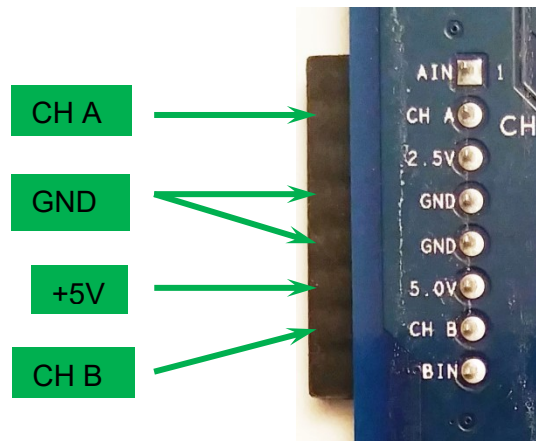
1.2 Connections:

It is assumed that the reader is somewhat familiar with the functionality and capabilities of the ADALM1000 hardware. For more on the ADALM1000 hardware please refer to the documents listed in the main ALICE User Guide:

First a few notes on nomenclature used in this document:

CH A refers to the Channel A SMU input / output pin on the 8 pin connector

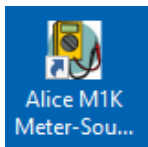
CH B refers to the Channel B SMU input / output pin on the 8 pin connector



ADALM1000 8 Pin Connector

1.3 Screen Setup:

Once the program is run by double clicking on the desktop icon:



as shown in figure 1, should appear. Be sure that the ALM1000 is plugged into the USB port before starting the program.

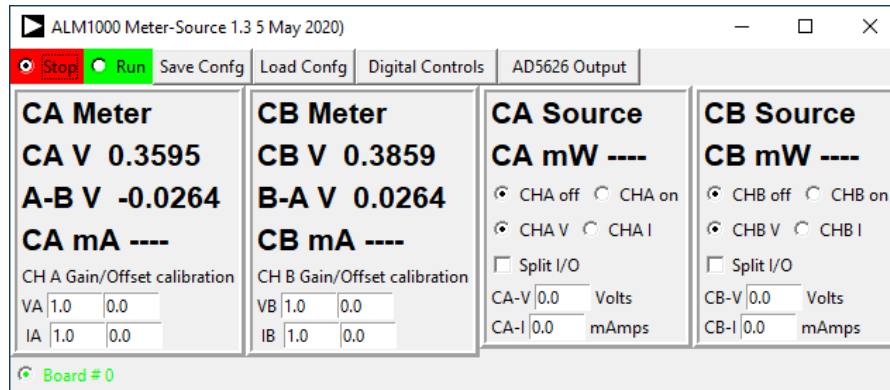


Figure 1, DC Meter-Source screen.

At the top of the screen are two buttons to run and stop (pause) the program looping and taking measurements continuously. There are also buttons to save and load the present configuration. There is a button to open the static logic control interface for the 4 digital I/O pins. There is also a button to open the control interface for the 12 bit AD5626 serial DAC from the ADALP2000 parts kit.

There are four frames below that. The two on the left labeled CA Meter and CB Meter display the measured voltage for channel A and channel B As well as the calculated voltage difference between the channels. Below the voltage display the current flowing in that channel's source output is displayed if the source is turned on. When a channel's source is off (channel is in Hi-Z mode measuring voltage only) the current display is blanked out with dashes and operates much the same as the DC Voltmeter tool.

Below the voltage and current displays at the bottom are entry slots for both voltage and current, gain and offset calibration factors for each channel. Using the calibration procedure from the Voltmeter tool is the best way to get the voltage calibration factors. Refer to the Voltmeter Tutorial Lab 0 for that procedure. Once you have the gain and offset numbers they can be entered here. To save these calibration values to a file for future use, use the Save and Load Config buttons.

The other two frames, on the right, are for controlling the DC voltage/current sources for each channel A and B. The CHA off and CHA on selector turns on the channel as a source (same for CH B). The CHA V and CHA I selector set the source mode to either source voltage or source current (same for CH B). The

CA-V and CA-I entry slots are used to set the voltage and current source values (same for CHB). The channels can source voltage from 0 to 5V, entry is in volts. The channels can source (positive number) or sink (negative number) from -200 mA to 200 mA, entry is in milliamps.

1.4 Use Examples:

To demonstrate using the ALM1000 to measure current while sourcing voltage consider the resistor current divider circuit shown in figure E1. We would like to measure the currents flowing in resistors R_S and R_2 while supplying 5V to R_S . We will be using $R_S = R_1 = R_2 = 100\Omega$ for this example.

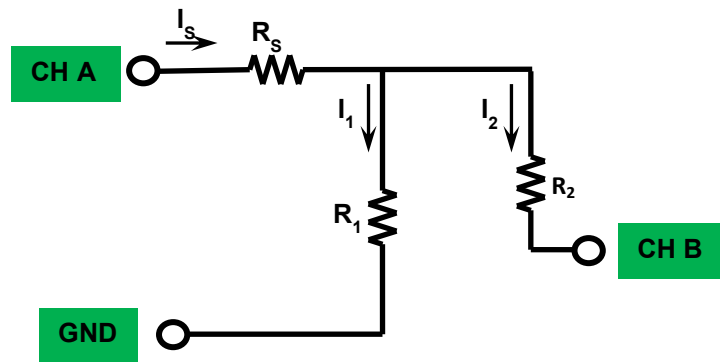


Figure E1, Using CH B as an ammeter

With CA source turned on set to source 5.0V and CB source turned on set to source 0.0V (same as if R_2 were connected to GND) we measure the currents in CA and CB as shown in figure E2. CA current is positive because it is sourcing current and CB is negative because it is sinking current.

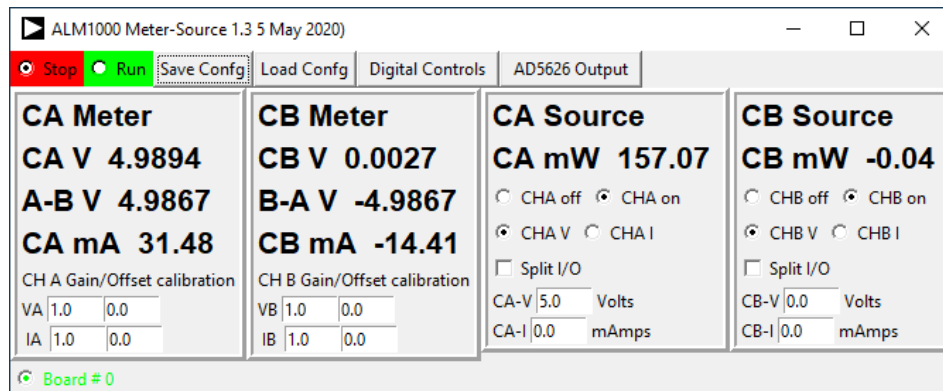


Figure E2, Measuring Currents

We know that the current in channel A (I_S flowing into R_S) is the sum of the currents in R_1 and R_2 . We are measuring the current in R_2 with CH B. We can calculate the current in R_1 by subtracting the CH B current from the CH A current, $31.48 \text{ mA} - 14.41 \text{ mA} = 17.07 \text{ mA}$.

Also note that the power supplied by CH A is calculated ($I \times V$) and displayed in the CA Source frame. The CH B power is also calculated but because the voltage is essentially zero the power even with 14 mA flowing will also be zero.

If we now increase the voltage setting of CH B to 2.5 V. See figure E3.

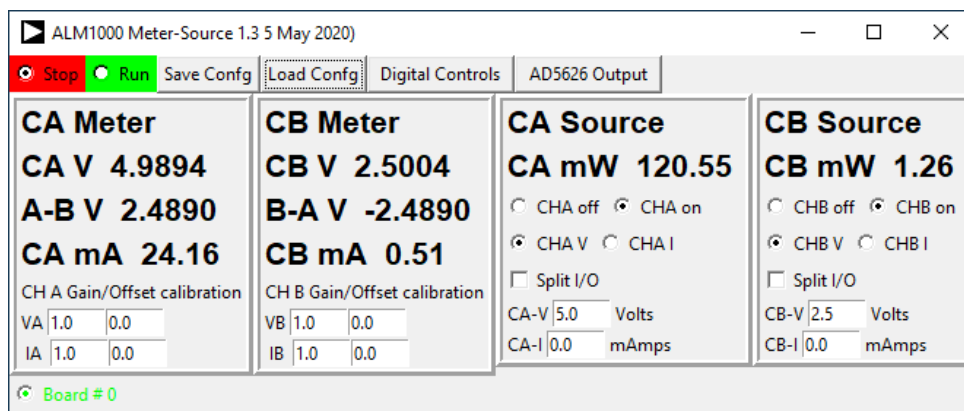


Figure E3, Measuring Currents with CH B sourcing 2.5 V

Now the current sourced by CH A is smaller and the current sunk by CH B has gone down to a small number of mA. The calculated power for the channels also reflects this change in the currents and voltages.

Now let's power an LED. Using the same 100Ω resistor for R_1 connect it to the CH A source and LED as shown in figure E4. Note that the LED has a longer and a shorter lead. The longer lead is the positive (+) terminal and the shorter lead is the negative terminal (-). The LED must be connected the right way around in order to light up. The voltages used from the CH A source are not large enough to damage the LED if it were to be connected backward.

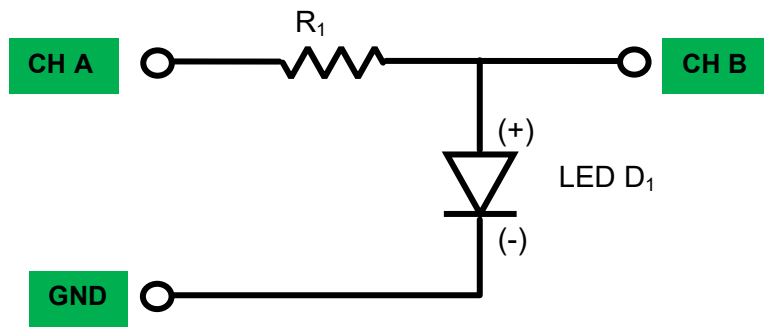


Figure E4, Powering an LED

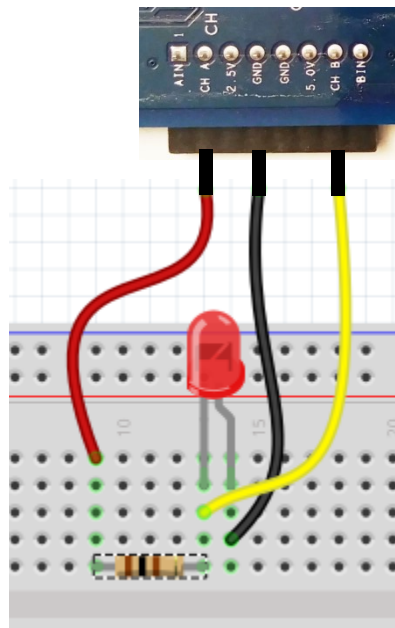


Figure B1, Solderless breadboard connections

Turn on CH A to source voltage and set the value to 5 V. Turn off CH B. Hit Run and you should get results something like figure E5. In this example a Red LED was used and has a forward voltage drop of about 2.0 V at 28 mA. A different color LED such as Yellow or Green will give a different result.

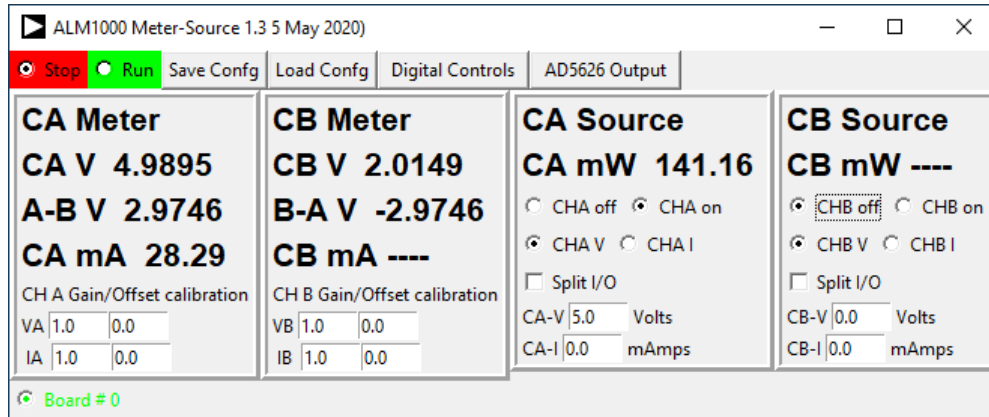


Figure E5, Measuring current in LED and voltage across LED

The LED may flicker slightly as the software cycles through taking measurements.

Try adjusting the Channel A voltage value and watch the brightness of the LED. At what point is it barely glowing? What is the current?

Next Steps – Time to Tinker!:

There are many more controls in the interface but these are the basics to get you started. You can explore the rest by exploring or tinkering on your own. The best way to become familiar with your M1k and the ALICE tools is to start tinkering with electronics. Using some parts from your parts kit put together some other simple circuits that you might have seen or been interested in learning more about (measuring the M1k fixed power supply rails, a resistor voltage divider, powering LEDs or a DC motor, etc.) or explore new electrical components or circuits. Eventually you will be building and analyzing more and more complex circuits.

Appendix:

Customizing the program

There may be times when the user wishes to customize the program to add additional functions such as controlling the digital I/O pins on the ALM1000 or combining the Voltmeter tool with the Ohmmeter tool. This should be relatively easy to do in the Python program file. There are a number of Python tutorial examples [here](#).