

- 1) Build the 8 bit resistor ladder circuit shown in figure 1, preferably on your solderless breadboard. It is best to use 1% resistors for this project if you can, but 5% resistors will work as well.

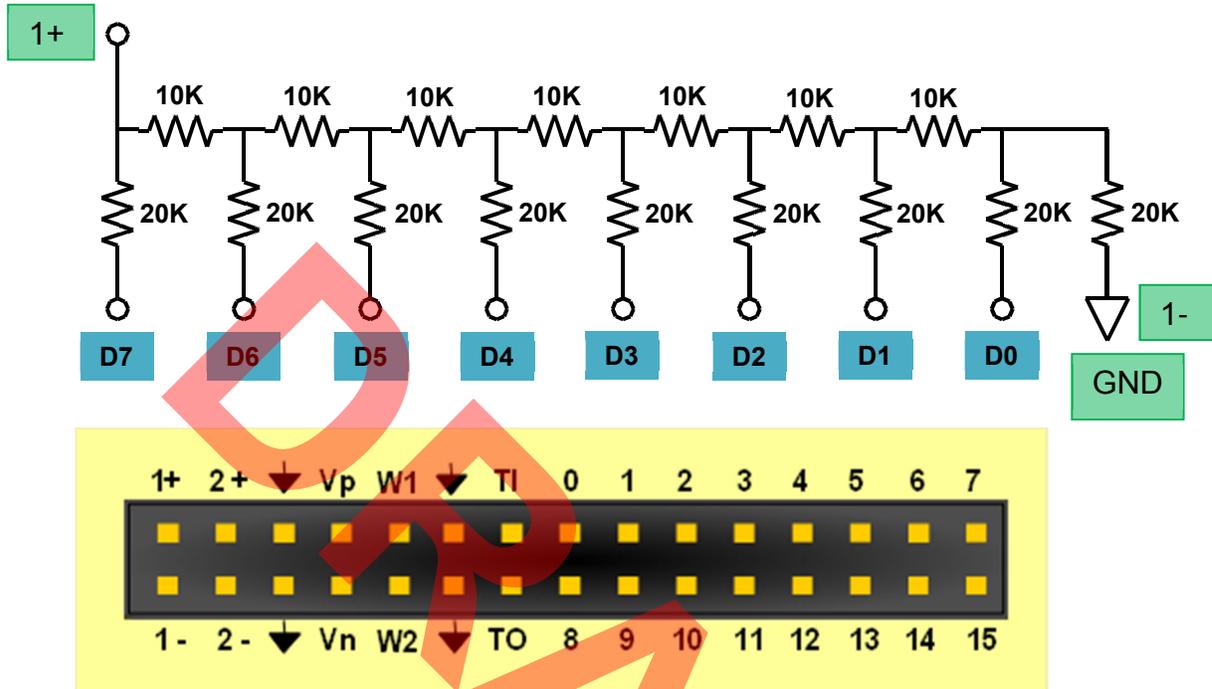
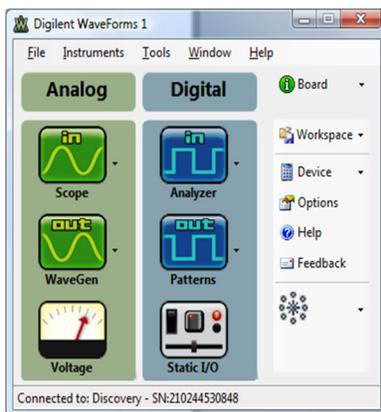


Figure 1 Resistor Ladder circuit

- 2) Connect the 8 digital outputs and the scope channel to the resistor ladder circuit as shown.
- 3) Start the Waveforms software.

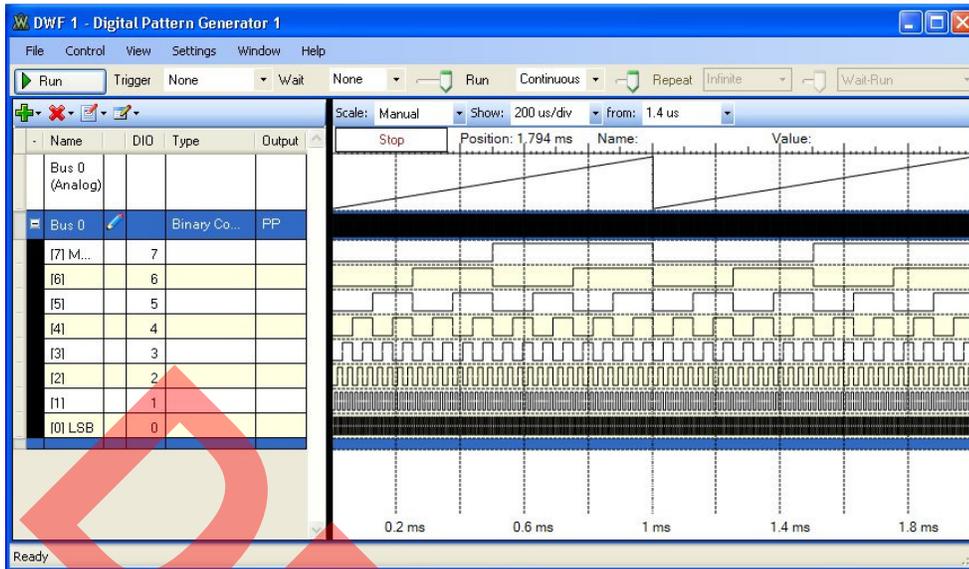


Note: When closing, the WaveForms software stores the last configuration (if set to do so). This tutorial assumes the factory default settings are used. To make sure that you have the factory default settings (even if somebody previously saved a different WaveForms configuration on your machine), click “Options” in the WaveForms main window, then “Erase configuration” in the Options window. Close the Options window.

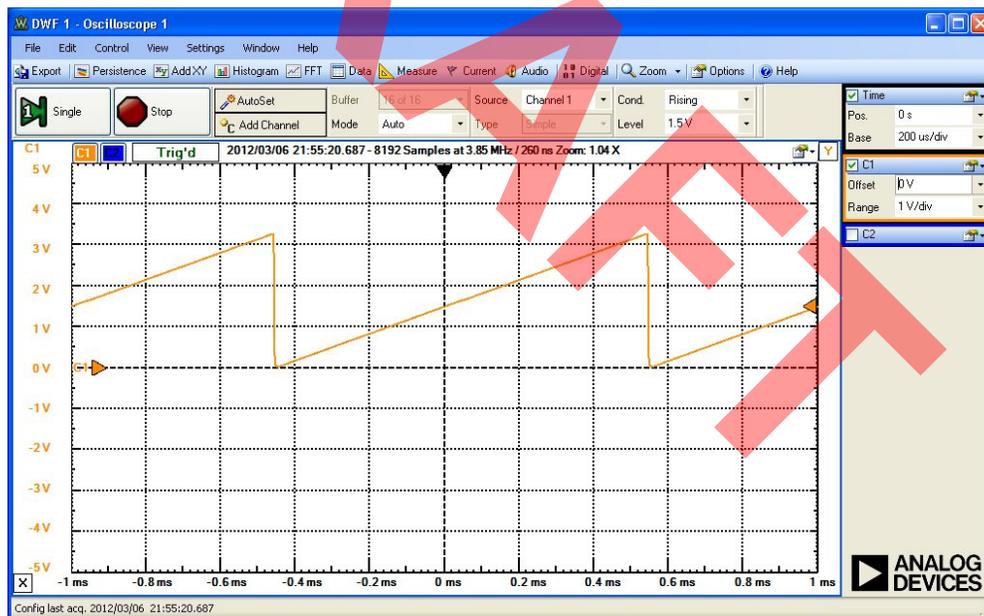
- 4) Start the Digital Patterns tool. Add a bus signal, select DIO 0 - 7 and click on the right green arrow to add signals to the bus. Click on the *show analog* representation check box. Select the *binary* format, the *endianness* to be LSB (the MSB should be DIO 7 and the LSB should be DIO 0), and click OK. Open the parameters box for bus 0, set the output to PP (for push-pull), the frequency to 256KHz, and click close.

Real Analog Lab Module
8-bit DAC built from an R-2R Resistor Ladder

You should see something that looks like the screen below. Lastly, hit the Run button.



- 5) Start the Scope tool, turn channel 1 on, set the range to 1V/ div, and set the time base for 200us/div. Hit the green Run button to start the scope. You may want to adjust the vertical range to see the waveform better. Observe the voltage ramp up from 0 volts to 3.3 volts over a period of 1mSec.



- 6) Congratulations – you have generated your first digital pattern with the Analog Discovery!
- 7) Change the digital pattern. Try the Random pattern and open the FFT window on the scope. You can also load custom patterns by making a plain text csv file with a column of numbers ranging from 0 to 255 (for the 8 bit wide bus). Load your custom pattern and see what happens!