ATX7006

Using histogram multi-test script to measure 16-bit ADC sub-LSB linearity

This application note describes how to measure the linearity of a 16-bit A/D converter with sub-LSB linearity errors, and how using the multi-test histogram Lua script on the ATX7006 can improve measurement accuracy.

The ATX7006 is the ideal instrument for measuring 16-bit A/D converter linearity. The AWG20 module is capable of generating a high accuracy analog ramp with 20-bit resolution; good enough to measure 16-bit ADC linearity accuracy down to 1/16th LSB. The ATView7006 software helps to quickly setup a measurement. When using the histogram test mode, measurement accuracy can be improved even further.

The converter

The Device Under Test (DUT) is an Analog Devices AD7671 16-bit 1MSps CMOS charge redistribution SAR ADC.

The device is used in serial mode. It requires an external reference voltage of 2.5V, a 5.0V analog supply voltage and a digital (output) supply voltage of 2.7 to 5.25 Volts.

The device contains four single-ended analog inputs (INA..IND) with series resistors of different values, so the analog input range can be selected by tying different inputs together to GND or V_{in} or V_{ref}. INGND can be used as a GND Sense input.

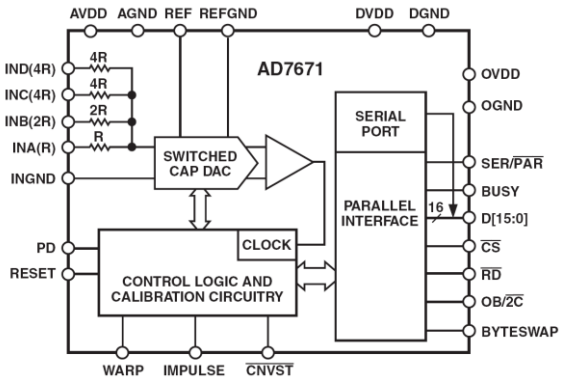

Figure 1 Bock schematic of the AD7671 ADC.

Measurement hardware setup

Figure 2 shows a simplified schematic of the measurement setup.

The 2.5V reference voltage is applied by the DRS20 reference source in the ATX7006, and the 5.0V analog supply voltage (AVDD) is supplied by one of the DPS16 module channels. The digital output voltage (OVDD) is derived from AVDD, with a serial resistor of 22 ohm and additional decoupling.

The analog stimulus signal is generated by the AWG20 module.

The DIO module in the ATX7006 is used to apply update clocks to the AWG module and the DUT. The conversion result is also read by the DIO module in a serial manner.

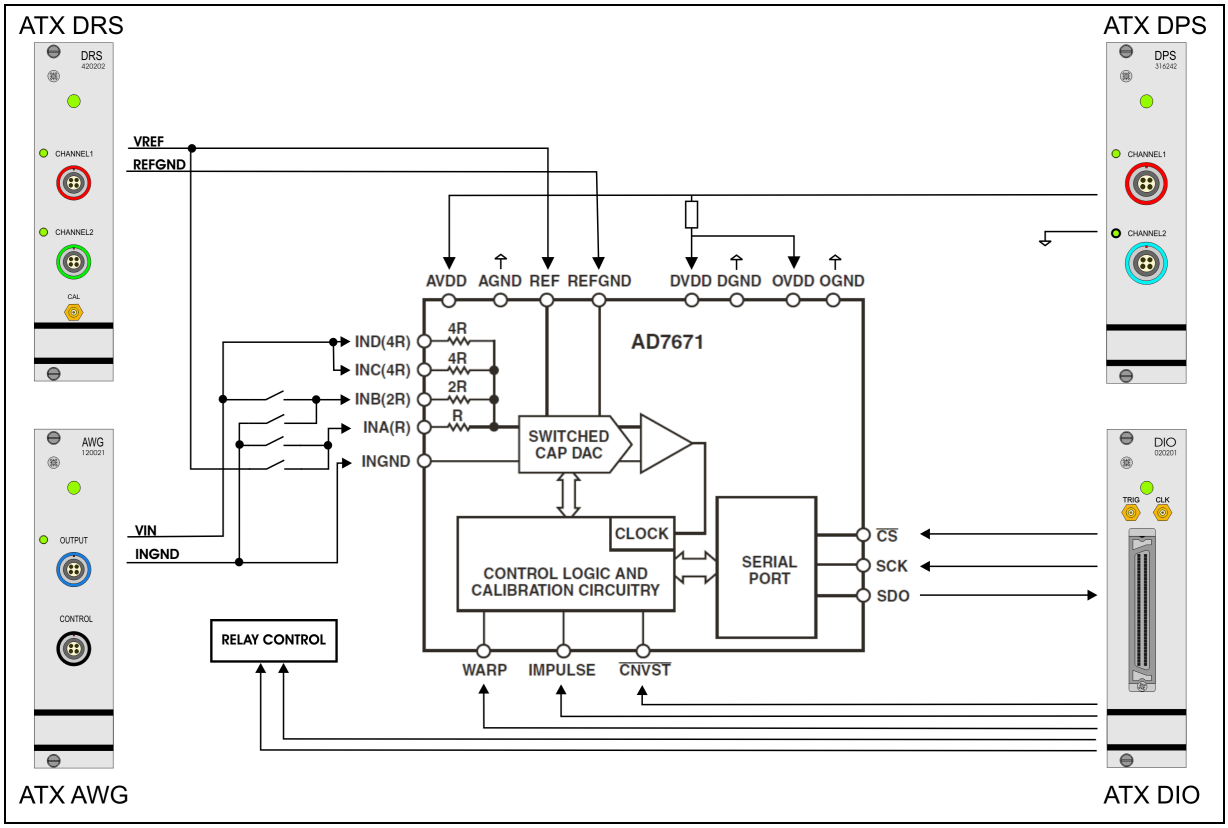

Figure 2 Measurement setup schematic.

DIO module setup

The DIO module is used to capture the conversion results from the AD7671, and to generate the AWG update and DUT sample clocks. The DIO setup screen is shown in Figure 3.

Because of the serial data read mode. The "I/O mode" field is set to "Serial_MSBFirst". Serial word length is set to 16bit.

The "I/O voltage" field of the DIO module is set to 3.30 Volts.

The static data bits on the SCSI connector are used to control the relays on the test fixture which select the input range of the ADC. In this case, to select 0..10V range, bit SDB3 is set.

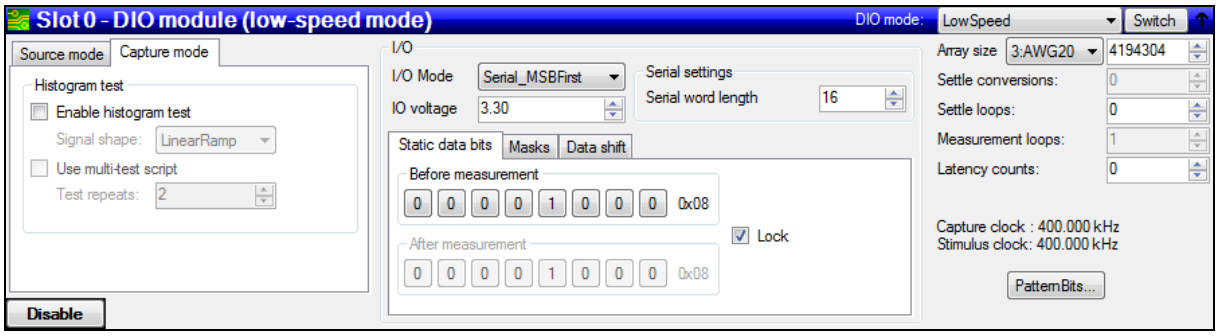

Figure 3 DIO module setup screen.

In serial I/O mode, the serial data output of the DUT is connected to D0 of the DIO, which is internally connected to a shift register.

Figure 4 shows how the DIO pattern bits are programmed to start a conversion and read the conversion result from the DUT.

The "User0" output signal of the pattern bits is connected to the SCLK signal on the DUT. Signal "User1" is connected to the CS# pin on the DUT. "User2" is connected to the CONVST# pin on the DUT, initiating a conversion when it goes low.

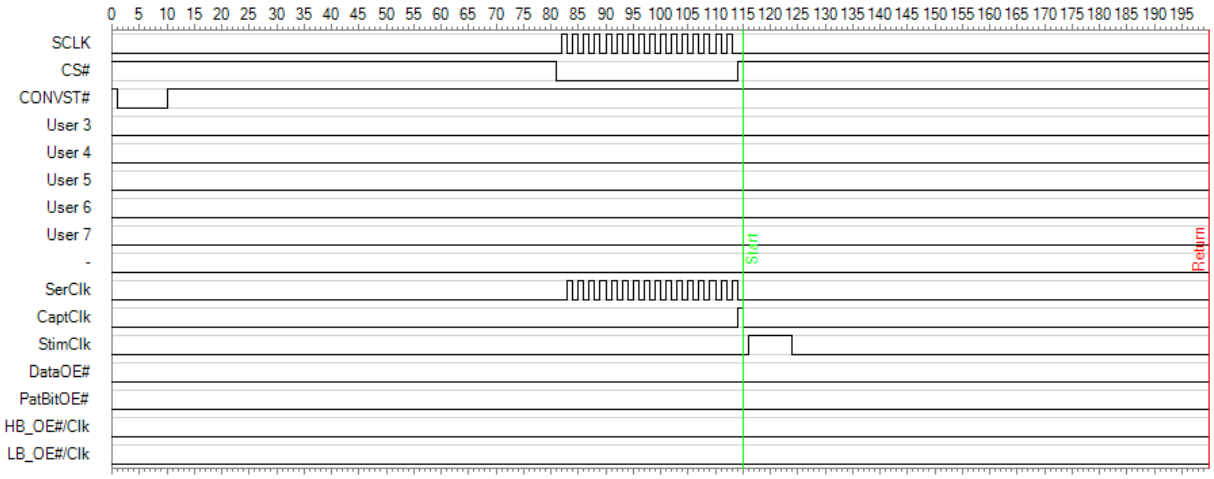

Figure 4 The DIO Pattern bits control the measurement timing.

A conversion is started by activating CONVST#. Then the capture result can be read from the ADC. The "SerClk" signal is the clock of the shift register; a rising edge on this signal captures the state of D0 (which is connected to SDOUT of the AD7671). Shortly after SCLK is high, the AD7671 presents a new databit on D0, so the next step is to generate a rising edge on "SerClk". Once all 16 databits are shifted into the shift register, the complete conversion result can be captured from the shift register and stored in the capture memory. This is done by programming a rising edge on the "CaptClk" signal. Now, the AWG20 can be clocked (by applying a rising edge to the "StimClk" signal) to apply the next voltage to the DUT input.

Since the maximum SCLK period time of the AD7671 is 25 ns, which corresponds to 40MHz clock frequency, the pattern bits clock source is set to 80MHz (see Figure 5). This way the pattern bits step time is 12.5 ns, which is perfect for generating a 25ns clock signal.

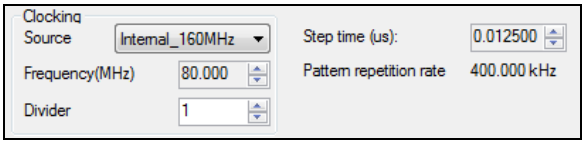

Figure 5 Pattern bits clock settings.

AWG20 module setup

The AWG20 module is used to generate the ramp signal required for the linearity measurement. The AWG20 setup screen is shown in Figure 6.

The DUT is configured with 0..10V input range. When the 5.12Vp range is selected together with a 5.0 volt common-mode voltage, the output range of the positive output of the AWG20 is -0.12 to 10.12 Volt; perfect for a little overdrive on the DUT input.

All settings can be seen in Figure 6.

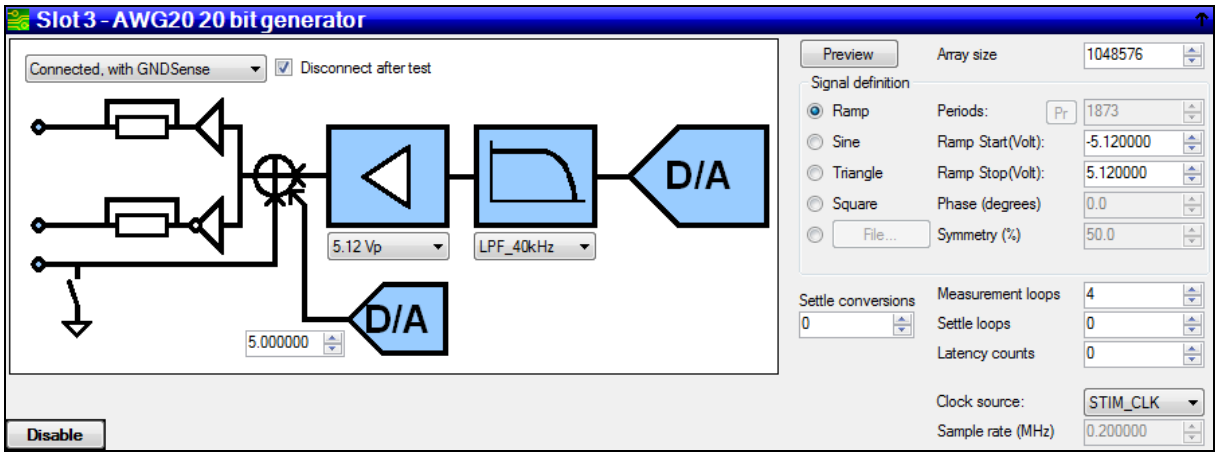

Figure 6 AWG20 module setup screen.

Other modules

Finally, we need to setup two more modules. The used DPS channel is set to 5.0 Volt (no signal generation), and the DRS channel in use is set to 2.5V. For both modules, four-wire connection is selected to compensate for voltage drop in the cables due to the current drawn from the modules.

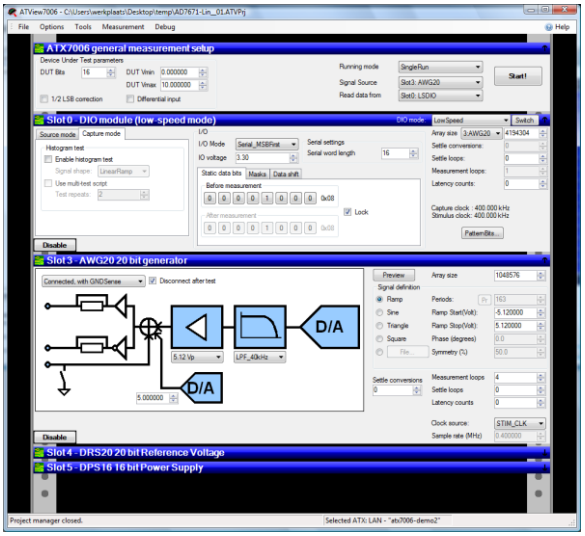

Figure 7 ATView7006 measurement setup screen.

Initial results

Now that the measurement is setup, it is time to run a measurement and look at the result.

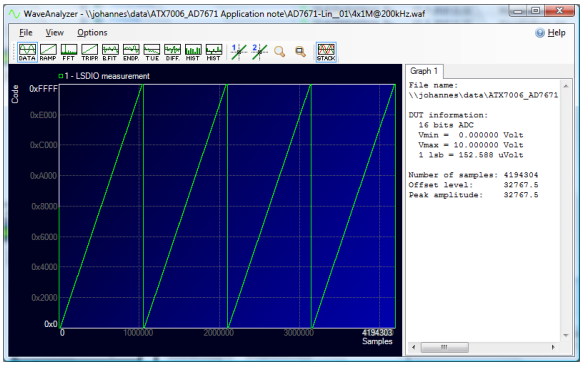

Figure 8 4M ramp measurement array.

After the first measurement, WaveAnalyzer pops up (see Figure 8) and we can see the four ramps of 1M points. By clicking the PlotMode buttons at the top of the window, we can switch to the linearity error plot we're interested in.

Figure 9 and Figure 10 show the linearity errors of the first measurement.

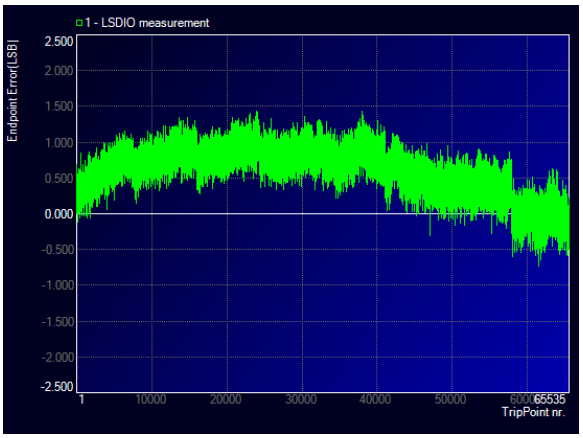

Figure 9 4M array Integral Nonlinearity Error.

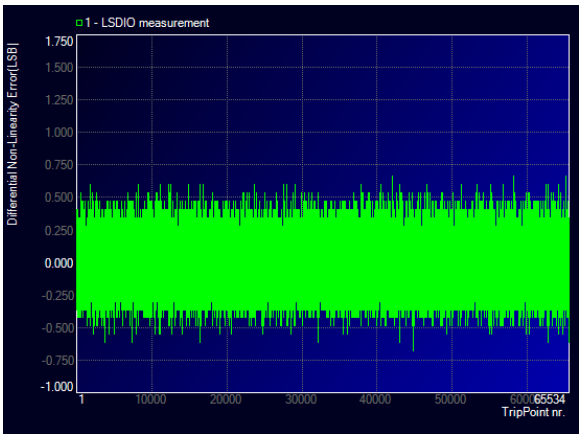

Figure 10 4M array Differential Nonlinearity Error.

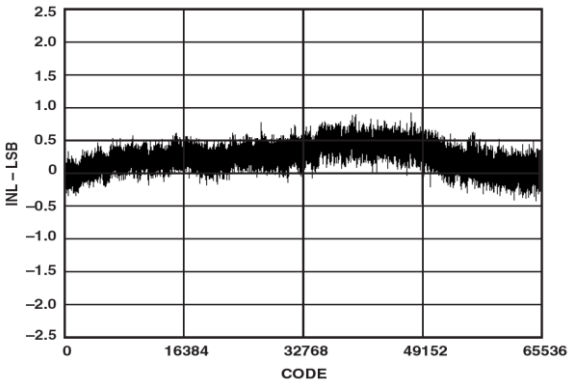

Figure 11 AD7671 datasheet typical INL.

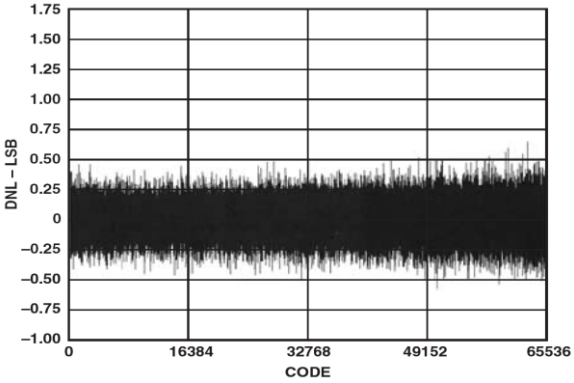

Figure 12 AD7671 datasheet DNL.

Figure 11 and Figure 12 show the typical linearity error graphs as published in the AD7671 datasheet by Analog Devices. In rough lines, the measured result looks similar to the plots in the datasheet. However if we look closer, it appears that the noise band of our measurement, especially in the DNL plot, is larger. Also, if we repeat the measurement, the second DNL plot does not correlate to the first one. This indicates that the wide band seen in the DNL plot is indeed due to noise.

Figure 13 Two DNL plots don't correlate with 4M points.

Increasing accuracy with a Lua script

Since we already use all of the 4M capture memory points of the DIO, we cannot reduce this noise band by simply increasing the measurement loops. Luckily the ATX7006 is equipped with the Lua scripting language.

There is an example script on the ATX7006 (ex_histogram_test.lua, which can be found in the scripts directory in the UserData folder) which allows us to run the whole test for a number of times, and read a histogram array of the captured DUT codes afterwards. WaveAnalyzer can calculate IND and DNL error plots from this histogram.

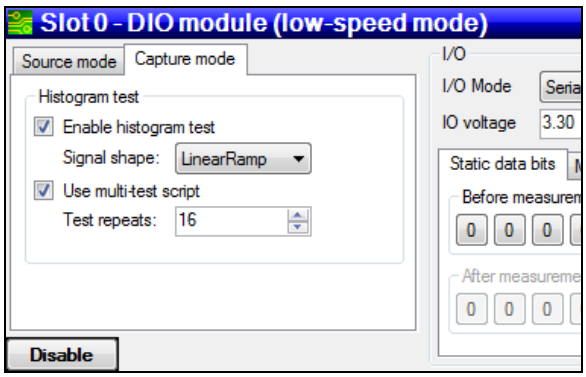

Figure 14 Click "Use multi-test script" to enable the multi-histogram test.

ATView7006 can make use of this script. Simply enable the histogram test and check “Use multi-test script” (as in Figure 14). Now all we have to do is specify the shape of the applied signal (in this case it is a ramp), and the number of times we want to repeat this whole test.

Figure 15 shows the DNL plot for a 16-times repeated histogram test.

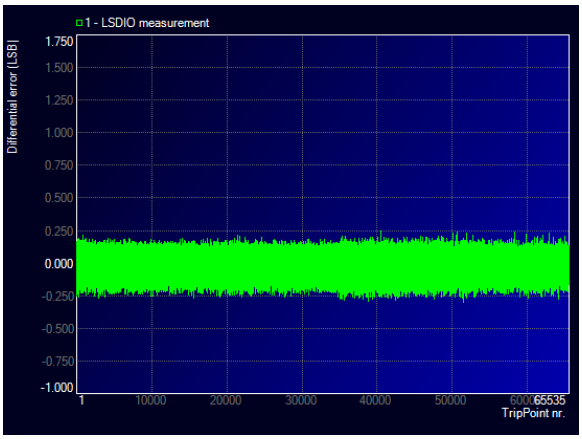

Figure 15 DNL plot with 16 histogram test repeats.

We can see the noise band is much smaller, compared to the single 4M measurement. Also, if we run the same measurement again, the DNL plots perfectly correlate (Figure 16). Now compare this plot with the DNL plot from the AD7671 datasheet (Figure 12). We see that the noise band of this measurement is much smaller than Analog Devices’ measurement.

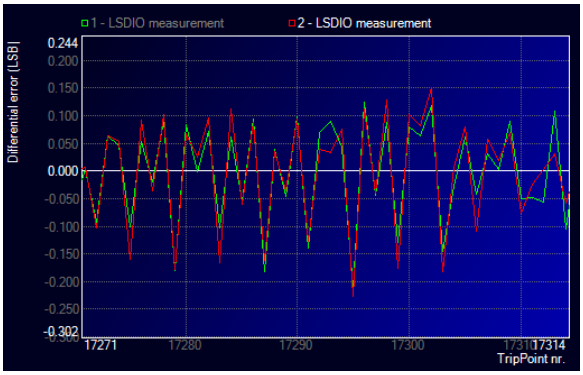

Figure 16 Two correlating DNL plots wit 16x4M points

Conclusion

We have seen that measuring 16-bit ADC linearity with the ATX7006 can be done very easily with great precision, but for extremely low-DNL converters, the multi-test histogram option in ATView7006 can go even further in reducing noise and increasing accuracy.

Document revision 1.0
Printed 22-Mar-10