

Getting the Max Resolution from Your Analog Measurements

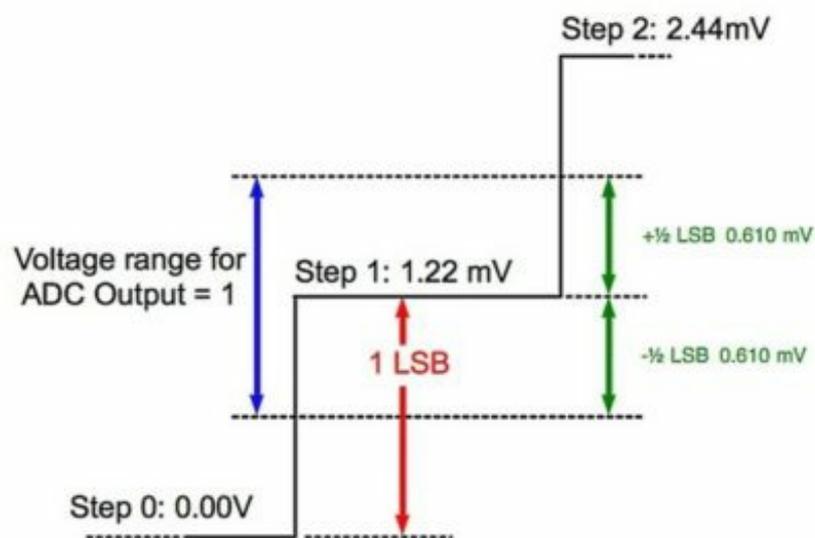
sealevel.com/support/getting-the-max-resolution-from-your-analog-measurements

Q: How do resolution and accuracy relate to an ADC? Do they mean the same thing?

A: No, each term refers to something different. Say I have an inexpensive plastic ruler with a centimeter scale marked at 1-millimeter increments. That scale gives me a 1-mm *resolution*. So I can measure, or *resolve*, a length to within the closest 1-mm mark.

When I place the plastic ruler next to an engraved metal engineering ruler with 1-mm marks, the plastic ruler has an error of about 0.5 mm over a distance of 15 mm. So the plastic ruler lacks the accuracy of the metal ruler. I also used a digital micrometer and found the same error for the plastic ruler.

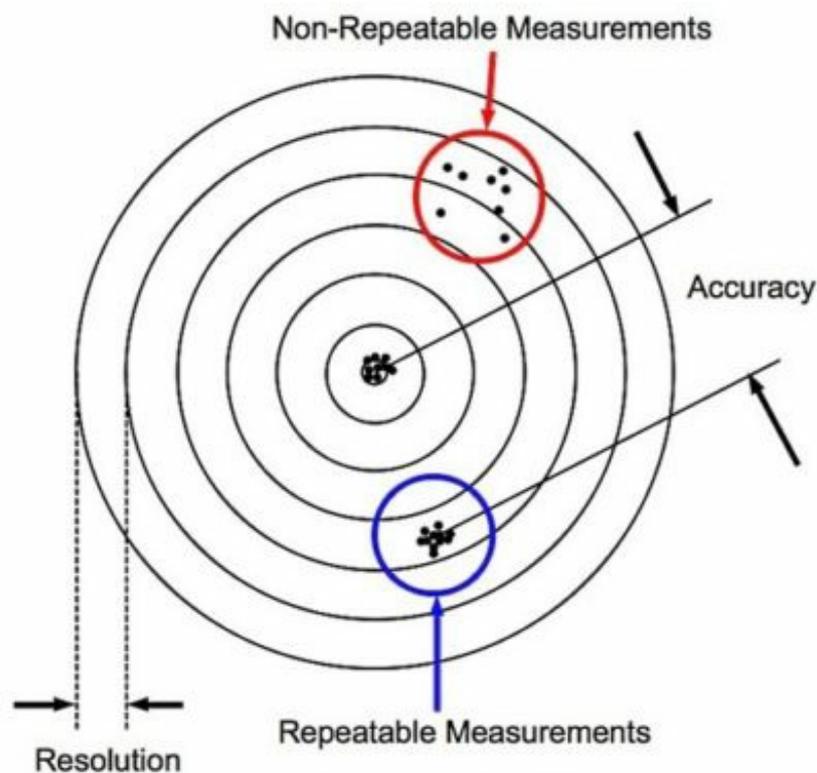
Now let's use as an example a 12-bit ADC with a 0 to 5.00-volt input range. For an unknown voltage within that range, this ADC will produce 2^{12} , or 4096, unique 12-bit *codes* for 4095 *voltage steps*. (Data sheets might note this specification as the number of *quantization levels* or the number of *digital output codes*). Our 12-bit ADC has a voltage step of $5\text{ V} / 4095\text{ steps}$, or 0.00122 volts (1.22 mV), as shown in the figure below.



Resolution: If we ground the ADC input we get the 12-bit value 000000000000_2 , equivalent to Step 0. Next we remove the ground connection and supply just enough DC voltage to the ADC input so the output changes to 000000000001_2 (Step 1). Does that mean the ADC has a 1.22 mV signal at its input? No, it means the signal can have a voltage within ± 0.610 mV of the Step 1 voltage as indicated by the green arrows. That span represents $\pm 1/2$ of the voltage “weight” assigned to each step. (That voltage also represents the voltage assigned to the ADC’s least significant bit, LSB). The red arrows mark this voltage span. So although we have a 12-bit resolution, we can only measure a signal to within ± 0.610 mV of each “step” voltage. Thus resolution equals ± 0.610 mV.

Accuracy: This specification defines how well the ADC output represents the applied signal. Does the ADC produce the code 0000000001_2 for an input voltage between $1.22\text{ mV} - 0.610\text{ mV}$ (0.61 mV) and $1.22\text{ mV} + 0.61\text{ mV}$ (1.83 mV)? Probably not.

Small errors, called quantization errors, determine how well the ADC's 12-bit output code represents the signal applied to it. You can have an ADC with 12-bit resolution that delivers slightly inaccurate results, caused by a reference-voltage error, electrical noise, minor circuit defects, and so on. ADC data sheets express accuracy in several ways, usually in terms of the LSB voltage step. Integral nonlinearity, for example, might cause a maximum accuracy error of $\pm 1\frac{1}{2}$ LSB, or $\pm 1.83\text{ mV}$. So the ADC still has a 12-bit resolution of $\pm 0.610\text{ mV}$, but an accuracy of only $\pm 1.83\text{ mV}$. The "target" diagram below illustrates resolution and accuracy.



You also should know about *repeatability*, a specification that indicates how well a series of measurements match each other. You might connect a stable, standard voltage source to an ADC or data-acquisition system and take 100 readings two minutes apart. A plot of the voltages vs. time should show a straight line within the accuracy specified for your measurement equipment. And if someone else sets up the equipment for the same measurement, the results should correlate well with the ones you took. If the line varies significantly the equipment might have a reproducibility problem, which means you can't tell the real measurements from the ones with voltage errors. You must eliminate the sources of reproducibility errors. For an overview of error analysis, please visit: <http://physics.appstate.edu/undergraduate-programs/laboratory/resources/error-analysis>.

Q: The ADC in my equipment has a 0-to-2.5-volt input range and must measure a sensor signal that varies only from +0.52 and +2.1 volts. Does that mean I lose some resolution in the ADC results?

A: No. A 12-bit ADC with a 2.5-volt input range always offers $2^{12} - 1$, or 4095 voltage steps, each separated by $2.5\text{ V} / 4095\text{ steps} = 0.000612\text{ volts}$, or 0.612 millivolts (mV). So you

don't lose resolution, although you get fewer than 4096 values from the ADC because the sensor's output does not "fill" the ADC's full 0-to-2.5-volt input span.

Q: How many voltage steps, or 12-bit values, can I expect for my sensor signal?

A: Your sensor has an *output range* of 0.52 to 2.1 V, which amounts to 1.58 V. Divide that range by 0.000612 volts / step (0.610 mV / step) and you will have 2581 steps, or values, from the ADC for your sensor's signal.

Q: So that means I can't take advantage of all 4096 possible values from the ADC?

A: Correct, but you might not care. The sensor-voltage limits you specified, +0.52 and +2.1 volts, have only two significant figures so you might not need all 2581 ADC values. Look at measurement results in the table below. (To make the small changes easy to see, the table below shows *calculated* voltages with four significant figures. On the Internet you can find information about how to interpret and use significant figures in measurements and calculations).

When the sensor puts out a 0.52-volt signal, the ADC produces the value 850₁₀. (The 0.52 V value divided it by 0.000612 volts / step.) When the sensor produces a 0.5206 V signal the ADC output equals 851. Likewise, a 0.5212 V signal from the sensor causes the ADC to put out the value 852. Do you need to detect those small voltage changes from the sensor? If not, your software could round the ADC data to values such as 850, 855, 860, and so on, or perhaps it could even round them to 850, 860, 870... Then you could relate those values to the physical phenomenon the sensor measures.

ADC Output Value	Sensor-Output Voltage
(Decimal)	(Volts)
850	0.5200
851	0.5206
852	0.5212
853	0.5219
854	0.5225
855	0.5231
856	0.5237
857	0.5243
858	0.5249
859	0.5255
860	0.5261
861	0.5268
862	0.5274
863	0.5280

864	0.5286
865	0.5292
866	0.5298
867	0.5304
868	0.5310
869	0.5316
870	0.5323
871	0.5329
872	0.5335
873	0.5341
874	0.5347
875	0.5353
876	0.5359
877	0.5365
878	0.5372
879	0.5378
880	0.5384
881	0.5390
882	0.5396
883	0.5402

Q: So how many rounded values would I get over the sensor's 0.52 to 2.1 V to range?

A: If you want values in steps of five values, divide the 2581 ADC values by 5 and you get 516 values. If instead you can use every 10th value, you would have 258 across the sensor's output-signal span. You might decide that number of values meets your measurement requirements.

Q: But what if I want to "spread" the sensor's 0.52 to 2.1 V signal across the entire 0 to 2.5 V ADC-input range. How do I do that?

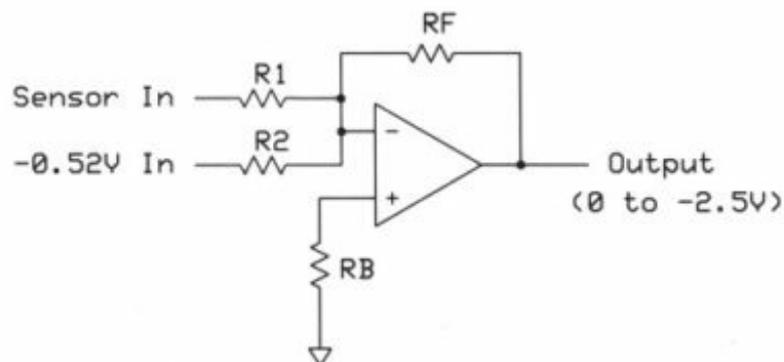
A: You need a summing-amplifier circuit. As an example your sensor signal varies from 0.52 to 2.1 V. First, the summing circuit must subtract 0.52 volts from the sensor's signal. The subtraction decreases a 0.52 V sensor output to 0 volts and it decreases the sensor's 2.1 V output signal to 1.58 V (2.1 V – 0.52 V). To stay with two significant figures, round the 1.58 V result to 1.6 volts. Second, the circuit must amplify the 1.6 V signal by a factor, x, so it becomes 2.5 volts. In this example, x equals 1.56, or 1.6. We call this an amplification factor, or gain.

$$1.6 \text{ V} * x = 2.5 \text{ V} \quad \text{or} \quad x = 2.5 \text{ V} / 1.6 \text{ V}$$

The schematic diagram that follows illustrates a standard *inverting* operational-amplifier (op-amp) circuit that performs the voltage subtraction and amplification. The ratio of the feedback resistance (RF) to each input resistance (R1 or R2) determines the amplification, or gain, for each input.

$$\text{Sensor gain} = R_F / R_1 \quad \text{and} \quad \text{Offset Gain} = R_F / R_2$$

Resistor RB simply provides a ground, or 0-volt, connection for the op-amp's positive input. The positive and negative inputs have signals—not power—applied to them. I left out the op-amp's power connections for the sake of clarity.



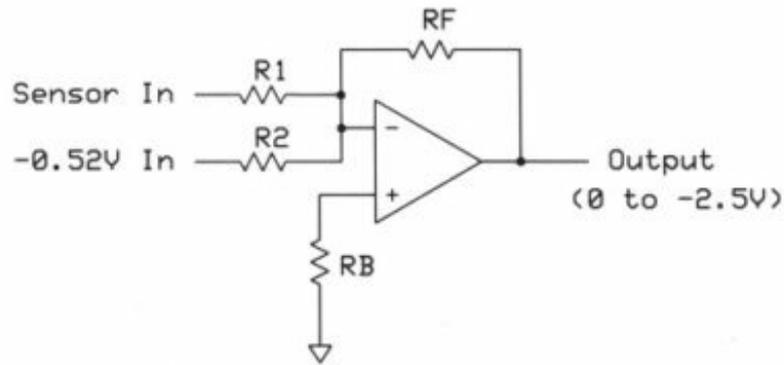
To obtain a 1.6-times gain for the sensor signal I chose a 10 kohm resistance for R1 and a 16 kohm resistance for RF. That ratio means the sensor signal alone (0.52 to 2.1 V) would appear at the output with a range from -0.83 to -3.32 volts. This op-amp circuit inverts the polarity of the output voltage.

The same gain factor, 1.6, must apply to the 0.52 V signal we want to subtract, so R2 equals 10 kohms, too. With only the -0.52 signal applied to the 10 kohm R2 resistor, you would measure becomes +0.83 volts at the output. As a result, the op-amp output goes from 0 V (-0.83 V + 0.83 V) to -2.49 V (-3.32 V + 0.83 V). That gives us the proper voltage-range expansion, but with negative voltages.

Another op amp could invert the signals again with a gain of one, or no amplification, to yield a 0 to 2.5 V signal for the ADC. This circuit requires a negative and a positive supply voltage, and a source of the -0.52 voltage we want to subtract.

An alternate approach uses a *non-inverting* op-amp circuit such as the one shown next. In this configuration, the resistances RF and RGND determine the amplification factor, which is always 1 or greater:

$$\text{Gain} = (R_F / R_{GND}) + 1$$



This circuit needs a negative-voltage source to produce the -0.52 V you want to subtract from the sensor signal. Other types of op-amp circuits can apply the 0.52-volt offset signal to an input to the negative terminal of the op-amp, so you don't need a negative supply. For more information about this type of circuit and design equations, see: Ardizzoni, John, "Efficiently Design An Op-Amp Summer Circuit," *Electronic Design Magazine*, August 26, 2013. Available at: <http://electronicdesign.com/ideas-design/efficiently-design-op-amp-summer-circuit>. I also recommend David Terrell's book, "Op Amps: Design, Application, and Troubleshooting," 2nd ed., Newnes, 1996. ISBN: 978-0750697026.

Q: So now can I amplify a signal to the full input range of a 12bit ADC and get useful information from all 12 bits?

A: In theory, yes. But from a practical perspective you want a sensor signal that *will not exceed* the range of the ADC you use. That means you leave some "leeway" at the upper and lower limits of the input range. For a 0-to-2.5-volt input range, you could adjust the sensor's signal via one of the amplifier circuits so it would only produce a signal for the ADC from, say, 0.1 to 2.4 volts.

Suppose a signal from a sensor causes a 12-bit ADC to produce the largest value it can: 4095. Does that mean the sensor created the maximum voltage or that the sensor output exceeded, or overloads, the ADC's maximum? We cannot tell because both conditions will produce the same value: 4095. Likewise, if a sensor signal goes below an ADC's lower voltage-input limit of 0 volts, you also get erroneous results. As a rule of thumb, don't push ADC inputs to their limits.

Q: I see the abbreviations SNR, SINAD, and ENOB in an ADC data sheet. What do they mean?

A: They stand for signal-to-noise ratio (SNR), signal-to-noise-and-distortion ratio (SINAD), and effective number of bits (ENOB). And they're important specifications for equipment designers. If you have an off-the-shelf data-acquisition system or other equipment that incorporates an ADC, the manufacturer might provide these specifications in a manual or data sheet.

Instead of trying to explain these measurements and what they mean in this answer, please refer to: Walt Kester's tutorial, "Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor," Tutorial MT-003, Analog Devices, 2008. <http://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf>. See also, "Count Your Effective Bits," *Design News Magazine*, April 7, 2008.

http://www.designnews.com/author.asp?section_id=1386&doc_id=224147.

Keep in mind that the effective number of bits—which you can calculate—never equals the accuracy of a real ADC. A 12-bit ADC, for example, could have an ENOB specification of 10.5 bits. That takes into account noise and imperfections in all electronic circuits. So even though you have an n -bit ADC you always have fewer effective bits.