



[12-bit ADCs in the 32-bit Atmel SAM D20](#)

[Steve Taranovich](#) - July 02, 2013

My friend and colleague, Paul Rako, and the former editor of this blog, is now at Atmel. I just wanted to convey to you his following view on the 12 bit ADC inside an ARM M0+ device and what the added dynamic range can do for you. I found it very informative and educational and entertaining, but that's Paul. ☺

As a life-long analog aficionado, I was delighted to see Atmel put 12-bit ADCs (analog to digital converters) in the new ARM M0+ based SAM D20 microcontrollers. A 12-bit converter gives you that extra edge of precision. Eight bits is 256, so nine is 512 and 10 is 1024. You might think that with a 10-bit LSB (least significant bit) being 1/1000 or 0.1%, you could do accurate measurements. You have to remember is that a 10-bit converter can only give you ten bits under the most ideal conditions. You can't be clocking it very fast since the ENOB (effective number of bits) drops off the faster you clock the converter. And even though the converter has a nominal 0.1% resolution, your accuracy will depend on the reference, your analog signal chain errors, and any external noise and drift that can seep into your system. So starting with a 12-bit converter is a good way to insure that you will have margin for all those other analog errors that are sure to get into your system.

You will note some new oscilloscopes have adopted 12-bit sampling. The benefit is the increase dynamic range. You can take one sweep and get more details of the smaller variations. This is why I used a 12-bit converter in an automotive diagnostic tool when I worked at HP years ago. The ignition coil signal has wide dynamic range. When the "points" or transistor opens, the coil primary voltage will shoot up to well over 100 volts. Then the spark begins and the signal falls to 12 volts, the so-called "firing line". You want to know the peak voltage to understand the initiation of the spark event, but you also want to know the details of all the variations on the firing line. A 12-bit converter let us do this, all in one acquisition event.

So keep the Sam D20 in mind if you are doing data acquisition. Even if you think 10 bits is enough, it's always nice to have 12 bits to have some design margin. Oh and here are a couple tips that helped me. One of your best tools is a fully-differential scope probe. At HP I was getting 8-bits out of our 12-bit converter, until my buddy Wayne Yamaguchi showed me how you can use a diff-probe to track down where the bits were going. You can't just invert a scope channel and combine channel one and two—the ac response is not identical in two scope channels. You need a real high-zoot big-dollar externally-powered honest-to-gosh differential scope probe.

Now you can poke around and start looking at the ac on various "grounds" on your PCB. We were losing 4 bits because the PCB layout person had "grounded" the reference IC with a tiny little trace that wound through the other signals before it nailed to the ground plane. The diff probe showed the ground pin on the reference jumping around relative to the ground pin on the ADC. It took a few seconds to find the problem after we saw that.

The other tip is to synchronize the ADC acquisition to your switching power supply or other major

current draw event in the system. With that diff probe we got to 11 bits. But Wayne is as much a perfectionist as I am. We were losing that last bit because I had to have 4 switching power supplies on the board, as well as a battery charging switcher for a local Ni-MH battery pack in the product. It soon became obvious we could not shield the switcher noise out of the converter chain.

Some of the switcher noise was magnetic fields and the only way to eliminate those is with thick sheets of steel or μ -metal. So rather than fight the switching noise, we just made the system dance to its tune. First I redesigned the switching converters so they were all synchronized off a master, I think it was the 5V rail converter. Then I just did the acquisitions synchronously with that master clock. Sure there was power supply ripple that caused error and switching noise that caused error. But by taking a sample at the exact same time in the switching cycle it made those errors effectively disappear. I actually designed the switching converters so their frequency was equal to the acquisition frequency I wanted. Even if you don't have that luxury, you can still null out a lot of converter noise by synchronizing your power.

Let's say you are taking 10 acquisitions for every cycle in the power system. Well now you can short the acquisition inputs and take the data. You will get a few bits of signal, but that signal will be repeatable and predictable, so you can just store the magnitudes and then subtract them out of your actual acquisition. You do need to know the phase of the converter clock relative to your acquisitions, so you can subtract the right offsets from the right data, but this method can get you closer to your 12- or 16- or 18-bit design intent.

OK, I said two tips, but have worked myself into a system-design frenzy. The last tip is to not cut your ground planes up way analog semiconductor companies tell you to. Since the SAM D20 converters are inside the chip, at least you can't make that mistake. The advice to have separate analog and digital grounds on the board, and to connect them at only one point under the IC was dreamed up by semiconductor company application engineers that only had to put a converter IC on demo board all by itself.

What are you supposed to do when you have 6 converters? These apps engineers were more like test engineers trying to get the best out of their one little chip, rather than system design engineers that have to get an entire product working. No, you want a solid ground plane under your entire PCB (printed circuit board). You want to nail it to the chassis at multiple points around its periphery. The Ham radio and RF engineers are much more intuitive when it comes to grounding, shielding, and the difference between a power supply return, which we all *call* ground, and a shield, and a real earth ground— a 10-foot copper-clad steel rod hammered into the dirt.

Take my advice; just buy [Henry Ott's](#) famous book, [*Electromagnetic Compatibility Engineering*](#). It's the best \$106.56 you can spend to learn grown-up engineering. Having worked at the Long Lines part of Bell Labs, Henry understands that noise can creep into your circuits. His solution is not to cut up ground planes, which will just make you fail FCC (Federal Communications Commission) EMC (electro-magnetic compatibility) laws. Henry advises that the way to keep your digital noise out of your analog is 1) part placement and 2) routing discipline. Even better than just buying Henry's book is going to his seminar, which does cost a lot, but he gives you a free book and you get the course notes from the man himself.

I see Henry is in a seminar at the [2013 IEEE EMC Society Symposium](#) in Colorado August 2013. The next [Henry Ott seminar is in September in Detroit](#). It costs \$1295 before August 16 and \$1495 after. I have attended one and it is the best money your company will ever spend. When your next product cannot meet its specs or pass FCC or TUV or CE, you will be the person that knows exactly what to do to fix the problem, rather than sprinkling "prayer beads" (ferrite beads over circuit traces) on your board and hoping it passes. Then you will get that big promotion and you can make sure the

next system is designed to pass from the start, using Henry's time-tested principles.