# Design a DSP lock-in amplifier, Part 2: Design methodology

Steve Hageman - December 27, 2017

In Part 1, I discussed the background behind lock-in amplifiers. Here, I'll cover analog and digital lock-in amplifiers, software-defined radio, de-embedding, and the hardware and software architecture of the design. A sidebar at the bottom of this page discusses why I code the components for this design.

In the mid-1980s, analog lock-in amplifiers gave way to digital signal processing (DSP) based designs. These DSP-based lock-in amplifiers were based on the common quadrature or IQ detection method still used in digital demodulators, including software-defined radios (SDRs) (**Figure 9**). The enabling technology was the availability of high speed, 16-bit analog-to-digital converters (ADCs).

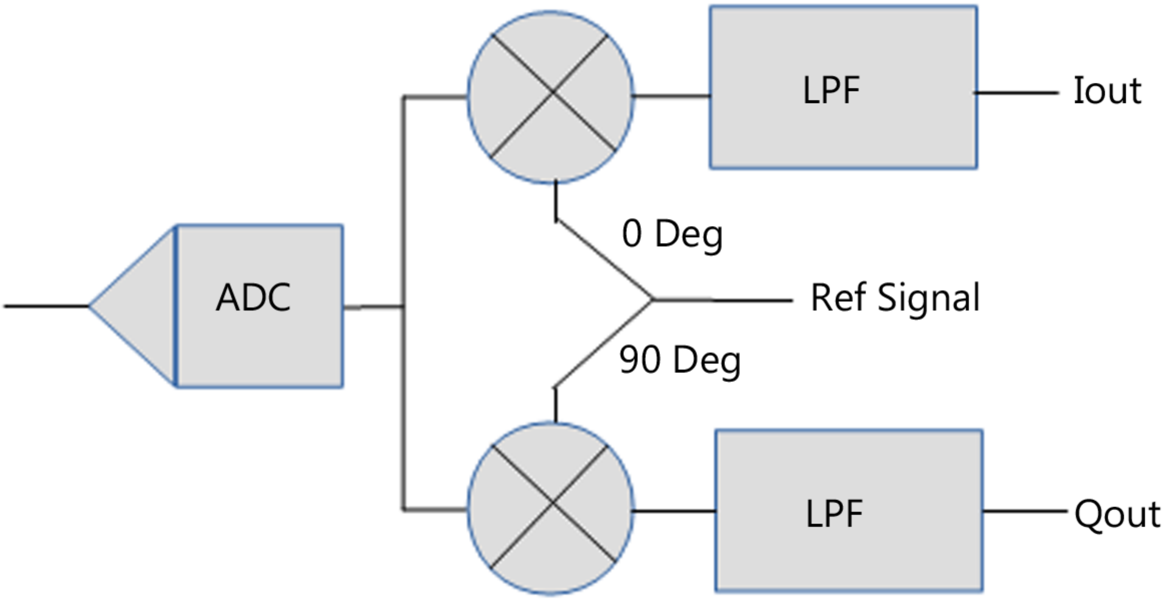

**Figure 9** In a DSP-based lock-in amplifier, everything after the ADC input is implemented digitally.

DSP-based designs have three advantages over analog designs:

- All the processing and filtering after the ADC is done digitally. That eliminates analog's matching, accuracy, drift, and tuning problems.
- The demodulator implements both an in-phase and a quadrature detector section so that the actual

magnitude and phase of the input signal can be determined for any phase relationship between input signal and the reference. This was a very important performance improvement.
- The reference signal is implemented as a digital sine wave. When combined with a true, digital multiplying demodulator, this technique eliminates the third-order response problem of the classic analog demodulator in **Figure 6**.

Commercial DSP designs have improved in performance over the years and now can operate from DC to several hundred Megahertz.

**A lock-in for experimentation**
Commercially available lock-in amplifiers contain many features and usually have built in displays and user Interfaces. This makes them easy and quick to apply to a variety of experiments. But, the processing algorithms and hardware configurations are "as supplied" by the manufacturer. This "closed system" configuration stifles experimentation into new approaches.

Experimentation today is defined as an "open ecosystem" configuration of hardware and software that can be modified or extended at will and as needed. With this need in mind, I designed a lock-in amplifier platform for quick experimentation in both the hardware and software domains.

**Hardware design**
The basic hardware design of **Figure 10** is built on two main PCB sections. The main board contains the power supplies, 32 Bit microprocessor, direct digital synthesis (DDS) source and two fast 16 bit ADC’s. These base functions are needed for all conceivable configurations. See "Choosing appropriate hardware" at the bottom of this page for the rationale behind my choice of components.

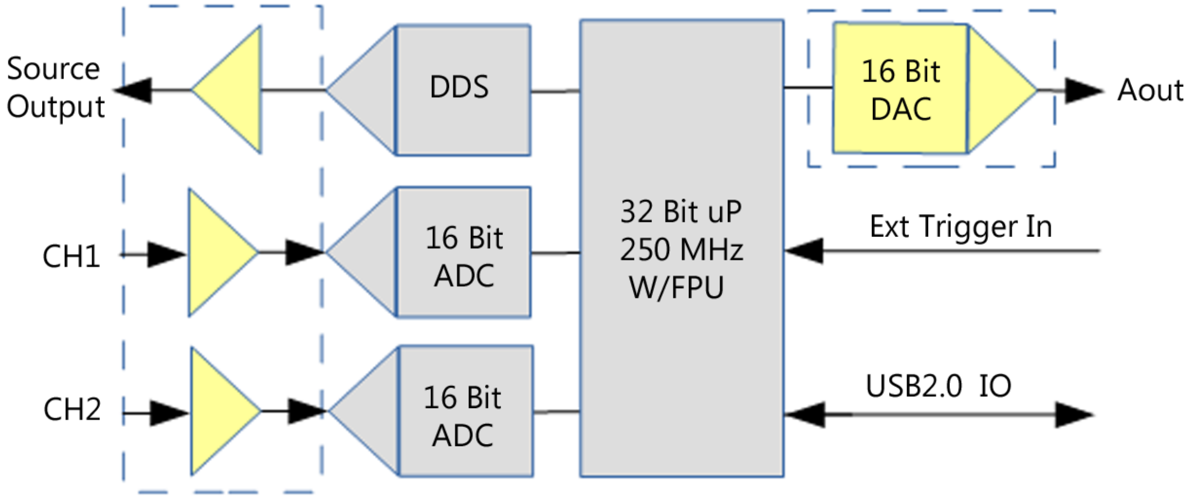

**Figure 10** A modernized Lock-In platform designed for experimentation. A single 32 bit microprocessor controls all the digital functions of the instrument. The Analog sections (yellow blocks) are quickly replaceable via a mezzanine PCB for easy adaptability to any requirement.

The ADC core functions consist of two high performance 16-bit ADCs that sample at up to 2 Msamples/s in 16-bit mode or up to 2.4 Msamples/s in 8-bit mode. Most designs that use high-speed ADCs have a separate FPGA to control data acquisition and storing the ADC samples to memory. This design uses the latest high-speed, MIPS based, 32-bit microprocessor to do this function without the need for an FPGA. The microprocessor contains 512 kbytes of on-board RAM for saving ADC data directly and by using the microprocessor to directly control the ADCs eliminates both and FPGA and and external RAM chip. More importantly, however, this configuration saves you from having to deal with yet another piece of code + compiler + programmer that would be needed in working with a separate FPGA.

The microprocessor chosen also includes a peripheral divide-down counter that is clocked from the low noise system clock. The output from the divider is a low jitter square wave used to generate the sample trigger clock for the ADCs. The ADC sample trigger clock is easily programmable from DC to 2.0 Msamples/s or 2.4 Msamples/s as dictated by the desired sampling rate.

A commercial direct-digital synthesis (DDS) IC provides the source output. It's a sine wave of up to 25 MHz, but you can modify it to become any wave shape. The main board contains attenuation and offset controls that control the output up to ±5 V maximum with 2 mV resolution. The offset can also be controlled over the full ±5 V range. Source harmonics are better than -55 dBc, which is comparable to a "good" analog oscillator.

To adapt this platform to any experimental configuration, you need to change the analog IO shown in yellow in **Figure 10**.

The replaceable analog front end (AFE) is built on a plugin mezzanine PCB that can be quickly designed around any specific application need. The AFE typically contains the signal input connectors and the input signal conditioning as shown in **Figure 11**. Connection between the boards are made through four 10-pin board-to-board connectors. The connectors provide power and digital IO to the AFE mezzanine. The connectors also provide an analog path to the input of the ADC converters.

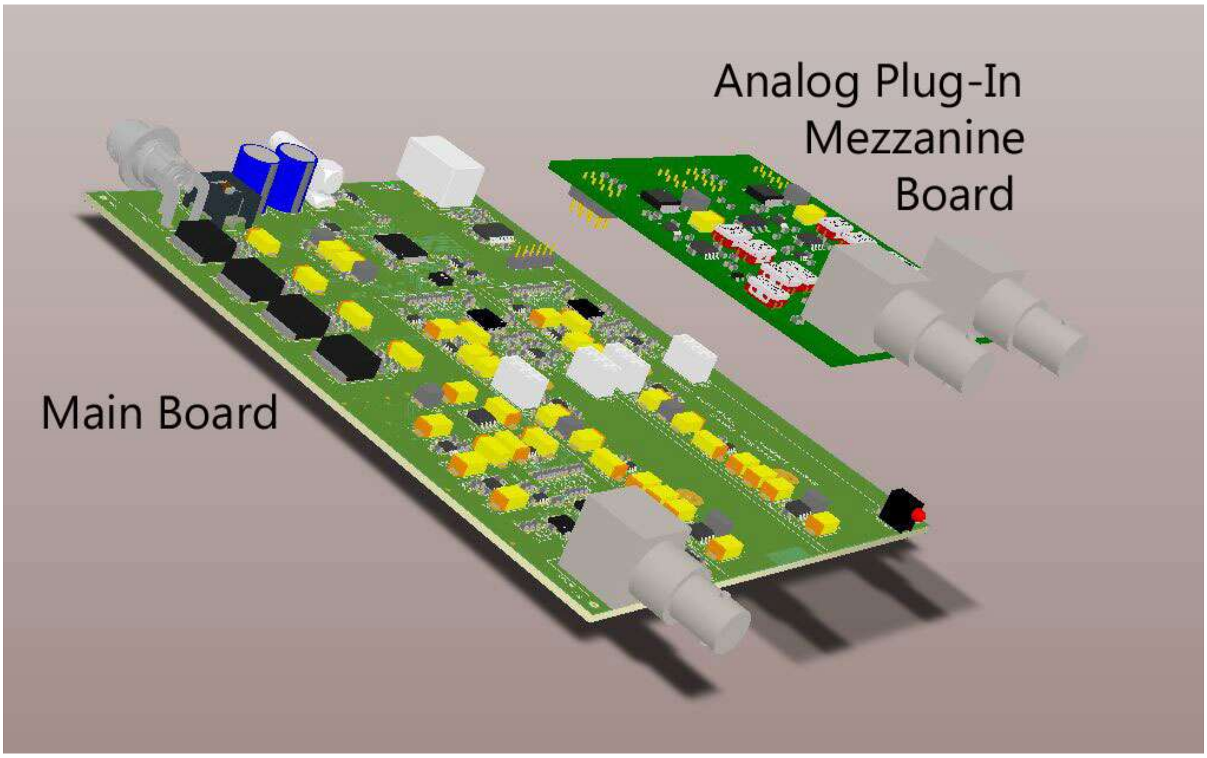

**Figure 11** The analog IO is what needs to change to adapt the new lock-in amplifier platform to any possible need. In this design the analog IO functions are placed on an easily replaceable mezzanine PCB that plugs into the main PCB.

Modern analog functions still need digital control and this is accomplished by having two serial digital IO (SPI) controls, one for each analog channel brought up to the mezzanine PCB. These two channels may be further expanded by using SPI IO expanders to any number of digital channels.

To provide maximum isolation between the AFE channels, each AFE channel is supplied by independent, low noise, linear regulators supplying +6 V and −5.5 V. A system power rail of +7 V at high current is available to the mezzanine for driving high current loads such as: LEDs, relays, etc.

One AFE design that has been developed (**Figure 11**) is a general purpose AFE that uses 4 nV/Hz JFET input amplifiers, has a gain range of 0.1-1000, input impedance of 1 MΩ and a bandwidth of up to 15 MHz. This general-purpose AFE lets you measure signals from: ±2 mV to ±20 V full scale.

Other designs have included:

- Ultra low-noise bipolar input amplifiers
- Photodiode/transimpedance amplifiers
- Ultra high impedance amplifiers
- Piezo transducer (charge) amplifiers
- Low DC drift, chopper stabilized amplifiers

In some applications, the lock-in amplifier is part of a closed-loop control system that usually includes some form of analog output. Because the lock-in processing algorithms are digital in this design, the processing results in a digital number. This digital result can then be fed to a suitable DAC scaled as required for your needs. The analog output in this design resides on the AFE Mezzanine board. This location makes it easy to modify and adapt the output to fit any system requirements.

Because imperfect parts are used for the source and AFE gain setting resistors, the gain varies from board to board. Thus, a calibration process is required for all designs. The main board contains a general purpose 32 kB EEPROM that can be used for calibration memory. This EEPROM can be used to store up to 8 k floating point numbers and these can be used for storing calibration constants. General purpose calibration routines have been developed that use the on-board DDS source as an AC test signal. The source is first calibrated with the help of a high resolution true RMS reading DVM.

Using the calibrated DDS source with suitable attenuators and making actual input comparison measurements with a DMM, you can calculate the AFE board's gains and offsets and store them on board. A calibration performed in this manner takes only a few minutes and has the accuracy and traceability of the DMM.

### Software design for experimentation

Modern digital radios are said to be "software defined" because the software defines the demodulation characteristics of the radio. In analog radios, hardware does the demodulating. In SDR, once the signal is properly digitized, any demodulation may be performed simply by changing the demodulation algorithm. Coupled with open software, this approach allows for quick experimentation and "breadboarding" of new ideas in software without having to change the underlying hardware.

This design for a lock-in amplifier uses the same technique. The signal is digitized early in the processing chain. By varying the number of samples taken, sample rate, and the processing algorithms, you can achieve nearly any desired demodulation.

One common digital processing technique is to oversample the input signal. This is used to increase the effective number of bits (ENOB) in the ADC. Because this design uses a 16-bit ADC, the natural dynamic range is approximately 6 dB times the number of bits or approximately of 96 dB. If there is sufficient random noise in the signal to "dither" the LSBs of the ADC, then by sampling the signal 16 times for each desired output sample, we can increase the effective resolution by a factor of four (square root of 16 = 4). This is like adding two bits to the ADC. Oversampling by dither noise is usually easily accomplished because, whatever signal transducer is used, it will usually have sufficient noise to dither the ADC LSBs. The AFE gain needs to be set just high enough to get the signals natural noise floor above the ADC noise by a few LSB bits [4]. If required, external dither can also be added to the signal path.

Undersampling is another common processing step often found in SDRs. With undersampling, the ADC's sample-and-hold function acts like a mixer, which causes multiple digital images to appear in the undersampled output. The resulting baseband signal can be digitally low-pass filtered to remove any other mixing products. Thus, you can change the down-conversion at will because all the down-conversion parameters are adjustable in software. Another benefit of undersampling is the required ADC sampling rate and therefore the data rate is greatly reduced, simplifying the ADC memory hardware design.

You can exploit undersampling in lock-in amplifiers. For example, assume you need to measure not only the fundamental response, but measure the second and possibly third harmonic of the signal at the same time. Some analog lock-in amplifiers have the capability to measure the second harmonic, but not the third and never all at once. With this software defined lock-in amplifier, you can measure all three signals at once, as shown in **Figure 12**.

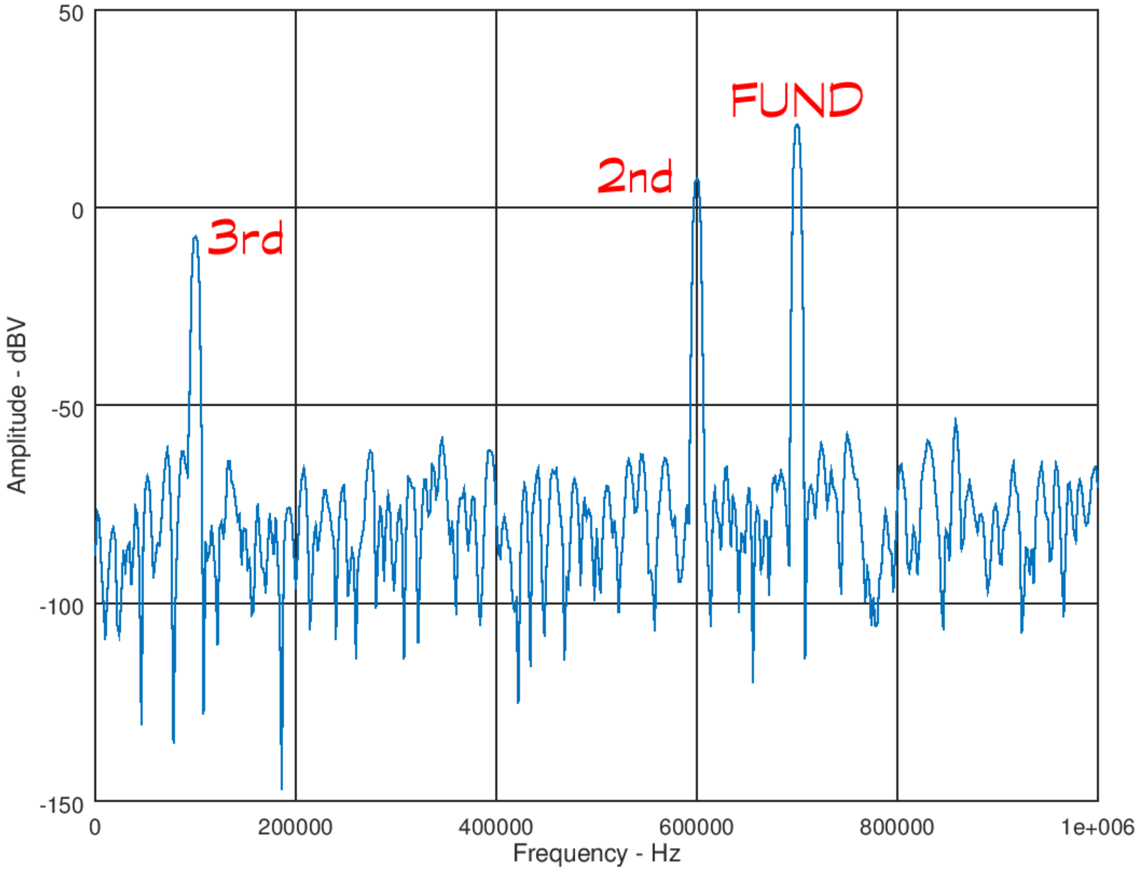

**Figure 12** A 10.7 MHz signal was digitized at 2 Msamples/s, the resulting fundamental signal, the second and third harmonics of the signal all alias back to baseband and are still easily distinguishable.

In **Figure 13**, a 10.7 MHz signal is sampled at 2 Msamples/s. The AFE analog signal bandwidth was >32 MHz and the ADC used in this design has a sample-and-hold bandwidth of 50 MHz. This allowed the fundamental, second and third harmonics to be digitized simultaneously. The fundamental then shows up at an apparent base band frequency of 700 kHz, the second harmonic appears as 600 kHz and the third harmonic appears as 100 kHz. Because the experiment's frequency of operation is determined by the internal source frequency from the lock-in amplifier, the fundamental and harmonic frequencies are also known and the multiple aliasing doesn't matter because all the signals' frequencies are known and separate.

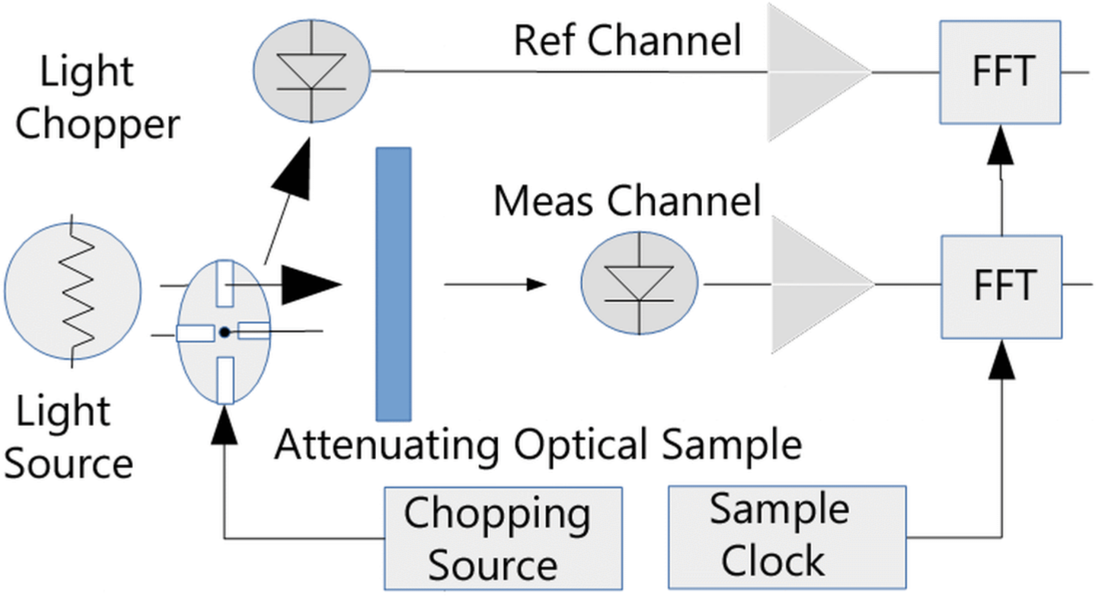

**Figure 13** With dual input channels, a reference channel measurement can be compared to the attenuated sample measurement. From that measurement you can find the complex gain and phase properties of the sample.

The ability to separate and identify aliased signals is a unique feature of a source/receiver instrument. The signal and all the harmonic frequencies are known because they are set by the on-board source and they can be avoided or combined as desired in the resulting output spectrum.

In the example shown in **Figure 13**, all three signals can be processed digitally at once because they can all be arranged as baseband signals at known and different frequencies. This processing capability is simply unobtainable in any analog or current digital lock-in amplifier and is a direct result of the software defined nature of this design.

Digital lock-in amplifiers use the classical processing approach that is shown in **Figure 9**. This DSP scheme mimics a zero-intermediate-frequency analog down converter as found in SDR designs, but this is not the only available approach. Complex fast fourier transforms (FFTs) can be performed on the digitized signals that will yield similar and possibly more useful results.

All FFTs return a complex result that provides magnitude and phase information similar to the I/Q output of the conventional DSP processing lock-in amplifier of **Figure 9**. By using one of the lock-in amplifier's channels to measure the source—as a reference and the other channel on the detector output—gain and phase information may be reliably measured and processed. A further benefit is that a gain-ratio measurement of the output/input signal can be made effectively eliminating any light source intensity fluctuation from the measurement result. This results in improved measurement stability. Using an FFT approach, a conversion scheme like was shown in **Figure 13** is possible, where each signal can easily be separated in frequency, as each signal shows up in a different FFT bin.

Demodulating a signal like the one in **Figure 13** simultaneously is impossible using the conventional DSP approach of **Figure 9**. You can, however, apply noise floor reduction techniques used extensively in SDR's to this open software lock-in amplifier design. One such technique is noise floor de-embedding where you can accurately measure the true input noise. Any signal that measures 3 dB above the noise floor is actually *at* the noise floor because the noise powers add to a signal giving an apparent 3 dB amplitude above the noise. A correction can then be made for the noise floor, resulting in a better estimate of the signal's actual amplitude. In this case, the signal would be reported with a 3 dB lower amplitude than what was measured. This processing has effectively de-embedded the noise floor from the measurement. Practically, this technique may increase the dynamic range by 7 dB to 10 dB.

Another technique you can use to reduce noise is cross-spectrum analysis (**Figure 14**). Here, two or more analog input channels connect to the same signal source. By vector summing one channel's FFT result directly with the complex conjugate of the other channel, you can get the noise of the analog channels to average out, but the signal won't. With enough averages, the noise floor of the instrument can be effectively lowered by 20 dB or more.

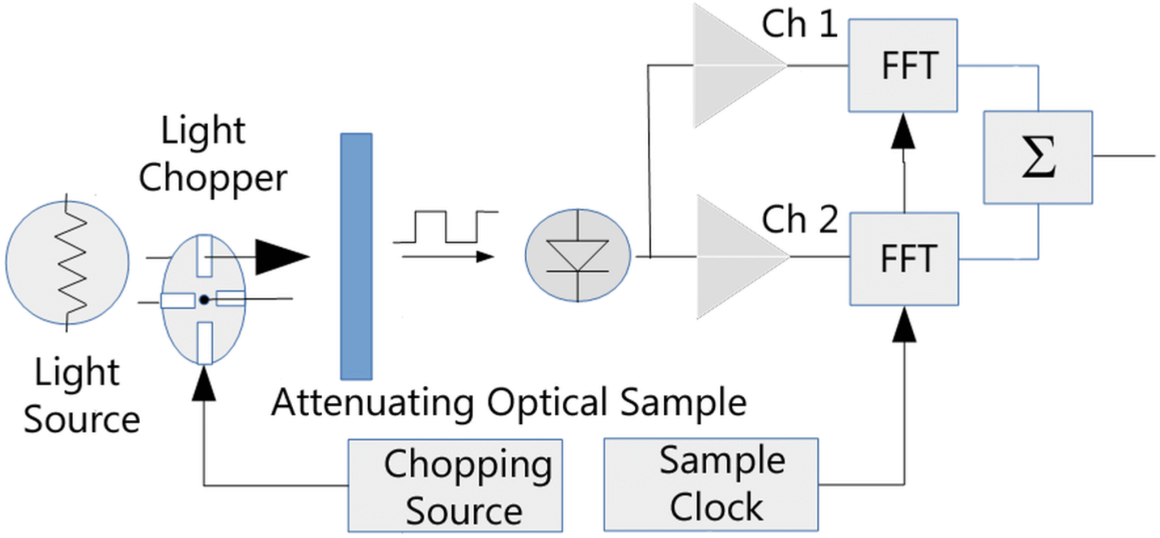

**Figure 14** Perform cross-spectrum analysis by vector summing one complex FFT output with the complex conjugate of another identical channel. If the additive noise is uncorrelated in both analog channels, then the noise will cancel with averaging. Given enough averages, this technique can enhance the noise floor by 20 dB or more.

Cross-spectrum analysis works because both analog channels will have uncorrelated noise and this noise can be vector averaged out, but the real signal in each path is correlated and will not average out.

The processing gain of a cross spectrum analysis is,

Reduction dB = 5 dB × $\text{Log}_{10}$(Averages)

For 1000 FFT averages, the noise floor can be reduced by 15 dB (or 5.6× less voltage noise).

Once the hardware design reaches the achievable limits of the input amplifier device noise, noise de-embedding or cross-spectrum analysis is the only way to get real reductions in the noise floor.

You can apply any or all of the above techniques to this common hardware platform because of the open software nature of the design. These techniques cannot be easily applied to any commercial instruments because of the impossibility of changing a commercial instruments software.

**Signal processing**

Depending on the requirements, the DSP may be performed entirely by the on-board microprocessor or raw data may be transferred to a computer for further processing. The microprocessor used here can perform a highly optimized 16 kpoint FFT in under 30 ms. The 32-Bit microprocessor's highly optimized DSP commands along with 512 kbytes of on-board data RAM allow for extensive signal processing performed by the instrument itself.

The 32-Bit microprocessor also includes is an integrated floating-point processor unit (FPU), which can provide both single and double precision floating point results in a single instruction. Real double-precision math in hardware opens a whole new realm of computational possibilities that aren't available when emulating floating point in software, because of the speed advantages.

With the open-software approach, the signal processing can be partitioned between the on-board microprocessor and the control PC in any way that makes sense for the specific application.

**Hardware control**

The microprocessor on the main board runs self-contained applications, which are useful for closed loop system where the output is an analog signal. Many times, however, lock-in amplifier applications use the instrument as a measuring device transferring commands and data to a PC that is used for further processing, display and storage.

To facilitate PC control, a USB 2.0 connection is supplied in this lock-in amplifier design. The USB connection is connected to the on board microprocessor's UART and runs at up to 3 Mbaud.

If isolation is required to the PC to cut ground loops, a simple add-on USB isolator such as the Analog Devices ADuM4160 can be inserted in the USB cable between the instrument and the PC and the instrument may be battery powered.

Control commands to the lock-in amplifier are processed through a SCPI like command parser. For instance, a command to start a data acquisition sequence, and read the result would be: "CHANel1:MEASure?", which can be shortened to: "CHAN1:MEAS?". The command above would initiate a data capture on channel 1 and return the data array result.

With the open software nature of the instrument, any command may be added or multiple operations can be combined into one as may be desired for the experiment at hand.

This makes control of the instrument easy and intuitive. In SCPI, a standard set of commands is always present. For example:

*IDN?, which asks the instrument to identify itself,
*RST, which causes the instrument to preset itself to the power on state.

The command parser API is easy to extend making the addition of application specific commands straightforward. It's useful because different AFE designs will almost certainly require custom command sets for control.

**Wrapping the design**

The hardware design was sized to fit a commercial extruded enclosure that measures 5×6×2 in. tall. The front and rear panels of this design are flat plates which allows custom front and rear panels to be easily made to fit any custom application.

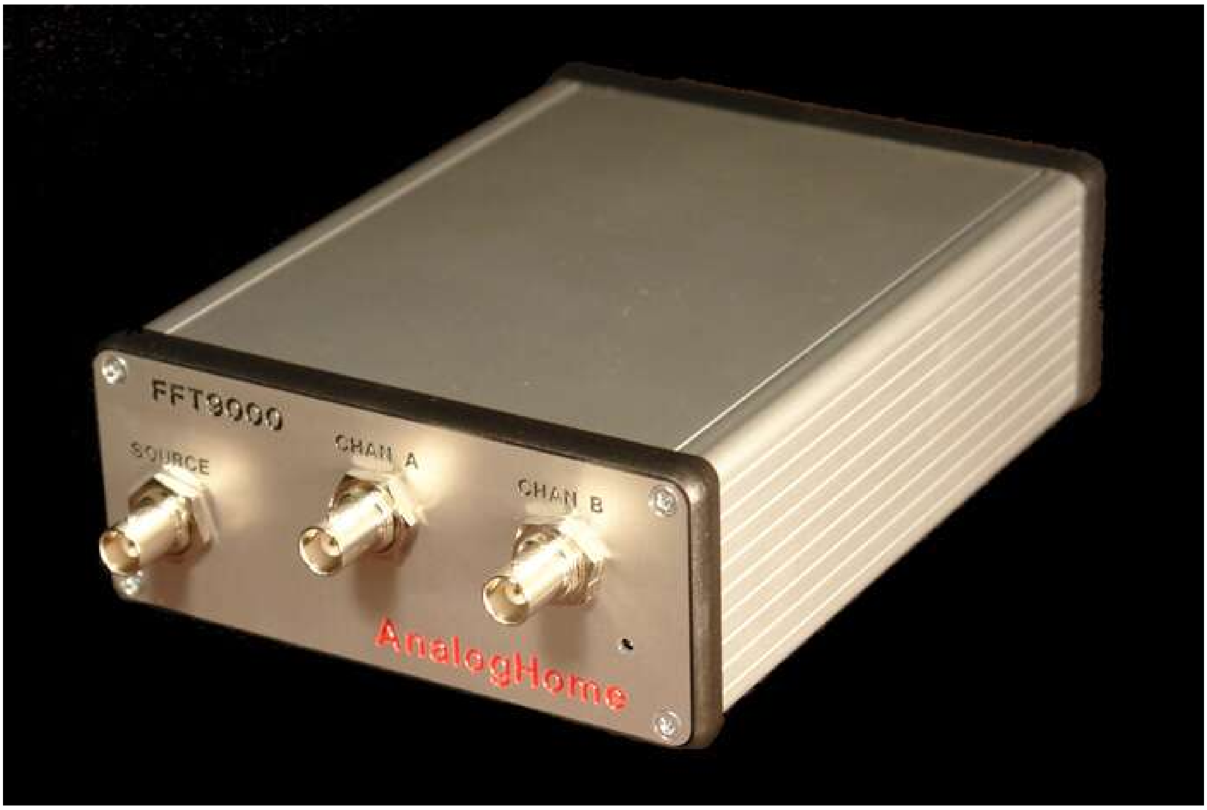

**Figure 15** The experimental lock-in amplifier design fits in a 5×6×2 in. commercial enclosure.

Using a simple hardware chassis design allows for easy access to the hardware for development and troubleshooting purposes as the hardware slides out of the chassis for access. The all aluminum enclosure also provides electrical shielding and environmental isolation from drift inducing air currents.

The design with the universal JFET input AFE consumes about 7.5 W total from a 9 VDC source, which leads to only a few degrees C temperature rise of the enclosure.

**Notes / References**

[1] Even chopper amplifiers have 1/f noise. Though choppers are better than other types of amplifiers, their noise will eventually have a 1/f shape if you look at a low enough frequency.

[2] Some "advanced" analog lock-in amplifiers implemented a quasi-sine wave demodulator based on the Walsh function and implemented with a four-step quantized sinewave multiplier. These designs had the effect of reducing the odd harmonic responses about 20 dB from their square wave based counterparts. See: Princeton Applied Research Model 5210.

[3] The same Analog Dialog Magazine that introduced an IC form of Figure 5, the AD630, also had articles on DSP-based chips and methods that could duplicate these analog functions digitally. See: Analog Dialog V17n1, 1983. The AD630 is limited to demodulation frequencies of less than 200 kHz.

[4] The rule of thumb here is that for oversampling to work well that the random noise that dithers or modulates the ADC LSB's should be a couple of bits or at least: 2 * 6 dB = 12 dB above the natural ADC noise floor. The noise should have a normal distribution. Usually the signal transducers and AFE amplifier noise and gain can be adjusted to meet this goal. Sometimes extra dither is added to the circuit directly ahead of the ADC.

*—Steve Hageman is owner of AnalogHome.*

**Related articles:**

- Design a DSP lock-in amplifier, Part 1: Background
- FFT: Equations and history
- Measuring small signals accurately: A practical guide
- SCPI programming: Strengths and weaknesses
- FFT plots provide insight to A/D performance
- Tips and techniques for power supply noise measurements
- Use synchronous detection to make precision, low-level measurements

---

**Choosing the hardware**

Most hardware is now a commodity, but there are still valid reasons for picking one set of ICs over another, often for leveraged code base considerations. Here are the reasons for my hardware choices.

**Microprocessor**
I use both Microchip, MIPS based PIC32 processors and the STMicro ARM based STM32 chips. Both these processors have DSP oriented instruction sets and have upper middle performance range core frequencies of 200 MHz plus, which was plenty for this project. Ultimately for this design I used a PIC32MZ processor in this design because of a few hardware features.

**ADC**
I needed a parallel output 16 bit IC that ran at greater than 2 MSPS for this design. I particularly wanted a wide bandwidth sample and hold of >25 MHz as I have some interesting experiments that have center frequencies of 10.7 and 21.4 MHz. An internal reference was a must to keep the noise uncorrelated between the two ADCs as much as possible. A Bipolar input was important because at very low frequencies the AC coupling capacitors get huge and a Bipolar input can eliminate the coupling capacitor in many low frequency noise tests. This lead to one choice really—The Linear Technology LTC2369-16. The LTC2369-16 also has tri-statable outputs. I wired each ADC output to one 16-bit wide port on the microprocessor and ping-pong between ADC's to get a full 16-bit word from each on a single IO port. Reading 32 bits at a 2 Msample/s rate is equivalent to a 64 Mbps data rate. The ADC data is stored directly in the processors 512 kbytes of RAM eliminating the need for an external FPGA and memory.

### DDS source

There is really only one source for these parts now: Analog Devices. I used the ubiquitous AD9834 for this design. The only downside to these chips is that the output DAC is only 10 bits. This limits the harmonics and spurious performance quite a bit, but the performance is still better than a "good" analog oscillator and the output is programmable to 28 bits of frequency resolution. To control the output amplitude, an AD5452, 12 bit multiplying DAC is used. It's the only one available with the bandwidth high enough for this application.

### Analog front end

The op amps used are combination of Linear Technology, Burr-Brown (Now TI), and Analog Devices. High performance op amps are an art and here is one market that is not so commoditized. Each manufacturer has one niche that they excel in, hence you need to buy from all to get the best performance.

### USB interface

Why use an FTDI RS232RL chip? For one thing, it has stable drivers. Plus, FTDI provides a DLL interface instead of just the usual COM port interface. Even though the PIC32MZ has USB hardware, having a separate USB interface IC speeds development for a low-volume instrument. This is because you can burn a new set of microprocessor bits or reset the processor and the USB interface does not reset. Thus, your PC application doesn't crash and you don't have to spend time restarting everything. Once you try this, you'll never go back.

The PIC32 has an independent programmable (and low jitter) clock divider that was perfect for driving the ADC sampling clock. Also the PIC32MZ family has a double precision floating point unit in a small pin count 64 pin package. I had no code base to leverage here so I didn’t need to take that into consideration. Both processors have excellent GNU based compiler tool chains and DSP support libraries available for free.

### Power supplies

I use Linear Technology and Texas Instrument parts almost exclusively. LT3080s are nearly everywhere except when lower dropout is needed then the I switch to TPS77301 and TPS72301 parts. To generate negative voltages, I use heavily filtered LTC1144 charge pumps. Heavy filtering keeps the charge pump switching frequency out of the instrument's noise floor. With an FFT-based instrument, you can see every "glitch and whistle" inside of your designs, so use extra filtering. Each analog section has separate linear regulators to maximize isolation.