The Poor Ham's Scalar Network Analyzer

Nick Kennedy—WA5BDU kennick@gmail.com

I can't seem to start a talk or article without first mentioning that it's a great time to be a ham builder and experimenter. Availability of inexpensive components and modules, free software and a huge knowledge base available on the internet, both in static form and in consultations via discussion groups. OK, I got that out of the way. Onward to the Poor Ham's Scalar Network Analyzer (PHSNA).

In September 2013, Jim Giammanco, N5IB and Jerry Haigwood, W5JH announced via discussion group postings their collaborative project for a versatile measurement system integrating a DDS RF generator, an Arduino microcontroller board, and an RF log power meter based on the AD8307 chip.

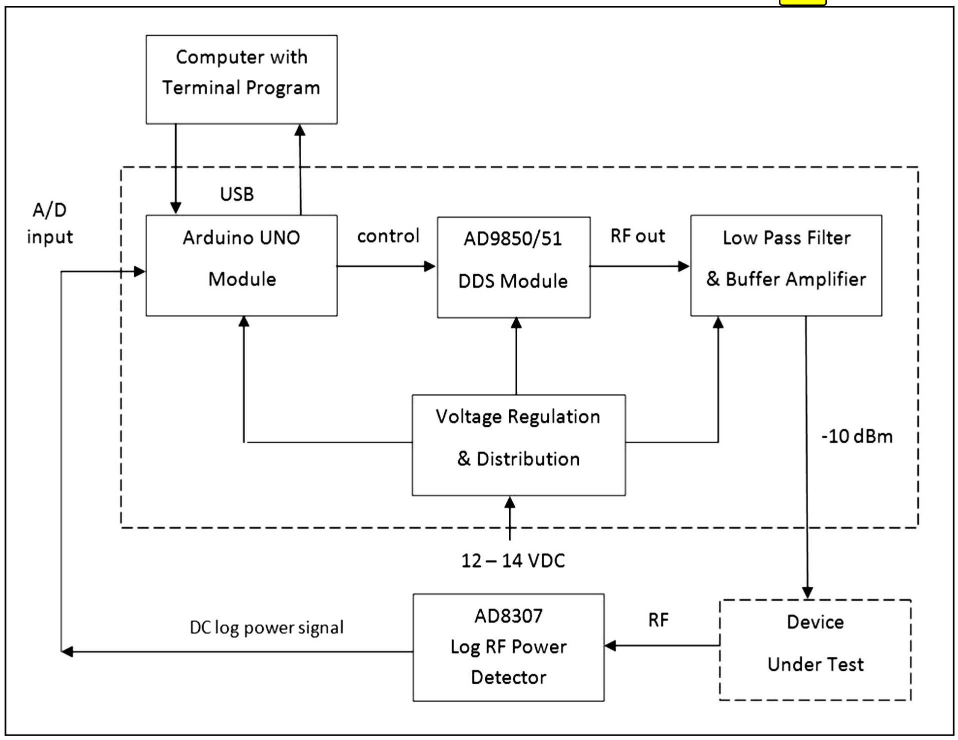

Figure 1—PHSNA block diagram.

What's in a Name?

In the original announcement post, the project was jokingly called SSNAFHU, for Simple Scalar Network Analyzer for Hams —Unveiled. We've standardized on PHSNA, Poor Ham's Scalar Network Analyzer, emphasizing the low cost of construction and because it's the name of the Yahoo Group. Lately I also hear SSNA for Simple Scalar Network Analyzer.

But the key term is "scalar," suggesting a modest reference to the fact that this isn't a Vector Network Analyzer (VNA). It gives the magnitude of gain or loss through a network, but not the phase shift. But I think we're seeing that magnitude only measurements can be quite useful.

What is it? What does it do?

The system is built around two off-the-shelf modules. Refer to Figure 1 for this discussion. First is an AD9850 DDS board which includes the clock module and other components necessary to make a complete DDS RF generator, excluding controller. Next is an Arduino board which will provide user interface communications, control of the DDS, and necessary ADC conversions on the signal from the log power meter.

The heart of the project is a small (2.75" by 4") motherboard onto which both modules are mounted. The board has 0.1" female headers on both faces to accommodate matching pins on the DDS and Arduino modules, putting it in the middle of a three board stack. The board makes the required connections between the two modules, plus provides voltage regulation, post-DDS filtering, an amplifier stage, and attenuators to control level and keep impedances constant. The DDS and Arduino modules and their motherboard reside within the dashed rectangle of Figure 1. Figures 2 and 3 show the DDS and Arduino sides of the board, respectively.

Thus far, we have a functional and useful RF signal source. To turn this into an analyzer we need an RF detector. The project uses a version of the circuit published in 2001 in *QST* by Wes Hayward and Bob Larkin [Ref. 1], with several variations published in succeeding years. It uses the amazing AD8307 chip which can measure RF signal strength from about –80 dBm to +15 dBm. Originally, the project was going to let participants be on their own for this component, but then Jim found that he could squeeze the circuit onto the standard production board size when he did the layouts, and got the OK from Mr. Hayward to publish his circuit on the PHSNA Yahoo site. Because of its extreme sensitivity, the log power measurement circuit needs to be in a separate enclosure and its board is separated from the main board. Users can include an analog meter to make it a stand-alone instrument as in the original article, or omit the meter and just let the Arduino convert the voltage output to a dBm reading.

Which brings us to the "what does it do?" part. Again turning to Figure 1, we see the system in action, sourcing a signal through a Device Under Test (DUT) (crystal, filter, amplifier, attenuator, etc.) and then into the log power detector, which sends a DC voltage output proportional to the log of RF power back to the Arduino. The Arduino converts that signal to a number representing RF power in dBm and reports it back to the user via serial / USB port. By sweeping the DDS over a desired range of frequencies, we can create an array of data showing the response of the DUT and display it graphically as seen in the lovely plots we've included in the article.

As hams, we know the value of seeing the response curves of our filters: lowpass, highpass, bandpass, IF, and other networks. Also, we can characterize individu-

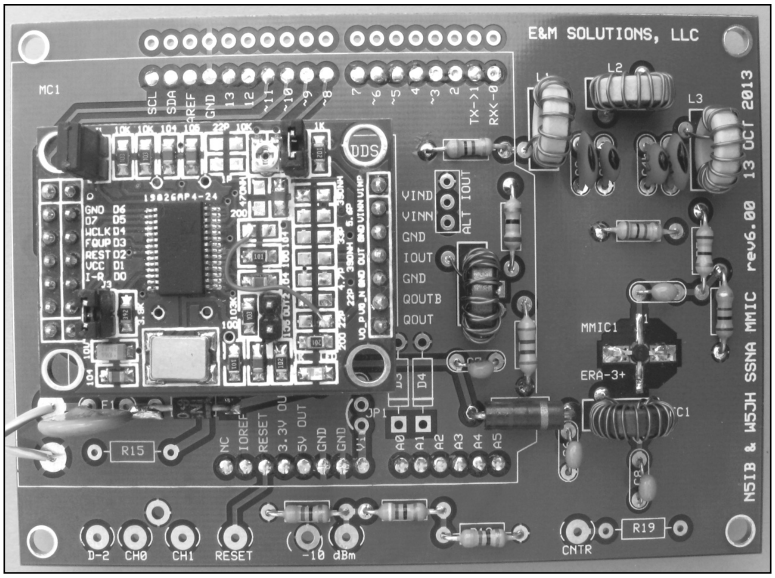

Figure 2—Main board with DDS, filter and post amplifier.

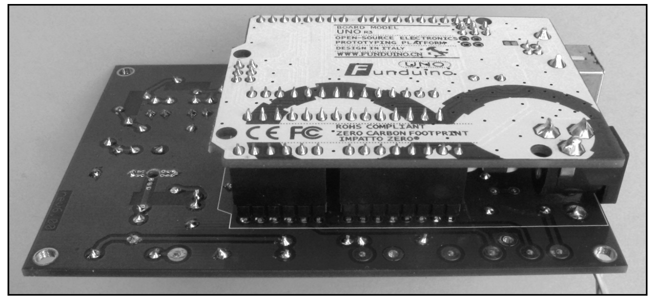

Figure 3—Main board from the Arduino side.

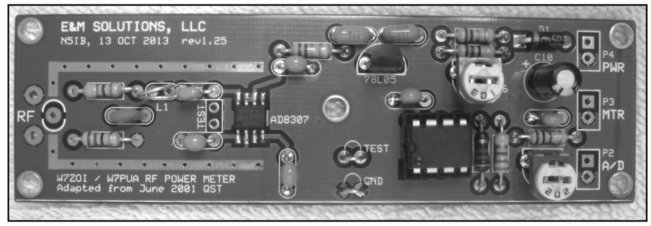

Figure 4—Log power meter board.

al crystals and use the parameters we've measured to design and build our own crystal filters. And we can see how "broadband" our broadband amplifiers actually are. Add a simple outboard bridge and we can check the SWR of our antennas.

Board Details

The power supply section requires 11 to 13 VDC input and uses an LM7805 regulator to provide +5 VDC for the Arduino and DDS. The full supply voltage is required for the post-DDS amplifier. Also provided is an adjustable LM317L regulator circuit to allow setting the ADC reference voltage used by the Arduino to the desired value.

The DDS module includes a filter, but it is made from tiny SMT components of questionable Q value, so a higher quality replacement filter using toroidal inductors is included in the board layout. It's flat out to 33 MHz. (That's for the AD9850 version. The AD9851 version will allow going beyond 6 meters.) There are resistive attenuators before and after the filter and after the amplifier to maintain 50 ohm terminations for the filter and assure a 50 ohm generator output resistance. These also set the target output level of approximately –10 dBm. The basis for this choice is to allow driving crystals at a proper level.

What about that amplifier? This is an evolving project and two versions were designed with separate board runs made for each. The first was based on EMRFD Figure 2.57 and used a 2N5109 transistor providing about 16 dB of gain. Another version uses a MMIC amplifier. This might be a good choice for anyone choosing to substitute an AD9851 module for the AD9850 originally proposed, taking the maximum frequency from about 30 MHz to about 60 MHz. I built the 2N5109 version and adjusted the amplifier and attenuator chain to get about +8.5 dBm output.

When putting it all into a box, these connectors and controls are needed: An RF output jack, typically BNC. A log power signal input jack. A +12 VDC power jack. For serial / USB communication and programming the Arduino, a cutout is needed to allow access to the Arduino's USB jack. My Arduino came supplied with a USB cable. A USB to serial driver makes it look like a serial port within my PC. In addition, a reset pushbutton for the Arduino is a good idea so you don't have to interrupt power to re-start the program.

Not Too Complicated So Far, eh?

The design for the log power board follows the Hayward / Larkin version (Ref. 1) fairly closely. It contains a battery, 5V regulator, the AD8307 chip, and a following op-amp with adjustable gain, plus an input compensation circuit giving it a wide bandwidth of over 500 MHz. Jim made boards for both DIP and SOIC versions of the AD8307. A number of suitable op-amps have also been proposed in lieu of the original LM358, so use your junk box. A unique feature of the board is inclusion of ground foils placed to allow building a copper "Quonset hut" shield over the AD8307 chip. This is optional and since my enclosure for this board is very RF tight, I did not include it in my build.

Calibration pots will allow scaling the maximum output of the log power board to match the ADC reference voltage set in the PHSNA board.

Controlling it—The User Interface

No display, no keyboard and almost no switches. The simplicity of this design comes in part from having the user interface made via a serial terminal. It seems almost archaic, but even in this modern age we can still find serial terminal programs to run on our PCs and talk to the Arduino via its virtual serial port. The Arduino spools out a menu when booted and waits for the user's input. I chose term232.exe as my terminal program. It's free, small and simple and runs fine on my laptop under Windows Vista. When working in the Arduino IDE environment, one can also just pull up its serial monitor and talk to the PHSNA from there. This project is a hacker's delight though. One user has already added an LCD to his PHSNA box showing current frequency and dBm reading.

The Software

Here's another place where hackers can have some fun. The PHSNA software is completely open source, so if you like to program, dive in. The software was rolled out with two functions. The SNA function scans a user specified range and step size and outputs to the serial terminal the current frequency and the level in dBm read back for each frequency. This information is transmitted in the "comma delimited" format that can be picked up by Excel or other programs for graphing. Just do a "capture" with the terminal program and you've got your data file.

The second function is the Generator function. The user specifies a frequency and the DDS goes there. A dBm reading is also displayed. Arrow keys and adjustable step size allow moving the DDS around and seeing the change in response, if the AD8307 board is connected.

I jumped on the software bandwagon and added several functions. Some of them were developed mindful of users who don't want to alter program code, or even "burn" an existing program to the Arduino. I added the ability to calibrate the DDS's clock to a standard (such as WWV) and to determine the AD8307's slope and intercept constants used to convert the ADC reading to dBm. Also the ability to set and/or change the start-up frequency of the DDS was added. All this information is saved to EEPROM and read at each start-up, so there's no need to "hard code" it into the source code and burn a custom version.

I initially did the slope and intercept using two points of data, but Jim pointed me to a statistics site that gave a method for calculating a best fit slope and intercept from a set of multiple data points, so I added that capability to the routine. SWR measurements using the return loss bridge became a separate menu item.

I also added ten memories accessed with a keystroke (0 through 9) to act similar to band switches. And finally a "Scan" function simply sweeps the DDS over a specified range and repeats indefinitely.

Normalization and Frequency Compensation

Often when plotting the response of filters, you don't care about the actual dBm numbers but instead want zero attenuation to be graphed as a 0 dB reference and everything goes down from that. This is where normalization comes in. The software can automatically normalize readings so the reading with no losses will be 0 dBm.

In addition, there will usually be a slight falling off in RF level from the PHSNA as frequency increases. The user just wants to see amplitude changes caused by his DUT, not by the RF generator. The software also can address this, using a 5th order polynomial equation to flatten the reported response. This hasn't been automated (yet) but Jim has placed a detailed description on how to get Excel to compute the coefficients from a "self scan" of the PHSNA in a PDF on the Yahoo site. I added a function to allow entering these constants from within the program which then saves them to EEPROM so no editing of source code is required.

For those who do want to edit source code though, note that the IDE is a free download and no external programming hardware is required to program the Arduino—just plug in the USB cable and go. Here's what's on the start-up screen:

Current reference clock frequency is:
  179987712
Mode flag: Normal Mode
Retrieved:
slope = 0.10110 intercept = -84.62
DDS is running on: 7040000 Hz
  AD9851 mode
PHSNA Standard rev 3.01

Select:
  0 - Restore this menu screen
  1 - SNA Functions
  2 - Generator Functions
  3 - Return Loss Bridge & SWR
  4 - Measure Crystal Parameters
  5 - Memories
  6 - Scan
  * - Options & Calibrations Menu

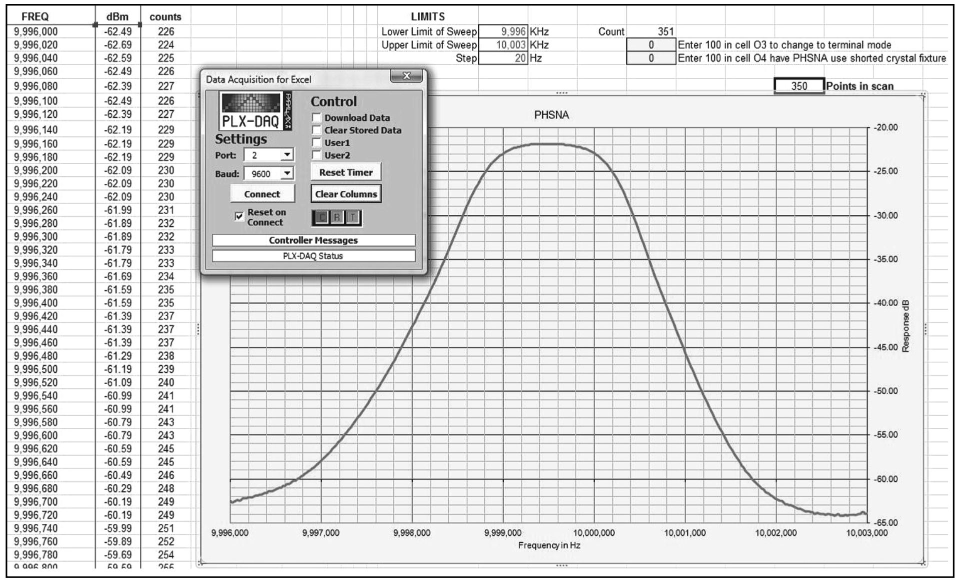

Figure 5—Spreadsheet running PLX-DAQ link to PHSNA.

And here's the separate menu for setup and calibrations:

Choice:
  A - Calculate or enter AD8307 slope / intercept
  C - Curve fit polynomical constants: View/Enter/Clear
  D - Calibrate DDS reference frequency
  P - Change start-up mode to PLX (Excel) or terminal
  S - Set Start-Up Frequency
  M - Toggle AD9850/AD9851 modes
  Q - About PHSNA
  $ - Re-display menu
  ^ - Back to main menu

PLX-DAQ Interface to Excel

Option 6 requires some explanation. Jack Generaux, WØFNQ found an Excel macro widget called PLX DAQ that's freeware from Parallax, maker of the Basic Stamp chips. This add-in gives Excel the ability to communicate via serial port, importing data from external hardware. Jack customized the PHSNA's Arduino software so it would ask for upper and lower limits and step size from specific Excel cells, then proceed to scan that range and send the frequency and dBm reading for each point to back to Excel where it could be plotted. This avoids the need for a terminal program entirely. Figure 5 shows PLX-DAQ in action. The data for the filter plot was imported directly into Excel from PHSNA using the PLX-DAQ interface.

I'm highly impressed with PLX-DAQ

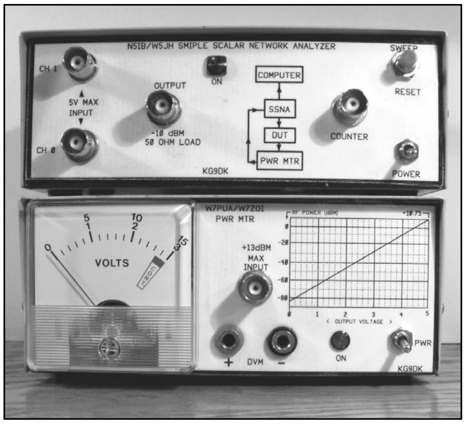

Figure 6—PHSNA and power meter by KG9DK.

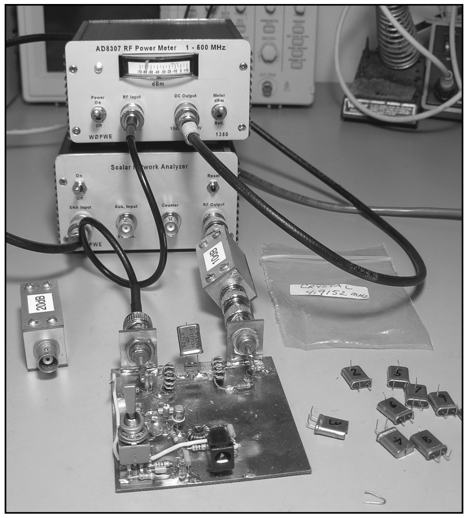

Figure 7—WØWPE's PHSNA and meter at work testing crystals.

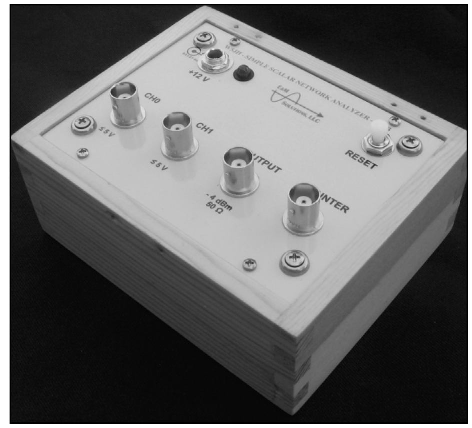

Figure 8—N5IB's PHSNA in a custom wood box.

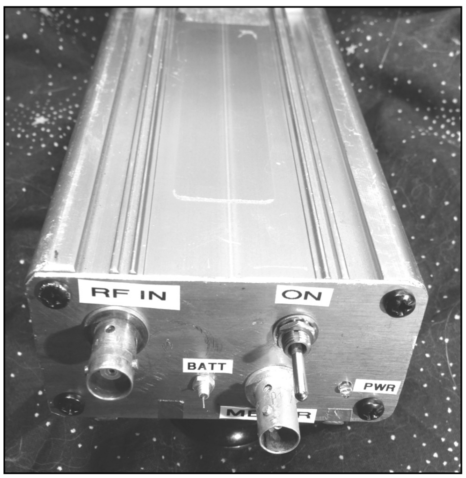

Figure 9—WA5BDU's log power circuit in box.

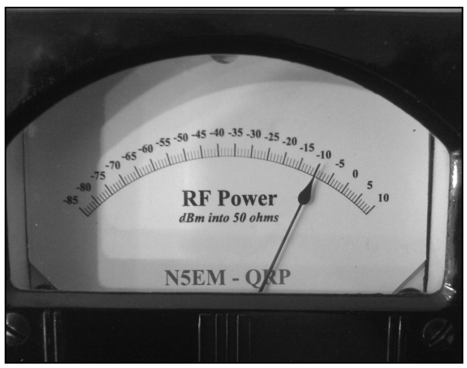

Figure 10—N5EM's classic meter movement with a custom scale.

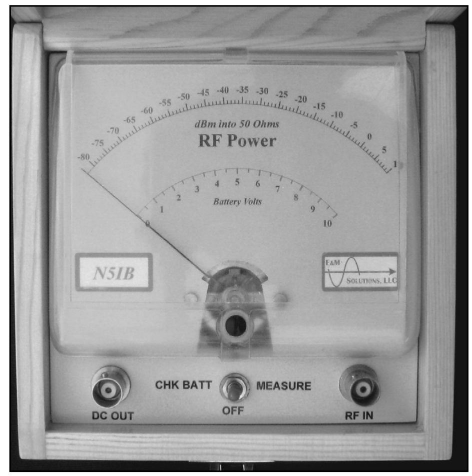

Figure 11—N5IB's meter with custom scale and box.

and decided to integrate it into the mainstream program so instead of having an "either/or" firmware configuration, the user could toggle between "terminal" and "PLX DAQ" modes as desired. You can manually tell it to switch modes, but it can also detect how it's connected and swap automatically.

Crystal Scanning Function

Finding the 3 dB bandwidth of crystals and their series resistance is required to determine crystal parameters Lm, Cm and Rs for use in filter design. One can easily go through tabulated or plotted data and find the bandwidth, and the Rs value can be found by resistor (potentiometer) substitution. This was all outlined by Jim Kortge K8IQY in a couple of influential AmQRP articles (Ref. 4) and by Hayward and Demaw before that. Then you crank through the math with your calculator or spreadsheet to determine the parameters.

I thought it would be a natural application for software to automate this process and wrote routines that automatically find the resonant peak and loss at that point, as well as finding the upper and lower half power points. With this information, the Arduino can (and does) crunch the numbers to get a crystal's model parameters (including Rs) using standard formulas, and do it in a jiffy. Now you can take a stack of crystals and find their motional parameters in a small fraction of the time it took previously. You could easily do one crystal per minute once you get rolling.

How Much Does it Cost?

We'd all like a workbench full of sophisticated test equipment, but price is important. I can't tally up what a complete system would cost since it varies so much. But I can say approximately what I paid for the major items. I made good use of the eBay / China supply chain when possible. These prices are postpaid and are what I paid. Others paid more or less.

Arduino UNO	$12.30
DDS AD9850 (30 MHz)	$6.88
PHSNA board set	$6.50
(one for the DDS/Arduino and onefor AD8307 power meter)	

The board set is not currently available as the last run is sold out. However, if evidence of sufficient interest is shown, the organizers of this project, N5IB and W5JH, have agreed to run another board set in the near future. If you are interested in the project, the best approach is to join the PHSNA group on Yahoo and tell the organizers of your interest. Likely cost for a board set is about $15 plus shipping.

Of course you have to populate the boards with the usual resistors, capacitors, regulators, connectors and toroids, and provide enclosures. I have a pretty deep

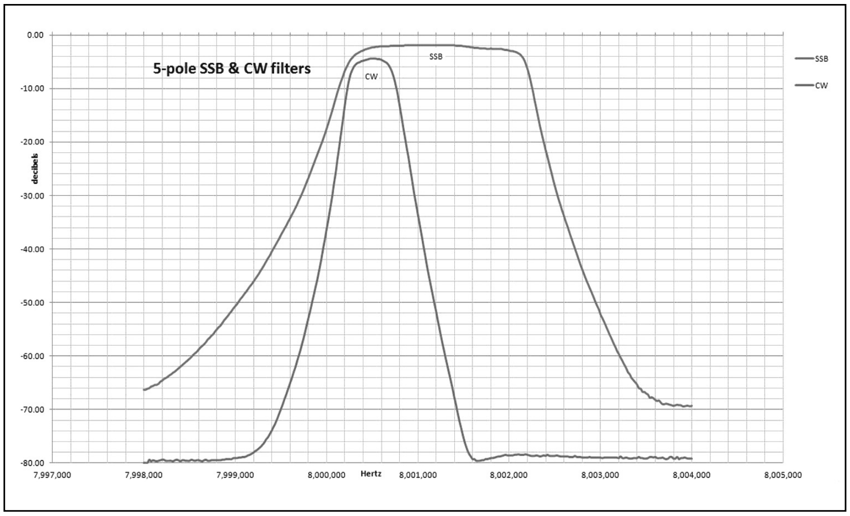

Figure 12—Passband plots for 8 MHz SSB and CW filters.

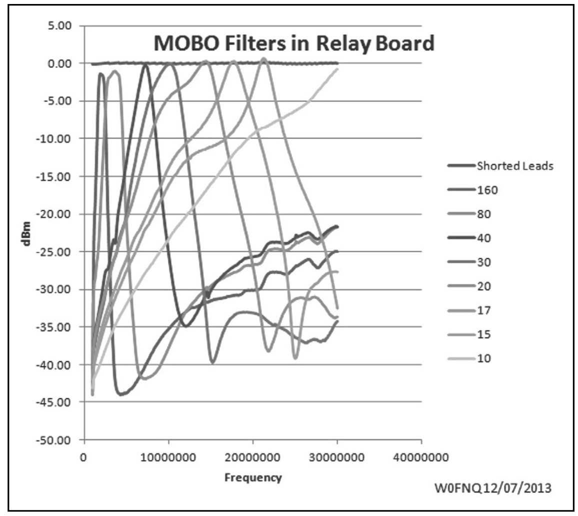

Figure 13—Multiple bandpass filter plots.

junk box but others might have a pretty good shopping list. The AD8307 chip could set you back $12 or so, or maybe you could get a free sample. I went to eBay and bought ten (10) of them in the SOIC-8 package for $9.50 postpaid, less than a dollar apiece.

On the PHSNA site is a spreadsheet BOM with info on where parts can be purchased. It includes photos of the Arduino and DDS boards, since you want to buy the correct form factor. An eBay search for "AD9850 module" or similar is suggested. Suggested vendors on eBay include "chip_partner," "chipworld," and "survy2014". Compare the hits you get with the photo in Figure 2.

Builds and Boxes

As you see from the photos, there are some real craftsmen in the group. A PHSNA in an ugly "repurposed" box (like mine) can work well too, but we'll focus on the pretty ones here. Figures 6, 7 and 8 show implementations by KG9DK, WØPWE and N5IB. I show my own log power circuit enclosure in Figure 9 as an example of one without a meter movement. If you do use a meter, it's hard to beat the beautiful old Weston movement with custom scale by N5EM shown in Figure 10. He used the Galva application to create the scale. Likewise, look at N5IB's beautiful meter and custom enclosure in Figure 11.

Again, two boxes are required. The one for the DDS / Arduino "sandwich" doesn't require much shielding but the one for the AD8307 log power measurement circuit should be in an RF tight box.

Getting Down to Business…

This is the fun part—actual measurements. To measure filters and other networks just put the DUT in circuit per Figure 1 and run a plot. You can save plotted data to a CSV file for plotting by Excel or other plotting programs, or import directly into Excel with PLX-DAQ.

In Figure 12 you see that I plotted the response of CW and SSB bandwidth filters on the same graph. This view will help determine the best placement of the BFO injection for each mode.

Figure 13 shows plots of eight separate bandpass filters in a WØFNQ project.

The PHSNA system is great for plotting the SWR of antennas when used with a return loss bridge. Figure 14 shows a plot I did of my tribander on 20 meters. The return loss bridge (RLB) is a simple device, discussed later.

You needn't limit yourself to checking passive devices. I used the PHSNA to plot the gain of a homebrew 35 dB gain broadband utility amplifier over the range of 1 MHz to 60 MHz. Just put a big attenuator at the input end so as not to exceed the AD8307's measurement range or drive the amplifier into saturation.

When doing crystal characterizations, the numerical results tell more than plots. Table 1 shows tabular data captured from PHSNA in testing a set of several crystals. This data is all you need to design your own crystal filters using any of a number of popular programs available free.

Accessorize!

You can do a lot with just the PHSNA and log power meter, but there are some other gadgets that will extend its usefulness.

Power Level Calibration

For most measurements, the relative levels of two powers are all that matters. That's what decibels are all about, after all. So we mostly need an accurate slope measurement on the log power measurement circuit. But some of us want to know the absolute power level being measured

Press 'Y' to test another w/same parameters,
Press other key to return to menu
Assign/change crystal ID (Y/N)?

ID#, F_peak, loss(dB), BW, Rs, Cm (pF), Lm (mH), Q
JH01, 4432251, 4.25, 74.0, 15.8, 0.0147079, 87.667694, 154899
JH02, 4432271, 5.06, 79.0, 19.7, 0.0143055, 90.133056, 127162
JH6, 4432243, 9.91, 136.0, 53.2, 0.0140858, 91.540275, 47899
JH9, 4432246, 4.55, 75.0, 17.2, 0.0143952, 89.572540, 144944
JH10, 4432308, 3.34, 66.0, 11.7, 0.0145663, 88.518142, 210561

Table 1—Crystal test results in tabular form.

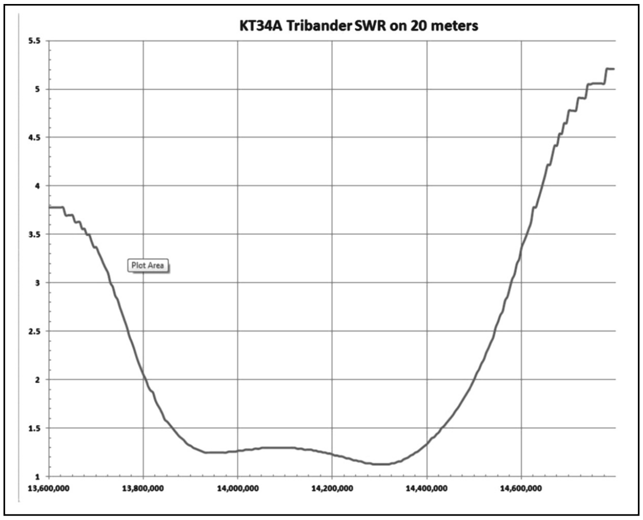

Figure 14—SWR plot of a KT-34A tribander for the 20 Meter band.

and/or being sourced by the PHSNA generator. You can do adequate calibrations with things like compensated diode detectors or oscilloscopes to establish a reference power level. Or you might use a 1 watt QRP transmitter and run it through the 40 dB tap described in the Hayward / Larkin article log power meter article [Ref. 1].

But some of us wanted to get fancy and tried a simple but accurate calibration source initially described by Bob Kopski K3NHI in the Jan/Feb 2004 issue of *QEX*. It uses a common computer clock module as a source. The calibrator and an extension of it are described in VE7BPO's popular QRP Homebuilder website (Ref. 2). Speaking of accessories, a step attenuator is nice to have for calibration and slope determination. Separate chainable attenuators work well too. I built the Kopski standard and the VE7BPO sine wave source that calibrates to it and now feel like I have a pretty accurate indication of true power level.

Crystal (and other) Measurement Jigs

You can just stick a crystal in between the PHSNA's RF out and the AD8307's input and measure it in a 50 ohm environment, but you're going to want some kind of socket or grabbers. So while you're at it, just make a dual 4:1 transformer arrangement to put the crystal in a 12.5 ohm environment for better accuracy. This was described in the K8IQY series [Ref. 4]. In Figure 7 you see WØWPE's set-up using the transformers of his K8IQY circuit for crystal measurements with PHSNA.

Return Loss Bridge

A return loss bridge (RLB) is a simple device consisting of three 49.9 ohm 1% resistors and one ferrite transformer plus three BNC jacks. But it expands greatly what you can do with the PHSNA, notably read the SWR for your antennas. But, a good 50 ohm match can also be important in other circuits such as filters and amplifier inputs. Jim and Jerry came up with a beauty and put it out as a kit including a Hammond box and a circuit board with holes that align perfectly with the BNC pins—no leads. Well worth having. RLBs are easy enough to build on your own if you want to go that route.

Mutations, Bifurcations and Tangential Developments

With a popular "open source" project, it's hard to keep up with developments. Hardly a day goes by that I don't see some new customized features added by a user. At least one person has added a text display with I^2C interface and added coding to the Arduino program to send it frequency and dBm information. Just yesterday (as I write in February) George N2APB and Dave AD7JT showed preliminary photos of a hand held terminal for the SSNA that accepts a keyboard connection and has a color graphics display. No computer required! A run of kits is expected.

Jerry, W5JH is working on a "measurement receiver" to scan and quantify harmonic levels from an RF source.

I'm personally working steadily on a stand-alone Windows application to interface directly with the PHSNA using Visual C#. The idea is to have the PC do all the hard work and just have the Arduino accept frequency information and return ADC (log power) information. Ideally, the program will also have graphing routines so no outside graphing program will be needed.

Where's the Schematic?

Really, I'm not holding anything back. But the project has a number of variations in configuration. Two or three different DDS modules, plus some are using NJQRP / AmQRP DDS-30 and DDS-60 boards. Going to 60 MHz changes the filter component values. There is also a version with a 2N5109 post amplifier and another with an MMIC amplifier. You get the idea. But the schematic diagrams and other technical information are presented in depth on the PHSNA Yahoo website. If you are interested in giving it a shot, sign up and take a look. Jim has also produced a number of illustrated guides in PDF format to help the user build and use the various components. You'll also find a lot more photos, graphs and other information from group participants.

Acknowledgements

We thank Terry WB4JFI, Bill W2EB, Leon G1HSM and Tracy N4LGH for providing some excellent information on working with the inexpensive DDS modules being sold by offshore electronics vendors. This allowed Jim and Jerry to move forward and make the PHSNA project a reality.

Conclusions

I believe the PHSNA is a project that results in an extremely useful and versatile RF measurement system at low cost and only moderate building complexity. I hope readers will see it the same way. We'd love to have more folks jump in and give this project a try.

References

1. "Simple RF-Power Measurement," by Wes Hayward, W7ZOI and Bob Larkin, W7PUA, *QST* June 2001.
2. QRP SWL Homebuilder: http://www.qrp.pops.net/RF-workbench-5.asp
3. PHSNA discussion group: https://groups.yahoo.com/neo/groups/PHSNA/info
4. "Tools and Methods for Measuring Crystal Parameters," by Jim Kortge, K8IQY, NJQRP *Homebrewer* Issue #9