PURESIGNAL for HermesLITE2

(feedback from the TRX relay crosstalk)

Christoph van Wüllen, DL1YCF

August 03, 2020

Table of contents

1. The AD9866 chip

The HermesLite2 is built around an AD9866 single-chip modem and a small Cyclone IV (25k logic elements) FPGA. See the block diagram of the AD9866 from Analog's data sheet:

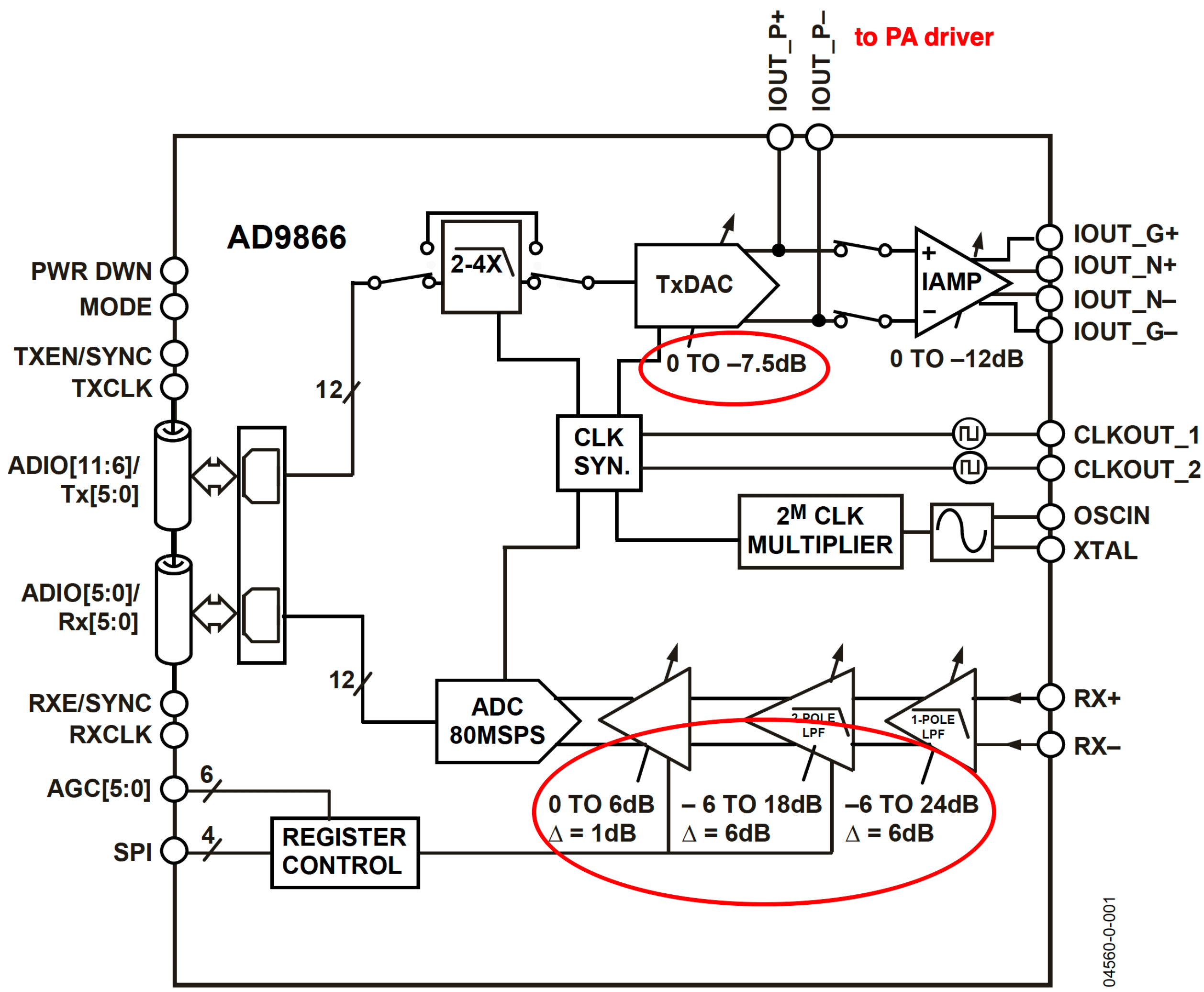

Fig. 1. AD9866 block diagram (from the AD9866 data sheet), comments added in red

In the RX path, there are three programmable amplifiers/attenuators, such that there is a user-programmable amplification of –12 … +48 dB. All "preamp" settings mentioned below refer to that scale. In the piHPSDR software used to perform these tests, there is a "RX gain" slider with values from –12 to +48 that directly translates into these "preamp" settings. Note that "conventional" HPSDR boards have a fixed 20 dB preamp and a programmable attenuator (0–31 dB) such that the "real" input amplification there goes from –11 to +20 dB.

Setting the preamp to 0 dB in the HL2 and feeding a –20 dBm input signal does not produce a RX IQ sample amplitude of 0.1, but this is what the software (piHPSDR, PowerSDR) signal strength measurement is based on. Therefore, piHPSDR uses an internal variable "rx_gain_calibration" for correction. It is the preamp setting at which a –20 dB input signal produces a RX IQ amplitude of 0.1. The standard value in piHPSDR is +14 but this can be changed in the "Radio" menu.

The PA driver is connected to the IOUT P+/- outputs of the AD9866 chip. This output is digitally programmable from 0 to –7.5 dB in steps of 0.5 dB, this explains why only the upper 4 bits of the “TX drive” HPSDR word are used in the HL2, and why the output level does not go down to zero (only down to about 1 Watt) if the “Drive” slider is moved to zero.

2. Prepearing for the measurements

My standard setup for performing measurements on a transmitter is a 500W dummy load which I have equipped with a –60dB attenuator. This is too much for the HL2 since I want to feed about 0dBm to the SpecAnalyzer. Therefore I built a small dummy load that can safely withstand the 5W HL2 TX output and provides a port for measurements which features ~0dBm when the HL2 is transmitting at 5W:

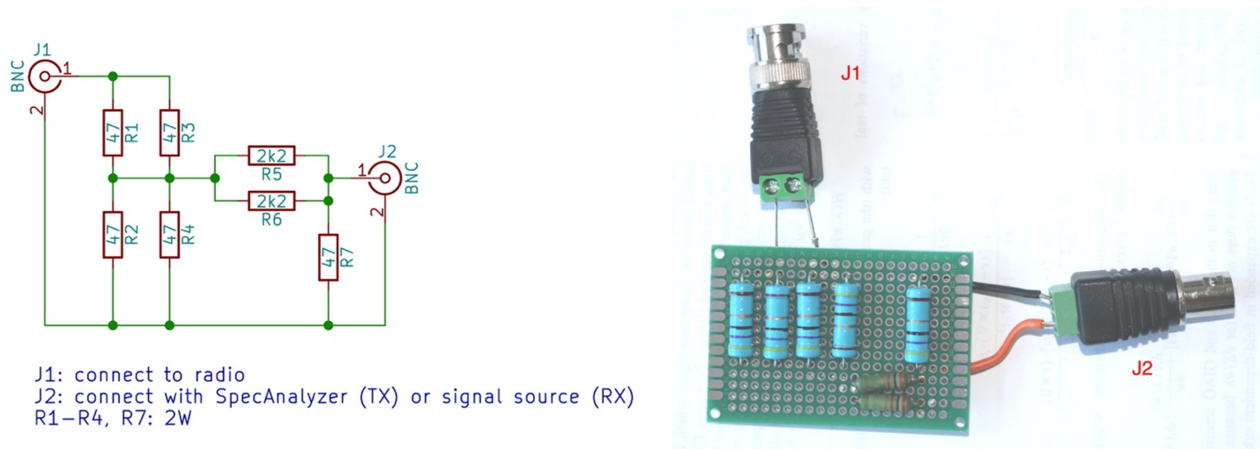

Fig. 2. A simple 5W dummy load with an attenuated output

R1–R4 and R7 are 2W metal film types, while R5/R6 came from the junk box. Using a SpecAnalyzer with a tracking generator, it was confirmed that the attenuation is "flat" at about –38dB from 1 to 30MHz. Connecting R5/R6 with the common ends of R1–R4 is somewhat unconventional, but results in a nice mechanical layout, as seen by the photo. Feeding a signal with ~1V_{pp} (as available from standard signal sources) to J2 will also result in a signal at J1 (about –30 dBm) that will not harm the HL2.

3. First connection

A Macintosh computer running piHPSDR and the HL2 were connected with a standard ethernet cable connecting the RJ45 plugs of the Mac and of the HL2. No DHCP server was running on the Mac.

While piHPSDR recognized the HL2, it could not start it. The "automatic" (APIPA) internet address of the HL2 (169.254.xx.yy) was outside the netmask of the Macintosh ethernet adapter, and piHPSDR therefore complained that the HL2's IP address is on the wrong subnet.

This I fixed in my version of piHPSDR: the "subnet" check is simply suppressed whenever the IP address is of the form 169.254.xx.yy. Note xx and yy are the two least significant bytes from the HL2's hardware address. This change is not necessary on all systems.

After that, piHPSDR could start the radio.

4. RF front end, RX gain calibration, ADC clipping

As a signal generator I used my "HAMlab", a RedPitaya-based versatile measurement device. The signal generator was set to 14.1 MHz. The signal was fed to the "J2" connector of the attenuator (Sec. 2) and "J1" was connected to the radio. To calibrate things, both an HL2 and my ANAN-7000 was used. In piHPSDR, first the standard HL2 value for the "RX calibration" (Radio menu) of 14 was used, and the readings were done with the preamp set to maximum attenuation (–12 dB) at the HL2. The measurements were done with and without the N2ADR filters engaged. The results are displayed in Table 1.

Signal V_{pp}	Anan-7000 reading	HL2 reading with N2ADR	HL2 reading without N2ADR
1V	–34 dBm	–31 dBm	–30 dBm
0.71 V	–37 dBm	–34 dBm	–33 dBm
0.5 V	–40 dBm	–37 dBm	–36 dBm
0.1 V	–54 dBm	–51 dBm	–50 dBm
0.01 V	–74 dBm	–71 dBm	–70 dBm

Table 1. Signal strength reading for various signal generator amplitudes

Note that 1 Vpp means 0.354 V_{eff} and corresponds to 2.5 mW (+4 dBm) at a 50-Ohm load, so the ANAN values nicely agree with the nominal attenuation (38 dB) of our simple circuit and can be taken as a reference here. The HL2 readings have a constant offset (+ 3 dB) and this can easily be corrected by changing the default value of the RX calibration from 14 to 11. Note that this setting is valid for 20m reception with the N2ADR low-pass filters engaged. Disabling them during RX gives 1 dB more sensitivity and then the correct RX calibration value would be 10.

The RX calibration value in piHPSDRs Radio menu has been set to 11 in the further measurements.

To investigate the onset of ADC clipping, the RX gain slider was slowly moved from –12 dB upwards (1 V_{pp} signal source). During this move, the input signal as displayed by HL2 stays at –34 dBm for a while (the IQ amplitude grows, but this is compensated internally by piHPSDR) until the RX gain slider has reached a gain of +33 dB, moving the slider further lets the displayed signal strength go down. This is so because the maximum IQ amplitude has been reached, further increase of the gain does not lead to larger IQ amplitudes but is taken into account internally. In a second experiment, this has been nailed down more accurately: while still feeding a signal at 14.1 MHz, the radio was set to 28.2 MHz where a weak signal is detected (–74 dBm, the first overtone of our signal source). The panadapter does not change when moving the RXgain slider from maximum attenuation (–12 dB) upwards until it reaches +33 dB, moving further up to +34 dB then shows a lot of artifacts in the spectrum.

This means that with a –34 dBm input signal (–35 dBm at the HL2 board input), and a nominal amplification of +33 dB of the built-in preamp, the ADC starts to clip. So with zero nominal amplification (preamp at 0 dB) it should be possible to go up to 0 dBm, but I do not recommend this experiment since 0 dBm is a rather strong signal to feed to delicate electronics.

Frequency [MHz]	HL2 w/o LPF, w/o HPF	HL2 only HPF	N2ADR standard settings
1.850	–52 dBm	–71 dBm	–53 dBm
3.600	–53 dBm	–53 dBm	–54 dBm
5.360	–54 dBm	–54 dBm	–55 dBm
7.100	–53 dBm	–53 dBm	–54 dBm
10.100	–53 dBm	–53 dBm	–54 dBm
14.100	–53 dBm	–53 dBm	–52 dBm
18.100	–52 dBm	–52 dBm	–51 dBm
21.100	–50 dBm	–50 dBm	–51 dBm
24.900	–48 dBm	–48 dBm	–49 dBm
29.000	–50 dBm	– 50 dBm	–51 dBm

Table 2. Insertion loss of N2ADR high- and low-pass filters

Finally, the insertion loss of the N2ADR board was measured on all bands (Table 2). The high-pass filter (OC bit 7) gives no detectable insertion loss except on the 160m band, where the insertion loss is large. Note that the high-pass is not used on the 160m band by default. For all bands the insertion loss of the low-pass filters is rather uniformely ~1 dB. Please ignore that the reading increase when going to higher frequencies. This is most likely caused by stray capacitances on the simple attenuator board (Fig. 2). On the ANAN-7000, the reading at 29.2 MHz is –50 dB and agrees within 1 dB with the HL2 reading. The "hard fact" here is the 1 dB loss caused by the low-pass filters.

A single experiment was done on the 20m band, tuning with the output set to 5 Watts without the TX low-pass filter and then inserting it via the "OC" menu of piHPSDR. When activatint the output low-pass, the output power reduced to 4 W. It thus seems that assuming a 1 dB insertion loss while TXing one is in the right ball-park.

5. Isolation/crosstalk in the RX/TX relay

Note: This experiment has been done with a temporarily modified version of piHPSDR where the preamp setting is not changed when switching to TX. Normally, the preamp setting during TX is derived from the "TX att" value (see below). The values reported here have been obtained with a –12dB preamp setting but the same values result with, say, the preamp set to 0dB.

Band	PA calib. value	RX level
160m	42.0 dB	–38 dBm
80m	42.0 dB	–33 dBm
40m	41.5 dB	–26 dBm
20m	42.0 dB	–20 dBm
10m	41.0 dB	–16 dBm

Table 3: RF isolation (crosstalk) in the TRX relay

The HL2 was connected to a dummy load. piHPSDR was set to "duplex" mode, that is, one continues receiving while transmitting. Then a single-tone output at full nominal output power was emitted. The "PA calibration" values were set such that about 5 Watt was emitted (a little bit less in the 10m band). Table 3 shows the PA calibration values, and the reported strength of the RX input signal, for various bands.

The PA calibration values are remarkably constant, but it is not possible to adjust the output level very accurately, since the AD9866 chip regulates the TX output in steps of 0.5 dB (therefore it also makes no sense to change the PA calibration value in smaller steps). So at best you can reach 5W ± 0.5 dB, which ranges from 4.5 to 5.6 Watt. I do not bother about this, since I am not sure whether I can actually *measure* 5W RF with higher accuracy.

The large differences in the RX level indicate that the cross-talk in the RX/TX relay substantially increases with the frequency (this is not unexpected). Given that we put about +37 dBm to the dummy load, the isolation in the RTX relay is ~75 dB on the 160m band and drops to 53 dB for the highest frequencies (10m band). This cross-talk can be used to produce a feedback signal for adaptive pre-distortion without any external hardware involved, the challenge is however the wide range of feedback levels for different bands.

6. "TX DAC feedback" level

This measurement is done in software. When a TX IQ signal with unit (~1.0) amplitude is emitted, the TX DAC feedback signal has a smaller amplitude. This is normally 0.406 for P1 HPSDR hardware and 0.2899 for P2 HPSDR hardware. The HL2 firmware is different and produces a level of about 0.235. This can be "measured" by temporarily adding some print statement in the piHPSDR program logging the maximum IQ amplitude (sqrt(i*i+q*q)) in each batch of TX and TX feedback IQ sample batches. The "measured" number is entirely determined by the HL firmware. At the same time, we can "measure" (that is, add a debug statement that prints) the IQ level in the RX feedback channel. While the TX-IQ and the TX-DAC-feedback levels are entirely software determined (and do not change when the drive slider is moved), the RX-feedback signal strength depends on the amount of attenuation and the RF level. Therefore we here report the IQ level of the RX feedback signal that is "optimal" for the PURESIGNAL engine.

The experiment was performed on the 20m band. A two-tone signal with about 5W PEP was sent. The preamp was (automatically) adjusted for optimum RX feedback level ("feedbk" value in the PS menu was 153). This corresponds to the following maximum IQ amplitudes:

TX IQ signal	0.982
TX DAX signal	0.234
RX feedback signal	0.141

The TX IQ signal is determined by the piHPSDR software, which uses two tones each with amplitude 0.49 (to prevent overflows). The TX DAC signal amplitude is determined by the HL2 firmware, this level is also measured by the PURESIGNAL engine and reported in the PS menu in the GetPk ("get peak") field. The optimal RX feedback signal then has an amplitude of about 0.141, which corresponds to about –17 dBm input signal strength when the RXgain slider is at the position of the RX-gain calibration value.

For a RX calibration value of 11 (Sec. 4) and a real (physical) RF input level of –38 (160m band) to –16 dBm (10m band) from the TRX relay cross-talk (Sec. 5) one can thus deduce that the optimum preamp setting for PURESIGNAL feedback from the TRX relay crosstalk should be between +32 dB (160m band) and +10 dB (10m band). This range is not fully covered by my original PURESIGNAL support in piHPSDR (made without any access to the hardware), which allows preamp ranges from –12 dB to +19 dB. This will be demonstrated in the next section and is easy to fix, since one can easily "move" a window 31 dB wide into the correct position.

7. Adaptive pre-distortion in "barefoot" mode, crosstalk feedback

The HL2 was set to full output power and connected to a dummy load (no external PA, no external attenuators, no feedback cable). Then the pre-distortion "PURESIGNAL" menu was activated and a two-tone signal emitted. In this table we record the strength of the feedback signal as reported by the PURESIGNAL menu, and the "TXatt" (Tx attenuation) value, which is in the range 0 to 31 dB. During TX, the preamp is set to (19-TXatt) dB (in my original version of setting TXatt for HL2 in the piHPSDR program), so that the preamp during TX is between –12 and +19 dB.

To obtain a valid measurement of the strength of the feedback signal, the RX-gain slider had to be set manually to (19-TXatt) dB, such that the preamp setting does not change between RX and TX.

Band	FeedBk Lvl	TX att	RX gain	RX signal
160m	33	0	19	–42 dBm
80m	68	0	19	–36 dBm
40m	133	0	19	–30 dBm
20m	160	4	15	–24 dBm
10m	158	5	14	–23 dBm

Table 4: standalone PURESIGNAL feedback

For adaptive pre-distortion, the optimal "feedback" level is about 145 (arbitrary units, as reported by the WDSP library). Below ~70, the feedback signal is too weak for the predistortion to work, and above ~350, there is clipping. piHPSDR's engine records the feedback level and changes the TXatt value until a good feedback level is reached (in the range 130–160, which is the optimum value ± 1dB). The measured values are displayed in Table 4 (previous page).

No adaptive pre-distortion was effective on the 160m band, since the feedback signal was simply too weak. On the 80m band, it was still much too weak but sufficient for the predistortion algorithm to jump in. On the 40m band, the feedback signal strength (from the cross-talk in the RX/TX relay) was sufficient to allow adaptive pre-distortion at a nominal "TX ATT" value of 0, which corresponds to a +19 dB setting of the preamp. At higher bands, the "TX ATT" value is larger than zero, this means the preamp value gets smaller.

piHPSDR modification: To be able to use "barefoot" pre-distortion on the 160m band, the amplification range (31 dB wide) has to be shifted. The preamp is set to to (33 - TXatt) upon TX, if “Internal” feedback is selected in the piHPSDR program. This maps to preamp settings from +2 to +33 dB. If “EXT1” or “ByPass” feedback is selected in the PS menu (see the following figures), the preamp range is from –12 dB to +19 dB to allow strong feedback levels that will dominate the cross-talk from the TRX relay (see Sec. 8). Note that choosing “Ext1” or “ByPass” does not result in any relay switching (since the HL2 does not have such relays), this information if just “hi-jacked” to tell piHPSDR which range of attenuation to use.

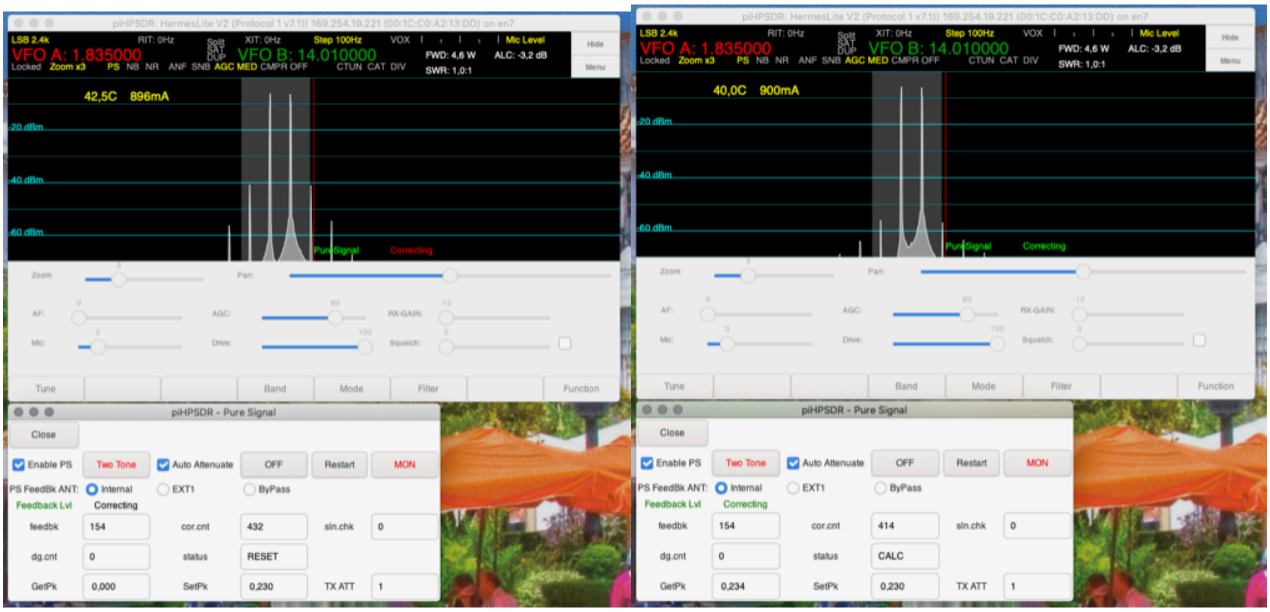

Fig. 3. PURESIGNAL on 160m (left: PS off, right: PS on). Note TXatt = 1, such that the internal preamp is at +32 dB

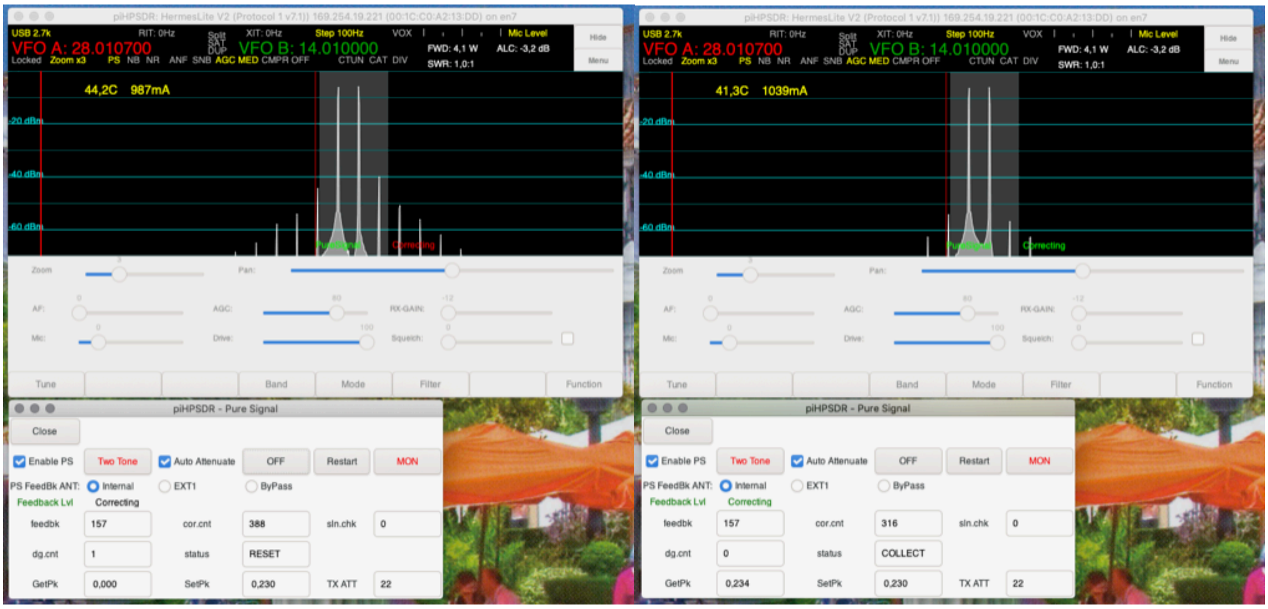

Fig. 4. PURESIGNAL on 10m (left: PS off, right: PS on). Note TXatt=22, such that the internal preamp is at +11 dB

Result of pre-distortion: According to piHPSDR's panadapter (Fig. 3 and 4), I can reach IM3 values of 50 dB (satellite peaks vs. main peaks) which is 56 dB according to the ARRL definition. In the PS menu of piHPDR, you can deactivate PS by hitting the "OFF" button and activating it by hitting the "Restart" button, and this is how the following screen shots have been produced. In both cases, the panadapter shows the RX feedback signal (MON button red).

To measure the "real world" performance, the HL2 has been connected to a (low-cost) SpecAnalyzer via a –40 dB attenuator (see Sec. 2). The experiment has been done on the 20m band:

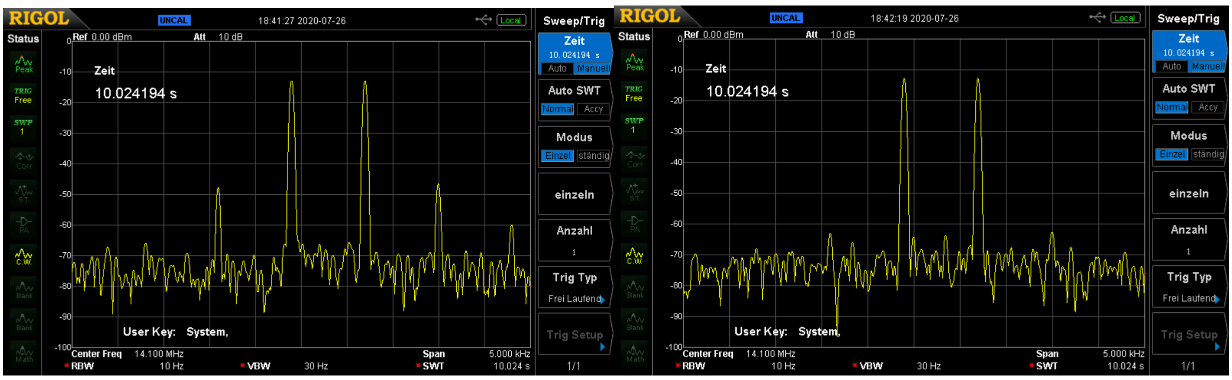

Fig. 5. TX signal on 20m band, PURESIGNAL off (left) and on (right). The N2ADR low-pass filters were active in both cases.

This matches what we have found using piHPSDR's on-board facility alone, the PA has 35 dBc IM3 which is already a rather good value. Then, we activated PURESIGNAL and obtained about 50 dBc which also is in agreement with the measurement reported above. Finally, the N2ADR filters were switched off by manually disabling the OC bits 4 and 7 (which are normally set on 20m TX), and the result (PURESIGNAL) is almost indistinguishable from the previous one:

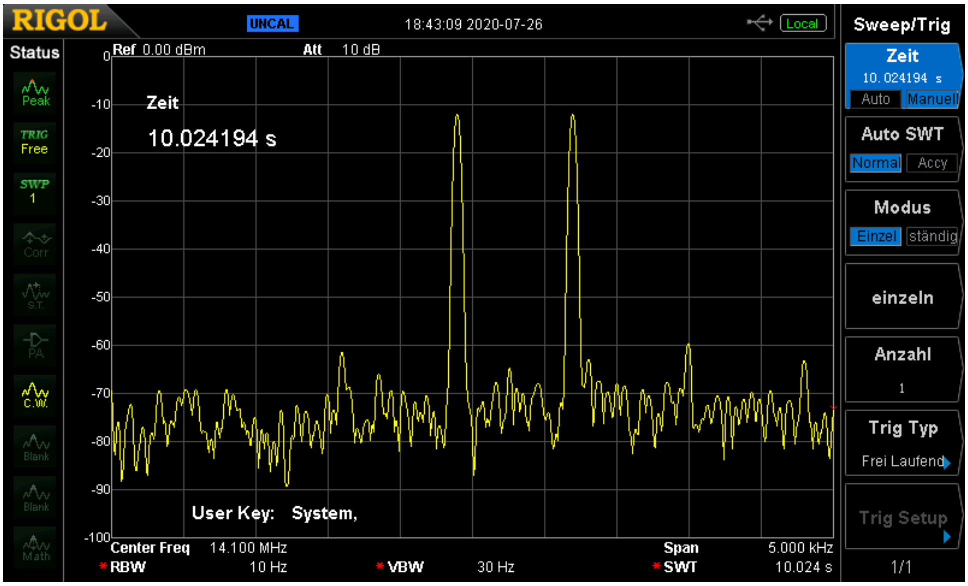

Fig. 6. TX signal on 20m band, with PURESIGNAL and ***without*** *N2ADR low-pass filters.*

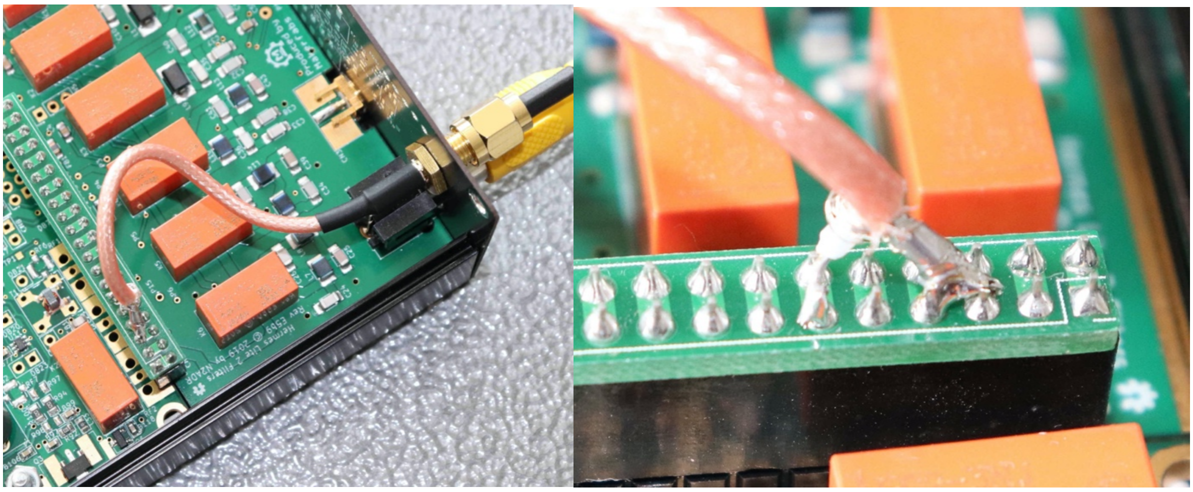

Fig. 7. Small coax cable with female SMA jack soldered to the HL2/N2ADR bridge (left panel: overview, right panel: detail of the bridge)

8. Using an external feedback path

The HL2 and the N2ADR filter board are connected via a small daughter-board bridging the connectors on the two boards, such that it is relatively easy to solder a small coax cable there and get a female SMA “RF3” connector mounted in the case. Fig. 7 (see previous page) shows the result, and Fig. 8 shows the HL2 case now featuring the additional "RF3" connector:

Fig. 8. Back view of the HL2 case, showing the new RF3 jack near the top left corner

The RF3 jack can now be used to provide a feedback signal from an attenuator, it should be much stronger than the "cross-talk" feedback which we cannot switch off! Therefore, "EXT1" feedback should be selected in the piHPSDR PURESIGNAL menu. While selecting "EXT1" instead of "Internal" feedback does not switch any relays (there aren't any!), it changes the mapping of TX attenuation to preamp values.

Using the HL2 "barefoot" the same results as with cross-talk feedback have been obtained, external feedback is however necessary when using external PAs. So I made a test with a 50-year old Yaesu FL2100 tube amplifier which makes about 50–70 watt output if driven with 4–5 watts from the HL2. This was connected to a dummy load with a –60dB attenuated output. The result is shown in Fig. 9, and IM3 satellites at –48 dB below the main peaks were observed.

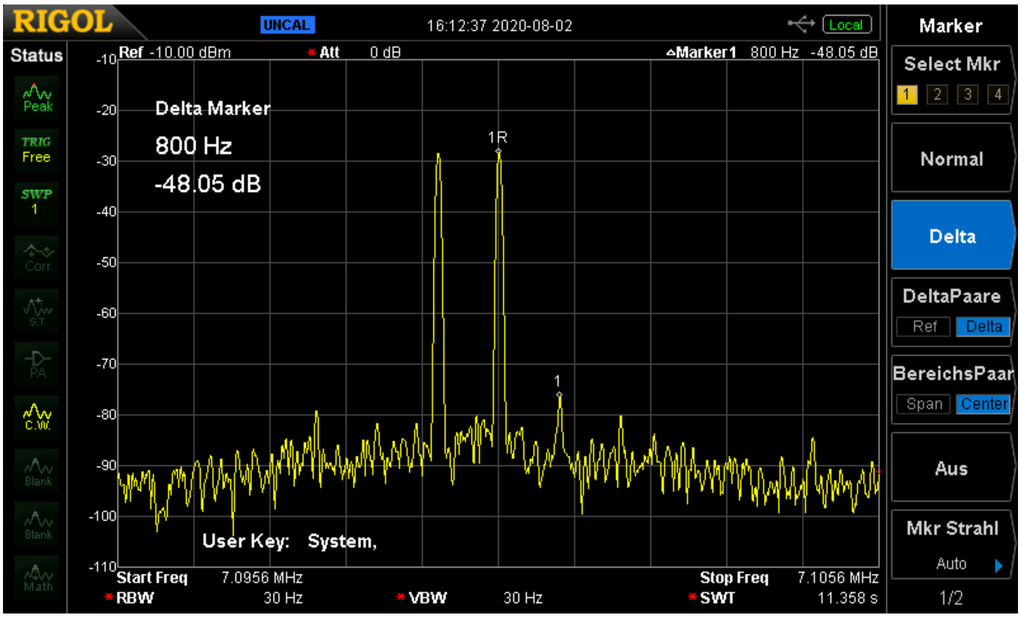

Fig. 9. 50 watt output of HL2 + external PA, using PURESIGNAL with external feedback

Note further that now there are clearly visible IM5 satellites but more than 50 dB below the main peaks. This is a very good result, since the PA tends to make rather strong IM5 satellites which are now obviously corrected (the strength of the IM5 satellites depends somewhat on how well the PA is tuned). To demonstrate the improvement gained by external feedback, the feedback has been disconnected and only crosstalk feedback has been used. That means, the PA itself is not part of the correction loop but is feed with a rather pure signal from the HL2.

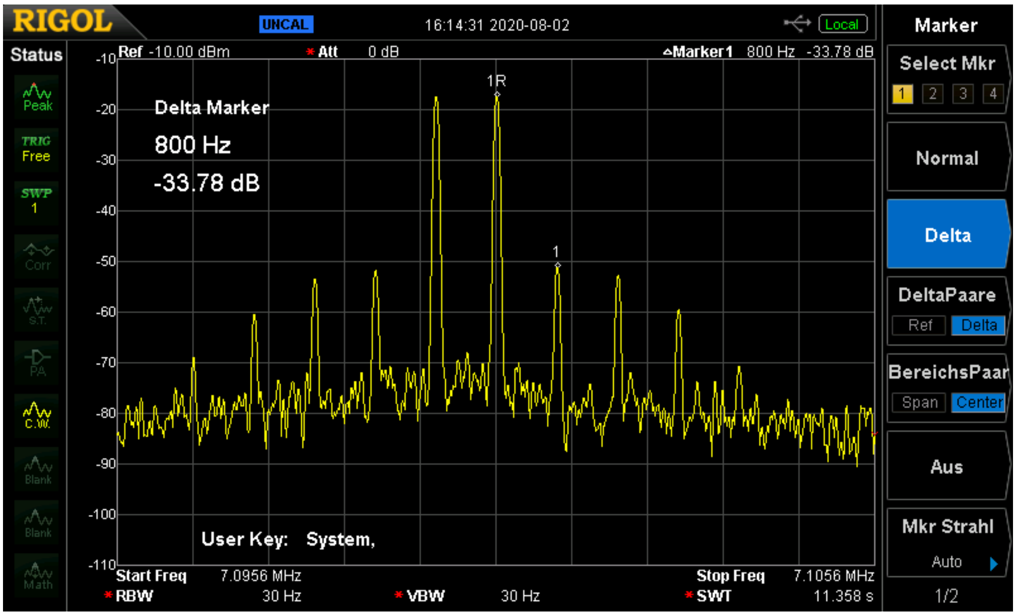

Fig. 10. 50 watt output of HL2 + external PA, using PURESIGNAL with internal feedback to correct the HL2 output, but the external PA does not contribute to the feedback

The result (Fig. 10) is not bad (IM3 satellites at –34 dBc, IM5 satellites only slightly below) and show the intrinsic linearity of the PA. This is the best result one can reach without external feedback.

Finally, one can speculate why "only" –48 dBc have been reached with the external PA, while –52 dB have been obtained in the same experiment when the PA was switched off (figure not shown). A slight degradation is to be expected because the RX "sees" a superposition of the external feedback and the crosstalk from the TRX relay, such that the PURESIGNAL algorithm does not "know" what is going on at the antenna. If this were the reason, an improvement should be seen when replacing the –60 dB attenuator (which is on the strong side for a 50 W signal) by –50 dB, but this is at the moment not available.

9. Operating CW with the HL2

One possibility is, of course, to create the CW signal entirely within the SDR application (piHPSDR in my case). To this end, the CW key/paddle need be connected to a GPIO input line (if piHPSDR is run on a single-board computer such as the Raspberry Pi) or key/paddle actions have to be reported as MIDI note on/off events. The latter case is the only option if piHPSDR runs on a Macintosh or Linux PC/laptop. I have soldered three wires to the PCB of my Behringer MIDI console such that pushing the left or right paddle shortens one of two buttons. Others have reported that they use a small Arduino micro-controller which can be programmed such that it is "seen" as a MIDI device if connected to a computer, and wrote a small Arduino code that translates key presses (Paddle connected to Arduino I/O pins) to MIDI messages. This is all not specific to the HL2, so we need to follow this further here. But let us note here that there will always be a small (at least 20 msec) delay of the side tone produced, which comes from the audio system of the computer, and this may lead to errors if you go beyond 25 wpm CW speed. So if it frequently happens that you send a digit "4" where you intended to send a letter "v", read the next paragraph.

The HL2 has an 1/8 inch stereo jacket where a CW and PTT signal can be delivered to the HL2. Note that there is no internal keyer, so *electrically* a straight key (or a mechanical bug) has to be connected here. An external tone generator could generate a side tone while using a straight key. So I took a K1EL WinKey external keyer, and connected the "Key" and "PTT" outputs to the 3.5 jack in the HL2 (a PTT lead-in delay was chosen in the WinKey device, mostly to make a downstream PA happy). With this setup, doing CW with the HL2 was easy and smooth, except that the side tone (which comes from a small speaker within the WinKey device) is somewhat low if you use headphones. If a WinKey device is to be used by default, one might consider feeding the sidetone output electrically to the headphone. The big advantage of this setup is a nearly delay-free side tone, the disadvantag is, of course, that you need the HL2 in close distance. If you plan remote operation "local" CW (paddle connected to the computer) is the only reasonable option.