DEMONSTRATIONS

OPEN RESEARCH INSTITUTE

MARCH 2026

INTRODUCTION

March Forth

It's March 2026 and this issue has a special focus on upcoming prototype demonstrations. Demonstrations, documentation, and deadlines are the daily drivers of ORI's volunteer open source community. What demonstrations are coming up? How did each of them come about? What purpose do they serve?

Open Research Institute is a non-profit dedicated to open source digital radio work on the amateur bands. We do both technical and regulatory work. Our designs are intended for both space and terrestrial deployment. We're all volunteer and we work to use and protect the amateur radio bands. You can get involved by visiting https://openresearch.institute/getting-started

Membership is free. All work is published to the general public at no cost. Review and download our work at https://github.com/OpenResearchInstitute

We equally value ethical behavior and over-the-air demonstrations of innovative and relevant open source solutions. We offer remotely accessible lab benches for microwave band radio hardware and software development. We host meetups and events at least once a week. Members come from around the world.

We are in excellent shape for a successful 2026, with FutureGEO prototypes planned, AMSAT-UK FunCube+ Mode Dynamic Transponder deliveries, Opulent Voice ASIC designs, Haifuraiya HEO/GEO satellite demonstrations, and a lot of R&D… all funded and scheduled.

We could use more staff. If you've ever wanted to be more involved in high tech amateur radio work, then now is the time and we can help you get there. We're here to help ordinary people do ambitious work. You do not need to be an expert to join ORI. You just have to be willing to become more of one along the way.

Get your questions answered at hello@openresearch.institute
Visit https://openresearch.institute/getting-started to join mailing lists and workspaces.

Want more Inner Circle Newsletters? Use the QR code at left or go to http://eepurl.com/h_hYzL and sign up.

Issue Contents

Find Us Online

YouTube https://www.youtube.com/@OpenResearchInstituteInc
Twitter (X) https://x.com/openresearchins
FaceBook https://www.facebook.com/openresearchinstitute/
LinkedIn https://www.linkedin.com/company/open-research-institute-inc
Fediverse https://micro.blog/OpenResearchIns

CONTENTS

The Case of the Missing Transmit Power

How a 4-bit Misalignment Stole 24 dB from the Opulent Voice Modem

The Opulent Voice modem for the LibreSDR graduated from the lab to the field in late March 2026. Instead of coaxial cables connecting transmitter to receiver, and receiver to transmitter, we now connected our brave little radios to filters and outdoor antennas.

And, nothing was received! The signal levels appeared to be very low. Even moving the antennas right next to each other resulted in only a few scattered frames demodulated and decoded. Obviously, we needed an amplifier. Fortunately, we had plenty in stock from collaborating with University of Puerto Rico's RockSatX team. They used an earlier version of Opulent Voice on their sounding rocket.

From the original listing at https://www.ebay.com/itm/363233702995

Microwave RF Power Amplifier Board SBB5089+SHF0589 40MHz-1.2GHz Gain 25DB 10PCS

Specifications:

– Input voltage: 10~30V DC
– Input power: about 5W
– Working frequency: 40MHz~1.2GHz (0.04~1.2GHz)
– Gain: about 25dB (may be higher)
– Power: 2W (may be higher)

Attention:

We measured 80.7% ultra-high efficiency in tests, and the official chip manual also mentioned that there is more than 50% efficiency at P1dB. Overall, this SBB5089+SHF0589 is better than SBB5089+SHF0289.

On 24 March 2026, we selected one of the amplifiers at random in order to characterize it in ORI's Remote Labs. We connected the input of the amplifier to the output of the DSG821A signal generator. The signal generator was set to 431 MHz, which was the frequency we wanted to use. We connected the output of the amplifier through a 6 dB attenuator to the Rigol RSA5065N Spectrum Analyzer. We fitted a JST-HX power cable to the power connector of the amplifier. We provided 12 volts of power from the DP832 lab power supply.

The amplifier made 27 dB of gain from -100 dBm input to about -3 dBm input, made 20 dB at 0 dBm input, and worked pretty well up to 9 dBm input.

The next test was to remove the signal generator and connect a LibreSDR running Opulent Voice. Instead of a carrier wave from the signal generator we'd be sending an 81 kHz wide minimum shift key (MSK) signal from a real modem through the amplifier. We were intending to repeat the measurements we'd made with the signal generator. However, we noticed something very interesting. The signal level from the LibreSDR was expected to be about 0 dBm, which would provide enough drive to the amplifier to create enough gain to help our over-the-air tests succeed. However, when the LibreSDR, running Locutus and Dialogus, was commanded to transmit with PTT and audio frames from Interlocutor, the peak of the main lobe of the MSK signal was at 1 microwatt. If this was the true power output of the LibreSDR, then no wonder the over-the-air tests had failed.

The transmit power hardware attenuation setting was confirmed to be at 0 dB. This is set through an Industrial Input and Output (IIO) library attribute call, was correctly reported, and we saw that changing the attribute caused the signal to increase or decrease by the exact amount of gain. So, it wasn't a configuration error. As far as the hardware was concerned, it was transmitting at 0 dBm.

The other possibility was that the I and Q signals were not being generated for transmit at full scale. If we weren't filling up the registers correctly, then maybe we were accidentally dividing our signal down before it got to the antenna. Investigation turned to the Hardware Descriptive Language (HDL) files.

The Opulent Voice VHDL language modem, called Locutus, runs inside the LibreSDR FPGA. Data frames arrive via direct memory access, pass through the Opulent Voice frame encoder, are convolutionally encoded (K=7, rate 1/2), go through a byte-to-bit deserializer, and the resulting bits are sent to the MSK modulator. The modulator produces the I and Q samples that drive the AD9363 digital to analog converter (DAC). Software in the general purpose processor of the LibreSDR configures the IIO context and controls PTT.

The direct memory access transfers protocol data frames into the LibreSDR, and not IQ samples. So, the classic PlutoSDR bug of 12-bit samples being miscounted in a 16-bit word did not apply here. The modulator itself generates all I and Q waveforms. The frequencies are set by Dialogus at startup.

The Integrated Logic Analyzer (ILA) in the bitstream already had probes on two very important signals, tx_i_sync and tx_q_sync. These signals were measured right at the point where the samples enter the AD9361 core. A January 2026 ILA capture told the story clearly. The waveform showed clean MSK signals. No corruption, no skips, and with the exact right relationship to each other. At the time, this was a big milestone and part of the process of troubleshooting the porting of the HDL code from the PlutoSDR to the LibreSDR. But we'd overlooked something critical. The bug was right there in an otherwise perfect image.

The peak values of the I and Q waveforms were only plus and minus 1100 or so, in a 16-bit signed word. At first glance, a value of 1100 in a 16-bit word might not raise any red flags. The alarm bells ring when you know how the axi_ad9361 core actually reads those 16 bits.

There are two different conventions on the same bus. The axi_ad9361 core uses 16-bit data buses internally. However, the AD9361 and AD9363 (the chips used in these software-defined radios) have only 12-bit digital to analog converters. The documented convention, confirmed by a tour through Analog Devices Engineer Zone forum, is as follows.

RX (ADC Output) is 12-bit value in [11:0], sign extended to [15:12]
TX (DAC Output) is 12-bit value expected in [15:4], which is the top 12 bits

In plain English, RX gives you the data right-justified. TX expects it left-justified. These are opposite conventions on the same 16-bit bus, and the apply whether the interface is CMOS (PlutoSDR) or LVDS (LibreSDR).

Our code, from msk_modulator.vhd, in the carrier_mod_proc section looks like this.

```
tx_samples_I <= std_logic_
vector(resize(s1s + s2s,
SAMPLE_W));
tx_samples_Q <= std_logic_
vector(resize(s1c + s2c,
SAMPLE_W));
```

s1s and s2s are each signed 12-bit values from the numerically controlled oscillator (NCO). The lookup table fills using the command

`ROUND(SIN(theta) * 1024.0)`, which gives a peak value of plus or minus 1024. VHDL addition of two such values produces a 12-bit result that ranges from -2048 to +2048. So far so good. The resize call then sign-extends that 13-bit result into 16 bits. This is a right-justified 16-bit word, which is the opposite of what the Analog Devices core expects.

The full chain of what happens to the signal amplitude can be calculated.

The lookup table output is [11:0] signed and is a

12-bit sinusoid.
s1s + s2s is [12:0] signed and is a 13-bit sum. Resize(…, 16) [15:13] sign extension with [12:0] as the data. This is right-justified. Analog Devices chip reads transmit values as [15:4], sending the top 12 bits to the DAC. Analog Devices reads [15:4], we drive [12:0], and this is a divide by 16 to the amplitude.

What's the damage? -24 dB.

Why did this work in the PlutoSDR? Well, it didn't. It did not produce full power, either. The same modulator code drove the Pluto variant of Opulent Voice. The -24 dB bug was there too. Why did we not notice it? We never graduated to over-the-air tests with the PlutoSDR. All of the tests transmissions were in the lab and were either conducted through coaxial cables or done with Vivaldi lab antennas right next to each other on the bench. With conducted tests, everything worked perfectly.

For ORI's LibreSDR work, we were now in the field. We wanted to characterize the modem output before adding an amplifier. That scrutiny revealed the long-lived bug in the HDL.

Matthew Wishek NB0X implemented a fix on the tx_sample_scale branch of the published repository, with changes to two submodules, the NCO and the msk_modulator. No changes to the block design TCL or to msk_top.vhd were required.

In the NCO (sin_cos_lut.vhd), a new constant was introduced: CONSTANT FULL_SCALE : INTEGER := 2**(SINUSOID_W-1) -1. And, the lookup table fill function was changed from the hardcoded 1024.0 to * real(FULL_SCALE). With SINUSOID_W = 12, this gives FULL_SCALE = 2047, filling the entire signed 12-bit range. The fix is fully generic. It works for any value of SINUSOID_W.

```
-- Before:
tmp := ROUND(SIN(theta) * 1024.0);
-- After:
CONSTANT FULL_SCALE : INTEGER :=
2**(SINUSOID_W-1) - 1;
tmp := ROUND(SIN(theta) *
real(FULL_SCALE));
```

In the modulator (msk_modulator.vhd), a new 3-bit input port tx_shift : IN std_logic_vector(2 DOWNTO 0) was added. The IQ output assignment was changed from a plain resize() to a shift_left() whose amount is driven by tx_shift at runtime.

```
-- Before:
tx_samples_I <= std_logic_
vector(resize(s1s + s2s,
SAMPLE_W));
-- After:
tx_samples_I <= std_logic_vector(
    shift_left(resize(s1s + s2s,
SAMPLE_W),
        to_integer(unsigned(tx_
shift))));
```

The full 12-bit scale was achieved. With the sum now peaking at plus or minus 4094, left-shifting by 3 puts the signal in the correct place, which is [15:3]. The Analog Devices core reads [15:4], which is the full DAC scale. Making tx_shift a configurable port rather than a hardcoded constant is an elegant touch. Dialogus sets it through the register map at runtime, with no bitstream rebuild needed.

With the tx_sample_scale fix integrated and a new bitstream loaded, the Opulent Voice modem then achieved its first successful over the air transmission. This was from one building to another, with the full signal chain, from a LibreSDR to another LibreSDR. Voice traffic and text messages were received, with excellent audio quality. The ~30 dB shortfall that had been quietly sitting in the hardware since the original modulator was gone.

Lessons Learned

RX and TX use opposite justify directions in axi_ad9361. This is documented, but really only in a so-called Verified Answer on Analog Devices Engineer Zone forum. It's not prominently documented in the IP wiki. The

wiki describes the 16-bit format and mentions that the IP “always works in 16 bits”, but does not call out the left/right justification asymmetry in a way that is easy to find. If you are writing custom HDL that drives DACs, then you should read the forum thread at https://ez.analog.com/fpga/f/q-a/112155/axi_ad9361-data-format

ILA probes are worth their cost. The screenshot from the ILA capture back in January 2026 told us the answer, if we had known what the question was. Running ILA and keeping the results pays off because you can go back and look at signals that may not be accessible otherwise. Wire up ILA early and often and be curious about your signals. Go for a tour. Explore your design and the design of any infrastructure that you are working with.

-24 dB is a recognizable signature. In fact, any multiple of -6 dB is significant. Each bit of DAC resolution is 6 dB, so if you’re missing something like 24 dB, then an inadvertent four-bit shift might be the culprit.

Fix things at the right layer. The initial discussions included assumptions such as “the fix should live in the block design TCL file” or maybe in msk_top. Matthew chose to fix it inside the modulator and NCO submodules. This is the better choice. It makes the modules self-consistent, removes the need for platform-specific fancy workarounds or settings, and ensures that any future target automatically benefits. When a submodule’s output format is wrong, fix the submodule rather than papering over it at the integration layer.

Given that Opulent Voice will be demonstrated at BSides San Diego on 4 April, resolving this bug makes that demonstration possible.

Acknowledgements

The modulator and NCO were written by Matthew Wishek NB0X, whose clean modular architecture made the bug straightforward to trace, and whose tx_sample_scale branch fix resolved it elegantly at the right layer.

Thanks to the ADI FPGA team (Laszlo) for the EngineerZone Verified Answer that became our primary citation. Thanks to Paul KB5MU and Michelle W5NYV for working through this signal chain, characterizing the amplifier, and methodically testing the new firmware.

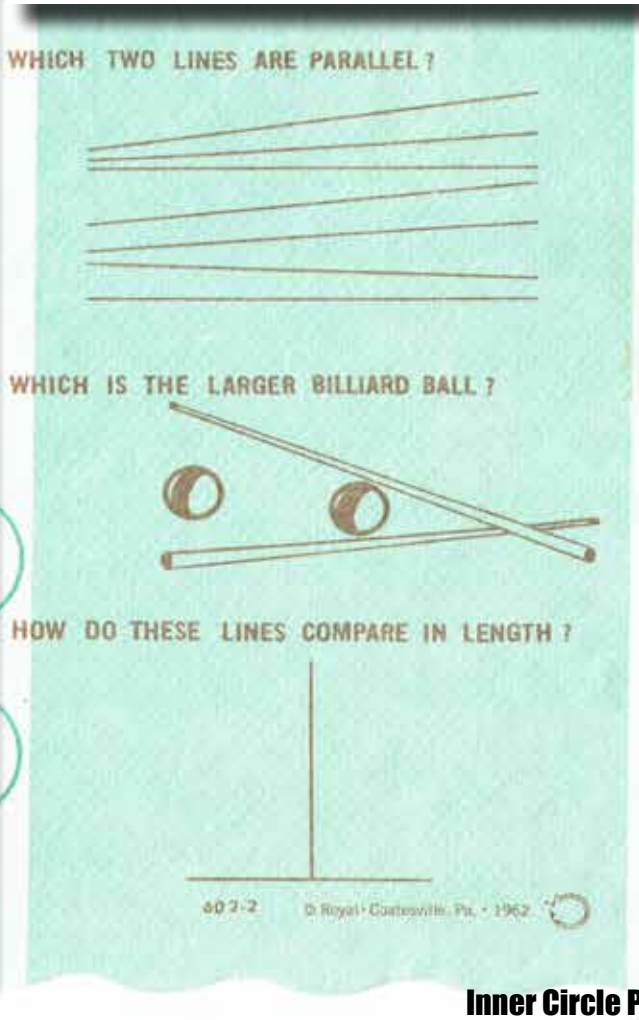
ADI EngineerZone — AXI_AD9361 Data Format (Verified Answer): ez.analog.com/fpga/f/q-a/112155/axi_ad9361-data-format

ADI Wiki with AXI_AD9361 IP documentation: wiki.analog.com/resources/fpga/docs/axi_ad9361

ORI pluto_msk repository (tx_sample_scale branch): github.com/OpenResearchInstitute/pluto_msk

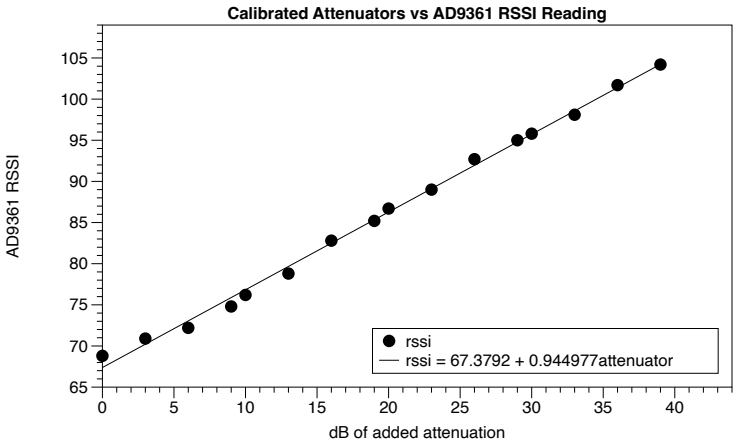
ORI msk_modulator repository: github.com/OpenResearchInstitute/msk_modulator

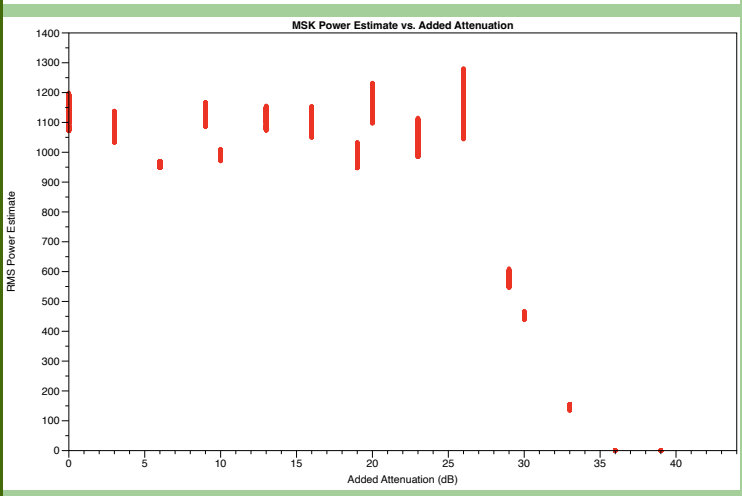
ORI nco repository (sin_cos_lut fix): github.com/OpenResearchInstitute/nco

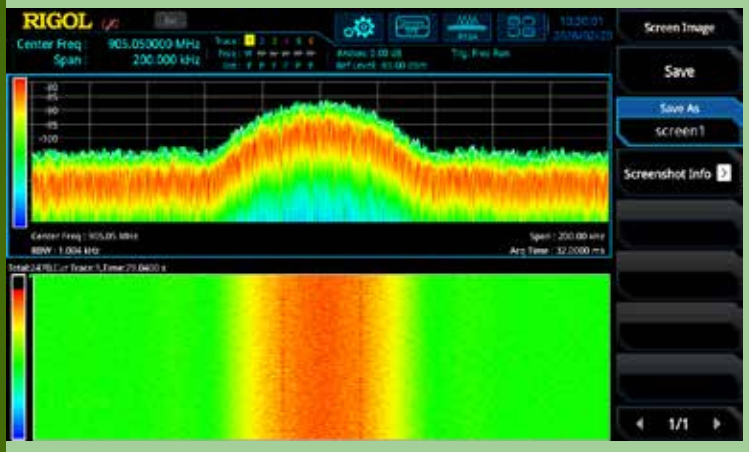


A Picture is Worth a Thousand Words

Paul KB5MU







Top left, is a power estimation calibration on LibreSDR pluto_msk commit 9db8

“I used LibreSDR4 as an Opulent Voice signal source, followed by a 20 dB inline attenuator and a 20-inch skinny coax cable. Extra attenuation was inserted after that, built from 3, 6, 10, 20, and 30 dB inline attenuators (accuracy previously checked on the Rigol spectrum analyzer).

I ran each test for about a minute, and extracted two values from each 10ms debug sample: the AD9361’s RSSI (assumed to be nominally in negative dB), and the MSK modem’s I-channel RMS power estimate as read from the rx_power register.

Middle left, the receiver’s analog AGC is evident here. All tests with 26 dB or less of extra attenuation show in the power estimators about the same value, 962.5 to 1167. The lock threshold seems to be around 30 dB of extra attenuation, at which point the power estimator shows around 450. The power estimator is still giving meaningful results another 3 dB down, but a further 3 dB attenuation gives power estimates of zero.

Bottom left is what the spectrum looks like at the threshold of 30 dB added attenuation. (Approximately, because there’s a different cable involved and one additional adapter.) Notice that the side lobes are completely submerged in the noise floor, and the peak amplitude is around -88 dBm.”

If a picture is worth a thousand words, then a video is worth a thousand pictures. Visit our YouTube channel at https://www.youtube.com/@OpenResearchInstituteInc to see all our presentations, lab demonstrations, and project meetups. Subscribe to get notifications of new content.

Inner Circle Sphere of Activity

4 April 2026 - BSides San Diego RF Village participation and demonstation.

26 - 28 June 2026 - ESA FutureGEO Workshop, Friedrichshafen HAMRADIO 2026, participation and demonstration. Find us at AMSAT-DL booth.

6 - 9 August 2026 - DEFCON RF Village participation and demonstration.

If you know of an event that would welcome ORI, please let your favorite board member know at our hello at openresearch dot institute email address.

Thank you to all who support our work! We certainly couldn't do it without you.

Anshul Makkar, Director ORI
Steve Conklin, CFO ORI
Michelle Thompson, CEO ORI
Matthew Wishek, Director ORI

Locutus MSK Modem: Symbol Synchronization Lock

Summary

Synchronization detection is an important metric for communications systems. It provides a mechanism to qualify received data as invalid or possibly valid. That is, if synchronization is unlocked the received data is invalid, and if locked data is likely valid. Additionally, lock signal at various levels in the hierarchy are especially useful in diagnosing misbehaving communication links. Here we review the synchronization lock detection in the Locutus MSK Modem/PHY.

OSI 7-Layer Model

The OSI (Open Systems Interconnect) Seven Layer model provides a framework for describing and standardizing network communication functions into layers. The figure below shows the seven layers and how computer networking functions map to each layer.

Table 1. OSI Seven Layer Model

7. Application Layer	User interface, HTTP, FTP, etc
6. Presentation Layer	Data formatting, encryption, compression
5. Session Layer	Session Management
4. Transport Layer	End-to-end connections - TCP, UDP
3. Network Layer	Routing - logical addressing (IP)
2. Data Link Layer	Point-to-Point - Ethernet II Frames using MAC addressing
1. Physical Layer	Physical medium - cables, radio waves

Symbol/Bit Lock in Ethernet

When an Ethernet cable is connected between two devices there is an indiction that the link is active via an LED on the device connector. When the cable is plugged into the first device LED does not light as the link is not yet active. When the cable is then plugged into the second device we see the LED on both devices will turn on. This is a form of Layer 1 / PHY symbol lock.

When the second connection is made the Ethernet PHY chips at each end start receiving a signal from the other end, they then negotiate link parameters like link speed (e.g. 10 Mbps, 100 Mbps, 1000 Mbps). Once the negotiation is complete the PHY chip will light the LED.

OSI Model as Applied to Opulent Voice and the Locutus MSK Modem

The OSI model can be applied to any communications system. Some layers may not be used, or may be merged, but the framework is still useful. The table below shows the two lower layers of the OSI model as applied to the OPV stack using the Locutus MSK Modem.

PHYs and Modems

Note here that the Ethernet PHY chip mentioned previously is a modem of sorts that takes Layer 2 data frames and converts the frame bits into signals transmitted across the Ethernet cable. The Locutus MSK Modem does the same taking Layer 2 frames (any Layer 2 frame, not necessarily Ethernet II frames) and converts the frame bits into signals transmitted across space using radio waves. So we can use the terms PHY and Modem interchangeably, although we will use Modem for the remainder of this discussion.

Table 2. OSI Seven Layer Model

Layer	Name	OPV Function
2	DataLink Layer	Opulent Voice Frame
1	PHY Layer	MSK Modem/PHY

In the Opulent Voice system the MSK Modem is equivalent to the Ethernet PHY Layer and the Opulent Voice frame is equivalent to an Ethernet II frame. Now let us consider how synchronization locks are applied in this system.

Symbol Locks and Frame Locks

Layer 2 OPV Frame Sync Lock

The OPV frame has a synchronization preamble that is detected at the receiver. Frame synchronization is necessary to identify the start of incoming OPV frames which allows the frames to be decoded. The decoder only starts decoding when frame synchronization is detected, and we call this state *frame sync lock*. The frame synchronization detector continues to detect the frame sync preamble for every frame, and if it doesn't detect the frame sync at the start of the next frame *frame sync unlock* is declared, after which the frame decoder will stop decoding frames, as the received data is undefined.

Layer 1 MSK Modem Symbol Lock

The MSK transmitter takes the incoming bits of the OPV frame and translates them into I/Q symbols that are then transmitted over the wireless radio link. The MSK receiver has to then *lock* to the received symbols. This process is done using a *Phase-Lock Loop* (PLL) that detects and synchronizes to the incoming radio waves. Once the PLL achieves synchronization *symbol sync lock* is declared

and the received symbols are converted back the bits making up the OPV frame.

The *symbol sync lock* from the MSK receiver is an important qualifier to the OPV frame decoder. If the OPV framer decoder is receiving invalid bits because we don't have symbol lock then the frame sync detector will be trying to lock to invalid bits from the MSK receiver. The detector might incorrectly detect random bits as a frame sync. This results in the frame detector assuming that it has found a frame sync and will look for the sync again at the next frame. That the data is invalid can cause the frame sync detector to miss the real frame once the MSK receiver achieves symbol lock. This isn't a fatal condition, but it will mean that one or more OPV frames are lost while the frame synchronizer recovers. The frame synchronizer uses the MSK receiver symbol lock status as a gating signal to only start frame detection once symbol lock achieved.

Locutus MSK Symbol Synchronization

Costas Loop

The MSK receiver uses two Costas Loops to detect the F1 and F2 frequencies used in MSK modulation. The Costas loop is a form of Phase-Locked Loop. The following diagram shows one of the Costas loops in the MSK receiver.

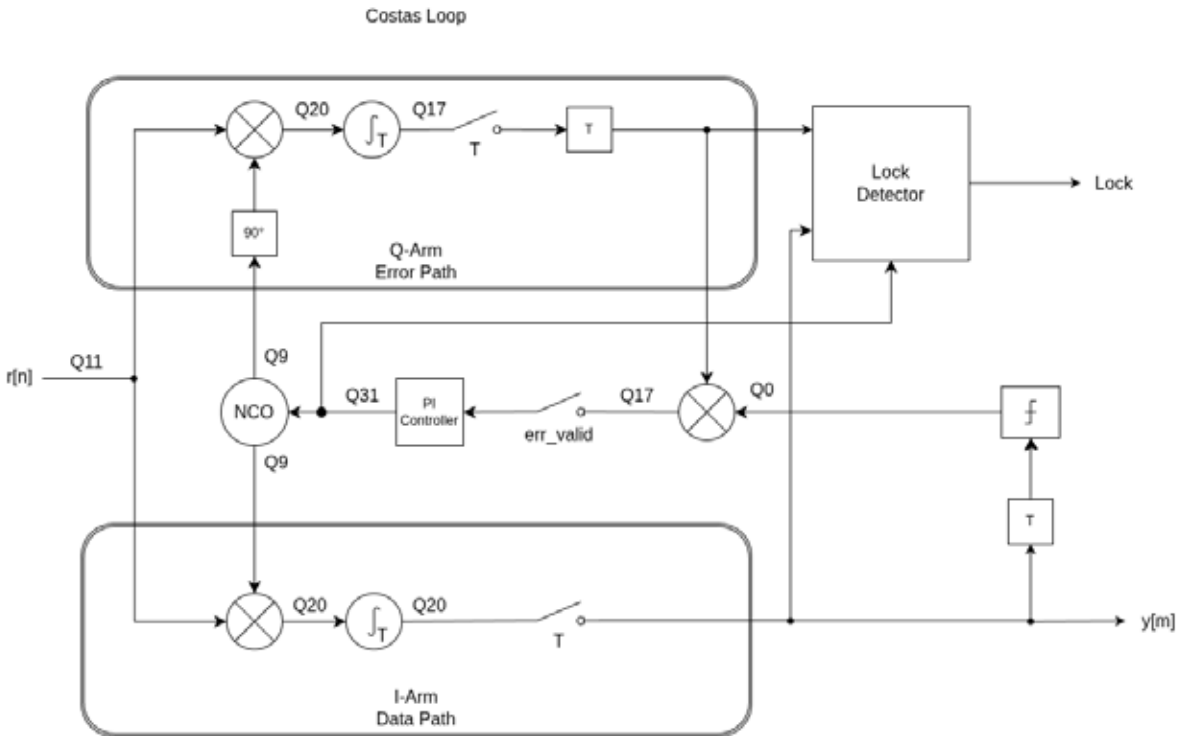


Figure 1. Image Costas Loop PLL for MSK Receiver

The received MSK signal consists of a cosine wave that switches between two frequencies f_1 and f_2. For the current discussion let examine one input frequency f_x.

The Costas loop takes the received signal and sends it to two "arms", the I (in-phase) and Q (quadrature-phase). The Costas loop then attempts to lock to the frequency and phase of the incoming signal.

In-Phase Arm

The I arm mixes the incoming signal with the NCO cosine output and accumulates mixer output over 1 bit time (T_b). The incoming signal is:

$$r_n = \cos(2\pi f_x n)$$

We expect that the NCO signal has both a frequency and phase offset from the incoming signal:

$$nco_{i,n} = \cos(\theta_{nco_n}) = \cos(2\pi(f_x + \Delta f)n + \theta)$$

where θ_{nco} is the NCO phase output, Δf is the frequency offset, and θ is the phase offset.

The mixer output mix_i is:

$$\begin{aligned} mix_{i,n} &= r_n \cdot nco_{i,n} \\ &= \cos(2\pi f_x n) \cdot \cos(2\pi(f_x + \Delta f)n + \theta) \\ &= \frac{1}{2}[\cos(2\pi f_x n - 2\pi(f_x + \Delta f)n - \theta) + \cos(2\pi f_x n + 2\pi(f_x + \Delta f)n + \theta)] \\ &= \frac{1}{2}[\cos(-2\pi \Delta f n - \theta) + \cos(4\pi(f_x + \frac{1}{2}\Delta f)n + \theta)] \end{aligned}$$

The first term is near DC and it the signal of interest. The second term is near $4f_x$ and can be disregarded.

The mixer output is then accumulated over T_b to produce the accumulator output for bit k.

$$acc_{i,k} = \sum_{n=kN}^{((k+1)N)-1} \frac{1}{2}[\cos(-2\pi \Delta f n - \theta)]$$

where N is the number of samples in a bit period, such that $N = T_b/F_s$ where F_s is the system sample rate.

When the loop is unlocked the accumulation for bit k is

$$0 < acc_{i,k} < \frac{1}{2}N$$

Let's assume the accumulation mean will be

$$\bar{acc}_{i,k} \approx \frac{1}{4}N$$

with a large variance.

When the loop is locked then $\Delta f \approx 0$ and $\theta \approx 0$ which simplifies the accumulation to:

$$\begin{aligned} acc_{i,k} &\approx \sum_{n=kN}^{((k+1)N)-1} \frac{1}{2}[cos(2\pi\Delta f_{res} n + \theta_{res})] \\ &\approx \frac{1}{2}N \end{aligned}$$

where Δf_{res} is the residual frequency error and θ_{res} is the residual phase error, which will both be close to 0.

Quadrature Phase Arm

The Q arm computes an error signal that adjust the NCO frequency via a PI controller. This arm works similarly to the I arm where the incoming signal r_n is mixed with the sine output from the NCO (90 degree phase shift from the I arm).

$$nco_{q,n} = sin(\theta_{nco_n})$$

$$\begin{aligned} q_{mix} &= r_n \cdot nco_{q,n} \\ &= \sin(2\pi f_x n) \cdot \sin(2\pi(f_x + \Delta f)n + \theta) \\ &= \frac{1}{2}[\sin(2\pi f_x n + 2\pi(f_x + \Delta f)n + \theta) - \sin(2\pi f_x n - 2\pi(f_x + \Delta f)n - \theta)] \\ &= \frac{1}{2}[\sin(4\pi(f_x + \frac{1}{2}\Delta f)n + \theta) - \sin(2\pi\Delta f n - \theta)] \end{aligned}$$

As with the I arm, the term near DC is of interest and the term near $4f_x$ can be disregarded. The accumulation is now:

$$acc_{q,k} = \sum_{n=kN}^{((k+1)N)-1} \frac{1}{2}[-\sin(2\pi\Delta f n - \theta)]$$

We can see again that when the loop is unlocked

$$0 < acc_{q,k} < -\frac{1}{2}N$$

And let us assume the accumulation mean will be

$$\bar{acc}_{q,k} \approx -\frac{1}{4}N$$

with a large variance.

And when the loop is locked $\Delta f \approx 0$ and $\theta \approx 0$ resulting in the accumulated value being:

$$\begin{aligned} acc_{q,k} &= \sum_{n=kN}^{((k+1)N)-1} \frac{1}{2}[-\sin(2\pi\Delta F_{res} n + \theta_{res})] \\ &\approx 0 \end{aligned}$$

Symbol Sync Lock Detection

The accumulated values on both the I arm and Q arm provide a useful signal to detect symbol lock. When the loop is unlocked we can say:

$$|acc_i| \approx |acc_q|$$

And when the loop is locked:

$$|acc_i| > |acc_q|$$

The accumulator output represents the correlation between the received signal and the NCO reference over one bit period. Squaring this correlation produces an instantaneous (per bit) energy metric for each arm:

$$\begin{aligned} eng_{i,k} &= acc_{i,k}^2 \text{(squared correlation, proportional to energy)} \\ eng_{q,k} &= acc_{q,k}^2 \end{aligned}$$

The instantaneous energy will be very noisy and is not suitable for directly detecting symbol lock. Rather we can accumulate the per bit energy over detection period M.

$$\begin{aligned} sum\_eng_{i,m} &= \sum_{k=mM}^{((m+1)M)-1} eng_{i,k} \\ sum\_eng_{q,m} &= \sum_{k=mM}^{((m+1)M)-1} eng_{q,k} \end{aligned}$$

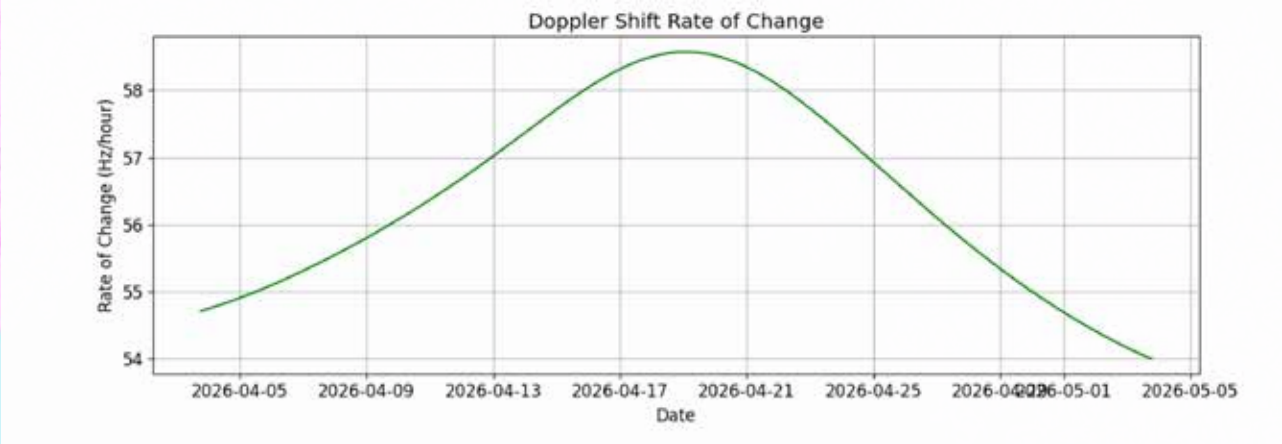
The energy sums can then be used for lock detection by looking at the differences between the I and Q arms. Since, these are energy metrics, we might ask why not compute a more standard RMS power for each arm and use it for lock detection. There are a couple of reasons why the energy sums is the preferred choice:

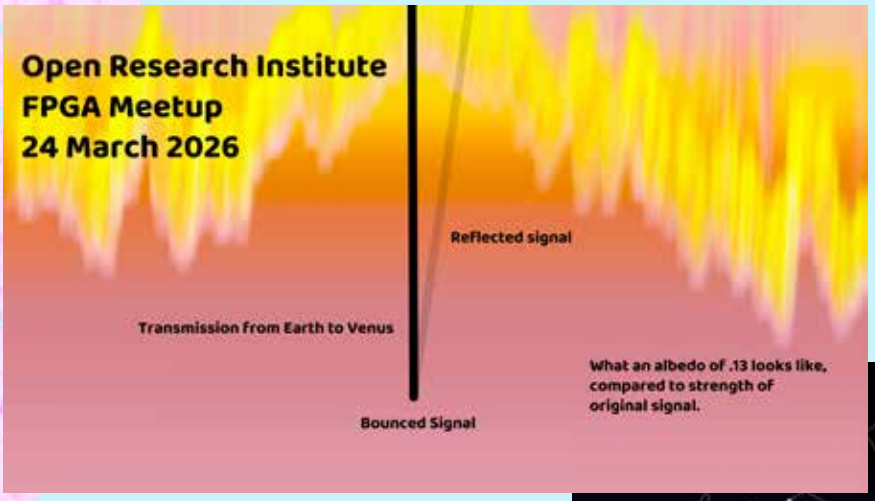
1. RMS power is $\sqrt{(\frac{1}{M}\sum_M eng_{i/q.m})}$ which requires computing a division by N and a square root. These are expensive (resources) and since are looking at the difference between the I and

Celebrating 730+ Subscribers on YouTube

Subscribe Here

Thank you for the support!





Q arms RMS power doesn't provide any more information than the energy sums.

2. Using energy sums provides a larger detection resolution. RMS power is normalized by N and the magnitude of the mean is further reduced by the square root. By using the un-normalized energy sums we have larger amplitude signals. When unlocked the I arm and Q arm are expected to have similar energy levels, and those relative levels will be about the same for both the energy sums and the RMS power. When the loop is locked the I arm energy sum will have a much larger mangitude than the RMS power, and the Q arm is expect to still be near zero for both the energy sums and RMS power. This provides a much larger range for selecting a lock threshold.

Now that we have the energy sums over the detection period M we can determine symbol sync lock status as follows:

$$L_m = sum\_eng_{i,m} - sum\_eng_{q,m}$$

$$L_m \geq L_{th} \rightarrow \text{locked}$$
$$L_m < L_{th} \rightarrow \text{unlocked}$$

where L_{th} is a configurable value.

The lock detection status is updated at each interval. There are two parameters that need to be configured for the lock detection to perform well: interval and threshold.

Selection of Lock Detection Interval and Threshold

Summary

Lock status from each layer of the OSI modem can be an indication to the higher layers whether the incoming data is valid or not. While system can function just fine with proper data/frame detection techniques using the relevant loc indicators can impove system robustness and may be necessary to meet system requirements.

Monument Peak Tower Collapse

February 2026 Storm Takes Down Communications Infrastructure on Mount Laguna, CA, USA

by Sudoku Ham for ORI

At exactly 10:00 AM on Wednesday, February 18, 2026, a communications tower on Monument Peak in the Laguna Mountains was blown over during a powerful wind event, captured in real time by a nearby wildfire camera system. The tower, owned by American Tower Corporation (ATC), had been carrying an AT&T cellular site and several microwave hops. According to sources familiar with the site, that was the only functional equipment on the structure at the time of its collapse.

The failure was documented by the HPWREN (High Performance Wireless Research and Education Network) camera system. This system is operated by UC San Diego's San Diego Supercomputer Center. Video assembled from the east-facing fixed-field-of-view camera shows a major wind event immediately preceding the collapse, with the tower going over at 10:00 AM. A before-and-after comparison of HPWREN still frames — one from February 13 showing the tower standing, another from later on February 18 showing it gone — confirms the loss. Snow visible in the post-collapse image and on the wreckage is consistent with the heavy winter weather that preceded the failure.

Damage photos obtained from CORA (Cactus Open Repeater Association) members, originally shared by Chris Baldwin, show the aftermath in stark detail. A lattice tower structure torn from its concrete foundation, the base ripped out of the ground with rebar exposed, and the wreckage draped across an equipment shelter labeled "FACILITY 3." The concrete pier appears to have failed catastrophically, with the entire foundation block uprooted rather than the tower buckling above the base. The combination of snow and ice loading on the structure, high sustained winds, and the age of the tower and presumed lack of recent maintenance all contributed to the failure.

Steve Hansen, W6QX, first drew attention to the HPWREN imagery showing the tower's disappearance.

The Storm

The collapse occurred during a series of storms that struck San Diego County over the span of four days. The first wave hit Monday, February 16, bringing heavy rain and winds gusting to 60 mph on Mount Laguna. A second, more intense wave arrived overnight Tuesday into Wednesday, the morning the tower fell. That wave produced winds of 80 mph at El Cajon Mountain, measured at 3:30 AM. Wind was measured at 76 mph at Birch Hill in the San Diego County mountains, and at 52 mph in the desert. The National Weather Service reported snow accumulations approaching a foot on Mount Laguna, with additional snow bands continuing through February 19. A third and final round brought further showers and gusty winds on Thursday the 19th before conditions improved Friday.

The HPWREN camera overlay on the February 18 image recorded conditions at Monument Peak of 31.1°F, 90.2% relative humidity, and 23.6 inHg barometric pressure. It was cold, and the front had clearly moved through.

What Was on the Tower?

Monument Peak (32.89°N, 116.42°W, at 6,271 feet) sits at the eastern edge of the Laguna Mountain Recreation Area within the Cleveland National Forest. It is one of the most significant multi-use communications sites in eastern San Diego County, with a coverage footprint extending from the Salton Sea south to the Mexican border and west across the county.


FACILITY 3

The ATC tower that collapsed was one of multiple structures at the site. The site hosts a diverse set of users and systems. Services known to operate from Monument Peak include the following.

Amateur Radio: The East County Repeater Association (ECRA) operates several repeaters from the Monument Peak site, including 147.240 MHz (+ offset, PL 107.2 Hz. K6KTA, which is a joint effort with CORA that participates in the CalZona Link), 446.750 MHz (- offset, PL 107.2 Hz), and 449.180 MHz (- offset, PL 88.5 Hz). These repeaters appear to have been on a different structure than the one that fell. Operators are encouraged to confirm current status on the air.

HPWREN/ALERT: This system from UC San Diego operates fixed-field-of-view and pan-tilt-zoom wildfire detection cameras, microwave backbone links, and a weather sensor suite from the site. Monument Peak is a backbone node in the HPWREN network and has been since the project transitioned from nearby Stephenson Peak. The HPWREN cameras that documented this collapse were themselves mounted on a separate structure and survived.

NASA Space Geodesy: The Monument Peak compound hosts NASA's MOBLAS-4 Satellite Laser Ranging (SLR) system, which has operated from this location since 1981, along with a GNSS antenna and an EarthScope seismic station.

Commercial and Public Safety: There are multiple microwave relay dishes and panel antennas visible in the HPWREN imagery on surviving structures. Historical records show San Diego County Sheriff's Office VHF low-band repeater infrastructure at the site dating to the 1960s.

According to sources familiar with the site, the only functional equipment on the collapsed ATC tower was the AT&T cell site and its microwave backhaul links. The full inventory of what had previously been on the structure versus what was still active is not entirely clear, but the tower appears to have been underutilized at the time of its failure.

What We Know and What We Don't

The video from the HPWREN cameras answers the biggest question. When did it fall? At 10:00 AM on February 18, during the second and most intense storm wave. Sources who have viewed the time-lapse describe it as showing a clear wind event immediately before the collapse. As is often the case with periodic camera captures of structural failures, the tower is there one frame and gone the next.

What was the failure mode? The damage photos show the concrete foundation pier uprooted from the ground rather than the tower folding at a structural joint. Whether ice loading, sustained wind, a gust event, or a combination caused the failure is unknown. No formal engineering assessment has been publicly released, but commenters seem surprised about the relatively small amount of concrete that was pulled up.

What services are currently offline? The ECRA repeaters and HPWREN systems appear to have survived on other structures. The primary loss appears to be AT&T cellular coverage and microwave backhaul from this site. Operators in the coverage area, particularly in eastern San Diego County and the Imperial Valley, may have noticed cellular outages.

What Happens Next

The central question is whether American Tower Corporation will rebuild. As one source familiar with the site put it (Chris KF6AJM), the only functional thing on the tower was the AT&T cell site with a few microwave hops. Whether that single-tenant revenue justifies the cost of constructing a new tower at a remote mountaintop location in a national forest, with all the permitting, environmental review, and logistics that entails, remains to be seen. It is not clear if that is profitable enough for ATC to

put money into the site.

Mountaintop tower sites in places like the Cleveland National Forest are expensive to build and maintain. Access roads can be difficult in winter. Construction requires Forest Service approval. And the economics of a single-carrier site are thin compared to a multi-tenant tower in an urban area. There has been a long-term trend away from using large mountaintop towers, with capacity replaced by fiber backhaul and fixed wireless broadband. The reason for this is that wide coverage is now less valuable than capacity per user. Higher data rates per commercial cellular user cannot be delivered by one large site covering a large land mass as easily and cheaply as can be delivered with more sites all closer to the ground.

On the other hand, Monument Peak provides cellular coverage to areas of eastern San Diego County and the Imperial Valley that are otherwise difficult to serve. The microwave hops that ran through this tower may also have been part of a backhaul chain serving other sites. The downstream effects of this loss on AT&T's network in the region are not yet clear.

Monument Peak has weathered storms before. HPWREN documented significant wind damage at their Big Black Mountain relay site in January 2018 during a Santa Ana event with 80-90 mph winds, and the HPWREN team rebuilt that site with an improved design to better withstand future weather. A similar assessment and improved rebuild process will likely be needed here for the commercial tower that fell if the economics support it.

For a site that serves as a critical node in the region's wildfire detection network, amateur radio infrastructure, scientific instrumentation, and commercial communications, the loss of even one tower could have cascading effects. The coming weeks will tell us more about the extent of the damage, any additional as-yet uniscovered damage on other towers, and the timeline for restoration.

If you have additional information about this event, particularly regarding which services are affected or the path forward for restoration, please contact us at ORI.

Photo Credits and Sources

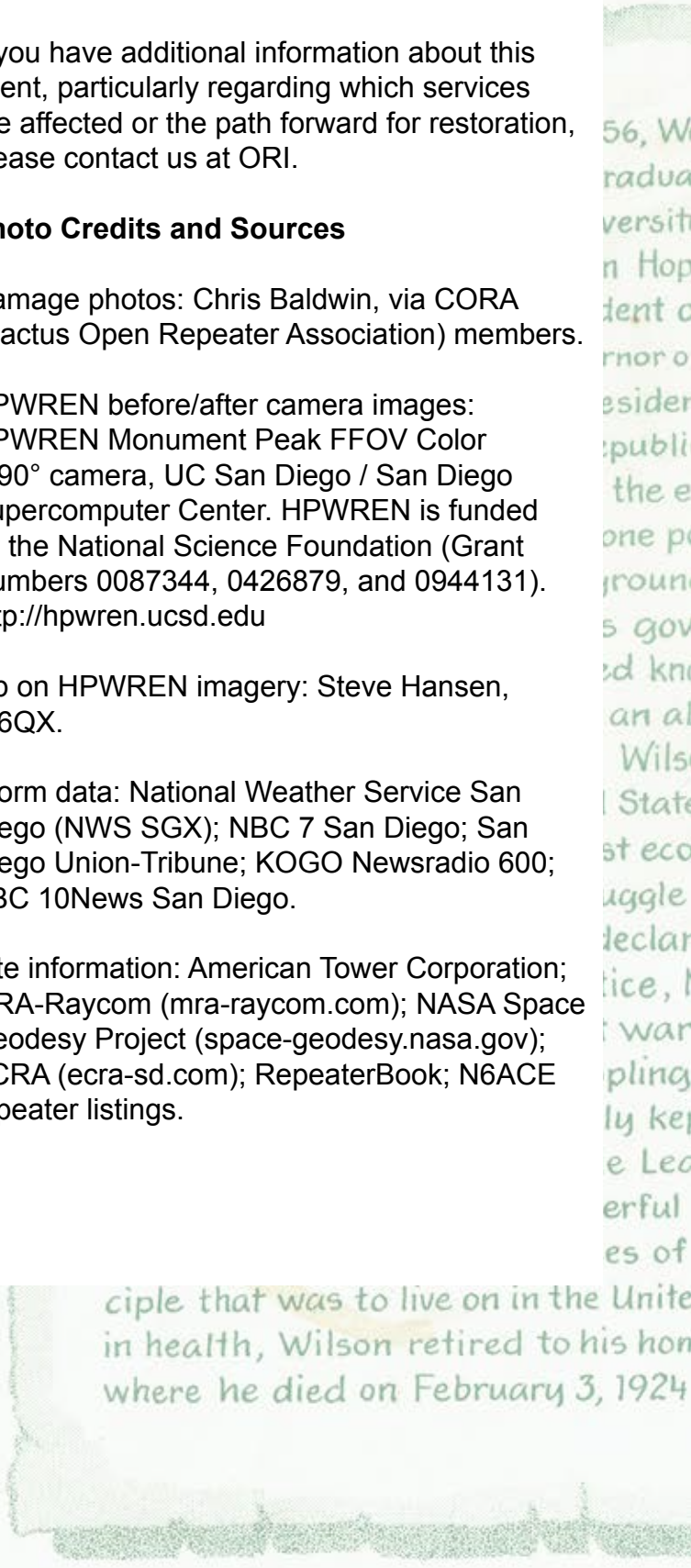
Damage photos: Chris Baldwin, via CORA (Cactus Open Repeater Association) members.

HPWREN before/after camera images: HPWREN Monument Peak FFOV Color E 90° camera, UC San Diego / San Diego Supercomputer Center. HPWREN is funded by the National Science Foundation (Grant Numbers 0087344, 0426879, and 0944131). http://hpwren.ucsd.edu

Tip on HPWREN imagery: Steve Hansen, W6QX.

Storm data: National Weather Service San Diego (NWS SGX); NBC 7 San Diego; San Diego Union-Tribune; KOGO Newsradio 600; ABC 10News San Diego.

Site information: American Tower Corporation; MRA-Raycom (mra-raycom.com); NASA Space Geodesy Project (space-geodesy.nasa.gov); ECRA (ecra-sd.com); RepeaterBook; N6ACE repeater listings.




HPWREN - Monument Peak East - http://hpwren.ucsd.edu 1771005649.510: PST
UC San Diego
PANO
Monument Peak FFOV Color E 90°
32.89°N, 116.42°W, 1910m/6266ft | 31.1 °F, 90.2 %RH, 23.5 inHg
HPWREN


HPWREN - Monument Peak East - http://hpwren.ucsd.edu 1771447909.036: PST
2026-02-18 12:51
UC San Diego
PANO
HPWREN

ORI Invited to Present Open Source Reference Design for IEEE P1954 UAV Communications Standard

Open Research Institute has been invited to present at the IEEE P1954 working group meeting on April 8th. Our topic: how to build an open source reference implementation for the emerging standard on self-organizing, spectrum-agile UAV communications.

What is IEEE P1954?

IEEE P1954 defines architecture and protocols that allow unmanned aerial vehicles to automatically form networks, dynamically access available spectrum, and coordinate communications without centralized infrastructure. Think of it as giving drones the ability to self-organize into mesh networks while intelligently sharing radio spectrum. These are critical capabilities for search and rescue, disaster response, infrastructure inspection, and beyond.

The standard is deliberately technology-agnostic. It specifies what UAV communication systems need to do, not how to build them. That's where reference implementations come in.

Why Open Source Matters Here

Standards without working implementations remain academic exercises. An open source reference design serves multiple purposes

Experimentation platform: Researchers and developers can test ideas against a working baseline
Conformance validation: Implementers can verify their systems behave correctly
Lowered barriers: Smaller players can participate without building everything from scratch
Vendor neutrality: No single company controls the reference, aligning with the standard's technology-agnostic philosophy

What ORI Brings to the Table

ORI's existing work maps remarkably well onto P1954's architecture. The standard envisions two distinct communication tiers:

Command & Control (C2): Safety-critical links requiring high reliability, low latency, and modest data rates
Payload: High-throughput channels for video and sensor data where best-effort delivery is acceptable

Our Opulent Voice protocol (MSK/CPFSK, constant envelope, narrowband) is designed for exactly the reliability-first requirements of C2 links. Our Neptune OFDM work addresses the high-throughput payload tier. Both have FPGA implementations in progress.

The standard also includes a SHALL-level requirement that UAVs "embed radio equipment such as software defined radios". This is precisely our domain.

The Path Forward

We're proposing to bring implementable chunks of P1954 into ORI repositories as open source FPGA and general-purpose processor designs. This isn't about implementing the entire standard overnight. It's about identifying the pieces most amenable to open source development and building momentum from there.

The April 8th meeting is our opportunity to discuss this approach with the working group and align our efforts with their priorities.

Get Involved

If you're interested then this is an opportunity to contribute to an emerging international standard from the ground floor. Watch for updates on our mailing lists and repositories.

Where is this Code From?

```
rocolcolate_eeor(udsr_rtioihngs):
>> adds dksl jkd and lfr, djlrfr itle to sfjlsbrn acisott tgjk dgjc

a lode [red square] 40
[white square]    lode_noz("sodece_shet_fejke.squ")
[white square]_ueres, n_miwfaos [red square] m,anpes

ratidgfgs [red square] (a lpha fgr:tr(rayhge(tsvg(ufery_rytuidges)))

[white square] .dahe [red square] ro.hstack[(n.dsta, rhshyyuk)]
[white square] .indichgrgs [red square] ro.hstack[(n.intaksc, usfe(s.
dahfy))]
[white square] .indptr [red square] rp.hstack[(.n.indptr, leu(a.dagy))]
[white square] ._shrper [red square] (n_ufgt [red square] l, n_mvioty)

erecornnshld N teoq ts nvg shuo
wicu opesg("mochyr.sxh", "rb") sj ptckish_in:
```

What Does This Code Do?

Send your answer to abraxas3d@openresearch.institute

The pink color in the text above is a hint. R=224, G=33, B=138.

Answers will be in the April Inner Circle Newsletter from Open Research Institute.

ORI now has its very own amateur radio club call. Membership in ORI's amateur radio club is free. Just sign up for the Inner Circle Newsletter. Below is the poster announcing the debut of W6ORI. The announcement was made in RF Village at BSides San Diego, held at SDSU Montezuma Hall on 4 April 2026.

Lunar Descent, the BSides San Diego 2026 RF Village Capture the Flag (CTF) from ORI

A capture-the-flag challenge based on a real signal processing problem in a radar altimeter!

https://github.com/OpenResearchInstitute/lunar-descent-ctf

Indian Space Research Organization (ISRO) designed a Ka band radar altimeter (KaRA) that guided Chandrayaan-3 to a soft lunar landing on 23 August 2023. The Radar Altimeter Processor (RAP) computes altitude and velocity from FMCW chirp signals, running on a single Xilinx Virtex-5 FPGA. This CTF uses a Python model of that system, faithful to the published paper in the Aeronautics and Electronic Systems Journal, where the altimeter feeds a landing autopilot. In our CTF, the altimeter works perfectly. The autopilot keeps crashing. Why? (solution in next newsletter!)

What did the participants see? A python script that could be installed on their computer and then run.

```
pip install numpy matplotlib
python lunar_descent_ctf.py --help        # See all options
python lunar_descent_ctf.py               # Run the mission, watch it crash
python lunar_descent_ctf.py --modes       # See the sweep mode table
python lunar_descent_ctf.py --test -p all # Test all three profiles
python lunar_descent_ctf.py --score       # Score your fix and earn flags
```

Rules

Edit ONLY the `MeasurementQualifier` class (clearly marked in the source)
Don't change the RAP, signal generation, autopilot, or scoring
The qualifier decides what the autopilot sees — fix it there
Submit flags at the RF Village table

Three Flags

None of the flags are "free". The buggy code scores 0 / 1000 points out of the box.

Flag	Points	Challenge
RECON	100	Explain the bug to RF Village staff. No hash on screen.
FIRST LIGHT	500	Land all three profiles without crashing.
NO GAPS	400	Zero qualifier rejections on all three profiles.

Total: 1000 points

The Scenario

The radar altimeter was tested on helicopters and aircraft at altitudes above 50 meters. It worked flawlessly. Field test performance met all mission specifications.

The altimeter is now integrated with a landing autopilot that uses both altitude and velocity measurements for thrust control during final approach. In simulation, the autopilot crashes the lander

every time below 15 meters altitude. The altitude readings are fine. Sub-meter accuracy all the way to touchdown. Something else is killing the lander. You need to find out what's going wrong and fix the measurement qualification logic so the autopilot can land safely.

Difficulty Curve

0 points: Running the code unmodified. The default run shows OK status all the way down until the final approach, then CRASH.
100 points: Explaining the problem to staff.
600 points: Fixing the `MeasurementQualifier` so it can land.
1000 points: Eliminating all qualifier rejections.

Validating Flag 1 (RECON)
No hash is printed on screen for Flag 1. Staff issue the flag manually.

Timing

The CTF ran all day alongside the workshop modules and talks. It's self-paced and doesn't require staff attention except for Flag 1 validation and prize distribution.

Test Profiles

Profile	Character	What It Tests
standard	Chandrayaan-3-like smooth descent, 10 km → 3 m, with altitude excursion at 20 m (thruster anomaly or drifting over crater)	Landing
aggressive	Fast exponential braking, 10 km → 3 m in 400 s	Rapid mode transitions at hight altitude + Landing
stepwise	Hover at guard band boundaries (9851/4795/2334/553/131/31/5 m), drop between them	Mode transitions + Low Hover

The Physics

The RAP uses FMCW radar. Up-chirp and down-chirp signals produce beat frequencies:

```
f_up = fb − fd (up-chirp)
f_dn = fb + fd (down-chirp)
```

Altitude comes from the sum: R = M × (f_up_index + f_dn_index).

Velocity comes from the difference: fd = (f_dn_index − f_up_index) × freq_res / 2.

The FFT is always 8192 points, but the number of real signal samples depends on the sweep time:
Mode 12 (high altitude): 8192 samples, full FFT
Mode 0 (3 m altitude): 14 samples, 99.8% zero-padding

Connection to Real Engineering

The problem is pedagogically framed but the pattern is real. Sensor qualification, knowing when to trust a measurement and when to reject it, is important.

- Radar and sonar tracking systems (Doppler reliability vs integration time)
- GPS/INS integration (knowing when satellite geometry is too poor to trust)
- Medical imaging (SNR-dependent confidence in measurements)
- Autonomous vehicle sensor fusion (camera vs lidar vs radar confidence)

The paper mentions “three sample qualification logics to generate the final altitude” without detailing them. What are those logics? Will some of those qualification logics help solve this CTF?

Source

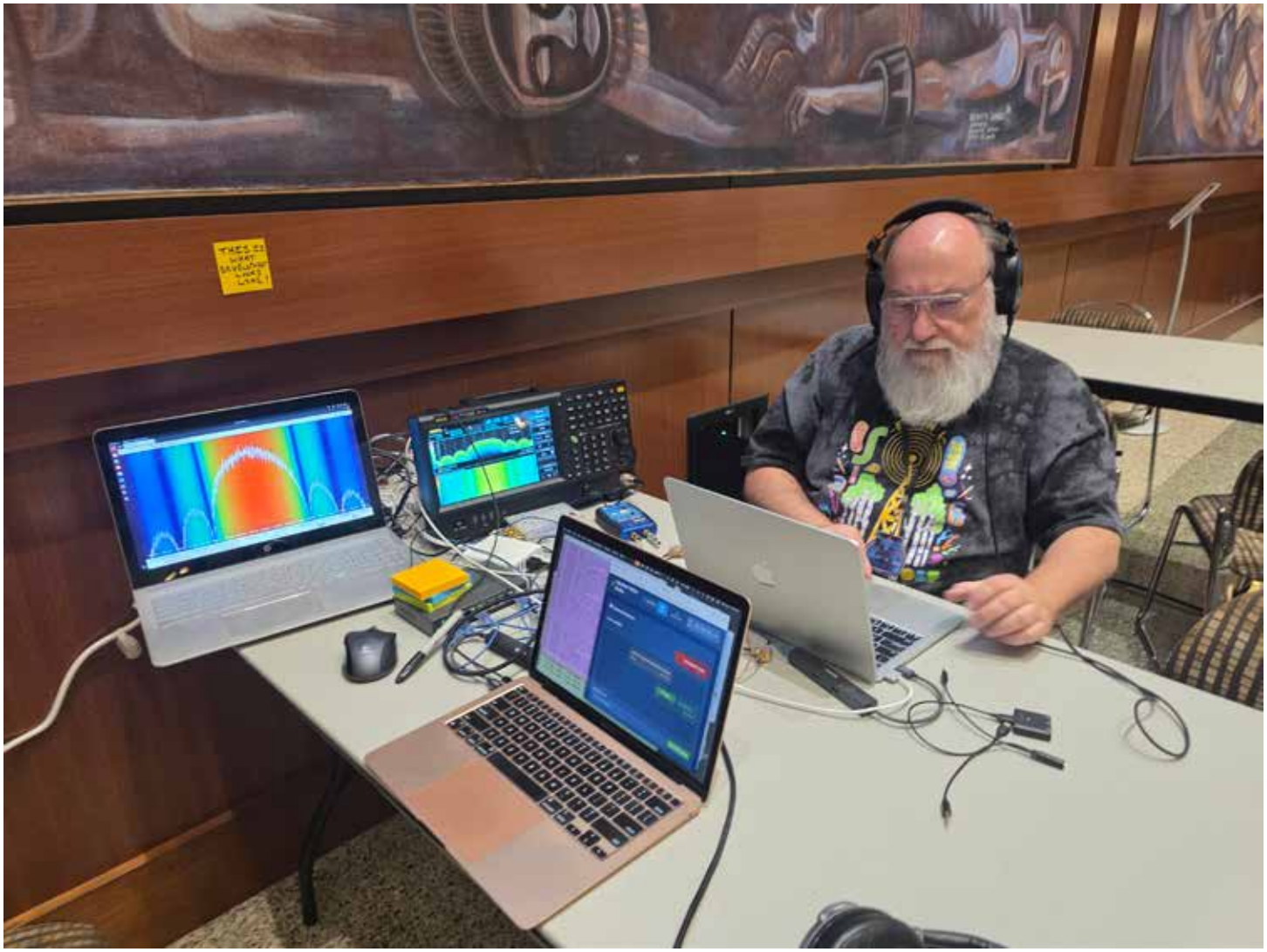
Based on: Sharma et al., “FPGA Implementation of a Hardware-Optimized Autonomous Real-Time Radar Altimeter Processor for Interplanetary Landing Missions,” IEEE A&E Systems Magazine, Vol. 41, No. 1, January 2026. DOI: 10.1109/MAES.2025.3595090

Poster created for Bsides San Diego RF Village

BSides San Diego 2026 RF Village Demonstrations

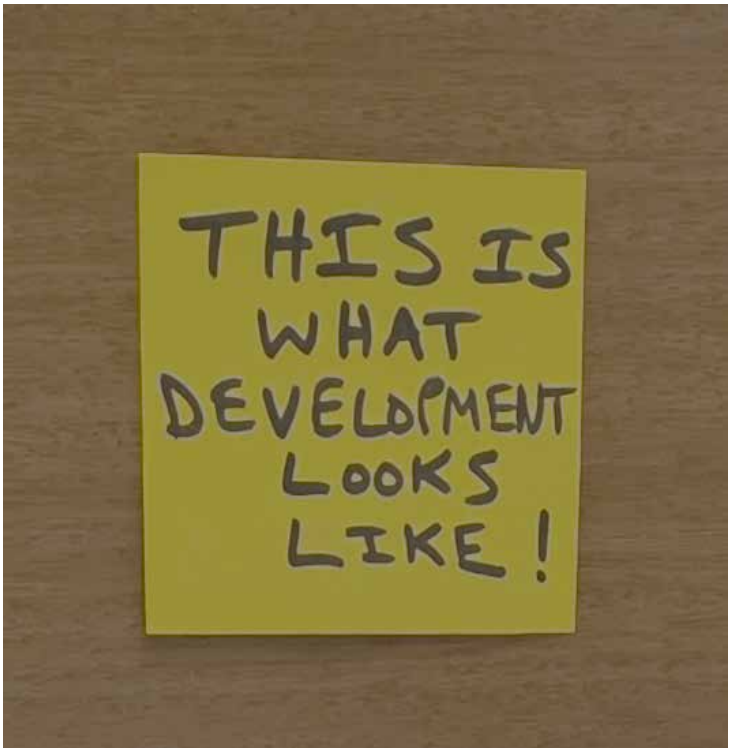
This is what development looks like!

Open Research Institute organized and executed the RF Village at BSides San Diego on 4 April 2026. This highly anticipated sold-out annual event has a focus on cybersecurity and DIY problem solving. Held at Montezuma Hall at San Diego State University, the one-day event had multiple speaking tracks, at least three Capture the Flag contests (CTFs), an electronic hackable badge, a variety of food and drink available throughout the day, extensive volunteer support, relevant and timely workshops, an After Hours party with more food and drink at Aztec Lanes bowling alley, and a vibrant Village Square that combined Villages and Vendors.



BSides San Diego RF Village Opulent Voice development station with Paul Williamson KB5MU.

ORI served as the organizer for RF Village, bringing four staff members and multiple exhibits. We debuted our Lunar Lander CTF, announced the W6ORI amateur radio club, and gave away a large box of Amateur Satellite handbooks. We had RFBitBanger kits available for donation (5 were sold), hosted a live Meshcore node, and exhibited a real live FPGA development station with lab equipment for Opulent Voice. Our poster session included Authentication and Authorization and the technical



Ephemeral signage in RF Village with hand-written "This is what development looks like!" in sharpie on yellow post-it note.

side of RFBitBanger. Thank you to BSides San Diego for the excellent support, with rotating village volunteer staffers, volunteer green room, excellent communications before during and after the event, and genuine care for positive participant experience. Organizations like this make demonstrating open source digital radio work a real joy instead of a daunting chore.

Demonstrations give you deadlines, documentations, and get things "done". The BSides Opulent Voice demonstration revealed some immediate problems with the 24 dB transmitter fix. The signal was clearly being transmitted at sufficient power, but the symbol and frame lock were not happening. We were seeing "garbage" frames where we were expecting to see actual live data.

Since the over-the-air voice call had worked so well just a few days prior, what was going on? The demonstration was still very successful, as plenty could be shown to the steady stream of people at the RF Village. As the day progressed, more information was gathered. It was very clear we'd have to go back to the lab to figure out how a badly needed transmitter fix had broken the receiver. Why was it working over the air, and not in RF loopback on the bench? First, we went back to the VHDL-only test bench. This sends 10 frames into the transmitter, routes the transmit I and Q streams right back to the receiver, and then demodulates, decodes, and displays the frames. The data in should match the data out. And, frames were coming out, but they were completely scrambled! Something had gone wrong in the RF loopback.

The difference between simulations and real life is usually a lot, and Opulent Voice is no different. A real hardware RF loopback has the radio chip in the loop. The VHDL test bench has only the FPGA contents. We don't have analog to digital converters (ADCs), digital to analog converters (DACs), or anything else that is in the radio chip. Since the transmitter fix had a lot to do with how the transmitter DAC dealt with the data, it seemed reasonable to assume that the transmitter fix had upset the receiver.

The missing gain was in the transmitter, but the fix applied some math changes to both the transmiter and the receiver, and investigating this took up part of the next day back in the lab. WIth two minor changes, the test bench started working flawlessly again. A new version of the firmware was created, and... it didn't work in hardware at all! Symbol lock and frame lock were totally non-functional. The plot had certainly thickened.

This is one of the many reasons why demonstrations are so valuable. We find things that we might not go looking for, and it forces us to regularly show things working end-to-end. Work continues this week in the lab to separate the transmitter fix from inadvertently affecting the receiver, and bring us back to working perfectly over the air as well as in simulation.