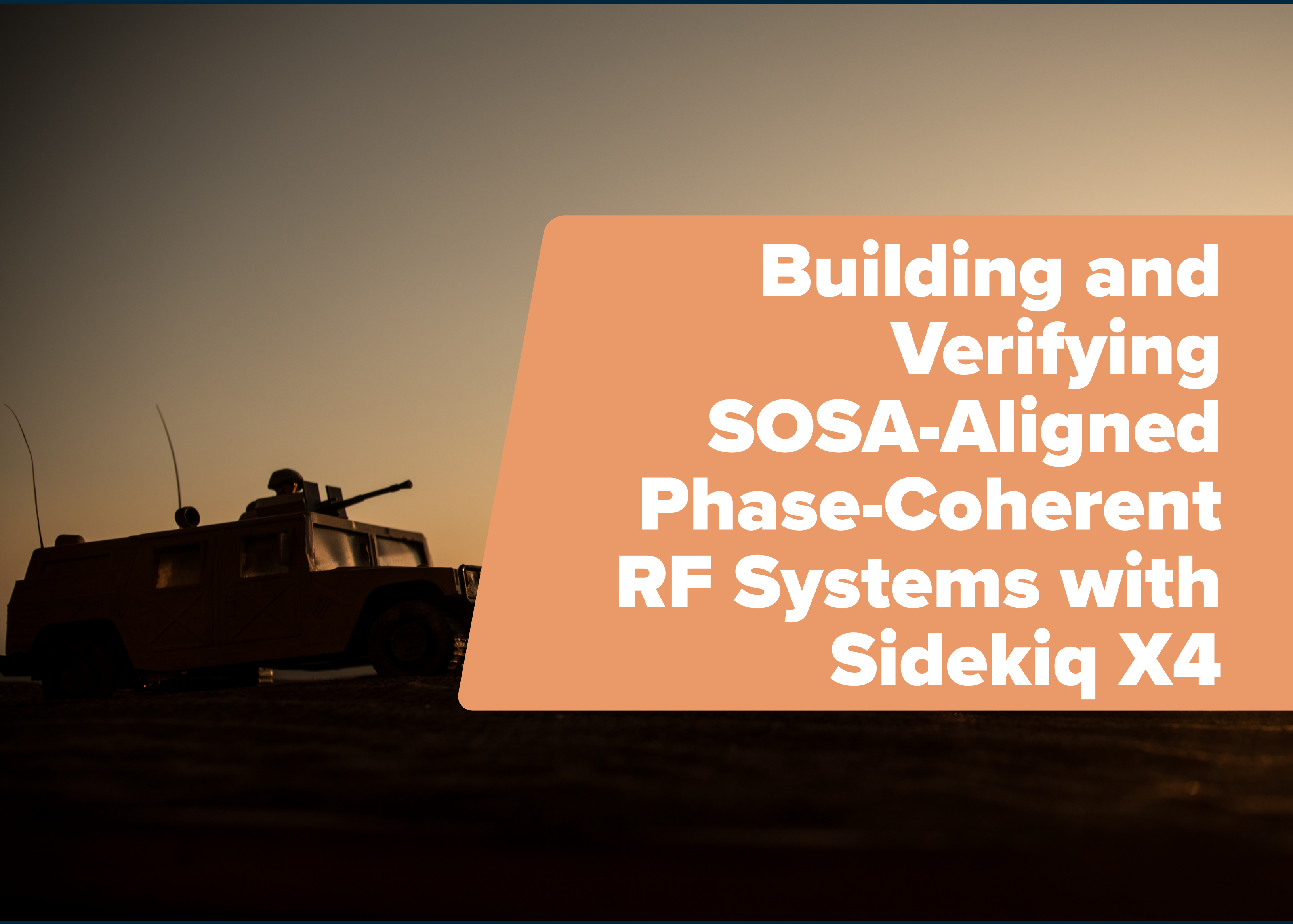


# Building and Verifying SOSA-Aligned Phase-Coherent RF Systems with Sidekiq X4

EPIQ SOLUTIONS

# Table Of Contents

## What is Phase-Coherent Operation and Why is it Important?

Our focus is on developing tools that provide situational awareness and detailed insight into RF environments so you can identify and take action against wireless threats.

At Epiq Solutions, our focus is on developing tools that provide situational awareness and detailed insight into RF environments so you can identify and take action against wireless threats. For these applications, it's often not good enough to just identify RF signals of interest; we want to know where they're coming from as well. Ideally, a line of bearing (LOB) can be provided to indicate the direction of an RF signal of interest relative to the receiver. Multiple techniques exist to provide a LOB, and phase-coherent RF reception is a key capability to support this use case.

Phase-coherent operation is the ability of an RF system's receive and/or transmit channels to be both time- and phase-aligned, where the signals have a defined and constant relative phase relationship. Applications including MIMO, beamforming, and direction finding require this alignment as well as a known phase relationship between each RF input or output. Typically, a multichannel software defined radio (SDR) transceiver is leveraged to provide time- and phase-aligned digital radio sample streams which can then be processed by a host system to extract information such as angle of arrival of an RF signal of interest.

In this whitepaper, we'll explore the phase coherence capabilities

> In this whitepaper, we'll explore the phase coherence capabilities of Epiq Solutions' Sidekiq X4, and how to measure this phase coherence using software applications delivered as part of the Sidekiq X4 Platform Development Kit.

of Epiq Solutions' Sidekiq X4 SDR card, and how to measure this phase coherence using software applications delivered as part of the Sidekiq X4 Platform Development Kit. Additionally, the reader will understand what is required to integrate a Sidekiq X4 into a real-world system, along with the calibration required to correct for real-world impairments such as cable loss to the antenna.

## How Phase-Coherent Operation is Commonly Achieved Today

Typical solutions for phase-coherent operation utilize one of two different mechanisms to derive coherence, as shown in Figure 1. First, an external RF transmitter can be utilized to emit an RF calibration signal that is then fed into the multi-channel receiver during operation of the system to measure and correct for phase ambiguity across the multiple receivers. While effective, this requires a transmitter to be integrated into the system. Additionally, generating an RF calibration signal and feeding it into all of your receivers may be required every time you change RF frequencies if phase coherence can't be guaranteed between tuning operations. So, if you're monitoring an RF signal at 2.4 GHz and then want to process a signal at 900 MHz, you would have to repeat the entire phase calibration process. This is time consuming and it requires transmitting a signal in situations where transmitting an RF signal may be the last thing you want to do.

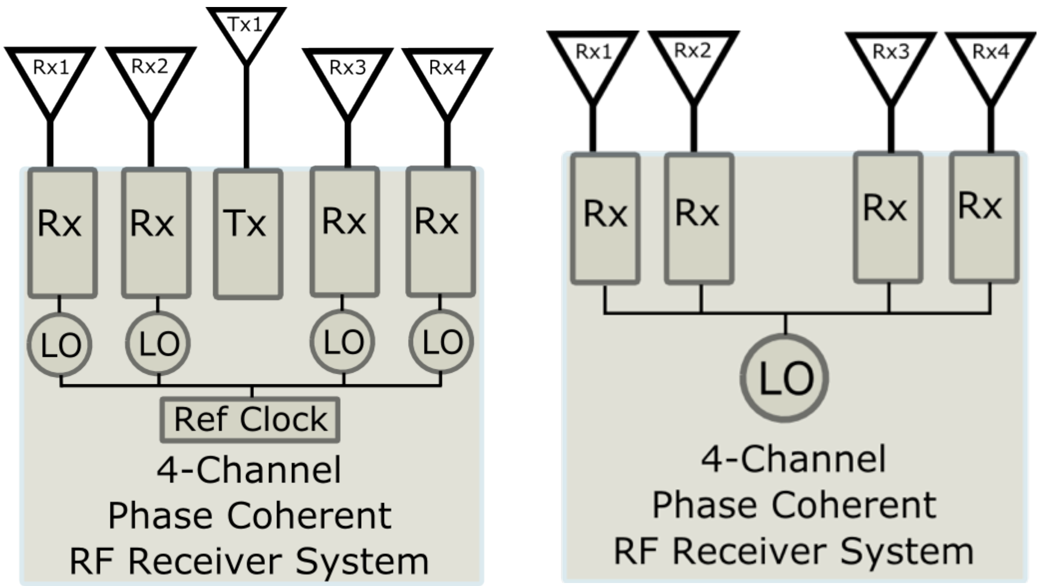



*Figure 1: Two common schemes are used today for achieving RF receiver phase coherence: use of an RF transmit calibration tone (on the left) or use of a common RF LO signal (on the right)*

Alternatively, a common local oscillator (LO) signal can be distributed to all of the RF receivers, thus ensuring that all receivers are phase-aligned. This, too, has a number of challenges, including the fact that it becomes very difficult to distribute a high fidelity, low phase noise LO signal to multiple receivers at the frequencies of interest (often up to 6 GHz or even higher in some cases). In cases where an externally distributed LO is being used with a quadrature down-conversion RF receiver, it is often necessary to generate and distribute an LO signal that is 2x the desired RF frequency of operation due to the way the quadrature LO signals are generated inside the RF receiver. Thus, operating up to 6 GHz would require the generation and distribution of a 12 GHz LO signal. And even if phase alignment is achieved at RF, this is just part of the solution. Synchronized baseband sampling in the analog-to-digital (ADC) converter is also required to ensure phase coherence.

A common local oscillator (LO) signal can be distributed to all of the RF receivers, thus ensuring that all receivers are phase-aligned.

**The Sidekiq X4 SDR leverages two ADRV9009 RFICs from Analog Devices to provide a VITA 57.1-compliant FPGA Mezzanine Card (FMC), providing four channels of phase-coherent receive and transmit capability.**


*Figure 2: The Sidekiq X4 FMC card supporting 4-Rx + 4-Tx, and a deployment option in a 3U VPX carrier*

## Multi-Channel Phase-Coherent Rx with the Sidekiq X4

The Sidekiq X4 SDR leverages two ADRV9009 RFICs from Analog Devices to provide a VITA 57.1-compliant FPGA Mezzanine Card (FMC), providing four channels of phase-coherent receive and transmit capability that can be deployed in any host system supporting FMC cards (including 3U/6U VPX platforms that are aligned to SOSA or CMOSS, or PCIe carrier cards for deployment in rack mounted servers, as shown in Figure 2). Epiq Solutions' libsidekiq software library provides a common software API across its entire portfolio of SDRs, with API extensions to support phase-coherent operation on Sidekiq X4. The libsidekiq API leverages a novel RF PLL synchronization capability available in the ADRV9009 RFICs to phase align one or more LO sources in a multi-ADRV9009 system such as Sidekiq X4. The Sidekiq X4 FPGA reference design is also utilized to complete the phase-coherent system, ensuring synchronized baseband sampling and providing timestamp management across all four receive or transmit channels to allow for proper alignment between sample streams.

The remaining sections of this white paper will focus on the following topics:

- System Configuration to Prepare for Phase-Coherent Reception
- Acquiring Phase-Coherent Samples
- Processing and Validation of Phase-Coherent Samples in a Test System

## System Configuration to Prepare for Phase-Coherent Reception

A typical Sidekiq X4-based system is shown in Figure 3. This system consists of several key hardware elements.

A typical Sidekiq X4-based system is shown in Figure 3. This system consists of several key hardware elements.

- **RF Signal Source:** This is responsible for generating our RF signal of interest.
- **Four-Channel Antenna Array:** This is the antenna that provides access to the RF signal, along with phase matched cables to deliver the four RF signals of interest to the receiver.
- **Sidekiq X4 Installed in an FPGA-based PCIe Carrier Card** (for lab evaluation, the Sidekiq X4 card + FPGA-based PCIe carrier card is delivered pre-installed in a Thunderbolt 3 chassis for ease of use): This hardware element is responsible for converting the fours RF signals to baseband, digitizing them, and streaming them via PCIe through a Thunderbolt 3 interface to a host computer.
- **Host Linux Computer Running Libsidekiq:** This hardware element is responsible for running a software application to configure Sidekiq X4 for phase-coherent operation, and receiving the four channels of timestamped digitized I/Q for post processing.

**In lieu of the four-element antenna described above, a common practice for lab evaluation and testing is to use an RF splitter to convert the RF signal source into four instances.**

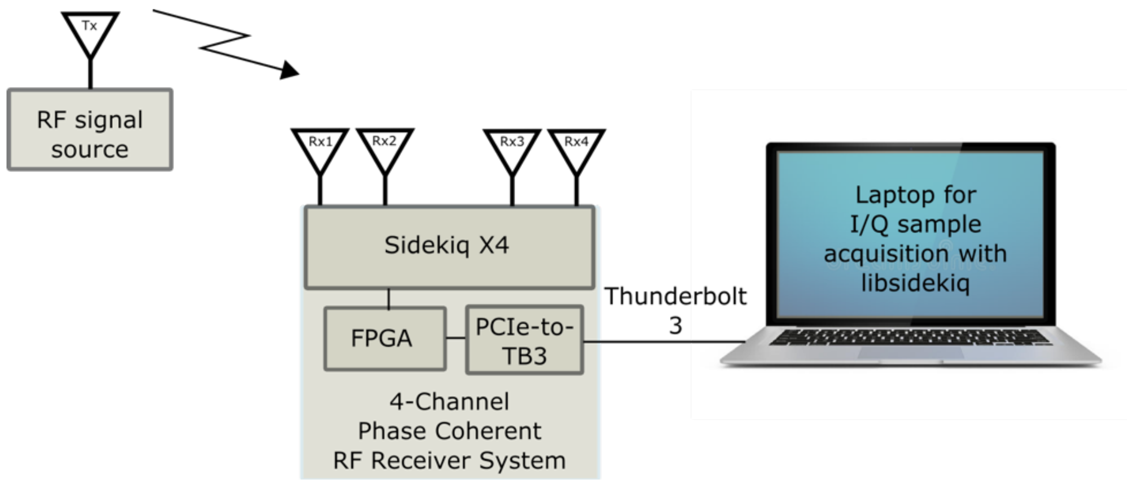



*Figure 3: Typical hardware configuration for operating a Sidekiq X4-based phase-coherent system*

**Note**: In lieu of the four-element antenna described above, a common practice for lab evaluation and testing is to use an RF splitter to convert the RF signal source into four instances. With the use of phase matched cables, it is possible to deliver the RF signal of interest into Sidekiq X4 in a cabled environment for controlled evaluation to ensure no ambient RF impairments interfere with the system. This configuration is shown in Figure 4 below.

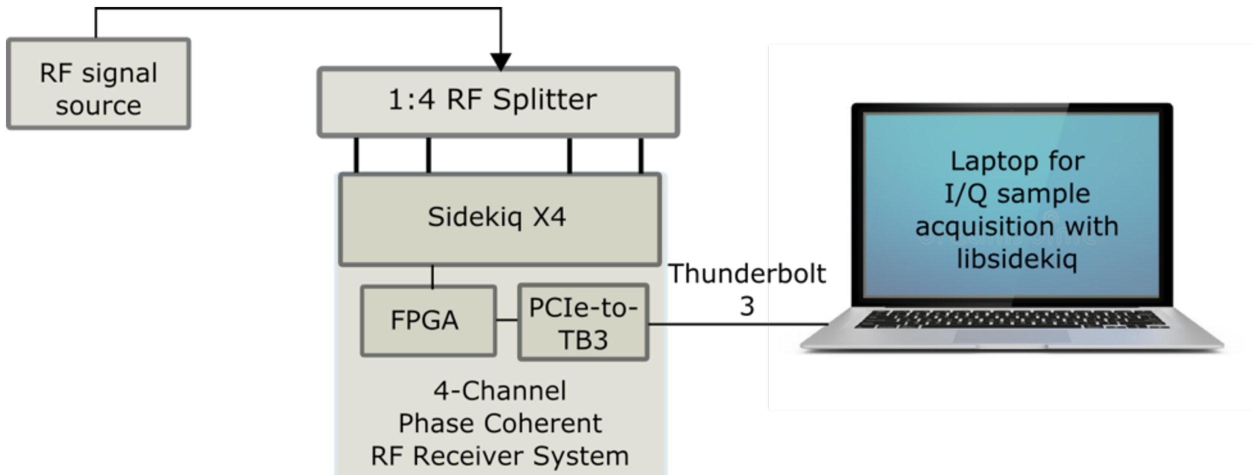



*Figure 4: Alternate hardware configuration for operating a Sidekiq X4-based phase-coherent system*

## Configuring the System and Acquiring Phase-Coherent Samples

> With the system connected, the next step is to configure Sidekiq X4 to perform phase-coherent reception at the RF frequency of interest, and stream the digitized I/Q samples up to the host laptop for verification of phase coherence.

With the system connected as shown in Figure 4, the next step is to configure Sidekiq X4 to perform phase-coherent reception at the RF frequency of interest and stream the digitized I/Q samples up to the host laptop for verification of phase coherence. The Sidekiq X4 Platform Development Kit (PDK) includes the C source code for more than a dozen test applications demonstrating usage of the libsidekiq API. This includes an application called **`rx_samples_on_trigger`**, which can be used to capture phase-coherent I/Q samples across all four receivers (commonly referred to as channels A1, A2, B1, and B2) and store them to a file for post-processing.

**The C source code for this application demonstrates all of the steps required to perform the following operation:**

- Initialize the Sidekiq X4 card
- Configure the card for phase-coherent operation
- Tune the four RF receiver channels to an RF center frequency of interest
- Set the sample rate and channel bandwidth of interest for all four RF channels
- Set the RF gain for all four RF channels to a user-defined level
- Start the streaming operation for all four RF channels up to the host laptop
- Acquire timestamped packets of I/Q samples across all four RF channels (A1, A2, B1, and B2)
- Align the timestamped packets of I/Q samples across all four RF channels
- Stop the streaming operation
- Store the four streams of aligned timestamped I/Q samples to a file
- Shutdown the Sidekiq X4 card
- Exit the application

A typical command line for executing the `rx_samples_on_trigger` application on the host Linux computer is shown below. This configures the first Sidekiq X4 card in the system (card 0) to an RF center frequency of 2.5 GHz, a baseband sample rate of 122.88 Msamples/sec, and an RF channel bandwidth of 100 MHz. It captures digitized I/Q across all four channel handles (A1, A2, B1, and B2), recording for a time period of one second, and storing the time-aligned phase-coherent I/Q files to the directory /tmp/phase_coherent_captures/ with a file prefix of my_file_prefix-:

```
sidekiq@ubuntu:~$ ./rx_samples_on_trigger
--card=0 --frequency=2.5e9 --rate=122.88e6
--bandwidth=100e6 --handle=ALL --time=1
--trigger-src=synced --destination=/tmp/
phase_coherent_captures/my_file_prefix-
```

The libsidekiq API and FPGA reference design take care of ensuring that the four RF receivers are operating phase-coherently, and that baseband sampling happens synchronously across all four channels.

The libsidekiq API and FPGA reference design take care of ensuring that the four RF receivers are operating phase-coherently, and that baseband sampling happens synchronously across all four channels. Each packet of I/Q samples received in the system includes a metadata header with a 64-bit system timestamp value that is inserted by the FPGA. This timestamp corresponds to the time when the first sample in an I/Q packet was received by the system, and serves as a common timebase across all four RF receive channels (A1, A2, B2, and B2). Using these timestamps, a user application receiving I/Q packets from all four RF receivers can align the I/Q samples in the packets to ensure their processing application is operating on phase-coherent samples that were acquired precisely at the same time. A diagram showing the typical I/Q packet header for each received on the host system via libsidekiq is shown on the next page.

**It is imperative to use the timestamps embedded in the metadata of each I/Q packet to actually know which samples are phase-coherent with each other.**

| Field | Width (Bits) | Description |
|---|---|---|
| rf_timestamp | 64 | RF timestamp associated with the received sample block |
| sys_timestamp | 64 | System timestamp associated with the received sample block |
| hdl | 6 | Receive handle indicating the receive handle associated with the received sample block |
| overload | 1 | RF Overload indicating whether or not the RF input was overloaded for the received sample block |
| rfic_control | 8 | RFIC control word carrier metadata from the RFIC, typically the receive gain index |
| id | 8 | Channel ID used by channelizer (currently unused) |
| system_meta | 6 | System metadata (unused / reserved) |
| version | 3 | Packet version field |

*Figure 5: Standard metadata header accompanying all I/Q sample packets from libsidekiq*

It should be noted that during the initiation of the streaming operation, it is possible to begin receiving I/Q packets for one channel slightly ahead of another while all four channels are starting up. However, both the RF hardware and the baseband A/D conversion process are operating coherently. It is imperative to use the timestamps embedded in the metadata of each I/Q packet to actually know which samples are phase-coherent with each other. Figure 6 below illustrates how the Sidekiq platform captures blocks of RX data from multiple channels and performs the alignment within the `rx_samples_on_trigger` test application. At time ts_a1[0], the application begins capturing samples from channel A1. A short time after that, at ts_a2[0], the same process begins for channel A2, then for channel B1, and finally channel B2. The example test application stops capturing samples from all channels when the requested number of bytes per channel have been captured (determined based on the requested runtime duration and the requested sample rate). Therefore, there is a portion of samples in the beginning of the A1, A2, and B1 captures and the end of the A2, B1, and B2 captures that are discarded. Only the samples which were captured simultaneously across all four channels will be used for the purposes of verifying phase coherence.

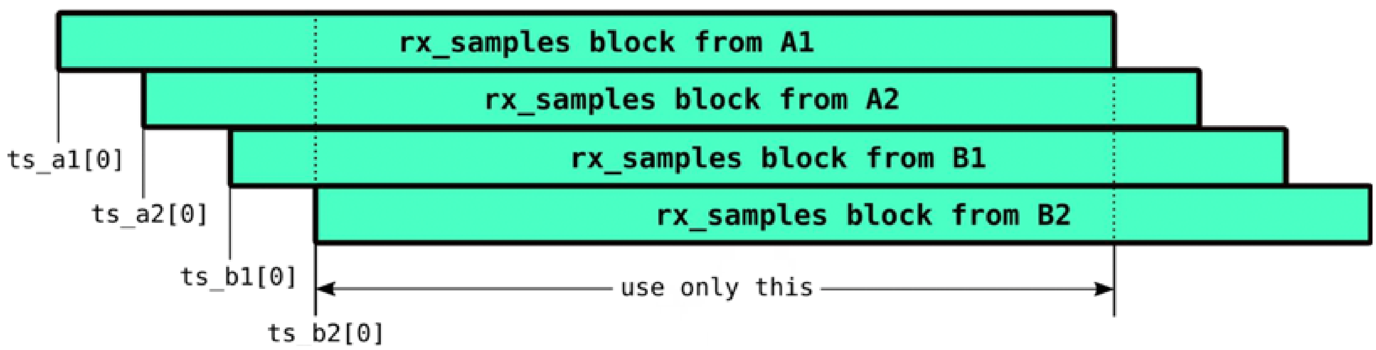



*Figure 6: Alignment of samples across all four Rx channels based on timestamp*

## Measuring Phase Coherence

In order to measure phase coherence, a well-defined test setup is required to ensure proper measurements can be captured for post-processing validation of coherence.

In order to measure phase coherence, a well-defined test setup is required to ensure proper measurements can be captured for post-processing validation of coherence. One method for quantitatively measuring the phase/time difference between channels is to use a Gaussian noise source as the RF signal source stimulus and then correlating the samples at baseband. Gaussian noise is useful because it:

- Excites all frequencies across the capture bandwidth of the receiver
- Is easily available without the need for expensive equipment
- Correlates as a Dirac delta function (δ function)

The phase of the correlation function is a direct measurement of the phase difference between any two given channels. The correlation can then be used to quantitatively determine the phase difference between the two channels at the given RX LO frequency and sample rate and store these calibration values in a calibration look up table.

As shown in Figure 7 below, the recommended test setup to measure phase coherence replaces the RF signal source with a broadband amplified RF noise source. The noise source must be high enough to overcome the noise floor of the Sidekiq X4 receiver.

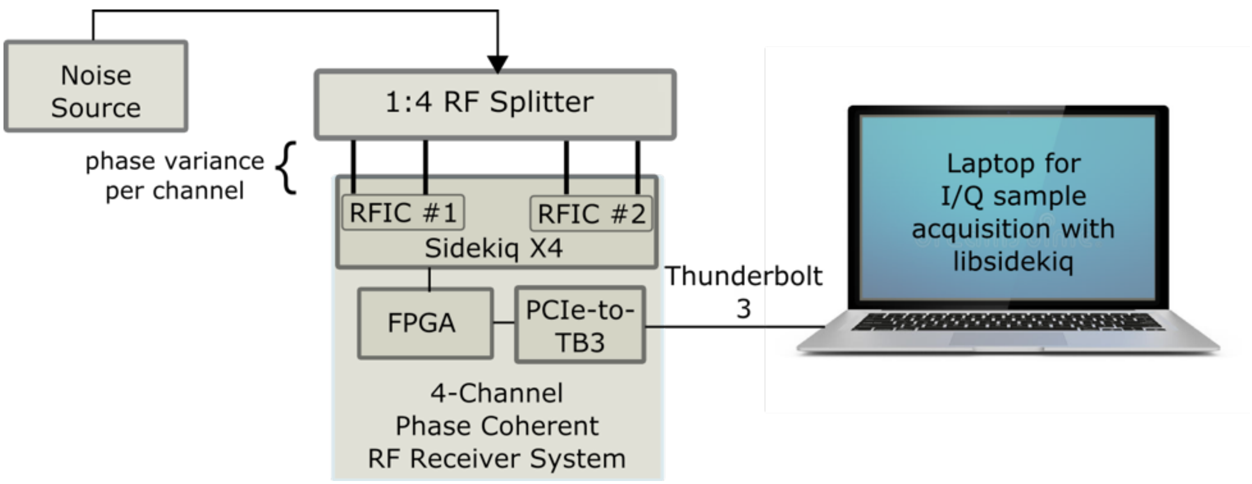
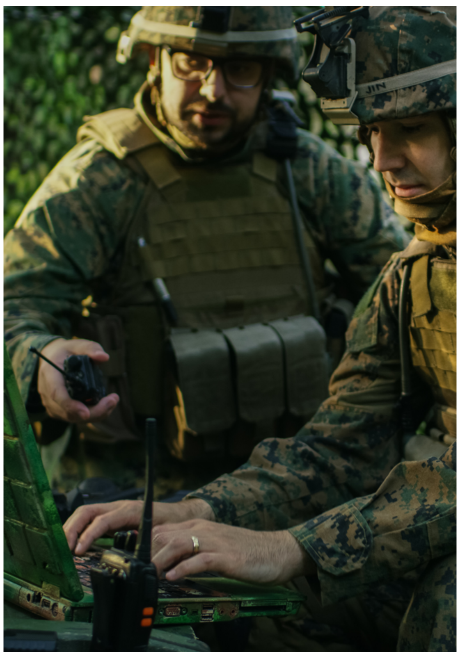



*Figure 7: Typical lab setup for measuring/verifying phase coherence on Sidekiq X4*

As shown above, a single amplified noise source is used to excite all four channels. The receive samples are time-aligned and correlated in the following manner:

$$\boldsymbol{R_{fg}[n] = f * g = \Sigma_{m=0}^{n} f * [m] g[m+n]}$$



This yields the correlation array $R_{fg}[n]$ of length (N + M) where N is the length of F and M is the length of G (ideally the same length). The mid-point of this array (N + M)/2 corresponds to a zero-sample shift between the F and G; that is, F and G are overlapped entirely.

Values to the left of the mid-point correspond to G leading F, and values to the right correspond to F leading G. The correlation array magnitude of $R_{fg}[n]$ represents the amount of correlation between F and G. This correlation function is often normalized since the magnitude depends on the number of samples and the individual values of F and G.

For our calibration, we are using a broadband noise source which behaves very much like Gaussian noise. The auto-correlation function of true Gaussian noise is a Dirac delta. However, in our case with the Sidekiq X4, the signal is not true Gaussian noise because it has a bandwidth limitation set by the bandwidth of the receiver. In the case of bandwidth-limited Gaussian noise, the output of the autocorrelation is closely approximated by the classic sinc function. Referring to Figure 8 below, if a block of I/Q samples were captured for two channels which had zero-time difference between the two, we would expect the output to be a sinc function with the maximum exactly at the 0th sample.

In Figures 8 and 9, the dashed line represents the normalized sinc function. The discrete stem plot represents the actual output values of our correlation function since it is a sampled system. The x-axis is the shift in samples between the two received signals. The sample rate, Fs, in our case is 122.88 Msamples/sec and the bandwidth, B, is 100 MHz. Each sample represents 1/122.88e6 seconds (8.138 ns). Some other points of note are the zero crossings of the sinc function which occur at multiples of 1/B.

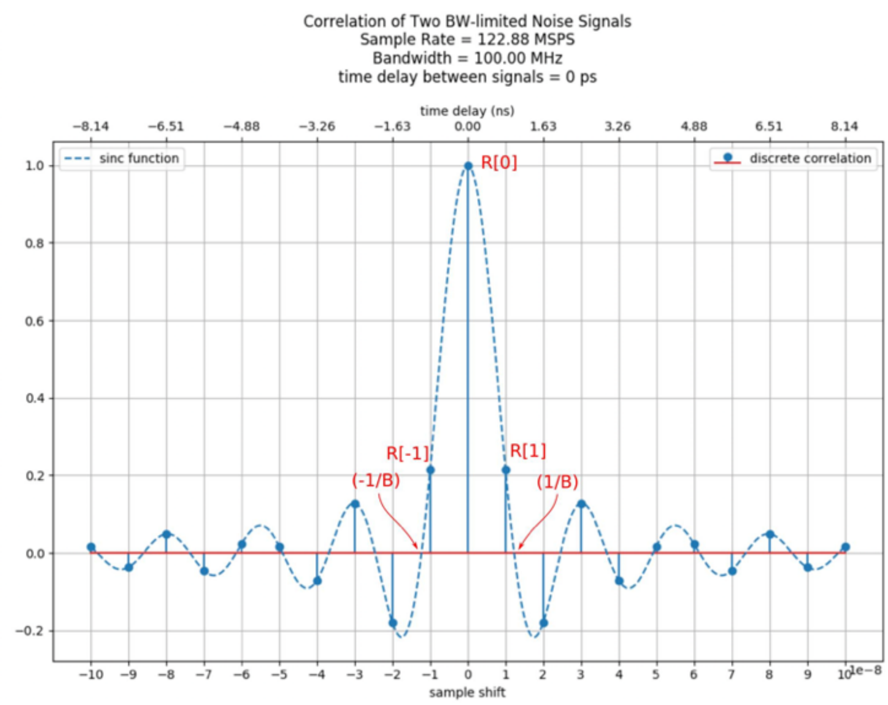



*Figure 8: Correlation of BW-limited noise with no time delay*

When the two channels in question are perfectly matched in phase, that is, there is no time delay between the two channels, the correlation will look like Figure 8 above. However, the RF front ends of each channel on the Sidekiq X4 are not perfectly matched. Each RF front end also has its own bank of selectable band-pass filters for pre-selection. Although they are the same filters, there are tiny phase differences between them. Also, Sidekiq X4 implements an option to allow RF receiver Rx1 on each ADRV9009 RFIC to alternately be used as a wider bandwidth Observation Receiver (ORx) in order to support up to 400 MHz of Rx bandwidth in a single Rx channel. This requires the inclusion of an RF switch in the hardware lineup that is not present on the Rx2 front end. Therefore, there will be a small time delay between Rx1 and Rx2 for each ADRV9009 RFIC on Sidekiq X4 which must be accounted for in order to phase match the channels.

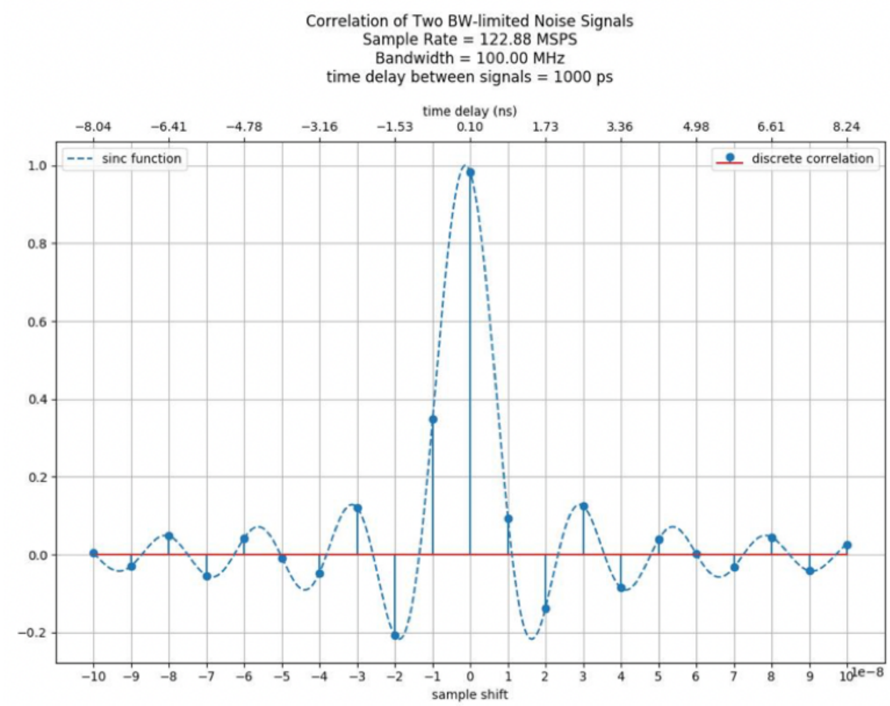



*Figure 9: Correlation of BW-limited noise with 1ns delay*

In Figure 9 above, the sinc function is shifted to the left by 1ns, which is only a fraction of a sample. We can see that the sinc function is shifted slightly from zero. In fact, it's shifted by 0.12288 samples since a full sample is 8.13ns. Also, from the plot above, it can be seen that R[1], the first sample to the right, is approaching 0. Meanwhile, R[-1], the first sample to the left is increasing.

For simplicity, the figures above show the correlation of two real (not complex) sample streams. However, in practice, these are streams of quadrature samples, with both a real and imaginary component. Since the signals we are using are complex, the correlations are also complex and thus actually produce two sinc functions. The phase difference between two streams of samples is the angle of the correlation at the peak sample - the point at which both signals correlate.

Figure 10 below shows one such correlation operation across four channels taken from a Sidekiq X4. Note that there are two sinc functions in each plot.

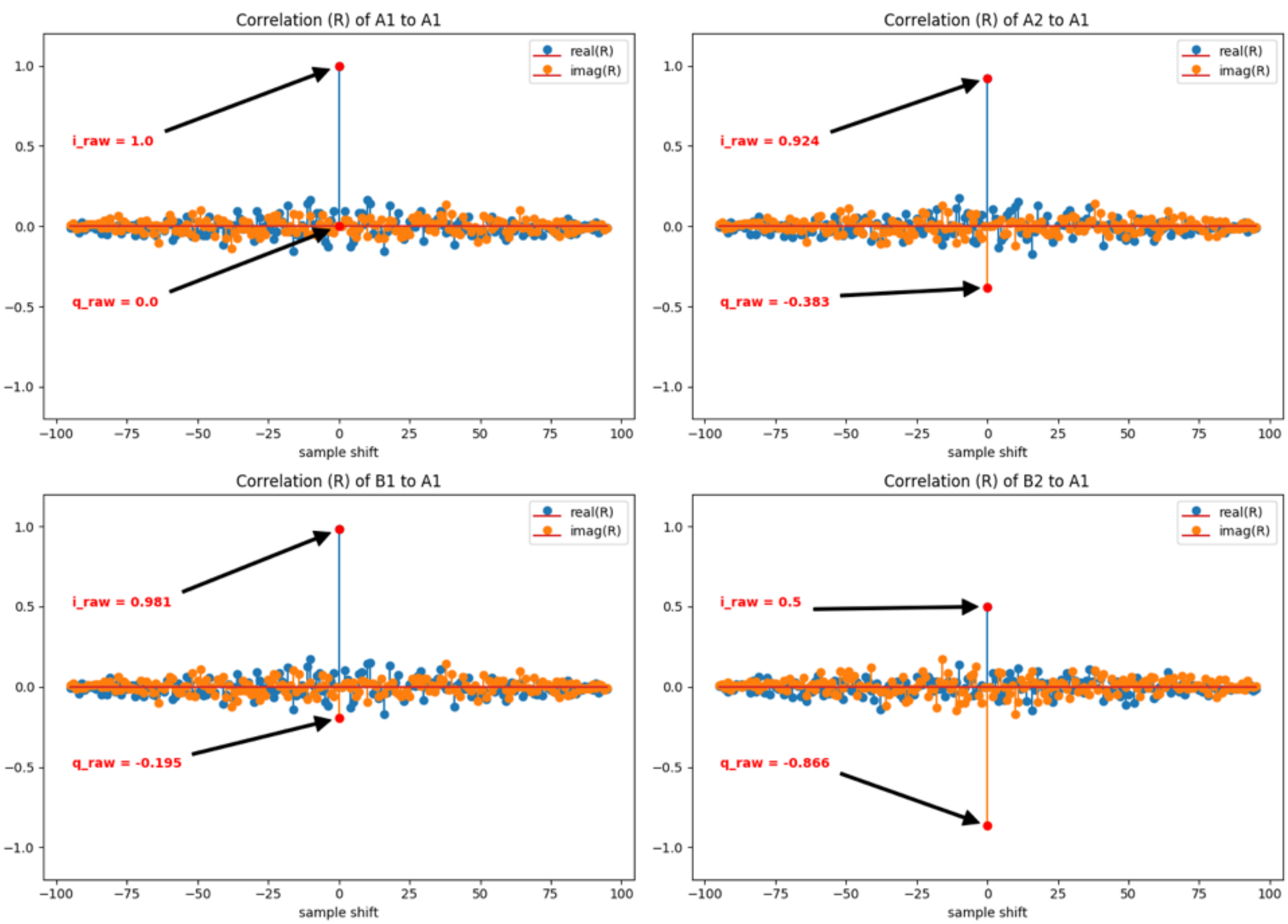



*Figure 10: Complex correlation of noise across all four channels of Sidekiq X4 referenced to channel A1*

Correlation can only compare two streams of samples at a time, so one channel is chosen to be the reference channel for all others. The reference channel is arbitrary, but for the figure above, channel A1 has been chosen. For completeness, the A1-to-A1 correlation has also been shown (top left), although this will always correlate perfectly since it is the exact same data. And indeed, the top left plot shows that there is no phase difference in the A1-to-A1 correlation since the imaginary part is zero.

Correlation can only compare two streams of samples at a time, so one channel is chosen to be the reference channel for all others.

However, the bottom right chart in the above figure, which shows the correlation for B2-to-A1, is quite a bit different. In this case, there is an angle of (0.5 - j*0.866) between the two signals. This equates to -1.04 radians or -60 degrees. If the B2 samples could be rotated by +60 degrees, the phase difference between B2 and A1 would be zeroed out. This can be accomplished by performing a complex multiplication of the B2 samples by (0.5 + j*0.866). The formula for finding the phase calibration factor for a single LO frequency at a single sample rate is:

$$\varphi_{channel-of-interest} = -1 * angle\left(peak\left(C(s_{reference-channel}, s_{channel-of-interest})\right)\right.$$

where C is the correlation function.

**Because any two channels will have both phase and amplitude differences due to path differences in the hardware, an amplitude scaling factor (k) will also need to be found.**

Comparing the amplitudes of channels is fairly straightforward. Simply perform an FFT operation and then take the mean of the entire array.

Because any two channels will have both phase and amplitude differences due to path differences in the hardware, an amplitude scaling factor (k) will also need to be found. As with the phase, there needs to be a common reference channel. When finding k, however, the reference channel cannot be arbitrarily chosen. The reason for this is because any corrected sample values should not exceed the maximum representable sample value of the ADC. In the case of Sidekiq X4, receive ADCs are 16 bits with a sign bit, which has a valid range of -32768 to +32767. So, if for instance, there was a scaling factor of 1.2 between channel X and Y, and this scale factor was applied to a sample with a value of 28000, this would result in a value of 33600 which is outside of the valid range for a signed 16 bit number. This operation would cause a clipping effect which would manifest as noise and unwanted products in the baseband signal. Therefore, only k values of 1.0 or less are valid and the way this is accomplished is as follows:

1. Find the channel with the lowest relative magnitude. This is the reference channel for finding the k value.

2. Compare all other channels to the reference channel. This ensures that all of the other channels will need to be attenuated (k <= 1.0) in order to be equal in amplitude.

Comparing the amplitudes of channels is fairly straightforward. Simply perform an FFT operation and then take the mean of the entire array. The diagram below shows a 96 point FFT (not logarithmic) of noise along with a line representing the mean.

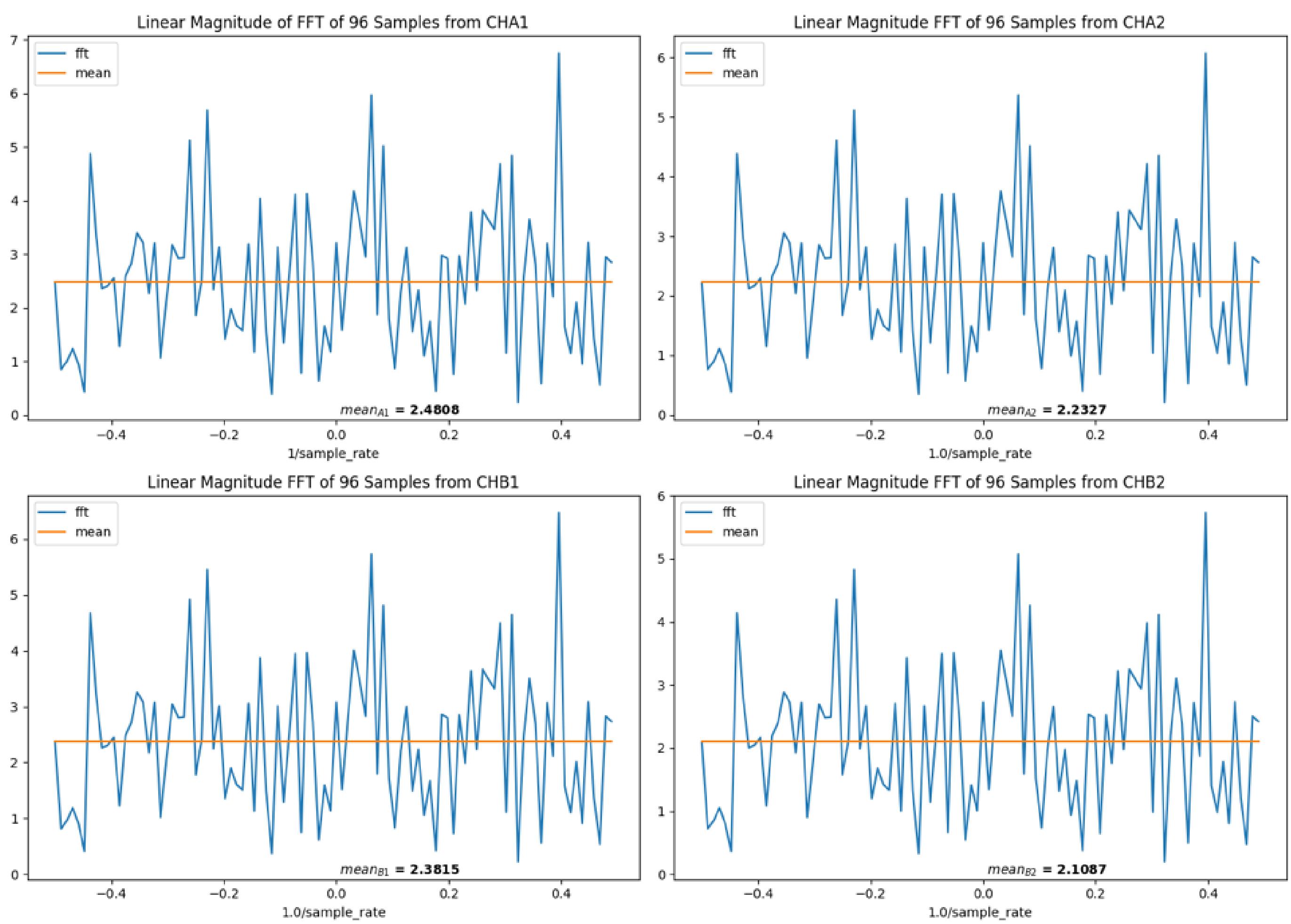



*Figure 11: FFT of noise across all four channels of Sidekiq X4*

From the figure above, the weakest channel is channel B2, with a mean FFT value of 2.1087. Therefore, in order to keep scaling values to 1 or less, channel B2 is used as the reference for all other channels. The formula for the scaling factor is as follows:

$$k_{channel-of-interest} = mean(FFT_{reference-channel})/mean(FFT_{channel-of-interest})$$

**The phase and scale factors are all that is needed in order null out any differences in the channels for a given frequency and sample rate.**

Epiq Solutions provides software functions to apply calibration values to each individual stream of receive samples using complex multipliers in the Sidekiq FPGA reference design.

The phase and scale factors are all that is needed in order null out any differences in the channels for a given frequency and sample rate. Epiq Solutions provides software functions to apply calibration values to each individual stream of receive samples using complex multipliers in the Sidekiq FPGA reference design. When enabled, each received I/Q sample is complex multiplied in the FPGA with the real_cal and imag_cal values to rotate and scale the samples. The real_cal and imag_cal values are simply the k and phi values in Cartesian form. The values themselves are between -1 and +1, scaled into fixed-point 16-bit 2's complement (-32768 to 32767). These values are applied to the sample stream by the FPGA by calling the libsidekiq function `skiq_write_iq_complex_multiplier_absolute()` after changing the RF LO frequency (see Sidekiq Software Developer's Manual for more details). Following on the example sample captures from above, these calibration values would be applied as shown in the blue columns below.

| Channel | Unitless | Degrees | real_cal | imag_cal |
|---|---|---|---|---|
| A1 | 0.85 | 0 | 0.85 | 0 |
| A2 | 0.944 | 22.5 | 0.872 | 0.362 |
| B1 | 0.885 | 11.24 | 0.926 | 0.184 |
| B2 | 1.0 | 60.0 | 0.5 | 0.866 |

*The Sidekiq X4 uses these calibration parameters as i and q complex multiplication values.*

For comparison, a plot of the uncalibrated phase-coherent performance across all four RF receive channels in Sidekiq X4 is shown in Figure 12 below, swept from 75 MHz to 6 GHz with a sample rate of 122.88 Msamples/sec and an RF channel bandwidth of 100 MHz. Phase coherence measurements were automatically performed over the course of 1000 execution cycles to assess repeatability over time and between initializations. Figure 12 shows the measured phase coherence before calibration is applied, where a constant phase offset can be observed between each of the four channels at any given RF LO frequency. Even though there is a constant phase offset between the four channels at a given RF LO frequency, the standard deviation of the phase difference between all four channels is less than 2 degrees, indicating a high degree of phase coherence.

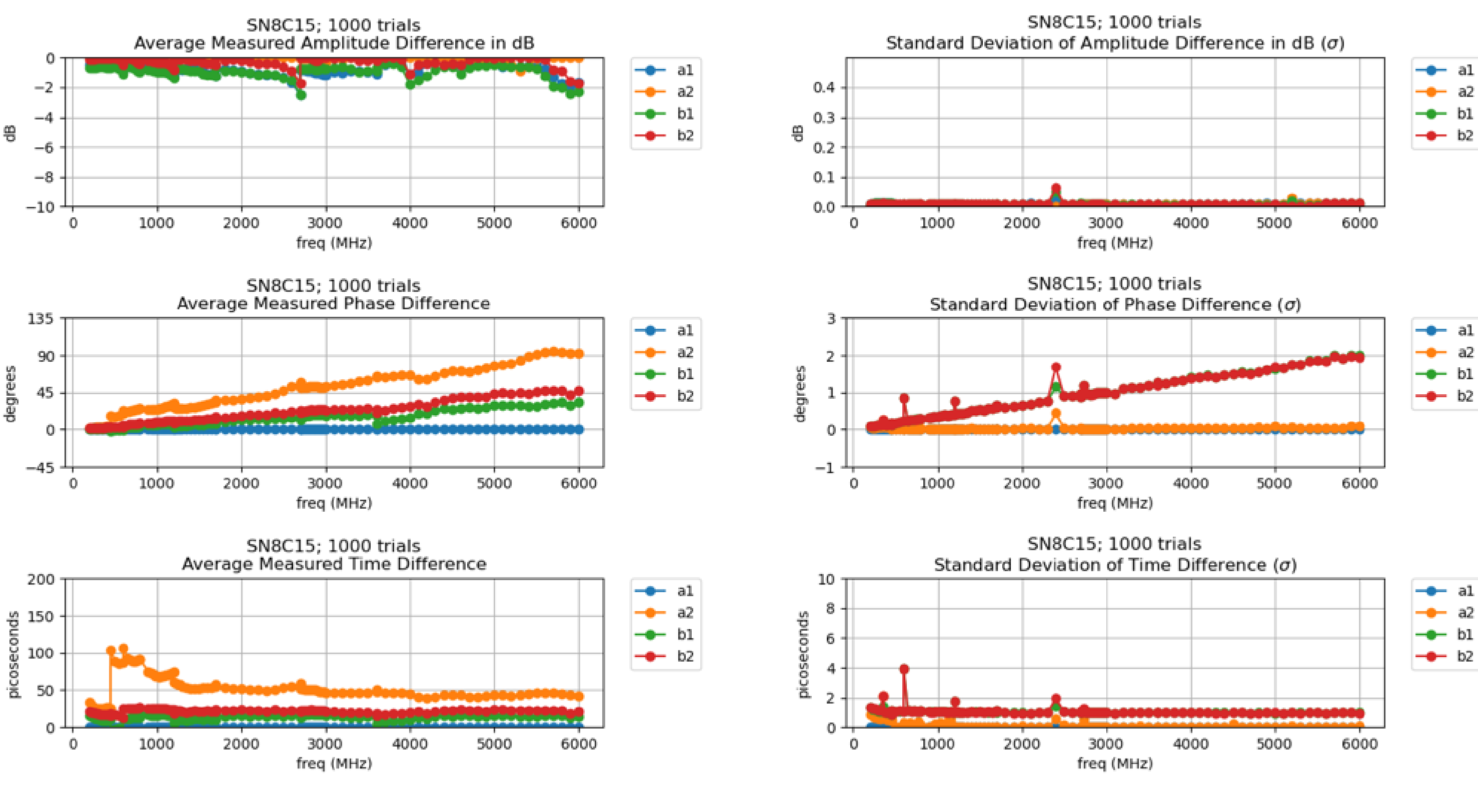



*Figure 12: Measured phase coherence results for Sidekiq X4 over 1000 trial runs (without phase and amplitude calibration)*

**During the integration of Sidekiq X4 into a host system, many factors can influence the synchronization of a multi-channel radio system.**

## Real-World Considerations for Building and Deploying Phase-Coherent Systems Based on Sidekiq X4

Every element in the system introduces an unknown phase/time difference. The summation of these phase/time differences can be accounted for through the use of a one-time calibration routine that measures these differences and creates a structured record of calibration values vs RF frequency for the card.

During the integration of Sidekiq X4 into a host system, many factors can influence the synchronization of a multi-channel radio system. These factors can be external to the platform such as the length of cables, filters, switches, antennas or internal such as the backplane and PCB trace routing. Every element in the system introduces an unknown phase/time difference. The summation of these phase/time differences can be accounted for through the use of a one-time calibration routine that measures these differences and creates a structured record of calibration values vs RF frequency for the card. While an uncalibrated system with Sidekiq X4 will technically still be phase-coherent, there will be a non-zero (but constant) group delay between each of the four RF receiver channels. This group delay will remain constant through changes in the RF LO center frequency as well as power cycles, and can thus be calibrated out.

For deployments where it is desirable to have this non-zero (but constant) group delay calibrated out, customers can apply calibration factors to each I/Q sample stream, which are complex multiplied in the FPGA. The hardware setup required to perform this calibration (along with any additional filtering/antennas/etc. included as part of the custom RF front end needed for the user's system) is shown in Figure 13 below. An amplified broadband noise source is used to inject noise over the air, received by the system, and processed to perform the actual calibration procedure, as described below.

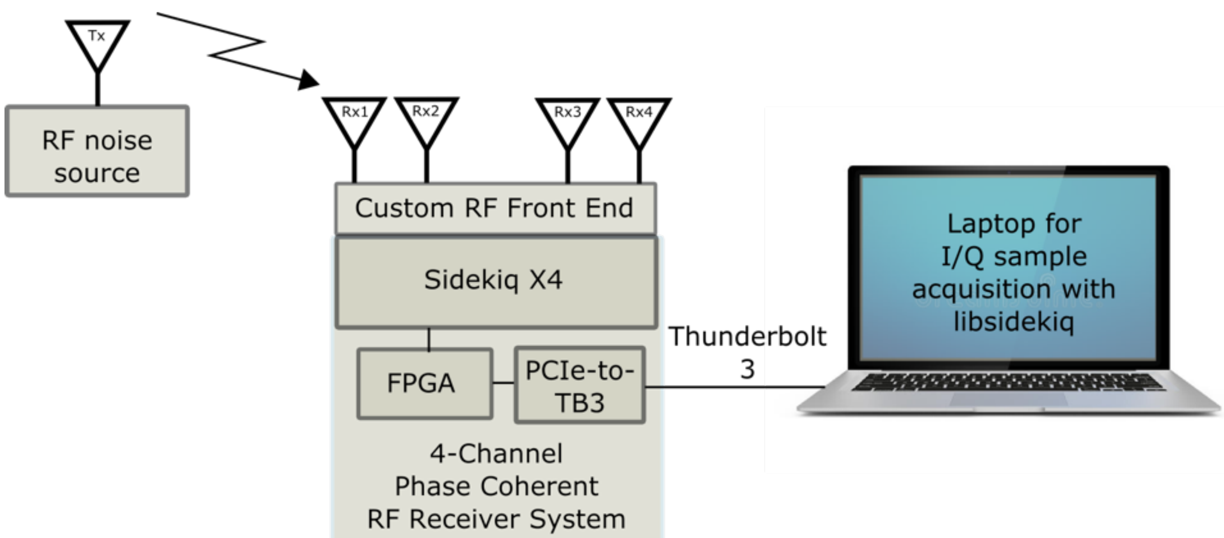



*Figure 13: Field calibration test setup for Sidekiq X4*

## Calibration Procedure

The flowchart for the general calibration procedure to compensate for any phase differences between the antenna and the RF receivers on Sidekiq X4 is shown in Figure 14 below. Since these are comparative measurements, one channel must be chosen as the reference channel. In this case, we've chosen channel A1. Samples are captured simultaneously for all four channels at each frequency. FFTs are performed on the samples. To determine the amplitude error, k, the magnitudes of each FFT are compared (divided by) the reference channel (A1). For the phase error, phi, each channel is cross correlated with the reference channel (A1). The phase of the correlation function at maximum correlation represents the phase difference between the channels. These two values, k and phi, are used to calculate the real_cal and imag_cal values which can then be stored in a file or other non-volatile storage available to the host system for use in operation (as shown in Figure 14).

The phase of the correlation function at maximum correlation represents the phase difference between the channels.

For Sidekiq X4, the group delay is constant between frequency tuning events, as well as power cycling events.

The calibration procedure generates real_cal and imag_cal values for each RF frequency of interest . While Sidekiq X4 provides an LO tuning resolution of ~4 Hz, it is not necessary to pre-calculate calibration values at this resolution. The required resolution will be application specific based on accuracy requirements, but a reasonable rule of thumb would be to perform and store calibration data every 10 MHz to 100 MHz. The libsidekiq software library provides an API to program the desired real_cal and imag_cal values down to the FPGA to perform the complex multiplication on each I/Q sample for each of the four Rx channels at run-time to correct for the phase and amplitude differences between channels. The calibration procedure generates real_cal and imag_cal values for each RF frequency of interest. During operation, the user's software application can apply the calibration values (through a call to `skiq_write_iq_complex_multiplier_absolute()` ) after each LO tuning operation, using the appropriate real_cal and imag_cal values previously determined for the LO frequency of interest.

Note that the calibration procedure is typically performed at a single ambient temperature point. Changes in the operational temperature of the system at run time may introduce a modest variance in the phase coherence, but due to the fact that the four RF receivers are in very close physical proximity to each other on Sidekiq X4, the temperature shift of the system is seen uniformly by all four receivers. Thus, the change in temperature doesn't typically have a substantial impact on the phase coherence operation.

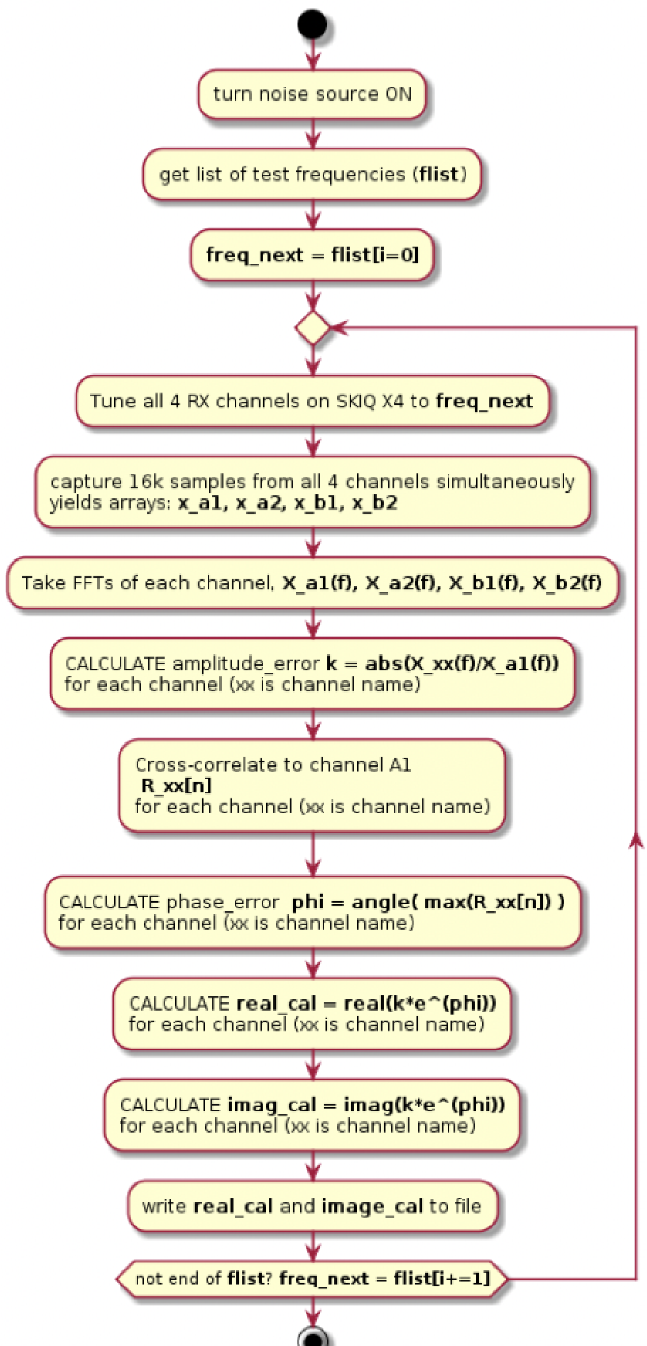



*Figure 14: The algorithm for one-time phase calibration of an RF system with Sidekiq X4 (factoring in all external antennas/cables/filters/etc.)*

Figure 15 below shows a plot of the final measured phase coherence of Sidekiq X4 with calibration applied.  This shows outstanding phase coherence performance between all four channels over the course of 950 cycles of measurement sweeping from 75 MHz to 6 GHz. The standard deviation of the phase difference is less than 1 degree for all RF frequencies.

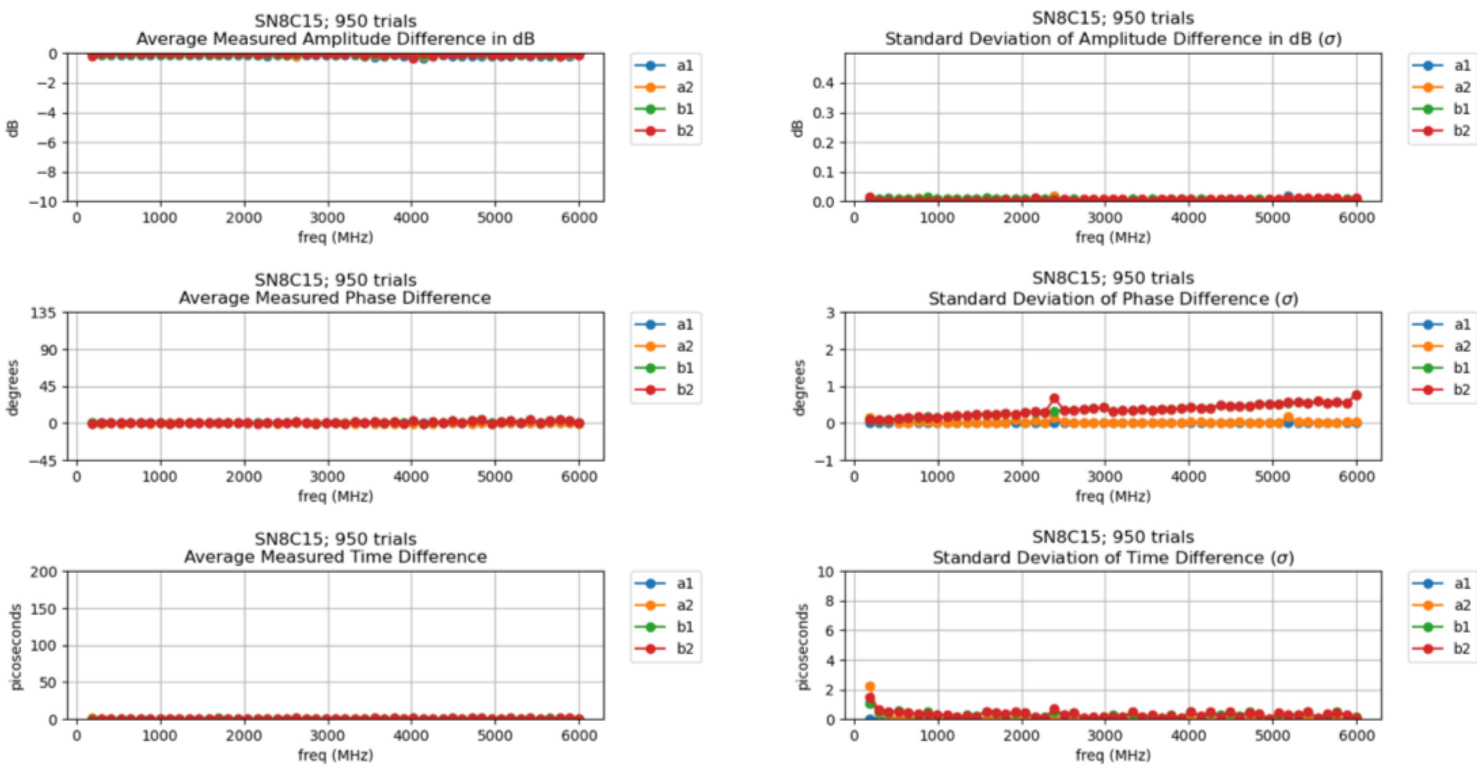



*Figure 15: Measured phase coherence results for Sidekiq X4 over 950 trial runs (with phase and amplitude calibration applied)*

## Conclusion

Four-channel phase-coherent receiver systems form the cornerstone of many types of MIMO and direction-finding systems on the market today. There is often a substantial amount of engineering required to develop and calibrate a phase-coherent system, as well as measure the phase coherence performance. This whitepaper provided an overview of this process from start to finish using Epiq Solutions' Sidekiq X4 and associated test software application as a basis. Typical phase coherence is better than 1 degree of phase accuracy between all four channels, and this level of coherency is maintained even when changing RF center frequencies or power cycling the system. Sidekiq X4 is available in a SOSA-aligned 3U VPX card so vendors can bring high-performance phase-coherent RF reception to their DoD application sets. Sidekiq X4 is part of Epiq Solutions' portfolio of SDR transceiver modules that lead the way in size, weight, and power consumption. For more information about how we can help you get up and running fast with your mission-critical defense applications, reach out to us here.

CONTACT US

With more than a decade in business, Epiq Solutions understands how important speed, cost, and performance are for defense and security applications. Our radically small SDR transceiver modules and turnkey RF sensing tools lead the way in size, weight, and power consumption. Whether you're developing mission-critical defense applications for the battlefield or protecting sensitive information, you can trust us to get you there fast.

LEARN MORE

EPIQ SOLUTIONS

epiqsolutions.com