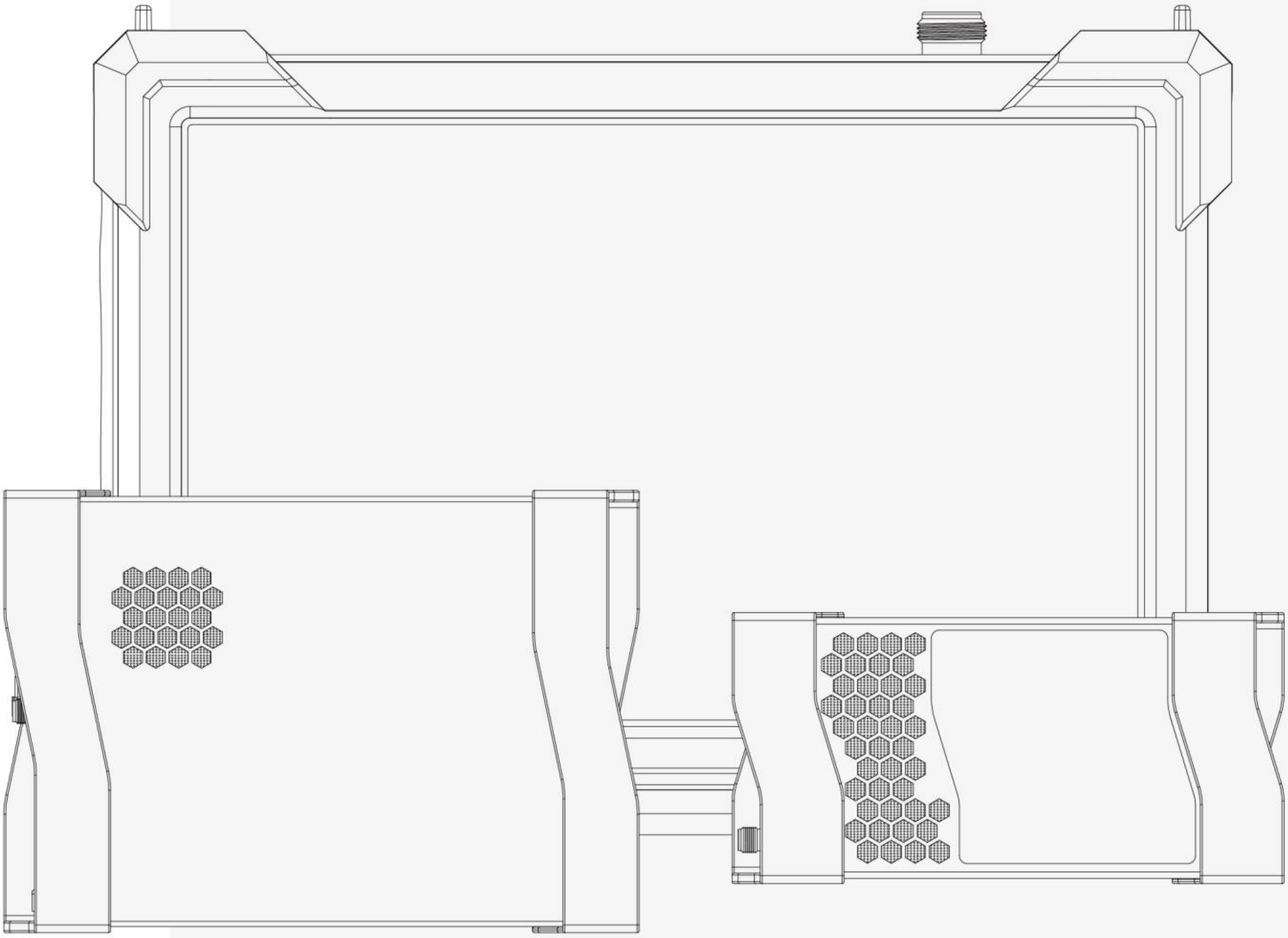
USER
GUIDE

HTRA FAQ and Troubleshooting Guide



V2.2
12/15/2025

HAROGIC

目录

Version Management

Updated Description Sheet

Version	Description	Date
V2.2	1. Modified: Moved the FAQ section to Chapter 1 2. Modified: Updated the content formatting 3. Modified: Updated the Version Management section	12/12/2025
V2.1	1. Added: Added Version Management chapter	11/18/2025
V2.0	1. Modified: Updated the description of the existing content 2. Added: Added supplementary content	10/10/2025
V1.9	1. Added: Added a FAQ section 2. Modified: Updated the description in the Troubleshooting Guide section	9/24/2025
V1.8	Initial Version	8/8/2025

1. FAQ

1.1 API Usage and Development

Q: In RTA mode, why does the displayed power change not accurately reflect the actual signal's power change?

A: The RTA mode has a quantization and floor characteristic of 0.75 dB per division. Therefore, any change in displayed power is always an integer multiple of 0.75 dB. This means that any real change that is not an integer multiple of 0.75 dB will be truncated or ignored.

Q: When calling an API, only part of the parameters are configured. How will the unconfigured parameters be handled on the device?

A: The unconfigured parameters will automatically be passed using the default values set after initialization**.**

Q: How to set the BusTimeout_ms?

A: The theoretical value for BusTimeout_ms is the time to capture one data packet (points per packet * decimation factor / sample rate). In practice, after Configuration, a certain trigger waiting time is needed. Therefore, BusTimeout_ms should be ≥ trigger arrival time + one packet capture time. It can be set to twice the theoretical timeout time.

Q: What is the difference between Realtime and non-Realtime query functions? For example, Device_QueryDeviceState and Device_QueryDeviceState_Realtime.

A: All Realtime and non-Realtime query functions adhere to the following rules:

- Device_QueryDeviceState: Obtains the device state from the device pointer and does not require communication with the device (the device state is updated after calling Get and Configure functions)
- Device_QueryDeviceState_Realtime: Requires communication with the device to obtain the device state in real-time.

Q: In SWP mode, which parameters affect the device sweep time and the number of spectrum points, and how do they affect it?

A: Sweep time is inversely proportional to sweep speed.
Parameters affecting device sweep time are:

- RBW: The larger the RBW, the faster the sweep speed. When RBW ≥ 40 kHz, the FFT is processed by the FPGA, resulting in even faster speed;
- VBW: The larger the VBW, the faster the sweep speed. When VBW ≥ 10 * RBW, the VBW filter is bypassed, and sweep speed no longer increases with larger VBW;
- Spur Suppression: The lower the spur suppression level, the faster the sweep speed. Sweep speed is fastest when spur suppression is turned off;

- Window Factor: The smaller the window factor, the faster the sweep speed. Choosing the Nuttall window results in faster sweep speed than the FlatTop window;
- Power Balance: The smaller the power balance value, the faster the sweep speed;
- Frequency: Switching times for components differ across frequency bands, and LO switching times for different frequency points vary, leading to different device sweep times in different frequency bands.

Parameters affecting the number of spectrum points are: Span, RBW, trace points strategy, trace detector, and spur suppression strategy.

Q: In SWP mode, after obtaining spectrum data by calling SWP_GetPartialSweep, why does calling SWP_GetPartialSweep again after a period of time not return the latest data?

A: When SWP_Configuration is called, the device captures several extra initial frames of spectrum data. If data acquisition is interrupted for a period and then SWP_GetPartialSweep is called again, the first few frames retrieved will be this stored old data, not the latest real-time data.

Q: How should I choose among the three window types: Flattop, Blackman_Nuttall, and LowSideLobe?

A:

- FlatTop Window: Provides the highest amplitude accuracy. It is used for power measurements.
- B-Nuttall Window: Offers higher frequency selectivity. It is used for separating dense spectrums, radar signal processing, and frequency accuracy measurements
- LowSideLobe Window: Provides higher measurement accuracy for low-frequency signals and strong leakage suppression capability. It is used for detecting weak signals in the presence of strong interference

Q: Can the API obtain timestamp and latitude/longitude information?

A: When the device is equipped with a GNSS option and has successfully locked onto a signal, this information can be obtained by accessing the measurement auxiliary information structure MeasAuxInfo or by using GNSS-related query functions such as Device_GetGNSSInfo.

Q: Do the software's recording files include GNSS information?

A: If the device is equipped with a GNSS option and has successfully locked onto a signal, the recording files will contain GNSS information, including UTC time, longitude, and latitude.

Q: What are the differences between the High-Pass Filter, Manual Offset, and Automatic Offset settings within DC Cancellation?

A: DC cancellation is only required when the analysis bandwidth is set to 100MHz.

- Enable High-Pass Filter: Activating the high-pass filter can completely suppress DC spurious signals, but it also results in some signal loss.

- Enable Automatic Offset: Based on the internal API algorithm, this setting suppresses DC spurious signals without causing signal loss. However, the suppression level is limited to approximately -50 dBc, and residual DC drift may still remain.
- Enable Manual Offset: This requires manually adjusting the DC cancellation offset values for the I and Q channels. The adjustment method involves first modifying the offset of one channel until the DC suppression effect no longer changes, then adjusting the other channel's offset until its suppression effect also stabilizes. By iterating this process, a better suppression effect can be achieved compared to automatic offset. It's important to note that the effectiveness of both automatic and manual offset in suppressing DC spurious signals is affected by the device's temperature. As the device temperature increases, the suppression performance tends to degrade.

Q: Can the sample rate parameter be set directly in SWP mode?

A: The sample rate cannot be set directly. In SWP mode, the sample rate is automatically calculated and set by the device's internal strategy based on the spectrum parameters configured by the user (such as center frequency, span, RBW, etc.). This ensures optimal frequency-domain measurement performance.

Q: Is the number of sampling points adjustable in SWP mode?

A: It cannot be manually adjusted; only the number of trace points can be set. The number of time-domain sampling points required for the device to complete frequency-domain analysis is automatically determined by the device's internal algorithm based on the frequency-domain parameters set by the user.

Q: When calling APIs or using the software, can the attenuation and gain values be set directly?

A: The device's attenuation and gain are linked to the reference level and cannot be set manually.

- Attenuation details: Reference Level = Attenuation Value - 10. In the API, the parameter for setting attenuation is SWP_ProfileIn.Atten, with a default value of -1 indicating automatic attenuation configuration. The attenuation range is 0 to 33 dB. If a reference level is manually configured while attenuation is set to a value greater than -1, the API will prioritize mapping the attenuation value to the reference level, disregarding the configured reference level.
- Gain details: When the reference level is below -30 dBm, the device automatically decides whether to enable the pre-amplifier based on the current configuration. The gain amount also cannot be manually controlled.

Q: What are the differences between IQ data precision options, and how should they be chosen?

A:

- Complex16bit: The acquired IQ data has 16-bit precision with the data type int16_t. It offers a dynamic range of 96 dB, with an actual measurement range of approximately 80 to 90 dB. Its advantages include good dynamic range, high precision, and strong versatility. It is suitable for general spectrum analysis, communication testing, and scenarios that balance speed and accuracy.
- Complex32bit: The acquired IQ data has 32-bit precision with the data type int32_t. It offers a dynamic range of 192 dB. Its strengths are extremely high precision, low likelihood of overflow, and ease of calculation. However, it generates a very large data volume. It is suitable for in-depth signal analysis.
- Complex8bit: The acquired IQ data has 8-bit precision with the data type int8_t. It offers a dynamic range of 48 dB, with an actual measurement range of approximately 40 to 45 dB. Its benefits are small data size, fast transmission, and fast processing. However, its dynamic range is low and precision is poor. It is suitable for experimental environments with low dynamic range requirements but a need for high-speed processing.

Q: What is the difference between the IQ data obtained via API and the IQ data displayed in the software?

A: The IQ data in the software is in units of microvolts (uV). The IQ data obtained via the API is raw, unitless IQ data. To convert the raw IQ data into IQ data in volts (V), it must be multiplied by IQS_ScaleToV.

Q: What is the difference between the trigger sources "Free Run" and "Bus Trigger"?

A: Free Run does not require a trigger. Bus Trigger requires a software trigger (rising edge or falling edge) to be received before data acquisition can begin.

Q: What is the relationship between RBW and frequency interval?

A: RBW = Current Frequency Interval / Trace Detection Ratio * Window Factor (under the premise of no zero-padding).
Window Factor: The window factor for FlatTop is 3.77, and for Nuttall window it's 1.976.
Trace Detection Ratio: This is the value of TraceDetectRatio in the SWP_TraceInfo_TypeDef structure.

Q: After AM/FM demodulation, is the data still digital IQ data?

A: The data after demodulation is no longer IQ data, but it is still digital signal data.

Q: In SWP mode, step-like spectrum patterns appear outside the DS segment.

A: This phenomenon is normal. The steps represent the segmentation points during the sweep. The power value at these frequency points is correct and will not affect the normal operation of the equipment.

Verification method: Input a single-tone signal at one of these frequency points and observe that the power reading at that point is accurate.

Q: How does the preview time in the software correspond to the TriggerLength set via the API call?

A: The conversion between the preview time and TriggerLength is related to the decimation factor and sampling rate. The relationship is as follows:

$$\text{PreviewTime} = \text{TriggerLength} * \frac{\text{DecimationFactor}}{\text{NativeSampleRate}}$$

Introducing API parameters:

In IQS mode:

$$\text{PreviewTime} = \text{IQS\_ProfileIn.TriggerLength} * \frac{\text{IQS\_ProfileIn.DecimationFactor}}{\text{IQS\_ProfileIn.NativeSampleRate\_SPS}}$$

In DET mode:

$$\text{PreviewTime} = \text{DET\_ProfileIn.TriggerLength} * \text{StreamInfo.TimeResolution}$$

Q: How to convert IQ raw data into power data of the spectrum in dBm units?

A: Please refer to the following formula:

$$\text{dBm} = 10 * \log_{10}(20(I^2 + Q^2) * \text{IQS\_ScaleToV}^2)$$

Q: When calling the Device_SetFreqResponseCompensation function to perform amplitude compensation for a specific frequency band, are the correction values the same for different reference levels?

A: Theoretically, the correction value is independent of the reference level and is only related to frequency.

Q: When calling the API, what is the difference between the two methods of obtaining spectrum data, SWP_GetFullSweep and SWP_GetPartialSweep?

A:

SWP_GetFullSweep: Fetches a complete spectral trace all at once, with no need for splicing.

SWP_GetPartialSweep: Each call fetches only a segment of the complete trace. It requires multiple calls and splicing to obtain the full trace, allowing the user to observe the spectrum being generated step by step

Q: Why is the spectrum obtained wider than the configured frequency, and how can it be resolved?

A: During the acquisition process, the device by default captures a frequency band wider than the parameters set. This can be resolved by calling the DSP_InterceptSpectrum function to perform extraction.

Q: How can the noise floor be reduced?

A: By decreasing the Reference Level and RBW, the device's noise floor can be lowered.

Q: After synchronizing the reference clocks of multiple spectrum analyzer modules, can the acquired IQ data or spectrum data be seamlessly spliced

A: Spectrum data can be spliced, but care must be taken to avoid signals falling exactly at the junction between two devices, as this may distort the signal shape.
IQ data cannot be spliced because splicing requires a strict phase relationship between different segments of IQ waveforms, which cannot be guaranteed solely by clock synchronization

Q: How is the Resolution Bandwidth (RBW) calculated?

A: RBW can be calculated based on the sample rate, window factor, decimation factor, and the number of sample points:

$$\text{RBW} = \frac{\text{Sample Rate}}{\text{Decimation Factor} * \text{Sample Points}} * \text{Window Factor}$$

Q: Code written on Windows that calls APIs, can it be used on Ubuntu under Linux?

A: Yes, the libraries provided for Linux and Windows are synchronized. However, you need to confirm whether the code references any header files or libraries exclusive to Windows (such as Windows.h). If it contains such dependencies, it cannot be used directly on Ubuntu.

Q: In IQS mode, how can data of a fixed duration be acquired in the FixedPoints trigger mode?

A: The FixedPoints mode acquires data based on the number of points, which indirectly enables fixed-duration acquisition. The method is to calculate the acquisition duration based on the sample rate, decimation factor, and trigger length:

$$\text{Acquisition Duration} = \frac{\text{Decimation Factor}}{\text{Sample Rate}} * \text{Triggerlength}$$

Q: When calling API functions, can configuration-related functions (such as parameter setting functions) be called multiple times?

A: Yes, except for Device_Open and Device_Close, all other functions can be called multiple times. However, the calling constraints must be met, and it must be ensured that only one function call is executed at any given time. Additionally, after each execution of Device_Open, a corresponding Device_Close must be called to properly close the device.

Q: In SWP mode, what is the difference between the Detector and the Trace Detector?

A:

- Detector: At the same local oscillator frequency point, it collects frame data based on the detection ratio. According to the detector's characteristics, it performs detection point-by-point across multiple frames of data, ultimately generating a characteristic value frame;
- Trace Detector: Based on the selected trace detector, it performs detection on the entire spectrum trace in steps defined by the trace detection ratio, thereby generating a characteristic value trace.

Q: In the software's IQ mode, with full span and the trigger mode set to FixedPoints, how should the external trigger period and preview time be set to avoid error -10?

A: The preview time must satisfy the following relationship:

$$\text{Preview Time} = \frac{\text{External Trigger Period}}{2} * 0.8$$

Q: When using a new spectrum analyzer, does the existing program need to be recompiled due to the replacement of calibration files?

A: No recompilation is required. You only need to copy all the calibration files from the CalFile folder in the accompanying materials of the new device to the program's CalFile folder.

Q: Can the sweep time be obtained? How are the trace sweep time and frame sweep time calculated?

A:

- When SweepTimeMode = SWTMode_Manual, the trace sweep time is equal to the set value of SweepTime (lower limit 0.1 s). In other modes, it can only be calculated by adding timers before and after Get;
- Frame sweep time = Trace sweep time / TraceInfo.TotalHops;
- Additional note: TraceInfo.FrameTime represents the acquisition time per frame of the device, excluding processing overhead such as FFT and data transfer.

Q: When continuously acquiring IQ data, is it necessary to cyclically call the API interface for data acquisition?

A: Yes, it is necessary to cyclically call the API interface for data acquisition. It is recommended to continuously call the Get function.

Q: Can the analysis bandwidth in IQS mode be set?

A: It can be set indirectly. It can be set indirectly through the decimation factor. A decimation factor of 1 corresponds to 100 MHz, and a factor of 2 corresponds to 50 MHz. The range of the decimation factor is: 2^0 to 2^12.

Q: Can the FM and AM demodulation interfaces directly return a sine wave?

A: If the input FM or AM signal is modulated with a single-tone signal, then the result after demodulation will be a sine wave.

Q: Why does RBW affect the noise floor?

A:

The noise floor is the sum of noise power within the RBW bandwidth. Therefore:

- When RBW decreases, the total power within the bandwidth decreases, and the noise floor is lowered;
- When RBW increases, the total power within the bandwidth increases, and the noise floor is raised.

Q: How is the time per frame of data calculated in RTA mode?

A:

After calling RTA_Configuration, use the parameters in the RTA_FrameInfo_TypeDef structure for calculation:

- FrameInfo.PacketAcqTime / FrameInfo.PacketFrame;
- FrameInfo.TimeResolution * FrameInfo.FFTSize.

Q: What is the difference between the Adaptive and FixedPoints trigger modes?

A:

- FixedPoints: After the trigger rising edge arrives, the device collects a fixed number of IQ points (TriggerLength). During collection, it does not respond to new triggers. After completion, it waits for the next trigger;
- Adaptive: Collection starts when the trigger rising edge arrives and stops when the trigger falling edge arrives. The collection length is determined by the duration of the trigger high level.

Q: Can the device only acquire IQ data for a single point? Can the bandwidth in IQS mode be set to 0?

A: No. In IQ mode, a minimum of 32 points must be acquired per acquisition. In IQS mode, the minimum bandwidth is 125 MHz / 4096 ≈ 30.5 kHz.

1.2 Software Usage

Q: Are there any requirements for the computer screen resolution when using SAStudio4?

A: Yes, there is a requirement. The minimum resolution must be 1280×800.

Q: What are the trace points parameter and the actual trace points parameter in the software?

A: The trace points parameter is the expected value set by the user, while the actual trace points parameter is the real value adjusted by the device according to its internal strategy. The device acquires spectrum data based on the actual trace points.

Q: How to estimate the total number of spectrum points recorded by the software in SWP mode?

A: The following formula can be used:

$$\text{Total points} = \frac{\text{Total recording time}}{\text{Trace sweep time}} * \text{Trace points}$$

where trace sweep time is the time required to acquire one trace, and trace points are the number of points contained in each trace.

Q: When using the recording function of the software, how long can it record continuously at one time?

A: Theoretically, there is no fixed time limit. Recording can continue as long as the following two conditions are met:

- The device has sufficient space (the remaining disk capacity is greater than the set file size limit);
- An appropriate sampling rate is selected according to the device model (not exceeding the maximum sampling rate in that mode).

Q: In SWP mode, are measurements like channel power, phase noise, ACPR, IM3, and OBW performed inside the spectrum analyzer or processed by the host computer?

A: They are processed by the host computer.

Q: In DET mode of the software, why does switching between different detectors not change the trace display?

A: When the Detect Ratio = 1, each frame of data is output directly without comparison or averaging between multiple frames. In this case, the results from different detectors on a single frame are the same, so the trace display does not change. When the Detect Ratio > 1, switching the detector option will cause differences in the trace display.

Q: Is the maximum preview time for DET mode limited to 20ms?

A: Yes, the software restricts the maximum preview time for DET mode to 20ms.

Q: When using the phase noise measurement function in SWP mode of the software, why are the results inaccurate at 100 Hz and 10 MHz frequency offsets?

A:

- Inaccurate phase noise measurement result at 10 MHz offset: It is recommended to set the span to 2 * Maximum Frequency Offset
- Inaccurate phase noise measurement result at 100 Hz offset: This issue can be resolved by setting the start frequency offset in the software's main settings area to be less than 100 Hz (e.g., 99 Hz)

Q: When using SAStudio4 or calling the API, how to ensure that the set trace points number equals the actual returned trace points number in SWP mode?

A:

- TraceBinBW_Hz = RBW

  Set TracePointsStrategy_TypeDef to BinSizeAssined. Then, Trace Points = (Span / RBW). This ensures the frequency bin interval strictly equals the RBW, but the sweep speed will be relatively slower
- TraceBinBW_Hz ≈ RBW

  Refer to the example SWP_RBW_Spaced_Trace.cpp. By reasonably setting the detection ratio, the trace interval can be made close to the RBW

Q: What is the reason for the maximum bandwidth limit of 32.5 MHz in digital demodulation?

A: To ensure good EVM (Error Vector Magnitude), the digital demodulation algorithm requires a minimum of 4x oversampling. Except for the SAN-45/SAN-60 series, other series devices have a native maximum sampling rate of 130 MSPS. Therefore, the maximum bandwidth is 32.5 MHz.

Q: Can recording files from the software be read by other software?

A:

- IQ Mode Streaming Data: Can be played back using SDR#;
- SWP, DET, RTA Data: Currently only supports playback within the software.

Q: Why is the trace sweep time in the software less than the frame sweep time?

A:

This is because the actual acquired Span is often larger than the configured Span:

- Trace Sweep Time: Calculated based on the configured Span;
- Frame Sweep Time: Calculated based on the actual acquired Span

Therefore, when the spectrum has only one frame of data, it is possible for the trace sweep time to be less than the frame sweep time.

Q: In Linux, why does the software intermittently stutter?

A: Because Linux performance is slightly inferior to Windows. Additionally, stuttering can occur when the screen resolution is higher than the resolution at which the software was compiled, but this does not affect data transmission.

Q: In the software, is the spectrum displayed above the IQ mode obtained via the SWP_GetPartialSweep function?

A: No. The spectrum is obtained by first acquiring IQ data and then calling the DSP_FFT_IQSToSpectrum function to perform an FFT operation on the IQ data.

1.3 Device Feature

Q: What is the minimum input power for the phase noise measurement?

A: The typical minimum input power for the phase noise function is -50 dBm.

Q: Which device series support SDR++?

A: As of now, all device series with version 55 support using SDR++. Other versions are not currently adapted.

Q: What is the duty cycle of the timer trigger in IQS/DET/RTA modes?

A: 50%.

Q: Does using the IF gain grade affect the amplitude of the acquired signal?

A: Yes, there is some impact. Differences of approximately 1 dBm in amplitude may occur at certain frequency points between IF gain grade 1 and grade 4.

Q: What protocol is used for the network interface of NX series devices?

A: TCP protocol.

Q: What is the accuracy of the timer trigger in IQS/DET/RTA modes?

A: 8 ns.

Q: Can the spurious rejection function be added in IQS/RTA modes?

A: No. The spurious rejection algorithm relies on the distribution characteristics of spurs during the frequency sweep process to identify and eliminate them. In the fixed-frequency mode of IQS/RTA, the device does not perform frequency hopping, making it impossible to utilize this characteristic for spurious rejection.

Q: In IQS/DET/RTA modes, can the analysis bandwidth be set to any value?

A: No, it cannot be set arbitrarily. The device's native sampling rate is determined by the ADC. The analysis bandwidth is adjusted by changing the Decimation Factor:

$$\text{Analysis Bandwidth} = \frac{\text{Native Sample Rate}}{\text{Decimation Factor}}$$

Q: In outdoor environments below 0 ℃, can the standard spectrum analyzers of each series operate normally?

A: The minimum operating temperature for standard spectrum analyzers is 0 ℃. Operation below this temperature cannot be guaranteed. If use in environments ranging from -20℃ to 65℃ is required, the T1 temperature extension option can be selected to ensure normal operation.

Q: Is there a specific connection order requirement for the device's power port and data port (USB/Ethernet)?

A: There is no strict requirement. As long as the power and data cables are correctly connected, devices of all series can operate normally.

Q: Can any series device support long-term outdoor operation?

A: Provided that the temperature range and power supply requirements specified in the product manual are followed, and the data transmission interface and power interface are properly connected, the device can operate outdoors for extended periods.

Q: How large is the internal data buffer of the device?

A: 128 MByte.

Q: When using level triggering, what is the low-level judgment time before the signal reaches the threshold?

A: 1 ns.

Q: When using level triggering as the trigger source, what is the function of the debounce safe time?

A: The debounce safe time is used to determine the validity of the high-level signal.

Q: What is the valid range of the trigger delay?

A: 0 to (2^32 - 1) * 8 ns = 0 to 34,359,738,360 ns (default is 0).

Q: Why is it necessary to inspect local oscillator-related spurs at multiples of 125 MHz?

A: The device's high-speed reference clock is 125 MHz, and the RF LO (Phase-Locked Loop) uses this clock as a reference. Therefore, LO-related spurs mainly originate from this clock. Spurs appear at positions offset from the carrier by N * 125 MHz.

Q: What is the "spur" signal of the device at the 125 MHz frequency multiples?

A: This is the residual response of the device. The receiver system uses a 125 MHz system clock, which generates harmonic components at N * 125 MHz. These harmonics couple into the RF input and LO paths through paths such as the chassis and circuits, resulting in residual responses at some 125 MHz frequency multiples.

Q: What is the total power consumption of the device?

A: Please refer to the following table:

Device Model	Mode	Power Port (W)	Data Port (W)	Sum(W)
E200 R3	SWP	8.6	2.53	11.13
	IQS	8.2	2.58	10.78
	DET	8.2	2.48	10.68
	RTA	8.2	4.01	12.21
E90 R3	SWP	7.3	2.46	9.76
	IQS	7.6	2.60	10.20
	DET	7.7	2.53	10.23
	RTA	7.7	4.07	11.77
M80 R5	SWP	5.1	2.02	7.12
	IQS	5.2	2.07	7.27
	DET	5.2	2.06	7.26
	RTA	5.2	2.95	8.15
M60 R5	SWP	3.5	2.09	5.59
	IQS	3.6	2.10	5.70
	DET	3.6	2.11	5.71
	RTA	3.6	2.99	6.59
N60 R5	SWP	3.0	1.87	4.87
	IQS	3.0	1.99	4.99
	DET	3.0	1.99	4.99
	RTA	3.0	2.61	5.61
N45 R5	SWP	3.0	1.73	4.73
	IQS	3.1	2.06	5.16
	DET	3.1	2.07	5.17
	RTA	3.1	2.49	5.59
N400 R2	SWP	7.5	2.57	10.07
	IQS	7.7	2.60	10.30
	DET	7.7	2.52	10.22
	RTA	7.7	4.06	11.76

Q: In the SWP mode of the software, why does the sweep speed slow down when the sweep span is reduced?

A: The device has different component configurations, switch transition times, and PLL lock times across different frequency bands. Therefore, in some cases, when the sweep span is reduced, the overall sweep speed may actually become slower.

Q: Does the data remain after the device is powered off?

A: The device itself does not have data storage capability. Data acquired will be lost after power-off.

Q: What is the polarity of the trigger output pulse?

A: The trigger output currently only supports positive pulses.

Q: What is the upper limit for pre-trigger time?

A: 8000 * 8 ns * decimation factor.

Q: What does it mean when the input power marked on the device needs to be less than 10 dBm?

A: It indicates the maximum destructive power that the device can handle in the DS segment or when the preamplifier is enabled.

Q: What is the difference between a solid and a blinking power indicator light?

A: Blinking indicates that the device's clock is out of lock; a solid light indicates normal operation.

Q: What is the meaning of DANL's dBm/Hz?

A: dBm/Hz represents the power density normalized to a 1 Hz bandwidth.

Q: Why is the analysis bandwidth equal to the sampling rate multiplied by 0.8?

A: Before acquiring IQ data and performing decimation on IQ data, anti-aliasing filtering must first be applied. The transition band of the filter is difficult to achieve a perfectly steep cutoff, so only 80% of the effective bandwidth is retained.

Q: In DET mode, what is the time interval between each data point?

A: The time interval is related to the sampling rate and the decimation factor.

$$T = \frac{\text{Decimation Factor}}{\text{Sample Rate}}$$

Where:
T: Time interval between adjacent points (seconds)
Sample Rate: Native sampling rate, e.g., 125 MHz
Decimation Factor: 1, 2, 4 ... up to a maximum of 4096

Q: When testing the device, does the RFIN require an external DC block?

A: If the test signal contains or it is uncertain whether it contains a DC voltage exceeding the maximum allowed input of the device, it is recommended to use an external DC block to protect the device.

1.4 Hardware Feature

Q: Is the device coupling method adjustable?

A: The device uses AC coupling, which is a fixed configuration and does not support adjustment.

Q: Does the power supply for NX devices need to comply with the PD3.0 protocol?

A: If using an adapter for power supply, it must comply with the PD3.0 protocol; if using a direct current source for power supply, it does not need to comply with PD3.0.

Q: Does the core module undergo three-proof treatment?

A: No, users need to implement corresponding protection in their integrated structure.

Q: What is the power consumption of a single fan in the device？

A: The power consumption of a single small fan is 5V × 300mA = 1.5W.

Q: Do all device series have ESD (Electrostatic Discharge) protection？

A: Yes, details are as follows:

1. RF input port

- N45 / N60 / M60 / M80 / E90: 4 kV protection
- E200: Class 2C (4 kV protection)
- N400: Class 1C (8 kV protection)

2. SA series

- Data port: 2 kV protection
- Power port: 2 kV protection

3. NX series

- Ethernet port: 2 kV protection
- Power port: 30 kV protection

4. PX series

- USB data port: 2 kV protection
- Power port: 30 kV protection

Q: By default, at what temperature do the fans turn on in SA / NX series devices？

A:

1. SAN / SAM series devices (excluding SAN-400):

- When using software: Fans automatically turn on at ≥50 °C; turn off at ≤40 °C.
- When using API: Fans automatically turn on at ≥50 °C; turn off at ≤40 °C.

2. SAE series / SAN-400 devices:

- Fans turn on automatically when the device is powered on.

3. NX series devices:

Fans turn on automatically when the device is powered on

Q: How to obtain the timestamp for each time-domain IQ point?

A:

Taking the timestamp of the first IQ point as the current GNSS second-level time, the timestamp of the Nth point is:

$$T_N = T_{GNSS} + N * T$$

Where:

T: Time interval between two points;

N: The Nth IQ point, counting from 0;

T_{GNSS}: The starting time of GNSS.

Q: What positioning systems and frequency bands does the provided GNSS board support?

A: Supported positioning systems and frequency bands: L26-T, M8T: GPS (L1C/A), Galileo (E1), BDS (B1I), QZSS (L1C/A).

Q: How is the data transfer rate of the device's USB interface calculated?

A: Native Sample Rate / Decimation Factor * 2 (for I and Q channels) * Number of Data Bytes.

Q: Can the power supply current of the host PC's motherboard USB port affect device usage?

A: As long as the power supply current of the motherboard's USB port complies with the USB 2.0 or USB 3.0 standard specifications, it will not affect the normal operation of the device.

2. Troubleshooting Guide

If the instrument does not perform as expected, please refer to the following steps for troubleshooting. If the problem persists, please contact official technical support.

2.1 Driver Installation Failed

Symptom:

After completing the driver installation according to the installation procedure, if the target device shows a yellow exclamation mark in the Device Manager, it indicates that the instrument is not functioning properly and cannot be recognized or used.

Root Cause:

Driver installation failed.

Resolution Steps:

1. In Device Manager, right-click the driver related to "HTRA Devices" and select "Uninstall device";

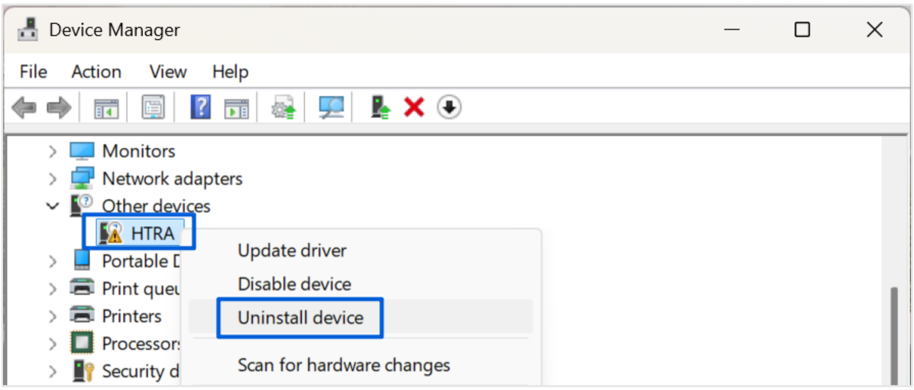


Figure 1 Select to uninstall HTRA Devices driver

2. In the pop-up dialog box, check the option "Attempt to delete the driver for this device" and then click "Uninstall";

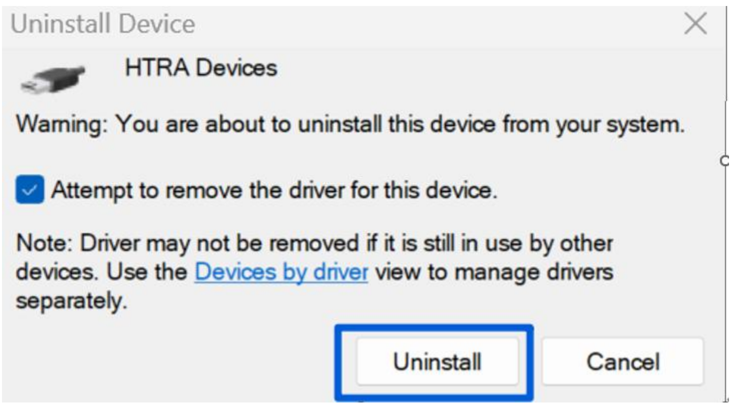


Figure 2 Uninstall Driver

3. After the driver uninstallation is complete, please re-plug the instrument's data cable, and then click the "Scan for hardware changes" icon at the top of the Device Manager. At this point, the instrument will usually appear under the "Other devices" branch. Right-click on the "HTRA" device and select "Update driver";

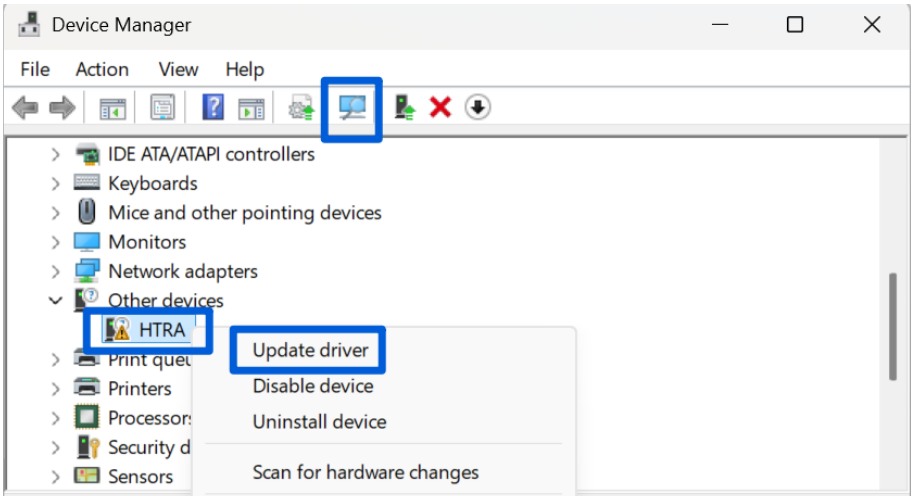


Figure 3 Manually update the driver

4. If "Other devices" does not appear, click "Action" on the top menu of Device Manager and select "Add driver" to manually install the driver;

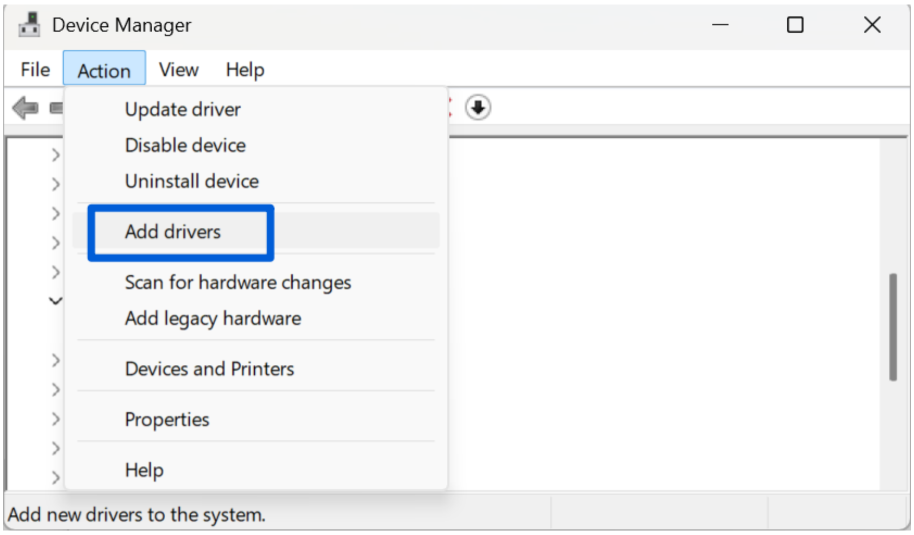


Figure 4 Manually add the driver

5. In pop-up window, click "Browse", select the \Windows\HTRA_Driver\Win10_x64 folder on the supplied USB drive, and click Next;

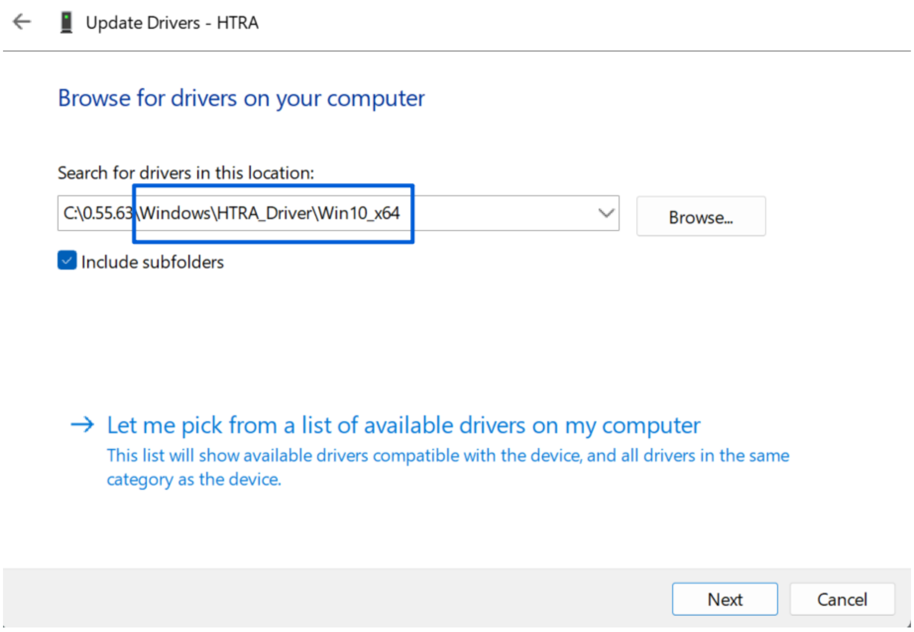


Figure 5 Select the driver file

6. The system will begin installing the driver. Please wait for the installation to complete, then close the window;
7. Reconnect the instrument, open "Universal Serial Bus controllers" in Device Manager, and the device driver should now be correctly recognized and displayed;

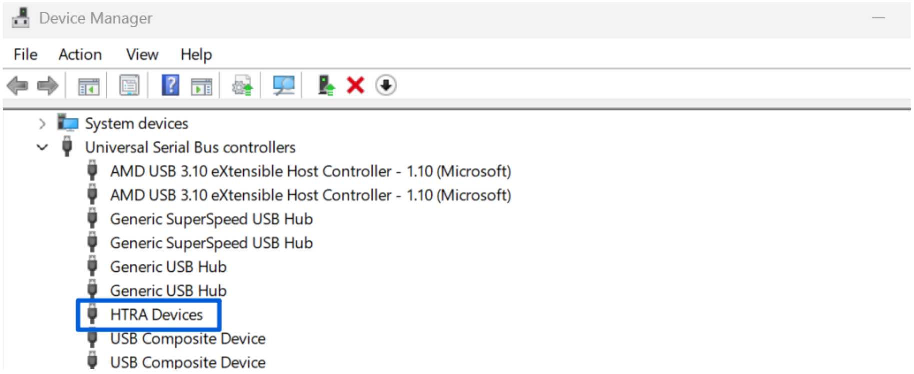


Figure 6 Driver installation successful

Note: On Windows 7 64-bit systems, if the above steps do not work, please contact technical support to obtain and install the kb4474419 and kb4490628 updates in sequence (be sure to follow this order). After restarting the computer, you will need to manually reinstall the drive.

2.2 Instrument Driver Not Recognized After Windows Update

Symptom:

After updating the host PC from Windows 10 to Windows 11, the device driver is no longer recognized.

Resolution Steps:

1. Refer to Steps 1 to 2 in the Driver Installation Failed section to uninstall the existing driver;
2. Install the driver by following Section 3.2.2 or 3.2.3 of the HAROGIC Quick Start Guide for Spectrum Analyzer.

2.3 Error Code -1

Symptom:

1. The API call to Device_Open returns error code -1;
2. The software interface displays "Bus open error -1".

Root Cause:

The instrument cannot initialize or connect properly. Possible causes include:

1. The driver is not installed correctly;
2. Abnormalities in the instrument's power supply, cables, or physical interfaces;
3. Incorrect software configuration parameters (e.g., DeviceNum, interface type, IP address);
4. The instrument is occupied by another process.

Resolution Steps:

2.3.1 SA Series Instruments

1. Check the instrument's connection status
   - Refer to Section3.2.1 of the HAROGIC Quick Start Guide for Spectrum Analyzer to connect the instrument.
   - Windows: Open "Universal Serial Bus controllers" in Device Manager. If an HTRA-related device appears, the instrument is connected successfully. If not, check the USB cable, interface, or try a different USB port.
   - Linux: Run the lsusb command in the terminal to verify the instrument connection. If it does not appear, check the USB cable, interface, or try a different USB port, and ensure the virtual machine's USB compatibility is set to USB3.1.

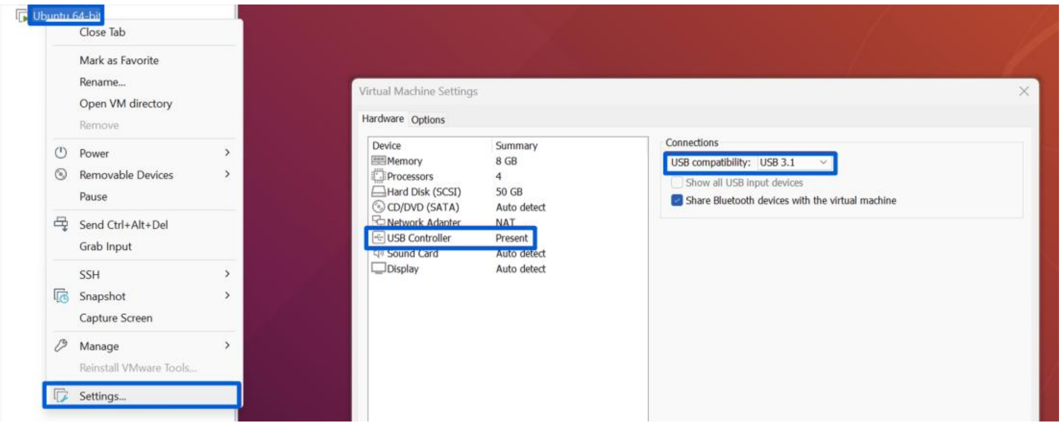


Figure 7 Set the virtual machine's USB compatibility

2. Check the driver installation
   - Windows: Ensure the driver is installed following Section3.2.2 of the HAROGIC Quick Start Guide for Spectrum Analyzer. In Device Manager, check "Universal Serial Bus controllers" or "Other devices." If the HTRA driver shows a yellow exclamation mark, reinstall the driver according to the "Driver Installation Failure" section.
   - Linux: Ensure the driver is installed following Section3.2.3 of the HAROGIC Quick Start Guide for Spectrum Analyzer.

     You can check if the installation was successful using the following command:

     ```
     ls /etc/ | grep htrausb.conf
     ls /etc/udev/rules.d/ | grep htra-cyusb.rules
     ```

3. Check for instrument occupancy
   Ensure that no other programs or companion software are occupying the instrument. If any are running, close them and try again.
4. Configure the interface parameters
   When using multiple instruments simultaneously, refer to the [Multiple Instruments Usage Troubleshooting] section for interface parameter configuration.

2.3.2 NX Series Instruments

1. Wait for the instrument to be ready:
   For NX series instruments, wait approximately 50seconds after power-on. Only use the software or call the API once the system has fully started.
2. Check the power supply:
   Ensure that the original power adapter is used. If unavailable, replace it with an adapter that supports 9V to 12V and a peak current of 2A.
3. Check for instrument occupancy:
   Ensure that no other programs or companion software are occupying the instrument. If any are running, close them and try again.
4. Network Connection and Configuration:
   Follow Chapter4 of the [HAROGIC Quick Start Guide for Spectrum Analyzer] to correctly connect the instrument and configure the network step by step. The instrument's default IP address is 192.168.1.100; if the IP has been changed, set it to the corresponding new address.
   - For API developers: set IPAddress to the instrument's IP address and PhysicalInterface = ETH;

```
// NX series model
BootProfile.PhysicalInterface = ETH;
BootProfile.ETH_IPVersion = IPv4;
BootProfile.ETH_RemotePort = 5000;
BootProfile.ETH_ReadTime0ut = 10000
//Configure IP address for NX series, default address is 192.168. 1. 100
BootProfile.ETH_IPAddress[0] = 192;
BootProfile.ETH_IPAddress[1] = 168;
BootProfile.ETH_IPAddress[2] = 1;
BootProfile.ETH_IPAddress[3] = 100;
```

   - For software interface users: open the configuration\Setting.ini file in the software installation directory, set Interface = ETH, and set Address to the instrument's IP address.
5. Resolve IP Conflicts:
   When the host PC has multiple network cards, ensure that the network card communicating with the NX instrument is on the same subnet as the instrument. For example, if the

instrument's IP is set to 192.168.1.100, there must be only one network card with an IP in the 192.168.1.x range; otherwise, an IP conflict will occur. If a conflict exists, follow the Device_GetAndSetIP.cpp sample file provided on the supplied USB drive to set the instrument and the connected network card to a new, independent subnet.

6. Check the hardware connections:

   If issues persist, try replacing the network cable or host PC. When using a dock's network port for data transmission, ensure it is functioning properly. Note that when using NX series instruments, changing the dock requires resetting the local IP address.

2.3.3 NX Series Instruments

1. Restart the instrument

   Try restarting the PX instrument and ensure it has sufficient power and is not in a low-battery state.

2. Check the startup status

   Ensure the instrument completes its self-startup and avoid manually opening the software during this process.

2.4 Driver Installation Failed

Symptom:

1. The API call to Device_Open returns error code -3, -4, or -43;
2. The software interface displays "CalFileLoss!".

Root Cause:

The instrument's calibration file is missing or mismatched, preventing the instrument from opening properly.

Resolution Steps:

1. Obtain the calibration file

   Copy all files from the CalFile folder in the root directory of the USB drive supplied with the instrument. If the USB drive is unavailable, contact technical support to obtain the files.

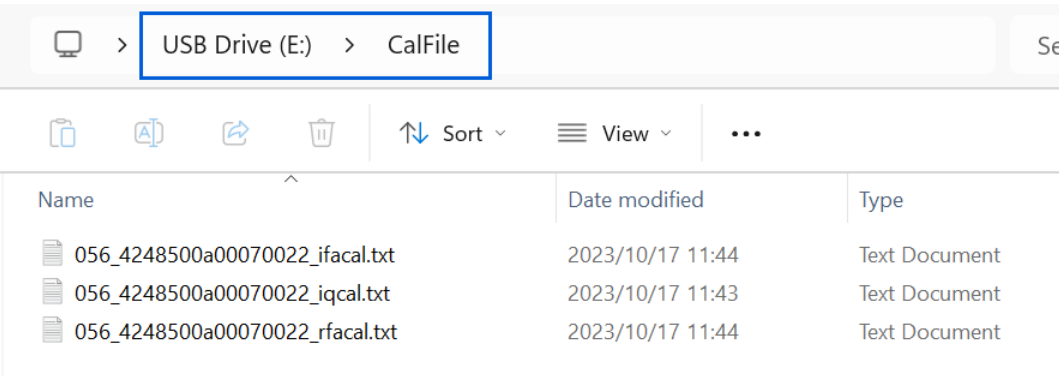


Figure 8 Calibration File Location

2. Place the calibration files

   - Software interface: Paste the calibration files into the \bin\CalFile folder in the software installation directory.

- Windows API programming: Paste all copied calibration files into the CalFile folder located in the same directory as the htra_api.dll library.

Example: When using the supplied C++ sample, the placement path can follow the illustration below. For other programming languages (e.g., C#), follow the same principle.

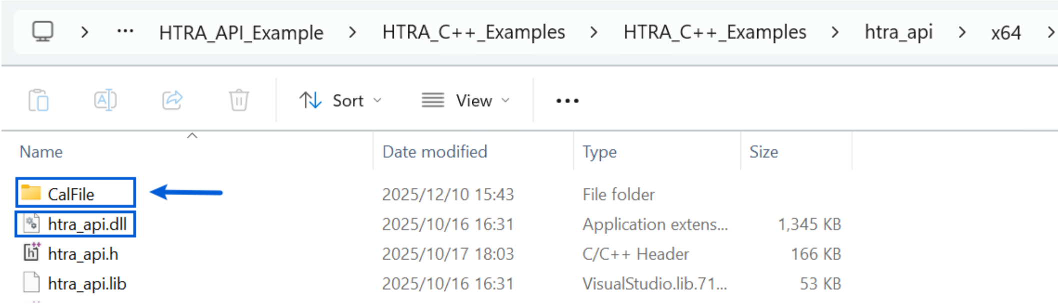


Figure 9 Calibration File Placement Path on Windows

- Linux API programming: Paste all copied calibration files into the CalFile folder under the bin directory.

For Python examples, copy the CalFile folder and all its contents to the current working directory of your Python interpreter. For detailed instructions, refer to the Python section in the HAROGIC HTRA API Examples Usage Guide.

Example: When using the supplied Qt sample, the placement path can follow the illustration below.

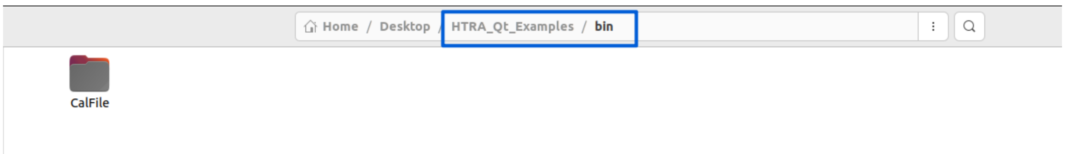


Figure 10 Qt Example Calibration File Placement Path

3. Place the calibration files

   If the calibration files are placed in the correct path but the issue persists, please verify that the name of the calibration file matches the instrument's UID number exactly. If they do not match, you will need to copy the correct files from the supplied USB drive or contact technical support to obtain the correct files.

2.5 Error Code -8

Symptom:

1. The API call to Device_Open returns error code -8;
2. The software displays a bus communication error.

Root Cause:

Power supply issues prevent the instrument from opening properly.

Resolution Steps:

2.5.1 SA Series Instruments

1. Follow the instructions in Sections3.1.1 and 3.2.1 of the HAROGIC Quick Start Guide for Spectrum Analyzer, use the original power adapter, and reconnect the instrument.

   Note: Avoid using the computer's USB port to power the instrument.
2. Check the instrument's indicator light status

   If the indicator light is steady, it means the connection is successful. If it is not lit, please replace the power adapter or power cable and try again.

- For SAE and SAN-400 series instruments, check the indicator light on the multifunctional interface;

Figure 11 Indicator light of SAE and SAN-400 series instruments

- For SAM and SAN series instruments, check the indicator light at the reference clock input;

Figure 12 Indicator light of SAM and SAN series instruments

3. Confirm whether any other programs or companion software are occupying the instrument. If so, please close them.

2.5.2 NX Series Instruments

1. Follow the instructions in Sections4.1.1 and 4.2.1 of the HAROGIC Quick Start Guide for Spectrum Analyzer, use the original power adapter, and reconnect the instrument;
   Note: Avoid using the computer's USB port to power the instrument.
2. Check the instrument's indicator light status;
   Check if the indicator light on the instrument’s multifunctional interface is steady. A steady light indicates a successful connection. If the light is off, please replace the power adapter or power cable and try again.

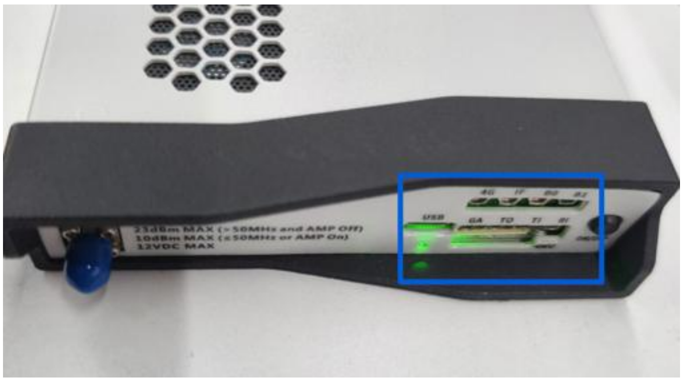


Figure 13 Indicator light of NX series instruments

3. Confirm whether any other programs or companion software are occupying the instrument. If so, please close them.

2.5.3 PX Series Instruments

Try restarting the PX instrument and ensure it has sufficient power and is not in a low-battery state.

2.5.4 Raspberry Pi 4B

The Raspberry Pi 4B uses the aarch64 (64-bit) architecture. Installing a 32-bit system may cause byte alignment issues, preventing the instrument from functioning. Please reinstall a 64-bit system on the Raspberry Pi 4B.

2.6 Error Codes -7, -9 or -11

Symptom:

1. The API call to any function returns error codes -7, -9, or -11;
2. The software pop-up indicates "Failed to send policy to instrument -7," "Data content error -9," or "Bus configuration error -11".

Root Cause:

Data anomalies are preventing the instrument from functioning properly. Possible causes include:

1. The instrument is occupied;
2. Unstable connection;
3. Insufficient power supply;
4. Conflict when connecting multiple instruments.

Resolution Steps:

2.6.1 Basic Troubleshooting

1. Close the program occupying the instrument
   Check if any other programs or companion software are using the instrument. If so, please close them first;
2. Reconnect the instrument
   If the issue persists after closing the programs, try disconnecting and reconnecting the instrument;
3. Restart the host PC and try a different interface
   If the issue persists, restart the host PC and use a different USB 3.0 port to connect the instrument;

4. Check the power supply

If the instrument occasionally reports error -9 during use, it may be due to insufficient power supply. Please confirm the following:

- SA series instruments: Power supply requirement is 5V/2A;
- NX series instruments: Power supply requirement is 12V/2A;
- It is recommended to use the original power adapter and avoid using the computer's USB port for power.

2.6.2 Multiple Instruments Usage Troubleshooting

If this error occurs while using multiple instruments simultaneously, please check the instrument configuration as follows:

1. SA Series Instruments

- Multiple programs running independently: Ensure that the DevNum parameter for each program is set to a different value.

```
int Status = 0;
void* Device = NULL;
int DevNum =0;
```

```
int Status = 0;
void* Device = NULL;
int DevNum =1;
```

- Single program controlling multiple instruments: As shown in the diagram, configure the Status, Device, DeviceNum, and related structures and functions separately for each instrument, and ensure that each instrument is assigned a unique DevNum value. For a complete code implementation, refer to the IQS_MultiDevSync_fixed.cpp example provided on the supplied USB drive.
- Software interface: Prepare separate companion software for each instrument. As shown in the diagram, set the DeviceNum to different values in the configuration\Setting.ini file under each companion software installation directory to open different instruments.

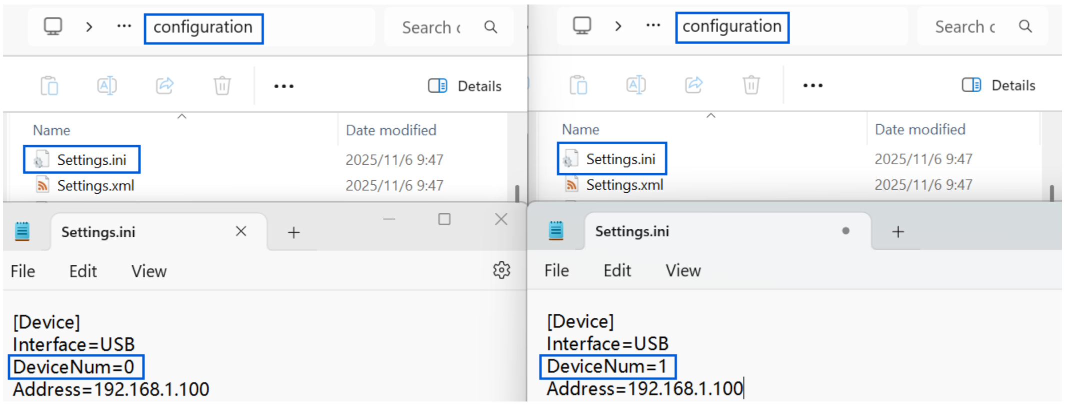


Figure 14 Setting the DeviceNum value when using multiple SA series instruments

2. NX Series Instruments
   - Multiple programs running independently: Refer to the Device_GetAndSetIP.cpp example and ensure that the IPAddress parameter for each program is set to a different value.
   - Single program controlling multiple instruments: Refer to the following code to configure the Status, Device, IPAddress, and related structures and functions separately for each instrument.

```
int Status = 0;
uint8_t DeviceCount;
// Device information, including IP address, subnet mask, UID, etc.
NetworkDeviceInfo_TypeDef NetworkDeviceInfo[100];

uint8_t LocalIP[4];
uint8_t LocalMask[4];
// Get the IP addresses, subnet masks, and other information of all network
devices in the network.
Status = Device_GetNetworkDeviceList(&DeviceCount, NetworkDeviceInfo,
LocalIP, LocalMask);

int Status0 = 0, Status1 = 0;
void* Device0 = NULL;
void* Device1 = NULL;
int DevNum0 = 0, DevNum1 = 0;

BootProfile_TypeDef BootProfile0, BootProfile1;
BootInfo_TypeDef BootInfo0, BootInfo1;
BootProfile0.DevicePowerSupply = Others;
BootProfile1.DevicePowerSupply = Others;

// Configure device 0.
BootProfile0.PhysicalInterface = ETH;
BootProfile0.ETH_IPVersion = IPv4;
BootProfile0.ETH_RemotePort = 5000;
BootProfile0.ETH_ReadTimeOut = 5000;
BootProfile0.ETH_IPAddress[0] = NetworkDeviceInfo[0].IPAddress[0];
BootProfile0.ETH_IPAddress[1] = NetworkDeviceInfo[0].IPAddress[1];
BootProfile0.ETH_IPAddress[2] = NetworkDeviceInfo[0].IPAddress[2];
BootProfile0.ETH_IPAddress[3] = NetworkDeviceInfo[0].IPAddress[3];

// Configure device 1.
BootProfile1.PhysicalInterface = ETH;
BootProfile1.ETH_IPVersion = IPv4;
BootProfile1.ETH_RemotePort = 5000;
```

```
BootProfile1.ETH_ReadTimeOut = 5000;
BootProfile1.ETH_IPAddress[0] = NetworkDeviceInfo[0].IPAddress[0];
BootProfile1.ETH_IPAddress[1] = NetworkDeviceInfo[0].IPAddress[1];
BootProfile1.ETH_IPAddress[2] = NetworkDeviceInfo[0].IPAddress[2];
BootProfile1.ETH_IPAddress[3] = NetworkDeviceInfo[0].IPAddress[3];

Status0 = Device_Open(&Device0, DevNum0, &BootProfile0, &BootInfo0);
// Open device 0.
Status1 = Device_Open(&Device1, DevNum1, &BootProfile1, &BootInfo1); //
Open device 1.
```

- Software interface: Prepare separate companion software for each instrument. As shown in the diagram, set the Address to different values in the configuration\Setting.ini file under each companion software installation directory to open different instruments.

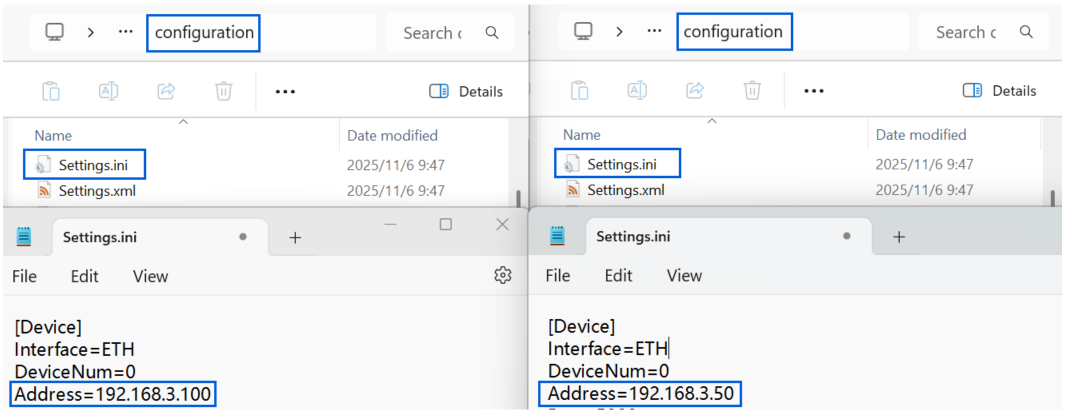


Figure 15 Modify the IP address when using multiple NX series instruments

2.7 Error Codes -10

Symptom:

1. In IQS, DET, and RTA modes, when the trigger source is set to "Bus Trigger" (TriggerSource = Bus), the instrument opens normally, but the API call to the Get function returns error code -10;
2. When using the software's disk streaming function, error code -10 occurs during the process when the analysis bandwidth is ≤ sample rate / 2;
3. In the software's DET mode, error code -10 occurs when performing a full sweep width, level triggering, and the input signal is a pulse signal.

Root Cause:

Data acquisition timeout.

Resolution Steps:

1. Check the physical connection
   - SA series instruments: Use a USB 3.0 cable to connect the instrument to the USB 3.0 port on the host PC;
   - NX series instruments: Use a Gigabit Ethernet cable to connect the instrument’s Gigabit Ethernet port to the host PC’s Gigabit Ethernet port.

2. Confirm the trigger function call sequence
   Before acquiring data, the trigger function IQS_BusTriggerStart must be called correctly. The calling logic for the two trigger modes is as follows:
   - Adaptive mode: First, call IQS_BusTriggerStart, then enter a loop to acquire data.

```
// Adaptive
Status = IQS_BusTriggerStart(&Device);
while (1)
{
Status = IQS_GetIQStream_PM1(&Device, &IQStream);
}
```

   - FixedPoints mode: Call IQS_BusTriggerStart once to collect data of a fixedpoint length.

```
// FixedPoints
while (1)
{
Status = IQS_BusTriggerStart(&Device);
for (int j = 0; j < StreamInfo.PacketCount; j++) {
Status = IQS_GetIQStream_PM1(&Device, &IQStream);
}
}
```

3. Check the decimation factor settings
   In the Adaptive mode of IQS and DET, the decimation factor must meet the following requirements:
   - SA Series Instruments: DecimateFactor ≥ 2 (SAN-45 and SAN-60 can be ≥ 1);
   - NX Series Instruments: DecimateFactor ≥ 16 (NXN-45 can be ≥ 1, NXN-60 can be ≥ When used in Linux, please appropriately increase the decimation factor according to the host computer's performance.
4. Optimize the data acquisition thread
   If the Adaptive mode still operates abnormally, ensure that the Get function runs in a dedicated thread, and that there are no other computational or blocking operations within that thread.
   For details, please refer to the example file "IQS_Multithread_GetIQ_FFT_Write.cpp" in the shipped USB drive.
5. In the software's DET mode, if the input is a pulsed signal with a short period, its data characteristics under a full span approximate a continuous stream, which may normally trigger error code -10 due to exceeding the USB 3.0 transmission bandwidth limit. It is recommended to try reducing the span or increasing the period of the pulsed signal.

2.8 Error Codes -12

Symptom:

1. The API call to the Get function returned error code -12;
2. The spectrum display in the software interface is abnormal, and the status bar on the right shows "Overflow!".

Root Cause:

The signal power exceeds the instrument's reference level, causing saturation in the intermediate frequency stage.

Resolution Steps:

1. Test a single-tone signal
   Please appropriately increase the reference level (RefLevel_dBm) to be higher than the power of the input signal.
2. Test a modulated signal
   If the reference level is already higher than the signal peak, but error -12 is still reported, continue to appropriately increase the reference level until the error disappear.
   Reason: A modulated signal typically contains multiple frequency components. Although each component may not exceed the ADC's acquisition range, their combined amplitude may still exceed the acquisition range at certain moments due to time-domain superposition, leading to ADC saturation.

Note: Avoid prolonged operation under intermediate frequency saturation (error -12) to prevent damage to the instrument's internal hardware.

2.9 Error Codes 10054, 10060 or 10062

Symptom:

1. API calls to any function return error codes 10054, 10060, or 10062;
2. The software spectrum refresh stops, and a pop-up window indicates that the device has disconnected from the network (error 10054), the connection attempt has failed (error 10060), or the device did not acquire data normally (error 10062).

Root Cause:

Network communication anomalies in the NX series instruments result in connection interruptions or data transmission failures between the instrument and the host computer.

Resolution Steps:

2.9.1 API Development Users

1. The program has added an exception handling mechanism:
   When it is detected that the instrument has been unresponsive for an extended period and returns the aforementioned errors, the following process is automatically executed:
   - Call the Device_Close function to close the instrument;
   - Loop by calling the Device_Open function until it returns 0 (indicating the instrument has been successfully reopened);
   - Re-deliver the configuration parameters;
   - For specific details, refer to the Error_handling.cpp example included in the accompanying USB drive.

2. Manual connection recovery: Manually unplug and replug the network cable to re-establish the instrument connection (the first method is recommended).

2.9.2 Software Interface User

1. Wait briefly:
   This issue is often caused by network instability. Click "Cancel," wait for a moment, and the connection will usually restore automatically.
2. Check the physical connection:
   If the connection cannot be restored after a long time, try the following actions:
   - Reconnect the network cable to ensure the network is stable;
   - Replace the network cable or switch to a different port.

2.10 Driver Installation Failed

Symptom:

When calling dynamic link libraries in a Linux system, an error may occur indicating that libraries such as libliquid.so, libhtraapi.so, or libusb.so cannot be found.

Root Cause:

1. The library files are not placed in a path that the executable can recognize;
2. The architecture of the library files does not match the system architecture;
3. The dynamically linked libraries with version numbers have not been correctly linked with symbolic links.

Resolution Steps:

1. Confirm that the library file paths are correct:
   Please ensure that the dynamic link libraries are stored in the correct target path during the compilation or build of the executable. For reference, you can check the Makefile in the Linux\HTRA_C++_Examples folder on the supplied USB drive.
2. Confirm that the architecture of the library files matches the system architecture:
   - Run the command uname -a in the terminal to check the system architecture;

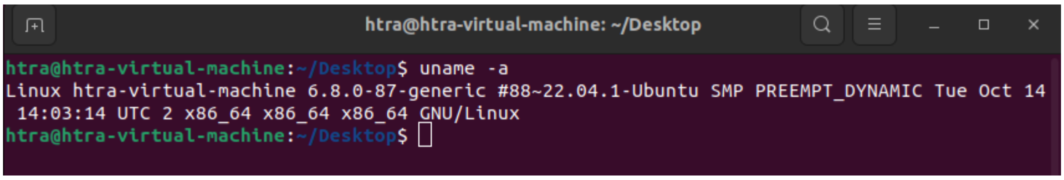


Figure 16 Check the Linux system architecture

   - Based on the system architecture, use the library files from the corresponding architecture directory on the supplied USB drive.
     Example: If the system architecture shows as x86_64, you should use the library files from the x86_64 or x86_64_gcc5.4 directory.

Note: The x86_64_gcc5.4 directory requires the system's GCC version to be 5.4 or higher.

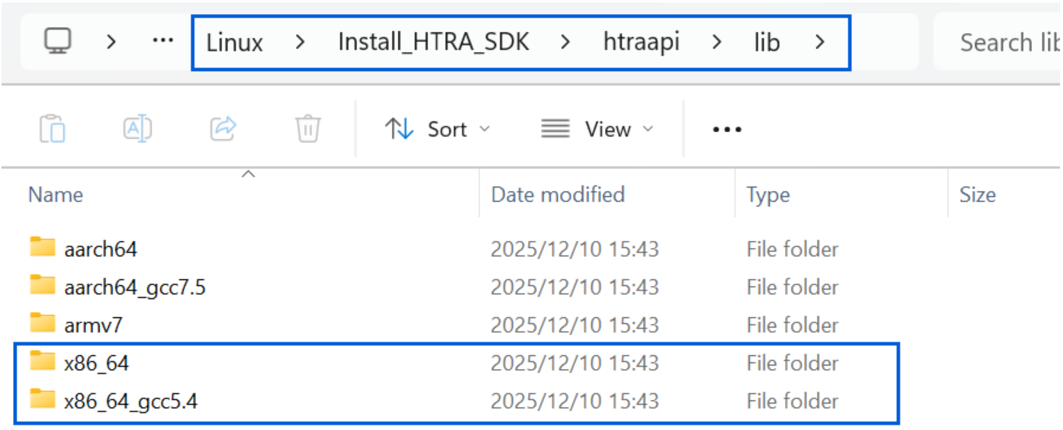


Figure 17 Library files corresponding to the Linux x86_64 architecture

3. Create the correct symbolic links:

   For example, with the 0.55.63 version API, execute the following commands in the directory where the library files are located to create the symbolic links:

```
ln -sf libhtraapi.so.0.55.63 libhtraapi.so.0
ln -sf libhtraapi.so.0 libhtraapi.so
ln -sf libusb-1.0.so.0.2.0 libusb-1.0.so.0
ln -sf libusb-1.0.so.0 libusb-1.0.so
ln -sf libgomp.so.1.0.0 libgomp.so.1
ln -sf libgomp.so.1 libgomp.so
```

   After executing the commands, the generated files will be as shown in the diagram below.

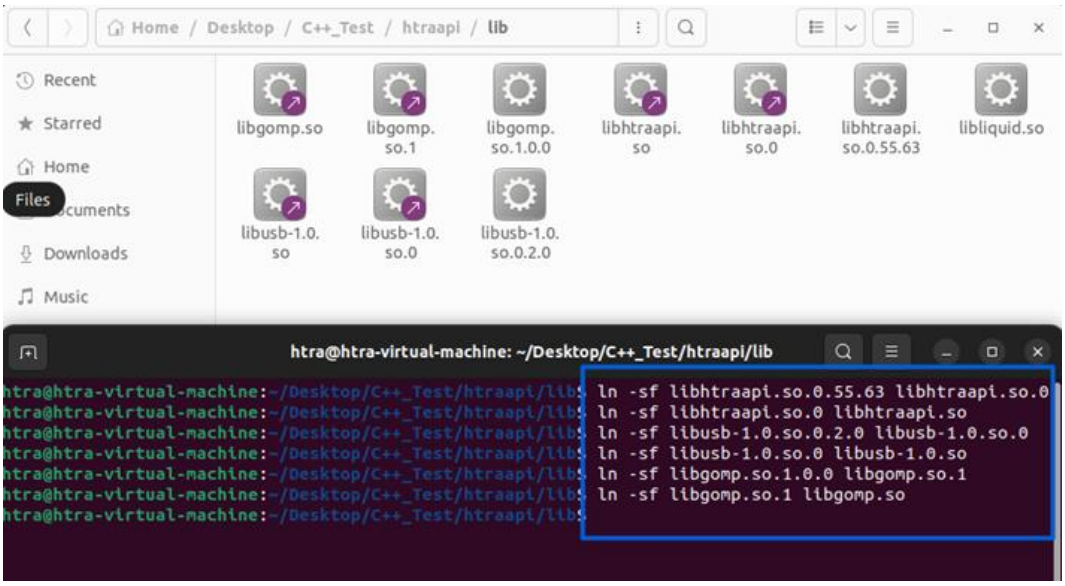


Figure 18 Symbolic links for dynamic link libraries

4. Confirm permission settings:

   Ensure the library files have executable permissions; you can grant permission using the command chmod +x *.so.

2.11 Parameters sent in SWP mode do not take effect

2.11.1 RBW or VBW

Symptom:

In SWP mode, the settings do not take effect when sending RBW or VBW parameters.

Root Cause:

The update method for RBW/VBW is not set to manual mode.

Resolution Steps:

1. Synchronize the setting of RBW Mode and VBW Mode to manual input mode.

```
SWP_ProfileIn.RBW_Hz = 300e3;
SWP_ProfileIn.RBWMode = RBW_Manual;
SWP_ProfileIn.VBW_Hz = 500e3;
SWP_ProfileIn.VBWMode = VBW_Manual;
```

2.11.2 Center Frequency and Span

Symptom:

In SWP mode, the settings do not take effect when sending Center Frequency (CenterFreq_Hz) and Span (Span_Hz) parameters.

Root Cause:

The frequency specification method is not set to CenterSpan mode.

Resolution Steps:

1. It is necessary to simultaneously set the Trace Detector Mode (TraceDetectMode) to the specified trace detection mode (TraceDetectMode_Manual).

```
SWP_ProfileIn.TraceDetector = TraceDetector_Bypass;
SWP_ProfileIn.TraceDetectMode = TraceDetectMode_Manual;
```

2.11.3 Trace Detector

Symptom:

In SWP mode, the settings do not take effect when sending the Trace Detector (TraceDetector) parameter.

Root Cause:

The Trace Detector mode is not set to manual specification mode.

Resolution Steps:

1. It is necessary to simultaneously set the Trace Detector Mode (TraceDetectMode) to the specified trace detection mode (TraceDetectMode_Manual).

```
SWP_ProfileIn.TraceDetector = TraceDetector_Bypass;
SWP_ProfileIn.TraceDetectMode = TraceDetectMode_Manual;
```

2.11.4 Trace Bin Size

Symptom:

In SWP mode, the settings do not take effect when sending the Trace Bin Size (TraceBinSize_Hz) parameter.

Root Cause:

The Trace Bin Size (or Frequency Spacing) does not support direct manual modification by default.

Resolution Steps:

1. Currently, the Trace Bin Size (TraceBinSize_Hz) defaults to -1, which means the instrument calculates the spacing automatically. To fully specify the Trace Bin Size, please refer to solution step 2 in the Spectrum acquired in SWP mode is slightly wider than specified interval.

2.12 Spectrum in SWP mode slightly wider than specified

Symptom:

When acquiring spectrum data in SWP mode, the actual acquired spectrum range is slightly larger than the configured start and stop frequencies.

Example: The configured frequency interval is 1 GHz to 2 GHz, but the actual acquired frequency range is 994 MHz to 2.007 GHz.

Root Cause:

This is a normal phenomenon under the two trace point strategies: "SweepSpeedPreferred" and PointsAccuracyPreferred. To ensure no spectrum information is lost, the instrument defaults to acquiring spectrum data slightly wider than the configured interval.

Resolution Steps:

1. Spectrum Intercept:

   To acquire spectrum data within a specified interval, the Spectrum Intercept interface can be called after data acquisition, as shown in the figure below;

   Note: The actual start and stop frequencies after intercept will not be exactly equal to the commanded values, but rather the closest values that enclose the commanded values.

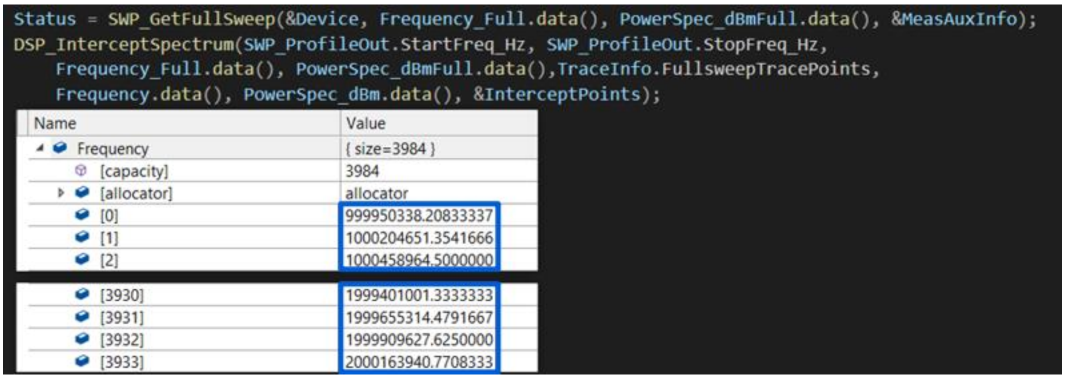


Figure 19 SWP Mode Spectrum Intercept Example

2. Use the specified TraceBinSize mode:

   In addition to the Spectrum Capture method, the trace point strategy can also use BinSizeAssigned. The characteristics of this mode are as follows:

   - The Start Frequency, Stop Frequency, and the number of trace points (FullsweepTracePoints) will be strictly executed according to the commanded values;
   - The sweep speed is relatively slower;
   - Trace Bin Size = Span_Hz / (TracePoints - 1).

2.13 SWP mode data acquisition delay

Symptom:

When acquiring spectrum data in SWP mode, the host machine display exhibits a delay of several seconds. As shown in the figure below , the signal input has been turned off, but the plotted spectrum trace still shows the signal

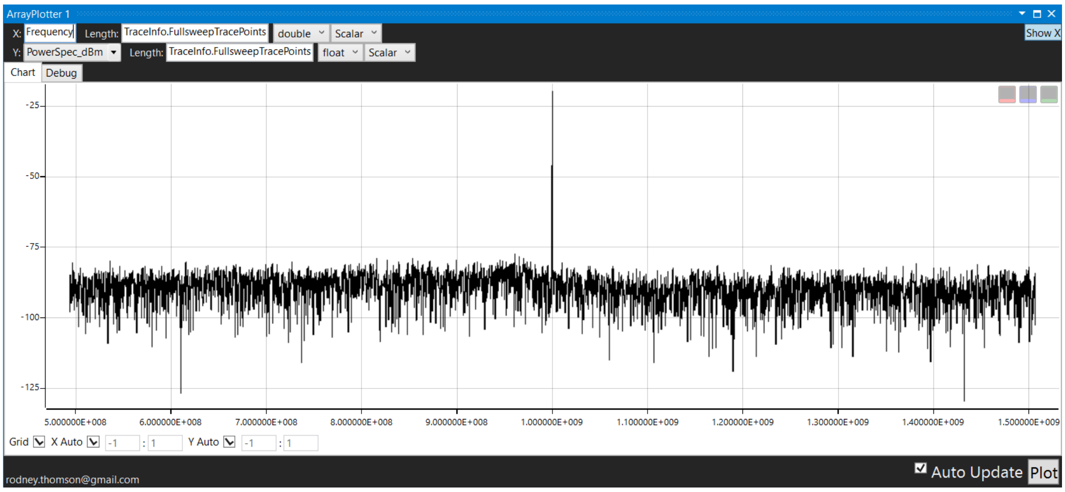


Figure 20 SWP mode data acquisition delay phenomenon

Root Cause:

Data display delay is usually caused by one or more of the following reasons:

1. Data Processing Blockage:

   Delay operations (such as Sleep, Delay, etc.) or complex calculations following the Get function block the data acquisition loop.

2. Single-Thread Architecture Limitations:

   Executing operations like "data acquisition, data processing, and image plotting" sequentially within the same thread results in an extended acquisition cycle.

3. Insufficient Host Machine Performance:

   NX series instruments have a built-in data buffer. If the host machine's data processing speed is lower than the instrument's data output speed, data will accumulate in the buffer, causing display latency.

Resolution Steps:

1. Check and Remove Delay Functions:

   Ensure that no delay functions, such as sleep(), are used after the Get function.

2. Adopt a Multi-threaded Architecture:

   Separate data acquisition, data processing, and graph plotting into different threads.

3. If the NX instrument still exhibits latency after fully satisfying the preceding conditions, the host machine performance may be insufficient. It is recommended to mitigate this issue by upgrading the host machine's configuration.

2.14 Converted spectrum from IQ data differs from SWP mode

Symptom:

When converting the acquired IQ data into a spectrum, the displayed spectrum trace exhibits a clear difference compared to the spectrum observed directly in SWP mode, specifically manifesting as:

- Signal Amplitude Difference: For example, when inputting a 1 GHz, -20 dBm signal, the signal amplitude displayed in SWP mode is approximately -20 dBm (as shown in Figure 21), where the spectrum after IQ conversion displays approximately -50 dBm (as shown in Figure 22).
- Sideband Effect: Unwanted sidebands appear on both sides of the spectrum after IQ conversion (as shown in Figure 22).

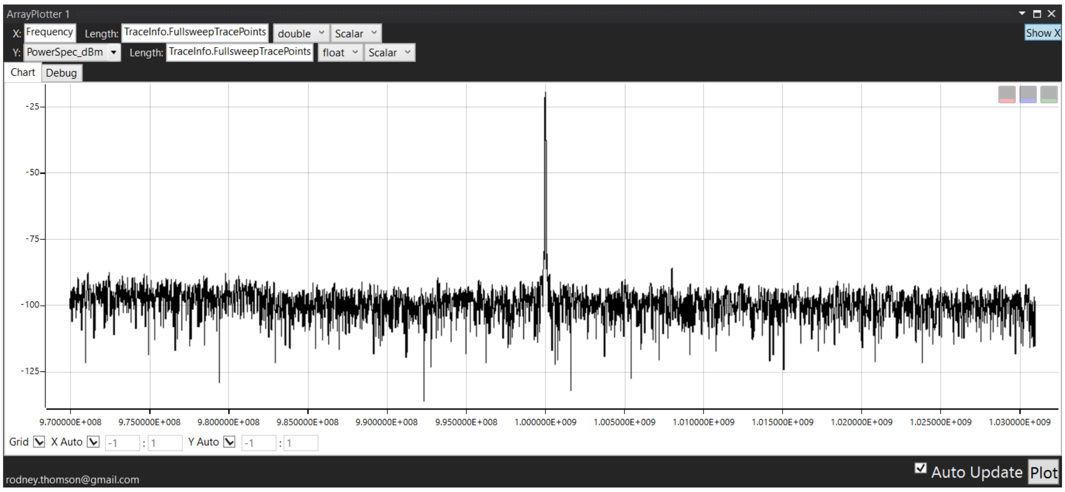


Figure 21 Spectrum trace in SWP mode

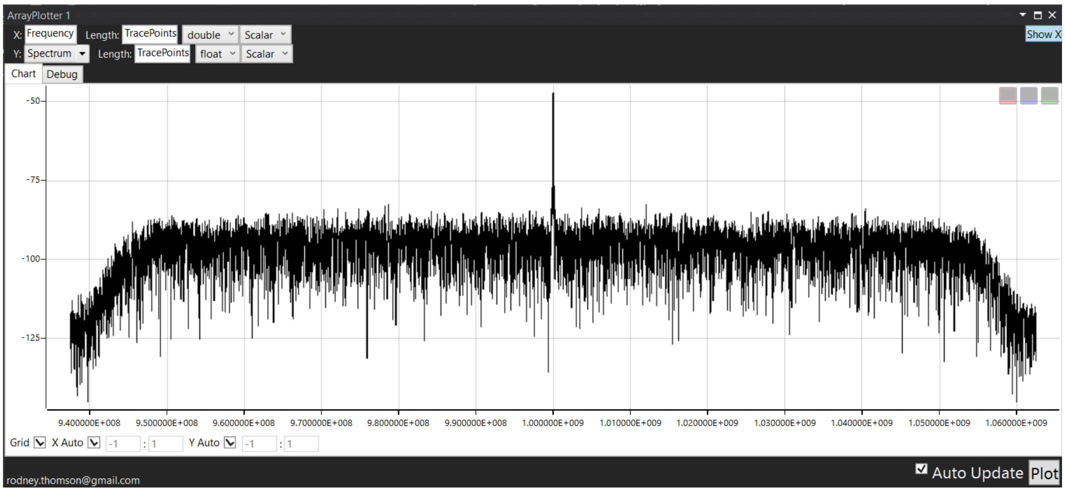


Figure 22 Spectrum trace after data conversion

Root Cause:

1. Anti-Aliasing Filtering Effect: Anti-aliasing filtering is required before acquiring IQ data and before decimation of the IQ data. The transition band of the filter cannot be perfectly steep, leading to a reduction in effective bandwidth;
2. DC Offset Impact: DC offset may be present in the IQ data, which affects the accuracy of the spectrum amplitude;

3. Processing Parameter Difference: The IQ data conversion process is influenced by parameters such as the decimation factor, IQ bit depth, and FFT point count, which introduces differences from the internal processing flow of SWP mode;
4. RBW Inconsistency: The resolution bandwidth (RBW) in the two modes may be inconsistent.

Resolution Steps:

1. Amplitude Anomaly Correction (DC Suppression):
   In IQS mode, set the DC Canceler Mode (DCCancelerMode) to 'Enable High-Pass Filter and Automatic Offset' (DCCAutoOffsetMode):
   ```
   IQS_ProfileIn.DCCancelerMode = DCCAutoOffsetMode;
   ```
2. Eliminating Sidebands (Spectrum Capture):
   Due to the transition band effect of the anti-aliasing filter, the effective bandwidth is reduced to approximately 80%. It is recommended to set Output Spectrum Intercept to 0.8 to effectively eliminate the sidebands. For details, please refer to the DSP_IQSToSpectrum.cpp example file included on the accompanying USB drive.
   ```
   DSP_FFT_TypeDef IQToSpectrumIn;
   IQToSpectrumIn.Intercept = 0.8;
   ```
3. Optimize processing parameters:
   If a significant difference in the spectrum trace still exists after completing the above steps, you can refer to the DSP_IQSToSpectrum.cpp example file on the accompanying USB drive to modify the decimation factor, IQ bit depth, and FFT point count.
4. Keep the RBW consistent:
   RBW = (Sample Rate * Window Factor) / (Decimation Factor * Sample Points), ensure that the RBW of the IQ-converted spectrum is consistent with the RBW of the SWP mode.

2.15 IQ mode signal power deviation

Symptom:

When using IQ mode in software, the signal power value observed on the spectrum plot has a significant deviation from the actual input power. For example, when inputting a signal with a power of -20 dBm to the instrument, the signal power displayed on the spectrum plot is approximately -60 dBm, a deviation of about 40 dBm.

Figure 23 Observed signal power deviation phenomenon in IQ mode

Root Cause:

This phenomenon is caused by DC offset. The DC component present in the IQ receiving link interferes with signal processing, which affects the accuracy of the spectrum amplitude.

Resolution Steps:

1. Enable the High-Pass Filter Function

Click System -> Setting Mode -> Professional, then select High-Pass Filter on the right side.

Figure 24 Enable High-Pass Filter Function in IQ Mode

www.harogic.com
info@harogic.com
+65-8299 8857