

WSA IQ Capture and Decode Guide

Sage Instruments - Advanced Application Note - 2026-05-02

Introduction

The WSA-408 from *Sage Instruments* is a self-contained product that has an embedded PC and DSP built-in, and all the software components are contained within. Users generally do not need to know anything about the internal software architecture. Even for the software upgrade process, all the necessary components are pre-packaged within a single “*WSA_update.bin*” file, and all users need to do is to save this single file into the root folder of a USB drive and then perform upgrade by pressing two buttons on an actively running WSA-408 with the USB drive plugged-in (that contains the “*WSA_update.bin*” file, of course).

However, if a user needs to control a WSA-408 or a WSA-308 from his/her own PC for applications in a lab environment where all screen capture results or IQ data capture results need to go to the PC directly, then some understanding of the WSA software architecture is very useful in order to fully utilize all the potential power of the WSA platform.

To control a WSA-408/308 from a user’s PC, there are two options:

1. Install the WSA_App software package on a PC, and run the WSA-408 like a WSA-308. For the WSA-308, you MUST install the WSA_App software on a PC. For the WSA-408, the built-in GUI software needs to be closed first with the touch of a few buttons, then the PC’s WSA_App can take over. The rest process is identical to that of a WSA-308.
2. Install the WSA_API software package on a PC, and control the WSA-408 (or 308) in parallel while the WSA-408 (or 308) is still running the normal WSA_App. The WSA_API is like a “parasitic” parallel control thread that lets users control the WSA_App remotely via the ethernet network without having to touch any of the buttons on the normal WSA_App GUI.

Prerequisites on a PC

All WSA software packages are very efficient in terms of CPU requirements and memory footprint, and are agnostic of Windows OS versions (from XP to Win-11). However, WSA does need a computer with a decent GigE ethernet port. Although one can get by with a USB-based ethernet adapter, the performance will be the best if one can use a PC with built-in GigE ethernet port. With ethernet connection, one must deal with a few basic IP networking setups in order to achieve the best performance:

1. For best initial networking connections, one should disable the Wi-Fi port as shown in Figure 1. This also prevents unnecessary PC Wi-Fi signal from being picked up by WSA’s antenna.

2. The ethernet port's IPv4 address should be set to "static", and the best address to use is "192.168.24.252", as shown in Figure 1. This is to avoid the two IP addresses used inside a WSA-408, "192.168.24.5" for the DSP, and "192.168.24.254 or 127" for the embedded PC. The WSA-308's internal DSP also uses the address "192.168.24.5".
3. Before launching the WSA_App or WSA_API applications, it's always a good idea to verify the network connections first. Connect the PC with the WSA-408 or 308 through an ethernet cable (an ethernet switch in the middle is OK), and then run the command "ping 192.168.24.5" to make sure the DSP inside the WSA-408/308 is reachable. If not, check your network setups again.

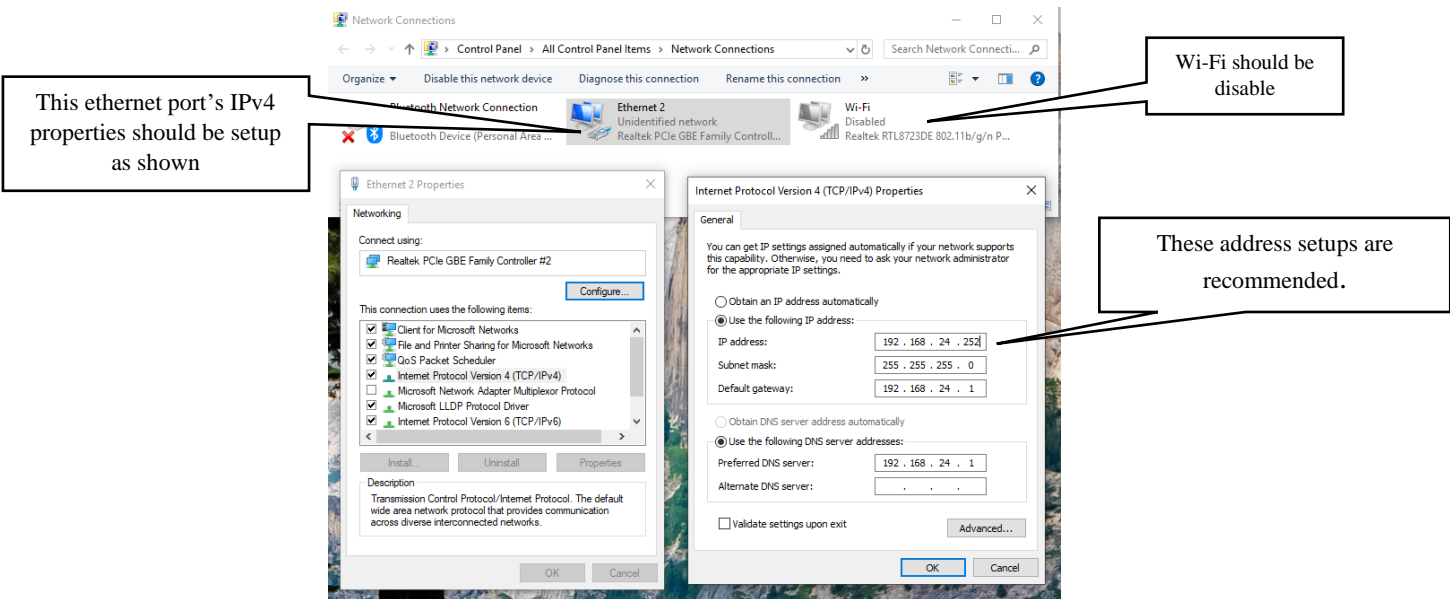


Figure 1 - Example network setup on a PC for best inter-working with a WSA-408/308

WSA_App software components

The "WSA_App" package does not need any elaborate installation procedures. Neither will it insert any weird files everywhere in your PC. All you need to do is to unzip (extract) the "WSA_App.zip" file from Sage Instruments. For simplicity, the "WSA_App.zip" file should be extracted onto a PC's Desktop and maintain the sub-folder structure as shown in Figure 2.

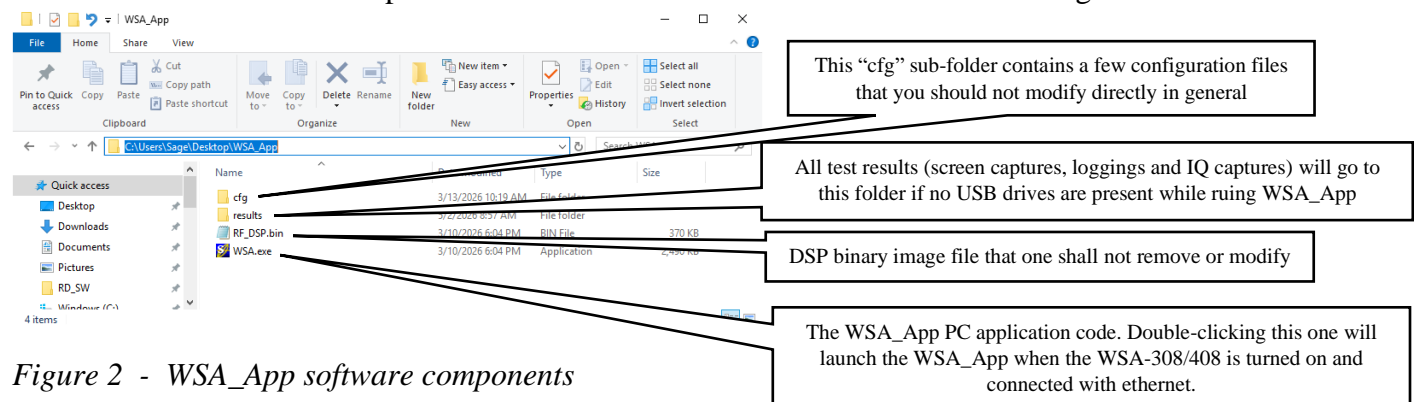


Figure 2 - WSA_App software components

To run a WSA-308, just connect it with the PC via ethernet cable, power it ON, verify the network connections by “pinging”, and then launch the WSA_App by double-clicking “WSA.exe” as shown in Figure 2.

Reducing WSA-408 to WSA-308

To control a WSA-408 from a PC that has “installed” the WSA_App as shown in Figure 2, you first need to close the GUI on the WSA-408 first. Press the lower right “MODE” button, you’ll see dialog as shown in Figure 3. Select “Close the GUI”. You’ll see the warning dialog shown at the right in Figure 3. Just dismiss it by pressing “OK”. After which, the WSA-408’s screen will totally go dark, but the power is still ON and the internal fan should still generate audible noise.

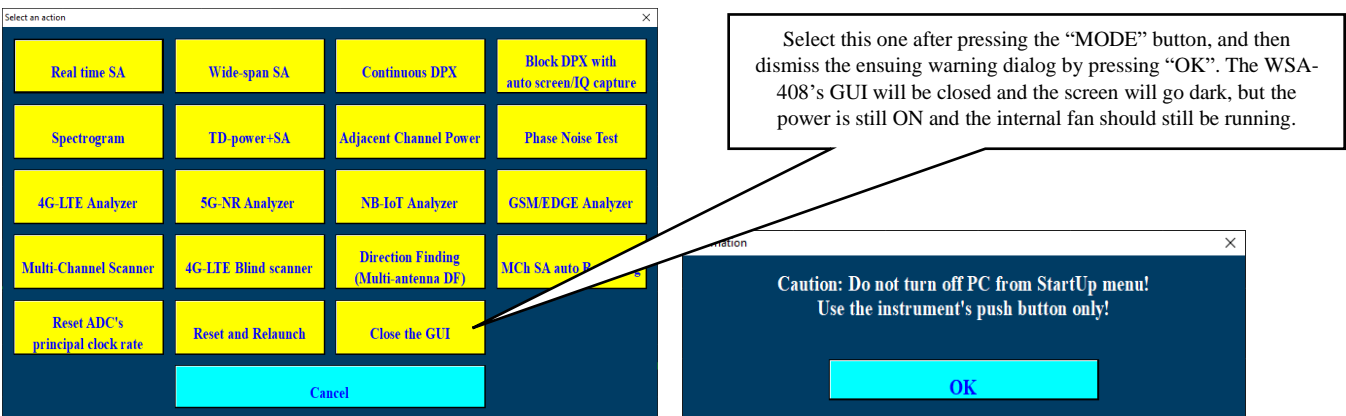


Figure 3 - Steps to close the GUI on a WSA-408 in order for an external PC to take control of it

After the steps in Figure 3, and with proper ethernet cable connection between WSA-408 and the PC and proper network connection verification via “ping 192.168.24.5”, you can launch the “WSA.exe” as shown in Figure 2, and the PC is now in control of the WSA-408’s internal DSP, and the WSA-408 is now functioning like a WSA-308, with one minor difference as shown in Figure 4.

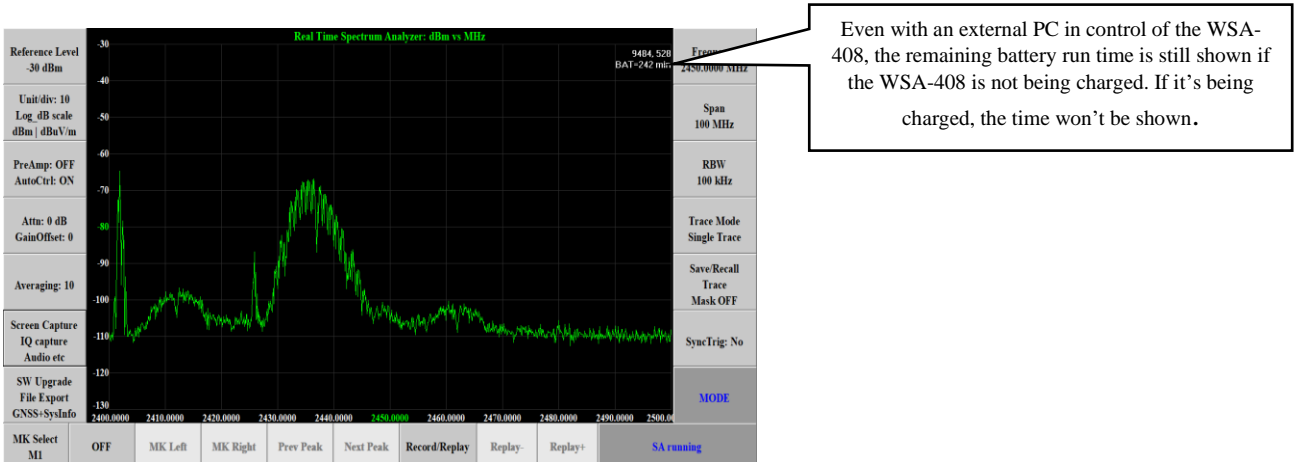


Figure 4 - When controlling a WSA-408 through the WSA_App on a PC, the GUI still displays the battery time as long as the WSA-408 is not being charged.

Resuming the WSA-408's normal operations

To return the “ghosted” WSA-408 to its normal operations through its own touch screen GUI, you must turn it off first. To turn the WSA-408 off, you must close the PC GUI first, disconnect the ethernet cable, and then press the power button of the WSA-408 for about 3 seconds. The dark screen of the WSA-408 should splash momentarily and the turning-off process is initiated. The built-in Windows OS will be shut off and in about 10 seconds, the power will go off and the fan noise will stop. You can then press the power button for 3 seconds again to turn it on, and it will resume the normal operations.

Storage drive precedence, where do the files go?

Whenever you perform a screen capture, or results logging under the 4G-LTE or 5G-NR tests, or IQ data capturing, the data will be saved into files with matching file extensions of “.png” for screen capture, “.csv” for results logging and “.iqr” for IQ capturing. The files will be saved in a storage drive according to the following rules:

1. If a movable drive such as a USB thumb-drive or external USB hard drive is connected to the PC running the WSA_App or plugged into the USB port of a WSA-408, all the above files will be saved to this movable drive.
2. If more than 1 movable drives are found, you'll be prompted to select one to use.
3. If no movable drives are found, the files will go to the “results\” sub-folder as shown in Figure 2. You can directly navigate to that folder to retrieve all the test results if you're running the “WSA_app” on a PC.

Moving the results files out of a WSA-408

When using a WSA-408 in the field, and no USB drive was plugged in, all the above test results and IQ captures would have been saved inside the WSA-408's internal solid-state hard drive that a user does not have easy access to. For this reason, you must use the following approach to move the files out.

1. Plug-in a USB drive into the WSA-408's USB port and turn on the unit if it's not already ON.
2. Press the lower left button and then “Export/File management” and you'll see dialogs as shown in Figure 5. Select the proper file types and follow the dialog instructions.
3. While exercising the above steps, the external USB drive must be inserted and detected, otherwise, a warning dialog about missing USB drive will be shown and no further actions will be taken until the USB drive is detected.

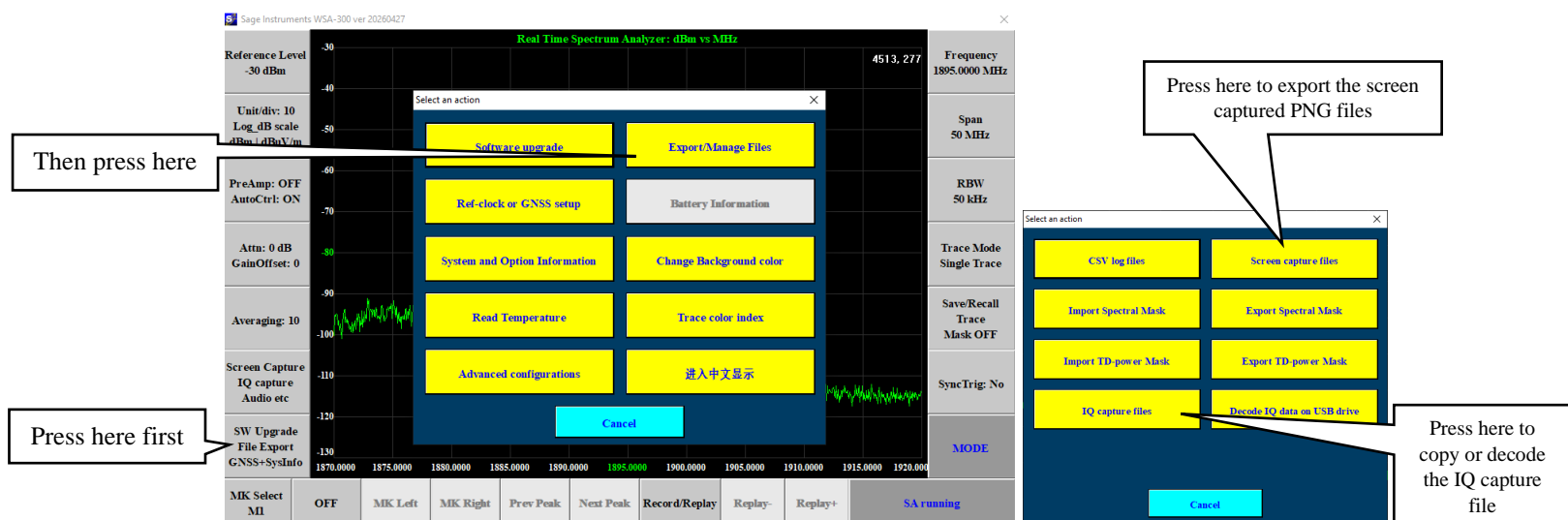


Figure 5 - Steps to take to export the various test results files from a WSA-408 to a USB drive

How to perform IQ data capture

The high-quality IQ data inside a WSA-408/308 can also be captured and saved into a file (or a sequence of files) for users. The WSA platform offers three categories of IQ captures:

1. A single-instance of IQ capture of finite IQ data length whose length is determined by the application setup (such as RBW) or explicit time length set by users.
2. Automated periodic IQ capturing (coupled with simultaneous screen capturing) of a single-instance of IQ capture of finite duration. This is particularly useful for RF-AI model trainings.
3. Continuous IQ data capturing with max sampling rate of 20 MHz by saving a sequence of files into a user-selected external hard drive (preferably). The storage hardware's max capacity will be detected, and 90% of it will be used circularly. Assuming the hard drive can hold 5 days of IQ data recording, and a WSA-408 is left running for a month, then only the last 5 days of IQ data will be kept.

This document will only cover the first two operations. The last one will be covered in a dedicated separate document.

IQ capture of a single-instance "at will"

Except for the "Wide-span sweeping mode SA", "Continuous DPX" and "Spectrogram" tests, all other test functions will allow users to capture a single-instance of IQ data stream used by that particular test function. The exact IQ data length and sampling rate used is available to users

with the touch of a button. Figure 6 shows an example of IQ capture instance under the most commonly used “Real-time SA” feature.

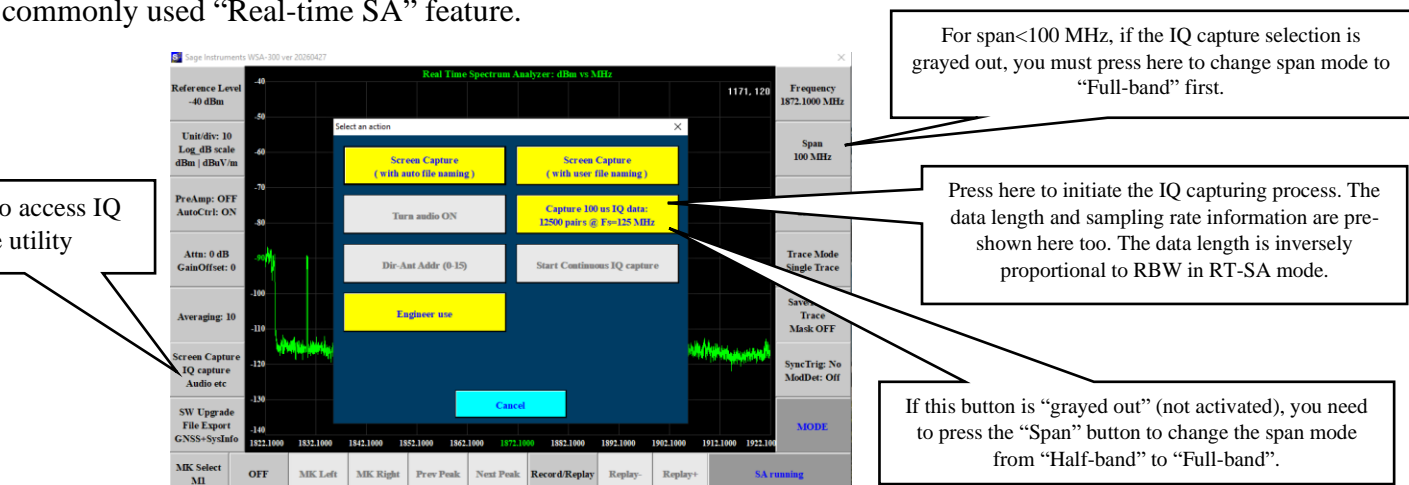


Figure 6 - Steps to take to initiate a single-instance of IQ capture at will

For span<100 MHz cases, if the IQ capturing selection is grayed out (disabled”) in Figure 6, you then need to press the “Span” button and change the span mode from “Half-band” to “Full-band” as shown in Figure 7.

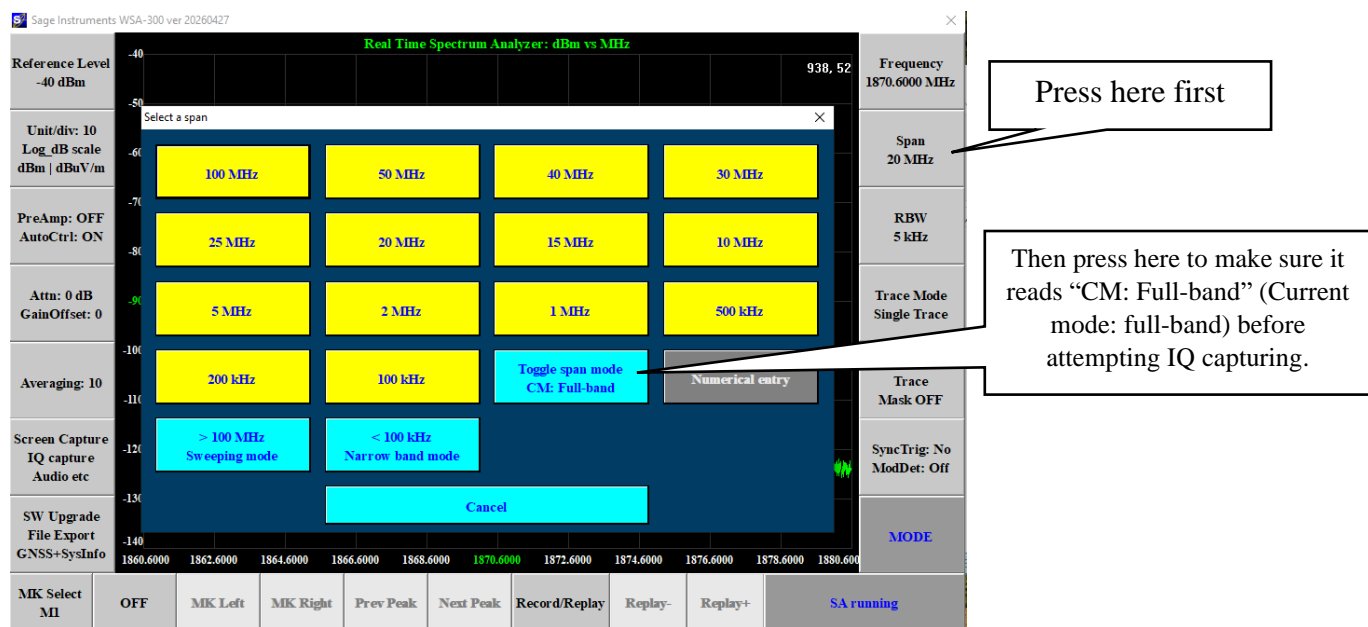


Figure 7 - For span <100 MHz, these are the steps to make sure it's in “Full-band” mode before capturing IQ.

Single-instance IQ capturing of user-definable duration

The best way to perform IQ capturing is under the Block DPX test function as shown below in Figure 8.

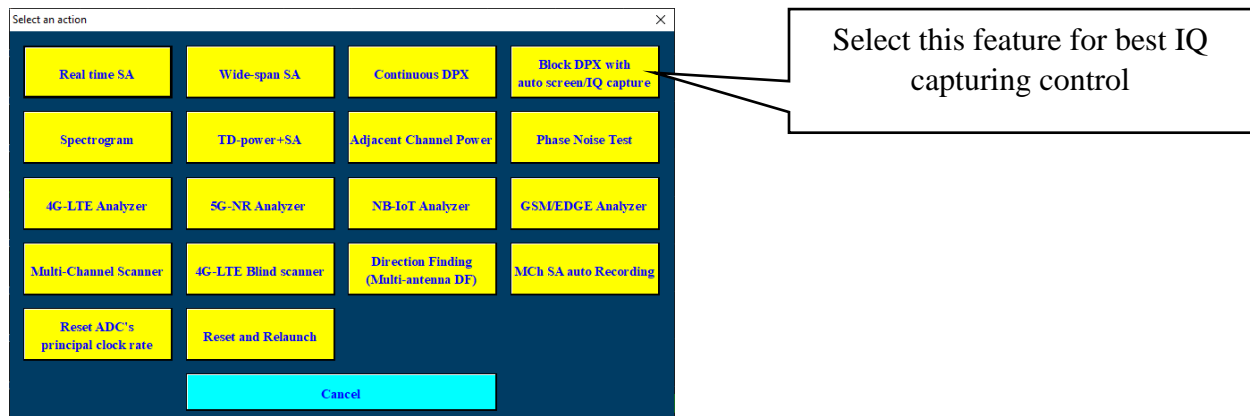


Figure 8 - Test mode selection menu. Select “Block DPX” for best IQ capturing control.

Once inside the Block DPX feature, first, you also need to make sure the span mode is set to “Full-band” mode when span is under 100 MHz, as shown in Figure 9.

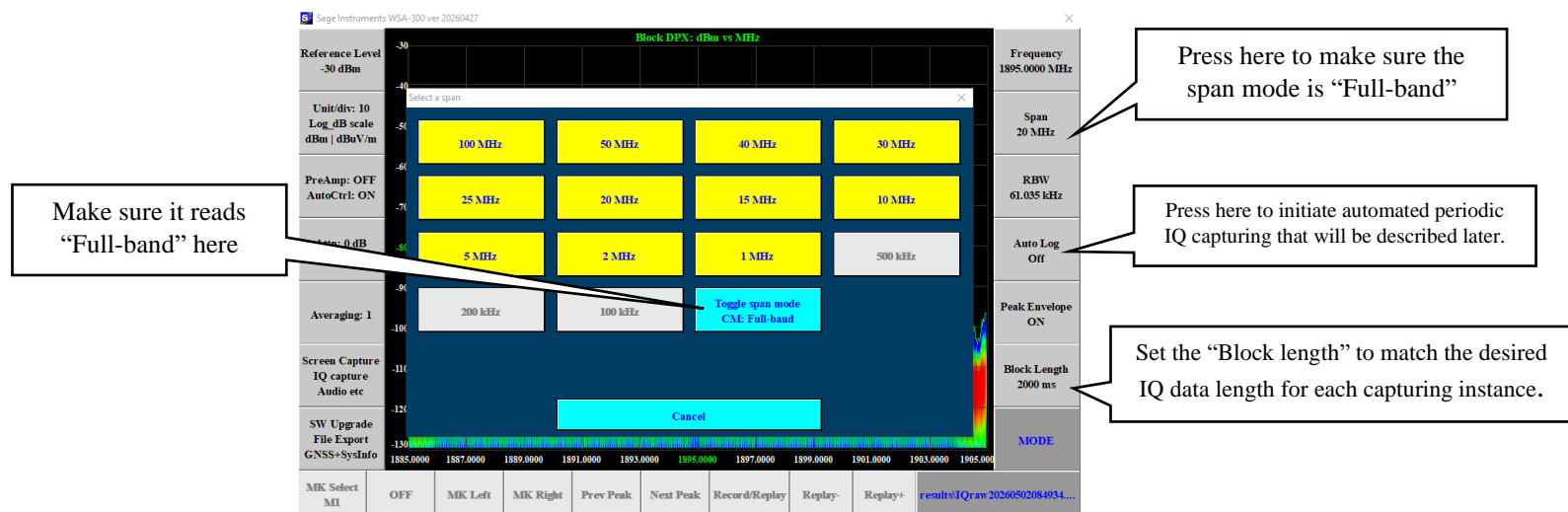


Figure 9 - Setting up the Block DPX properly for IQ capturing

As shown in Figure 9, once entering the Block DPX mode, you should check the following before attempting IQ capturing:

1. Make sure the span mode is in “Full-band” mode when span is less than 100 MHz. If not in “Full-band” mode, the “Auto-log” can still be turned on, but no actual IQ capturing will be performed. You MUST make sure it’s in “Full-band” mode first each time you change the span setting.

2. Select proper “Block length”. The smaller the span setting, the longer the “Block length” can be set to. For the example shown in Figure 9, with span=20 MHz, the max block length can be set to 2000 ms, meaning that users can capture a single instance of contiguous IQ data stream of 2s long. Please be aware that capturing long IQ data may take a while. A 2s IQ data may take up to 10 seconds to gather, then transport through the ethernet and then be saved into a file. One must be patient! Selecting shorter block length is always more efficient if you do not need long IQ data length to begin with.
3. After proper setups, you initiate the actual IQ capturing process using the same steps as shown in Figure 6.

Starting automated periodic IQ capturing and screen capturing

As shown in Figure 9, under the Block DPX test function, the right 4 button “Auto Log” allows user to initiate automated periodic screen capturing and IQ data capturing. The intention is to use these two sets of data to train certain RF-AI models for signal identification and recognition. The two data sets (DPX screen capture and raw IQ data) are time-synchronized and refer to the same data sets, so the AI model can either be trained by the screen capture PNG file (which is what we recommend for its simplicity, efficiency and suitability for AI models), or be trained with the raw IQ data if necessary. After pressing “Auto Log” in Figure 9, you’ll see dialog as shown below in Figure 10.

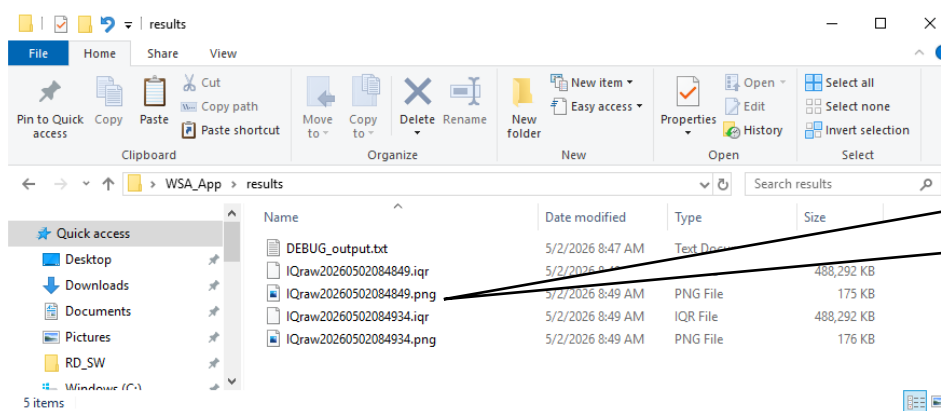


Figure 10 - Buttons to select and start the automated periodic capturing

Once proper selection is made in Figure 10, the software will do this:

1. Immediately perform a screen capturing (DPX display goes to a PNG file) and a companion IQ capturing. Both files refer to the same IQ data set and the file names are identical (with the same timestamp) but differing file extensions.
2. After 20 seconds (or 40 or 60 seconds), the actions in 1 will be repeated again and again.

Example files as a result of setup in Figure 10 are shown in Figure 11.



Two capture instances are shown here. The matching .iqr and .png files have identical name (same timestamp) but differing extensions.

Figure 11 - Example file names as a result of automated periodic capturing

What to do with the captured IQ data?

The captured IQ data in “.iqr” file format contains proprietary hardware information unique to the WSA platform. Since it’s also the most efficient data format, we strongly urge users to archive their IQ data using this format. A single “.iqr” file not only contains the raw IQ data but also crucial information on sampling rate, center frequency and internal filter and gain settings etc. But to use it, you must decode each “.iqr” file into a pair of files:

1. A plain text header file with extension “.txt” that provides sampling rate, center frequency, usable bandwidth and GPS timing information if available.
2. A companion “.dat” file that contains a single array of IEEE-754 32-bit floating point data with I and Q interleaved. The IQ data stream is also normalized in such a way that $|z(n)|^2 = |I(n) + i * Q(n)|^2$ being 1.0 means the measured instantaneous power is 1 mW (0 dBm); being 0.01 means 0.01 mW (-20 dBm).

How to decode the “.iqr” data files?

You have two ways to decode the “.iqr” files into format that you can further process in your own ways:

1. Use the built-in utility as shown in Figure 5. This is suitable for the light-weight users who perform IQ capturing occasionally using a WSA-408 in the field. And while capturing IQ data, a USB drive was not available, so the .iqr files were saved inside the WSA-408. A user has to get these files out to a USB drive later by steps listed in Figure 5. On the file type selection at the right of Figure 5, the user should select the lower left one “IQ capture file” and then select all the .iqr files and then select “Decode then delete the original” as shown below in Figure 12. For large IQ files or lots of IQ files, this may take a while. So be patient!
2. Use a PC application software provided by *Sage Instruments* for free.

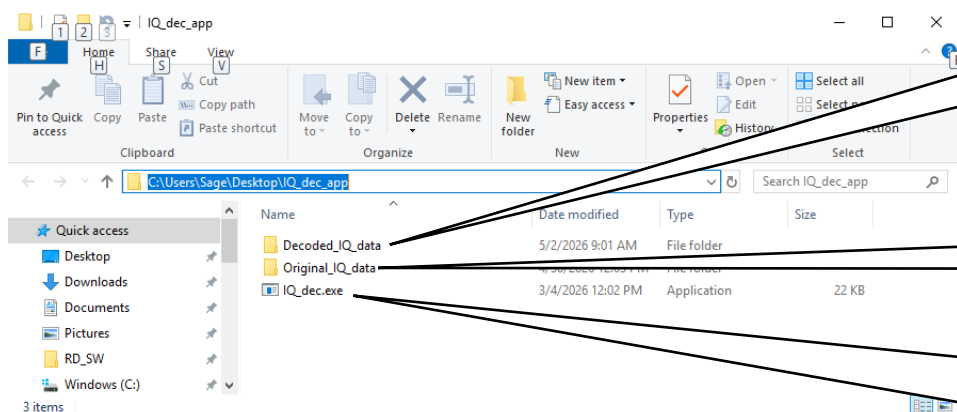


Select this for IQ decoding. After which, all the internal .iqr files will be decoded into companion pairs of files and saved to the USB drive while original .iqr files will be deleted to conserve storage space.

Figure 12 - Using the built-in utility to decode the IQ data files

Use the IQ_Dec app to decode the .iqr files

Sage Instruments provides for free a PC application program to decode the “.iqr” data files on any Windows PC. The application is in the “IQ_dec_app.zip” file. You should unzip (extract) this file onto the desktop on a PC and maintain the following file structure as shown below in Figure 13.



The decoded “.txt” header files and “.dat” IQ data files will go into this folder

The original “.iqr” files to be decoded MUST reside in this folder

Execute this IQ decoder application by either double clicking here, or open a DOS-cmd prompt and navigate to this folder and type “IQ_dec” and return.

Figure 13 - The IQ decoder app file and folder structure

Using data collected in Figure 11 as examples, we move the “.iqr” files into the “Original_IQ_data” folder shown in Figure 13, the resulting content inside “Original_IQ_data” folder should look like that shown in Figure 14.

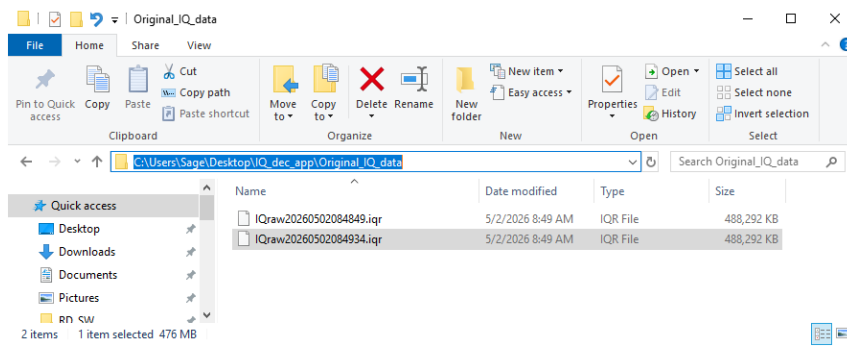


Figure 14 - The captured “.iqr” files (from Figure 11) are moved to the “Original_IQ_data” folder inside the “IQ_dec_app”, ready to be decoded.

After steps in Figure 14, back to the “IQ_dec_app” folder and execute “IQ_dec.exe”. If there is only one “.iqr” file, the file will be decoded automatically without any prompt. If there are more than one “.iqr” files as in the case of Figure 14, you’ll see a prompt shown in Figure 15.

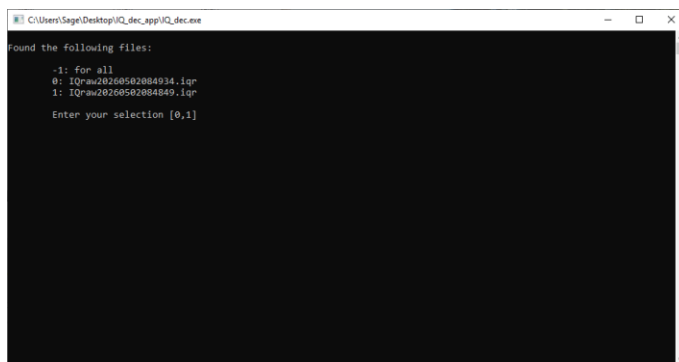


Figure 15 - File selection when executing “IQ_dec.exe” app with more than one “.iqr” files detected

In Figure 15, enter (on your keyboard) 0, 1 or N-1 (N being the number of files listed) to decode a specific file, or enter “-1” to decode all. Once again, for large files or lots of files, this may take a while, so patience is gold. The decoding process example is shown in Figure 16.

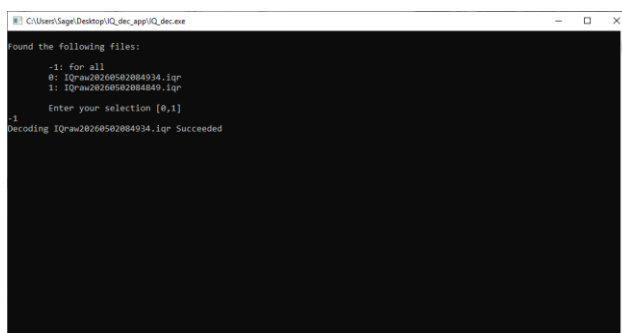


Figure 16 - An exemplary decoding process. This console display will disappear once all files are decoded, unless you started the “IQ_dec.exe” application inside a cmd prompt.

Before decoding, the “Decoded_IQ_data” folder in Figure 13 should only contain two example MATLAB script files as shown in Figure 17.

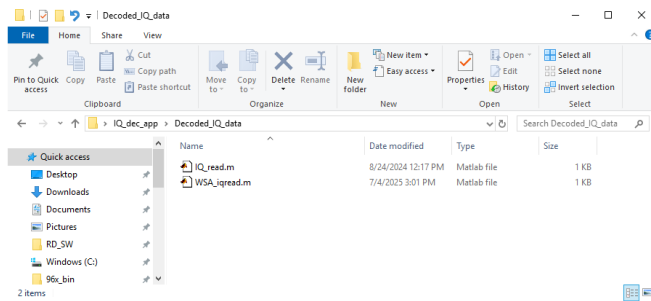


Figure 17 - File contents inside the “Decoded_IQ_data” folder before any decoding

After the decoding actions in Figure 15 and 16, the file contents inside the “Decoded_IQ_data” folder will look like that shown in Figure 18.

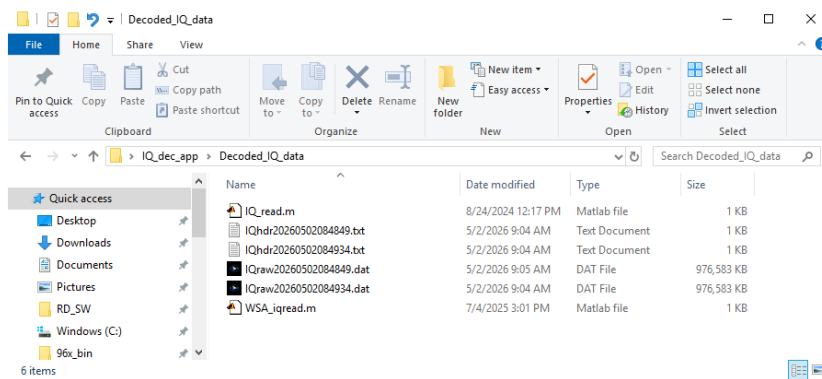


Figure 18 - After decoding, the “Decoded_IQ_data” folder will contain two sets of files matching the original two “.iqr” files. There are two “.txt” header files and two “.dat” files. Both file’s timestamps remain the same as the original “.iqr” data files (time stamp is year-month-date-hour-minute-second).

The “.txt” header file content will look like that shown in Figure 19.

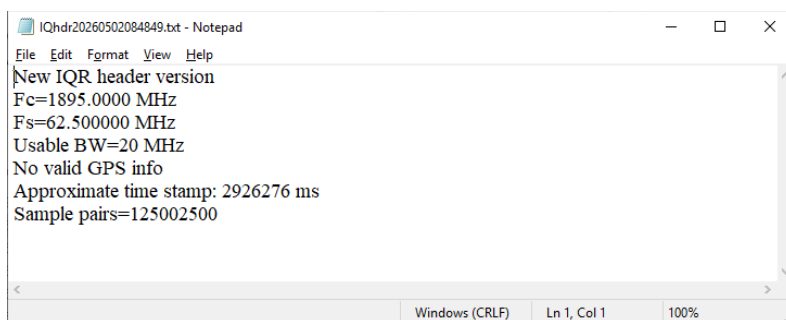
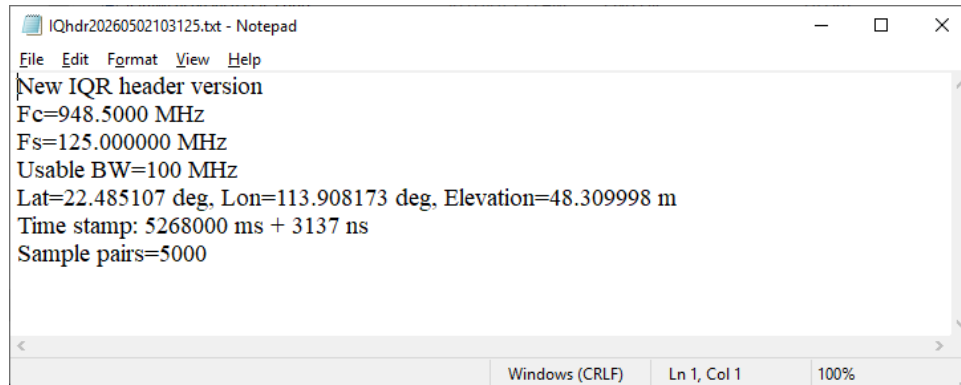


Figure 19 - Example header file showing the center frequency (F_c), sampling rate (F_s), usable bandwidth (20 MHz) and timestamp (UTC time of the day in ms) and IQ sample length in pairs.

The header displayed in Figure 19 contains no GPS information. If the WSA-408/308 is locked to the GPS (how to set up the GPS lock will be in another document) when performing IQ capturing, then the GPS information will also be displayed in the “.txt” header file as shown in Figure 20.



```

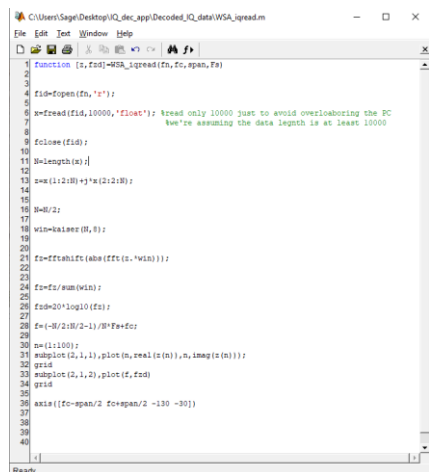
IQhdr20260502103125.txt - Notepad
File Edit Format View Help
New IQR header version
Fc=948.5000 MHz
Fs=125.000000 MHz
Usable BW=100 MHz
Lat=22.485107 deg, Lon=113.908173 deg, Elevation=48.309998 m
Time stamp: 5268000 ms + 3137 ns
Sample pairs=5000
Windows (CRLF) Ln 1, Col 1 100%

```

Figure 20 - In this exemplary .txt header file, GPS information is available in terms of latitude (Lat=..., will be negative numbers for Southern hemisphere), longitude (Lon=..., will be negative numbers for Western hemisphere) and elevation. The time stamp shows a precise “3137 ns” offset. This is relative to GPS’s 1PPS pulse (1 pulse per second). With this information, if one captured multiple data sets using different WSA units located at different places, but all locked to GPS and performed capturing at roughly the same time, you can post-synchronize the data from different units to perform a TDoA (Time Difference of Arrival) type of analysis.

Dealing with the decoded .dat files

The two MATLAB script files shown in Figure 17 provide examples on how to read and process the IQ data stream stored inside the .dat files. More specifically, the “WSA_iqread.m” file looks like that shown below in Figure 21.



```

C:\Users\Sage\Desktop\IQ_dec_app\Decoded_IQ_data\WSA_iqread.m
File Edit View Window Help
function [z, fcd]=WSA_iqread(fc, span, Fs)
1
2
3
4 fid=fopen(fc, 'r');
5
6 x=fread(fid, 10000, 'float'); %read only 10000 just to avoid overloading the PC
7 %we're assuming the data length is at least 10000
8
9 fclose(fid);
10
11 N=length(x);
12
13 x=(1:2:N)+j*(2:2:N);
14
15
16 N=N/2;
17
18 vis= Kaiser(B, 1);
19
20
21 f0=fftshift(fft(abs(fft(x.*vis)))));
22
23
24 f0=f0/sum(vis);
25
26 fcd=20*log10(f0);
27
28 f0=(-N/2:N/2-1)/(N*Fs+Fc);
29
30 N=1:100;
31 subplot(2,1,1), plot(N, real(z(N)), N, imag(z(N)));
32 grid
33 subplot(2,1,2), plot(f, fcd)
34 grid
35
36 axis([fc-span/2 fc+span/2 -130 -30])
37
38
39
40
Ready

```

Figure 21 - File content of the “WSA_iqread.m” MATLAB script file. This file shows how to best read the .dat file.

For the decoded files shown in Figure 18, we open a MATLAB command window, and navigate to the “Decoded_IQ_data” folder and issue the MATLAB commands as shown below in Figure 22, you’ll see resulting plots shown at the right of Figure 22.

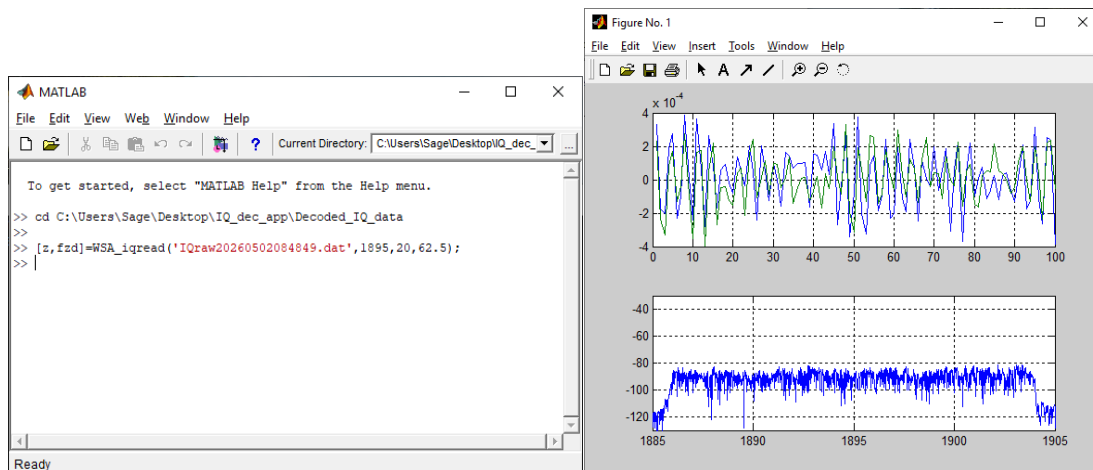


Figure 22 - Issuing the two lines of MATLAB commands shown at the left will yield results shown at the right for the .dat files (shown in Figure 18)

About WSA_API

While a WSA-408 is running the embedded “WSA.exe” application or a WSA-308 is connected with a PC running “WSA.exe” GUI, you can also start the “WSA_API” application to simultaneously control the same WSA’s internal DSP and RF hardware. For a WSA-408, the WSA_API by nature must run on an external PC connected with it through ethernet cable. For a WSA-308, the WSA_API can run either on the same PC, or on another PC. If running on another PC, the other PC must also have network access to the WSA-308. This is achieved by connecting the two PCs and the WSA-308/408 into the same ethernet switch.

The WSA_API package (inside the WSA_APP.zip file) contains the following:

1. A complete set of C source files and ready-to-use project management setup using Microsoft’s Visual Studio tool.
2. A precompiled executable “WSA_API.exe” that you can readily use under a console command prompt (the good old DOS command). Once executed, you can remotely control an actively running WSA-408/308 from another computer without ever having to use the normal “WSA.exe” GUI, and all forward commands will be reflected correctly on the actively running “WSA.exe” GUI.

The purpose of the WSA_API C source code is to provide the following:

1. Provide the crucial underlying network sockets and protocols for communicating with the DSP hidden inside the WSA-408/308 unit. This is such that the users do not have to labor through the low-level networking and multi-packet handling software, and eliminate the

necessity of having to understand the communication and control protocols as much as possible.

2. Provide exemplary simulated program for how to initiate test changes and change RF settings and getting test results etc so that the users can further expand them to whatever applications they have in mind.

Understand the WSA_API source code

You should unzip (extract) the “WSA_API.zip” file onto a PC’s folder and maintain the following file structure.

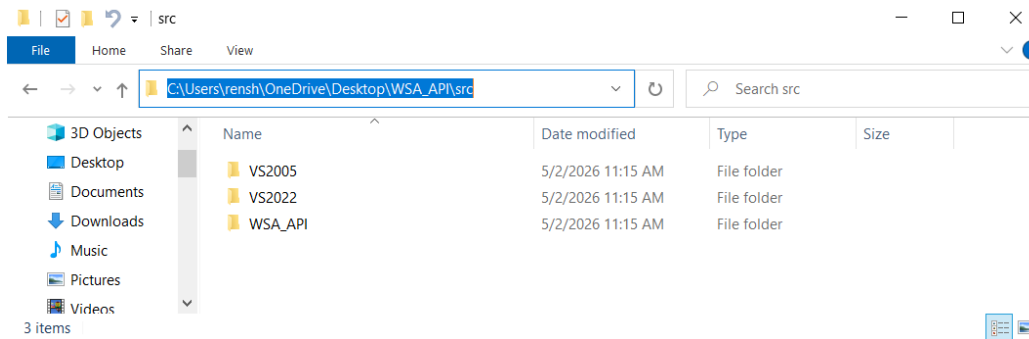


Figure 23 - Inside the “src” folder of the unzipped “WSA_API.zip” files, you’ll see sub-folders as shown here.

As shown in Figure 23, all the C-source files are under the “WSA_API” sub-folder as shown below:

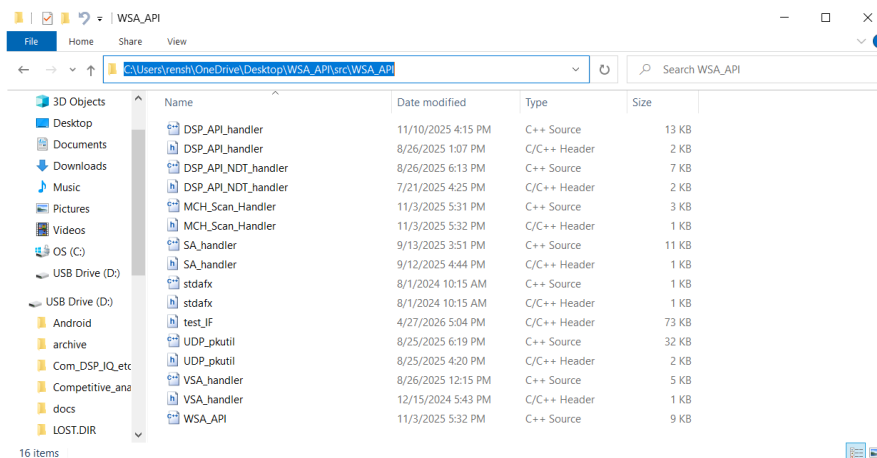


Figure 24 - All the C source files are contained inside the “WSA_API” sub-folder as shown here

Although one can import all the above C source files into any software programming or project management tool environments, we suggest you use Microsoft's (VS) Visual Studio tool. Assume you already have the VS tool installed, you can go back to the "src" folder shown in Figure 23, and navigate to the VS2005 folder if you have VS version prior to 2022, or navigate to VS2022 folder if you have VS version 2022 or later. Assume you have the latter, go to the VS2022 folder, you'll see the following:

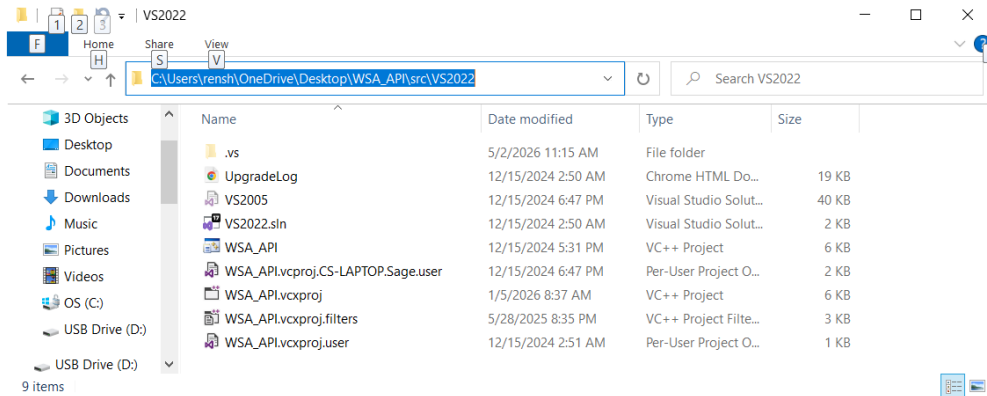


Figure 25 - VS project management files. Double clicking the "VS2022.sln" file should automatically launch the VS tool and present a nice project list as shown in Figure 26.

After clicking "VS2022.sln" in Figure 25, with proper VS tool, you should see the following in Figure 26.

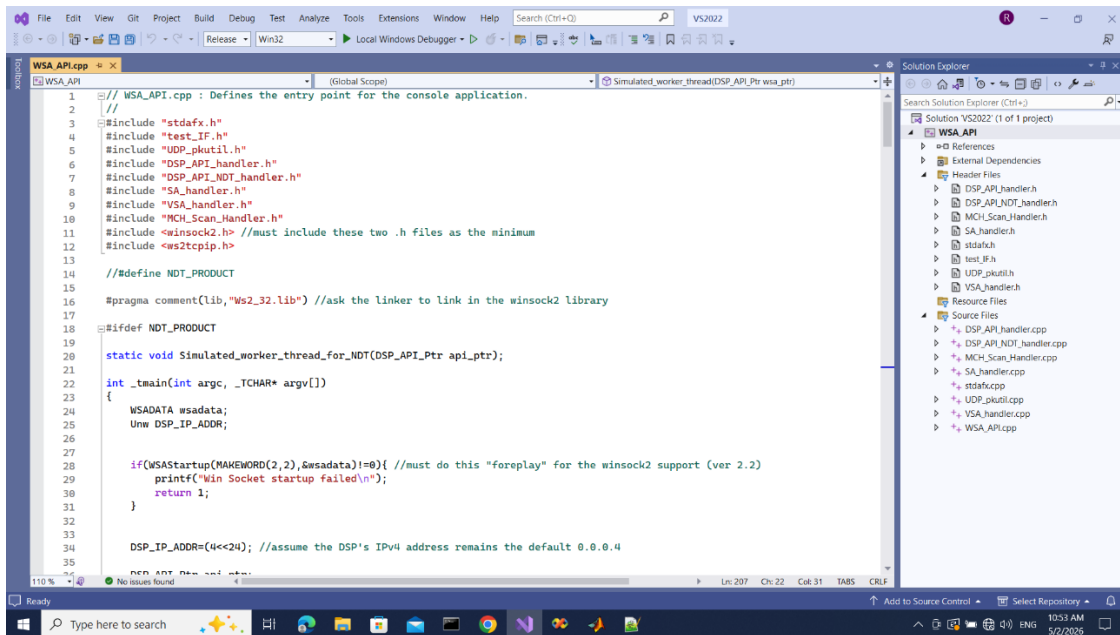


Figure 26 - A simple project list of all the C source files associated with WSA_API application

With the IDE shown in Figure 26, you can also browse the API command list as shown in Figure 27.

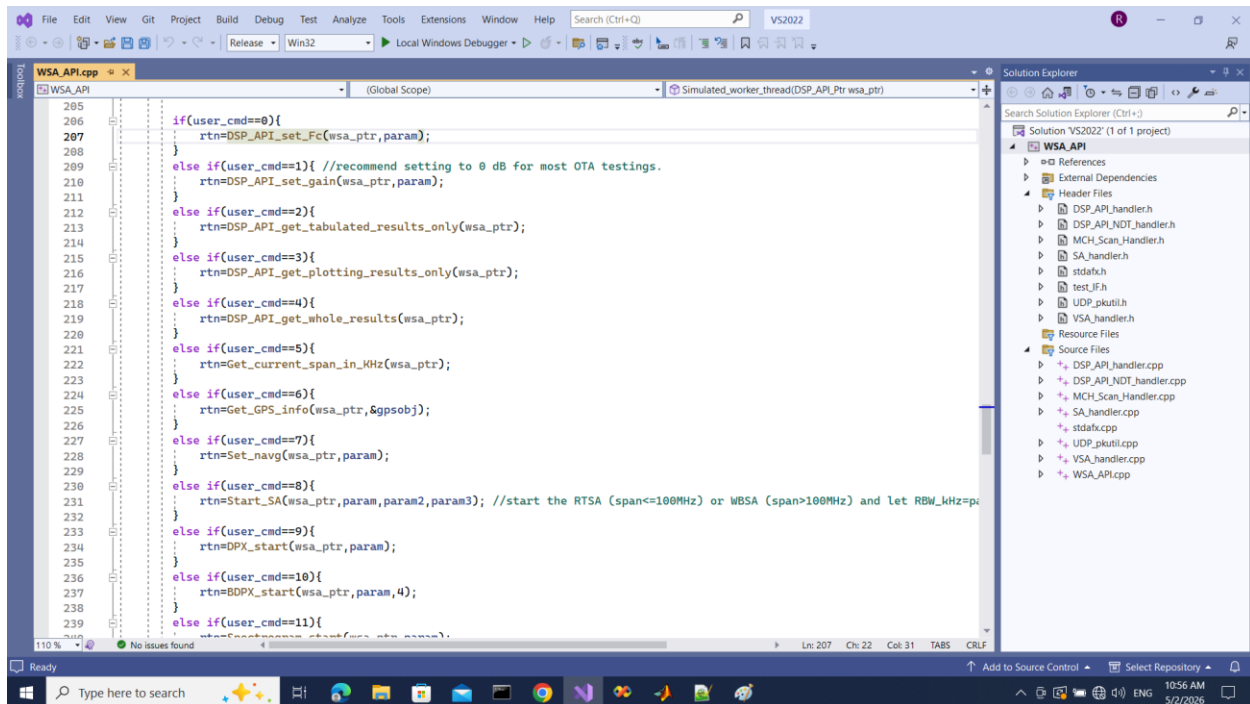


Figure 27 - A snapshot of various function calls for remotely controlling an actively running WSA-408/308. By following the function definition, you can easily figure out the inner working mechanisms.

Using WSA_API without dealing with the C code

If you don't have the patience or time to deal with the C source code, you can play with the pre-compiled "WSA_API.exe" application to familiarize yourself with the WSA_API commands. To do this, follow these steps:

1. Turn on a WSA-408, for simplicity. WSA-308 works the same way although you'll have to have a PC with "WSA_App" installed.
2. Connect the WSA-408 with a PC through ethernet, and set up the PC's ethernet IPv4 properties according to what this document described at the beginning sections. Verify connection by "ping 192.168.24.5".
3. Open a cmd prompt (the good old DOS command prompt), navigate to the "WSA_API\out" folder as shown below in Figure 28 and start the "WSA_API.exe" as shown. You can see that you can remotely change the center frequency and also get an instance of the spectrum analyzer results.

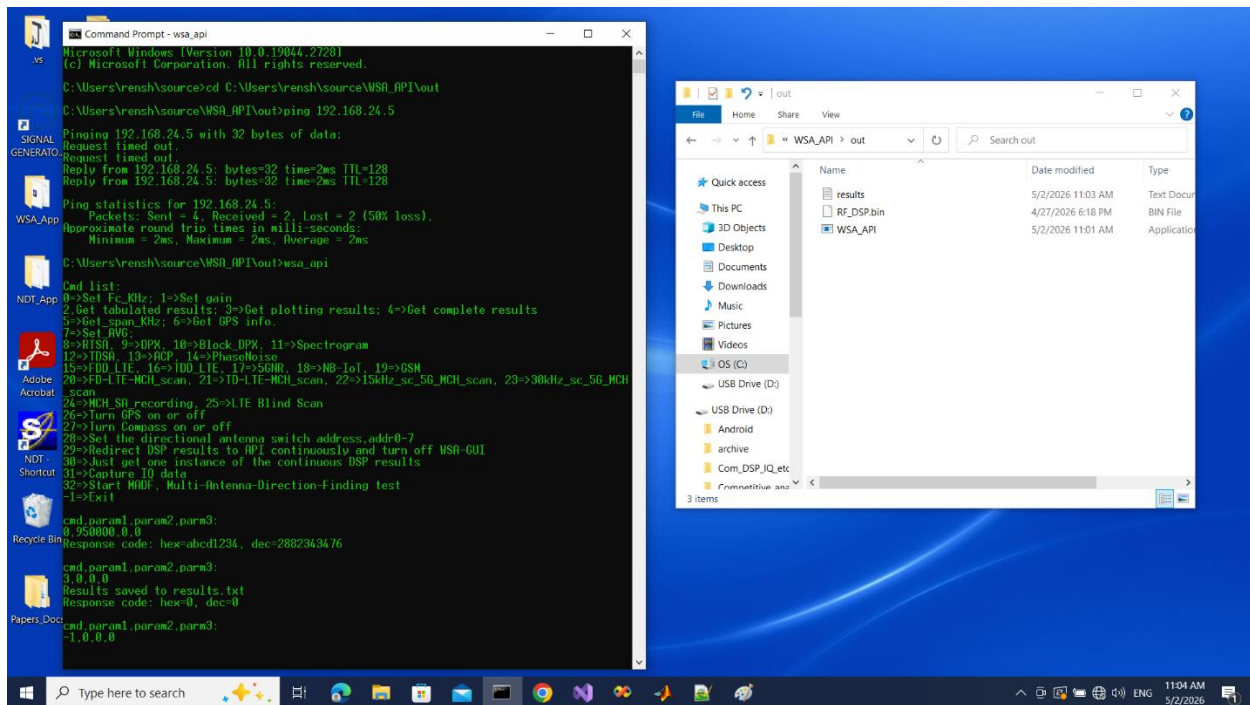


Figure 28 - Example of how to use the pre-compiled “WSA_API.exe” application.

The “results.txt” file shown in Figure 28 at the right and obtained via command “3,0,0,0” at the left contains readable data array (corresponding to spectrum in unit of dBm) as shown below:

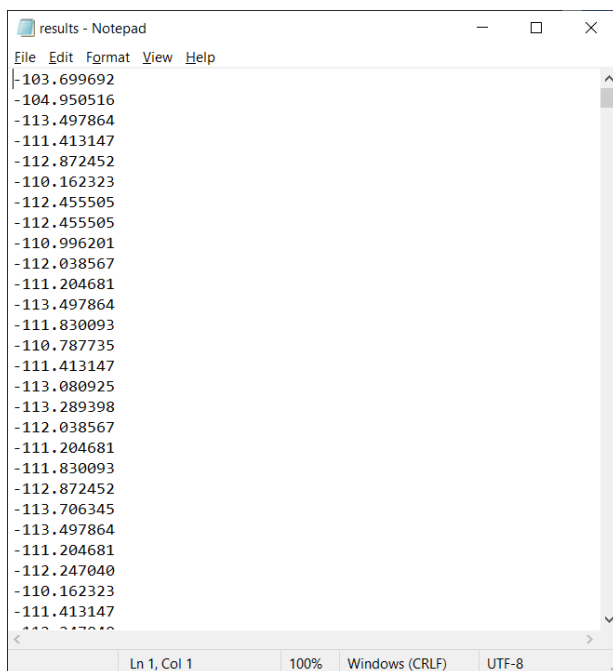


Figure 29 - Example Spectrum Analyzer results obtained through the WSA_API

Limitations of the WSA_API

Although the “WSA_API” has the potential to be expanded to be a powerful substitute of the factory provided “WSA.exe” GUI application, at its current implementation, one must be aware of the following:

1. For the majority of applications, one can use the WSA_API to change the test and hardware settings and get any instances of test results.
2. But for the communication bandwidth intensive application such as IQ capturing, the WSA_API support is not provided. Such support is possible, but requires major software rework at all 3 fronts (the WSA.exe GUI, the WSA_API and the DSP codes). This will only be provided to users who are committed to product purchase of adequate quantities. For IQ capturing, we strongly urge you to use the existing WSA.exe GUI interface, especially the automated periodic capturing method described in this document. For a WSA-408, we suggest you install the “WSA_App” package on a PC and control the WSA-408 as a WSA-308 using steps described in this document so that all IQ files are naturally stored in user’s PC.