# Tektronix RSA Device
# MATLAB Driver
# User Guide
# API Build 3.9.0029

**Version 2.02**

# Table of Contents

# Overview

The RSA_API MATLAB Driver allows communication between MATLAB and RSA API. The driver uses a generic driver from MATLAB's Instrument Control Toolbox to make individual calls to the API. This driver supports version 3.9.0029 of the API.

# Requirements

The following hardware, software and files are needed to use the RSA_API MATLAB driver.

1. RSA device. Supported models: 306, 306B, 503A, 507A, 603A, 607A
2. MATLAB and the Instrument Control Toolbox. (MATLAB R2015b has been tested).
3. Version 3.9.0029 of the RSA API, included in the MLTBX package.
4. MATLAB instrument driver file (RSA_API_Driver.mdd), included in the MLTBX package.

# Installation

MATLAB and the Instrument Control Toolbox, are assumed to be previously installed. Add the RSA MATLAB Driver package to MATLAB my running the "Tektronix USB Spectrum Analyzer Driver.mltbx" installer, or by using the "Get Add-Ons" menu in MATLAB. The API installer, examples, and documentation will be added to your Documents\MATLAB\Add-ons\Toolboxes folder.

An installer for the Tektronix RSA API is included in this package. Please run RSA_API_Install.m to install the API, or run the setup.exe file from the RSAAPI-64_3.9.0029 folder if the API has not yet been installed.

The MATLAB driver will look in the default install location of C:\Tektronix\RSA_API for the API files. If you installed the API in a different location, you will need to change the driver path to point to the new location:

1. Open the RSA_API_Driver.mdd in the MATLAB Instrument Driver Editor. See Figure 1.
2. Change the default value of the apiIncludePath and apiLibPath properties. See Figure 2.

Fig 1. Open Driver Editor

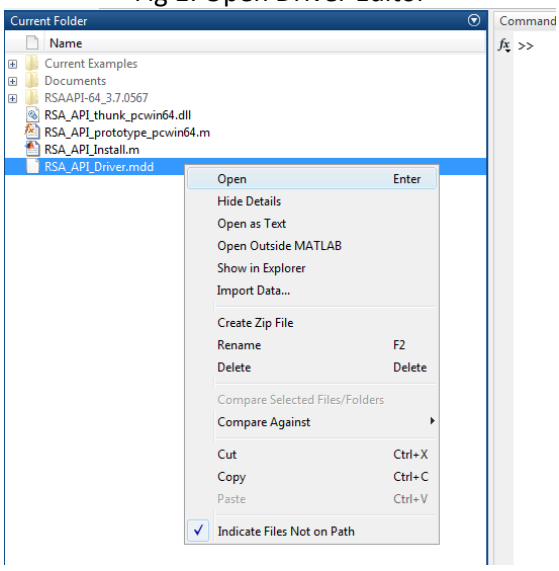
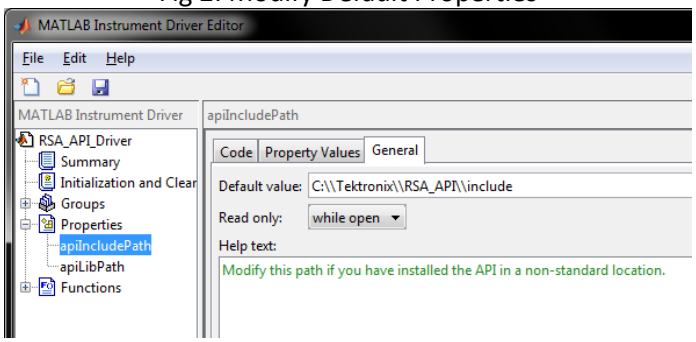



Fig 2. Modify Default Properties



# How to Use the Driver

Using the Driver requires the following basic steps:

1. Create the device object.
2. Connect to the device object.
3. Set/Get properties or invoke functions.
4. Disconnect from the device object.
5. Delete and clear the device object from memory.

Example of the following basis steps:

1. See Installation section.
2. Create the device object using the icdevice(**'driver'**) function.
    a. **'driver'** is the file name for of the MATLAB Instrument Driver.
        i. **File**: Use this optional parameter for file playback

```
device = icdevice('RSA_API_Driver');
device = icdevice('RSA_API_Driver', 'File');
```

3. Connect the object to the driver with the connect(**device**) function.
        i. **device** is the device object created in Step 2.

```
connect(device);
```

4. Set/ Get properties or invoke functions.
    a. Set properties using the set(**obj**, '**propertyName**', **propertyValue**) function.
        i. **obj** is the device group object.

```
set(device.Configure, 'CenterFreq', 2.4453e9);
```

    b. Get properties using the get(**obj**, '**propertyName**') function.

```
centerFrequency = get(device.Configure, 'CenterFreq');
```

    c. Invoke functions using the invoke(**obj**, '**functionName**', **input1**, **input2**, …) function.

```
invoke(device.Device, 'Run');
```

5. Disconnect from the device object using the disconnect(**device**) function.

```
disconnect(device);
```

6. Delete and clear the device object from memory.

```
delete(device);
clear('device');
```

## Example: Generate IQ Data

The following script provides an example that generates IQ data with the MATLAB driver.

```
%Create the device object
    dev = icdevice('RSA_API_Driver');

%Connect to the device object
    connect(dev);

%Set the center frequency to 2.4453GHz
    set(dev.Configure, 'CenterFreq', 2.4453e9);

%Set the reference level to -60dB
    set(dev.Configure, 'ReferenceLevel', -60);

%Set the IQ record length to 5000 samples
    set(dev.Iqblock, 'IQRecordLength', 5000);

%Begin data acquisition
    invoke(dev.Device, 'Run');

%Wait for the IQ data to be ready. It has a timeout value of 5000ms
    invoke(dev.Iqblock, 'WaitForIQDataReady', 5000);

%Get IQ data. The start index is set to 0 and the record length is 5000 samples.
    [I, Q] = invoke(dev.Iqblock, 'GetIQDataDeinterleaved', 0, 5000);

%Multiply the Q data by sqrt(-1) to make it imaginary
    Q = Q*1i;

%Add the I and Q arrays together to get IQ data
    IQ = I+Q;

%Stop data acquisition
    invoke(dev.Device, 'Stop');

%Disconnect from the device object
    disconnect(dev);

%Delete and clear the device object from memory
    delete(dev);
    clear('dev');
```

# Appendix A: Driver Properties

**Table 1. Property Value Notation**

| Notation | Description |
|---|---|
| {x,y,z} | Set of discrete values: x, y and z. |
| [x,z] | Continuous values between x and z, inclusive. |
| [x:z] | Discrete values in increments of 1 between x and z. |
| [x:y:z] | Discrete values in increments of y between x and z. |

**Table 2. Audio Properties**

| Property Name | Description | Values |
|---|---|---|
| FrequencyOffset | Set or query a frequency offset from center frequency. | See RSA_API documentation for more information. |
| Mode | Set or query audio demodulation mode. | {'ADM_FM_8KHZ', 'ADM_FM_13KHZ', 'ADM_FM_75KHZ', 'ADM_FM_200KHZ', 'ADM_AM_8KHZ'} |
| Mute | Set or query whether audio mute is enabled or disabled. | {true, false} |
| Volume | Set or query volume level. | [0, 1] |

**Table 3. Configure Properties**

| Property Name | Description | Values |
|---|---|---|
| AutoAttenuationEnable | Set or query whether auto gain is enabled or disabled. | {true, false} |
| CenterFreq | Set or query center frequency. | See RSA_API documentation for more information. |
| ExternalRefEnable | Set or query whether external reference is enabled or disabled. | {true, false} |
| PreampEnable | Set or query whether preamp is enabled or disabled. | {true, false} |
| ReferenceLevel | Set or query reference level. | See RSA_API documentation for more information. |
| RFAttenuator | Set or query whether RF attenuator is enabled or disabled. | {true, false} |

**Table 4. DPX Properties**

| Property Name | Description | Values |
|---|---|---|
| Enable | Set or query whether DPX is enabled or disabled. | {true, false} |

**Table 5. GNSS Properties**

| Property Name | Description | Values |
|---|---|---|
| AntennaPower | Set or query whether antenna power is enabled or disabled. | {true, false} |
| Enable | Set or query whether GNSS is enabled or disabled. | {true, false} |

**Table 6. IQBlock Properties**

| Property Name | Description | Values |
|---|---|---|
| IQBandwidth | Set or query IQ bandwidth. | [0, 40 MHz] |
| IQRecordLength | Set or query amount of IQ data samples in each data acquisition. | See RSA_API documentation for more information. |

**Table 7. Spectrum Properties**

| Property Name | Description | Values |
|---|---|---|
| Enable | Set or query whether spectrum is enabled or disabled. | {true, false} |

**Table 8. Trigger Properties**

| Property Name | Description | Values |
|---|---|---|
| IFPowerLevel | Set or query IF power detection level. | See RSA_API documentation for more information. |
| TriggerMode | Set or query trigger mode. | {'freeRun', 'triggered'} |
| Position | Set or query trigger position. | [0%, 100%] |
| Source | Set or query trigger source. | {‘TriggerSourceExternal’, ‘TriggerSourceIFPowerLevel’} |
| Transition | Set or query trigger transition mode. | {‘TriggerTransitionLH’, ‘TriggerTransitionHL’, ‘TriggerTransitionEither’} |

**Table 9. Trkgen Properties**

| Property Name | Description | Values |
|---|---|---|
| Enable | Set or query whether tracking generator is enabled or disabled. | {true, false} |
| OutputLevel | Set or query output level of tracking generator. | See RSA_API documentation for more information. |

# Appendix B: Driver Functions

**Table 10. Alignment Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetAlignmentNeeded | Determine if an alignment is needed. | None | needed: returns true if alignment is needed. |
| RunAlignment | Run alignment for device. | None | None |
| GetWarmupStatus | Determine warm up status of device. | None | warmUp: returns true if device is warm up. |

**Table 11. Audio Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetData | Store audio data in data array. | **inSize:** maximum number of audio samples. | **data:** an array of audio data. **outSize:** number of audio samples in array. |
| GetEnable | Determine if audio is playing. | None | **enable:** returns true if audio is streaming to speaker. |
| Start | Start streaming audio speaker. | None | None |
| Stop | Stop streaming audio speaker. | None | None |

**Table 12. Configure Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetExternalRefFrequency | Return the external reference frequency in Hz. | None | **extFreq:** external reference frequency. |
| GetMaxCenterFreq | Return maximum frequency of RSA unit in Hz. | None | **maxCF:** maximum center frequency. |
| GetMinCenterFreq | Return minimum frequency of RSA unit in Hz. | None | **minCF:** minimum center frequency. |
| Preset | Set device to preset values. See RSA_API documentation for more information. | None | None |

**Table 13. Device Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| Connect | Connect RSA device. | **deviceID:** number between 0-19, indicating RSA device connected by USB 3.0 | None |
| Disconnect | Disconnect RSA device. | None | None |
| GetAPIVersion | Return API version number of device. | None | **apiVersion:** API version number. |
| GetDeviceNomenclature | Return name of device. | None | **nomenclature:** name of device. |
| GetDeviceNomenclatureW | Return wide name of device. | None | **nomenclature:** name of device. |
| GetDeviceSerialNumber | Return serial number of device. | None | **serialNum:** serial number of device. |
| GetEnable | Return data acquisition status of device. | None | **enable:** returns true if data acquisition is running. |
| GetEventStatus | Obtain event status of device. | **eventID:** desired event to determine if occurred. | **eventOccurred:** returns true if event occurred. **eventTimestamp:** timestamp of event. |
| GetFPGAVersion | Return FPGA version number of device. | None | **fpgaVersion:** FPGA version number. |
| GetFWVersion | Return firmware version number of device. | None | **fwVersion:** firmware version number. |
| GetHWVersion | Return hardware version number of device. | None | **hwVersion:** hardware version number. |
| OverTemperatureStatus | Determine if device is over temperature. | None | **overTemperature:** return if device is over temperature |
| PrepareForRun | Prepare device for data transfer. | None | None |
| Reset | Reset RSA device. | **deviceID:** RSA device being reset. | None |
| Run | Start data acquisition. | None | None |
| Search | Search for RSA devices connected to PC. | None | **numDevicesFound:** number of devices found. **deviceIDs:** an array of device IDs. **deviceSerial:** device serial number. **deviceType:** name of device. |
| StartFrameTransfer | Start data transfer. Call PrepareForRun first. | None | None |
| Stop | Stop data acquisition. | None | None |

**Table 14. DPX Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| Configure | Enable or disable DPX spectrum and spectrogram. | **enableSpectrum:** enable/disable DPX spectrum.<br>**enableSpectrogram:** enable/disable DPX spectrogram. | None |
| FinishFrameBuffer | This function must be called before getting a new frame buffer. | None | None |
| GetFrameBuffer | Return latest Frame Buffer. | None | **frameBuffer:** see Appendix D: DPX GetFrameBuffer. |
| GetFrameInfo | Return live update of frame count and FFT count. | None | **frameCount:** frame count<br>**fftCount:** FFT count |
| GetRBWRange | Return maximum and minimum RBW values. | **fspan:** span used to calculate RBW limits. | **minRBW:** minimum RBW allowed.<br>**maxRBW:** maximum RBW allowed. |
| GetSettings | Return current DPX spectrum settings. | None | **dpxSettings:** see Appendix E: DPX GetSettings. |
| GetSogramHiResLine | Return the specified spectrogram line after the system has stopped acquiring DPX data. | **lineIndex:** sogram line to retrieve.<br>**tracePoints:** length of sogram line to retrieve.<br>**firstValidPoint:** first point to retrieve from the sogram line. Set to 0 to retrieve the whole line. | **vData:** array containing the sogram line data.<br>**dataSF:** scaling factor for the sogram line data. |
| GetSogramHiResLineCountLatest | Return how many spectrogram lines are available. | None | **lineCount:** number of sogram lines. |
| GetSogramHiResLineTimestamp | Return timestamp of specified sogram line. | **lineIndex:** sogram line to retrieve a timestamp for. | **timestamp:** timestamp of the specified sogram line. |
| GetSogramHiResLineTriggered | Determine if trigger event occurred at specified sogram line. | **lineIndex:** sogram line to retrieve a trigger event for. | **triggered:** returns true if trigger event occurred for the specified sogram line. |
| GetSogramSettings | Return current DPX spectrogram settings. | None | **sogramSettings:** Appendix F: DPX GetSogramSettings. |
| IsFrameBufferAvailable | Determine availability of DPX frame. | None | **frameAvailable:** returns true if frame is available. |
| Reset | Reset DPX processor to default values. See | None | None |

| | | | |
|---|---|---|---|
| | RSA_API documentation for default values. | | |
| SetParameters | Set DPX processor parameters. | **fspan:** span in Hz.<br>**rbw:** resolution bandwidth in Hz.<br>**bitmapWidth:** width of bitmap in pixels.<br>**tracePtsPerPixel:** number of trace points in spectrum trace for each bitmap pixel.<br>**yUnit:** vertical units for bitmap.<br>**yTop:** vertical upper limit for bitmap.<br>**yBottom:** vertical lower limit for bitmap.<br>**infinitePersistence:** enable/disable for every data point to persist.<br>**persistanceTimeSec:** amount of time in seconds that previous signal will persist.<br>**showOnlyTrigFrame:** enable/disable to show only trigger frames. | None |
| SetSogramParameters | Set parameters for DPX spectrogram. | **timePerBitmapLine:** amount of time per bitmap line.<br>**timeResolution:** amount of time each sogram line represents.<br>**maxPower:** maximum power of sogram.<br>**minPower:** minimum power of sogram. | None |
| SetSogramTraceType | Set spectrogram trace type. | **traceType:** trace type to set for sogram. | None |
| SetSpectrumTraceType | Set spectrum trace type for specified spectrum trace. | **traceIndex:** specify trace to be set. 0, 1, or 2<br>**type:** trace type to set for spectrum trace. | None |
| WaitForDataReady | Wait for DPX data to be ready, otherwise timeout. | **timeoutMsec:** maximum time to wait for data to be ready, in milliseconds. | **ready:** returns true if the data is ready. |

**Table 15. GNSS Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| ClearNavMessageData | Clear navigation message in buffer. | None | None |
| Get1PPSTimestamp | Return 1PPS timestamp. | None | **isValid:** returns true if 1PPS is valid.<br>**timestamp1PPS:** timestamp of 1PPS. |
| GetHWInstalled | Determine if GNSS hardware is installed. | None | **installed:** returns true if GNSS hardware is installed. |
| GetNavMessageData | Return navigation message in buffer. | None | **msgLen:** length of navigation message.<br>**message:** NMEA message. |

**Table 16. IF Stream Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetActiveStatus | Determine if device is writing to disk. | None | **active:** returns true while streaming to disk. |
| SetDiskFileCount | Set number of files to generate. | **count:** number of files to stream to disk. | None |
| SetDiskFileLength | Set how long to record to disk in milliseconds. | **msec:** number of milliseconds to record data. | None |
| SetDiskFileMode | Set mode of IF stream. | **mode:** streaming mode. | None |
| SetDiskFilenameBase | Set name of file. | **base:** name of file streamed to disk. | None |
| SetDiskFilenameSuffix | Set suffix of file appended to name of file. | **suffixCtl:** suffix of file. | None |
| SetDiskFilePath | Set path of file. | **path:** location to save file. | None |
| SetEnable | Enable or disable streaming ADC to disk. | **enable:** enable/disable to stream ADC data to disk. | None |

**Table 17. IQ Block Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| AcquireIQData | Start data acquisition. | None | None |
| GetIQAcqInfo | Return current IQ header data. | None | **header:** see Appendix G: IQ Block GetIQAcqInfo. |
| GetIQData | Return IQ data in interleaved format. startIndex and length should not exceed RecordLength. | **startIndex:** starting index of IQ record.<br>**length:** number of IQ samples to acquire. | **iqData:** I and Q data at alternating indexes in array. |
| GetIQDataCplx | Return IQ data in interleaved format. startIndex and length should not exceed RecordLength. | **startIndex:** starting index of IQ record.<br>**length:** number of IQ samples to acquire. | **iqData:** I and Q data at alternating indexes in array. |
| GetIQDataDeinterleaved | Return IQ data in deinterleaved format. startIndex and length should not exceed RecordLength. | **startIndex:** starting index of IQ record.<br>**length:** number of IQ samples to acquire. | **iData:** I data array.<br>**qData:** Q data array. |
| GetIQSampleRate | Return sample rate. Based on bandwidth. | None | **iqSampleRate:** sample rate of device. |
| GetMaxIQBandwidth | Return maximum bandwidth of device. | None | **maxBandwidth:** maximum bandwidth. |
| GetMaxIQRecordLength | Return maximum record length of device. | None | **maxSamples:** maximum record length. |
| GetMinIQBandwidth | Return minimum bandwidth of device. | None | **minBandwidth:** minimum bandwidth. |
| WaitForIQDataReady | Wait for IQ data to be ready, otherwise timeout. | **timeoutMsec:** maximum time to wait for data to be ready, in milliseconds. | **ready:** returns true if data is ready. |

**Table 18. IQ Stream Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| ClearAcqStatus | Reset sticky bits in iqinfo for GetIQData. | None | None |
| GetAcqParameters | Return output bandwidth and sample rate in Hz. | None | **bwHz_act:** bandwidth.<br>**srSps:** sample rate. |
| GetDiskFileInfo | Return info about file output operation. | None | **fileinfo:** see Appendix H: IQ Stream GetDiskFileInfo. |
| GetDiskFileWriteStatus | Determine if device is writing to disk. isComplete is more designed for triggering. | None | **isComplete:** always returns true after first writing to disk is complete.<br>**isWriting:** returns true if writing to disk. |
| GetEnable | Determine if IQ stream processing is enabled. | None | **enable:** returns true if IQ stream processing is enabled. |
| GetIQData | Return IQ data in interleaved format. | **buffer:** buffer to store IQ data samples. | **iqdata:** I and Q data at alternating indexes in array.<br>**iqlen:** number of IQ data pairs.<br>**iqinfo:** see Appendix I: IQ Stream |

| | | | |
|---|---|---|---|
| | | | GetIQData. |
| GetIQDataBufferSize | Return maximum number of IQ sample pairs for GetIQData. | None | **maxSize:** maximum size of IQ data buffer. |
| GetMaxAcqBandwidth | Return maximum acquisition bandwidth. | None | **maxBandwidthHz:** maximum bandwidth |
| GetMinAcqBandwidth | Return minimum acquisition bandwidth. | None | **minBandwidthHz:** minimum bandwidth |
| SetAcqBandwidth | Set acquisition bandwidth in Hz. | **bwHz_req:** bandwidth. | None |
| SetDiskFileLength | Set how long to record to disk in milliseconds. | **msec:** number of milliseconds to record data. | None |
| SetDiskFilenameBase | Set name of file. | **filenameBase:** base filename for file output. | None |
| SetDiskFilenameBaseW | Set name of file. | **filenameBaseW:** base filename for file output. | None |
| SetDiskFilenameSuffix | Set suffix of file appended to name of file. | **suffixCtl:** suffix of file. | None |
| SetIQDataBufferSize | Set data buffer for IQ sample pairs. | **reqSize:** size of IQ data buffer. | None |
| SetOutputConfiguration | Set data type and output destination. | **dest:** destination for IQ sample output.<br>**dtype:** IQ data type. | None |
| Start | Initiate IQ stream processing and data output. | None | None |
| Stop | Terminate IQ stream processing and data output. | None | None |

**Table 19. Playback Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetReplayCompleted | Determine if file playback has completed. | None | **complete:** returns true if playback is complete |
| OpenDiskFile | Opens a .r3f file for playback and prepares for playback based on parameters. | **filename:** name of file to open.<br>**startPercentage:** starting location in file for playback.<br>**stopPercentage:** stopping location in file for playback.<br>**skipTimeBetweenFullAcquisitions:** amount of time to skip in file.<br>**loopAtEndOfFile:** enable/disable continuous looping of file.<br>**emulateRealTime:** enable/disable real time emulation mode. | None |

**Table 20. Power Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetHWStatus | Get any available battery HW info | None | **powerInfo:** see Appendix N: Power GetHWStatus. |

**Table 21. Reftime Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetCurrentTime | Return Unix time and internal timestamp. | None | **o_timeSec:** Unix time.<br>**o_timeNsec:** number of nanoseconds in specified second.<br>**o_timestamp:** internal timestamp. |
| GetIntervalSinceRefTimeSet | Return number of seconds since internal API time and timestamp was set. | None | **sec:** number of seconds. |
| GetReferenceTime | Return Unix time and internal timestamp set by user. | None | **refTimeSec:** Unix time.<br>**refTimeNsec:** number of nanoseconds in specified second.<br>**refTimestamp:** internal timestamp. |
| GetTimeFromTimestamp | Convert internal timestamp to Unix time. | **i_timestamp:** internal timestamp. | **o_timeSec:** Unix time.<br>**o_timeNsec:** number of nanoseconds in specified second. |
| GetTimestampFromTime | Convert Unix time to internal timestamp. | **i_timeSec:** Unix time.<br>**i_timeNsec:** number of nanoseconds in specified second. | **o_timestamp:** internal timestamp. |
| GetTimestampRate | Return clock rate of device. | None | **o_refTimestampRate:** clock rate. |
| SetReferenceTime | Set Unix time and internal timestamp. | **refTimeSec:** Unix time.<br>**refTimeNsec:** number of nanoseconds in specified second.<br>**refTimestamp:** internal timestamp. | None |

**Table 22. Spectrum Functions**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| AcquireTrace | Acquire trace for spectrum. | None | None |
| GetLimits | Get Spectrum limits. | None | **limits:** see Appendix J: Spectrum GetLimits. |
| GetSettings | Get Spectrum settings. | None | **settings:** see Appendix K: Spectrum GetSettings. |
| GetTrace | Get specified Spectrum trace data. | **trace:** trace index.<br>**maxTracePoints:** maximum number of trace points to retrieve. | **traceData:** spectrum trace data. |
| GetTraceInfo | Get spectrum trace information. | None | **traceInfo:** see Appendix L: Spectrum GetTraceInfo. |
| GetTraceType | Get specified Spectrum trace setting. | **trace:** trace index. | **enable:** returns true when trace is enabled. |

| | | | |
|---|---|---|---|
| | | | **detector:** detector type |
| SetDefault | Set Spectrum to default settings. See RSA_API documentation for more information. | None | None |
| SetSettings | Set Spectrum settings. | **settings:** See Appendix M: Spectrum SetSettings. | None |
| SetTraceType | Set specified Spectrum trace setting. | **trace:** trace index.<br>**enable:** enable/disable trace.<br>**detector:** detector type. | None |
| WaitForTraceReady | Wait for spectrum data to be ready, otherwise timeout. | **timeoutMsec:** maximum time to wait for data to be ready, in milliseconds. | **ready:** returns true when data is ready. |

**Table 23. Trigger Function**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| ForceTrigger | Force device to trigger. | None | None |

**Table 24. Trkgen Function**

| Function Name | Description | Inputs | Outputs |
|---|---|---|---|
| GetHWInstalled | Determine if tracking generator is installed | None | **installed:** returns true if tracking generator is found. |

# Appendix C: Error Values

The possible error return values are described below.

**Table 25. Error Return Status Values**

| ReturnStatus | Description |
|---|---|
| noError | Function has succeeded. |
| errorNotConnected | Device is not connected. |
| errorTimeout | Timeout has occurred. |
| errorTransfer | Device communication failed. |
| errorFileOpen | File did not open. |
| errorParameter | Parameter has an invalid value. |
| errorDataNotReady | Data is not ready to be queried. |
| errorInvalidCalibConstantFileFormat | Calibration constant file format is invalid. |
| errorMismatchCalibConstantsSize | Calibration constant size is invalid. |
| errorFailed | Function failed. |
| errorCRC | Calculated CRC and saved CRC do not match. |
| errorWriteCalConfigHeader | Calibration configuration header failed to write. |
| errorWriteCalConfigData | Calibration configuration data failed to write. |
| errorReadCalConfigHeader | Calibration configuration header failed to read. |
| errorReadCalConfigData | Calibration configuration data failed to read. |
| errorEraseCalConfig | Calibration configuration failed to erase. |
| errorCalConfigFileSize | Calibration configuration size is invalid. |
| errorChangeToFlashMode | Device failed to change to flash mode. |
| errorChangeToRunMode | Device failed to change to run mode. |
| errorIncompatibleFirmware | Firmware is not up to date. |
| errorStreamADCtoDiskFileOpen | Failure to open ADC streaming data file. |
| errorStreamADCToDiskAlreadyStreaming | Cannot change ADC file streaming parameters while streaming is active. |
| errorStreamADCToDiskBadPath | Nonexistant path or insufficient privileges to open file in specified path. |
| errorStreamADCToDiskThreadFailure | ADC streaming to disk thread failure. |
| errorRebootFailure | Failed to reboot instrument. |
| errorLOLockFailure | Local oscillator did not achieve lock. |
| errorPOSTFailureFPGALoad | Device internal HW configuration failure. |
| errorPOSTFailureHiPower | USB port power insufficient for device. |
| errorPOSTFailureI2C | Device internal control bus test failure. |
| errorPOSTFailureGPIF | Device internal data bus test failure. |
| errorPOSTFailureUsbSpeed | Device failed to connect as USB 3.0 |
| errorPlaceholder | Error checking has not been implemented. |
| notImplemented | Function does nothing. |

# Appendix D: DPX GetFrameBuffer

**Table 26. FrameBuffer Format**

| Field Name | Description |
|---|---|
| fftPerFrame | Number of FFTs performed in this frame. |
| fftCount | Total number of FFT performed since DPX acquisition started. |
| frameCount | Total number of DPX frames performed since DPX acquisition started. |
| timestamp | Acquisition timestamp for the current frame. |
| acqDataStatus | Acquisition data status for the current frame. |
| minSigDuration | Current minimum signal duration for 100% POI. |
| minSigDurOutOfRange | Minimum signal duration is currently out of range. |
| spectrumBitmapWidth | Width of the spectrum bitmap in pixels. |
| spectrumBitmapHeight | Height of the spectrum bitmap in pixels. |
| spectrumBitmapSize | Total number of pixels in the spectrum bitmap. |
| spectrumTraceLength | Number of trace points in spectrum trace. |
| numSpectrumTraces | Number of spectrum traces available. |
| spectrumEnabled | DPX Spectrum is enabled. |
| spectrogramEnabled | DPX Spectrogram is enabled. |
| spectrumBitmap | Vector containing the spectrum bitmap. |
| spectrumTraces | Vector containing the spectrum traces. |
| sogramBitmapWidth | Width of the spectrogram bitmap in pixels. |
| sogramBitmapHeight | Height of the spectrogram bitmap in pixels. |
| sogramBitmapSize | Total number of pixels in the spectrogram bitmap. |
| sogramBitmapNumValidLines | Number of valid lines in the spectrogram bitmap |
| sogramBitmap | Vector containing the spectrogram bitmap. |
| sogramBitmapTimestampArray | Vector containing the timestamps for each line in the spectrogram bitmap. |
| sogramBitmapContainTriggerArray | Vector containing the trigger events for each line in the spectrogram bitmap. |

# Appendix E: DPX GetSettings

**Table 27. dpxSettings Format**

| Field Name | Description |
|---|---|
| enableSpectrum | Checks current status for DPX Spectrum. |
| enableSpectrogram | Checks current status for DPX Spectrogram. |
| bitmapWidth | Width of the spectrum bitmap in pixels. |
| bitmapHeight | Height of the spectrum bitmap in pixels. |
| traceLength | Length of the spectrum trace. |
| decayFactor | Current decay factor. |
| actualRBW | Current RBW value in Hz. |

# Appendix F: DPX GetSogramSettings

**Table 28 sogramSettings Format**

| Field Name | Description |
|---|---|
| bitmapWidth | Width of the spectrogram bitmap in pixels. |
| bitmapHeight | Height of the spectrogram bitmap in pixels. |
| sogramTraceLineTime | Time per spectrogram trace line. |
| sogramBitmapLineTime | Time per spectrogram bitmap line. |

# Appendix G: IQ Block GetIQAcqInfo

**Table 29. acqInfo Format**

| Field Name | Description |
|---|---|
| acqDataStatus | See RSA_API documentation for more info about bit masking. |
| acquisitionTimestamp | Timestamp of current IQ data. |
| acquisitionUnixTime | Unix time of current IQ data. |

# Appendix H: IQ Stream GetDiskFileInfo

**Table 30. fileInfo Format**

| Field Name | Description |
|---|---|
| numberSamples | Number of IQ pairs written to file. |
| sampleOTimestamp | Timestamp of first sample written to file. |
| triggerSampleIndex | Sample index where trigger event occurred. |
| triggerTimestamp | Timestamp of the trigger event. |
| acqStatus | See RSA_API documentation for more info about bit masking. |
| filenames | Retrieve Data filename only. Header filename is currently not supported. |
| unixTime | Timestamp of first sample written to file. |
| triggerUnixTime | Unix time of the trigger event. |

# Appendix I: IQ Stream GetIQData

**Table 31. iqInfo Format**

| Field Name | Description |
|---|---|
| timestamp | Timestamp of first sample of block. |
| triggerCount | Number of trigger events occurring during block. |
| triggerIndices | List of sample indices where triggers(s) occurred. |
| scaleFactor | Convert integer data types to standard voltage values. |
| acqStatus | Acquisition status flags to indicate block status. See RSA_API documentation for more information. |
| unixTime | Unix time of first sample of block. |

# Appendix J: Spectrum GetLimits

**Table 32. limits Format**

| Field Name | Description |
|---|---|
| maxSpan | Maximum Span. |
| minSpan | Minimum Span. |
| maxRBW | Maximum RBW. |
| minRBW | Minimum RBW. |
| maxVBW | Maximum VBW. |
| minVBW | Minimum VBW. |
| maxTraceLength | Maximum trace length. |
| minTraceLength | Minimum trace length. |

# Appendix K: Spectrum GetSettings

**Table 33. settings Format**

| Field Name | Description |
|---|---|
| span | Span of spectrum. |
| rbw | RBW of spectrum. |
| enableVBW | Check current status of VBW. |
| vbw | VBW of spectrum. |
| traceLength | Length of the spectrum trace. |
| window | Windowing used in spectrum. See RSA_API documentation for more information. |
| verticalUnit | Trace output unit (dBm, Watt, Volt, Amp, dBmV) |
| actualStartFreq | Actual start frequency set for spectrum. |
| actualStopFreq | Actual stop frequency set for spectrum. |
| actualFreqStepSize | Actual frequency step size set for spectrum. |
| actualRBW | Actual RBW set for spectrum. |
| actualVBW | Actual VBW set for spectrum. |
| actualNumIQSamples | Actual number of IQ samples set for spectrum. |

# Appendix L: Spectrum GetTraceInfo

**Table 34. traceInfo Format**

| Field Name | Description |
|---|---|
| timestamp | Timestamp of first acquisition sample. |
| acqDataStatus | See RSA_API documentation for more info about bit masking. |
| unixTime | Unix time of first acquisition sample. |

# Appendix M: Spectrum SetSettings

**Table 35. settings Format**

| Field Name | Description |
|---|---|
| span | Span of spectrum. |
| rbw | RBW of spectrum. |
| enableVBW | Check current status of VBW. |
| vbw | VBW of spectrum. |
| traceLength | Length of the spectrum trace. |
| window | Windowing used in spectrum. See RSA_API documentation for more information. |
| verticalUnit | Trace output unit (dBm, Watt, Volt, Amp, dBmV) |

# Appendix N: Power GetHWStatus

**Table 36. powerStatus Format**

| Field Name | Description |
|---|---|
| externalPowerPresent | Status of external power connection. |
| batteryPresent | Battery installed. |
| batteryChargeLevel | Current charge of the battery in percent. |
| batteryCharging | Status of battery charging. |
| batteryOverTemperature | Battery overheating. |
| batteryHardwareError | Hardware error status from battery charging circuit |