

Tektronix RSA 306

MATLAB Driver

User Guide

API Build 3.4.0253

Version 1.02

Updated 1/16/2015 – Matt Sowa

Updated 1/14/2015 – Matt Sowa

Updated 10/9/2014 – Matt Sowa

Created 9/12/2014 – John Akre

Table of Contents

Overview	1
Requirements.....	1
Installation	1
How to Use the Driver.....	1
Example: Generate IQ Data	3
Appendix A: Driver Properties	4
Appendix B: Driver Functions.....	5
Appendix C: Error Values	8
Appendix E: DPX Settings Format	10
Appendix F: DPX Spectrogram Settings Format.....	10

Overview

The RSA306 MATLAB Driver allows communication between MATLAB and the RSA300 API. The driver uses a generic driver from MATLAB's Instrument Control Toolbox to make individual calls to the API. This driver supports version 3.4.0253 of the API.

Requirements

The following hardware, software and files are needed to use the RSA306 MATLAB driver.

1. RSA306 device.
2. Version 3.4.0253 of the RSA300API.
3. MATLAB and the Instrument Control Toolbox. (MATLAB R2014a has been tested).
4. MATLAB instrument driver file (RSA306_Driver_3_4_0253.mdd).
5. Folder locations of the following files:
 - a. BaseDSPL.dll (version 3.4.0253)
 - b. GPMeasDSP.dll (version 3.4.0253)
 - c. ippcore-8.0.dll
 - d. ippi-8.0.dll
 - e. ippiy8-8.0.dll
 - f. ippm-8.0.dll
 - g. ippmu8-8.0.dll
 - h. ippv-8.0.dll
 - i. ippsm7-8.0.dll
 - j. ippvm-8.0.dll
 - k. ippvmmx-8.0.dll
 - l. RSA300API.dll (version 3.4.0253)
 - m. RSA300API.h (version 3.4.0253)

Installation

MATLAB and the Instrument Control Toolbox are assumed to be previously installed. The MATLAB R2014a version has been tested with this driver. The MATLAB instrument driver file (RSA306_Driver_3_4_0253.mdd) can be placed in the working directory or on MATLAB's path. Also, the files listed under step 5 of the requirements must be added to MATLAB's path.

How to Use the Driver

Using the Driver requires the following basic steps:

1. Add the folder paths containing the files listed under step 5 of the requirements section to the search path.

2. Create the device object.
 3. Connect to the device object.
 4. Set/query properties or evaluate functions.
 5. Disconnect from the device object.
 6. Delete and clear the device object from memory.
1. Add the folders containing the required files to the search path with the `addpath('folderName')` function. The required files are listed in the overview section.

`addpath('C:\Folder');`

2. Create the device object using the `icdevice('driver', 'id')` function.
 - a. **'driver'** is the file name for of the MATLAB Instrument Driver.
 - b. **'id'** is the optional connection ID if you wish to connect to a specific device when more than one are attached.

`device = icdevice('RSA306_Driver_3_4_0253', '0');`

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

`connect(device);`

4. Set/Query properties or evaluate functions.
 - a. Set proeprties using the `set(obj, 'propertyName', propertyValue)` function.
 - i. **obj** is the device group object.

`set(device.Acquisition, 'CenterFrequency', 2.4453e9);`

- b. Query properties using the `get(obj, 'propertyName')` function.

`centerFrequency = get(device.Acquisition, 'CenterFrequency');`

- c. Evaluate functions using the `invoke(obj, 'functionName', input1, input2, ...)` function.

`invoke(device.Waveform, '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.

%Add the folder(s) containing the required files to the search path. The files are listed in the overview section.

```
addpath('C:\ReqFileFolder');
```

%Create the device object

```
device = icdevice('RSA306_Driver_3_4_0253');
```

%Connect to the device object

```
connect(device);
```

%Set the center frequency to 2.4453GHz

```
set(device.Acquisition, 'CenterFrequency', 2.4453e9);
```

%Set the reference level to -60dB

```
set(device.Amplitude, 'ReferenceLevel', -60);
```

%Set the IQ record length to 2000 samples

```
set(device.Acquisition, 'IQRecordLength', 2000);
```

%Set the trigger mode to free run

```
set(device.Trigger, 'TriggerMode', 'freeRun');
```

%Begin data acquisition

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

%Wait for the IQ data to be ready. It has a timeout value of 2000ms

```
invoke(device.Waveform, 'WaitForIQDataReady', 2000);
```

%Get IQ data. The start index is set to 0 and the record length is set to 2000 samples.

```
[I, Q] = invoke(device.Waveform, 'GetIQDataDeinterleaved', 0, 2000);
```

%Multiply the Q data by sqrt(-1) to make it imaginary

```
Q = Q*i;
```

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

IQ = I+Q;

%Stop data acquisition

invoke(device.Waveform, 'Stop');

%Disconnect from the device object

disconnect(device);

%Delete and clear the device object from memory

delete(device);

clear('device');

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. Acquisition Group Properties

Property Name	Description	Values
CenterFrequency	Sets or queries the center frequency.	[0 Hz, 6.2 GHz]
IQRecordLength	Sets or queries the IQ record length.	[1 sample:5.25 Msamples]
IQSampleRate	Queries the IQ sampling Rate	{56 MHz}
IQBandwidth	Sets or queries the IQ bandwidth	Up to 40MHz

Table 3. Amplitude Group Properties

Property Name	Description	Values
ReferenceLevel	Sets or queries the reference level.	[-130 dB, 30 dB]

Table 4. Audio Group Properties

Property Name	Description	Values
Mode	Sets or queries the audio demod mode.	{'ADM_FM_8KHZ', 'ADM_FM_13KHZ', 'ADM_FM_75KHZ', 'ADM_FM_200KHZ', 'ADM_AM_8KHZ'}
Mute	Sets or queries the mute status.	{true, false}
Volume	Sets or queries the volume level.	[0, 1]

Table 5. Connection Group Properties

Property Name	Description	Values
SerialNumber	Gets the serial number for the connected device.	String

Table 6. Trigger Group Properties

Property Name	Description	Values
IFPowerLevel	Sets or queries the IF power trigger level.	[-130 dB, 30 dB]
TriggerMode	Sets or queries the trigger mode.	{'freeRun', 'triggered'}
Position	Sets or queries the trigger position.	[0%, 100%]
Source	Sets or queries the trigger source.	{'TriggerSourceExternal', 'TriggerSourceIFPowerLevel'}
Transition	Sets or queries the trigger transition mode.	{'TriggerTransitionLH', 'TriggerTransitionHL', 'TriggerTransitionEither'}

Appendix B: Driver Functions

Table 7. Audio Group Functions

Function Name	Description	Inputs	Outputs
StartAudio	Starts streaming audio to the speaker	None	None
StopAudio	Stops streaming audio to the speaker	None	None

Table 8. DPX Group Functions

Function Name	Description	Inputs	Outputs
Configure	Configures whether DPX spectrum and/or spectrogram are enabled.	spectrumEnabled := DPX spectrum enabled. sogramEnabled := DPX spectrogram enabled.	None
FindRBWRange	Finds the max and min allowed RBW for the given span.	span := Span used to calculate RBW limits.	minRBW := minimum RBW allowed. maxRBW := maximum RBW allowed.
FinishFrameBuffer	This function must be called before getting a new DPX frame buffer.	None	None
GetFrameBuffer	Retrieves the latest DPX Frame Buffer.	None	FrameBuffer := See Appendix D for buffer struct format.

GetSettings	Retrieves the current DPX spectrum settings.	None	Settings := See Appendix E for settings struct format.
GetSogramHiResLine	Retrieves the specified spectrogram line after the system has Stopped and acquired DPX data.	lineIndex := sogram line to retrieve. tracePoints := length of sogram line to retrieve. firstValidPoint := first point to retrieve from the sogram line. Set to 0 to retrieve the whole line.	sogramLine := array containing the sogram line data. scalingFactor := scaling factor for the sogram line data.
GetSogramHiResLineCountLatest	This function reports how many spectrogram lines are available.	None	lineCount := number of available spectrogram lines.
GetSogramHiResLineTimestamp	Gets the timestamp for the specified sogram line.	lineIndex := sogram line to retrieve a timestamp for.	timestamp := timestamp for the specified sogram line.
GetSogramHiResLineTriggered	Reports if the specified sogram line contains a trigger event.	lineIndex := sogram line to retrieve trigger event for.	trigger := trigger event for the specified sogram line.
GetSogramSettings	Retrieves the current DPX spectrogram settings.	None	Settings := See Appendix F for settings struct format.
IsFrameBufferAvailable	Retrieves the status of DPX buffer processing.	None	bufferAvailable : = returns true if buffer processing is complete.
ResetDPX	Resets the DPX processor to defaults.	None	None
SetDPXEnabled	Enables the DPX processor.	None	None
SetParameters	Sets DPX processor parameters.	fspan := Current span to process. rbw := resolution bandwidth. bitmapWidth := Width of the bitmap. tracePtsPerPixel := number of points in	None

		<p>spectrum trace for each bitmap pixel.</p> <p>yUnit := vertical units for the bitmap.</p> <p>yTop := vertical upper limit for the bitmap.</p> <p>yBottom := vertical lower limit for the bitmap.</p> <p>persistanceTimeSec := amount of time in seconds that the spectrum bitmap will persist.</p> <p>showOnlyTrigFrame := enable/disable DPX triggering.</p>	
SetSogramParameters	Sets parameters for the DPX spectrogram.	<p>timePerBitmapLine := The amount of time represented by one sogram bitmap line.</p> <p>timeResolution := The desired time resolution</p> <p>maxPower := The maximum power shown on the spectrogram.</p> <p>minPower := The minimum power shown on the spectrogram.</p>	None
SetSogramTraceType	Sets the trace type for the spectrogram.	traceType := traceType to set for the DPX spectrograms.	None
SetSpectrumTraceType	Sets the trace type for the specified spectrum trace.	<p>trace := DPX spectrum trace to set the type for. Currently only the first trace is supported.</p> <p>type := traceType to set for the DPX spectrum trace.</p>	None
WaitForDPXDataReady	Waits for DPX data acquisition to complete.	timeout := time to wait for acquisition to complete.	DataReady := reports if the data is ready or not.

Table 9. Trigger Group Functions

Function Name	Description	Inputs	Outputs
---------------	-------------	--------	---------

ForceTrigger	Forces a trigger to occur in the data stream.	None	None
--------------	---	------	------

Table 10. Waveform Group Functions

Function Name	Description	Inputs	Outputs
GetIQData	Stores I and Q data in an array. Their values are at alternating indexes of the array.	index := index in the record to start receiving data from. length := amount of IQ data samples to query.	IQData := an array of alternating I and Q data. I values are at odd indexes. Q values are at even indexes. The Q data is not imaginary.
GetIQDataDeinterleaved	Stores I and Q data in separate arrays.	index := index in the record to start receiving data from. length := amount of IQ data samples to query.	iData := an array of I data. qData := an array of Q data. The Q data is not imaginary.
Run	Start input signal acquisition	None	None
Stop	Stop input signal acquisition	None	None
WaitForIQDataReady	Waits for IQ data to be ready or until timeout.	timeout := maximum time to wait for data to be ready. Measured in ms.	dataReady := returns 1 when data is ready. Returns 0 when data is not ready.

Appendix C: Error Values

The possible error return values are described below.

Table 11. Error ReturnStatus 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.
errorBootLoaderNotRunning	Bootloader is not running.
errorPlaceholder	Error checking has not been implemented.
notImplemented	Function does nothing.

Appendix D: DPX FrameBuffer Format

DPX.GetFrameBuffer returns a Frame Buffer, the format is described below.

Table 12. DPX FrameBuffer Values

Field Name	Description
fftPerFrame	Average number of FFTs processed per frame.
fftCount	Number of FFTs processed so far.
frameCount	Frames produced so far.
timestamp	Timestamp for the current frame.
acqDataStatus	Acquisition Status parameters 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	Length of the spectrum trace.
numSpectrumTraces	Number of spectrum traces available. (Currently the MATLAB driver only supports one trace.)
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 Settings Format

DPX.GetSettings returns a DPX Settings struct, the format is described below.

Table 13. DPX Settings Values

Field Name	Description
enableSpectrum	DPX Spectrum is enabled.
enableSpectrogram	DPX Spectrogram is enabled.
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.

Appendix F: DPX Spectrogram Settings Format

DPX.GetSogramSettings returns a DPX Sogram Settings struct, the format is described below.

Table 14. DPX Spectrogram Settings Values

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.