Tektronix®

RSA7100A Real-Time Spectrum Analyzer Application Programming Interface (API)

Programmer Manual

077-1496-00

Tektronix®

RSA7100A Real-Time Spectrum Analyzer Application Programming Interface (API)

Programmer Manual

www.tek.com

077-1496-00

Contacting Tektronix

Tektronix, Inc.
14150 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit www.tek.com to find contacts in your area.

Table of Contents

Preface

In the typical API scenario, the custom user program calls functions from the API to fetch data. This is known as a “pull” model. The user’s program must “pull” the data from the instrument. This is not the case with the RSA7100A IQFlow™ streaming API.

The RSA7100A IQFlow™ streaming API uses a “push” model to “push” the data to the user’s code by calling user supplied callback functions. These callback functions have immediate access the latest acquired data and can then use it however the user has programmed.

IQ data is structured as an array of 16-bit signed integer pairs and needs to be scaled by multiplying it with a scaling factor. The AuxData structure returned by the USERDATA_GetAuxData() function contains this scaling factor.

Function calls

dataCallbackFxn	Type definition for the user's data callback function. Called automatically by the API when new data is available.
Declaration:	dataCallbackFxn(IQDataPtr data, size_t size, size_t startOfBlockSample, bool streamOverload, double sampleRate, double centerFreq);
Parameters:	
data	IQDataPtr structure. Provides the IQ pair data block.
size	size_t of the IQ pair data block.
startofBlockSample	size_t sample number of the first sample in the IQ data block.
streamOverload	bool. True indicates an IQ data block was missed. False indicates no skipped IQ data block.
sampleRate	double of the sample rate for the current acquisition.
centerFreq	double of the center frequency for the current acquisition.

triggerCallbackFxn	Type definition for the user's trigger callback function. Called automatically by the API when the RSA is triggered.
Declaration:	triggerCallbackFxn(size_t triggerSample);
Parameters:	
Triggersample	size_t of the sample number for the current trigger event.

USERDATA_Connect()	Connects the user's callback functions to the IQFlow™ streaming API.
Declaration:	ReturnStatus USERDATA_Connect(size_t &dataSize, dataCallbackFxn dCbFxn, triggerCallbackFxn tCbFxn);
Parameters:	
dataSize	size_t passed by reference. Sets the buffer size of the IQ pair data block. This value is bound between 2^{21} to 2^{28}.
dCbFxn	dataCallbackFxn. Sets the user supplied callback function to be called by the API each time a new data block is available. Can be set to null if real-time IQ data is not required.
tCbFxn	triggerCallbackFxn. Sets the user supplied callback function to be called by the API each time a trigger event occurs. Can be set to null if trigger timestamp is not required.
Return values:	
noError	The function completed successfully.
errorBufferTooSmall	The buffer size specified is too small.
errorBufferTooLarge	The buffer size specified is too large.

USERDATA_Disconnect()	Disconnects the user's callback functions from the SignalVu-PC streaming API.
Declaration:	USERDATA_Disconnect();
Return values:	
noError	The function completed successfully.

USERDATA_GetAuxData() Fetches a data structure containing metadata for the acquired data.

Declaration: ReturnStatus USERDATA_GetAuxData(AuxData &auxData);

Parameters:

AuxData structure defined as follows:

Float version	Version of structure
bool skippedFrame:	True indicates a skipped frame. False indicates no skipped frame.
bool adcOverrange:	True indicates adc over range occurred. False indicates no adc overrange.
bool refFreqUnlock:	—
bool ifPowerOverloadStartAcq;	IF power overload occurred during instrument initialization.
bool ifPowerOverloadRuntime;	IF power overload occurred during run time.
bool dmaOverflow;	True indicates dma overflow occurred. False indicates no dma overflow.
bool gpsStatus;	True indicates GPS is on. False indicates GPS is off.
double gpsLat;	Returns GPS latitude if GPS status is true.
double gpsLong;	Returns GPS longitude if GPS status is true.
double gpsAlt;	Returns GPS altitude if GPS status is true.
bool validTimeRef;	Time source used is valid.
uint64_t seconds;	Returns time of IQ data block in seconds (based off posix time) if GPS status is true.
uint64_t nSeconds;	Nanosecond resolution of current IQ data block.
uint64_t timestamp;	Returns the timestamp of the first IQ pair in data block.
uint64_t timeRefTimestamp;	Returns the current time reference source timestamp associated with time.
double sampleRate;	Returns the sample rate of the current acquisition.
double cf;	Returns the center frequency of the current acquisition.
double scalingFactor;	Returns the scaling factor for the IQ data. Multiply the signed 16-bit integer values by this to scale the data factor.

Return Values:

noError:	The function completed successfully.
errorInvalidShared-Memory:	No data found in auxiliary data.
errorTimedOut:	The function timed out.
errorNotConnected:	The function had no valid connection to SignalVu-PC.

Additional Detail: GFRM_OFF (0) is returned when GNSS source is not selected.

USERDATA_GetData ()	Fetches the latest IQ data using a traditional pull method.
Declaration:	ReturnStatus USERDATA_GetData(IQDataPtr data, size_t size, size_t &startofBlockSample, bool &streamOverload, double &sampleRate, double &centerFreq);
Parameters:	
data	IQDataPtr structure. Provides the IQ pair data block.
size	size_t of the number of samples in the IQ pair data block.
startofBlockSample	size_t sample number of the first sample in IQ data block.
streamOverload	bool. True indicates an IQ data block was missed. False indicates no skipped IQ data block.
sampleRate	double of the sample rate for the current acquisition.
centerFreq	double of the center frequency for the current acquisition.
Return values:	
noError	The function completed successfully.
errorBufferTooSmall	The buffer size specified is too small.
errorBufferTooLarge	The buffer size specified is too large.

USERDATA_FindData ()	Searches for the sample number passed to it in the API's local circular buffer and returns a data set containing the requested sample, if it can be found.
Declaration:	ReturnStatus USERDATA_FindData(IQDataPtr data, size_t size, size_t findSample, size_t &startofBlockSample, bool &sreamOverlaod, double &sampleRate, double &centerFreq);
Parameters:	
data	IQDataPtr structure. Provides the IQ pair data block.
size	size_t of the number of samples in the IQ pair data block.
findSample	—
startofBlockSample	size_t sample number of the first sample in the IQ data block.
streamOverload	bool. True indicates an IQ data block was missed. False indicates no skipped IQ data block.
sampleRate	double of the sample rate for the current acquisition.
centerFreq	double of the center frequency for the current acquisition.
Return values:	
noError	The function completed successfully.
errorInvalidSize	The function parameter size is 0.
errorInvalidDataRate	The function parameter sampleRate is 0 or negative.
errorStaleTrigger	The function failed to acquire the trigger since the data has already passed.
errorTimedOut	The function timed out while looking for a trigger.
unknownError	The function received an unknown error.

Index