2023

Intan RHX TCP Guide

Control software and hardware using text commands that can be sent from any language with TCP functionality.

Connect the Intan software to a different application on the host computer or a different computer via networking.

Stream acquired data to the client application for a real-time alternative to writing to disk.

3.3.1

Contents

TCP Communication Overview

The IntanRHX software features the ability to control almost every aspect through TCP (Transmission Control Protocol) commands, in addition to the traditional Graphical User Interface. This is accomplished by the IntanRHX software acting as a server that an external client can connect to as a client. Through this document we will use MATLAB as the platform for our client software, but any language that has the ability to connect as a client via TCP and receive/send data can be similarly used.

Note that when transferring data via TCP, at high enough data rates TCP can take too long for the data transfer to keep up with real-time acquisition. This issue is most relevant when multiple channels have TCP data output enabled and acquisition is run at a high sample rate. For example, at Intan we've seen that with around 10 channels of data at 30 kS/s, real-time data transfer begins to slow down, causing a data backup and for transferred data to be delayed. For cases in which large amounts of data need to be transferred to an external application, a more efficient alternative would be to save data in File Per Signal Type or File Per Channel format, with a low WriteToDiskLatency, and to write this external application to read that data from disk in real-time. When large amounts of data are being transferred, this read-from-disk method can generally keep up with a higher data-rate than TCP.

Opening IntanRHX Server

Before any TCP communication can take place, the IntanRHX host software must open a server to listen for connections from any client software. From the Network menu at the top of the IntanRHX window, select Remote TCP Control to open the Remote TCP Control dialog. There are 3 different server connections that can be opened:

1. "Commands" server – all commands (Execute, Get/Set, LiveNotes) are communicated through this socket. In order for any TCP control of the operation of the software or controller to occur, this socket must be connected. Default port: 5000
2. "Waveform Output" server – all waveform data for all TCP-output-enabled channels (for amplifier signals waveform data includes Wideband, Lowpass, Highpass, DC, and Stimulation data) are communicated through this socket. The data rate of this socket is predetermined, depending on how many channels and filter bands are enabled, as well as the sample rate; the actual activity on these channels has no impact on how much data is sent per second. For example, the same amount of waveform data will be sent for rapidly firing tissue as completely flatlined tissue. Default port: 5000. Each block of data (128 data frames, each of which contains a timestamp and a sample from every enabled channel) starts with a constant 32-bit unsigned integer 'magic number', which is 0x2ef07a08. This should be used during reading to ensure that alignment with each incoming data block is preserved.
3. "Spike Output" server – all spike data for all TCP-output-enabled amplifier channels are communicated through this socket. The data rate of this socket is not predetermined, and changes depending on how much spiking activity is occurring on enabled channels. For example, rapidly firing tissue with a low spike threshold will result in much more spike data being sent than for flatlined tissue with a high spike threshold. Default port: 5000. Each chunk of spike data (the native name of the spiking channel, the timestamp of the spike, and the spike ID of the spike) starts with a constant 32-bit unsigned integer 'magic number', which is 0x3ae2710f. This

should be used during reading to ensure that alignment with each incoming spike chunk is preserved.

All 3 servers have the same default host address: “127.0.0.1”, a.k.a. “localhost”, which refers to the same computer that the IntanRHX software is running on.

Example: Connecting to Commands server on localhost

```
tcommand = tcpclient('localhost', 5000);
```

Clear All Data Outputs

Before any TCP channel outputs are enabled, it’s a good idea to clear all data outputs first. This ensures that any channels that may have been previously enabled are no longer enabled, so that the only channels with data output are those explicitly enabled later on by the client.

Example: Clearing all TCP data outputs

```
write(tcommand, uint8('execute clearalldataoutputs'));
```

Enable Data Output for Desired Channel(s)

Each channel (and filter band, if a non-wideband waveform for an amplifier channel is desired) must be individually enabled for TCP output.

Example: Setting Analog-In-4 and highpass of A-008 data outputs enabled

```
write(tcommand, uint8('set a-008.tcpdataoutputenabledhigh
true'));

write(tcommand, uint8('set analog-in-4.tcpdataoutputenabled
true'));
```

Start Controller Running

Before any data that can be output via TCP can be acquired, the controller must start running.

Example: Start controller running

```
write(tcommand, uint8('set runmode run'));
```

Read N DataBlocks From Desired Channel(s)

While the controller is running, the client application should periodically check how much data is available on the TCP data output port. When an arbitrary amount of data is available (for this example,

10 data blocks), the client application should read and parse that data. Each data block has 128 timesteps of data, and examples of how to calculate the size in bytes of each data block can be found in the provided MATLAB example scripts.

Example: Read 10 DataBlocks

```
if twaveformdata.BytesAvailable >= waveformBytes10Blocks
    waveformArray = read(twaveformdata, waveformBytes10Blocks);
end
```

Stop Board Running

When no more data acquisition is desired, the controller should stop running.

Example: Stop controller running

```
write(tcommand, uint8('set runmode stop'));
```

Syntax of commands

Each command can be sent individually, or multiple commands can be sent in one transmission. In this case, each command needs to be separated by a semicolon. For example, *set a-000.recordingenabled true; set a-001.recordingenabled false*

Get: "get" + *parameter*

Returns (from server to client) the current value of the parameter (true, false, text, or a numerical value depending on the parameter), or an error if the structure of the command is wrong (more or less than 1 word following "get").

Set: "set" + *parameter* + *value*

Sets (from client to server) the parameter to the desired value. An error will only be returned to the client if the syntax structure of the command is wrong (more or less than 2 words following "set").

Execute: "execute" + *action* or "execute" + action + parameter

Executes (from client to server) the desired action. An error will only be returned to the client if the syntax structure of the command is wrong (more or less than 1 word following "execute" if the execute command doesn't use a parameter, or more or less than 2 words following "execute" if the execute command does use a parameter). As opposed to a "set" command which changes the state of a variable in the software, an "execute" command immediately triggers an event to be activated.

LiveNotes: "livenotes" + *note*

Appends the given *note* alongside a *timestamp* to a .txt file corresponding to the current recording session. The syntax of the text file is:

time1 note1

time2 note2 …

Each line is separated by the characters "\r\n", and the whitespace between *time* and *note* is a single space. *Time* is a single-decimal precision number measuring the time in seconds since this recording session began. Due to USB communication delay, this measurement can be no more precise than 1/30th of a second, so only a single decimal point of precision is given.

If this is the first note of the recording session, the .txt file is created prior to appending the note. The name of the file is *basefilename* + "_" + *timestamp* (the same filename as the saved .rhx file, but with a .txt extension instead). If the board is not recording when this command is set, the command is ignored.

Dictionary of parameters (used in "get" and/or "set" commands)

System Wide

Intrinsic variables

Type

- Which type of controller is interfacing with the IntanRHX software
- *"Get" command only – cannot be changed without restarting the IntanRHX software*
- Returned values:
    - "ControllerRecordUSB2" – Intan USB Interface Board
    - "ControllerRecordUSB3" – Intan 512-ch or 1024-ch Recording Controller
    - "ControllerStimRecord"- Intan 128-ch Stimulation/Recording Controller

Synthetic

- Whether or not the board being controlled by the IntanRHX software is synthetic. If no Intan controller can be detected over USB (or if there's a problem with the drivers allowing communication with the controller), the user will be able to select a type of controller to synthesize for testing purposes
- *"Get" command only – cannot be changed without restarting the IntanRHX software*
- Returned values:
    - "True" – Controller is synthetic and all data is being generated by the software
    - "False" – Controller is not synthetic and all data is streaming from a physical device

Playback

- Whether or not the software is running in playback mode. Playback mode allows for pre-recorded data files to be opened and viewed as if it were streaming in realtime
- *"Get" command only – cannot be changed without restarting the IntanRHX software*
- Returned values:
    - "True" – Software is running in playback mode and playing an already-recorded data file
    - "False" – Software is not playing data back, and instead is streaming incoming data from connected hardware

SampleRateHertz

- The sampling rate of the RHD/RHS amplifiers. This value also controls stimulation time resolution for the Stimulation/Recording controller (1 / SampleRateHertz)
- *"Get" command only – cannot be changed without restarting the IntanRHX software*
- Returned values:
    - "1000" Hz (not applicable to Stimulation/Recording Controller)
    - "1250" Hz (not applicable to Stimulation/Recording Controller)
    - "1500" Hz (not applicable to Stimulation/Recording Controller)
    - "2000" Hz (not applicable to Stimulation/Recording Controller)
    - "2500" Hz (not applicable to Stimulation/Recording Controller)
    - "3000" Hz (not applicable to Stimulation/Recording Controller)
    - "3333" Hz (not applicable to Stimulation/Recording Controller)
    - "4000" Hz (not applicable to Stimulation/Recording Controller)
    - "5000" Hz (not applicable to Stimulation/Recording Controller)
    - "6250" Hz (not applicable to Stimulation/Recording Controller)
    - "8000" Hz (not applicable to Stimulation/Recording Controller)
    - "10000" Hz (not applicable to Stimulation/Recording Controller)
    - "12500" Hz (not applicable to Stimulation/Recording Controller)
    - "15000" Hz (not applicable to Stimulation/Recording Controller)
    - "20000" Hz
    - "25000" Hz
    - "30000" Hz

StimStepSizeMicroAmps

- The stimulation step size, which also controls stimulation current range (±255 * StimStepSizeMicroAmps)
- *"Get" command only – cannot be changed without restarting the IntanRHX software*
- *Only applies to Stimulation/Recording Controllers*
- Returned values:
    - "0.01" µA
    - "0.02" µA
    - "0.05" µA
    - "0.1" µA
    - "0.2" µA
    - "0.5" µA
    - "1" µA
    - "2" µA
    - "5" µA
    - "10" µA

ExpanderConnected

- Whether or not an I/O expander is connected to the controller
- *"Get" command only – cannot be changed without physically connecting an expander and restarting the IntanRHX software*
- Returned values:

- "True" – An expander is connected
- "False" – No expander is connected

Version

- The version of the IntanRHX software
- *"Get" command only – cannot be changed from outside the software's source code*
- Returned values:
  - "M.m.r" where M is major version, m is minor version, and r is revision. For example, Beta release returns "0.9.0"

UploadInProgress

- Whether or not an upload (UploadBandwidthSettings, UploadChargeRecoverySettings, UploadAmpSettleSettings, or UploadStimParameters) is still in progress. Since the same USB bus is used for all these uploads as well as data transmission, users should ensure this value is False before performing an upload or starting the controller's data acquisition. If an upload or setting the controller to run is attempted, that function will be aborted and an error message will be sent to the client
- *"Get" command only – cannot be changed directly, only by waiting for a previously started upload to complete*
- Returned values:
  - "True" – An upload execute command is still in progress
  - "False" – No upload execute command is currently in progress

HeadstagePresent

- Whether or not at least one headstage has been detected on the connected system
- *"Get" command only – cannot be changed directly, only by connecting/disconnecting a headstage and rescanning ports*
- Returned values:
  - "True" – At least one headstage has been detected
  - "False" – No headstage has been detected

Saving Data

FileFormat

- Which file format will be used to save recorded data to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - "Traditional" – saves all waveforms and settings in one file **(Default)**
  - "OneFilePerSignalType" – saves a separate file for each signal type, for example all amplifier recordings in one file and controller digital inputs in another file
  - "OneFilePerChannel" – saves a separate file for each channel

WriteToDiskLatency

- Level of latency between when data is fully processed and it is written to disk

- More latency allows for more efficient writing to disk due to fewer, larger writes having less overhead than many, smaller writes. However, if data is being read during acquisition this will also increase the amount of time before data is fully written to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "Highest" – Most latency before writing for best performance **(Default)**
    - "High" – High degree of latency
    - "Medium" – Medium degree of latency
    - "Low" – Low degree of latency
    - "Lowest" – Least latency before writing for other applications to have most immediate access to written data

CreateNewDirectory

- Whether a new save directory whose name ends with a timestamp is created for each recording
- The default, recommended behavior is that a new directory will be created. This is safest because when recording multiple sessions, each session will have a unique directory name to be written to. If this value is set False, .rhd/.rhs and .dat files will be written directly to the location set by Filename.Path, potentially overwriting any other files of the same name in that location
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – New directory will be created at the start of each recording **(Default)**
    - "False" – Data will be saved directly into the directory set by Filename.Path

SaveWidebandAmplifierWaveforms

- Whether or not wideband amplifier data will be saved to disk
- "*Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – Wideband amplifier data will be saved to disk **(Default)**
    - "False" – Wideband amplifier data will not be saved to disk

SaveLowpassAmplifierWaveforms

- Whether or not lowpass amplifier data will be saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – Lowpass amplifier data will be saved to disk
    - "False" – Lowpass amplifier data will not be saved to disk **(Default)**

LowpassWaveformDownsampleRate

- What the rate of downsampling will be when saving lowpass amplifier data to disk.
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*

- Acceptable values:
  - "1" – no downsampling will occur **(Default)**
  - "2" – every 2nd sample will be saved to disk
  - "4" – every 4th sample will be saved to disk
  - "8" – every 8th sample will be saved to disk
  - "16" – every 16th sample will be saved to disk
  - "32" – every 32nd sample will be saved to disk
  - "64" – every 64th sample will be saved to disk
  - "128" – every 128th sample will be saved to disk

SaveHighpassAmplifierWaveforms

- Whether or not highpass amplifier data will be saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - "True" – Highpass amplifier data will be saved to disk
  - "False" – Highpass amplifier data will not be saved to disk **(Default)**

SaveAuxInWithAmplifierWaveforms

- Whether or not auxiliary input waveforms will be saved in the amplifier.dat along with the wideband amplifier waveforms
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - "True" – Auxiliary input waveforms will be saved in amplifier.dat
  - "False" – Auxiliary input waveforms will not be saved in amplifier.dat **(Default)**

SaveSpikeData

- Whether or not spike data will be saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - "True" – Spike data will be saved to disk
  - "False" – Spike data will not be saved to disk **(Default)**

SaveSpikeSnapshots

- Whether or not spike snapshots will be saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - "True" – snapshots surrounding spikes will be saved to disk
  - "False" – snapshots surrounding spikes will not be saved to disk **(Default)**

SpikeSnapshotPreDetectMilliseconds

- How many milliseconds prior to a spike event are saved as a spike snapshot
- Acceptable values:

- "-3" – 3 milliseconds prior to a spike event are saved
- "-2" – 2 milliseconds prior to a spike event are saved
- "-1" – 1 millisecond prior to a spike event is saved **(Default)**
- "0" – no data prior to a spike event is saved

SpikeSnapshotPostDetectMilliseconds

- How many milliseconds after a spike event are saved as a spike snapshot
- Acceptable values:
  - "1" – 1 millisecond after a spike event is saved
  - "2" – 2 milliseconds after a spike event are saved **(Default)**
  - "3" – 3 milliseconds after a spike event are saved
  - "4" – 4 milliseconds after a spike event are saved
  - "5" – 5 milliseconds after a spike event are saved
  - "6" – 6 milliseconds after a spike event are saved

SaveDCAmplifierWaveforms

- Whether or not DC amplifier waveforms will be saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
  - "True" – DC amplifier waveforms will be saved to disk
  - "False" – DC amplifier waveforms will not be saved to disk **(Default)**

NewSaveFilePeriodMinutes

- How many minutes of recording will pass until a new save file is started
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - Integer value within the range "1" to "999", in minutes **(Default)**

Filename

- The name and location of the file(s) containing recorded data
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*

  Filename.BaseFilename

  - The base name for the file(s) containing recorded data. The final name of the file(s) will also include a date and time stamp, as well as a file extension (for example, .rhd or .rhs)
  - Acceptable values:
    - Any alphanumeric text (for example, "ExtracellularRecording2")

  Filename.Path

  - The path to the file(s) containing recorded data.
  - Acceptable values:
    - Any alphanumeric text (for example, "C:/Users/Intan/Desktop/Recordings")

TriggerSource

- Which input source (digital or analog input) is used to trigger recording if board's RunMode is Trigger (as opposed to Record, Run, or Stop)
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - For USB Interface Board, "DIGITAL-IN-XX" where XX is a 2-digit number from 00 to 15 **(Default: DIGITAL-IN-00)**
    - For USB Interface Board, "ANALOG-IN-XX" where XX is a 2-digit number from 00 to 07
    - For Recording Controller or Stimulation/Recording Controller, "DIGITAL-IN-XX" where XX is a 2-digit number from 01 to 16 **(Default: DIGITAL-IN-01)**
    - For Recording Controller or Stimulation/Recording Controller, "ANALOG-IN-X" where X is a 1-digit number from 1 to 8

TriggerPolarity

- Which polarity (high or low) is used to trigger recording if board's RunMode is Trigger (as opposed to Record, Run, or Stop). This trigger can be further configured through TriggerSource
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "High" – Recording triggers when trigger source receives a logic high **(Default)**
    - "Low" – Recording triggers when trigger source receives a logic low

TriggerAnalogThresholdVolts

- What voltage threshold must be crossed to trigger recording if board's RunMode is Trigger (as opposed to Record, Run, or Stop) and an analog input is used as a trigger source. This value is also used for triggering the PSTH tool if PSTHDigitalTrigger is an analog input.
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Decimal value within the range "-10.24" to "10.24", in Volts **(Default: 1.65)**

PreTriggerBufferSeconds

- How many seconds prior to a trigger event beginning are saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "1" to "30", in seconds **(Default: 1)**

PostTriggerBufferSeconds

- How many seconds after a trigger event ending are saved to disk
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "1" to "9999", in seconds **(Default: 1)**

TriggerSave

- Whether or not a trigger source is guaranteed to be saved, even if recording of that channel is disabled through the GUI
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - "True" – Trigger source will be saved to disk **(Default)**
  - "False" – Trigger source may not be saved to disk (still depends on if the trigger source channel is enabled in the GUI)

Note1

- The first of 3 notes that can be saved alongside recorded data
- For notes that can be quickly written and saved *while* recording, not before, see the LiveNotes section
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - Any alphanumeric string (for example, "Trial Number 24")

Note2

- The second of 3 notes that can be saved alongside recorded data
- For notes that can be quickly written and saved *while* recording, not before, see the LiveNotes section
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - Any alphanumeric string

Note3

- The last of 3 notes that can be saved alongside recorded data
- For notes that can be quickly written and saved *while* recording, not before, see the LiveNotes section
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
  - Any alphanumeric string

TCP

TCPNumberDataBlocksPerWrite

- How many data blocks (each of which contains 128 samples of data for each enabled data source) must be accrued before writing the data to its TCP output socket
- Increasing this value will introduce more delay between data acquisition and it appearing on the TCP socket, but may improve performance both for the IntanRHX software, or for whatever client software is reading the data because larger chunks of data can be processed at once.

- Decreasing this value will minimize delay between data acquisition and it appearing on the TCP socket, but processing this data in small chunks may be less efficient if performance is an issue.
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "1" to "100" **(Default: 10)**

TCPWaveformDataOutputHost

- The host address for the TCP waveform data output socket
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Any alphanumeric string **(Default: 127.0.0.1)**

TCPSpikeDataOutputHost

- The host address for the TCP spike data output socket
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Any alphanumeric string **(Default: 127.0.0.1)**

TCPWaveformDataOutputPort

- The port for the TCP waveform data output socket
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "0" to "9999" **(Default: 5001)**

TCPSpikeDataOutputPort

- The port for the TCP spike data output socket
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "0" to "9999" **(Default: 5002)**

TCPWaveformDataOutputConnectionStatus

- The connection status of the TCP socket used to output waveform data
- *"Get" command only – cannot be changed from outside the software's source code. To attempt to connect/disconnect, execute the "ConnectTCPWaveformDataOutput" or "DisconnectTCPWaveformDataOutput" command*
- Returned values:
    - "Connected" – socket is connected
    - "Pending" – socket is open and waiting for a client to connect
    - "Disconnected" – socket is disconnected

TCPSpikeDataOutputConnectionStatus

- The connection status of the TCP socket used to output spike data
- *"Get" command only – cannot be changed from outside the software's source code. To attempt to connect/disconnect, execute the "ConnectTCPSpikeDataOutput" or "DisconnectTCPSpikeDataOutput" command*
- Returned values:
    - "Connected" – socket is connected
    - "Pending" – socket is open and waiting for a client to connect
    - "Disconnected" – socket is disconnected

Processing

AvailableXPUList

- A list of available XPUs (CPUs or GPUs) that can be used to process acquired data
- This command should be used to determine the XPUIndex of each available XPU on your machine. Then, you can set which XPU to use with the "UsedXPUIndex" command
- *"Get" command only – only returns a list of available hardware from your machine. To select which XPU to use, set the "UsedXPUIndex" parameter*
- Returned values:
    - 0:[FirstXPUName]…1:[SecondXPUName]… [etc.] For example, *0:CPU...1:GeForce GTX 1660 Ti...2:Intel(R) UHD Graphics 630...*

UsedXPUIndex

- Which XPU (by index) should be used to process acquired data
- Prior to using this command, the index of each XPU should be determined by using the "AvailableXPUList" command
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "0" to "N-1" where N is the number of available XPUs on your machine

Running

RunMode

- Whether the controller is currently stopped, or if it is acquiring data, whether that data is just being displayed, saved to disk, or waiting for a trigger to start recording
- Since the same USB bus used for data acquisition when this value is not Stop is also used for UploadAmpSettleSettings, UploadChargeRecoverySettings, and UploadStimParameters, users should ensure UploadInProgress is False before executing this command if any of these uploads were recently executed. If setting this value is attempted while UploadInProgress is True, this command will be aborted and an error message will be sent to the client
- When setting this value via TCP, changes do not immediately take effect. If writing a script in a client application that sets this value before following up with additional commands, it is

recommended to pause for a fraction of a second or continually query RunMode until confirming the change has taken effect.

- Acceptable values:
  - "Run" – controller acquires data, which is only displayed graphically and not saved to disk
  - "Record" – controller acquires data, which is displayed graphically and saved to disk
  - "Trigger" – controller acquires data, which is displayed graphically, and waits for a trigger event to begin saving to disk
  - "Stop" – controller does not acquire data

CurrentTimestamp

- If the controller is running, returns the most recently read timestamp at the time this command is received. Meaningless if the controller is not running.
- Due to USB and TCP latency, this may return a value around 100 ms after this timestamp actually occurred, and should not be used for applications requiring precise timing.
- *"Get" command only*
- Returned values:
  - Positive integer value marking the number of timesteps (1 / SampleRateHertz) that have passed since acquisition began
  - -1 if this command is received while the controller is not running

CurrentTimeSeconds

- If the controller is running, returns the most recently read time in seconds at the time this command is received. Meaningless if the controller is not running.
- Due to USB and TCP latency, this may return a value around 100 ms after this time actually occurred, and should not be used for applications requiring precise timing.
- *"Get" command only*
- Returned values:
  - Poistive decimal value marking the number of seconds that have passed since acquisition began
  - -1 if this command is received while the controller is not running

Software Audio

AudioEnabled

- Whether or not the currently selected channel will have its audio played using the computer's audio
- Acceptable values:
  - "True" – Audio is enabled
  - "False" – Audio is disabled **(Default)**

AudioFilter

- Which filter band of the currently selected channel will have its audio played, as long as AudioEnabled is True
- Acceptable values:

- "Wide" – Wideband data of currently selected channel will be played
- "Low" – Lowpass data of currently selected channel will be played
- "High" – Highpass data of currently selected channel will be played

AudioVolume

- The volume level at which the currently selected channel will played, as long as AudioEnabled is True
- Acceptable values:
  - Integer value within the range "0" to "100" **(Default: 50)**

AudioThresholdMicroVolts

- The noise slicer threshold, in microVolts, that must be surpassed in order for software audio to play (using this threshold filters out low-amplitude noise, allowing audio to be clearer)
- If no thresholding is desired, this value can be set to 0
- Acceptable values:
  - Integer value within the range ±"0" to ±"200" **(Default: 0)**

Hardware Audio/Analog Out

AnalogOutGainIndex

- The index which corresponds to Electrode to Analog Out gain, which controls the amplitude of the Analog Out channels
- Acceptable values:
  - "0" – 1.6 mV/µV **(Default)**
  - "1" – 3.2 mV/µV
  - "2" – 6.4 mV/µV
  - "3" – 12.8 mV/µV
  - "4" – 25.6 mV/µV
  - "5" – 51.2 mV/µV
  - "6" – 102.4 mV/µV
  - "7" - 204.8 mV/µV

AnalogOutNoiseSlicerIndex

- The index which corresponds to the noise slicer threshold that must be surpassed in order for hardware audio to play (using this threshold filters out low-amplitude noise, allowing audio to be clearer)
- If no thresholding is desired, this value can be set to 0
- To get the threshold value in µV, multiply this index value by 3.12
- Acceptable values:
  - Integer value within the range "0" to "64" **(Default: 0)**

AnalogOutHighpassFilterEnabled

- Whether or not a highpass filter will be applied to Analog Out channels
- The cutoff frequency of this filter is specified by AnalogOutHighpassFilterFreqHertz
- Acceptable values:
  - "True" – A highpass filter will be applied to all Analog Out channels
  - "False" – A highpass filter will not be applied to all Analog Out channels

AnalogOutHighpassFilterFreqHertz

- The cutoff frequency of the highpass filter that will be applied to Analog Out channels, as long as AnalogOutHighpassFilterEnabled is True
- Acceptable values:
    - Decimal value within the range “0.01” to “9999.99”, in Hertz **(Default: 250)**

Specific Channel Analog Out

AnalogOut1LockToSelected

- Whether or not Analog Out 1 will be locked to the single channel that is currently selected in the GUI
- Acceptable values:
    - “True” – Analog Out 1 will be locked to the currently selected channel
    - “False” – Analog Out 1 will not be locked to the currently selected channel **(Default)**

AnalogOut1Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 1
- If no routing is desired, a value of “Off” will prevent Analog Out 1 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut2Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 2
- If no routing is desired, a value of “Off” will prevent Analog Out 2 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut3Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 3
- If no routing is desired, a value of “Off” will prevent Analog Out 3 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut4Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 4
- If no routing is desired, a value of “Off” will prevent Analog Out 4 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut5Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 5
- If no routing is desired, a value of “Off” will prevent Analog Out 5 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut6Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 6

- If no routing is desired, a value of “Off” will prevent Analog Out 6 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut7Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 7
- If no routing is desired, a value of “Off” will prevent Analog Out 7 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOut8Channel

- The NativeChannelName of the amplifier channel that is being routed to Analog Out 8
- If no routing is desired, a value of “Off” will prevent Analog Out 8 from being output
- Acceptable values:
    - Any alphanumeric text, for example, “A-000” **(Default: “Off”)**

AnalogOutRefChannel

- The NativeChannelName of the amplifier channel that is used as a global reference that is subtracted from all Analog Out signals in real time.
- If no external reference is desired, a value of “Hardware” will simply use the hardware reference
- Acceptable values:
    - Any alphanumeric text, for example, “A-000”) **(Default: “Hardware”)**

AnalogOut1ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 1 in order for the low-latency comparator on Digital Out 1 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut1ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut2ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 2 in order for the low-latency comparator on Digital Out 2 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut2ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut3ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 3 in order for the low-latency comparator on Digital Out 3 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut3ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut4ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 4 in order for the low-latency comparator on Digital Out 4 to trigger a digital signal

- In order for this thresholding to occur, AnalogOut4ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut5ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 5 in order for the low-latency comparator on Digital Out 5 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut5ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut6ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 6 in order for the low-latency comparator on Digital Out 6 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut6ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut7ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 7 in order for the low-latency comparator on Digital Out 7 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut7ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut8ThresholdMicroVolts

- The threshold, in MicroVolts, that must be surpassed on the signal routed to Analog Out 8 in order for the low-latency comparator on Digital Out 8 to trigger a digital signal
- In order for this thresholding to occur, AnalogOut8ThresholdEnabled must be True
- Acceptable values:
    - Integer value within the range “-6000” to “6000”, in µV **(Default: 0)**

AnalogOut1ThresholdEnabled

- Whether or not Digital Out 1 will trigger a digital signal when activity on the channel routed to Analog Out 1 surpassed AnalogOut1ThresholdMicroVolts
- Acceptable values:
    - “True” – Digital Out 1 will trigger a digital signal based on Analog Out 1 activity **(Default)**
    - “False” – Digital Out 1 will not trigger a digital signal based on Analog Out 1 activity

AnalogOut2ThresholdEnabled

- Whether or not Digital Out 2 will trigger a digital signal when activity on the channel routed to Analog Out 2 surpassed AnalogOut2ThresholdMicroVolts
- Acceptable values:
    - “True” – Digital Out 2 will trigger a digital signal based on Analog Out 2 activity **(Default)**
    - “False” – Digital Out 2 will not trigger a digital signal based on Analog Out 2 activity

AnalogOut3ThresholdEnabled

- Whether or not Digital Out 3 will trigger a digital signal when activity on the channel routed to Analog Out 3 surpassed AnalogOut3ThresholdMicroVolts
- Acceptable values:
  - "True" – Digital Out 3 will trigger a digital signal based on Analog Out 3 activity **(Default)**
  - "False" – Digital Out 3 will not trigger a digital signal based on Analog Out 3 activity

AnalogOut4ThresholdEnabled

- Whether or not Digital Out 4 will trigger a digital signal when activity on the channel routed to Analog Out 4 surpassed AnalogOut4ThresholdMicroVolts
- Acceptable values:
  - "True" – Digital Out 4 will trigger a digital signal based on Analog Out 4 activity **(Default)**
  - "False" – Digital Out 4 will not trigger a digital signal based on Analog Out 4 activity

AnalogOut5ThresholdEnabled

- Whether or not Digital Out 5 will trigger a digital signal when activity on the channel routed to Analog Out 5 surpassed AnalogOut5ThresholdMicroVolts
- Acceptable values:
  - "True" – Digital Out 5 will trigger a digital signal based on Analog Out 5 activity **(Default)**
  - "False" – Digital Out 5 will not trigger a digital signal based on Analog Out 5 activity

AnalogOut6ThresholdEnabled

- Whether or not Digital Out 6 will trigger a digital signal when activity on the channel routed to Analog Out 6 surpassed AnalogOut6ThresholdMicroVolts
- Acceptable values:
  - "True" – Digital Out 6 will trigger a digital signal based on Analog Out 6 activity **(Default)**
  - "False" – Digital Out 6 will not trigger a digital signal based on Analog Out 6 activity

AnalogOut7ThresholdEnabled

- Whether or not Digital Out 7 will trigger a digital signal when activity on the channel routed to Analog Out 7 surpassed AnalogOut7ThresholdMicroVolts
- Acceptable values:
  - "True" – Digital Out 7 will trigger a digital signal based on Analog Out 7 activity **(Default)**
  - "False" – Digital Out 7 will not trigger a digital signal based on Analog Out 7 activity

AnalogOut8ThresholdEnabled

- Whether or not Digital Out 8 will trigger a digital signal when activity on the channel routed to Analog Out 8 surpassed AnalogOut8ThresholdMicroVolts
- Acceptable values:
  - "True" – Digital Out 8 will trigger a digital signal based on Analog Out 8 activity **(Default)**
  - "False" – Digital Out 8 will not trigger a digital signal based on Analog Out 8 activity

Impedance Testing

ImpedancesHaveBeenMeasured

- Whether or not impedance values have been measured

- *"Get" command only – cannot be changed without running an impedance measurement*
- Returned values:
    - "True" – Impedance values have been measured
    - "False" – Impedance values have not been measured **(Default)**

ImpedanceFreqValid

- Whether or not the current impedance frequency is valid given restrictions based on amplifier bandwidth and sampling rate
- *"Get" command only – cannot be changed without setting DesiredImpedanceFreqHertz*
- Returned values:
    - "True" – Current impedance frequency is valid
    - "False" – Current impedance frequency is invalid **(Default)**

DesiredImpedanceFreqHertz

- The desired frequency at which the impedance measurement will run
- Restrictions based on amplifier bandwidth, sampling rate, and discretization keep certain frequencies from being achievable, so the user should set this value, which updates ActualImpedanceFreqHertz to the closest possible match
- Acceptable values:
    - Decimal value between "0" and "7500", in Hz **(Default: 1000)**

ActualImpedanceFreqHertz

- The actual frequency at which the impedance measurement will run
- Restrictions based on amplifier bandwidth, sampling rate, and discretization keep certain frequencies from being achievable, so the user should set DesiredImpedanceFreqHertz, which updates this value to the closest possible match
- *"Get" command only – cannot be changed without setting DesiredImpedanceFreqHertz*
- Returned values:
    - Decimal value between "0" and "7500", in Hz **(Default: 1200)**

ImpedanceFilename

- The name and location of the file containing measured impedance data

  ImpedanceFilename.BaseFilename

    - The base name for the file containing measured impedance data. The final name of the file will also include a file extension (for example, .csv)
    - Acceptable values:
        - Any alphanumeric text (for example, "ImpedanceMeasurement1kHz")

  ImpedanceFilename.Path

    - The path to the file(s) containing recorded data.
    - Acceptable values:
        - Any alphanumeric text (for example, "C:/Users/Intan/Desktop/Impedance")

Referencing

UseMedianReference

- When setting a reference as an average of multiple of channels, whether a median is used instead of average for all reference calculations
- Depending on the application, may be more accurate than average, for example if a few channels are significant outliers
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – References calculated from multiple channels will use the median.
    - "False" – References calculated from multiple channels will use the average. **(Default)**

Filtering

DSPEnabled

- Whether or not the On-Chip DSP Offset Removal Filter is enabled
- The cutoff frequency of this filter is based on ActualDSPCutoffFreqHertz
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – On-Chip DSP Offset Removal Filter is enabled
    - "False" – On-Chip DSP Offset Removal Filter is disabled

DesiredDSPCutoffFreqHertz

- The desired cutoff frequency for the On-Chip DSP Offset Removal Filter. This value, along with DesiredLowerBandwidthHertz, can combine effects to both impact DesiredLower3dBCutoffFreqHertz.
- Restrictions based on sampling rate and discretization keep certain frequencies from being achievable, so the user should set this value, which updates ActualDSPCutoffFreqHertz to the closest possible match
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Decimal value between "0" and "4000", in Hz **(Default: 1)**

ActualDSPCutoffFreqHertz

- The actual cutoff frequency for the On-Chip DSP Offset Removal Filter. This value, along with ActualLowerBandwidthHertz, can combine effects to both impact ActualLower3dBCutoffFreqHertz
- Restrictions based on sampling rate and discretization keep certain frequencies from being achievable, so the user should set DesiredDSPCutoffFreqHertz, which updates this value to the closest possible match

- *"Get" command only – cannot be changed without setting DesiredDSPCutoffFreqHertz or DesiredLower3dBCutoffFreqHertz*
- Returned values:
    - Decimal value between "0" and "8000", in Hz **(Default: 1)**

DesiredLowerBandwidthHertz

- The desired lower bandwidth for the on-chip amplifiers. This value, along with DesiredDSPCutoffFreqHertz, can combine effects to both impact DesiredLower3dBCutoffFreqHertz
- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set this value, which updates ActualLowerBandwidthHertz to the closest possible match
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Decimal value between "0.1" and "500", in Hz **(Default: 0.1)**

ActualLowerBandwidthHertz

- The actual lower bandwidth for the on-chip amplifiers. This value, along with ActualDSPCutoffFreqHertz, can combine effects to both impact ActualLower3dBCutoffFreqHertz
- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set DesiredLowerBandwidthHertz, which updates this value to the closest possible match
- *"Get" command only – cannot be changed without setting DesiredLowerBandwidthHertz or DesiredLower3dBCutoffFreqHertz*
- Returned values:
    - Decimal value between "0" and "1000", in Hz **(Default: 0.1)**

DesiredLower3dBCutoffFreqHertz

- The desired lower 3dB cutoff frequency for the on-chip amplifiers, taking into account both the on-chip DSP filter and lower amplifier bandwidth. This value is impacted by both DesiredDSPCutoffFreqHertz and DesiredLowerBandwidthHertz, and setting this value with impact both of these variables in turn
- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set this value, which updates ActualLower3dBCutoffFreqHertz to the closest possible match
- *"Get" command only – cannot be changed outside of the GUI. This is intended as a GUI convenience function, and the user can instead set DesiredLowerBandwidthHertz and DesiredDSPCutoffFreqHertz*
- Returned values:
    - Decimal value between "0.1" and "500", in Hz **(Default: 1)**

ActualLower3dBCutoffFreqHertz

- The actual lower 3dB cutoff frequency for the on-chip amplifiers, taking into account both the on-chip DSP filter and lower amplifier bandwidth. This value is impacted by both ActualDSPCutoffFreqHertz and ActualLowerBandwidthHertz, and changes to this value with impact both of these variables in turn

- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set DesiredLower3dBCutoffFreqHertz, which updates this value to the closest possible match
- *"Get" command only – cannot be changed outside of the GUI. This is intended as a GUI convenience function, and the user can instead set DesiredLowerBandwidthHertz and DesiredDSPCutoffFreqHertz*
- Returned values:
    - Decimal value between "0.05" and "1000", in Hz **(Default: 1)**

DesiredUpperBandwidthHertz

- The desired upper bandwidth for the on-chip amplifiers
- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set this value, which updates ActualUpperBandwidthHertz to the closest possible match
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Decimal value between "100" and "20000", in Hz **(Default: 7500)**

ActualUpperBandwidthHertz

- The actual upper bandwidth for the on-chip amplifiers
- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set DesiredUpperBandwidthHertz, which updates this value to the closest possible match
- *"Get" command only – cannot be changed without setting DesiredUpperBandwidthHertz*
- Returned values:
    - Decimal value between "50" and "25000", in Hz **(Default: 7500)**

NotchFilterFreqHertz

- The center frequency for the notch filter, which is designed to remove noise at 50 or 60 Hz.
- The notch filter can be disabled by setting this value to "None"
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - "None" – Notch filter is disabled
    - "50" – Notch filter will attenuate 50 Hz signals from incoming amplifier data
    - "60" – Notch filter will attenuate 60 Hz signals from incoming amplifier data

LowpassFilterOrder

- The order of the lowpass filter, which can be further configured through LowpassFilterType and LowpassFilterCutoffFreqHertz
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "1" to "8" **(Default: 2)**

LowpassFilterType

- The type of the lowpass filter, which can be further configured through LowpassFilterOrder and LowpassFilterCutoffFreqHertz
- The two types of filters implemented are Bessel and Butterworth
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - "Bessel" – Lowpass filter will be a Bessel type filter
    - "Butterworth" – Lowpass filter will be a Butterworth type filter

LowpassFilterCutoffFreqHertz

- The cutoff frequency of the lowpass filter, which can be further configured through LowpassFilterOrder and LowpassFilterType
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - Decimal value within the range "1" to "5000" **(Default: 250)**

HighpassFilterOrder

- The order of the highpass filter, which can be further configured through HighpassFilterType and HighpassFilterCutoffFreqHertz
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "1" to "8" **(Default: 2)**

HighpassFilterType

- The type of the highpass filter, which can be further configured through HighpassFilterOrder and HighpassFilterCutoffFreqHertz
- The two types of filters implemented are Bessel and Butterworth
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - "Bessel" – Highpass filter will be a Bessel type filter
    - "Butterworth" – Highpass filter will be a Butterworth type filter

HighpassFilterCutoffFreqHertz

- The cutoff frequency of the highpass filter, which can be further configured through HighpassFilterOrder and HighpassFilterType
- *"Set" has no effect when board is recording. To change this parameter, stop the board from recording (for example, set RunMode to Stop)*
- Acceptable values:
    - Decimal value within the range "1" to "5000" **(Default: 250)**

Display options

FilterDisplay1

- Which filter band of each amplifier channel is displayed in position 1 of the waveform plot
- Acceptable values:
    - "None" – Position 1 will not display any filter band
    - "Wide" – Position 1 will display wideband waveforms **(Default)**
    - "Low" – Position 1 will display lowpass waveforms
    - "High" – Position 1 will display highpass waveforms
    - "Spk" – Position 1 will display spike rasters
    - "Dc" – Position 1 will display DC amplifier waveforms (Stimulation/Recording Controller only)

FilterDisplay2

- Which filter band of each amplifier channel is displayed in position 2 of the waveform plot
- Acceptable values:
    - "None" – Position 2 will not display any filter band **(Default)**
    - "Wide" – Position 2 will display wideband waveforms
    - "Low" – Position 2 will display lowpass waveforms
    - "High" – Position 2 will display highpass waveforms
    - "Spk" – Position 2 will display spike rasters
    - "Dc" – Position 2 will display DC amplifier waveforms (Stimulation/Recording Controller only)

FilterDisplay3

- Which filter band of each amplifier channel is displayed in position 3 of the waveform plot
- Acceptable values:
    - "None" – Position 3 will not display any filter band **(Default)**
    - "Wide" – Position 3 will display wideband waveforms
    - "Low" – Position 3 will display lowpass waveforms
    - "High" – Position 3 will display highpass waveforms
    - "Spk" – Position 3 will display spike rasters
    - "Dc" – Position 3 will display DC amplifier waveforms (Stimulation/Recording Controller only)

FilterDisplay4

- Which filter band of each amplifier channel is displayed in position 4 of the waveform plot
- Acceptable values:
    - "None" – Position 4 will not display any filter band **(Default)**
    - "Wide" – Position 4 will display wideband waveforms
    - "Low" – Position 4 will display lowpass waveforms
    - "High" – Position 4 will display highpass waveforms
    - "Spk" – Position 4 will display spike rasters
    - "Dc" – Position 4 will display DC amplifier waveforms (Stimulation/Recording Controller only)

ArrangeBy

- Whether the default listing of the waveforms in the waveform plot is arranged in order by channel or by filter
- Acceptable values:
  - "Channel" – Each channel will be listed in order, with all the filter bands of a channel being displayed before the next channel **(Default)**
  - "Filter" – Each channel will be listed in order, with the first filter band of all channels being displayed before the next filter band

ShowDisabledChannels

- Whether or not Disabled channels, which will not be saved to disk, are displayed on the waveform plot
- Acceptable values:
  - "True" – Disabled channels will be shown **(Default)**
  - "False" – Disabled channels will not be shown

ShowAuxInputs

- Whether or not on-chip auxiliary inputs are displayed on the waveform plot
- *Only applies to USB Interface Boards and Recording Controllers*
- Acceptable values:
  - "True" – auxiliary inputs will be shown **(Default)**
  - "False" – auxiliary inputs will not be shown

ShowSupplyVoltages

- Whether or not on-chip supply voltage measurements are displayed on the waveform plot
- *Only applies to USB Interface Boards and Recording Controllers*
- Acceptable values:
  - "True" – supply voltage measurements will be shown
  - "False" – supply voltage measurements will not be shown **(Default)**

BackgroundColor

- Which color will be used for the background of the waveform plot
- Acceptable values:
  - RGB values in the following format: "#RRGGBB" where each digit is a hexadecimal digit. For example, red would be #FF0000" **(Default: #000000, which is black)**

PlottingMode

- Which method will be used when plotting waveforms
- For large resolution monitors, the Original method of plotting may be too slow to keep up in real-time. The High Efficiency plotting mode significantly improves performance by only plotting the most recently acquired sections of the waveforms. This only works when RollMode is False.
- Acceptable values:
  - "Original" – data will be plotted with the original method that RHX has always used
  - "High Efficiency" – data will be plotted with a newer method that prioritizes speed **(Default)**

Playback options

RunAfterJumpToPosition

- Whether or not playback starts immediately after jumping to a position
- *Only applies when software is started in Playback mode, having recorded previously saved data*
- Acceptable values:
    - "True" – playback starts immediately after jumping to a position
    - "False" – playback does not start automatically after jumping to a position **(Default)**

Display trigger

DisplayTriggerMode

- Whether or not an oscilloscope-style trigger will update the display of the waveform plot. This trigger can be further configured through DisplayDigitalTrigger, DisplayTriggerPolarity, and DisplayTriggerPosition
- Acceptable values:
    - "True" – The display will only update when a trigger event occurs
    - "False" – The display will not rely on trigger events to update **(Default)**

DisplayDigitalTrigger

- Which input source (digital or thresholded analog input) is used to trigger the oscilloscope-style update of the display of the waveform plot if DisplayTriggerMode is True. This trigger can be further configured through DisplayTriggerPolarity and DisplayTriggerPosition
- Acceptable values:
    - For USB Interface Board, "DIGITAL-IN-XX" where XX is a 2-digit number from 00 to 15 **(Default: DIGITAL-IN-00)**
    - For USB Interface Board, "ANALOG-IN-XX' where XX is a 2-digit number from 00 to 07
    - For Recording Controller or Stimulation/Recording Controller, "DIGITAL-IN-XX" where XX is a 2-digit number from 01 to 16 **(Default: DIGITAL-IN-01)**
    - For Recording Controller or Stimulation/Recording Controller, "ANALOG-IN-X" where X is a 1-digit number from 1 to 8

DisplayTriggerPolarity

- Which polarity (high or low) is used to trigger the oscilloscope-style update of the display of the waveform plot if DisplayTriggerMode is True. This trigger can be further configured through DisplayDigitalTrigger and DisplayTriggerPosition
- Acceptable values:
    - "Rising" - Recording triggers when trigger source encounters a rising edge **(Default)**
    - "Falling" – Recording triggers when trigger source encounters a falling edge

DisplayTriggerPosition

- Which position the event used to trigger the oscilloscope-style update of the display of the waveform plot if DisplayTriggerMode is True will occupy. This trigger can be further configured through DisplayDigitalTrigger and DisplayTriggerPolarity
- Acceptable values:
    - "1/10" – Trigger event will be placed 1/10 along the time axis of the waveform plot

- "1/4" – Trigger event will be placed 1/4 along the time axis of the waveform plot **(Default)**
- "1/2" – Trigger event will be placed 1/2 along the time axis of the waveform plot
- "3/4" – Trigger event will be placed 3/4 along the time axis of the waveform plot
- "9/10" – Trigger event will be placed 9/10 along the time axis of the waveform plot

Plotting waveforms

RollMode

- Whether or not the display will be shown as a rolling plot with incoming data immediately being shown at the right-hand side of the display
- Acceptable values:
  - "True" – Display will plot in a rolling fashion, with data continuously coming in on the right
  - "False" – Display will plot in a sweeping fashion, with data occupying the same position until the left-to-right sweep overwrites it **(Default)**

ClipWaveforms

- Whether or not the display will clip waveforms that exceed their vertical scale boundaries
- This is a purely visual function, and will not impact the data saved to disk
- Acceptable values:
  - "True" – waveforms will be clipped when they exceed their vertical scale boundaries
  - "False" – waveforms will not be clipped **(Default)**

TimeScaleMilliseconds

- How many milliseconds the full horizontal length of the waveform plot represents
- Acceptable values:
  - "10" ms
  - "20" ms
  - "40" ms
  - "100" ms
  - "200" ms
  - "400" ms
  - "1000" ms
  - "2000" ms **(Default)**
  - "4000" ms

LowScaleMicroVolts

- How many microVolts the full vertical length of a single amplifier channel's lowpass plot represents
- Acceptable values:
  - "50" µV
  - "100" µV
  - "200" µV
  - "500" µV **(Default)**

- “1000” µV
- “2000” µV
- “5000” µV

WideScaleMicroVolts

- How many microVolts the full vertical length of a single amplifier channel’s wideband plot represents
- Acceptable values:
  - “50” µV
  - “100” µV
  - “200” µV
  - “500” µV **(Default)**
  - “1000” µV
  - “2000” µV
  - “5000” µV

HighScaleMicroVolts

- How many microVolts the full vertical length of a single amplifier channel’s highpass plot represents
- Acceptable values:
  - “50” µV
  - “100” µV
  - “200” µV
  - “500” µV **(Default)**
  - “1000” µV
  - “2000” µV
  - “5000” µV

AuxScaleVolts

- How many Volts the full vertical length of an auxiliary input’s plot represents
- *Only applies to USB Interface Boards and Recording Controllers*
- Acceptable values:
  - “0.1” V
  - “0.2” V
  - “0.5” V
  - “1.0” V **(Default)**
  - “2.0” V

AnalogScaleVolts

- How many Volts the full vertical length of a controller’s analog input’s plot represents
- Acceptable values:
  - “0.1” V
  - “0.2” V
  - “0.5” V
  - “1.0” V **(Default)**
  - “2.0” V

- "5.0" V
- "10.0" V

DCScaleVolts

- How many Volts the full vertical length of an amplifier's DC plot represents
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
  - "0.5" V
  - "1.0" V **(Default)**
  - "2.0" V
  - "5.0" V
  - "10.0" V

DisplayLabelText

- Which value is displayed as the label for a waveform plot
- Acceptable values:
  - "CustomName" – The CustomChannelName the user gave this channel. If no custom name exists, then the NativeChannelName is used **(Default)**
  - "NativeName" – The NativeChannelName assigned to this channel
  - "ImpedanceMagnitude" – The magnitude of measured impedance for this channel
  - "ImpedancePhase" – The phase angle of measured impedance for this channel
  - "Reference" – The reference used for this channel

LabelWidth

- How wide the label for a waveform plot will be
- Acceptable values:
  - "Hide" – The label for a waveform plot will not be displayed
  - "Narrow" – The label for a waveform plot will be narrow
  - "Wide" – The label for a waveform plot will be wide **(Default)**

ISI (Inter-Spike Interval Histogram)

ISIChannel

- The NativeChannelName of the amplifier channel that is being displayed in the ISI plot
- Acceptable values:
  - Any alphanumeric text, for example, "A-000" **(Default: whichever channel is selected when ISI window is first opened)**

ISITimeSpanMilliseconds

- The timespan, in milliseconds, used to plot ISI data
- Acceptable values:
  - "50" ms
  - "100" ms
  - "200" ms **(Default)**
  - "500" ms

- “1000” ms

ISIBinSizeMilliseconds

- The size, in milliseconds, of the bins used to plot ISI data
- Acceptable values:
  - “1” ms
  - “2” ms
  - “5” ms **(Default)**
  - “10” ms
  - “20” ms

ISIYAxisLog

- Whether or not the y axis of the ISI plot will be displayed logarithmically
- Acceptable values:
  - “True” – y axis of ISI plot is logarithmic
  - “False” – y axis of ISI plot is linear

ISISaveCsvFile

- Whether or not a CSV Text File is created when ISI data is saved
- Acceptable values:
  - “True” – CSV Text File will be created when ISI data is saved
  - “False” – CSV Text File will not be created when ISI data is saved

ISISaveMatFile

- Whether or not a MAT Data File (used by MATLAB) is created when ISI data is saved
- Acceptable values:
  - “True” – MAT Data File will be created when ISI data is saved
  - “False” – MAT Data File will not be created when ISI data is saved

ISISavePngFile

- Whether or not a PNG Screen Capture File is created when ISI data is saved
- Acceptable values:
  - “True” – PNG File will be created when ISI data is saved
  - “False” – PNG File will not be created when ISI data is saved

PSTH (Peri-Stimulus Time Histogram)

PSTHChannel

- The NativeChannelName of the amplifier channel that is being displayed in the PSTH plot
- Acceptable values:
  - Any alphanumeric text, for example, “A-000” **(Default: whichever channel is selected when PSTH window is first opened)**

PSTHPreTriggerSpanMilliseconds

- The amount of time, in milliseconds, before the trigger event that data is included in the PSTH
- Acceptable values:

- "50" ms
- "100" ms
- "200" ms
- "500" ms **(Default)**
- "1000" ms
- "2000" ms

PSTHPostTriggerSpanMilliseconds

- The amount of time, in milliseconds, after the trigger event that data is included in the PSTH
- Acceptable values:
  - "50" ms
  - "100" ms
  - "200" ms
  - "500" ms **(Default)**
  - "1000" ms
  - "2000" ms
  - "5000" ms
  - "10000" ms
  - "20000" ms

PSTHBinSizeMilliseconds

- The size, in milliseconds, of the bins used to plot PSTH data
- Acceptable values:
  - "1" ms
  - "2" ms
  - "5" ms **(Default)**
  - "10" ms
  - "20" ms
  - "50" ms
  - '100" ms

PSTHMaxNumTrials

- The maximum number of trigger events (trials) for which to gather PSTH data before older data is overwritten
- Acceptable values:
  - "10" – 10 trials will occur before the earliest is overwritten
  - "20" – 20 trials will occur before the earliest is overwritten
  - "50" – 50 trials will occur before the earliest is overwritten **(Default)**
  - "100" – 100 trials will occur before the earliest is overwritten
  - "200" – 200 trials will occur before the earliest is overwritten
  - "500" – 500 trials will occur before the earliest is overwritten

PSTHDigitalTrigger

- Which input source (digital or thresholded analog input) is used to trigger the PSTH data collection. This trigger can be further configured through PSTHTriggerPolarity
- Acceptable values:

- For USB Interface Board, “DIGITAL-IN-XX” where XX is a 2-digit number from 00 to 15 **(Default: DIGITAL-IN-00)**
- For USB Interface Board, “ANALOG-IN-XX” where XX is a 2-digit number from 00 to 07
- For Recording Controller or Stimulation/Recording Controller, “DIGITAL-IN-XX” where XX is a 2-digit number from 01 to 16 **(Default: DIGITAL-IN-01)**
- For Recording Controller or Stimulation/Recording Controller, “ANALOG-IN-X” where X is a 1-digit number from 1 to 8

PSTHTriggerPolarity

- Which polarity (high or low) is used to trigger the PSTH data collection. This trigger can be further configured through PSTHDigitalTrigger
- Acceptable values:
  - “Rising” – PSTH triggers when trigger source encounters a rising edge **(Default)**
  - “Falling” – PSTH triggers when trigger source encounters a falling edge

PSTHSaveCsvFile

- Whether or not a CSV Text File is created when PSTH data is saved
- Acceptable values:
  - “True” – CSV Text File will be created when PSTH data is saved
  - “False” – CSV Text File will not be created when PSTH data is saved

PSTHSaveMatFile

- Whether or not a MAT Data File (used by MATLAB) is created when PSTH data is saved
- Acceptable values:
  - “True” – MAT Data File will be created when PSTH data is saved
  - “False” – MAT Data File will not be created when PSTH data is saved

PSTHSavePngFile

- Whether or not a PNG Screen Capture File is created when PSTH data is saved
- Acceptable values:
  - “True” – PNG File will be created when PSTH data is saved
  - “False” – PNG File will not be created when PSTH data is saved

Spectrogram

SpectrogramChannel

- The NativeChannelName of the amplifier channel that is being displayed in the Spectrogram plot
- Acceptable values:
  - Any alphanumeric text, for example, “A-000” **(Default: whichever channel is selected when Spectrogram window is first opened)**

SpectrogramDisplayMode

- Whether the Spectrogram will be displayed in Spectrogram or Spectrum mode
- Acceptable values:
  - “Spectrogram” **(Default)**
  - “Spectrum”

SpectrogramTimeScaleSeconds

- When in Spectrogram mode, how many milliseconds the full horizontal length of the spectrogram represents
- Acceptable values:
  - "2" s
  - "5" s **(Default)**
  - "10" s

SpectrogramFFTSize

- How many samples are used in each iteration of the FFT algorithm
- There is an inherent tradeoff between time resolution and frequency resolution when running the FFT algorithm. Using a low number of samples will result in more precise time resolution, but less precise frequency resolution. Using a high number of samples will result in less precise time resolution, but more precise frequency resolution
- Acceptable values:
  - "256"
  - "512"
  - "1024"
  - "2048" **(Default)**
  - "4096"
  - "8192"
  - "16384"

SpectrogramFreqMinHertz

- Minimum frequency to be displayed in Spectrogram, in Hertz
- Acceptable values:
  - Any integer value within the range "0" to "fNyquist – 10" where fNyquist is ½ the amplifier sample rate, in Hz **(Default: 0)**

SpectrogramFreqMaxHertz

- Maximum frequency to be displayed in Spectrogram, in Hertz
- Acceptable values:
  - Any integer value within the range "10" to "fNyquist" where fNyquist is ½ the amplifier sample rate, in Hz **(Default: either fNyquist or 200, whichever is lower)**

SpectrogramFreqMarkerHertz

- Which frequency is shown as a marker in the Spectrogram, in Hertz. Whether a marker exists is controlled by SpectrogramShowFreqMarker
- Acceptable values:
  - Any integer value within the range "0" to "fNyquist" where fNyquist is ½ the amplifier sample rate, in Hz **(Default: 60)**

SpectrogramShowFreqMarker

- Whether or not the frequency specified in SpectrogramFreqMarkerHertz is displayed as a marker in the Spectrogram
- Acceptable values:

- "True" – Marker is shown **(Default)**
- "False" – Marker is not shown

SpectrogramFreqMarkerNumHarmonics

- How many harmonics of the marker frequency specified in SpectrogramFreqMarkerHertz are displayed
- Acceptable values:
  - Any integer value within the range "0" to "9" **(Default: 0)**

SpectrogramDigitalDisplay

- Which input signal (digital or thresholded analog input) is displayed at the top of the Spectrogram window
- Acceptable values:
  - "None" – no digital input signal is displayed
  - For USB Interface Board, "DIGITAL-IN-XX" where XX is a 2-digit number from 00 to 15 **(Default: DIGITAL-IN-00)**
  - For USB Interface Board, "ANALOG-IN-XX" where XX is a 2-digit number from 00 to 07
  - For Recording Controller or Stimulation/Recording Controller, "DIGITAL-IN-XX' where XX is a 2-digit number from 01 to 16 **(Default: DIGITAL-IN-01)**
  - For Recording Controller or Stimulation/Recording Controller, "ANALOG-IN-X" where X is a 1-digit number from 1 to 8

SpectrogramSaveCsvFile

- Whether or not a CSV Text File is created when Spectrogram data is saved
- Acceptable values:
  - "True" – CSV Text File will be created when Spectrogram data is saved
  - "False" – CSV Text File will not be created when Spectrogram data is saved

SpectrogramSaveMatFile

- Whether or not a MAT Data File (used by MATLAB) is created when Spectrogram data is saved
- Acceptable values:
  - "True" – MAT Data File will be created when Spectrogram data is saved
  - "False" – MAT Data File will not be created when Spectrogram data is saved

SpectrogramSavePngFile

- Whether or not a PNG Screen Capture File is created when Spectrogram data is saved
- Acceptable values:
  - "True" – PNG File will be created when Spectrogram data is saved
  - "False" – PNG File will not be created when Spectrogram data is saved

Spike Scope

SpikeScopeChannel

- The NativeChannelName of the amplifier channel that is being displayed in the Spike Scope
- Acceptable values:

- Any alphanumeric text, for example, “A-000” **(Default: whichever channel is selected when Spike Scope window is first opened)**

SpikeScopeScaleMicroVolts

- How many microVolts the full vertical length of the Spike Scope display represents
- Acceptable values:
  - “50” µV
  - “100” µV
  - “200” µV
  - “500” µV **(Default)**
  - “1000” µV
  - “2000” µV
  - “5000” µV

SpikeScopeTimeScaleMilliseconds

- How many milliseconds the full horizontal length of the Spike Scope display represents
- Acceptable values:
  - “2” ms **(Default)**
  - “4” ms
  - “6” ms

SpikeScopeNumSpikes

- How many spikes will be displayed in the Spike Scope before older spikes are overwritten
- Acceptable values:
  - “10”
  - “20” **(Default)**
  - “30”
  - “50”
  - “100”
  - “200”
  - “500”

ArtifactSuppressionEnabled

- Whether or not artifacts are suppressed, preventing them from being classified as spikes when data is first processed
- Artifacts are voltage spikes that surpass ArtifactSuppressionThresholdMicroVolts, and are intended to represent signal distortion from non-neural events like movement that technically surpass a spike threshold
- Acceptable values:
  - “True” – Events classified as artifacts will be suppressed, and will not be classified as spikes
  - “False” – Events classified as artifacts will not be suppressed, and will be classified as spikes

ArtifactsShown

- Whether or not artifacts that have been detected (only possible in ArtifactSuppressionEnabled is True) are displayed in a unique color in the Spike Scope
- Acceptable values:
    - "True" – Artifacts that have been detected are shown in the Spike Scope
    - "False" – Artifacts that have been detected are not shown in the Spike Scope

ArtifactSuppressionThresholdMicroVolts

- What threshold (in microVolts) must be crossed in order for a spike event to be classified as an artifact if ArtifactSuppressionEnabled is True
- Acceptable values:
    - Integer value within the range "0" to "5000", in µV **(Default: 300)**

Fast Amplifier Settling

ManualFastSettleEnabled

- Whether or not the manual amplifier fast settle functionality of the amplifiers is enabled
- *Only applies to USB Interface Boards and Recording Controllers*
- Acceptable values:
    - "True" – Amplifier fast settling will be manually enabled
    - "False" – Amplifier fast settling will not be manually enabled **(Default)**

ExternalFastSettleEnabled

- Whether or not the external amplifier fast settle functionality of the amplifiers is enabled
- The digital input channel used to trigger external fast settle is specified through ExternalFastSettleChannel
- *Only applies to USB Interface Boards and Recording Controllers*
- Acceptable values:
    - "True" – Amplifier fast settling will be externally triggered
    - "False" – Amplifier fast settling will not be externally triggered

ExternalFastSettleChannel

- Which digital input channel is used to trigger external amplifier fast settling
- *Only applies to USB Interface Boards and Recording Controllers*
- Acceptable values:
    - Integer value within the range "1" to "16". Note: For USB Interface Boards, the Digital Input with value 1 is DIGITAL-IN-00, whereas for Recording Controllers, the Digital Input with value 1 is DIGITAL-IN-01. **(Default: 1)**

Stimulation

DesiredLowerSettleBandwidthHertz

- The desired frequency, in Hertz, the amplifier lower bandwidth will temporarily be toggled to during an amp settle event if UseFastSettle is False

- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set this value, which updates ActualLowerSettleBandwidthHertz to the closest possible match
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadAmpSettleSettings
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
    - Decimal value within the range "0.1" to "1000", in Hz **(Default: 1000)**

ActualLowerSettleBandwidthHertz

- The actual frequency, in Hertz, the amplifier lower bandwidth will temporarily be toggled to during an amp settle event if UseFastSettle is False
- Restrictions based on discretization keep certain frequencies from being achievable, so the user should set DesiredLowerSettleBandwidthHertz, which updates this value to the closest possible match
- *"Get" command only – cannot be changed without setting DesiredLowerSettleBandwidthHertz*
- *Only applies to Stimulation/Recording Controllers*
- Returned values:
    - Decimal value within the range "0.05" to "1000", in Hz **(Default: 1)**

UseFastSettle

- Whether or not an amp settle event will trigger a traditional fast settle. If not, then the amplifier's lower bandwidth frequency will be temporarily toggled to a higher value of ActualLowerSettleBandwidthHertz to attenuate stimulation artifacts
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadAmpSettleSettings
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
    - "True" – Traditional fast settle will be used when an amp settle event occurs
    - "False" – Lower amplifier bandwidth will be changed when an amp settle event occurs **(Default)**

HeadstageGlobalSettle

- Whether or not an amp settle event will be triggered globally (across all amplifier channels)
- When this is enabled, an amp settle event on one channel will also be triggered for every other channel on that headstage
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadAmpSettleSettings

- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
    - "True" – Amp settle events will be triggered globally
    - "False" – Amp settle events will not be triggered globally **(Default)**

ChargeRecovery

- Whether or not the charge recovery switch is used during a charge recovery event. If not, then the Current-Limited Charge Recovery Circuit will be used, which can be further configured through ChargeRecoveryCurrentLimitMicroAmps and ChargeRecoveryTargetVoltage
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadChargeRecoverySettings
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
    - "True" – Charge recovery switch will be used during a charge recovery event
    - "False" – Current-Limited Charge Recovery Circuit will be used during a charge recovery event **(Default)**

ChargeRecoveryCurrentLimitMicroAmps

- The current limit, in microAmps, of the charge recovery circuit that is used during a charge recovery event, if ChargeRecovery is False. The charge recovery circuit can be further configured through ChargeRecoveryTargetVoltage
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadChargeRecoverySettings
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
    - "Min"
    - "0.001" µA
    - "0.002" µA
    - "0.005" µA
    - "0.01" µA **(Default)**
    - "0.02" µA
    - "0.05" µA
    - "0.1" µA
    - "0.2" µA
    - "0.5" µA
    - "1" µA

ChargeRecoveryTargetVoltageVolts

- The target voltage, in Volts, of the charge recovery circuit that is used during a charge recovery event, if ChargeRecovery is False. The charge recovery circuit can be further configured through ChargeRecoveryCurrentLimitMicroAmps
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadChargeRecoverySettings
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- *Only applies to Stimulation/Recording Controllers*
- Acceptable values:
    - Decimal value within the range "-1.225" to "1.215", in Volts **(Default: 0)**

Global Spike Detection Thresholds

AbsoluteThresholdsEnabled

- Whether or not an absolute threshold is globally set to all channels
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect, the parameters must be confirmed by Executing the command SetSpikeDetectionThresholds
- If this value is True, future calls to SetSpikeDetectionThresholds will set all spike detection thresholds to the value specified by AbsoluteThresholdMicroVolts
- If this value is False, future calls to SetSpikeDetectionThresholds will set all spike detection thresholds a channel-relative value which has a magnitude set by RmsMultipleThreshold and a polarity set by NegativeRelativeThreshold
- Acceptable values:
    - "True" – Global spike detection threshold setting will be absolute across all enabled channels
    - "False" – Global spike detection threshold setting will be relative for all enabled channels

AbsoluteThresholdMicroVolts

- The threshold voltage, in MicroVolts, to which spike detection will be set globally across all enabled channels
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect, the parameters must be confirmed by Executing the command SetSpikeDetectionThresholds
- This value will only be used if AbsoluteThresholdsEnabled is True
- Acceptable values:
    - Integer value within the range "-5000" to "5000", in MicroVolts **(Default: -70)**

RmsMultipleThreshold

- The factor by which a channel's RMS noise level will be multiplied to determine its relative spike detection threshold

- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect, the parameters must be confirmed by Executing the command setSpikeDetectionThresholds
- This value will only be used if AbsoluteThresholdsEnabled is False
- Acceptable values:
    - Decimal value within the range “3.0” to “20.0” **(Default: 4.0)**

NegativeRelativeThreshold

- Whether or not the polarity of the relative threshold to be set globally across all enabled channels is negative (the magnitude is determined by RmsMultipleThreshold).
- Setting this parameter updates its value in the IntanRHX software, but in order for the changes to take effect, the parameters must be confirmed by Executing the command SetSpikeDetectionThresholds
- This value will only be used if AbsoluteThresholdsEnabled is False
- Acceptable values:
    - “True” – The relative spike detection threshold for each channel will be negative **(Default)**
    - “False” – The relative spike detection threshold for each channel will be positive

Port Manipulation

A port’s name can be used in conjunction with a “port parameter” in order to conduct get and sets operations on an individual port. The port parameter should be included immediately after a period following the port name, much like accessing a member within a struct in the C programming language.

Example 1. To enable all analog inputs: *set analog-in.recordingenabled true*

Example 2. To determine the number of connected headstage channels on port B: *get b.channelspresent*

Prefix

- The prefix of each channel’s name. For example, amplifier channel A-012 has prefix “A”. Analog input channel ANALOG-IN-3 has prefix “ANALOG-IN”
- *“Get” command only – cannot be changed outside of source code*
- Returned values:
    - “A” – Port A
    - “B” – Port B
    - “C” – Port C
    - “D” – Port D
    - “E” – Port E (only applies to 1024-channel Recording Controllers)
    - “F” – Port F (only applies to 1024-channel Recording Controllers)
    - “G” – Port G (only applies to 1024-channel Recording Controllers)
    - “H” – Port H (only applies to 1024-channel Recording Controllers)
    - “ANALOG-IN”
    - “ANALOG-OUT”
    - “DIGITAL-IN”
    - “DIGITAL-OUT”

ManualDelayEnabled

- Whether or not a user-specified SPI delay is specified for this port
- *Only applies to amplifier ports A-H*
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – A user-specified SPI delay, ManualDelay, is used for this port
    - "False" – The automatically calculated default SPI delay is used for this port **(Default)**

ManualDelay

- The amount of delay expected across the SPI cable connecting this port to its headstage(s)
- This value has no units, and is just an index that can range from 0 to 15. The lower the index value is, the less delay is expected
- *Only applies to amplifier ports A-H*
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "0" to "15" **(Default: 1)**

AuxDigOutEnabled

- Whether or not the chip(s) connected to this port has its auxiliary digital output pin enabled. When this functionality is enabled, the output pin can be controlled from a digital input channel (AuxDigOutChannel) with a latency of 4-5 amplifier sampling periods
- *Only applies to amplifier ports A-H*
- *Only applies to USB Interface Boards and Recording Controller Boards*
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - "True" – Auxiliary digital output is enabled to trigger off AuxDigOutChannel
    - "False" – Auxiliary digital output is not enabled **(Default)**

AuxDigOutChannel

- Which digital input channel is used to trigger the auxiliary digital output pin of the chip(s) connected to this port, if AuxDigOutEnabled is True
- *Only applies to amplifier ports A-H*
- *Only applies to USB Interface Boards and Recording Controller Boards*
- *"Set" has no effect when board is running. To change this parameter, stop the board from running (for example, set RunMode to Stop)*
- Acceptable values:
    - Integer value within the range "1" to "16". Note: For USB Interface Boards, the Digital Input with value 1 is DIGITAL-IN-00, whereas for Recording Controllers, the Digital Input with value 1 is DIGITAL-IN-01. **(Default: 1)**

NumberAmplifierChannels

- How many amplifier channels are present on this port

- *Only applies to amplifier ports A-H*
- *"Get" command only – cannot be changed without connecting different hardware to the controller and rescanning ports*
- Returned values:
    - Integer value within the range "0" to "128" **(Default: 0)**

NumberAuxiliaryChannels

- How many auxiliary channels are present on this port
- *Only applies to amplifier ports A-H*
- *Only applies to USB Interface Boards and Recording Controllers*
- *"Get" command only – cannot be changed without connecting different hardware to the controller and rescanning ports*
- Returned values:
    - Integer value within the range "0" to "6" **(Default: 0)**

NumberSupplyVoltageChannels

- How many supply voltage channels are present on this port
- *Only applies to amplifier ports A-H*
- *Only applies to USB Interface Boards and Recording Controllers*
- *"Get" command only – cannot be changed without connecting different hardware to the controller and rescanning ports*
- Returned values:
    - Integer value within the range "0" to "2" **(Default: 0)**

Channel Manipulation

A channel's native name (which cannot be changed by the user) can be used in conjunction with a "channel parameter" in order to conduct get and set operations on an individual channel. The channel parameter should be included immediately after a period following the channel name, much like accessing a member within a struct in the C programming language. As an example, to disable channel 23 on Port B, the command would be: *set b-023.recordingenabled false*.

Accessing channels can vary from system to system. Use the syntax for your specific system in order to access of each of the signal group types:

USB Interface Board

- Amplifier signals: ***PortLetter*-*ChannelNumber*** where PortLetter can be A, B, C, or D, and ChannelNumber is a three-digit number between 000 and 127. Example: B-024
- Auxiliary input signals: ***PortLetter***-AUX***ChannelNumber*** where PortLetter can be A, B, C, or D, and ChannelNumber is a one-digit number between 1 and 6. Example: C-AUX3
- Supply voltage signals: ***PortLetter***-*VDD****ChannelNumber*** where PortLetter can be A, B, C, or D, and ChannelNumber is a one-digit number between 1 and 2. Example: A-VDD1
- Analog input signals: ANALOG-IN-***ChannelNumber*** where ChannelNumber is a two-digit number between 00 and 07. Example: ANALOG-IN-01

- Analog output signals: *cannot be directly controlled through TCP*
- Digital input signals: DIGITAL-IN-***ChannelNumber*** where ChannelNumber is a two-digit number between 00 and 15. Example: DIGITAL-IN-06
- Digital output signals: DIGITAL-OUT-***ChannelNumber*** where ChannelNumber is a two-digit number between 00 and 15. Example: DIGITAL-OUT-03

Recording Controller

- Amplifier signals: ***PortLetter-ChannelNumber*** where PortLetter can be A, B, C, or D, (if the controller is 1024 channels, PortLetter can also be E, F, G, or H) and ChannelNumber is a three-digit number between 000 and 127. Example: B-024
- Auxiliary input signals: ***PortLetter***-AUX***ChannelNumber*** where PortLetter can be A, B, C, or D, and ChannelNumber is a one-digit number between 1 and 6. Example: C-AUX3
- Supply voltage signals: ***PortLetter***-*VDD****ChannelNumber*** where PortLetter can be A, B, C, or D, and ChannelNumber is a one-digit number between 1 and 2. Example: A-VDD1
- Analog input signals: ANALOG-IN-***ChannelNumber*** where ChannelNumber is a one-digit number between 1 and 8. Example: ANALOG-IN-1
- Analog output signals: *cannot be directly controlled through TCP*
- Digital input signals: DIGITAL-IN-***ChannelNumber*** where ChannelNumber is a one or two-digit number between 1 and 16. Example: DIGITAL-IN-2 or DIGITAL-IN-13
- Digital output signals: DIGITAL-OUT-***ChannelNumber*** where ChannelNumber is a one or two-digit number between 1 and 16. Example: DIGITAL-OUT-3 or DIGITAL-OUT-12

Stimulation/Recording Controller

- Amplifier signals: ***PortLetter-ChannelNumber*** where PortLetter can be A, B, C, or D, and ChannelNumber is a three-digit number between 000 and 127. Example: B-024
- Auxiliary input signals: *cannot be directly controlled through TCP*
- Supply voltage signals: *cannot be directly controlled through TCP*
- Analog input signals: ANALOG-IN-***ChannelNumber*** where ChannelNumber is a one-digit number between 1 and 8. Example: ANALOG-IN-1
- Analog output signals: ANALOG-OUT-***ChannelNumber*** where ChannelNumber is a one-digit number between 1 and 8. Example: ANALOG-OUT-8
- Digital input signals: DIGITAL-IN-***ChannelNumber*** where ChannelNumber is a one or two-digit number between 1 and 16. Example: DIGITAL-IN-2 or DIGITAL-IN-13
- Digital output signals: DIGITAL-OUT-***ChannelNumber*** where ChannelNumber is a one or two-digit number between 1 and 16. Example: DIGITAL-OUT-3 or DIGITAL-OUT-12

Example 1. To disable channel 23 on Port B's recording ability:

set b-023.enabled false

Example 2. To set Analog Input 2's custom name to "SignalGenerator":

set analog-in-2.customchannelname

Example 3. To determine the color of Digital Output 4:

get digital-out-04.color

Color

- Which color this channel's waveform will be displayed as in the waveform plot
- Acceptable values:
  - RGB values in the following format: "#RRGGBB" where each digit is a hexadecimal digit. For example, red would be "#FF0000" **(Default: Assigned through software to cycle through the main colors of the rainbow)**

Reference

- Which channel is used as a reference for this channel, subtracting the reference's voltage from this channel's voltage
- If no external reference is desired, a value of "Hardware" will simply use the hardware reference
- *Only applies to headstage amplifier channels*
- Acceptable values:
  - Any alphanumeric text, for example, "A-000" **(Default: "Hardware")**

UserOrder

- How high in the waveform display this channel is, which changes when the user rearranges channels
- Acceptable values:
  - Integer value within the range "0" to "2000" **(Default: In order corresponding to channel number)**

GroupID

- Which group this channel is a part of
- Acceptable values:
  - Integer value within the range "0" to "2000" **(Default: 0)**

Enabled

- Whether or not recording is enabled for this channel
- Acceptable values:
  - "True" – **(Default for Amplifier, Auxiliary Input, and Supply Voltage Channels)**
  - "False" – **(Default for Analog Input, Analog Output, Digital Input, and Digital Output Channels)**

NativeChannelName

- The permanent, inherent name of a channel that cannot be changed by any means
- *"Get" command only – cannot be changed without connecting different hardware to the controller and rescanning ports*
- Returned values:
  - Alphanumeric text. For example, "A-000", "A-AUX1", "ANALOG-IN-03"

CustomChannelName

- The custom name of a channel that can be changed by the user
- Until this name is changed, the default CustomChannelName is identical to the NativeChannelName
- Acceptable values:
    - Any alphanumeric text excluding white-space **(Default: Same as NativeChannelName)**

TCPDataOutputEnabled

- Whether or not this channel will have its data output via TCP when a valid TCP data output connection is made. For amplifier channels, this refers to the wideband waveform
- This data is output via the socket configured through TCPWaveformDataOutputHost, TCPWaveformDataOutputPort, and TCPWaveformDataOutputConnectionStatus
- Acceptable values:
    - "True"
    - "False" **(Default)**

TCPDataOutputEnabledLow

- Whether or not this channel will have its lowpass data output via TCP when a valid TCP data output connection is made
- This data is output via the socket configured through TCPWaveformDataOutputHost, TCPWaveformDataOutputPort, and TCPWaveformDataOutputConnectionStatus
- *Only applies to headstage amplifier channels*
- Acceptable values:
    - "True"
    - "False" **(Default)**

TCPDataOutputEnabledHigh

- Whether or not this channel will have its highpass data output via TCP when a valid TCP data output connection is made
- This data is output via the socket configured through TCPWaveformDataOutputHost, TCPWaveformDataOutputPort, and TCPWaveformDataOutputConnectionStatus
- *Only applies to headstage amplifier channels*
- Acceptable values:
    - "True"
    - "False" **(Default)**

TCPDataOutputEnabledSpike

- Whether or not this channel will have its spike data output via TCP when a valid TCP data output connection is made
- This data is output via the socket configured through TCPSpikeDataOutputHost, TCPSpikeDataOutputPort, and TCPSpikeDataOutputConnectionStatus
- *Only applies to headstage amplifier channels*
- Acceptable values:
    - "True"
    - "False" **(Default)**

TCPDataOutputEnabledDC

- Whether or not this channel will have its DC amplifier data output via TCP when a valid TCP data output connection is made
- This data is output via the socket configured through TCPWaveformDataOutputHost, TCPWaveformDataOutputPort, and TCPWaveformDataOutputConnectionStatus
- *Only applies to Stimulation/Recording Controllers*
- *Only applies to headstage amplifier channels*
- Acceptable values:
    - "True"
    - "False" **(Default)**

TCPDataOutputEnabledStim

- Whether or not this channel will have its stimulation data output via TCP when a valid TCP data output connection is made
- This data is output via the socket configured through TCPWaveformDataOutputHost, TCPWaveformDataOutputPort, and TCPWaveformDataOutputConnectionStatus
- *Only applies to Stimulation/Recording Controllers*
- *Only applies to headstage amplifier channels*
- Acceptable values:
    - "True"
    - "False" **(Default)**

SpikeThresholdMicroVolts

- What voltage threshold must be crossed to trigger spike detection, in microVolts
- *Only applies to headstage amplifier channels*
- Acceptable values:
    - Integer value within the range "-5000" to "5000", in uV **(Default: -70)**

Stimulation Parameters

The following parameters are accessed for each channel in the same manner as the other Channel Manipulation parameters. However, these are all exclusively valid with the Stimulation/Recording Controller, and they dictate the parameters of triggered stimulation for Amplifier Channels, Analog Output Channels, and Digital Output Channels. Setting these parameters updates their values in the IntanRHX software, but in order for the changes to take effect when the board begins running again, the parameters must be uploaded to the controller by Executing the command UploadStimParameters.

Amplifier Channel Stimulation Parameters

Shape

- Which shape the stimulation waveform will have
- Acceptable values:
    - "Biphasic" – A positive/negative current at a given magnitude immediately followed by a current with the opposite polarity **(Default)**

- "BiphasicWithInterphaseDelay" – A positive/negative current at a given magnitude followed by a current with the opposite polarity, with a delay of zero current between the two
- "Triphasic" – A positive/negative current at given magnitude immediately followed by a current with the opposite polarity, which is immediately followed by a current with the first current's polarity, magnitude, and duration

Polarity

- Whether the stimulation waveform will begin with negative (cathodic) or positive (anodic) current
- Acceptable values:
  - "NegativeFirst" – The first phase of the stimulation waveform will inject negative charge
  - "PositiveFirst" – The first phase of the stimulation waveform will inject positive charge

Source

- Which Digital Input, Analog Input, or Keypress event will trigger this stimulation waveform
- Acceptable values:
  - "DigitalInXX" where XX is a 2-digit number from 01 to 16 **(Default: DigitalIn01)**
  - "AnalogInXX" where XX is a 2-digit number from 01 to 08
  - "KeyPressFX" where X is a 1-digit number from 1 to 8, corresponding to the F1-F8 keys on the keyboard

TriggerEdgeOrLevel

- Whether the event that triggers stimulation will be activated once on a rising/falling edge or continuously stay active above or below a certain level
- Acceptable values:
  - "Edge" – A trigger event will occur once when the value on the trigger source changes from inactive to active **(Default)**
  - "Level" – Trigger events will occur continuously when the value on the trigger source is active until the value becomes an inactive value

TriggerHighOrLow

- Whether the event that triggers stimulation will be activated when a high or low value is detected
- Acceptable values:
  - "High" – A trigger event will occur when the trigger source detects a high value **(Default)**
  - "Low" – A trigger event will occur when the trigger source detects a low value

PulseOrTrain

- Whether the stimulation waveform will occur once as a single pulse, or if it will be repeated multiple times as a pulse train
- Acceptable values:
  - "SinglePulse" – A single trigger event will result in a single pulse of the stimulation waveform **(Default)**
  - "PulseTrain" – A single trigger event will result in multiple pulses of the stimulation waveform (the number of pulses is controlled through NumberOfStimPulses)

StimEnabled

- Whether or not stimulation is enabled on this channel
- Acceptable values:
  - "True" – A trigger event will result in stimulation activation
  - "False" – A trigger event will not result in stimulation activation **(Default)**

MaintainAmpSettle

- Whether or not amp settle (if EnableAmpSettle is True) will be maintained during the entirety of the pulse train period, including the time between consecutive stimulation pulses
- Acceptable values:
  - "True" – Amp settle will remain active throughout the entirety of the pulse train
  - "False" – Amp settle will only activate during, immediately before, and immediately after each individual stimulation pulse **(Default)**

EnableAmpSettle

- Whether or not amp settle will activate during, immediately before, and immediately after stimulation
- The time period before and after stimulation for which amp settle shall activate are controlled through PreStimAmpSettleMicroseconds and PostStimAmpSettleMicroseconds
- Acceptable values:
  - "True" – Amp settle will activate when stimulation occurs **(Default)**
  - "False" – Amp settle will not activate when stimulation occurs

EnableChargeRecovery

- Whether or not charge recovery will activate after stimulation has finished
- The time period after stimulation and for how long charge recovery shall activate are controlled through PostStimChargeRecovOnMicroseconds and PostStimChargeRecovOffMicroseconds
- Acceptable values:
  - "True" – Charge recovery will activate when stimulation has finished
  - "False" – Charge recovery will not activate when stimulation has finished **(Default)**

FirstPhaseDurationMicroseconds

- How many microseconds the first phase of stimulation will last for
- Acceptable values:
  - Decimal value within the range "0" to "5000", in µs **(Default: 100)**

SecondPhaseDurationMicroseconds

- How many microseconds the second phase of stimulation will last for
- Acceptable values:
  - Decimal value within the range "0" to "5000", in µs **(Default: 100)**

InterphaseDelayMicroseconds

- How many microseconds between the first and second phases of stimulation zero current will be output for, if Shape is BiphasicWithInterphaseDelay
- Acceptable values:
  - Decimal value within the range "0" to "5000", in µs **(Default: 100)**

FirstPhaseAmplitudeMicroAmps

- The amplitude of the first phase of stimulation, in microAmps
- This value is limited to integer multiples (between 1 and 255) of stimulation step size
- Acceptable values:
    - Decimal value within the range “0” to “2550”, in µA **(Default: 0)**

SecondPhaseAmplitudeMicroAmps

- The amplitude of the second phase of stimulation, in microAmps
- This value is limited to integer multiples (between 1 and 255) of stimulation step size
- Acceptable values:
    - Decimal value within the range “0” to “2550”, in µA **(Default: 0)**

PostTriggerDelayMicroseconds

- How many microseconds between the stimulation trigger and the beginning of the stimulation waveform
- Acceptable values:
    - Decimal value within the range “0” to “500000”, in µs **(Default: 0)**

PulseTrainPeriodMicroseconds

- How many microseconds each single stimulation pulse lasts before the next stimulation pulse in the pulse train begins
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 10000)**

RefractoryPeriodMicroseconds

- How many microseconds after a complete stimulation event (either SinglePulse or PulseTrain) pass with no further stimulation before the next stimulation event can trigger
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 1000)**

PreStimAmpSettleMicroseconds

- How many microseconds before a stimulation waveform triggers for which amp settle is activated, if EnableAmpSettle is True
- Acceptable values:
    - Decimal value within the range “0” to “500000”, in µs **(Default: 0)**

PostStimAmpSettleMicroseconds

- How many microseconds after a stimulation waveform ends for which amp settle is activated, if EnableAmpSettle is True
- Acceptable values:
    - Decimal value within the range “0” to “500000”, in µs **(Default: 1000)**

PostStimChargeRecovOnMicroseconds

- How many microseconds after a stimulation waveform ends until charge recovery begins, if EnableChargeRecovery is True
- The total length of time that charge recovery is active is PostStimChargeRecovOffMicroseconds - PostStimChargeRecovOnMicroseconds

- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 0)**

PostStimChargeRecovOffMicroseconds

- How many microseconds after a stimulation waveform ends until charge recovery ends, if EnableChargeRecovery is True
- The total length of time that charge recovery is active PostStimChargeRecovOffMicroseconds – PostStimChargeRecovOnMicroseconds
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 0)**

NumberOfStimPulses

- How many stimulation pulses will occur for a single stimulation trigger, if PulseOrTrain is PulseTrain
- For a single stimulation pulse, PulseOrTrain should be SinglePulse
- Acceptable values:
    - Integer value within the range “0” to “256” **(Default: 2)**

Analog Out Channel Stimulation Parameters

Shape

- Which shape the stimulation waveform will have
- Acceptable values:
    - “Biphasic” – A positive/negative voltage at a given magnitude immediately followed by a current with the opposite polarity **(Default)**
    - “BiphasicWithInterphaseDelay” – A positive/negative voltage at a given magnitude followed by a voltage with the opposite polarity, with a delay at BaselineVoltageVolts between the two phases
    - “Triphasic” – A positive/negative voltage at given magnitude immediately followed by a voltage with the opposite polarity, which is immediately followed by a voltage with the first voltage’s polarity, magnitude, and duration
    - “Monophasic” – A positive/negative voltage at a given magnitude

Polarity

- Whether the stimulation waveform will begin with negative or positive voltage
- Acceptable values:
    - “NegativeFirst” – The first phase of the stimulation waveform will have negative voltage
    - “PositiveFirst” – The first phase of the stimulation waveform will have positive voltage

Source

- Which Digital Input, Analog Input, or Keypress event will trigger this stimulation waveform
- Acceptable values:
    - “DigitalInXX” where XX is a 2-digit number from 01 to 16 **(Default: DigitalIn01)**
    - “AnalogInXX” where XX is a 2-digit number from 01 to 08

- "KeyPressFX" where X is a 1-digit number from 1 to 8, corresponding to the F1-F8 keys on the keyboard

TriggerEdgeOrLevel

- Whether the event that triggers stimulation will be activated once on a rising/falling edge or continuously stay active above or below a certain level
- Acceptable values:
  - "Edge" – A trigger event will occur once when the value on the trigger source changes from inactive to active **(Default)**
  - "Level" – Trigger events will occur continuously when the value on the trigger source is active until the value becomes an inactive value

TriggerHighOrLow

- Whether the event that triggers stimulation will be activated when a high or low value is detected
- Acceptable values:
  - "High" – A trigger event will occur when the trigger source detects a high value **(Default)**
  - "Low" – A trigger event will occur when the trigger source detects a low value

PulseOrTrain

- Whether the stimulation waveform will occur once as a single pulse, or if it will be repeated multiple times as a pulse train
- Acceptable values:
  - "SinglePulse" – A single trigger event will result in a single pulse of the stimulation waveform **(Default)**
  - "PulseTrain" – A single trigger event will result in multiple pulses of the stimulation waveform (the number of pulses is controlled through NumberOfStimPulses)

StimEnabled

- Whether or not stimulation is enabled on this channel
- Acceptable values:
  - "True" – A trigger event will result in stimulation activation
  - "False" – A trigger event will not result in stimulation activation **(Default)**

FirstPhaseDurationMicroseconds

- How many microseconds the first phase of stimulation will last for
- Acceptable values:
  - Decimal value within the range "0" to "5000", in µs **(Default: 100)**

SecondPhaseDurationMicroseconds

- How many microseconds the second phase of stimulation will last for
- Acceptable values:
  - Decimal value within the range "0" to "5000", in µs **(Default: 100)**

InterphaseDelayMicroseconds

- How many microseconds between the first and second phases of stimulation BaselineVoltageVolts will be output for, if Shape is BiphasicWithInterphaseDelay

- Acceptable values:
    - Decimal value within the range “0” to “5000”, in µs **(Default: 100)**

FirstPhaseAmplitudeVolts

- The amplitude of the first phase of stimulation, in Volts
- Acceptable values:
    - Decimal value within the range “0” to “10.24”, in V **(Default: 0)**

SecondPhaseAmplitudeVolts

- The amplitude of the second phase of stimulation, in Volts
- Acceptable values:
    - Decimal value within the range “0” to “10.24”, in V **(Default: 0)**

BaselineVoltageVolts

- The default, baseline voltage which will be output when one of the phases of the stimulation waveform is not occurring
- Acceptable values:
    - Decimal value within the range “-10.24” to “10.24”, in V **(Default: 0)**

PostTriggerDelayMicroseconds

- How many microseconds between the stimulation trigger and the beginning of the stimulation waveform
- Acceptable values:
    - Decimal value within the range “0” to “500000”, in µs **(Default: 0)**

PulseTrainPeriodMicroseconds

- How many microseconds each single stimulation pulse lasts before the next stimulation pulse in the pulse train begins
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 10000)**

RefractoryPeriodMicroseconds

- How many microseconds after a complete stimulation event (either SinglePulse or PulseTrain) pass with no further stimulation before the next stimulation event can trigger
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 1000)**

NumberOfStimPulses

- How many stimulation pulses will occur for a single stimulation trigger, if PulseOrTrain is PulseTrain
- For a single stimulation pulse, PulseOrTrain should be SinglePulse
- Acceptable values:
    - Integer value within the range “0” to “256” **(Default: 2)**

Digital Output Channel Stimulation Parameters

Source

- Which Digital Input, Analog Input, or Keypress event will trigger this stimulation waveform
- Acceptable values:
    - "DigitalInXX" where XX is a 2-digit number from 01 to 16 **(Default: DigitalIn01)**
    - "AnalogInXX" where XX is a 2-digit number from 01 to 08
    - "KeyPressFX" where X is a 1-digit number from 1 to 8, corresponding to the F1-F8 keys on the keyboard

TriggerEdgeOrLevel

- Whether the event that triggers stimulation will be activated once on a rising/falling edge or continuously stay active above or below a certain level
- Acceptable values:
    - "Edge" – A trigger event will occur once when the value on the trigger source changes from inactive to active **(Default)**
    - "Level" – Trigger events will occur continuously when the value on the trigger source is active until the value becomes an inactive value

TriggerHighOrLow

- Whether the event that triggers stimulation will be activated when a high or low value is detected
- Acceptable values:
    - "High" – A trigger event will occur when the trigger source detects a high value **(Default)**
    - "Low" – A trigger event will occur when the trigger source detects a low value

PulseOrTrain

- Whether the stimulation waveform will occur once as a single pulse, or if it will be repeated multiple times as a pulse train
- Acceptable values:
    - "SinglePulse" – A single trigger event will result in a single pulse of the stimulation waveform **(Default)**
    - "PulseTrain" – A single trigger event will result in multiple pulses of the stimulation waveform (the number of pulses is controlled through NumberOfStimPulses)

StimEnabled

- Whether or not stimulation is enabled on this channel
- Acceptable values:
    - "True" – A trigger event will result in stimulation activation
    - "False" – A trigger event will not result in stimulation activation **(Default)**

FirstPhaseDurationMicroseconds

- How many microseconds the first phase of stimulation will last for
- Acceptable values:
    - Decimal value within the range "0" to "5000", in μs **(Default: 100)**

PostTriggerDelayMicroseconds

- How many microseconds between the stimulation trigger and the beginning of the stimulation waveform
- Acceptable values:
    - Decimal value within the range “0” to “500000”, in µs **(Default: 0)**

PulseTrainPeriodMicroseconds

- How many microseconds each single stimulation pulse lasts before the next stimulation pulse in the pulse train begins
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 10000)**

RefractoryPeriodMicroseconds

- How many microseconds after a complete stimulation event (either SinglePulse or PulseTrain) pass with no further stimulation before the next stimulation event can trigger
- Acceptable values:
    - Decimal value within the range “0” to “1000000”, in µs **(Default: 1000)**

NumberOfStimPulses

- How many stimulation pulses will occur for a single stimulation trigger, if PulseOrTrain is PulseTrain
- For a single stimulation pulse, PulseOrTrain should be SinglePulse
- Acceptable values:
    - Integer value within the range “0” to “256” **(Default: 2)**

Dictionary of “execute” commands

MeasureImpedance

- Runs an impedance measurement on all connected amplifier channels at the frequency specified by ActualImpedanceFreqHertz, and can take a few seconds to complete. Once the impedances have been measured, they can be viewed through the GUI and saved in CSV Format through the SaveImpedance command
- *This command has no effect when board is running. To execute this command, stop the board from running (for example, set RunMode to Stop)*

SaveImpedance

- Once an impedance measurement has been completed (for example, by executing the MeasureImpedance command), saves all channels’ impedance values in CSV Format. The location and name of this CSV file can be changed through the ImpedanceFilename parameter

RescanPorts

- Rescans all amplifier ports to re-establish a connection with all present amplifier headstages

- Should be executed if headstages have been added or removed from the controller after the software began
- *This command has no effect when board is running. To execute this command, stop the board from running*

ConnectTCPWaveformDataOutput

- Open a server TCP socket that another piece of software can connect to, either locally or via network, to stream waveform data (raw data from non-headstage channels, and wideband, low-pass, high-pass, dc amplifier, and stimulation data from headstage channels; spike data is sent separately)
- Details of this output socket can be further configured through TCPWaveformDataOutputHost, TCPWaveformDataOutputPort, and TCPWaveformDataOutputStatus

ConnectTCPSpikeDataOutput

- Open a server TCP socket that another piece of software can connect to, either locally or via network, to stream spike data from headstage channels
- Details of this output socket can be further configured through TCPSpikeDataOutputHost, TCPSpikeDataOutputPort, and TCPSpikeDataOutputStatus

DisconnectTCPWaveformDataOutput

- Disconnect a server TCP socket (previously opened through ConnectTCPWaveformDataOutput) to prevent further sending of waveform data

DisconnectTCPSpikeDataOutput

- Disconnect a server TCP socket (previously opened through ConnectTCPSpikeDataOutput) to prevent further sending of spike data

ClearAllDataOutputs

- Go through all channels of all signal groups to set TCPDataOutputEnabled False for each (for Amplifier Channels, TCP output for all filter bands are disabled as well)
- This command is useful to ensure no TCP outputs are enabled before configuring the intended TCP outputs

UploadBandwidthSettings

- Upload settings related to the amplifier bandwidth to controller (for example, parameters like DesiredLowerBandwidthHertz, DesiredUpperBandwidthHertz). If these parameters are changed via TCP, this command must be called before running the board in order to actually take effect in the hardware
- Since the same USB bus is used for this upload, UploadAmpSettleSettings, UploadChargeRecoverySettings, UploadStimParameters, and data transmission, users should ensure UploadInProgress is False before executing this command. If this command is attempted while UploadInProgress is True, this command will be aborted and an error message will be sent to the client
- *This command has no effect when board is running. To execute this command, stop the board from running*

UploadAmpSettleSettings

- Upload settings related to stimulation amplifier settle to controller (for example, parameters like ActualLowerSettleBandwidthHertz, UseFastSettle, HeadstageGlobalSettle). If these parameters are changed via TCP, this command must be called before running the board in order to actually take effect in the hardware
- Since the same USB bus is used for this upload, UploadBandwidthSettings, UploadChargeRecoverySettings, UploadStimParameters, and data transmission, users should ensure UploadInProgress is False before executing this command. If this command is attempted while UploadInProgress is True, this command will be aborted and an error message will be sent to the client
- *Only applies to Stimulation/Recording Controller*
- *This command has no effect when board is running. To execute this command, stop the board from running*

UploadChargeRecoverySettings

- Upload settings related to stimulation charge recovery to controller (for example, parameters like ChargeRecovery, ChargeRecoveryCurrentLimitMicroAmps, and ChargeRecoveryTargetVoltageVolts). If these parameters are changed via TCP, this command must be called before running the board in order to actually take effect in the hardware
- Since the same USB bus is used for this upload, UploadBandwidthSettings, UploadAmpSettleSettings, UploadStimParameters, and data transmission, users should ensure UploadInProgress is False before executing this command. If this command is attempted while UploadInProgress is True, this command will be aborted and an error message will be sent to the client
- *Only applies to Stimulation/Recording Controller*
- *This command has no effect when board is running. To execute this command, stop the board from running*

UploadStimParameters *[Optional Parameter]*

- Upload parameters related to the stimulation waveform to controller (all parameters listed in the Stimulation Parameters section). If these parameters are changed via TCP, this command must be called before running the board in order to actually take effect in the hardware
- The optional parameter can be a single channel's NativeChannelName to only upload stimulation parameters for that channel. If this optional parameter is omitted, then stimulation parameters from all Amplifier, Analog Out, and Digital Out channels present on the system will be uploaded. This takes a longer amount of time the more channels are present. At least a few seconds (roughly 3 seconds for 32 Amplifier channels) must be given after this command is issued before the USB port can be used for anything else (other uploading or streaming of data when board starts running) to avoid data corruption of multiple routines using the same USB bus
- Since the same USB bus is used for this upload, UploadBandwidthSettings, UploadAmpSettleSettings, UploadChargeRecoverySettings, and data transmission, users should ensure UploadInProgress is False before executing this command. If this command is attempted

while UploadInProgress is True, this command will be aborted and an error message will be sent to the client
- *Only applies to Stimulation/Recording Controller*
- *This command has no effect when board is running. To execute this command, stop the board from running*

ManualStimTriggerOn *[Parameter]*
- Sends a "trigger on" signal for the key corresponding to the specified parameter. For example, ManualStimTriggerOn F1 achieves the same effect as pressing down on the F1 key
- *Only applies to Stimulation/Recording Controller*
- Parameter can be "F1" – "F8"

ManualStimTriggerOff *[Parameter]*
- Sends a "trigger off" signal for the key corresponding to the specified parameter. For example, ManualStimTriggerOff F1 achieves the same effect as releasing the F1 key
- *Only applies to Stimulation/Recording Controller*
- Parameter can be "F1" – "F8"

ManualStimTriggerPulse *[Parameter]*
- Sends a "trigger on" signal followed by a "trigger off" signal for the key corresponding to the specified parameter. For example, ManualStimTriggerPulse F1 achieves the same effect as pressing down then quickly releasing the F1 key
- *Only applies to Stimulation/Recording Controller*
- Parameter can be "F1" – "F8"

SetSpikeDetectionThresholds
- Applies the current global spike detection threshold settings (governed by AbsoluteThresholdsEnabled, AbsoluteThresholdMicroVolts, RmsMultipleThreshold, and NegativeRelativeThreshold) across all enabled channels
- Disabled channels will be unaffected
- This operation cannot be undone
- *This command has no effect when board is running. To execute this command, stop the board from running*

SaveSettingsFile *[Parameter]*
- Saves the current software settings (all RHX settings) to the specified settings file. Depending on how long the settings file is, this may take a few seconds to complete, so it is recommended to pause for several seconds before moving on to subsequent commands
- *This command has no effect when board is running. To execute this command, stop the board from running*
- This operation cannot be undone
- Parameter must be full path and file name of settings file *with no spaces*, for example 'C:/Users/Intan/Desktop/settings.xml'

LoadSettingsFile *[Parameter]*

- Loads the specified settings file (all RHX settings). Depending on how long the settings file is, this may take a few seconds to complete, so it is recommended to pause for several seconds before moving on to subsequent commands
- *This command has no effect when board is running. To execute this command, stop the board from running*
- This operation cannot be undone
- Parameter must be full path and file name of settings file *with no spaces*, for example 'C:/Users/Intan/Desktop/settings.xml'

SaveStimulationSettingsFile *[Parameter]*

- Saves the current settings (stimulation parameters only, doesn't include other RHX settings) to the specified settings file. Depending on how long the settings file is, this may take a few seconds to complete, so it is recommended to pause for several seconds before moving on to subsequent commands
- *This command has no effect when board is running. To execute this command, stop the board from running*
- This operation cannot be undone
- Parameter must be full path and file name of settings file *with no spaces*, for example 'C:/Users/Intan/Desktop/settings.xml'

LoadStimulationSettingsFile *[Parameter]*

- Loads the specified settings file (stimulation parameters only, doesn't include other RHX settings). Depending on how long the settings file is, this may take a few seconds to complete, so it is recommended to pause for several seconds before moving on to subsequent commands
- *This command has no effect when board is running. To execute this command, stop the board from running*
- This operation cannot be undone
- Parameter must be full path and file name of settings file *with no spaces*, for example 'C:/Users/Intan/Desktop/settings.xml'