



PRELIMINARY

QS1R VERB Application Note: AN080004

March 2, 2008

QS1Server Command Protocol Description

SDRMAXII GUI sends commands to the QS1RServer over a socket using UDP. There are two types of commands: read and write. Read commands are prefixed by "r" and write commands are prefixed by "w".

The general format for a read command is:

r:*command*]

where **command** is the particular setting you want to read.

The general format for the write command is:

w:*command*[*value*]

where **command** is the particular setting you want to change and **value** is the value you want to set.

QS1RServer responds to read commands by returning the value or a "?" if it does not understand the command sent. The general response to write commands is "OK;" if the command value was accepted, "ER;" if the command value was in error, or "?" if QS1RServer did not understand the write command. All responses are terminated by a semicolon (;).

The following is a description of the currently implemented QS1RServer commands:

Frequency:

- w:freq[7250000]
Sets the frequency to 7250000 Hz.
- r:freq[]
Reads the current frequency in Hz.

AN080004

Filter:

- w:filterhi[4000]
Sets the filter high frequency to 4000 Hz.
- w:filterlo[-4000]
Sets the filter low frequency to -4000 Hz.
- w:filter[3000, -3000]
Sets the filter high and low values in Hz. The above command sets the filter high value to 3000 Hz and the filter low value to -3000 Hz.
- r:filterhi[]
Reads the current filter high frequency in Hz.
- r:filterlo[]
Reads the current filter low frequency in Hz.
- r:filter[]
Reads the current filter high and low values. The value are returned as two numbers separated by a comma.

Init:

- w:init[]
Initializes communication with the QS1R.

Set ID:

- w:setid[8]
Sets the ID that QS1RServer uses to attempt to connect to the QS1R.

Load Firmware:

- w:loadfw[filename]
Loads the firmware file specified into the microcontroller on QS1R.

Load FPGA:

- w:loadfpga[filename]
Loads the FPGA configuration file specified into the FPGA on QS1R.

PGA:

- w:pga[1]

Sets the LTC2208 PGA gain level. Sending a value of 1 sets the PGA to high, sending a value of 0 sets the PGA to low.

Dither:

- w:dither[1]

Sets the LTC2208 Dither on or off. Sending a value of 1 sets the Dither to on, sending a value of 0 sets the Dither to off.

Dither:

- w:random[1]

Sets the LTC2208 Randomizer on or off. Sending a value of 1 sets the Randomizer to on, sending a value of 0 sets the Randomizer to off.

Local Oscillator:

- w:locosc[10000]

Sets the local oscillator in the QS1RServer DSP chain to the frequency specified in Hz.

Start:

- w:start[]

Starts the DSP process and begins audio output.

Stop:

- w:stop[]

Stops the DSP process and ends audio output.

Server Kill:

- w:serverkill[]

Kills the QS1RServer immediately.

Binaural Mode:

- w:bin[1]

Sets the binaural mode to on or off. Sending a value of 1 turns on binaural mode, sending a value of 0 turns off binaural mode.

- r:bin[]

Reads the current binaural mode setting.

Mute:

- w:mute[1]
Sets the mute mode to on or off. Sending a value of 1 turns on mute mode, sending a value of 0 turns off mute mode.
- r:mute[]
Reads the current mute mode setting.

AGC Maximum:

- w:agcmax[90]
Sets the AGC maximum gain to the value specified in dB.
- r:agcmax[]
Reads the current AGC maximum value.

AGC Fixed Gain:

- w:agcfixedgain[90]
Sets the AGC fixed gain to the value specified in dB.
- r:agcfixedgain[]
Reads the current AGC fixed gain value.

DSP Mode:

- w:mode[lsb]
Sets the mode to LSB.
Valid modes are:
 - am
 - sam
 - dsb
 - usb
 - lsb
 - fmn
 - cwu
 - cwl
 - digu
 - digl

- drm
- spec
- r:mode[]
Reads the current mode.

AGC Mode:

- w:agcmode[slow]
Sets the AGC mode to slow.
Valid AGC modes are:
 - off
 - long
 - slow
 - med
 - fast
- r:agcmode[]
Reads the current AGC mode.

Noise Blanker 1:

- w:nb1[1]
Sets the noise blanker 1 mode to on or off. Sending a value of 1 turns on noise blanker 1mode, sending a value of 0 turns off noise blanker 1 mode.
- r:nb1[]
Reads the current noise blanker 1 mode setting.

Noise Blanker 2:

- w:nb2[1]
Sets the noise blanker 2 mode to on or off. Sending a value of 1 turns on noise blanker 1mode, sending a value of 0 turns off noise blanker 2 mode.
- r:nb2[]
Reads the current noise blanker 2 mode setting.

Noise Reduction:

- w:nr[1]

Sets the noise reduction mode to on or off. Sending a value of 1 turns on noise reduction 1mode, sending a value of 0 turns off noise reduction mode.

➤ r:nr[]

Reads the current noise reduction mode setting.

Automatic Noise Filter:

➤ w:anf[1]

Sets the noise filter mode to on or off. Sending a value of 1 turns on noise reduction 1mode, sending a value of 0 turns off noise filter mode.

➤ r:anf[]

Reads the current noise filter mode setting.

Noise Blanker 1 Threshold:

➤ w:nb1threshold[10]

Sets the noise blanker 1 threshold to the value supplied in dB.

➤ r:nb1threshold[]

Reads the noise blanker 1 threshold value.

Noise Blanker 2 Threshold:

➤ w:nb2threshold[10]

Sets the noise blanker 2 threshold to the value supplied in dB.

➤ r:nb2threshold[]

Reads the noise blanker 2 threshold value.

Noise reduction delay:

➤ w:nrdelay[500]

Sets the noise reduction delay to the value specified.

➤ r:nrdelay[]

Reads the noise reduction delay value.

Noise reduction gain:

➤ w:nrgain[2]

Sets the noise reduction gain to the value specified.

- r:nrgain[]
Reads the noise reduction gain value.

Noise reduction leak:

- w:nrleak[2]
Sets the noise reduction leak to the value specified.
- r:nrleak[]
Reads the noise reduction leak value.

Noise filter delay:

- w:anfdelay[500]
Sets the noise filter delay to the value specified.
- r:anfdelay[]
Reads the noise filter delay value.

Noise filter gain:

- w:anfgain[2]
Sets the noise filter gain to the value specified.
- r:anfrgain[]
Reads the noise filter gain value.

Noise filter leak:

- w:anfleak[2]
Sets the noise filter leak to the value specified.
- r:anfleak[]
Reads the noise filter leak value.

Remote:

- w:remote[1]
Sending a value of 1 sets QS1RServer to respond to remote commands over the socket interface. Sending a value of 0 sets QS1RServer to only respond to commands over the local command line interface.
- r:remote[]
Reads the current remote mode.

Input Device:

- w:indevice[0]
Sets the Port Audio input device index to the value specified.
- r:indevice[]
Reads the current Port Audio input device index.

Output Device:

- w:outdevice[0]
Sets the Port Audio output device index to the value specified.
- r:outdevice[]
Reads the current Port Audio output device index.

Host Api:

- w:hostapi[0]
Sets the Port Audio host api index to the value specified.
- r:hostapi[]
Reads the current Port Audio host api index.

Block Size:

- w:blocksize[1024]
Sets the DSP blocksize.
- r:blocksize[]
Reads the current DSP blocksize.

Clear Queues:

- w:clrqueue[]
Clears the audio buffers.

IP Address:

- w:ipaddr[127.0.0.1]
Sets the IP address of the machine that SDRMAXII is running on for the meter data and spectrum data sockets.
- r:ipaddr[]

Reads the IP address that QS1RServer sends the meter and spectrum data to.

Meter Port:

- w:meterport[55668]

Sets the port number that QS1RServer sends the meter data to SDRMAXII on.

- r:meterport]

Reads the current port number use to send the meter data.

Spectrum Port:

- w:specport[55669]

Sets the port number that QS1RServer sends the spectrum data to SDRMAXII on.

- r:specport]

Reads the current port number use to send the spectrum data.

Run Meter:

- w:runmeter[1]

Sending a value of 1 causes QS1RServer to begin sending the meter data to the meter data port. Sending a value of 0 causes QS1RServer to stop sending data to the meter data port.

- r:runmeter]

Reads whether the QS1RServer is sending data to the meter port.

Run Spectrum:

- w:runspec[55669]

Sending a value of 1 causes QS1RServer to begin sending the spectrum data to the spectrum data port. Sending a value of 0 causes QS1RServer to stop sending data to the spectrum data port.

- r:runspec[]

Reads whether the QS1RServer is sending data to the spectrum port.

Volume:

➤ w:volume[0.2,0.2]

Sets the volume to a percentage of full scale for both the left and right channels. Full scale is 100% or 1.0. The first value is for the left channel and the second value is for the right channel.

➤ r:volume[]

Reads the current volume setting. It returns two numbers separated by a comma. The first value is for the left channel and the second value is for the right channel. The returned numbers range from 0 to 1.0.

Volume Left:

➤ w:volleft[0.35]

Sets the left channel audio volume to 35%.

➤ r:volleft[]

Reads the current left channel volume level.

Volume Right:

➤ w:volright[0.40]

Sets the right channel audio volume to 40%.

➤ r:volright[]

Reads the current right channel audio volume level.