ghpsdr3

John Melton g0orx/n6lyt

Introduction

ghpsdr3 is a first attempt at developing software that works with the HPSDR hardware, specifically to work with multiple receivers either running concurrently in a single Mercury card or with multiple Mercury cards.

There is also some support for the softrock receivers and the Flex Radio SDR-1000.

The software has been broken out into multiple servers and clients. The hardware server takes the input either from Ozy if using the HPSDR hardware or from a sound card and outputs a UDP stream of I/Q samples along with some status information such as the sample rate. For the HPSDR running multiple receivers the I/Q samples are demultiplexed and sent out on separate streams.

There are currently three hardware servers:

server - HPSDR hardware server.
softrockserver - softrock server (both fixed frequency ans USB connected si570).
sdr1000server - SDR1000 server using either a parallel port or the USB dongle.

There are currently 2 clients that work with these server. The first is a modified version of ghpsdr that is receive only (ghpsdr was originally developed as a single application interfacing to the HPSDR hardware), and the second is a dspserver that connects to the hardware server and performs all the DSP processing (using DttSP) and has a TCP socket interface fro clients to connect to that they can then send commands to get spectrum samples, set frequency, filters, mode, etc and receive spectrum samples and and audio stream so they can display the spectrum and/or a waterfall and play out the audio. This is specifically design to have low bandwidth so it can work over the internet.

For the dspserver there are 2 clients. One is written in Java (jMonitor) and another written using Qt and C++ (qtMonitor). These are meant to be sample applications to demonstrate capabilites of the system.

My intention was that I would run 1 ghpsdr application on one of the Mercury receivers giving me the full features and then run several jMonitor applications that would allow me to monitor multple bands concurrently.

Source code

All the source code can be found on the HPSDR svn repository at:

svn://64.245.179.219/svn/repos_sdr_hpsdr/trunk/N6LYT

Protocols

The protocols used between the servers and clients is described in a separate document in the ghpsdr3/trunk/doc directory.

Manual Pages

The following manual pages describe the command line options for each of the applications.

NAME

server – the HPSDR server

SYNOPSIS

server [options]

DESCRIPTION

server is the hardware interface to the HPSDR Ozy USB interface.

It sends USB packets to Ozy to configure the hardware, set the frequency, etc, along with the audio output stream.

The server is started with:

server [options]

OPTIONS

--receivers <receivers>
The number of receivers to configure. Currently <receivers> can be in the range 1..4. The default is 1 receiver.

--samplerate <48000|96000|192000>
Set the sampling rate. The default is 96000.

--dither <off|on>
set the LT2208 dither state. The default is on.

--random <off|on>
set the LT2208 random state. The default is on.

--preamp <off|on>
set the preamp state. The default is off.

--10mhzsource <atlas|penelope|mercury>
set the source for the 10MHz clock source. The default is mercury.

--122.88mhzsource <penelope|mercury>
set the 122.88MHz clock source. The default is mercury.

--timing
Debug output timing information to process samplerate samples.
Generally the sample time should be 1000ms +/- 1ms.

NAME

softrock – the Softrock server

SYNOPSIS

softrock [options]

DESCRIPTION

softrock is the hardware interface to Softrock.

It will work with both fixed frequency softrock receivers and also softrock receivers with an si570 USB interface.

The server is started with:

softrock [options]

OPTIONS

--samplerate <48000|96000|192000>
Set the sampling rate. The default is 96000.

--input <audio_device>
Selects the input audio device. The default is 0.

--output <audio_device>
Selects the output audio device. The default is 0.

--iq
Samples in I/Q order. This is the default.

--qi
Samples in Q/I order.

--si570
Use si570 USB interface to set frequency. The default is to not use the si570.

--startupfreq <frequency>
si570 startup frequency. The default is 56.32.

--multiplier <m>
si570 frequency multiplier. The default is 4.

--correctedfreq <f>
si570 corrected xtal frequency. The default is 114.285

--verbose
some more debug output.

NAME

sdr1000 – the SDR-1000 server

SYNOPSIS

sdr1000 [options]

DESCRIPTION

sdr1000 is the hardware interface to Flex Radio SDR-1000.

The SDR-1000 can be connected using either a parallel interface or the USB dongle.

The audio interface uses Port Audio.

The server is started with:

sdr1000 [options]

OPTIONS

--rfe <0|1>
If RFE is present. The default is 0.

--pa <0|1>
If PA is present. The default is 0.

--sampleratet <4800|96000|19200>
Sets the sample rate. The default is 96000.

--usb <0|1>
SDR-1000 control via USB. The deafult is 0.

--lptaddr <hexaddress>
The hex address of the parallel port if not using usb.

--input <device>
Input port audio device.

--output <device>
Output port audio device.

NAME

ghpsdr – the ghpsdr receiver

SYNOPSIS

ghpsdr [options]

DESCRIPTION

ghpsdr is a version of the ghpsdr transceiver code modified to work with the hardware server but currently only supports receiving.

OPTIONS

--server <host>

The host IP address running the hardware server. The default is 127.0.0.1.

--receiver <receiver>

The receiver to connect to.

--sound-card <device>

Sets the soundcard device the for the source of I/Q samples. The default is HPSDR.

<device> can be:

UNSUPPORTED
SANTA_CRUZ
AUDIGY_2_ZS
MP3_PLUS
EXTIGY
DELTA_44
FIREBOX
EDIROL_FA_66
HPSDR"

--sample-rate <48000|96000|192000>

Set the sample rate. The default is 96000.

--offset <x>

Set the I/Q sample offset. The default is 0.
For some SoundBlaster cards the Q sample is delayed so the offset should be set to -1.

--local-audio <0|1>

Output audio locally using the direct audio interface.

--audio-device <device>

Specify the device to use (i.e. /dev/dsp) if local-audio is enabled.

--port-audio <0|1>

Output audio locally using Port Audio. The default is 0.

--port-audio-device <deviceid>
Specify the device to use for Port Audio.

NAME

dspserver – the DSP server

SYNOPSIS

dspserver [options]

DESCRIPTION

dspserver is the DSP processing client of ghpsdr3.

It processes an I/Q data stream fro the hardware server.

It accepts TCP socket connections from clients than can the set frequency, mode, filters, etc, and request spectrum samples and an audio stream.

The audio stream is sent as 8 bit aLaw samples at 8000 samples per second.

The server is started with:

dspserver [options]

OPTIONS

--server <host>

The host IP address running the hardware server. The default is 127.0.0.1.

--receiver <receiver>

The receiver to connect to. The default is 0.

--soundcard <device>

Sets the soundcard device the for the source of I/Q samples. The default is HPSDR.

<device> can be:

UNSUPPORTED
SANTA_CRUZ
AUDIGY_2_ZS
MP3_PLUS
EXTIGY
DELTA_44
FIREBOX
EDIROL_FA_66
HPSDR"

--offset <x>

Set the I/Q sample offset. The default is 0.
For some SoundBlaster cards the Q sample is delayed so the offset should be set to -1.

NAME

jMonitor – a simple internet capable receiver.

SYNOPSIS

java -jar jmonitor.jar [options]

DESCRIPTION

jMonitor is a simple client written in Java to connect to the DSP server.

It can be run as a standalone Java application or it can be run as a Java Applet in a web browser.

The client is started with:

java -jar jmonitor.jar [options]

OPTIONS

--server <host>

The host IP address running the hardware server. The default is 127.0.0.1.

--receiver <receiver>

The receiver to connect to. The default is 0.

--limit <seconds>

Sets a max connection time in seconds. This is really to use when the client is run from a web browser to to limit the amount of time a remote user can stay connected.

NAME

qtMonitor – a simple internet capable receiver.

SYNOPSIS

qtMonitor [options]

DESCRIPTION

qtMonitor is a simple client written in C++ using the Qt GUI toolkit to connect to the DSP server.

The client is started with:

qtMonitor

OPTIONS

There are no options as the host, receiver and audio are selected form the GUI.