Linux - HPSDR

John Melton

G0ORX/N6LYT

john.d.melton@googlemail.com
Existing software to support HPSDR

- PowerSDR – runs on Windows
  - Combination of C and C#

- KISS Konsole – runs on Windows
  - C#
• Goal

  • Write Linux software for HPSDR
SDR Forum
Dayton 2010

• HPSDR Hardware
  • Atlas Bus
  • Ozy/Magister USB I/O
  • Mercury receiver
  • Penelope transmitter
  • LPU Power Supply
  • Excalibur 10 MHz clock source
  • PennyWhistle RF Power Amplifier
  • Pandora Enclosure

• Coming soon …
  • Alex – RF Bandpass Filters
  • Ozy II – Ethernet I/O
  • Hermes – Single board HPSDR transceiver
• HPSDR Hardware

Atlas

Ozy/Magister

Mercury

Penelope
SDR Forum Dayton 2010

- HPSDR Hardware

Hermes
HPSDR Hardware

- FPGA on each card
  - Allows reprogramming of the hardware.
  - Original Mercury FPGA code implemented 1 receiver with either 48KHz, 96KHz or 192KHz bandwidth from anywhere within the 60MHz sampling range of the receiver.
  - New development to implement up to 8 receivers within a single Mercury card.
Software

- USB interface to Ozy (packet rate for 48000 bps samples)
  - 512 byte packets
    - 8 byte header
      - Mox
      - Frequency (Rx or Tx or Both)
      - Full Duplex
      - Rx speed (48000, 96000, 192000)
      - Clock sources
      - Rx preamp
  - 63 Tx I/Q samples and audio samples
    - I and Q samples 16 bits each
    - Audio samples are 2 channels of 16 bits
Software

- USB interface from Ozy (packet rate dependent on sample rate)
  - 512 byte packets
    - 8 byte header
      - PTT
      - Dash/Dot
      - ADC overflow
      - Software version of Ozy, Mercury, Penelope
      - Forward power
    - 63 I/Q and Microphone samples
      - I and Q samples 24 bits each
      - Microphone sample 16 bits
SDR Forum
Dayton 2010

- Software
  - USB bandscope interface from Ozy
    4096 16 bit samples
Software

- ghpsdr
  - Written in C
  - Uses DttSP for DSP functions
  - Uses GTK+ for GUI
SDR Forum
Dayton 2010

Software

Multi threaded
- thread to read USB I/Q and Microphone samples
- thread to read USB bandscope samples
- thread to process I/Q samples
- thread to process bandscope smaples
- thread for spectrum/waterfall display update
- thread for bandscope display update
• Software

• Modular source code

agc.c     filter.c     ozy.c     spectrum_update.c
audio.c   frequency.c  ozy_ringbuffer.c subrx.c
audiostream.c hpsdr_setup.c preamp.c  transmit.c
band.c    libusbio.c   receiver.c vfo.c
bandscope.c bandscope_control.c main.c  setup.c  volume.c
bandscope_update.c meter.c sinewave.c xvtr_setup.c
command.c  meter_update.c soundcard.c
display.c   mode.c    spectrum_buffers.c
display_setup.c ozy_buffers.c spectrum.c
SDR Forum
Dayton 2010

- Software
  ghpsdr
Software

ghpsdr
• Software

ghpsdr
Software

But what about Mercury supporting multiple receivers?
USB I/Q Data Stream

USB data is transferred in 512 byte packets.

8 bytes of header data.

Receiver I/Q data and Mic sample.

I and Q samples are 24 bits (3 bytes).
Microphone samples are 16 bits (2 bytes).

Number of samples per packet varies depending on number of receivers.
Software

- Commands
- USB
- I/Q samples
- RX-0 Buffer
- RX-n Buffer
- RX-0 I/Q
- RX-n I/Q
- ghpsdr (RX-0)
- ghpsdr (RX-n)

TCP connection for Commands
I/Q data sent over UDP
SDR Forum
Dayton 2010

- Software
• Software
Software

Bandwidth Requirements

24 bit I samples
24 bit Q samples

48000 samples per second = 2304000 bits per second
96000 samples per second = 4608000 bits per second
192000 samples per second = 9216000 bits per second

For just 1 receiver!

8 receivers at 192000 = 73728000 bits per second

My broadband uplink speed is 360 Kbps
• Software

**SDR Forum**
**Dayton 2010**

- **HPSDR Server**
  - **RX 0 Buffer**
  - **RX n Buffer**

- **DSP Server (RX-0)**
- **ghpsdr (RX-n)**

- **Internet Client**

**Connections:**
- USB I/Q samples
- Control & Audio
- Commands
- Spectrum Data
- Audio data

**Routes:**
- RX-0 I/Q
- RX-n I/Q
- Commands
Software

- Bandwidth requirements per internet receiver

  - Spectrum data sent as 8 bit values
  - Client requests number of samples - currently 480 at 10 requests per second.

  - Audio data sent as 8-bit aLaw at 8000 samples per second.

  - Spectrum data = 480*10*8 = 34800 bits per second
  - Audio data = 8000*8 = 64000 bits per second

  - 98800 bits per second per receiver
SDR Forum Dayton 2010

- Software

  - Java Applet Client
    Runs in any browser window.
    Can be run as a standalone application.
• Software
  – iPhone client
・Software

- Android Client
**Software**

Ongoing development to support other hardware

SDR1000 server
Softrock server (both fixed Xtal and Si570)
UHFSDR server
Thanks

- HPSDR Group
- TAPR Group
- Phil Harman VK6APH
- Bob McGwier N4HY
- Frank Brickle AB2KT
- Ken Hopper N9VV
- Kirk Weedman KD7IRS
- Bruce Walker W1BW
References

- Source code in HPSDR svn repository:

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