Development and Design of Firmware Programming Tools for the openHPSDR Hardware

David R. Larsen

1KV0S
Columbia, Missouri, USA
OpenHPSDR Development Group
openhpsdr.org

Digital Communication Conference, 2017
St Louis, Missouri
Profession

- Professor of Forestry.
- Work on Forest Dynamics.
- Growth Modeling.
- Statistical Analysis.
- Code development for Forestry applications.
- Unix, Linux programmer.
Amateur Radio

- First licensed in 1975.
- Live happened!
- Relicensed in 2004.
- First DCC, Chicago, 2008.
Scotty asked me to take it on in 2008.

Also started and attend the Friday night openHPSDR teamspeak sessions.

Role in organizing the repositories.
Motivation
Programming contributions
Clean start
Summary

A Little Personal History
Group Structure & Website

openHPSDR.org

High Performance Software Defined Radio
An Open Source Design

Project Description

Introduction -- What's It All About?

The HPiSDR is an open source (GNU type) hardware and software project intended as a "next generation" Software Defined Radio (SDR) for use by Radio Amateurs ("hams") and Short Wave Listeners (SWLs). It is being designed and developed by a group of SDR enthusiasts with representation from interested experimenters worldwide.

The rationale behind the project is to break the overall design up into a number of modules. Each module is designed by an individual or group and connects to other modules using a pre-defined and common bus -- rather like plugging boards into a PC motherboard.

This modular approach enables prospective users to incorporate just the modules that interest them as well as designing their own variants if desired. The approach also enables new ideas and circuits to be tested by replacing an existing module. Since the majority of modules will be retained, such experimentation can be done with minimum disruption to an existing working system.

The modules vary in complexity from simple bandpass filters and input/output interfaces, to full blown DSP functions. Such variety enables experimenters with varying degrees of experience to contribute.

Thus far, the modules have each been named for easier identification when talking or writing about them. On this website, each module has its own web page, as noted by the tab selections near the top of the page. Some of the modules are being designed so that they can be either used in conjunction with others or stand-alone. Each module board size (except the backplane) will be 100 mm. by 120 to 220 mm. and use either a 56 pin or 64 pin DIN41612 type connector.
Maintain the legacy of old work completed and not.
Help people to find what they need to get started.
Archive teamspeak, listserver, code, papers and videos.
The Wiki.
Early days

- First Boards.
  - Atlas back backplane
  - Janus sound interface
  - Ozymandias first board with
    - Cyclone II FPGA,
    - USB interface
    - Loaded firmware from PC through the USB line
First radio

- **Penelope**
  - Digital up converter (DUC) a 1/2-watt transmitter/exciter board,
  - Cyclone II FPGA
  - Initially programmed with a USB Blaster, Quartus programmer, using JTAG or Flash memory.

- **Mercury**
  - A 0-65MHz Direct Sampling Receiver
  - Cyclone III FPGA
  - Initially programmed with a USB Blaster, Quartus programmer, JTAG or Flash memory.
First radio
Search for Bandwidth

- Problems with USB2
  - Limited bandwidth.
  - USB Driver issues.
    - Windows Centric
    - General not open source
    - No Volunteers to write drivers
    - USB3 a possibility but not widely in 2010
  - Ethernet as an alternative
    - Very stable drivers in all OS platforms.
    - Sufficient bandwidth.
Motivation
Programming contributions
Clean start
Summary

A New Interface Board

Metis
- Ethernet Interface module.
- This required writing Basic TCPIP interface code in FPGA firmware.
- Introduction of the Bootloader Mode.
- Introduction of in Metis JTAG programmer.
- Introduction of simplified Programmer.
- Communication by raw PCAP or UDP protocols
Tools for programming firmware.

- First Tool HPSDRBootloader by John, G0ORX
  - Used PCAP
  - Written in C++ with Qt GUI
    - Required Administrator login.
    - Required setting jumpers on board.
    - Work flow was confusing to occasional users.
In 2012, I started maintain and writing programmer code.
Seperate functionality.
Use PCAP for recovery.
Use UDP for normal updates.
HPSDRBootloader is a program for recovery from programming failure.

HPSDRProgrammer is a program for normal updating.
Design Criteria

- Use PCAP protocol. (MAC Addressing)
- Require Administrator login on computer.
- Require jumpers being set on PCB board.
- Board Discovery uses bootloader firmware
- Allow Metis to function as a JTAG programmer for Penelope and Mercury.
HPSDRBootloader

- Metis
  - Jumper 1
- Hermes
  - Jumper 12
- Metis JTAG programmer
  - Jumper 1
- Angelia
  - Jumper 17
- JTAG
- Mercury
  - Last JTAG Jumper
- Penelope
  - Penny Lane
  - Last JTAG Jumper

David R. Larsen
OpenHPSDR FPGA Programmers
HPSDRBootloader

Motivation
Programming contributions
Clean start
Summary

HPSDRBootloader (version 2.0.4.4) 2015-1-29

File Tools Help

☐ IP ?

Computer Interface

Interface: enp0s20

Computer IP Address: 192.168.1.10
Computer MAC: 00:1F:C6:7E:52:DE

Board Bootloader

Board MAC address: 00:04:A3:64:25:95  Board with Bootloader Found

Test for Bootloader

Board Programmer

/home/dlarsen/Downloads/Hermes_v3.2.rbf

Browse Program

Use Board as a JTAG Programmer

Interrogate

Firmware RBF file

Browse Program

Reading rbf file: /home/dlarsen/Downloads/Hermes_v3.2.rbf

David R. Larsen  OpenHPSDR FPGA Programmers
Design Criteria

- Use UDP protocol. (TCP/IP Addressing)
- No jumpers being set on PCB board.
- Board Discovery uses past Radio firmware
- Can only program boards with an ethernet connector
HPSDRProgrammer

Motivation
Programming contributions
Clean start
Summary

HPSDRProgrammer_V2

Metis
Hermes
Griffin
Angelia

ethernet
Maintainence issues

- Stable and working.
- Changes in the Code (Easy to fix)
- Changes in installers (Easy if you use all the time, Hard if you use once every 3 years).
- Code detritus (Unused left over bits and pieces).
This is my Hobby!

Why start over?

- This is my hobby! I want to learn something new.
- Learning a new computer language.
- Want to clean up the structure and process.
Go Language

Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.

Download Go
Binary distributions available for Linux, MacOS X, Windows, and more.

Featured articles
Go 1.9 is released
Today the Go team is happy to announce the release of Go 1.9. You can get it from the download page. There are many changes to the language, standard library, runtime, and tooling. This post covers the most significant visible ones. Most of the engineering effort put into this release went to improvements of the runtime and tooling, which makes for a less exciting announcement, but nonetheless a great release.
Go Language (golang.org)

- Has the flavor of C.
- First developed at Google but has been open sourced
- Original Three designers.
  - Ken Thompson
  - Rob Pike
  - Robert Greismer
Go Language

Ken Thompson

- Formerly Bell Labs.
- Wrote the B programming language.
- Wrote Plan 9.
Go Language

Rob Pike
- Formerly Bell Labs.
- Author on The Practice of Programming and The Unix Programming Environment, and UNIX
- Author on Plan 9.
- Author of UTF-8
Go Language

Robert Greismer
- Formerly Bell Labs.
- Wrote Sawsall
- Little C background
Go Language


- The group was influenced by Hoare
Go Language

Feature of Go that I found helpful in this project.

- Function testing.
- Cross platform compiling. Linux, Windows, MacOS, Arm, BSD.
- Good packages for networking including PCAP, UDP, HTTP.
- Static binaries.
Go Language

The disadvantages of Go

- No GUI package, either Command line or HTTP
First Programmer in Go

- Started the process before Protocol 2 was available for testing.
- So Started with Protocol 1
- Built and test each component functions.
- Use the command line interface.
Motivation
Programming contributions
Clean start
Summary

First Programmer in Go

![Screenshot of the terminal output showing the HPSDR programmer's interface and address information.](image-url)
Second Programmer in Go

- Started the process Protocol 2 building to the Protocol.
- Built and test each component functions.
- Use the command line interface.
- Very few corrections to make the functions work once Protocol 2 was available.
Second Programmer in Go
At this point I have two fast and clean packages to program HPSDR boards.

Shared these with others for testing.

Most users we uncomfortable with the command line interface.
The next step

- Go have the package to allow the creation of small local httpservers
- The next step is to make a HTTP interface for the packages.
- Program can be contained in a single executable file.
- The server cannot be access outside of the excuting computer, unless you configure that access.
In this process, I just had to work on the user interface.
The packages from could be reused without modification.
In general the extra code is mostly html.
I need a small it of javascript to improve user feedback.
Local Web Programmer

```
$ dlarserndave-Radio --drlc/gocode/src/eak.snr.missouri.edu/daveradio/newradio
   2017/09/04 11:17:40 For a list of commands use --help

2017/09/04 11:17:40 RRF directory /home/dlarsern/Downloads/HPSDRfiles/
2017/09/04 11:17:40 Listening ...
2017/09/04 11:17:40 Point your web browser to: http://localhost:8228/intro/
```
Local Web Programmer

Overview

The HPSDR Programmer is a tool to load >1.7 protocol firmware into HPSDR boards. This program performs the same function as the HPSDRProgrammer. It perform the following tasks.

- Discovery of the HPSDR boards available.
- Changing the HPSDR board to a fixed IPv4 address within your subnet
- Erase and Program an RBF file to the HPSDR Board.

Select Interface  Quit
Local Web Programmer

Network Interfaces

请选择要执行的接口

接口：(网络) MAC (IPv4) (IPv6)

1:lo (): (127.0.0.1) (-1)
2:eth0 (): (00:80:97:98:72:2b) (0)
3:vga0 (): (00:1f:00:67:52:de) (192.168.1.10) (fe80::5448:da02:1c8:973d)
4:docker0 (): (172.17.0.1) (172.17.0.1)

已选择的网络接口

1:lo ()

Select

Quit
Local Web Programmer

HPSDR Programmer

By Dave, W3WS - Version 0.8.8. Protocol: R2.1. Last Updated 2018-8-17 - openhpsdr.org

Computer

Computer: 00:1f:67:e5:2d:65
OS: Linux (x86_64) 4 CPU(s)
User: David Larsen (dlarsen) homeldiansen
IPV4: 192.168.1.10
IPV6: fe80:644b:da02:1cb:9d7d

Radios

Please select from these available Radios


Selected Network interface: 3. enp0s20 (00:1f:67:e5:2d:65)

HERMES (0:4:a3:64:25:95) Select

Select HPSDR Board

Board: HERMES
Board Mac: 0:4:a3:64:25:95
Board Address: 192.168.1.50:1024
Board Status: not running
Protocol: 2.9
Firmware: 10.0
Receivers: 2

Frequency Input: Freq_input
IQ data format: Big-Endian IQ in 3 byte format

Change IP Program Quit
Local Web Programmer

HPSDR Programmer

Firmware File Information

- Found rbf file: /home/dlarson/Downloads/HPSDRfiles/Hermes_16_bit_14_Aug.rbf
- Size rbf file: 554487
- Size rbf in memory: 554496
- Packets: 2106

Program
Local Web Programmer

HPSDR Programmer

Erase flash memory: Erase 12.1 seconds
Programming: Pending

Return  Quit
Local Web Programmer

David R. Larsen

OpenHPSDR FPGA Programmers
Summary

- `openhpsdr.org` has a large number of new and old projects.
- Projects are both hardware and software.
- Many people ask for changes. The bottleneck is the number of people to work on each item.
- Join us, Have fun making your favorite feature or part.

Outlook

- The HPSDR project is over 12 years old at this point.
- I believe that together we have made an impact on Amateur Radio.
Open High Performance Software Defined Radio.
http://openHPSDR.org

Open High Performance Software Defined Radio.
http://openHPSDR.org/beta/