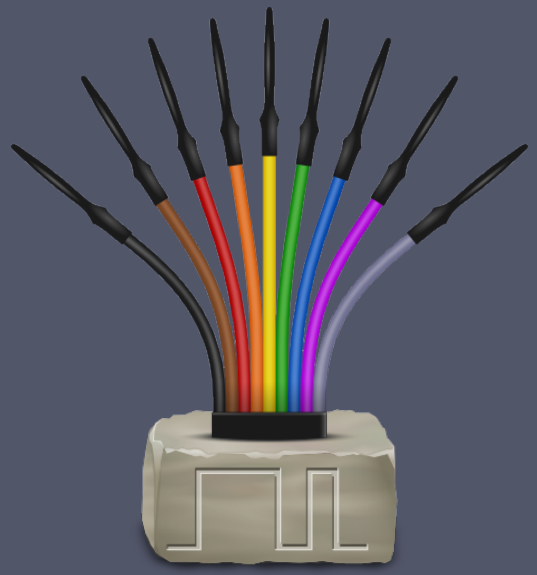


Sigrok: Using Logic to Debug Logic

Matt Ranostay
Intel Open Source Technology Center
matt.ranostay@intel.com

sigrok Overview

- Provides a simple Open Source solution for mostly proprietary and some Open Source Hardware digital logic devices
- Common framework which includes output format, device metadata, and H/W interfacing

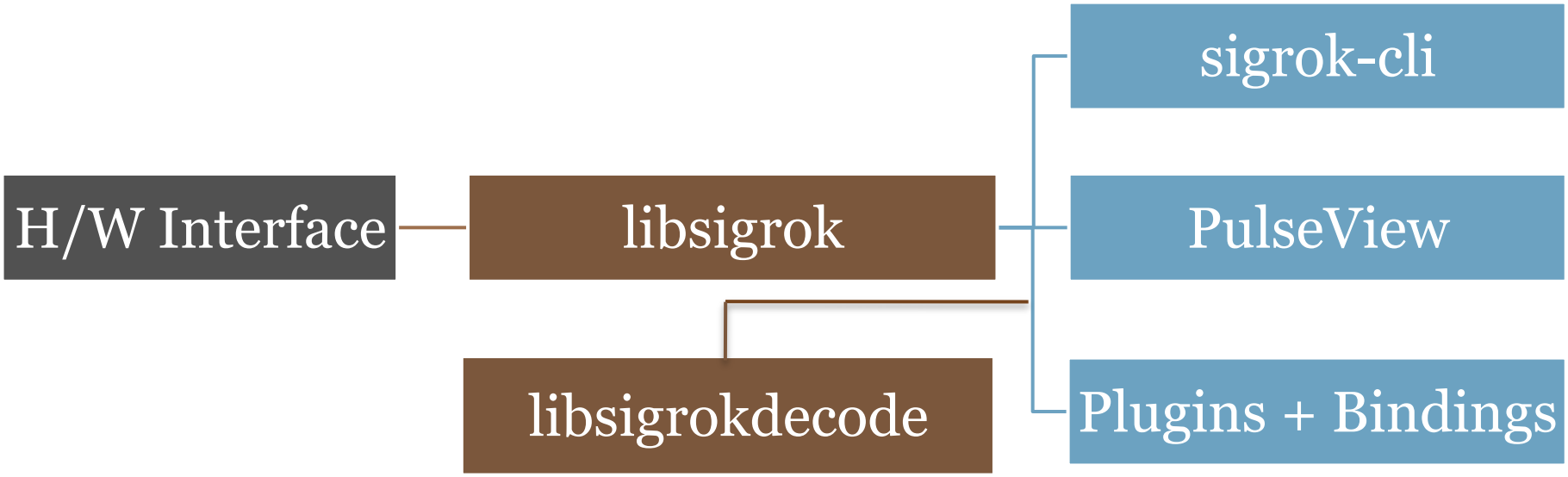
sigrok Project

- Blanket project with various libraries, backends, protocol decoders, third-party firmware, and graphical frontends
- Aims to make a common framework for a various of logic analyzers, oscilloscopes, and other analog/digital debugging devices

Meet The Family

- libsigrok – Heart and brains behind the device communication, functionality, and control
- libsigrokdecode – Python3 interfacing lib in C + protocol decoders
- sigrok-cli – Command line backend for sigrok
- sigrok-util – Various useful scripts + utilities
- sigrok-dumps – Collections of various captures
- fx2lafw – OSS Firmware for Cypress FX2 LAs
- PulseView – sigrok QT GUI frontend

sigrok Components



Examples of Supported Devices

- Logic Analyzers
  - Open Logic Sniffer
  - Saleae Logic/Logic 16
- Oscilloscope
  - Rigol DS1052E
- Mixed-Mode Devices
- Digital Multimeter
- Analog devices like thermometers, hygrometers, light meters, etc
- Full support list available on sigrok wiki

sigrok output format

- Device agnostic and interchangeable
- Simple hexdump to process
- Compressed with zip algorithm
  - Due to most samples being repeats it isn't rare to see compression rates of 100x
- Traces can be broken into chunks (e.g. logic-* files in archive)
- Common metadata that is useful for clients and protocol decoding

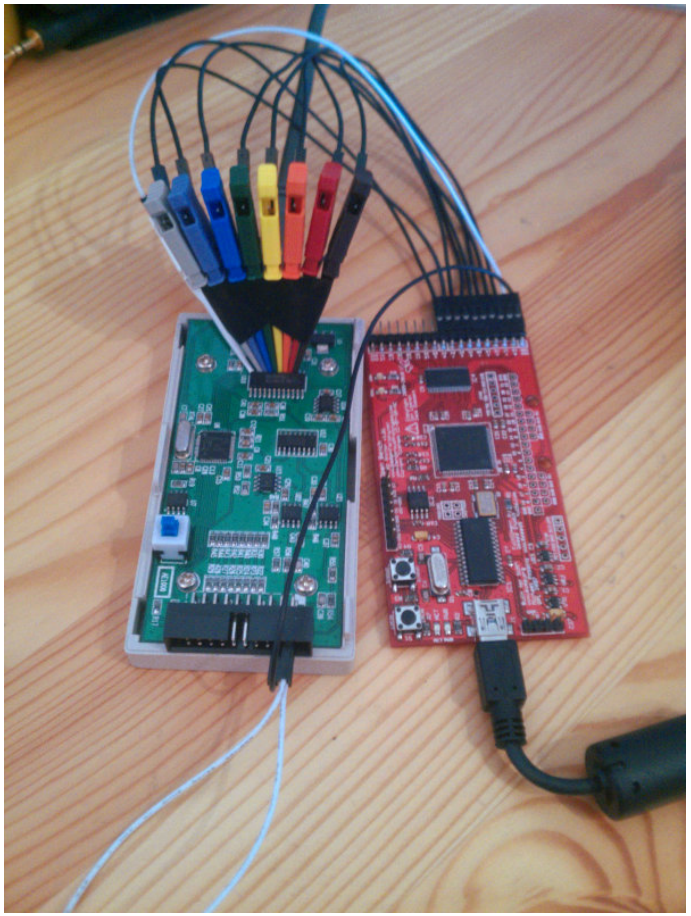
Other sigrok supported outputs

- VCD / Value Change Dump
- Analog (for DMMs and DSOs)
- Comma Separated Values
- GnuPlot
- Various vendor/device specific formats
  - OLS – Open Logic Sniffer Java client output
  - ChronoVu LA8

Logic Analyzer Visited

- Open Logic Sniffer (50$ USD)
  - Pros
    - Inexpensive + completely OSHW + OSS
    - 100mhz sampling
    - 200mhz DDR mode
    - H/W triggers
    - Run Length Encoding
    - External Clock + Trigger In/Out
  - Cons
    - Dead project + questionable design decisions
    - Only 24Kb sample buffer
    - High speed traces are useless without using RLE

Open Logic Sniffer



Logic Analyzers Visited Continued

- Saleae Logic (150$ USD, although clones can be found much cheaper)
  - Cypress FX2 microcontroller which has open source third-party firmware (sigrok-firmware-fx2lafw)
- Saleae Logic 16 ($299 USD)
  - Pros
    - Sampling three channels @ 100 mhz, 6 @ 50mhz, 9 @ 32mhz, 16 @ 16mhz with recent firmware
  - Cons
    - S/W only triggers
    - No H/W buffer, data is piped over the USB 2.0 interface, and random timeouts can end trace early

Saleae Logic 16

Protocol Decoders

- Magically translates the dump into user grokable output
- All the common bus protocols supported
  - I2C/SMBUS
  - I2S
  - SPI
  - CAN
  - 1Wire
  - UART
  - USB
- Side by side decoding of multiple protocols
- Stackable protocol decoders
  - Saves the user from hand decoding underlying protocol layers, e.g., SPI -> AVR ISP, i2c -> Wii Nunchuck, 1wire-link -> 1wire-network

Protocol Decoder Continued

- Protocol decoding can be done in CLI or GUI
- Examples of useful CLI based protocol decoding
  - UART -> UART dump (NMEA 0183 output)
    - sigrok-cli –i mtk3339_8n1_9600.sr -P uart:tx=TX:baudrate=9600 -B uart=tx
  - i2c -> i2c-filter -> RTC DS1307
    - sigrok-cli –i rtc_ds1307.sr -P i2c:scl=0:sda=1,i2cfilter:address=0x68,ds1307
- Examples of useful GUI base protocol decoding
  - Multiple signals and protocols side by side
    - NMEA 0183 + Pulse Per Second
    - I2c GPIO Expander + GPIO status
    - SPI UART + UART data

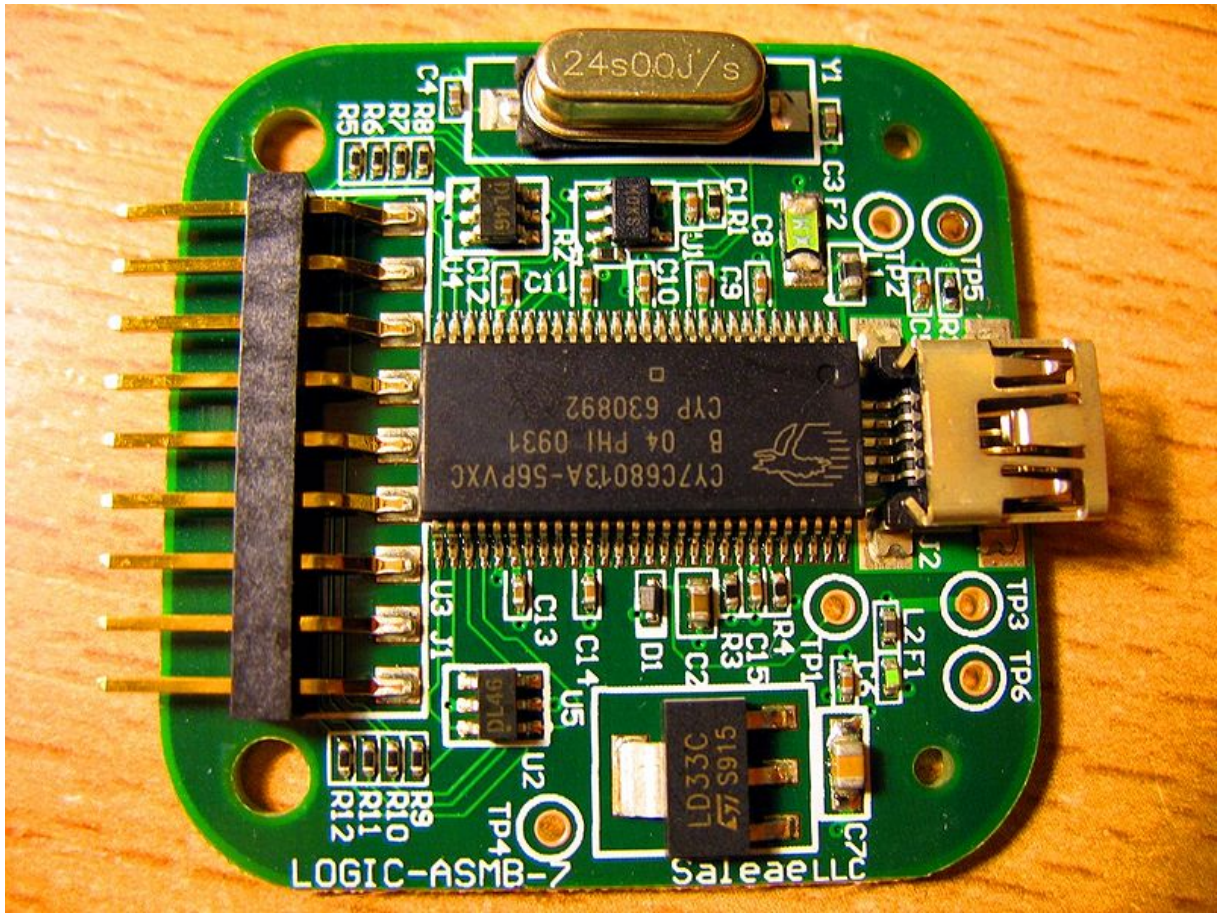
Creating Protocol Decoders

- sigrok protocol dumps are required for any new protocol decoder submissions
  - Split up commands into multiple dumps
  - Document everything that will likely be useful
- Decoders are written in Python so it's very easy to prototype and implement
- Annotations both in output and protocol forms
  - Allows stacking of PDs, for example i2c output into ds1307
- Decoder design should be scalable and extensible
- Most decoders are and should be written as state machines

FX2 Chipset Devices

- Open Source Source firmware available from sigrok project
  - Microcontroller firmware programmed every device initialization
- Various low end logic analyzers (aka clones of Logic) use this
- Good enough for 90% of your debugging
  - I2C + SPI + 1Wire busses
  - Once you require over the 24mhz sampling clock that probably requires going a little higher-end

Logic Breakdown + Cypress FX2 chip



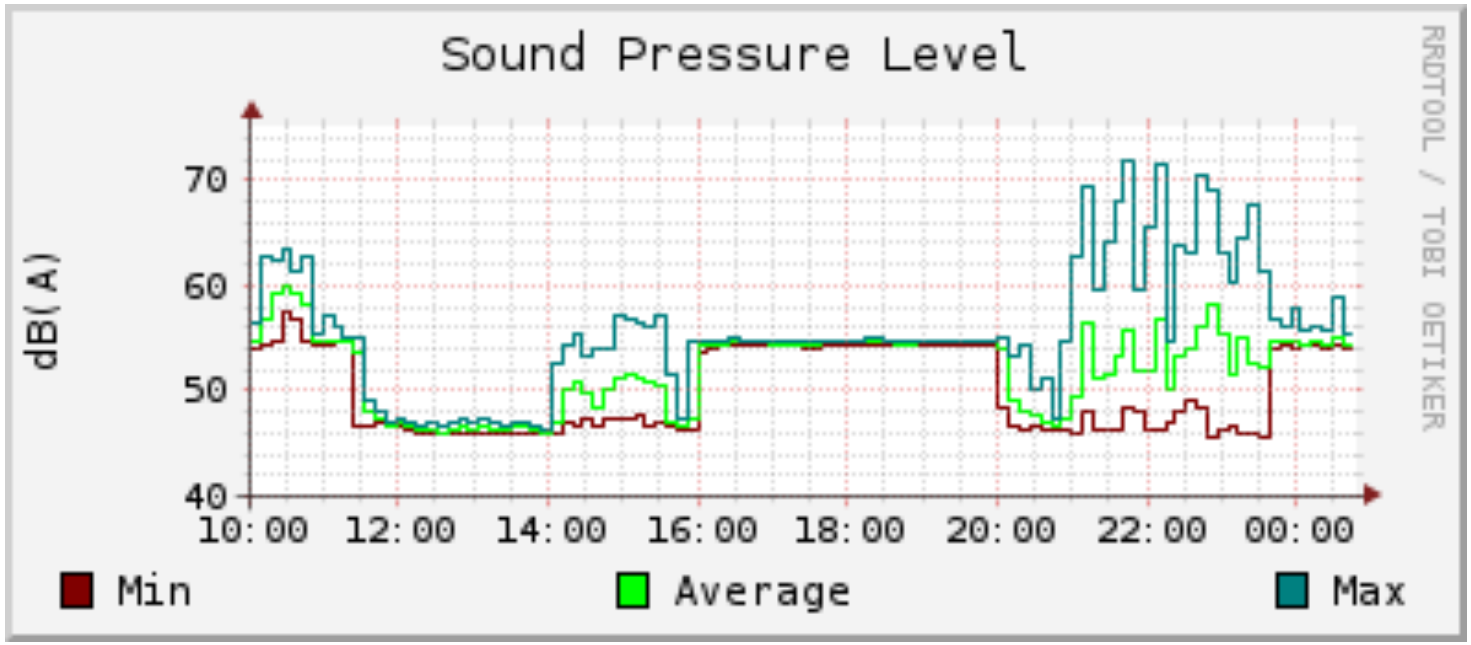
In Progress/Planned Features

- Advanced Triggers
  - Serial triggers
  - Multi stage triggers
- Software Triggers
  - Saleae Logic/Logic 16 (plus others) for example supports only SW triggers
  - Allow more dynamic options without hacking Verilog…
- Analog to Digital channel conversion
- Multiple devices per capture

sigrok Bindings + Plugins

- Python bindings (SWIG based)
- collectd plugin
  - Allows reporting of sigrok supported analog devices in an industry standard RRDtool format
  - Simply allows a circular database of data and creating useful graphs from them
  - Use cases would be using a sound level meter or in server room, or monitoring Carbon Monoxide levels near a furnace

collectd Example



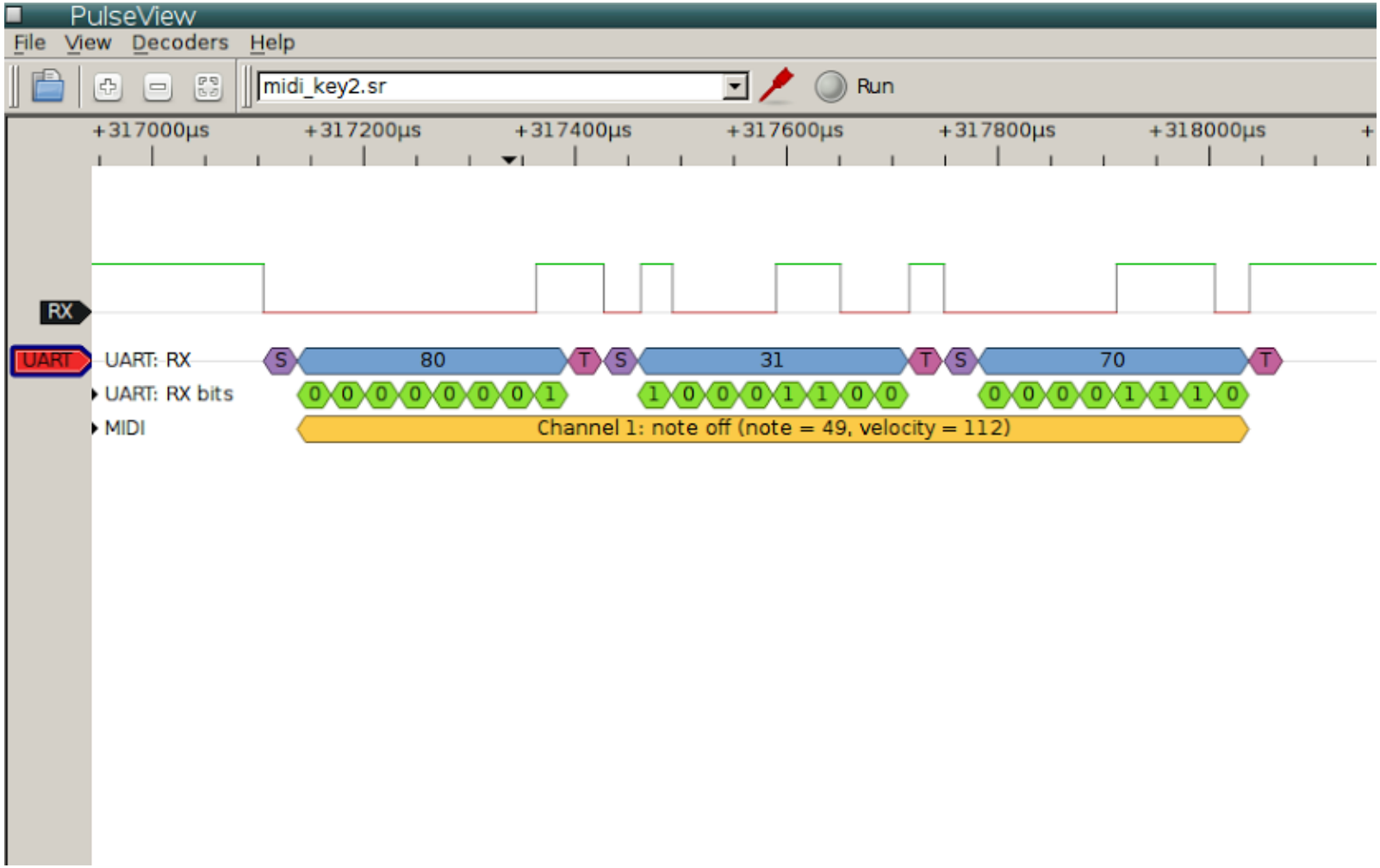
collectd Example Continued

```
LoadPlugin “sigrok”
<Plugin “sigrok”>
   LogLevel 3
   <Device “Sound level”>
         Driver “cem-dt-885x”
         conn “/dev/ttyUSB1”
         MinimumInterval 1
   </Device>
</Plugin>
```

PulseView

- First true graphic frontend for sigrok
  - Previous attempts really buggy and limited
- Allows running of traces and visual displaying of samples + protocol decoders
- Vital missing “killer app” for the sigrok project

PulseView continued



Logic Analyzer Hangups

- Continuous samples are a myth…
  - USB 3.0 may hold the hope for the future
- Run Length Encoding is a good solution but there are hold-ups
- Triggers type are important
- External clocking can be very useful
- Chaining with external triggers outs/ins

Weird Solutions

- Run Length Encoding hacks
  - Use more groups than you need
  - Waiting forever is possible…
- Simple microcontrollers/microprocessors as logic analyzers
  - Cypress FX2 + AM335x PRU
- Multiple devices chained or multiple channel groups for different logic levels

Demo

- sigrok-cli
  - Example of when using text based output is more useful than the UI interface
    - Audio playback of captured I2S data
  - Protocol Decoding
  - Tracing the ASCII way
- PulseView
  - Live capture + display of samples
    - Saleae Logic 16 tracing of data stream to WS2801 RGB LEDS string

Community

- Send us your protocol dumps in sigrok format
  - No protocol too odd or ancient for submission
  - Examples of ones we'd want are DMX512, MIDI, NMEA, CAN bus devices, and various I2C devices
  - First step to a protocol decoder… even if you aren't the one writing it
- Smoke out bugs and report any that are found
- Contribute to the wiki, post device breakdowns, protocol disassembly, etc
- Free hardware samples to support? (couldn't hurt to ask ☺)
- As always patches are welcome!

Questions?

- What are the features the users need most?
- Where is sigrok advanced or lacking compared to other OSS projects or even closed-source ones?

References + Links

- http://sigrok.org
- http://sigrok.org/wiki/Supported_hardware
- http://en.wikipedia.org/wiki/Sigrok
- http://dangerousprototypes.com/docs/Open_Bench_Logic_Sniffer
- http://www.saleae.com/logic16
- http://www.adafruit.com/datasheets/WS2801.pdf
- http://collectd.org/
- http://www.rigolna.com/products/digital-oscilloscopes/ds1000e/ds1052e/

Special Thanks

- Uwe Hermann
- Bert Vermeulen
- Joel Holdsworth