Open Source and Open Hardware

- Bruce Perens, Perens LLC
- bruce@perens.com
- +1-510-4PERENS

- One of the founders of the Open Source movement in software.
- Strategic Consultant for companies and government on issues of Open Source and Open Hardware.
- Expert witness on intellectual property cases.
- Currently building a wireless start-up.

Today we have an Open Source Track

- Last month I keynoted a law conference. This month SDR. Whatever field, it's always a sure thing that I'll meet incredibly bright and *motivated* people when I hang out with the ones involved in Open Source.
- Or maybe they're more interesting because they can actually *talk* about what they're doing, and show you everything about it without violating their NDA, and you can join their team if you can help.
- Attend the Open Source track today, and see for yourself.

My Background

- 19 years in computer graphics, 12 of them at Pixar.
- Credited on *Toy Story 2* and *A Bug's Life.*
- The tightly-held IPR side: Purchased a home with proceeds from Pixar' IPO, and of course their films.
- The widely-shared IPR side: Created the licensing structure of Open Source Software and Open Hardware. Created *Busybox,* a key component of embedded Linux.

Why Open Source?

- We just want to make things. Haven't you ever wished your company would *just **let** you make things?* Isn't it a business failure that they don't?
- Intellectual property is a pain! It gets in the way of engineering and business.
- But it's how we make money, right?
- Sometimes. But intellectual property, even our *own* intellectual property, is a *blocker* for our business at least as often as it is an asset.

There's Another Way

- Open Source demonstrates the failure of conventional intellectual property strategy. Open Sourcers produce great technical products that companies could not capitalize or develop successfully using the conventional paradigm.

Open Source in Your Backyard

- GNURadio and PowerSDR
- USRP, FlexRadio SDR platforms.
- OpenBTS, the Open Source GSM stack.
- OSLEC: Open Source Line Echo Canceler
- Mesh Potato
- Speex, Codec2, FLAC, Ogg (Vorbis, Theora), WebM and WebP.
- Arduino, Bus Pirate, Beagleboard, Pandaboard
- Linux as the OS platform. Algorithms a la carte.

And they Profit, Too

- For the most part, the people who are working on Open Source software today do so *on company time,* as part of their job.
- Their companies *profit* by sharing what might otherwise be considered their intellectual property.
- Later in this talk, I'll show you how.

Start with the Basics

- Open Source software licenses grant the right to use, modify, and redistribute software, royalty-free.
- Made by a diverse mix of individuals, business, academia.
- Largest developer on the Linux kernel is “Not Affiliated”. This is one of the most complex products in software, and put to mission-critical use everywhere.

Where do You Come In?

- Feel free to *use* Open Source software. All companies of any significance do.
- Leverage Open Source in your products *only* if you *perform due diligence on compliance.*
- Participate in Open Source development where it makes sense for your company.

An Open Source Radio Story

- In the 2000's, radio hams started using a commercial digital voice system called D*STAR.
- D*STAR is unique to ham radio, but it does many of the same things as APCO P25, the federal-government-mandated system for digital walkie-talkies used by police, firemen, etc.
- Most importantly, D*STAR and P25 both use an AMBE digital voice codec from Digital Voice Systems Inc. D*STAR uses AMBE+
- DVSI's AMBE codecs are *highly* proprietary.

Ham Radio's Open Tradition

- Hams are the original Open Sourcers.
- Their tradition, since the days when spark-gaps powered transmitters, has been to *understand how your radio works.*
- Their books and magazines are full of schematics.
- Real men *build* their radios and antennas.
- The sharing of innovation that their community fostered drove much of the development of radio.

Modern Hams

- Ham radio is experiencing a renissance, in part due to the removal of their Morse-code test, and in part due to exciting new technology.
- There are 700,000 U.S. Hams today, more than at any time in history!
- **Software is an enabler** for today's hams, many of them are SDR pioneers.Today *Real Women* write their radio modems in C.
- ***Unless*** it's software that is closed away from them, software that they can't modify.

A Really Bad Fit For The Purpose

- So, here were the hams with AMBE+, a black-box, trade-secret, patented, DMCA-encumbered piece of intellectual property in charge of encoding their voice transmissions.
- AMBE+ is mostly sold as a pre-programmed DSP chip with the “read” fuse blown, and even the radio *manufacturers* don't learn how it works.

But The Hams Decided to Use AMBE+ Anyway Because:

- Because their boss told them to.
- Because “that decision is above my pay grade”.
- Because legal required it.
- **Because they *respect* intellectual property!**

None of the above.

Religion

- “Respect of intellectual property” is a religious argument.
- It keeps you from thinking about, or understanding, what the best intellectual property strategy for your business (or non-business) really might be.
- In this case, the “customer requirement” of the hams was to ***stay in control.*** To be able to understand, implement, copy, and improve; without secrets, fees, royalties, patents, without intellectual property that would *remove their control* of their technology.

Pragmatism

- But we would have to keep using AMBE+ because codecs are rocket science, algorithmic magic that cost millions of dollars and are developed by rare geniuses.
- Besides, codec technology is so encumbered in patents that even if we tried to make a free one, we'd just end up in patent court.
- And we could never find someone who would help us write one for free.

Wrong.

Reality

We did it all. We now have a digital voice codec called **Codec2** that:

- Delivers telephone-quallity voice at only 2200 bits per second, as good as AMBE+.
- **Has no secrets!** No magic necessary, mostly fundamental DSP. All Open Source, read our algorithms, please!
- Has been engineered to use expired patents and algorithms that are in the public domain or original to our authors.

A Business Opportunity

- Codec2 is available to build into your next wireless product, without royalties, under easy “BSD-style” licensing that doesn't encumber *your* intellectual property.
- It can provide a $15 per unit cost reduction as well as significant technical improvements to any wireless digital voice product that doesn't need AMBE for interoperability.
- Obviously, we're going to see this go *really* far outside of amateur radio.

Prior Art Summary

- Sinusoidal Coding, Mcaulay & Quatieri, 1984
- Linear Predictive Coding, Makhoul, 1975
- Line Spectrum Pairs, Itakura, 1975
- MBE Voicing, Griffin & Lim, 1988
- Overlap Add, Tribolet & Crochiere, 1979
- NLP Pitch Estimation, Rowe, 1999
- LPC Amplitude Recovery (algorithm used here), Rowe, 1991, 1999, 2009
- Post Filter, Rowe, 2009

Pile On

- Part of the patent strategy is simply to pile on: many Radio Amateurs are attorneys, and we have a large, well-practiced national pro-bono law force as a result of the constant neighborhood challenges to the presence of our antennas on homes. Some of those attorneys are intellectual property specialists. We can, if necessary, mobilize a larger legal staff than most companies.
- Most patent challenges are not directly against the Open Source project with the intent to deter them. There is little to gain from directly suing

A Call to APCO

- P25 is an open standard. As a requirement for federal funding, it must be open enough for all manufacturers to build.
- When P25 was designed, there weren't any open codecs available of sufficient quality.
- Having a government-required standard depend on a single manufacturer's black-box technology was acceptable as long as it was the only possible solution. *That's not the case any longer.* Start following this development.

How Do Companies Participate in Open Source *AND* Profit?

- The key is understanding the different roles of software in your company.
- Some software provides ***business differentiation.*** It makes your product look better, directly to the customer, in some way.
- But most software doesn't differentiate your company at all. You need it to get the job done, and it might give you an efficiency advantage, but it doesn't make the sale to the customer.

Amazon Example

- Amazon had a recommendation system before other book-sellers had it. It close to *doubled* their business by recommending purchases related to what you were looking for.
- Obviously, Amazon couldn't Open Source their recommendation system at that time. If other companies used it, they'd lose their business differentiator.
- But now, every internet store has a recommendation system, and that no longer has differentiating value.

Differentiating vs. Non-Differentiating

- At the time that the recommender was business-differentiating software for them, Amazon made use of Linux, Apache, Perl, Mason, and other Open Source software.
- Amazon shared its work on this *infrastructure* software with other companies, while holding its differentiating software close – *not* sharing that.
- Amazon could tell Barnes and Noble everything that it knew about Apache and Linux without hurting their business.

Distributing Cost and Risk

- By sharing its work on Linux and Apache with other companies, Amazon could distribute the cost and risk of those projects, and they didn't have to sustain the entire expense.
- They could then move software development dollars into the software development that actually differentiated their business.
- And so they were able to invest more time and money into the things that made them a profit.

Intellectual Property Policy Must Be Different for Different Software

So, we learn some economic lessons from this:

- Some software is essential, but it isn't what makes you money.
- Some software has a disproportionately large effect on your bottom line.
- You have to pay to develop both kinds of software.
- You get the best use of development time and dollars if the intellectual property policy for the two kinds is entirely different.

Two Policies

- Infrastructure, enabling technology, “back office”, operating systems and servers are probably not business differentiating.
- Distribute the cost and risk of their development across multiple companies, sharing all intellectual property with them and the public.
- Customer-visible special features of your product must be developed under your control, and held close, lest you lose the business differentiation.

Depreciation

- Remember that Amazon's recommendation system *was* a business differentiator. It isn't one any longer. They still need it, but it doesn't give them the unique advantage it once did.
- The value of that software has depreciated tremendously.
- What does this say about the value of *your* software that isn't business differentiating?
- It probably isn't worth what you spent to develop it.

Apply to Wireless

So, let's apply this to the wireless industry. If you've worked for a cellular handset company, you might have noticed that there's a “holy of holies” among their intellectual property. Something of tremendous value, which they must hold close at all costs.

At least they think so. Because it cost a *ton* of money to develop. It must still be worth a lot, right?

What is it?

It's

- **The GSM stack!**
- I've met more than one company that treats it like gold.
- But 10 other companies have one too, and the Open Source community has its own GSM stack now. They've even tested it with a town's worth of live phones, with FCC authorization and .
- And when have you heard a proud new smartphone owner bragging about how sexy the device's GSM stack it?

Not Business Differentiating

- If 10 other companies have something, and the Open Source community has one too, you can bet it's not business-differentiating.
- You might as well start sharing its development. You're not saving anything by holding it close, and with the advent of an Open Source version, you're not deterring any potential competitors.
- It's not worth what it cost to develop any longer.

OS Kernels Don't Differentiate

- Blackberry came out with a tablet with the “QNX Neutrino” ultra-reliable OS, but without the applications enjoyed by Apple or Android. The units stagnated on retailer shelves until discounted from $500 to $200, probably below cost.
- Symbian spent Millions to implement IPV6 on their kernel, at a time when the proper strategy would have been to abandon it. They couldn't even get anyone interested after they Open Sourced it.
- Solaris didn't sell enough hardware to keep Sun

Economic Lesson

- Use Linux or BSD, share the development of features you need with other companies, and save your development dollars for things the customer sees.
- If your company uses cost of development to determine the cash value of its own software, their figures are unrealistic. If Open Source makes the same thing, and does it well, the value of your product may yet approach zero.

Introducing Open Hardware

- Designs of physical devices or electronics that are shared as if they are Open Source software.
- Large and active community.

AMSAT

- The largest private space program, operating since 1963. Many vehicles launched as hitchhikers.
- Rockets lift fixed payloads, secondary satellites are better than ballast.
- AMSAT's problem is international cooperation in the face of ITAR 121.

ITAR 121

- Zimmerman Cryptography export case - “PGP”.
- Phil Karn KA9Q (senior scientist, Qualcomm) sues U.S. Government, gains “public domain” exception to ITAR 121.
- Used by other Open Source groups such as “DIY Drones”. Project includes Wired Magazine senior editor Chris Anderson. Makes an Open Source UAV platform for model airplanes and helicopters.

Arduino

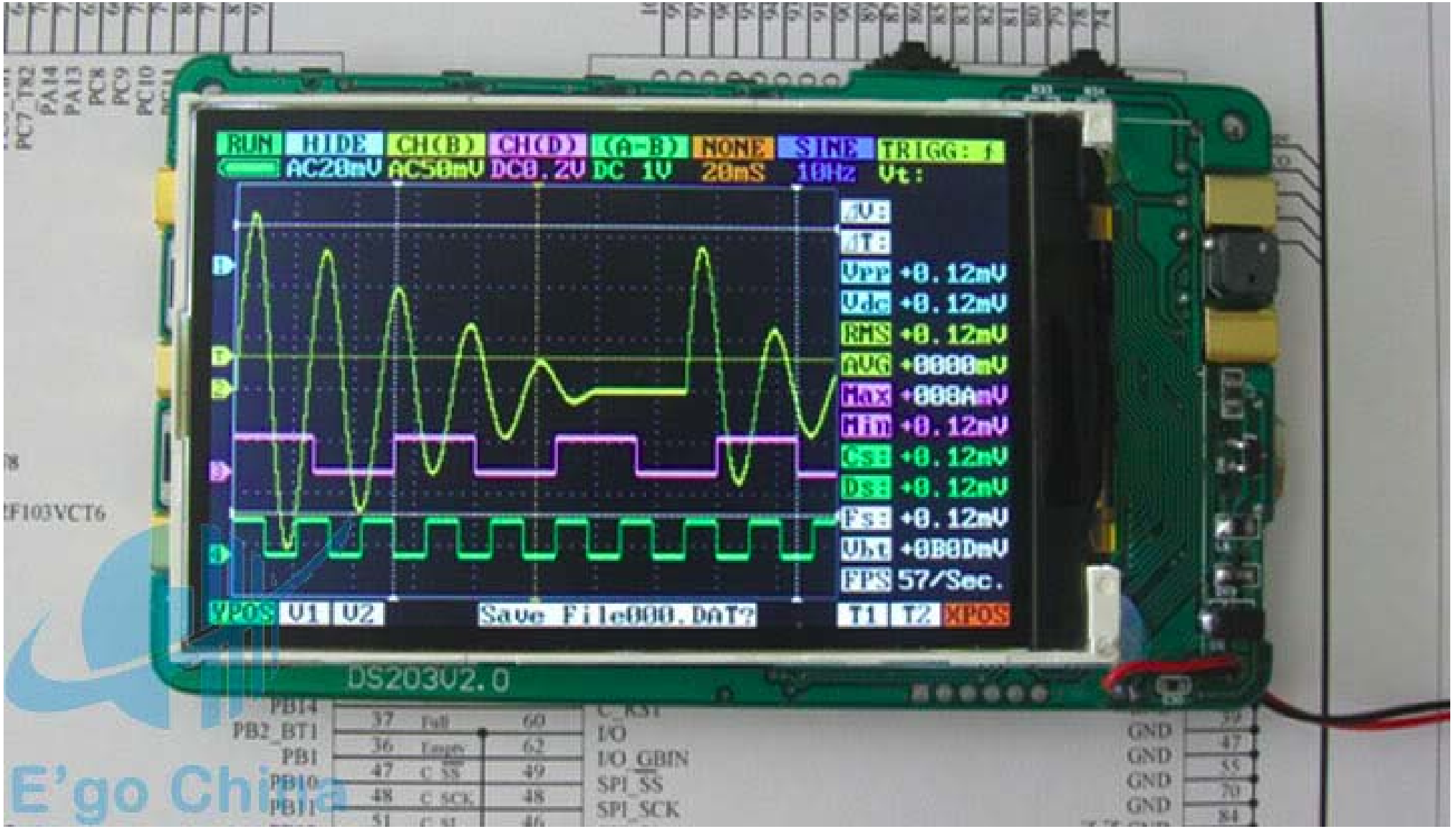
- Most used device of the Open-Source Hardware movement for now.
- General-purpose embedded CPU.
- Easier to program than PIC, more powerful than Basic Stamp.
- Standard form for

Why Arduino Instead of PIC, Stamp, etc.

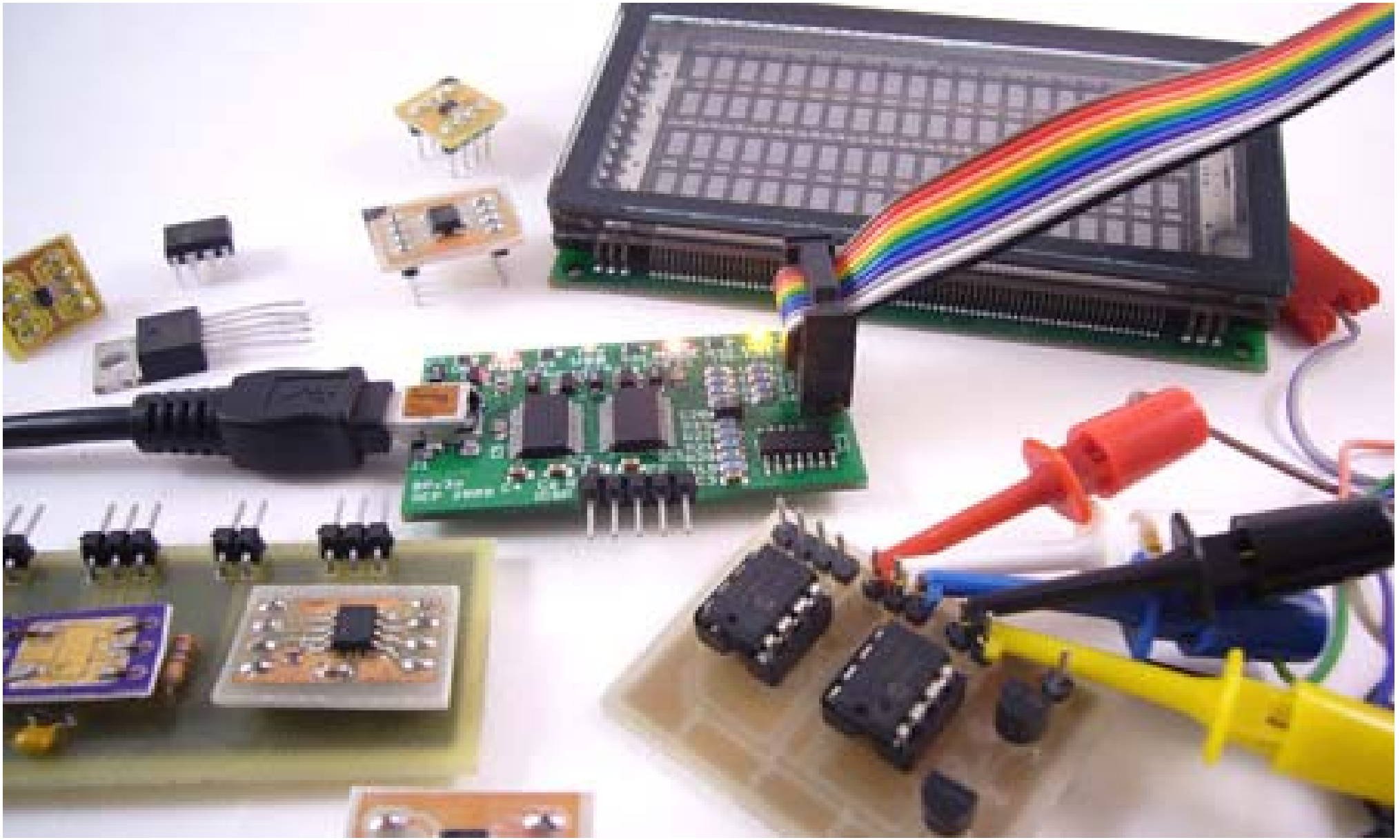
- Community, community, community.
- 100% Open Source including hardware design.
- Google selects Arduino-based USB interface platform as standard Android plug-in accessory. Using Apple's closed-ness against them.
- Use AVR CPU in production units, running same code as Arduino but without Arduino card. Approaches price for PIC, beats Basic Stamp.

Some Open Hardware Products

DSO Quad


RUN HIDE CH(B) CH(D) (A-B) NONE SINE TRIGG: f
AC20mV AC50mV DC8.2V DC 1V 20mS 10Hz Vt:
ΔV:
ΔT:
Vpp +0.12mV
Vdc +0.12mV
RMS +0.12mV
AVG +0000mV
MAX +008AmV
MIN +0.12mV
CSR +0.12mV
DSR +0.12mV
FSR +0.12mV
Vht +0B0DmV
FPS 57/Sec.
YPOS V1 V2
Save File000.DAT?
T1 T2 XPOS
DS203V2.0

Bus Pirate



Bus Pirate

- The Bus Pirate is an open source hacker multi-tool that talks to electronic stuff. It's got a bunch of features an intrepid hacker might need to prototype their next project.
- General-purpose 0-5.5V interface.
- $30 assembled.

Bus Pirate Features

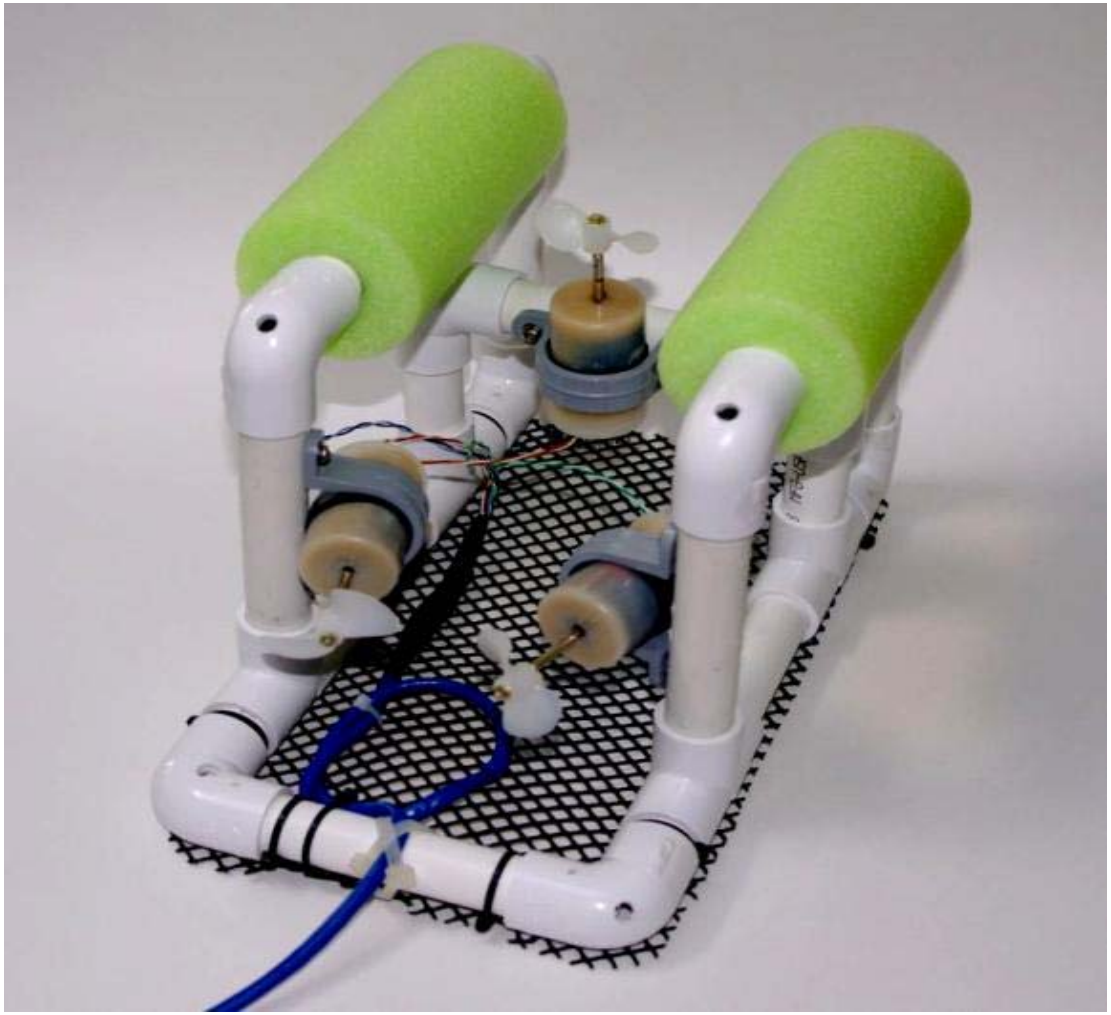
- 0-5.5volt tolerant pins
- 0-6volt measurement probe
- 1Hz-40MHz frequency measurement
- 1kHz - 4MHz pulse-width modulator, frequency generator
- On-board multi-voltage pull-up resistors
- On-board 3.3volt and 5volt power supplies with software reset
- Macros for common operations
- Bus traffic sniffers (SPI, I2C)

Bus Pirate Protocols

- 1-Wire
- I2C
- SPI
- JTAG
- Asynchronous serial
- MIDI
- PC keyboard
- HD44780 LCD
- 2- and 3-wire libraries with bitwise pin control

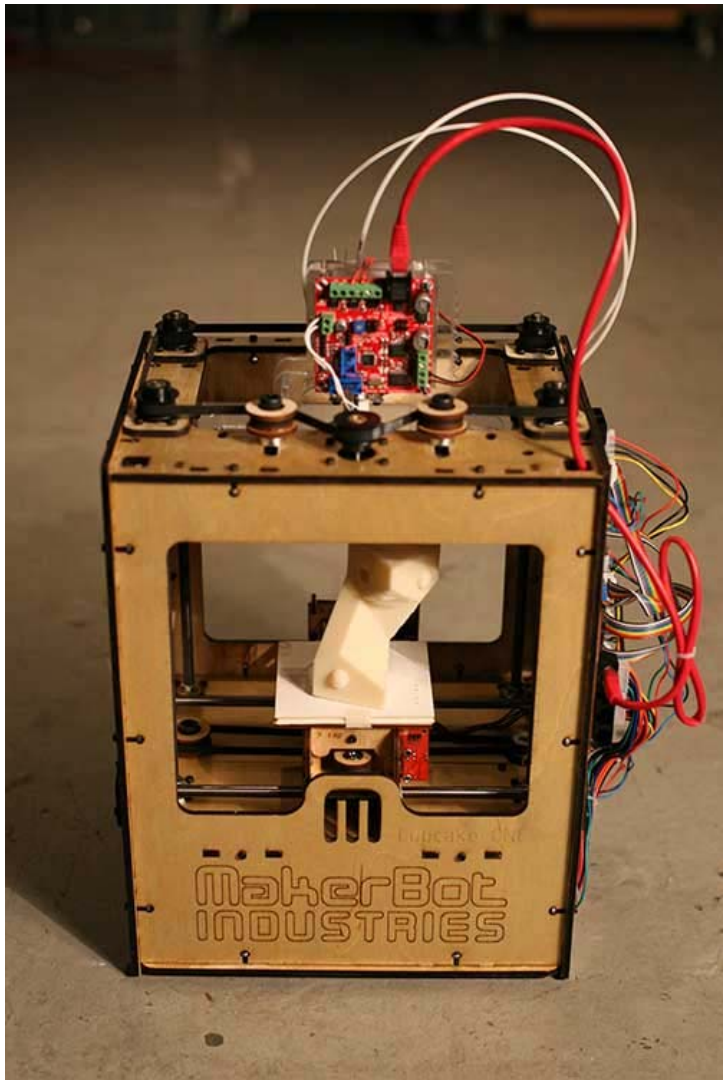
Sea Perch

- Main purpose is education.
- Remotely operated submarine built from PVC pipe, pill bottles, wax, pipe insulation, electronic parts from Digikey.



MakerBot

- 3D-printer. Makes small thermoplastic objects from your computer models.
- Has not reached the critical milestone: resolution is too low to make good Lego-compatible parts. The studs don't hold together like real Lego. When we get there...
- About $750.
- All Open Source hardware and software.



This is One Piece

- Can manufacture objects that would be physically impossible to fabricate in one piece using conventional manufacturing methods.

Can You Make a Makerbot With A Makerbot?

- You can print *some* of the mechanical parts to make another printer, using your printer.
- You can't come *close* to making the whole thing that way. Any claims otherwise are hyperbole.
- Output is jaggy like old computer graphics, and would require sanding of the entire surface to use components as radio cases or knobs. But fine for prototypes if you put in the work.
- 3D model can be sent to a mold-maker who will use CNC milling machine.

OpenPCR

- Thermal cycler for the Polymerase Chain Reaction.
- $600.
- Open Hardware.
- Genetic Engineering at Home!
- Other tools, such as electrophoresis, are well within reach of hobbyist.

Open Hardware Journal

- A new Open-Access journal of the Open Hardware community.
- Free to read, copy, redistribute.
- First issue on November 1, 2011
- http://OpenHardware.org/journal/

Why are Smartphones so smart, and HTs so dumb?

If the HT's software is so much simpler than the smartphone, Why is the HT so difficult to use, compared to the smartphone?

Potential HT Design Elements

RF section from USRP WBX-2 or upcoming release from Ettus which will operate DC-4 GHz.

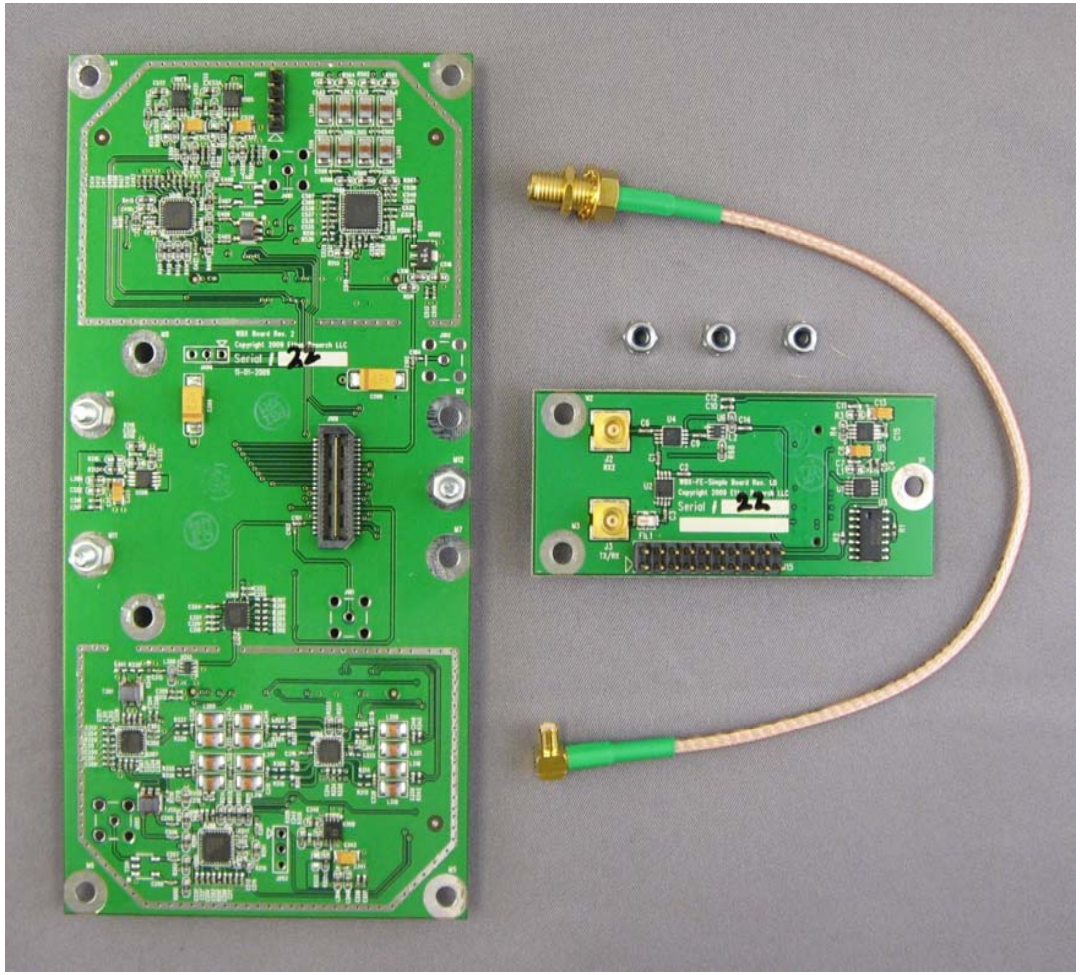
ADC/DAC.

Programmable gate-array for specialized processing.

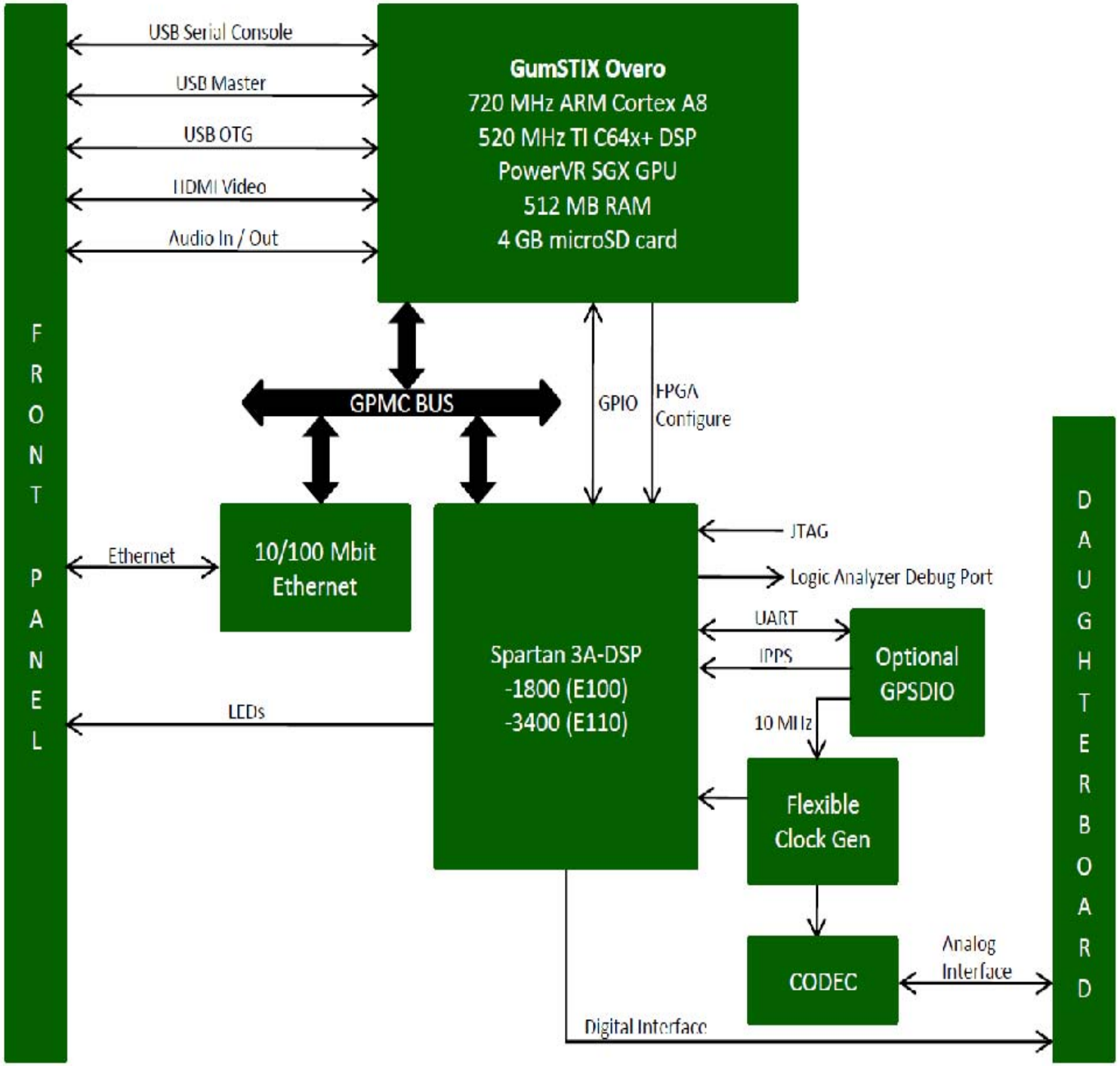
Embedded Linux computer and touch-screen display.

USRP WBX-2

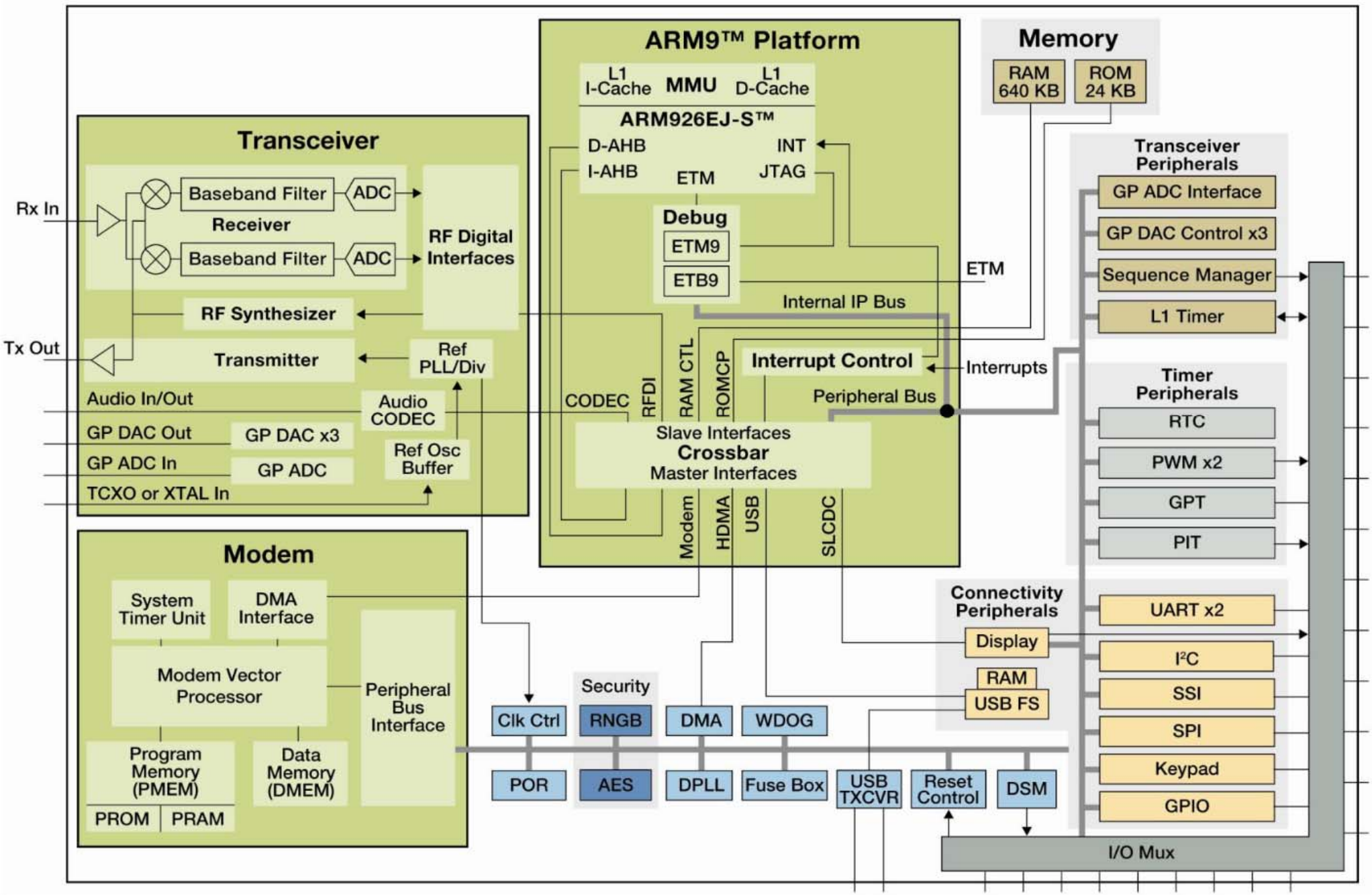
- 50 MHz to 2.2 GHz QRP transceiver, feeds to ADC and DAC on USRP board.
- Not very many parts, are there?
- Most of the signal processing work is done in the computer.



- USRP E100 motherboard block diagram.
- Two 64 MS/s 12-bit ADCs
- Two 128 MS/s 14-bit DACs
- Gate-array for hardware implementation of compute-intensive algorithms.
- On-board computer running Linux.



MC13260 System-on-Chip Two-Way Radio



Contact the Speaker

Bruce Perens

bruce@perens.com

+1 510-4PERENS