

The FSM Virtual Radio Kernel

Why, What, & How
(In That Order)

Frank Brickle, AB2KT
ARRL/TAPR DCC 2007

Copyright © 2007 by Frank Brickle
Creative Commons License ATTR/NC/ND

The Horseless Radio

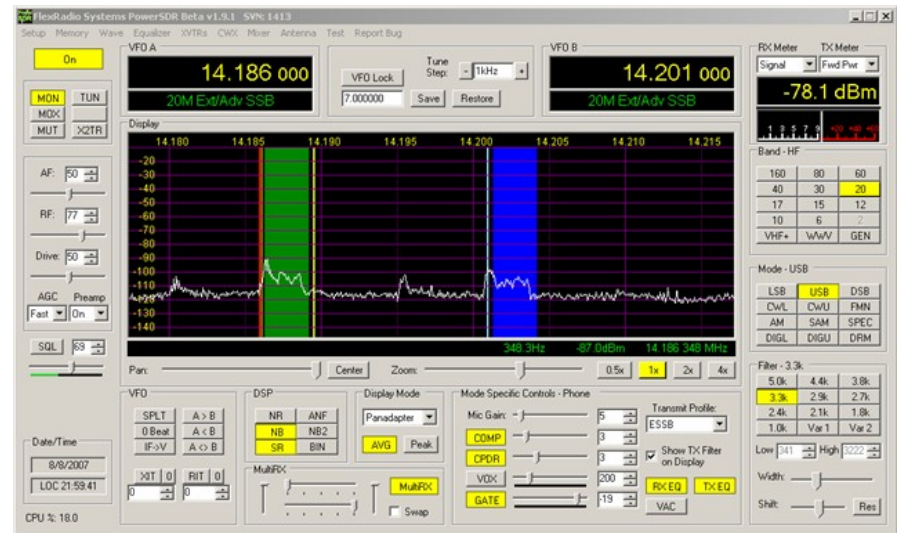
- SDR has reached maturity...*in some areas*
- SDR DSP is primarily a substitute for conventional components...*so far*
- SDR UI offers incremental improvements over a conventional interface...*as yet*
- C/SDR can do much, much more...*soon*
- Infrastructure for “more” is substantially in place...***already***

Roger's *D*ream *S*tation

1. Multiple SDRS (hw/sw both)
2. Simultaneously display consoles for all
3. Simultaneously view panadapters for all subRX
4. Follow UI focus for audio, PTT
5. Mute/unmute, pan output audio from each
6. Multiple control devices (footpedals, etc.)

The Hard Way

The PowerSDR Console...
a **good** example,
but you still don't
want to write
Roger's Dream
Station this way!



The Answer to Your Prayers!

- RDS isn't a single app, it's many smaller ones working together
- *VR-SDR* is the whole system
- *VR-Kernel* is the brain
- **FSM** is our *VR-Kernel*
- **GBG** is our *VR-SDR*
- Today we're talking mostly about **FSM**



What Makes Him So Smart?

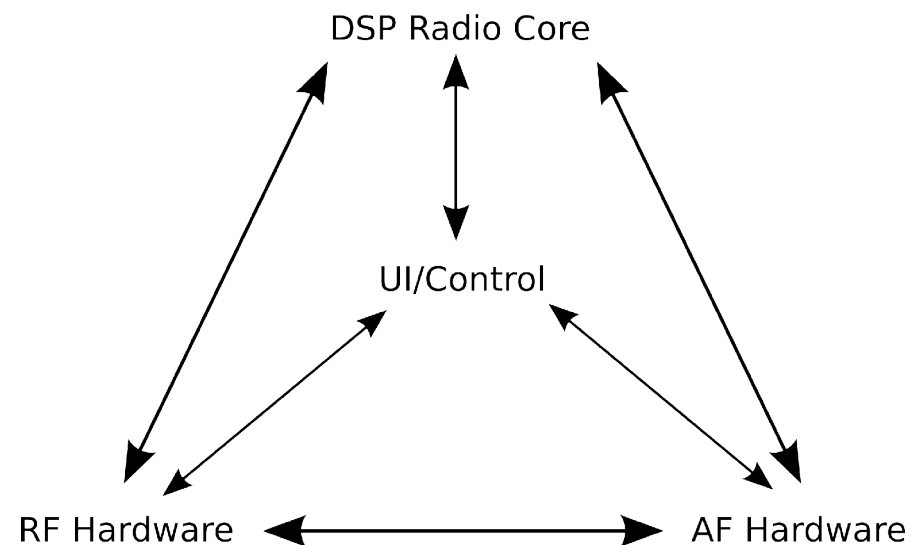
- Not the first time we've been here
- Computer music was in the same situation a generation ago
- There's still blood on the walls
- PLOrk (<http://plork.cs.princeton.edu>)
- Chuck (<http://chuck.cs.princeton.edu>)
- The principle is already proved

Moving from Glass to VR

1. Refactor SDR applications
2. Build operating environment using Erlang/OTP
3. Create new visuals based on 3D graphics support

1. Refactor SDR Apps

- Right now, tightly coupled and monolithic
- Mostly just a collection of APIs
- Hard-limited assumptions about sources and sinks
- Code-bound



The Remedy

- Complete decoupling of all pieces
- Needs *referential transparency* and *lexical scoping*
- That is, AFAP, many standalone programs
- Gain: recursive definition (nesting), hot-swapping, multi-language, multi-platform (distributed too)
- Dynamic control at user level or by proxy

Some Woodshedding

- DttSP/sdr-core built from the ground up with decoupling in mind
- *nix-based OSes have an easier time
- Windows will always be the odd one out

Hiding in Plain Sight

- “Cognitive Radio” needs heavy computational infrastructure
- Much will never be seen directly by users
- Large body of existing code for heterogeneous platforms
- Nobody wants to rewrite all of it
- Can be leveraged easily *as-is* in the VR-SDR environment

[What is Cognitive Radio?]

- A bad name for a good idea
- An *Intelligent Agent* whose primary sensors are for RF
- IA: an automaton to carry out different sequences of actions based on data from sensors
- Two possible actions: (1) create new sequences (2) encapsulate old ones
- GBG/FSM enable this (uniquely?)

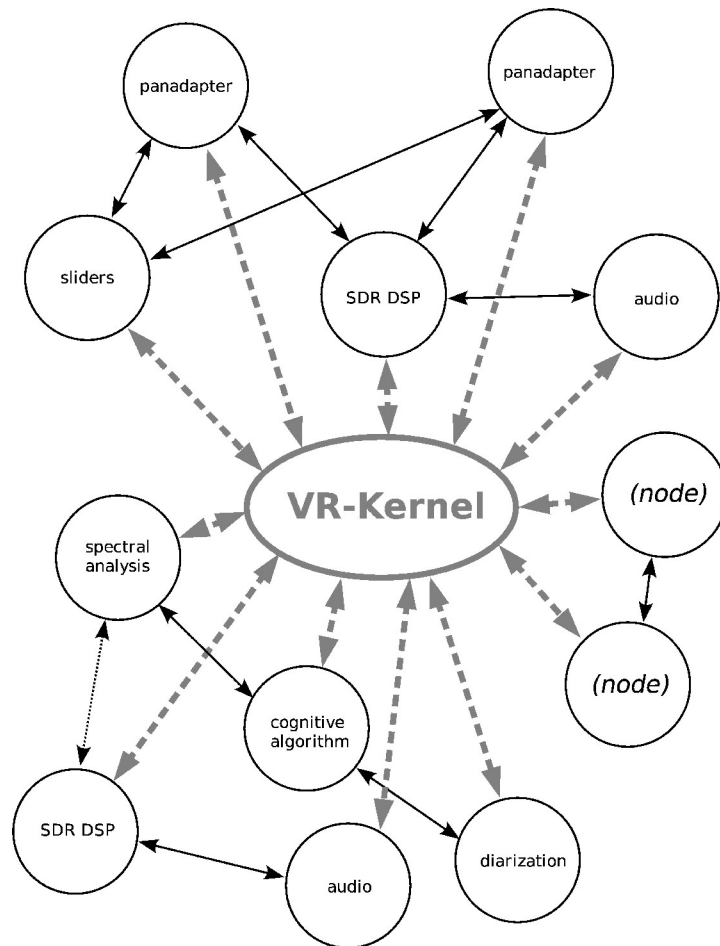
Learning by Doing

- VR-SDR is more like an OS than an application
- Based heavily on *syntax* (rules) and *constraints*, therefore
- *Bottom-up* and not *Top-down*
- *Protocol* not *API*
- Important: Anyone Can Play
- Thus, **GBG**

The Non-Negotiable Clause

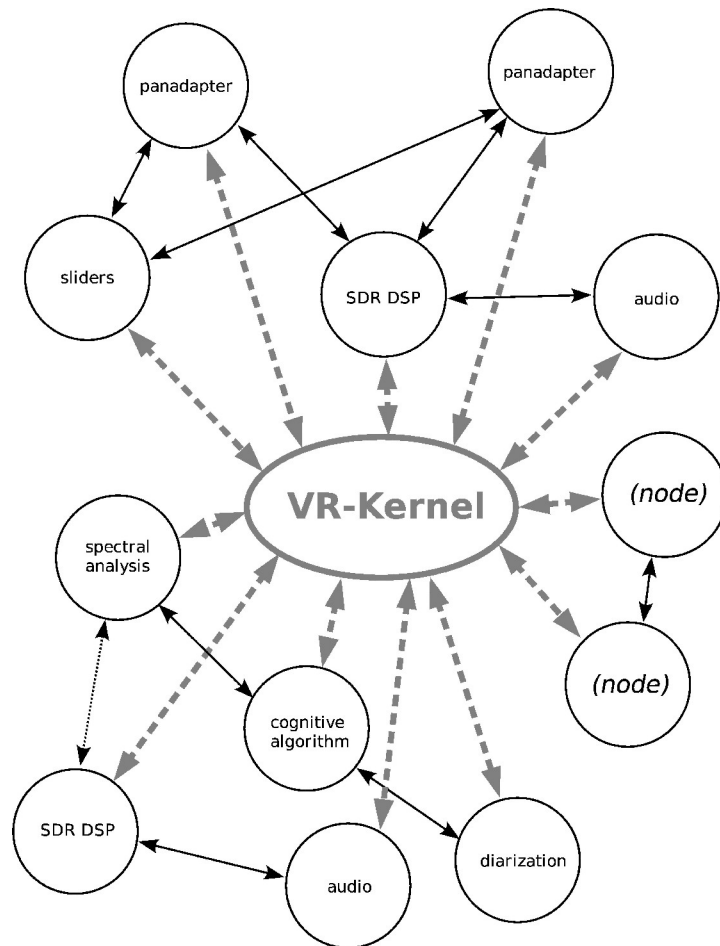
*Free** and *Open*
Source

2. The Engine



- A picture is worth a billion bits...

Not a Classless Society



- Two kinds of nodes
- Solid black are functional radio pieces
- Grey elliptical thing is the VR-Kernel
- VR-Kernel also a node, but with special privileges and responsibilities

VR-Kernel Specialties

- Only VR-Kernel can *create, control, and destroy* other nodes
- VR-Kernel tells new nodes about their neighbors (creates dynamic context)
- *All control* passes through it
- *Data* can pass through it (but can go directly too)
- [*At least 3-connected nodes on a torus*]

VR-Kernel Responsibilities

- Constrains possible system states, known or unknown [NB enumeration impossible]
- Known states: behaviors/personalities
- Unknown states: errors **or** innovations
- Parameter validation
- Monitoring and repair
- Persistency

Ontological Relativity

- Nodes and edges can be realized in different ways
- Nodes and edges can be hot-swapped
- Nodes can be distributed transparently
- Only VR-Kernel needs to know *where* nodes are running

2. Erlang/OTP

- Erlang == “Agner Krarup Erlang” & “Ericsson Language”
- OTP == Open Telecommunications Platform
- Complete implementation of full infrastructure for VR-SDR
- Robust, fault tolerant
- GPL, active + heavy development
- Fully interoperating + cross-platform

Development Strategy

- Entire system is hot-swappable
- Multiple versions can coexist
- *Therefore* system can evolve but keep backward compatibility
- Start with simple simulation of radio personalities
- Symbolic representation in XML

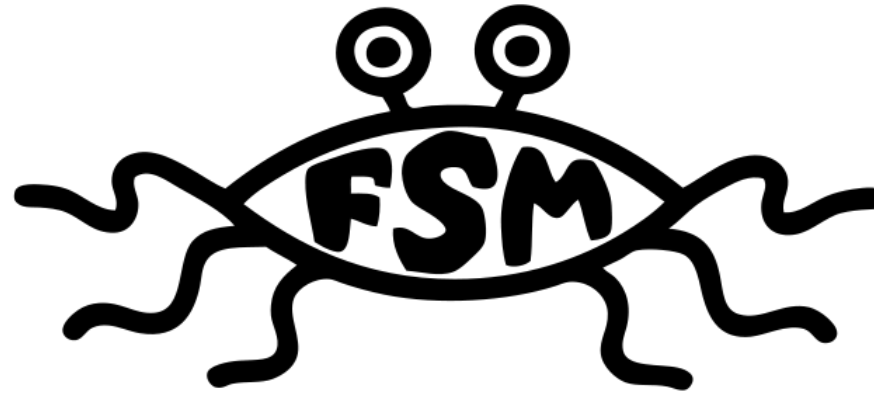
The Story Till Now:

- Everything behind the scenes in Roger's Dream Station, **plus**
- Distributed *and* multi-platform
- Robust
- Free
- Other technically necessary or appealing qualities...

We Are Not Alone

- Bob Cowdery G3UKB: Erlink-SR
- gnuradio
- CORBA/SCA
- heterogeneous hardware *and* software

The Truth Is Out There



- Note the uncanny resemblance...*

* The Truth Is...

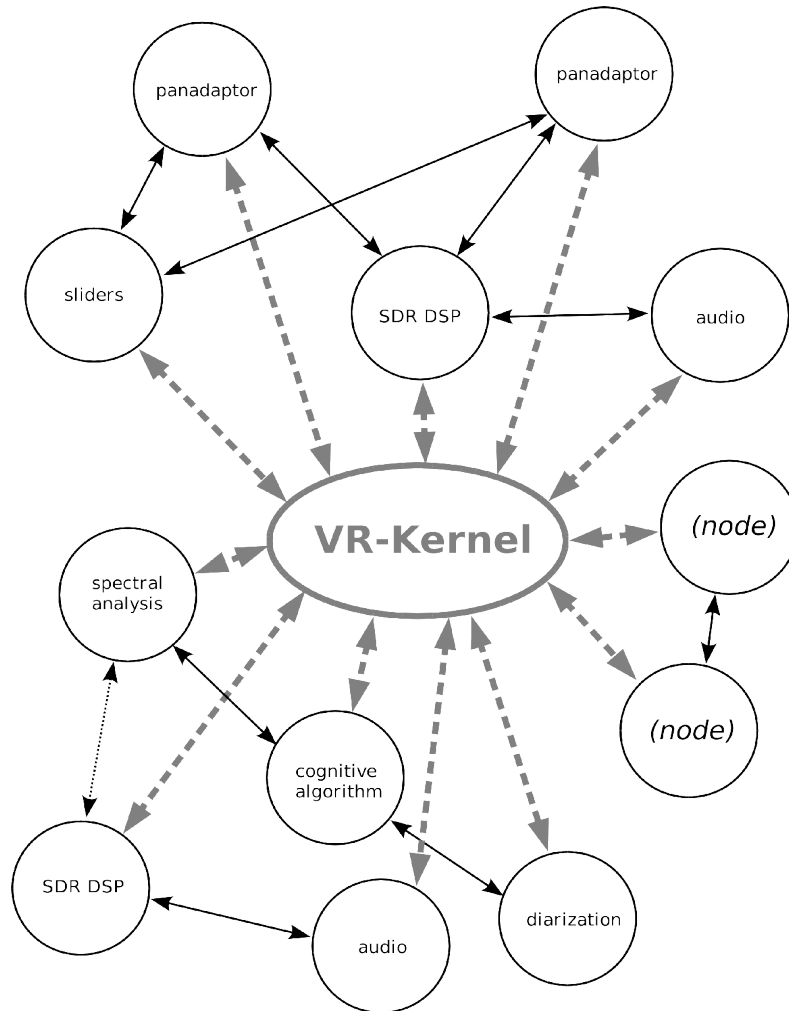


Fig. 2. A typical VR-SDR configuration.

- It's not a Finite State Machine at all, except locally

3. UI/Display in 3D

- Andy Moorer: “How do you do computer music without a lot of graphics?”
- In part, substitute rich graphics for tactile feedback
- Bitmapped display + mouse are not enough
- Needs (effectively) multiple visual dimensions and multiple parallel control streams from diverse devices

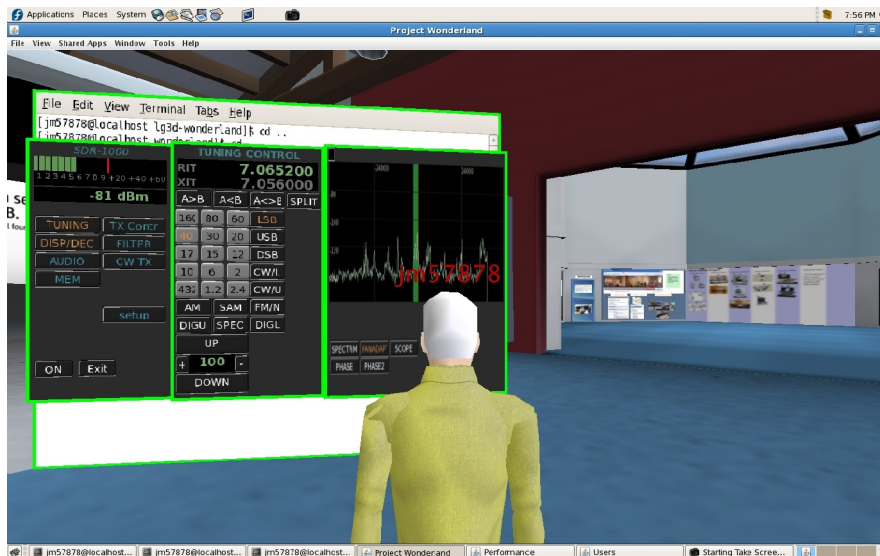
The Five Essentials

1. Information and control in 3D
2. Everything visible subject to user or program control
3. Everything visible can be transformed continuously and RT
4. 3D locations are in unified space
5. Mobility in *time* (not continuous)

Is This Some Kind of Game?

- Yes.
- However, a Compositing Window Manager does most of the work.

Proof of (Second) Life

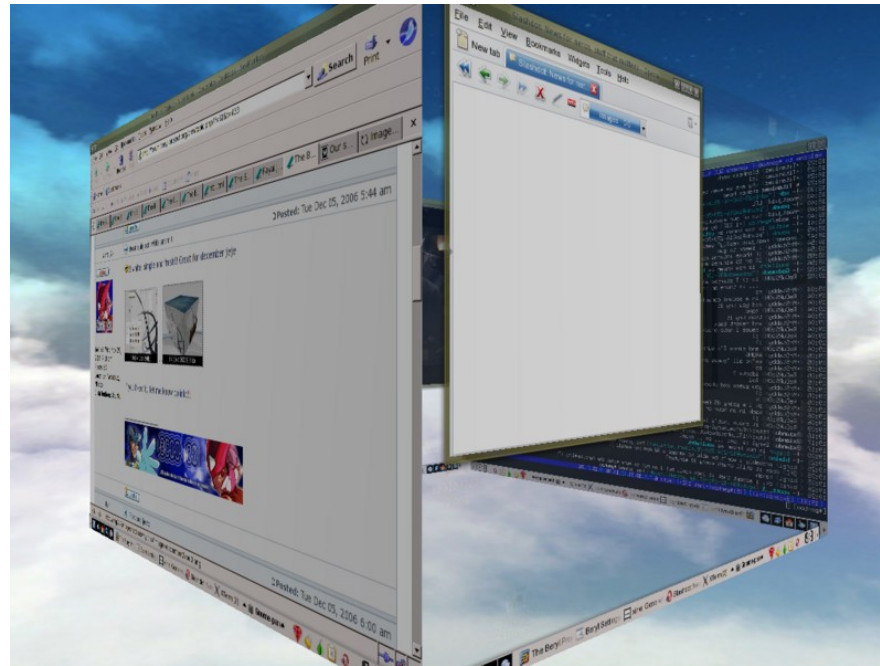


- *jsdr* by John Melton, G0ORX/N6LYT
- Embedded in *MPK20* by Sun
- Collaborative virtual environment
- Environment, not application

Operating Environment

- 3D Compositing Window Manager (Beryl, Compiz)
- Moderately-high-end graphics hardware
- Multiple, configurable controllers (MIDI)
- Interface via VR-SDR nodes

Some Pure WM Examples



- A Beryl Desktop Cube

More WM Examples



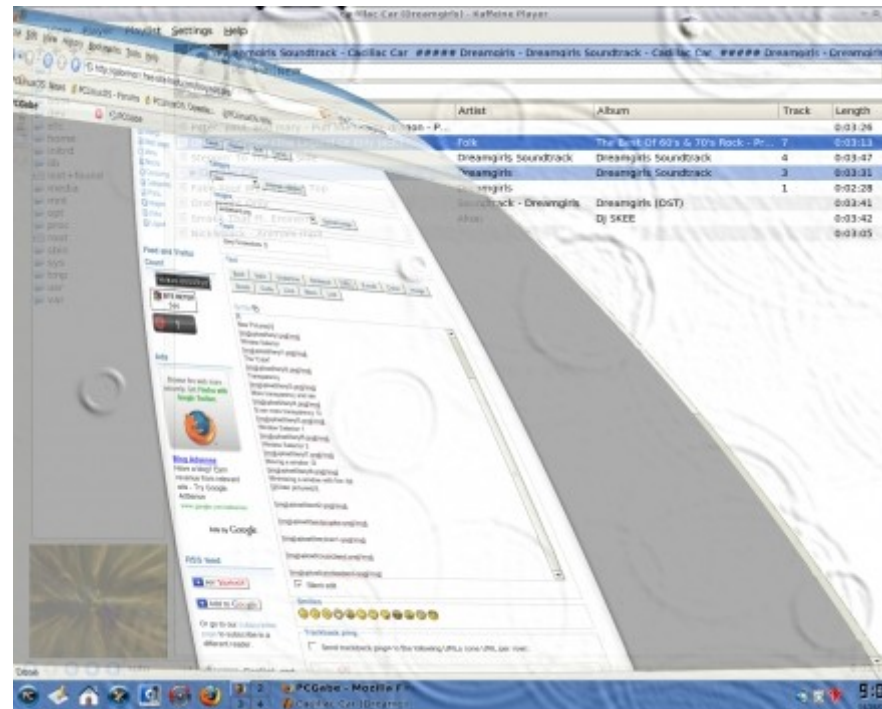
- Compositing WM arranging with effects

More WM Examples



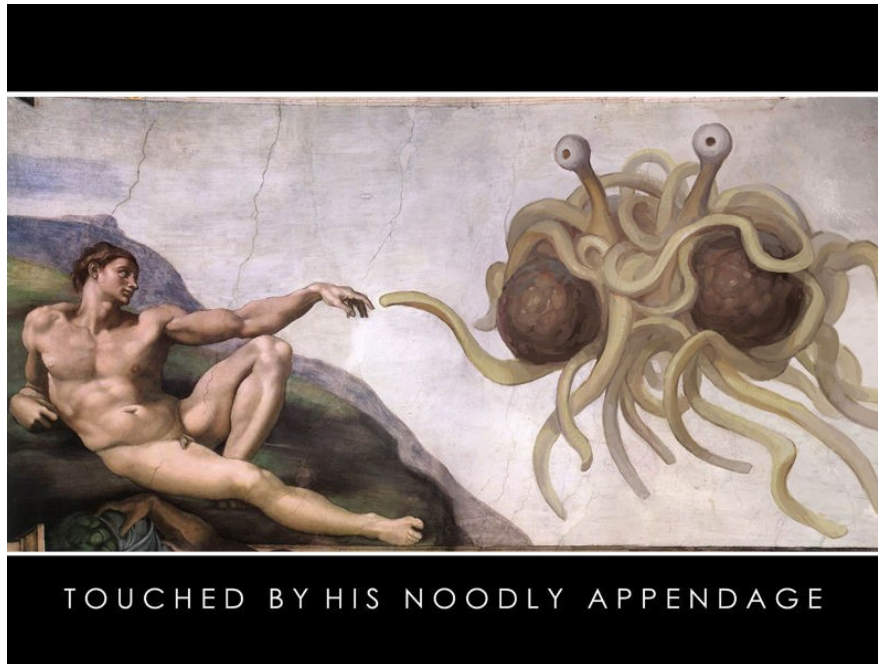
- Desktop Cube with stacked overlaid windows

More WM Examples



- Beryl dynamic warping

Postscript



- Thanks to N9DG, K3IB, W3SZ, AA6YQ, AB2WF, WA2DKJ, K7GNU, N2MJI, N4HY
- There's some math to be mined here too...