

# ***UDR56K-4 Software***

NW Digital Radio  
John D. Hays, K7VE  
Microhams 2013

# *Topics*

- Operating System
- Generic Applications
- Ported Applications
- Writing Command and Control
  - Web Based



# *Operating System*

- Debian Linux
  - Armel (Soft Floating Point)
  - Squeeze or Wheezy\*
  - Using the Raspberry Pi
    - Wheezy Armel (not Raspbian)
    - UDR56K-4 has higher performance on Armel

# *Applications*

- Existing applications from Debian
  - gpsd (with gpsd-utilities)
  - ntp (time synchronization)
  - xrdp (for use with remote desktop)
  - Other pre-built armel packages



# ***Packet Programs***

- Initial NW Digital Delivery
  - APRX (Digipeater, iGate)
  - RMS Gateway (Email, Winlink)
  - Paclink-Unix (client)
  - UDRTracker (APRS Tracker – adapted from KK7DS design)

# ***D-STAR Programs***

- NW Digital / G4KLX programs
  - ircDDBGateway
  - SplitRepeater
  - UDRrepeater (modified from  
DSTARrepeater)
  - D-STAR Voice
  - D-STAR Data



# *Command and Control Programs*

- NW Digital Radio
  - UDRconsole
    - Based on JSON and Websockets
    - node.js framework
    - HTML, CSS, and Javascript
    - Requires modern browsers
    - Extensible through open source

# ***Command and Control Our Approach***

- Low Level Drivers
  - Open Source
  - Socket Interfaces
- Sub-systems
  - I/Q modulators
  - Synthesizer Control
  - PA Control

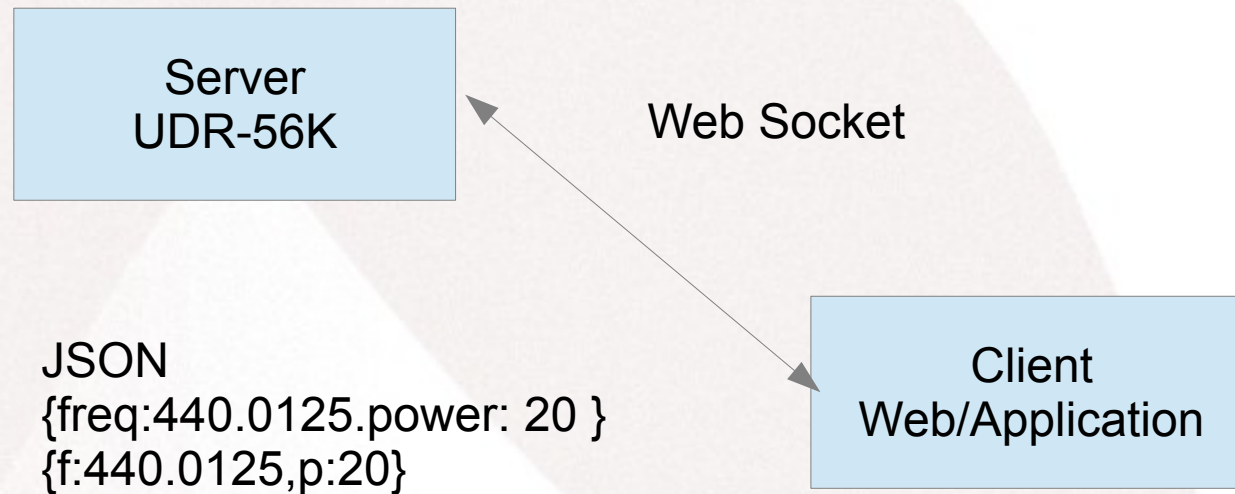


# *Command and Control Framework*

- node.js
  - Provides the container for server code
  - Services are written in JavaScript
    - Other languages can be used
  - Data is passed over sockets as JSON
  - Rich set of pre-written services
    - db, networking, filesystem, ...
  - Non-Blocking, use callbacks for asynchronous data

# *node.js*

## *Basic Communication*





# *node.js*

## *Chat Server Example*

```
var app = require('express').createServer()
var io = require('socket.io').listen(app);
app.listen(8080);
app.get('/', function (req, res) { res.sendFile(__dirname + '/index.html'); });
var usernames = {};
io.sockets.on('connection', function (socket) {
  socket.on('sendchat', function (data)
    {io.sockets.emit('updatechat', socket.username, data); });
  socket.on('adduser', function(username){
    socket.username = username;
    usernames[username] = username;
    socket.emit('updatechat', 'SERVER', 'you have connected');
    socket.broadcast.emit('updatechat', 'SERVER', username + ' has connected');
    io.sockets.emit('updateusers', usernames);
  });
  socket.on('disconnect', function(){
    delete usernames[socket.username];
    io.sockets.emit('updateusers', usernames);
    socket.broadcast.emit('updatechat', 'SERVER', socket.username + ' has
disconnected');
  });
});
```

# Chat Client Page

## The JavaScript

```
<script src="/socket.io/socket.io.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.min.js"></script>
<script>
  var socket = io.connect('http://localhost:8080');
  socket.on('connect', function(){ socket.emit('adduser', prompt("What's your name?")); });
  socket.on('updatechat', function (username, data) {
    $('#conversation').append('<b>'+username + ':</b> ' + data + '<br>');
  });
  socket.on('updateusers', function(data) {
    $('#users').empty();
    $.each(data, function(key, value) { $('#users').append('<div>' + key + '</div>'); });
  });
  $(function(){
    $('#datasend').click( function() {
      var message = $('#data').val();
      $('#data').val("");
      socket.emit('sendchat', message);
    });
    $('#data').keypress(function(e) {
      if(e.which == 13) {
        $(this).blur();
        $('#datasend').focus().click();
      }
    });
  });
</script>
```



# *Chat Client Page*

## *The HTML/CSS*

```
<div style="float:left;width:100px;border-right:1px solid black;height:300px;padding:10px;overflow:scroll-y;">
  <b>USERS</b>
  <div id="users"></div>
</div>
<div style="float:left;width:300px;height:250px;overflow:scroll-y;padding:10px;">
  <div id="conversation"></div>
  <input id="data" style="width:200px;" />
  <input type="button" id="datasend" value="send" />
</div>
```

An Example – <http://qso.k7ve.org>

# *Chat Client Page*

## *The HTML/CSS*

```
<div style="float:left;width:100px;border-right:1px solid black;height:300px;padding:10px;overflow:scroll-y;">
  <b>USERS</b>
  <div id="users"></div>
</div>
<div style="float:left;width:300px;height:250px;overflow:scroll-y;padding:10px;">
  <div id="conversation"></div>
  <input id="data" style="width:200px;" />
  <input type="button" id="datasend" value="send" />
</div>
```

An Example – <http://qso.k7ve.org>



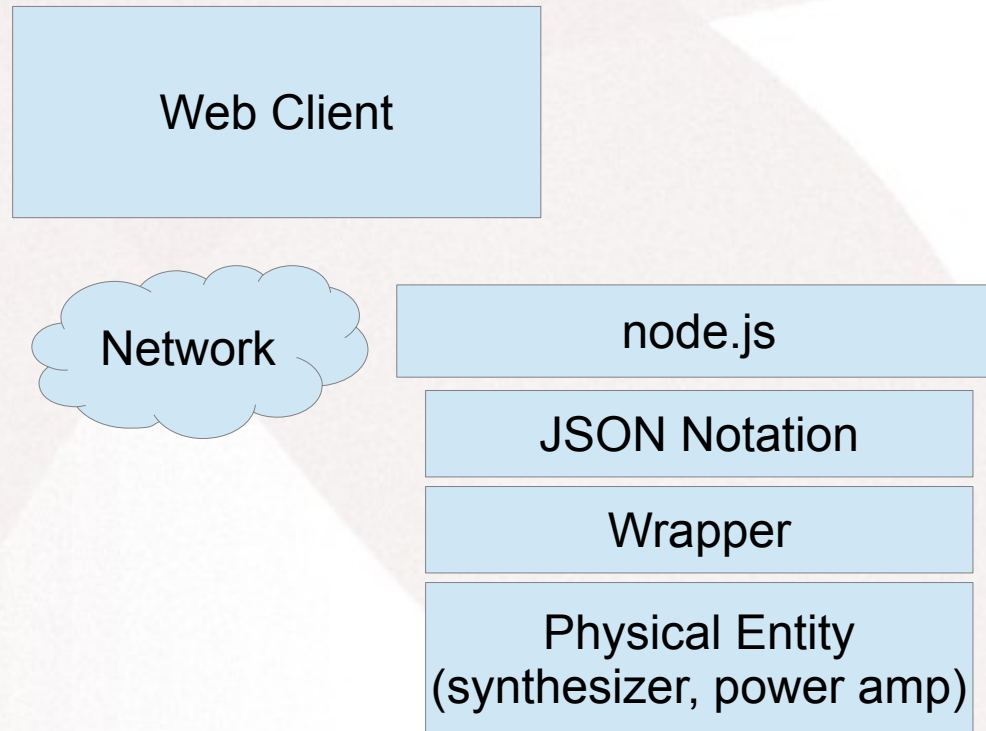
# *node.js*

## *Packet Sniffing*

```
var pcap = require("pcap"), pcap_session = pcap.createSession("",
    "dst port \ (9007 or 14580\)", matcher = /NW7DR/;
...
var httpserver = http.createServer(app).listen(app.get('port'), "192.231.186.2", function()
    {console.log("Express server listening on port " + app.get('port')); });
var io = require('socket.io').listen(httpserver);
io.sockets.on( 'connection', function(socket) {
    pcap_session.on( 'packet', function(raw_packet) {
        var packet = pcap.decode.packet(raw_packet),
            data = packet.link.ip.tcp.data;
        switch (packet.link.ip.tcp.dport) {
            case 9007: // Do Processing
                break;
            case 14580: // Do Processing
                break;
        }
    }
    ...
}
```

Example: <http://nw7dr.dstar-relay.net>

# *Command and Control through node.js*





# *Roll Your Own Interface*

- Install node.js
  - Linux, MacOS, Windows, ?
  - Simulate UDR56K-4 JSON
    - API still being ironed out
  - Write your user interface
    - Javascript, Python, C++, Ruby, ...
    - Use web socket
- When you get your UDR56K-4
  - Point your interface at it's node.js server
- Et Voila!

# *Low Level Interfacing*

- Our drivers our open source
  - Example of pattern
  - “You break it, you fix it!”
  - May need to build new kernel
  - Cross-compilation is your friend
    - Recommend using Debian/Ubuntu cross development platform
  - Honor licensing (GPL, MIT, Apache, ...)
  - Share with community
- Real Time Protocol for Web (future)



# ***Questions and References***

[Http://NWDigitalRadio.com](http://NWDigitalRadio.com)  
[info@NWDigitalRadio.com](mailto:info@NWDigitalRadio.com)

[Http://k7ve.org/blog](http://k7ve.org/blog)  
[john@hays.org](mailto:john@hays.org) / [k7ve@k7ve.org](mailto:k7ve@k7ve.org)

[Http://groups.yahoo.com/group/UniversalDigitalRadio](http://groups.yahoo.com/group/UniversalDigitalRadio)  
[Http://groups.yahoo.com/group/ircDDBGateway](http://groups.yahoo.com/group/ircDDBGateway)  
[Http://groups.yahoo.com/group/ARMedAmateurs](http://groups.yahoo.com/group/ARMedAmateurs)