Software Defined Radio Server

“A Radio Server for VHF+ Contesting And Weak Signal Work”

A Radio Server for HF, VHF+ Contesting, and Weak Signal Work using a port 80 (browser based) control approach

Phil Theis K3TUF

Digital Communications Conference
October 10, 2015
Background

SDR-1000

PowerSDR

DB15

Universal Controller Board

4 bit In: 16 DPDT Relay Out
Background

Flex 5000
Flex 3000
Flex 1500

I2C
FlexWire

I2C
GPIO
Expander

PowerSDR

Universal
Controller
Board

4 bit In: 16 DPDT Relay Out
UCB Daughter Board
Flex 5000
Flex 3000
Flex 1500

I2C
FlexWire

I2C GPIO Expander +
2 X Octal Darlington

16 Open Collector Outputs

PowerSDR
FlexWire Board

Used with:

Flex 1500
Flex 3000
Flex 5000
Finding the right MultiPole Relay
FlexWire(I2C) with RF Relays
Flex 6000 series
Initial Plans

- Need Band Data
- Switch Transverters
- 6700 is Great Radio (#1 on Sherwood Engineering List)
- No way to change uW bands
- Or HF bands for that matter
Put an Embedded Device to work

- Select Device
- Use Rapid Development Tools
  - Python
- Get on the air
- End of Story?
Python in Action
Elegance and Simplicity

- Integrated Development Environment
- Built In – Off the Shelf
  - Beagle Bone Black
  - Immediate Bone Script
  - Python
  - Ethernet or USB

October 2014
Talk Today

- Take you through the Process
- See what I learned along the way
- Much more that can happen
  - Transverter Control
  - Remote Control of 6K radios
  - Contest Mode Control
  - Tasks around the Shack
  - Monitoring
- All Via Ethernet
Device Choices

- Arduino
- Raspberry Pi
- Beagle Bone
Beagle Bone Black

- DC Power
- 10/100 Ethernet
- Ethernet PHY
- USB Client
- LEDs
- Reset Button
- eMMC
- HDMI Framer
- USB Host
- 512MB DDR3
- Sitara AM3358
- Serial Debug
- microHDMI
- uSD
- Boot Button
# 65 possible digital I/Os

<table>
<thead>
<tr>
<th></th>
<th>P9</th>
<th></th>
<th>P8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DGND</td>
<td>1</td>
<td>DGND</td>
</tr>
<tr>
<td></td>
<td><strong>VDD_3V3</strong></td>
<td>2</td>
<td><strong>VDD_3V3</strong></td>
</tr>
<tr>
<td></td>
<td><strong>VDD_5V</strong></td>
<td>3</td>
<td><strong>VDD_5V</strong></td>
</tr>
<tr>
<td></td>
<td><strong>SYS_5V</strong></td>
<td>4</td>
<td><strong>SYS_5V</strong></td>
</tr>
<tr>
<td></td>
<td><strong>PWR_BUT</strong></td>
<td>5</td>
<td><strong>SYS_RESETN</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_30</strong></td>
<td>6</td>
<td><strong>GPIO_60</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_31</strong></td>
<td>7</td>
<td><strong>GPIO_40</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_48</strong></td>
<td>8</td>
<td><strong>GPIO_51</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_4</strong></td>
<td>9</td>
<td><strong>GPIO_5</strong></td>
</tr>
<tr>
<td></td>
<td>I2C2_SCL</td>
<td>10</td>
<td>I2C2_SDA</td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_3</strong></td>
<td>11</td>
<td><strong>GPIO_2</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_49</strong></td>
<td>12</td>
<td><strong>GPIO_15</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_117</strong></td>
<td>13</td>
<td><strong>GPIO_14</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_125</strong></td>
<td>14</td>
<td><strong>GPIO_123</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_121</strong></td>
<td>15</td>
<td><strong>GPIO_122</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_120</strong></td>
<td>16</td>
<td><strong>VDD_ADC</strong></td>
</tr>
<tr>
<td></td>
<td><strong>AIN4</strong></td>
<td>17</td>
<td><strong>GND_ADC</strong></td>
</tr>
<tr>
<td></td>
<td><strong>AIN6</strong></td>
<td>18</td>
<td><strong>AIN5</strong></td>
</tr>
<tr>
<td></td>
<td><strong>AIN2</strong></td>
<td>19</td>
<td><strong>AIN3</strong></td>
</tr>
<tr>
<td></td>
<td><strong>AIN0</strong></td>
<td>20</td>
<td><strong>AIN1</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GPIO_20</strong></td>
<td>21</td>
<td><strong>GPIO_7</strong></td>
</tr>
<tr>
<td></td>
<td><strong>DGND</strong></td>
<td>22</td>
<td><strong>DGND</strong></td>
</tr>
<tr>
<td></td>
<td><strong>DGND</strong></td>
<td>23</td>
<td><strong>DGND</strong></td>
</tr>
</tbody>
</table>

In GPIO mode, each digital I/O can produce interrupts.
Apache Web Server

- Port 80
- PHP
- Available to any Device
FLEX-6000
HW System Architecture
SmartSDR Ethernet API
Interfaces

Streaming Data

WFALL DATA
PAN DATA
METER DATA
RF IQ DATA
DISCOVERY

Control & Status

CLIENT DATA

UDP
TCP
FLEX-6XXX
3rd Party App using Ethernet API

SmartSDR Ethernet API

SmartSDR

UDP

TCP

Linux/Mac/etc. Computer

Client Application

FLEX-6XXX
Flex Uses the API

- SmartSDR Windows client rests on FlexLib which rests on the internet API
- CAT and DAX also use FlexLib
- You can do anything done in SmartSDR
- Unprecedented control over a Radio Server
DAX & SmartCAT
SmartSDR API

Objectives

- Provide a common interface for FlexRadio products
- Support the building of an ecosystem around SmartSDR for the benefit of customers, developers and FlexRadio
- Provide a way to use a FLEX-6000 in a variety of applications, even ones that may not be mainstream
API Standards

Radio control is a TCP/IP socket with simple commands (no standard known):

slice create freq=14.1 ant=ANT1
mode=USB
slice tune 0 14.105

Streaming Panadapter/Waterfall/Meter/Discovery data are VITA-49 Extension

I/Q and Real IF is VITA-49 IF Data (24-192ksps)
SmartSDR TCP/UDP API

Command Format

- Command preface, sequence, v-bar, command
  C134|slice create freq=7.243

- Response preface, sequence, v-bar, response
  R134|500000002

- Status preface, handle, v-bar, status
  S67EF9A22|slice 0 freq=7.243
  S67EF9A22|slice 0 filter_lo=300
  filter_hi=2700
SmartSDR TCP/UDP API
Connecting to radio

- TCP/IP socket connection to port 4992
- API provides API version and a “handle”
  - v1.1.0.0
  - H35E61405
- Send commands!
- Interface is asynchronous, commands are non-blocking
Slice Receivers, example

Create a slice receiver

```
slice create [freq=<MHz>] [ant=<antenna>] [mode=<mode>]
```

```
C34|slice create freq=14.236 mode=FDV
R34|0
```

Tune a slice receiver

```
slice tune 0 [freq=<MHz>] [ant=<antenna>] [mode=<mode>]
```

```
C45|slice 0 freq=14.236
R45|0
```

Change slice receiver settings

```
slice set <slice> [<feature>=<value>]
```

```
C71|slice set 0 diversity=1 tx=0 record=1
R71|0
```
Sniffing TCP/IP API Using Wireshark
The Magic of the API
Eclipse Development Environment
Programming Finally

- Program written in GNU 'C'
- Subscribes to Slice information in radio
- Parses the responses
- Watches for Frequency to change
- Sends signal to Band change output
  - Either GPIO or I2C
- Expands to additional needs
  - Active Slice
  - Active TX
Flex Web Interface

- Apache HTTP Server
- Show Radio Status
- Send commands to Radio
- Perhaps display Panadapter
- Waterfall?
Technology: Languages

- HTML Hyper Text Markup Language
- AJAX Asynchronous JavaScript and XML
- DOM The Document Object Model is a platform and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents
- Apache / PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language
Technology: Languages cont'd

- C Programming Language for the server
- JavaScript is a dynamic computer programming language. It is most commonly used as part of Web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed
- JSON JavaScript Object Notation
- Python for early proof of concept
My Port 80 Plan
(Web Access)

PC

Radio

Ethernet

Tab

PC

Run FlexWeb here

WiFi

FlexWeb Executes here

BBB

XVTR
BndChg

GPIO

I2C

Serial up to 4kft

Key
Mic
Speaker

Ard

cell
Hi Current Control
- Instantaneous Re-Configuration
- Liaison to Run
- Split Audio
- No Loss of Focus
- Complete Control of Radio
- LED Feedback
Future Tasks

- Monitor Temperatures
- Control Power Supplies
- Turn Antennas / Switch Antennas
- Round out the Remote Experience
- Multiple Locations with Distributed Computing
- Beacon Monitoring: Propagation Notification
- Performance of Beacons: Real Time Status
- Operate Station from FL in Winter
Thank You

Phil Theis, K3TUF
phil@k3tuf.com