Software Defined Radio Server

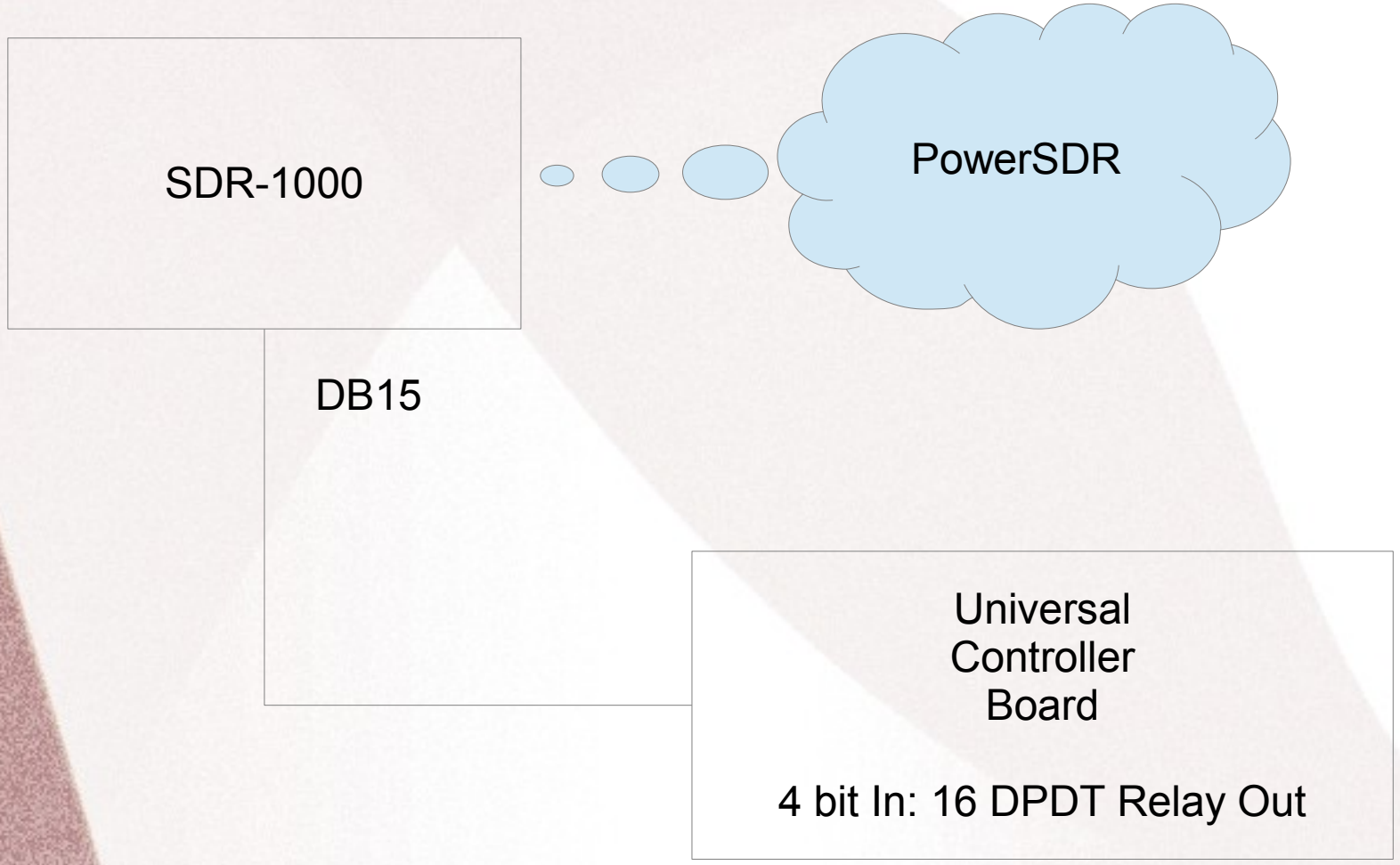
"A Radio Server for VHF+ Contesting
And Weak Signal Work"

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

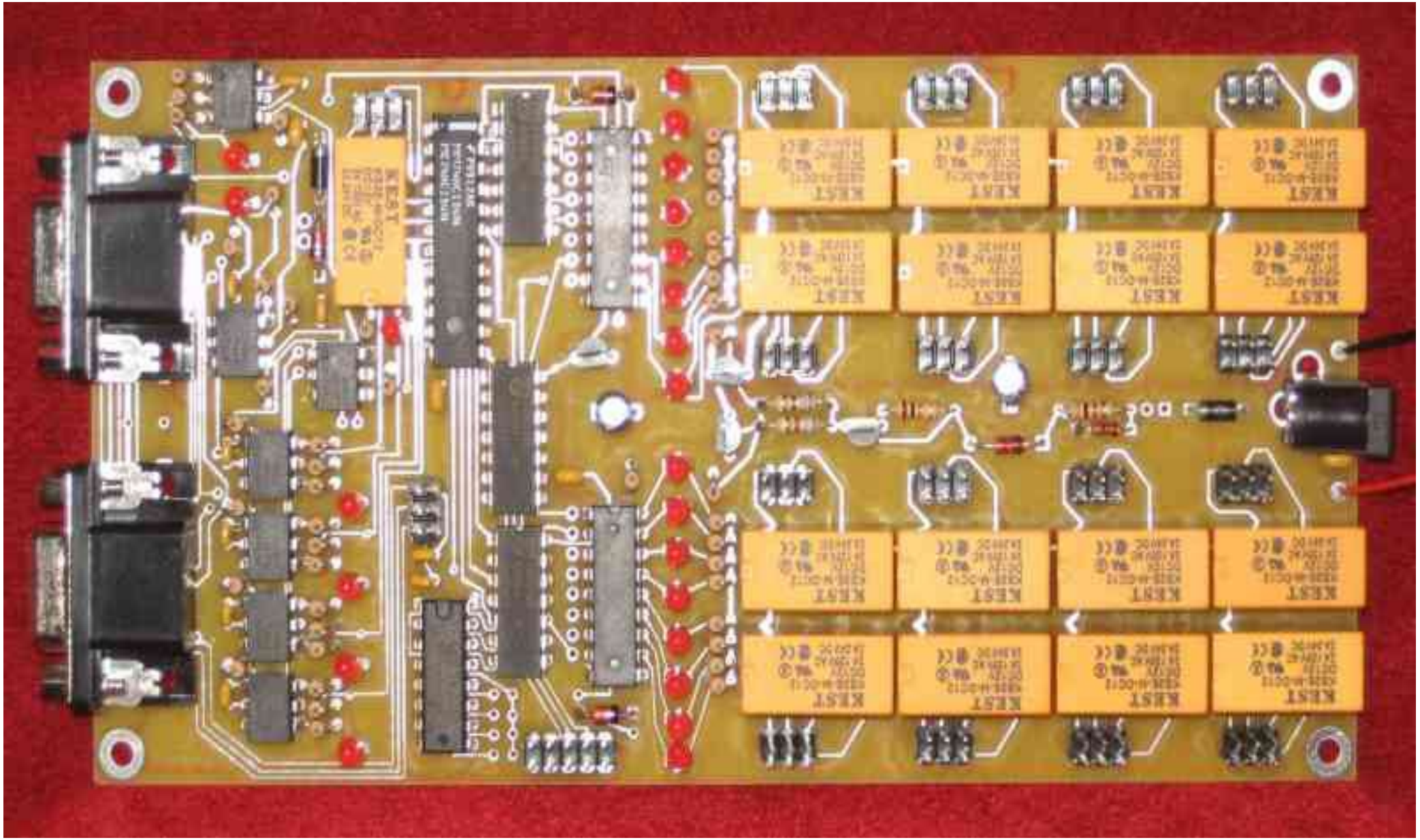
Phil Theis K3TUF

Digital Communications Conference
October 10, 2015

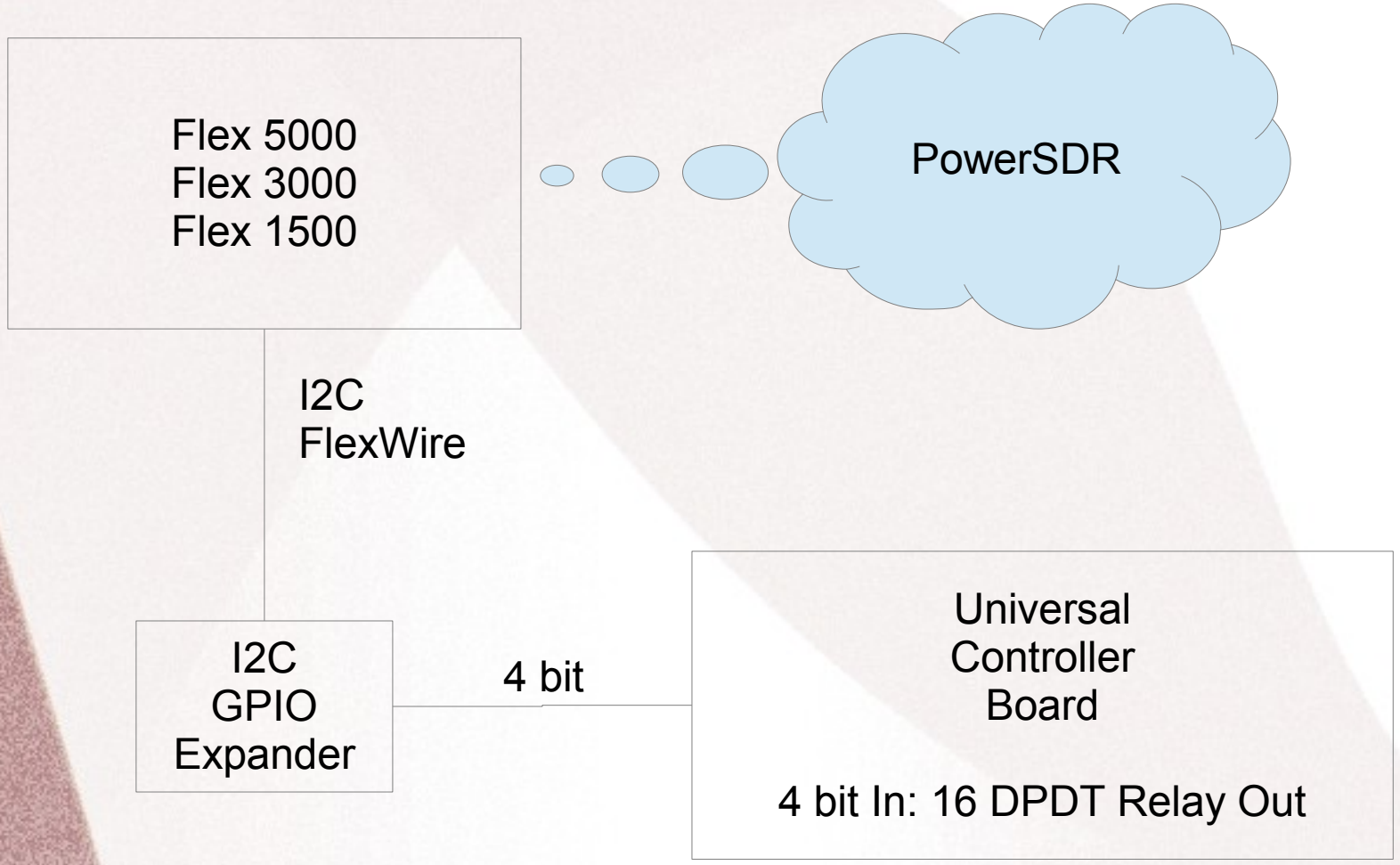
Background


SDR-1000
PowerSDR
DB15
Universal
Controller
Board
4 bit In: 16 DPDT Relay Out

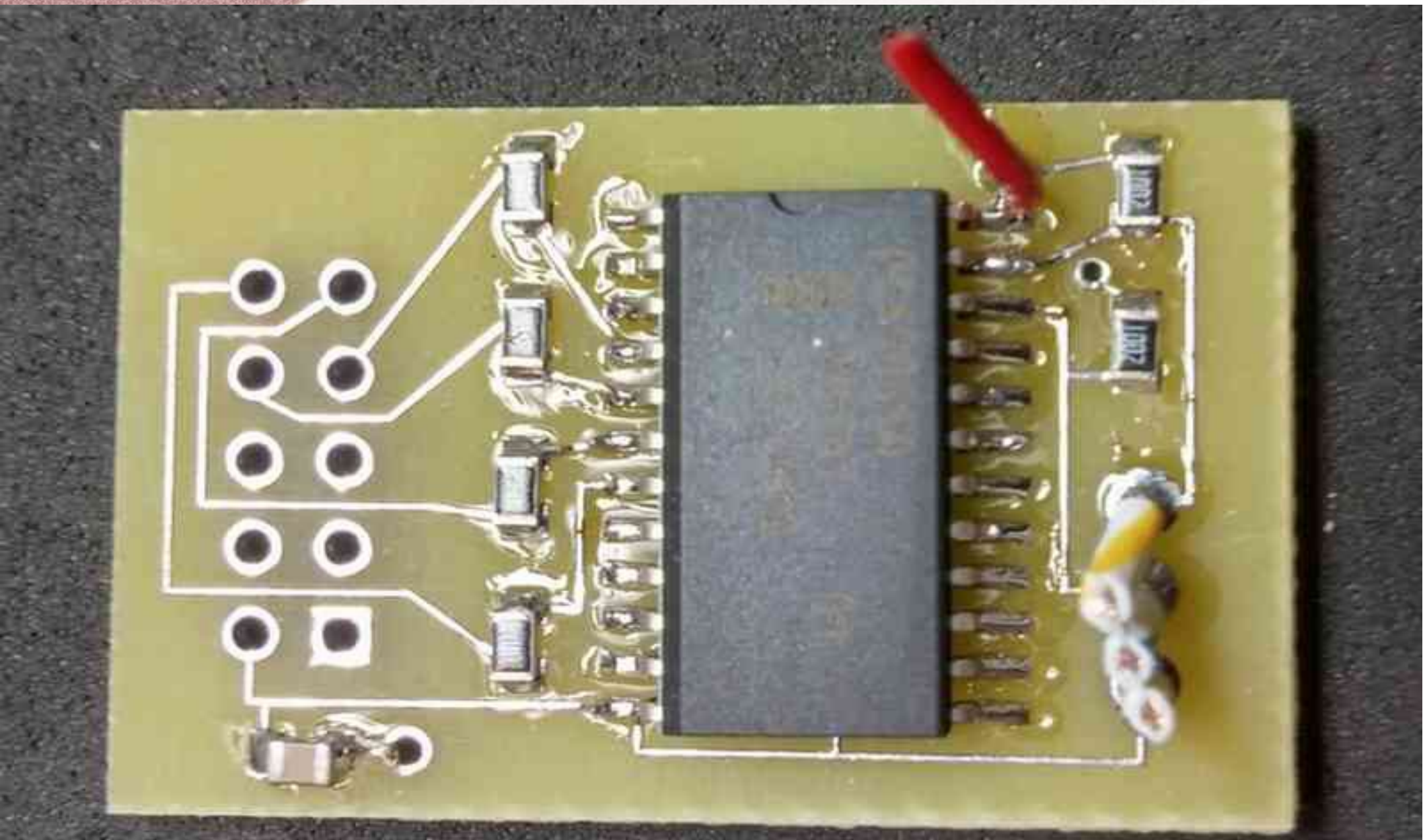
SDR1000 UCB



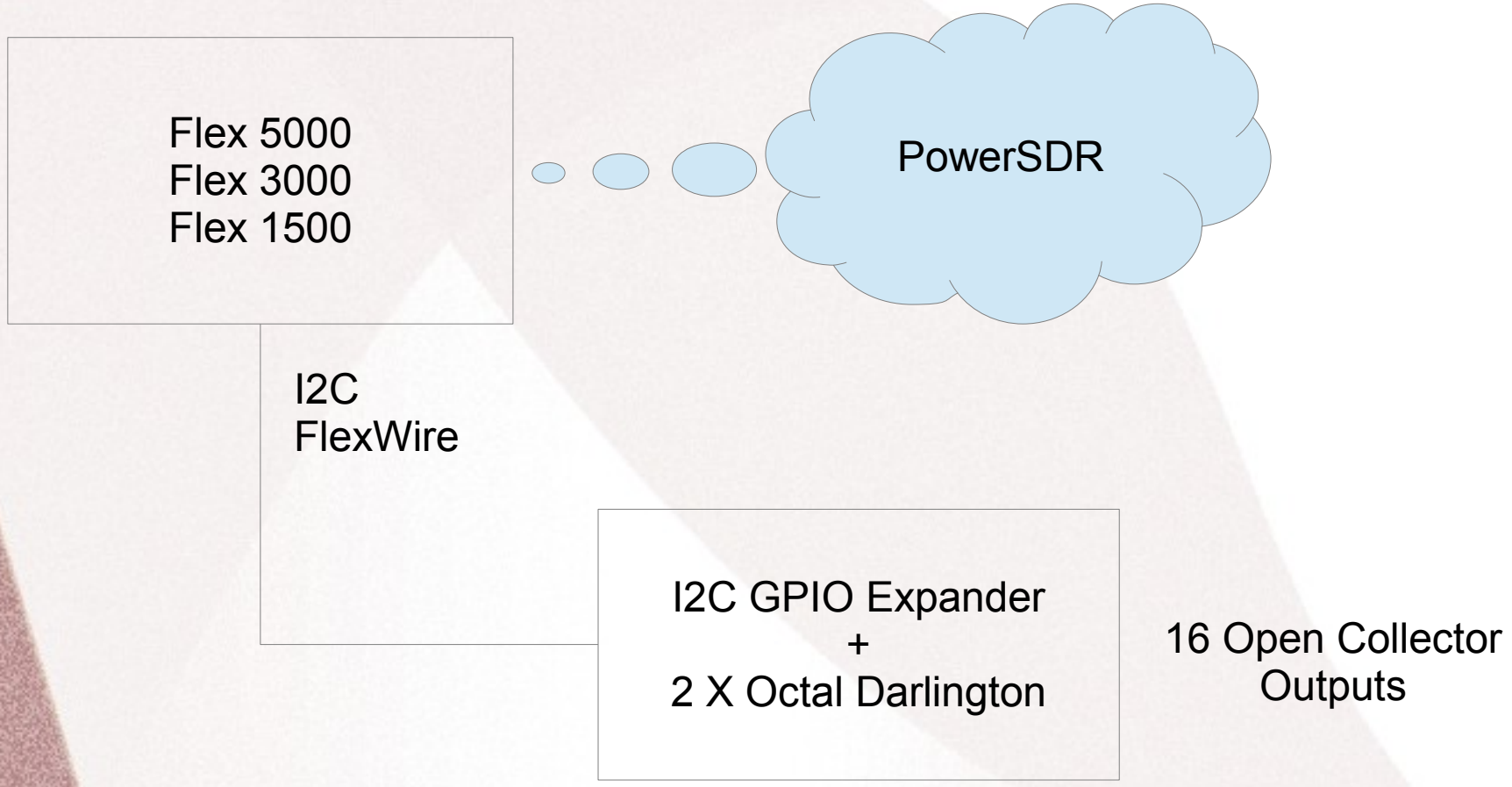
Background


Flex 5000
Flex 3000
Flex 1500
PowerSDR
I2C
FlexWire
I2C
GPIO
Expander
4 bit
Universal
Controller
Board
4 bit In: 16 DPDT Relay Out

UCB Daughter Board



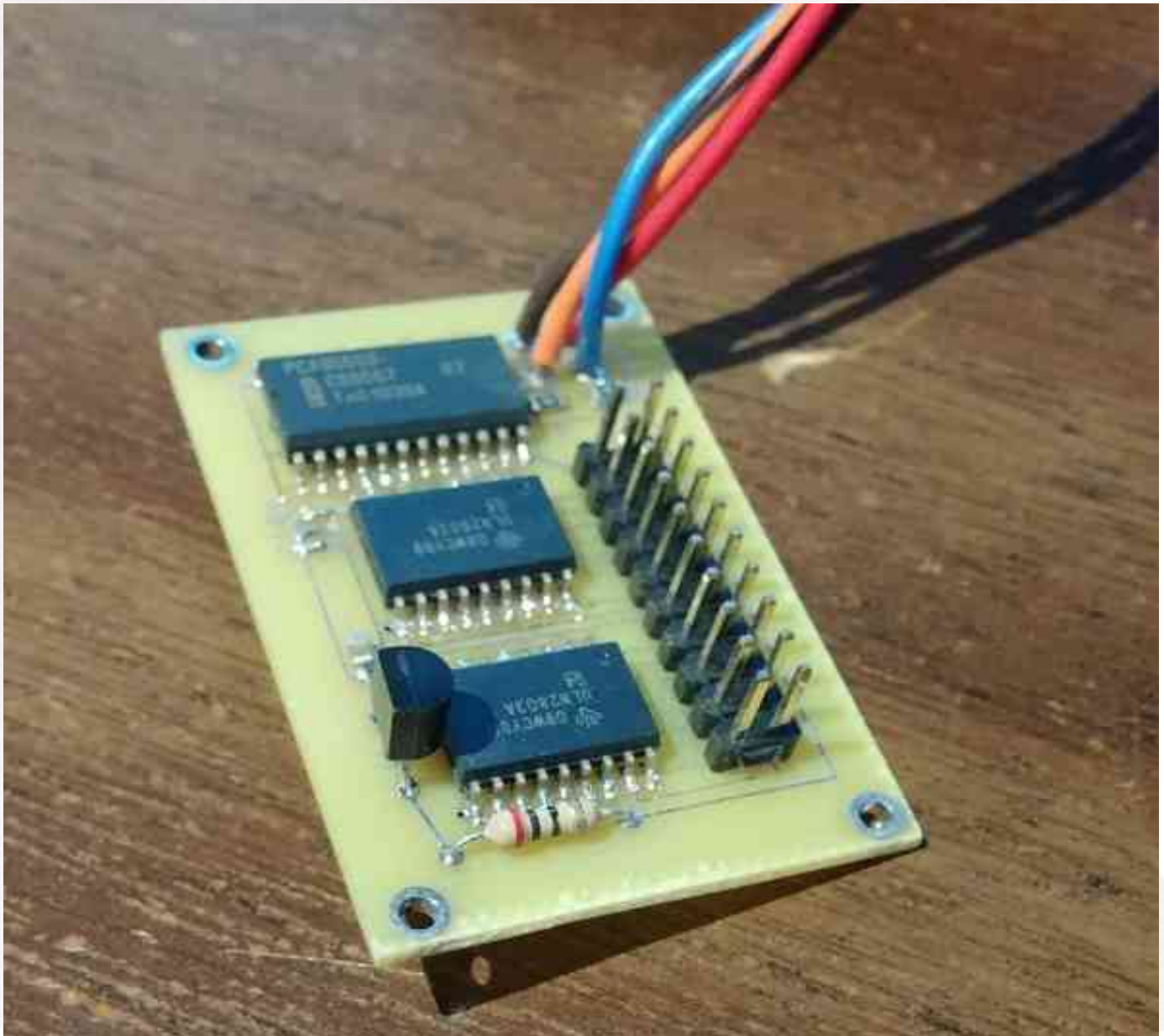
Background


Flex 5000
Flex 3000
Flex 1500
PowerSDR
I2C
FlexWire
I2C GPIO Expander
+
2 X Octal Darlington
16 Open Collector
Outputs

FlexWire Board

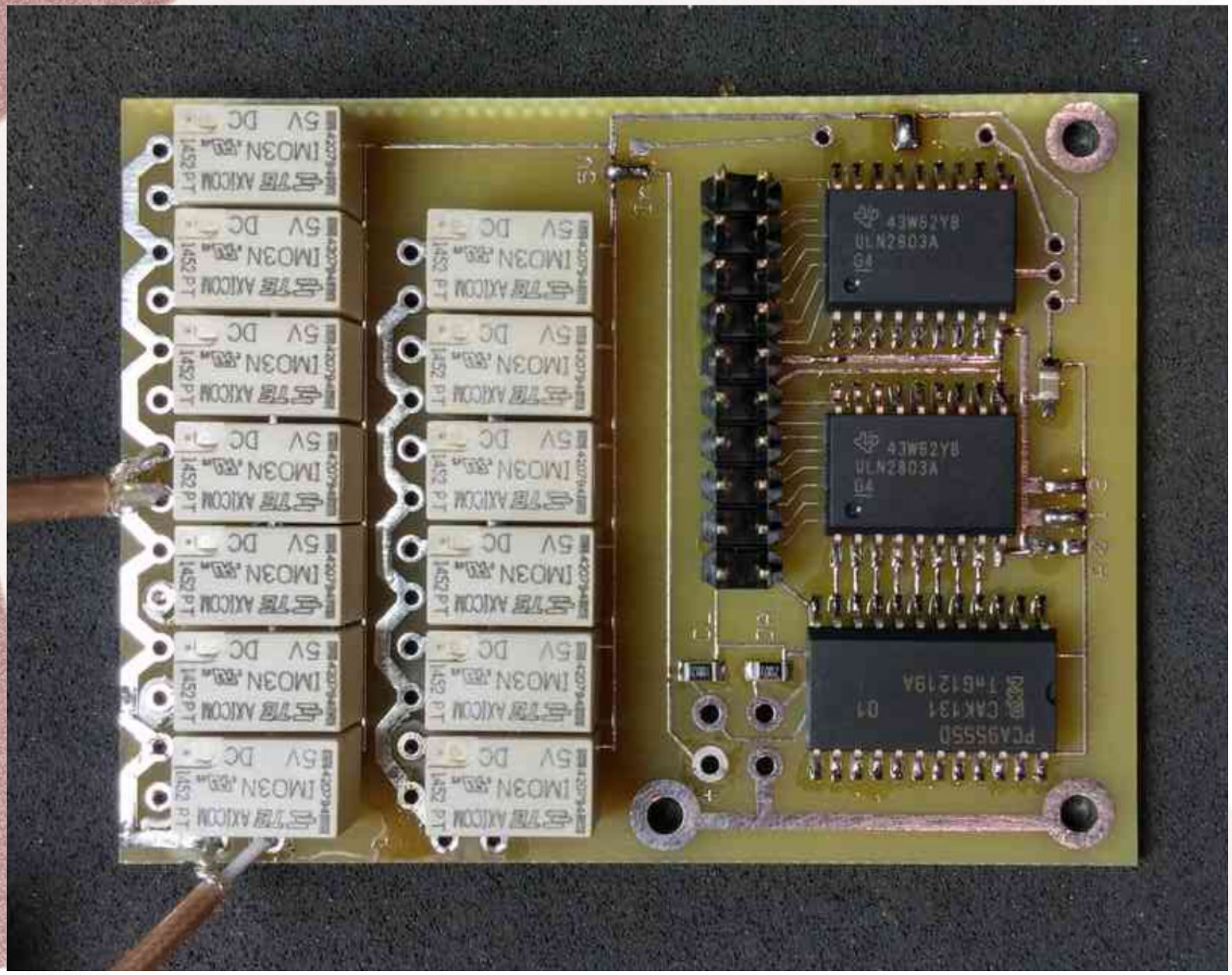
Used with:

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

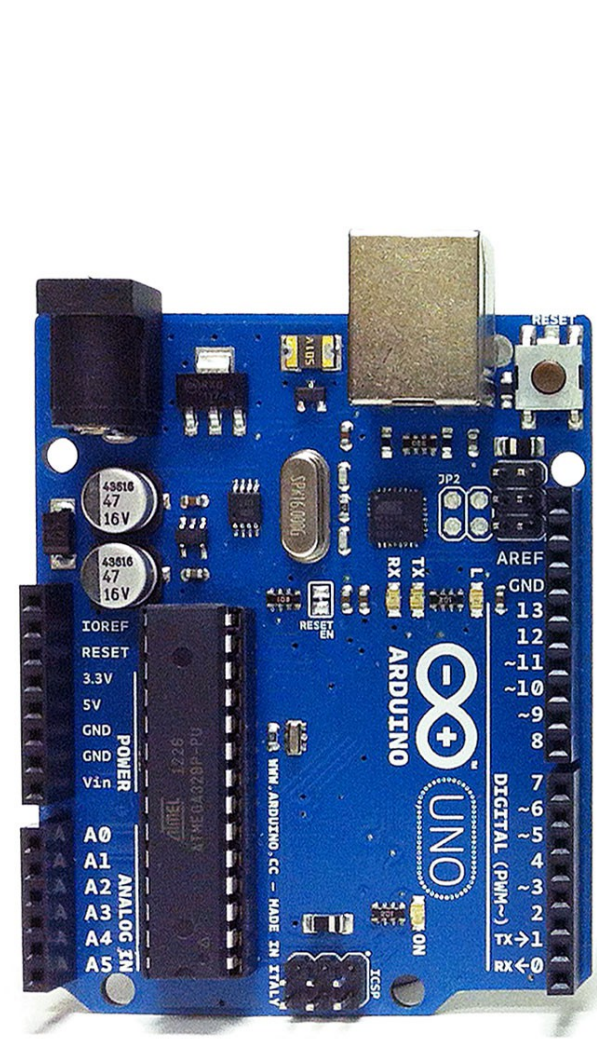
- Integrated Development Environment
- Built In – Off the Shelf
  - Beagle Bone Black
  - Immediate Bone Script
  - Python
  - Ethernet or USB

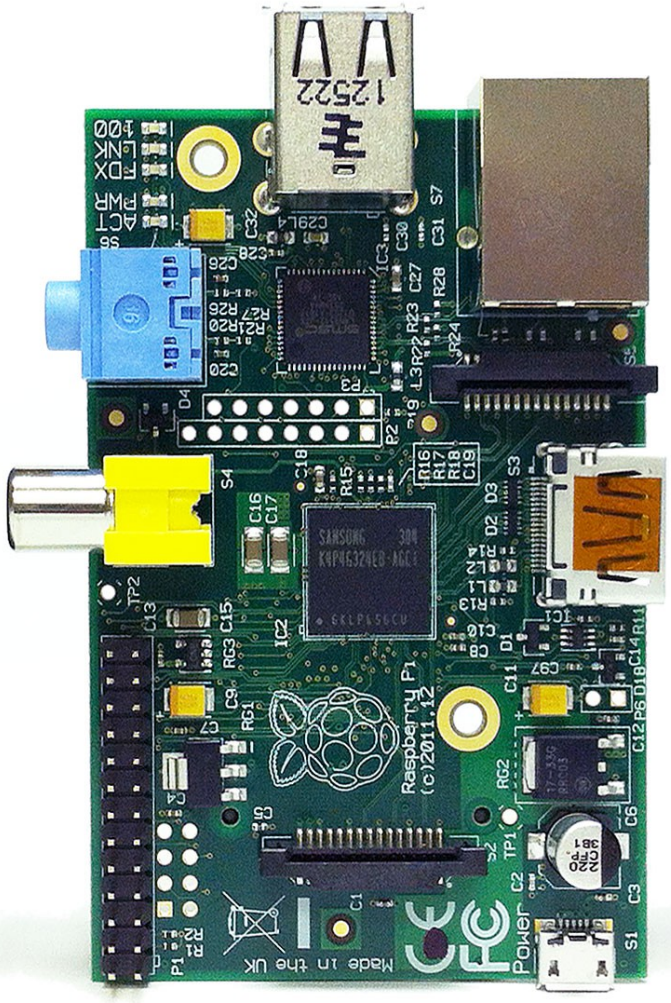
Talk Today

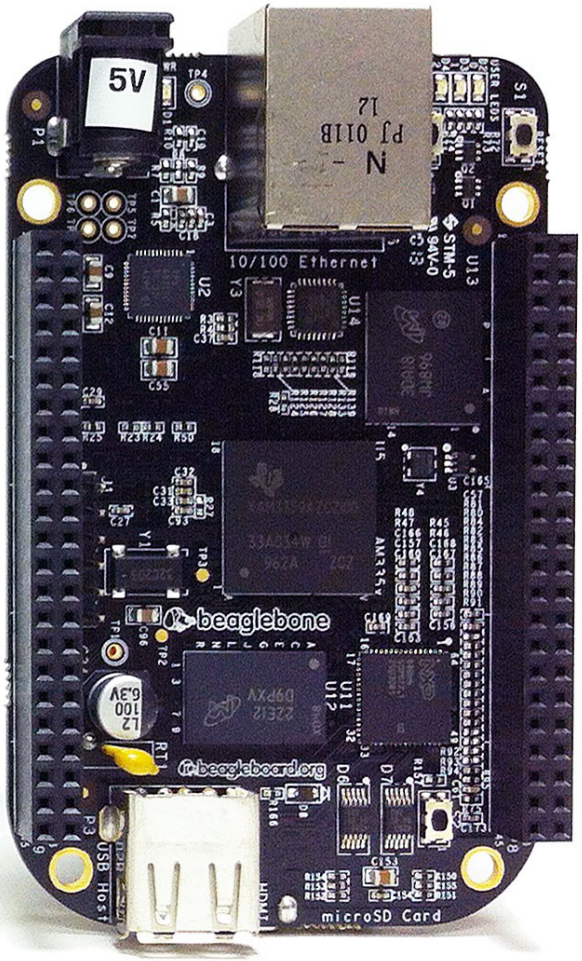
- Take you through the Process
- See what I learned along the way
- Much more that can happen
  - Transverter Control
  - Remote Control of 6K radios
  - Contest Mode Control
  - Tasks around the Shack
  - Monitoring
- All Via Ethernet

Device Choices

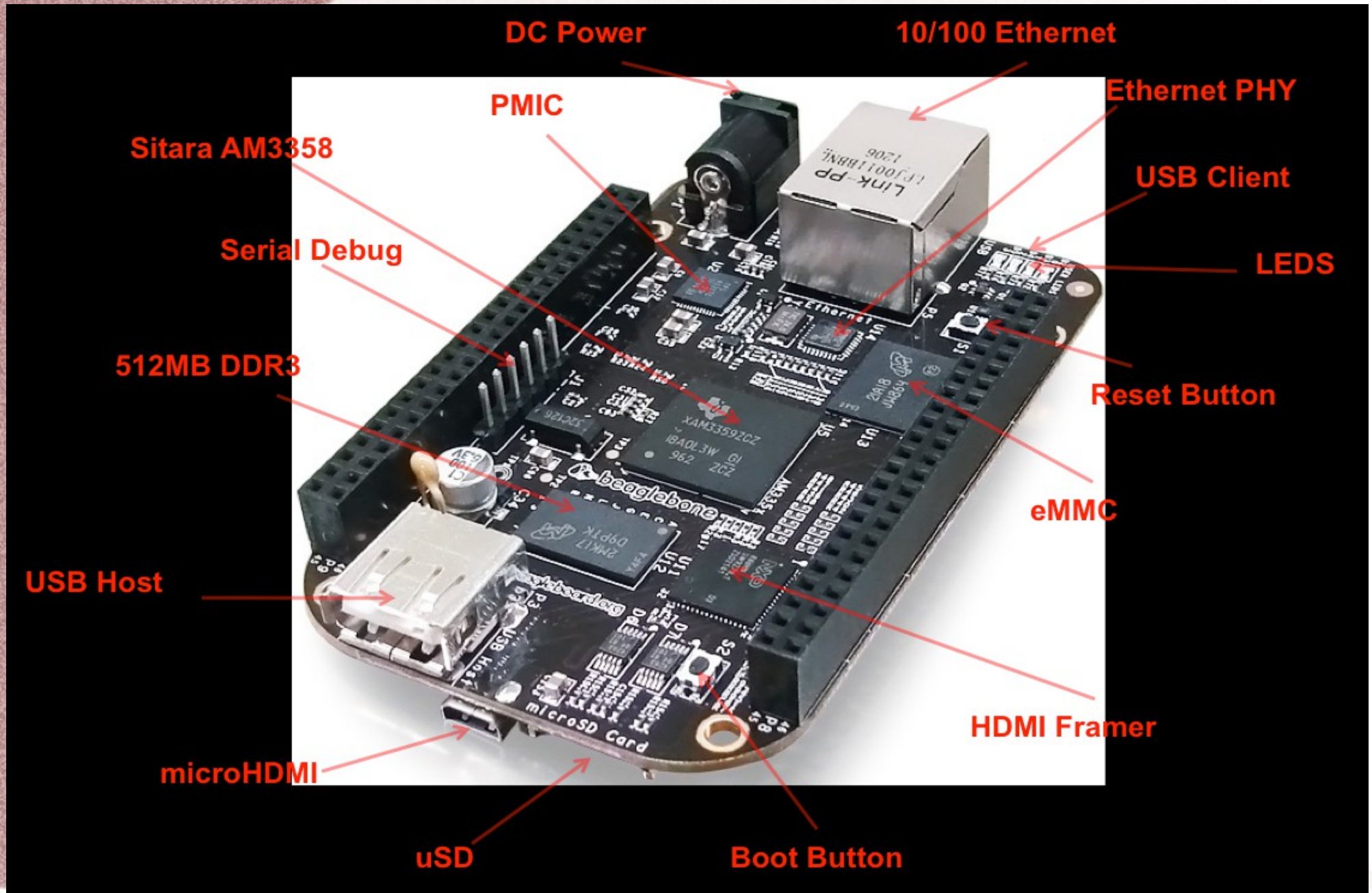
- Arduino – Rasberry PI – Beagle Bone







Beagle Bone Black



GPIO pins

65 possible digital I/Os

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_40
GPIO_48	15	16	GPIO_51
GPIO_4	17	18	GPIO_5
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_125	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

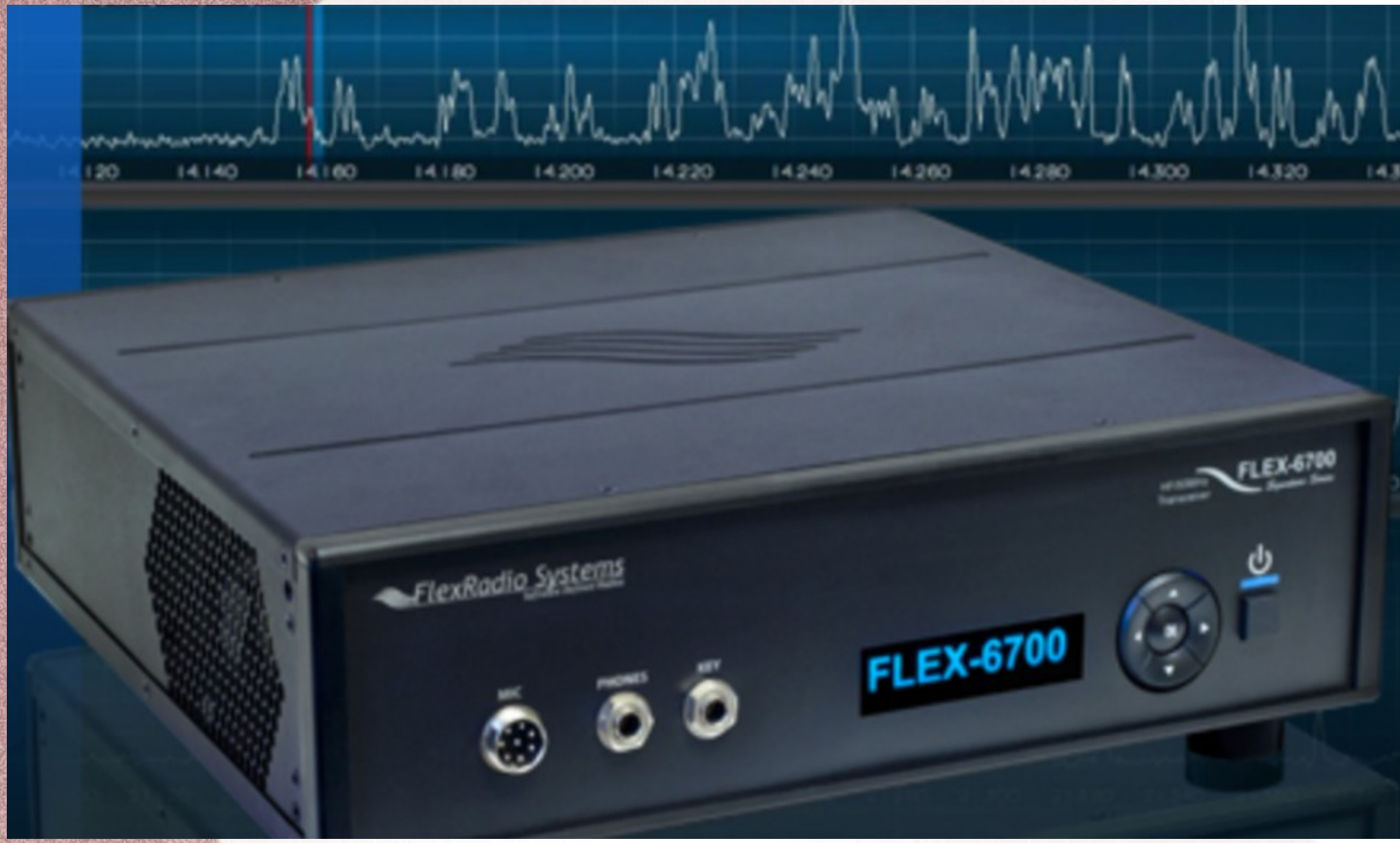
In GPIO mode, each digital I/O can produce interrupts.


LINK
10/100 Ethernet
beagleboard.org
5V
Esc
F1 RF
F2 PrtSc
F3
F4
F5
F6
F7
F8
BACK Del
Tab
Caps
Shift
Fn
Win Ctrl
Alt
AltGr
Enter CTRL+ALT+DEL

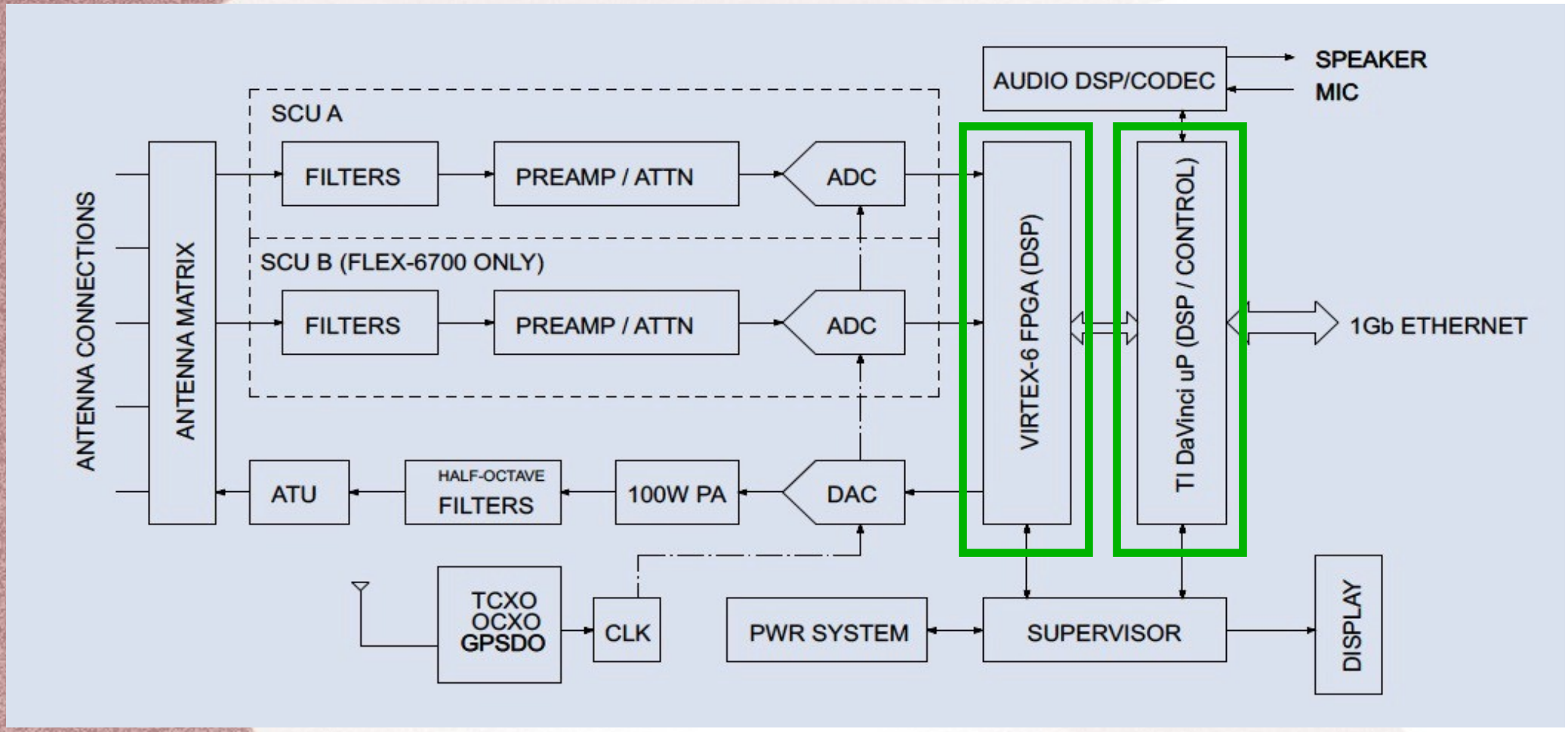
Apache Web Server

- Port 80
- PHP
- Available to any Device

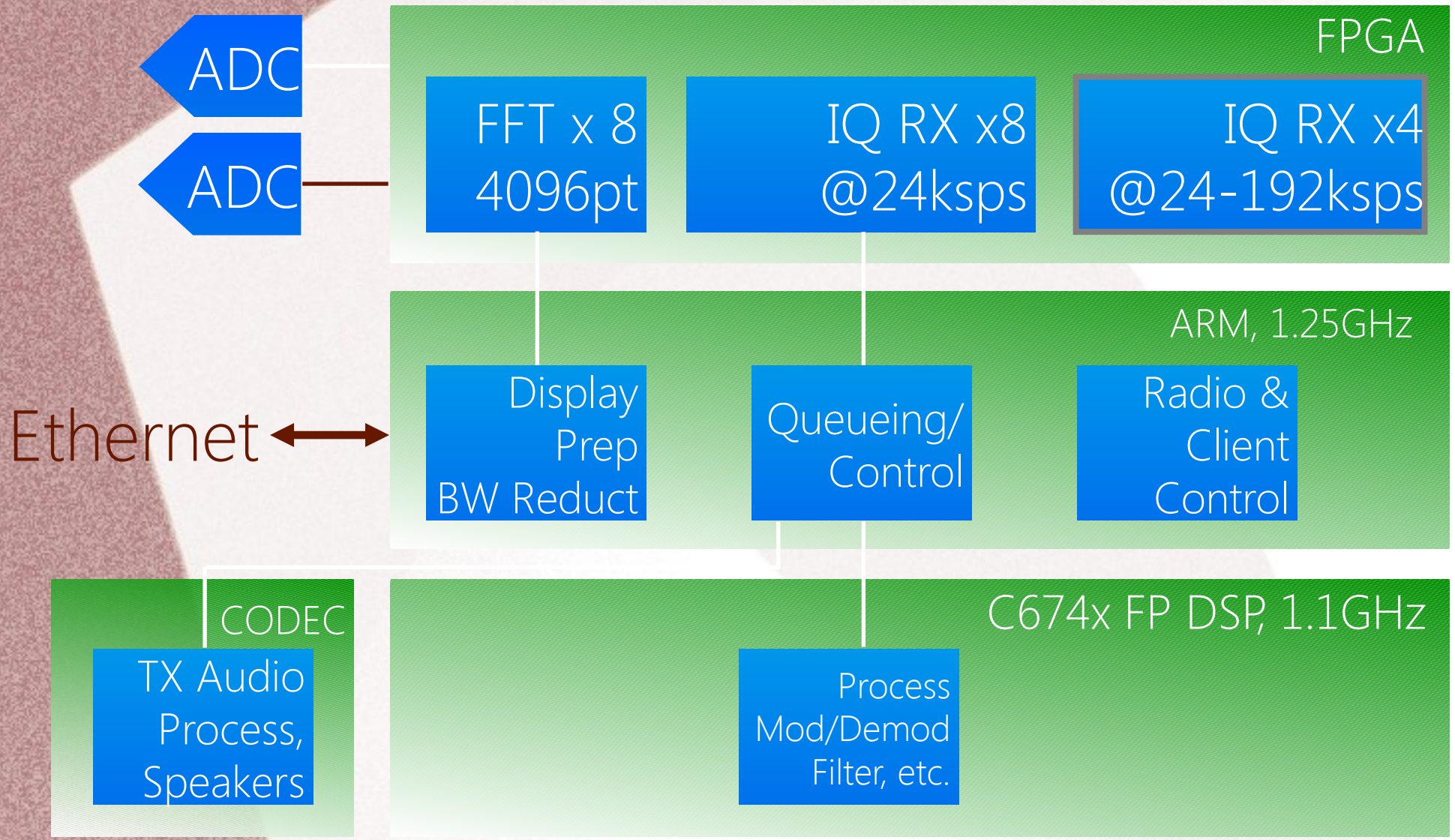
The Radio Server



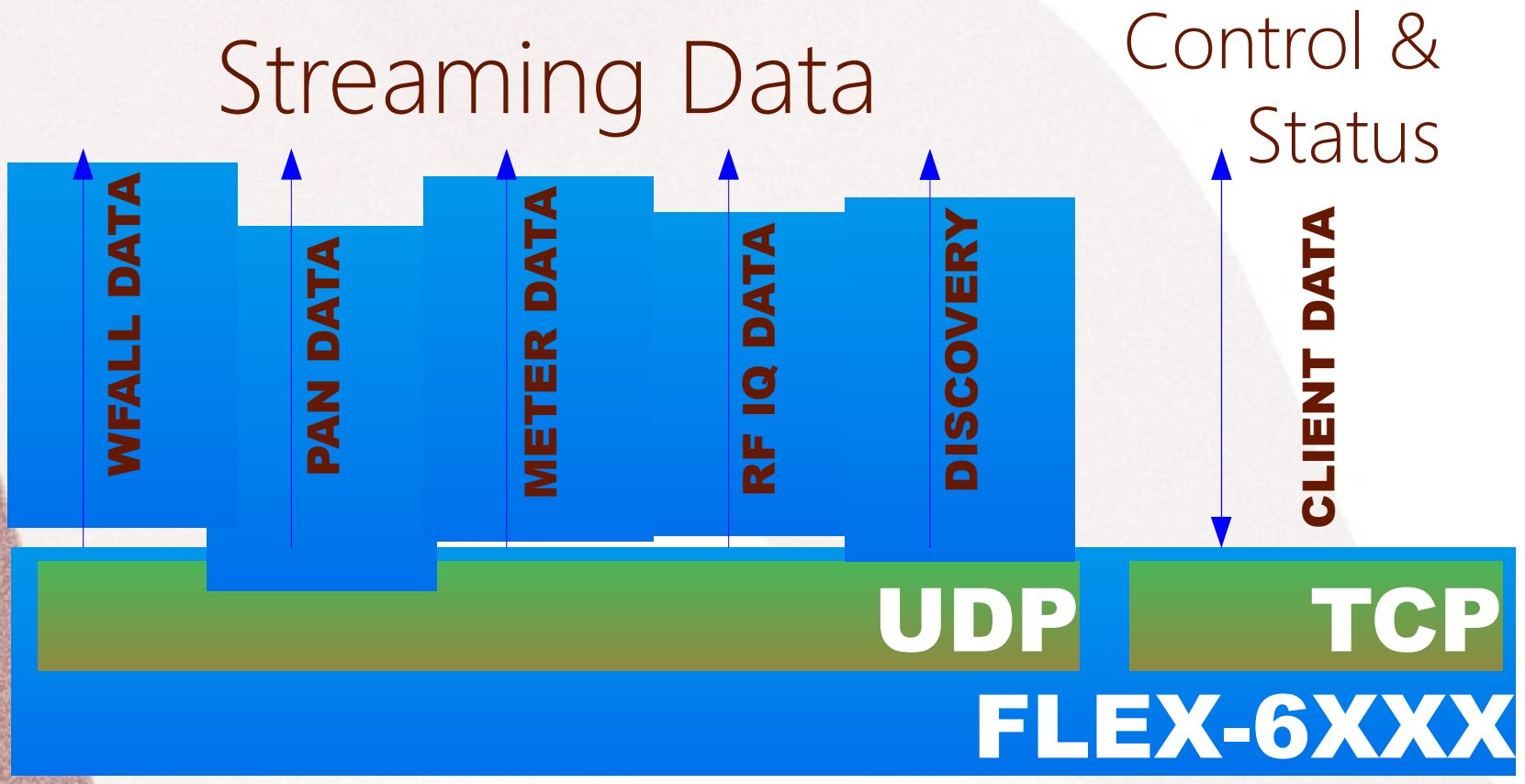
FLEX-6000
HW System Architecture



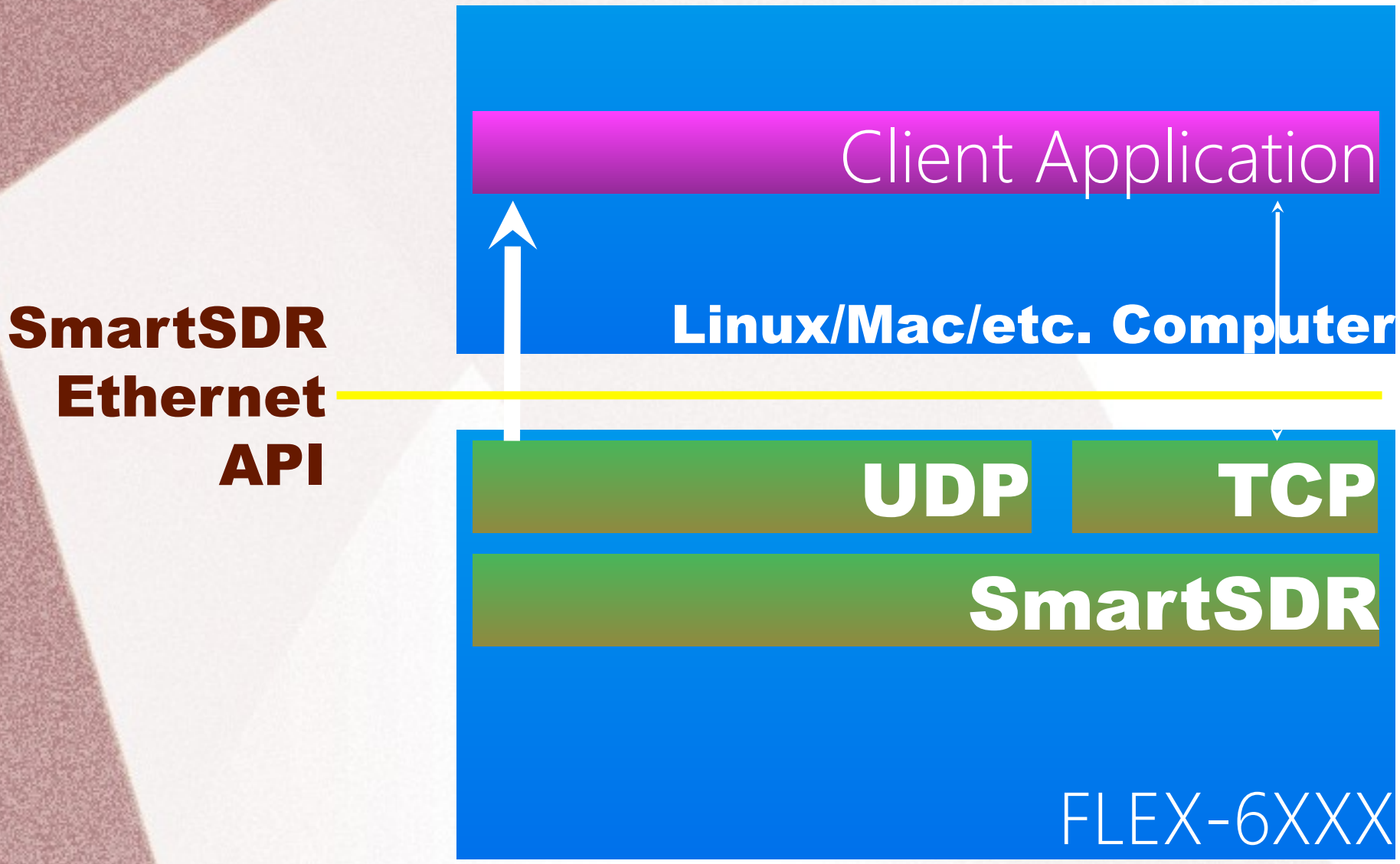
SmartSDR SW Architecture



SmartSDR Ethernet API Interfaces



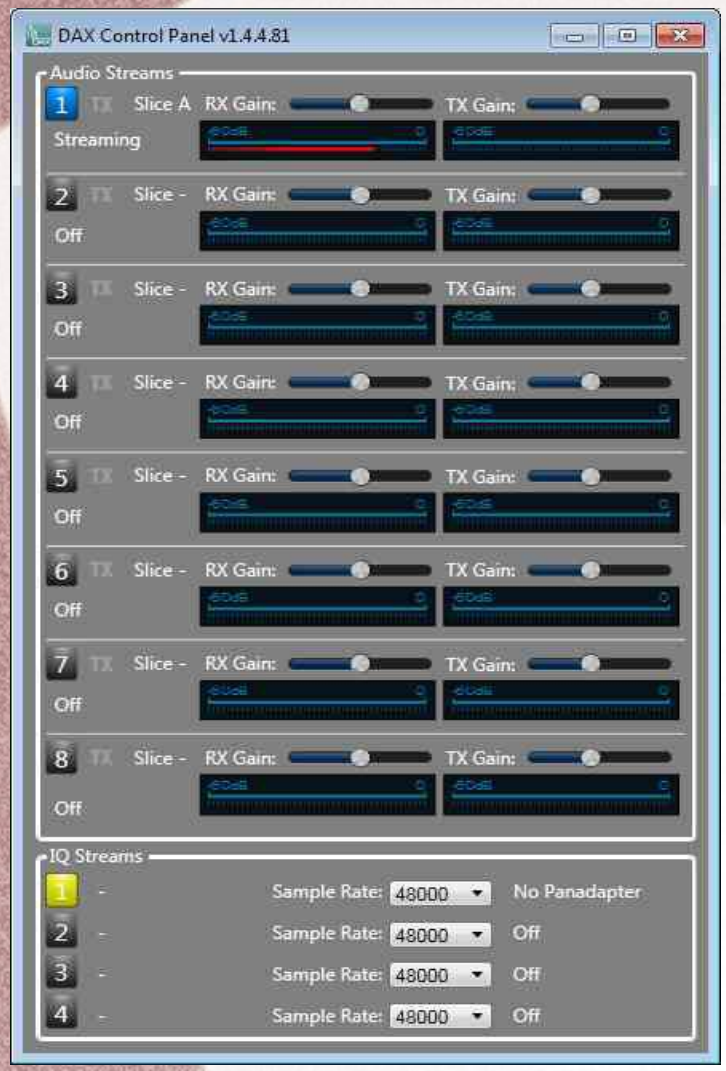
3rd Party App using Ethernet API

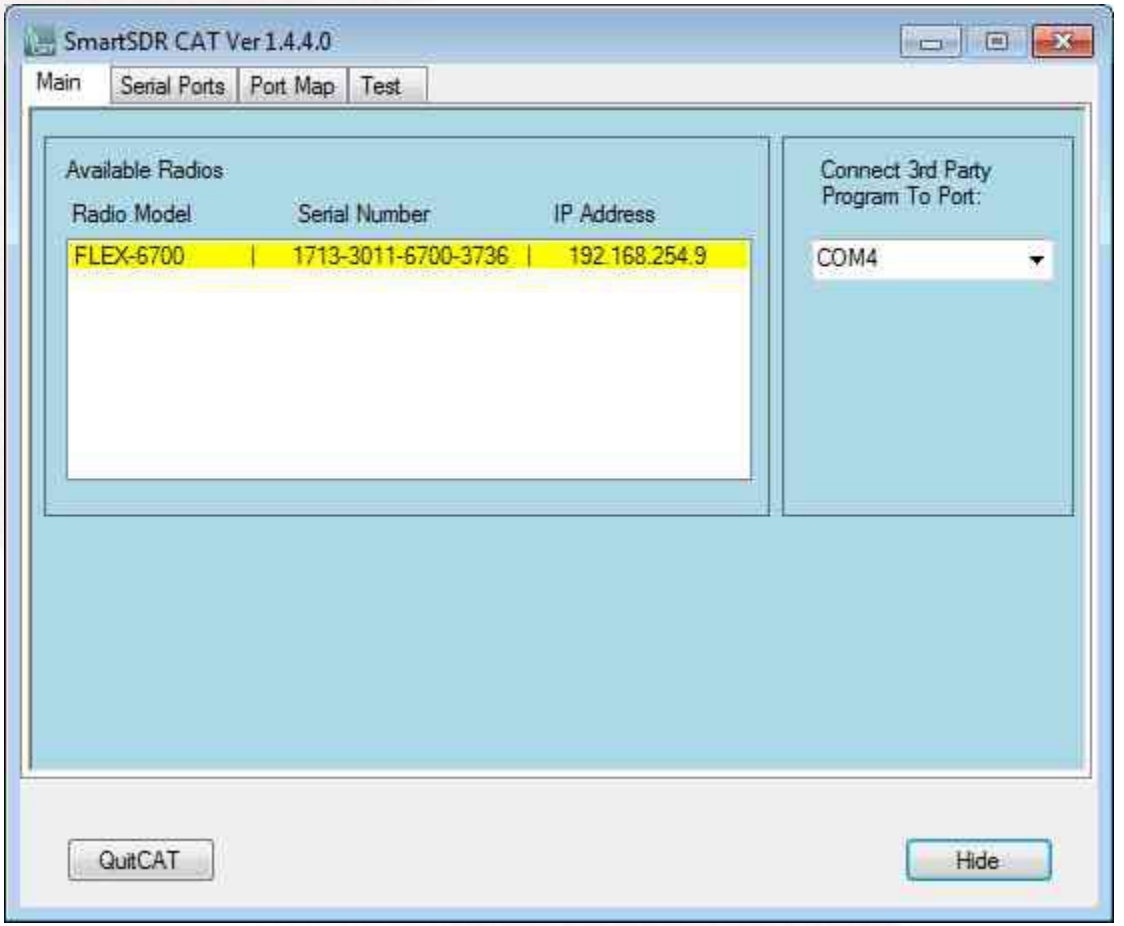


Flex Uses the API

- 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|50000002
```

- 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
```

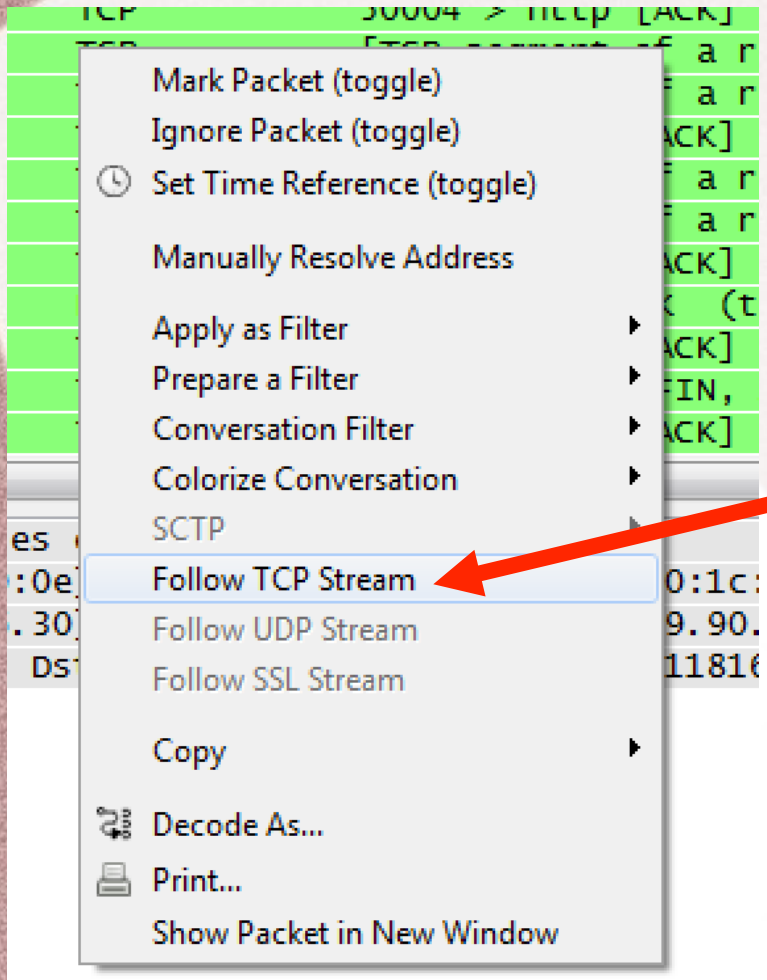
- Tune a slice receiver

```
slice tune 0 [freq=<MHz>] [ant=<antenna>]
[mode=<mode>]
C45|slice 0 freq=14.236
R45|0
```

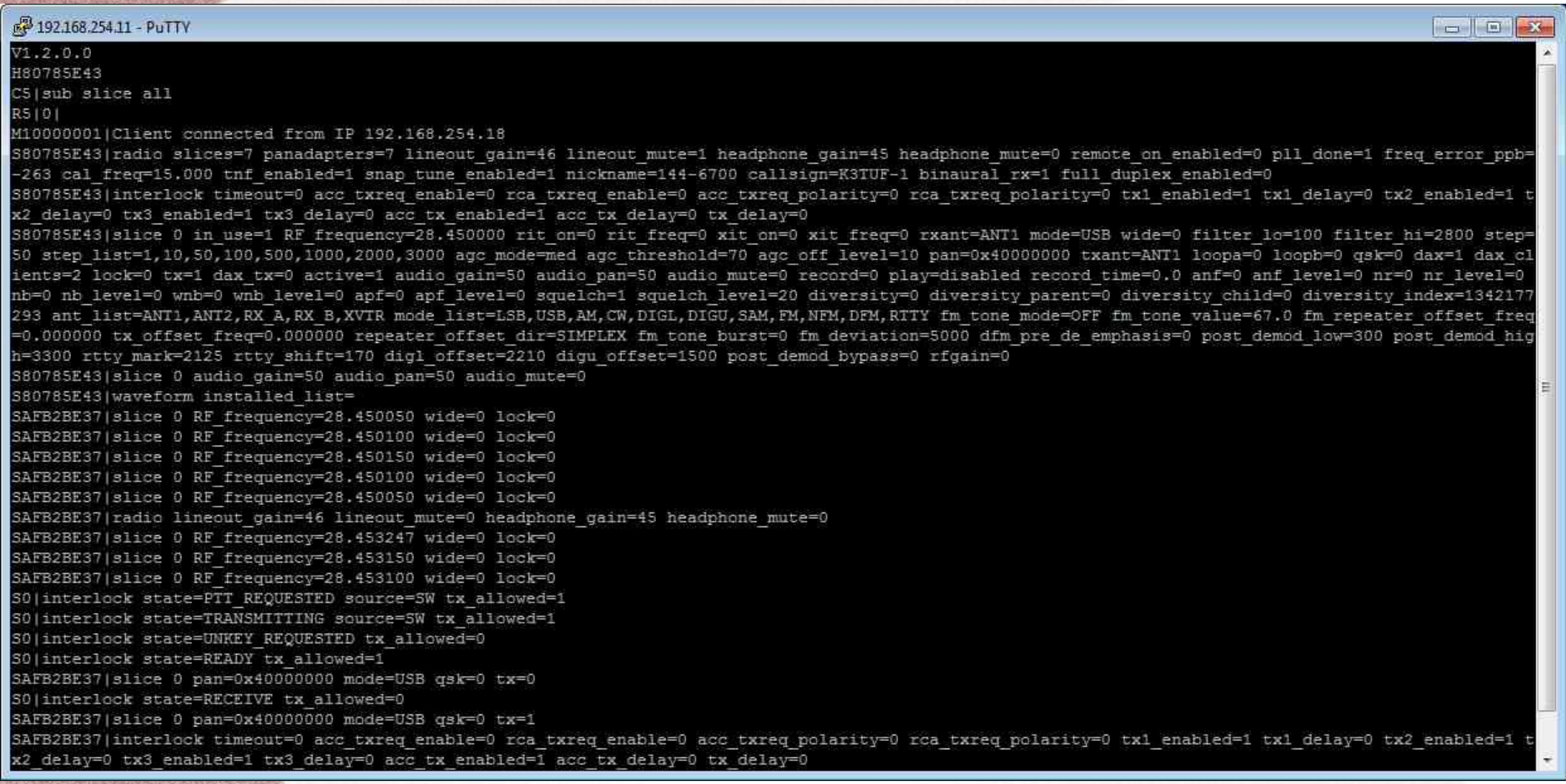
- Change slice receiver settings

```
slice set <slice> [<feature>=<value>]
C71|slice set 0 diversity=1 tx=0 record=1
R71|0
```

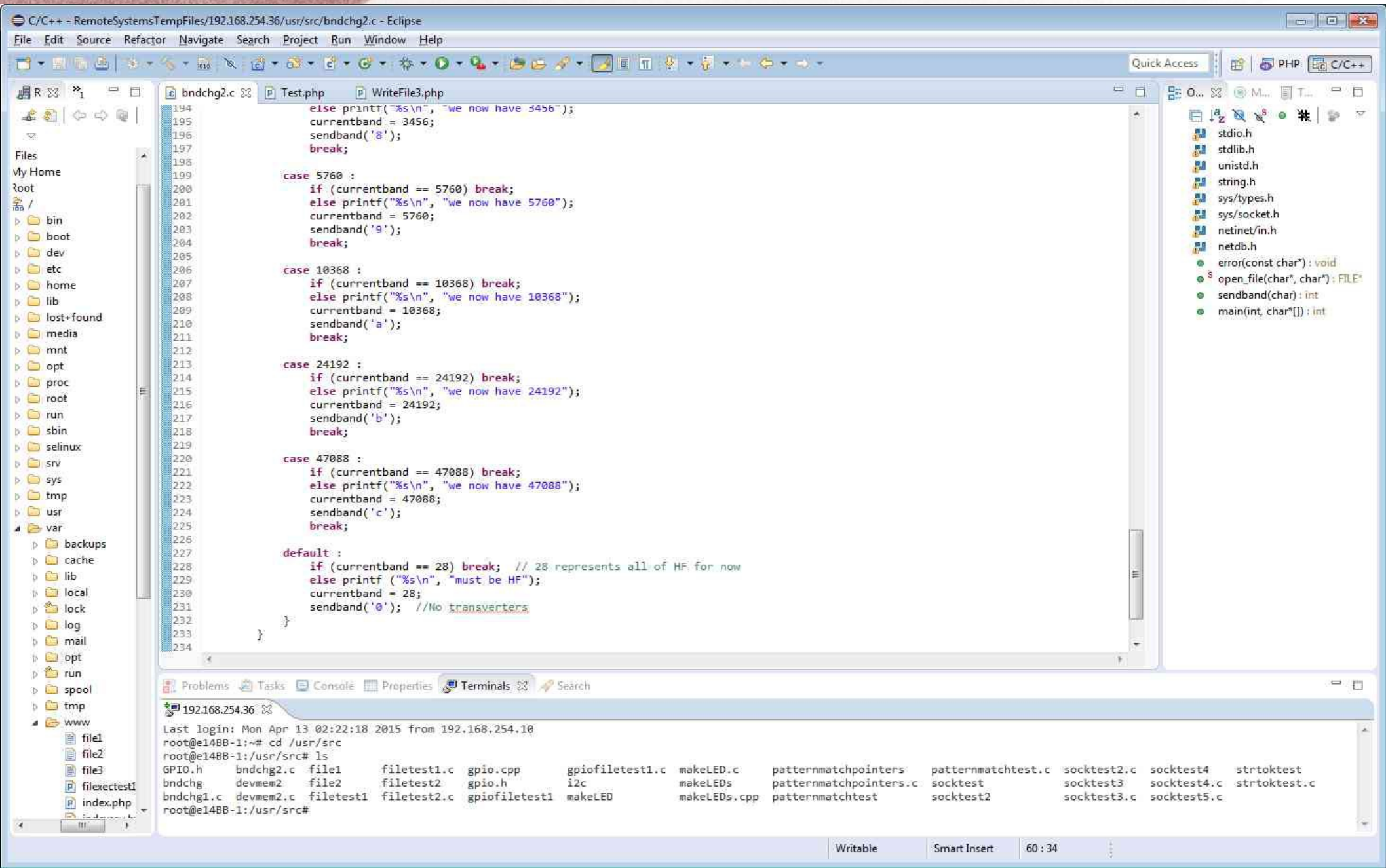
Sniffing TCP/IP A
Using Wiresh



The Magic of the API



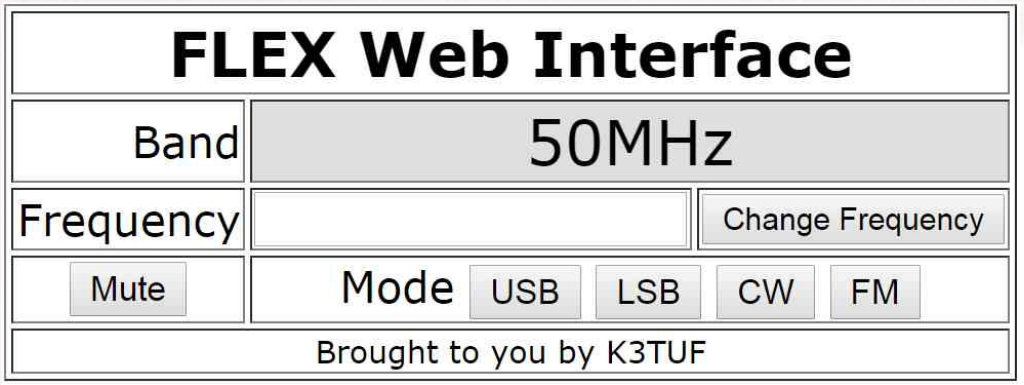
Eclipse Development Environment



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

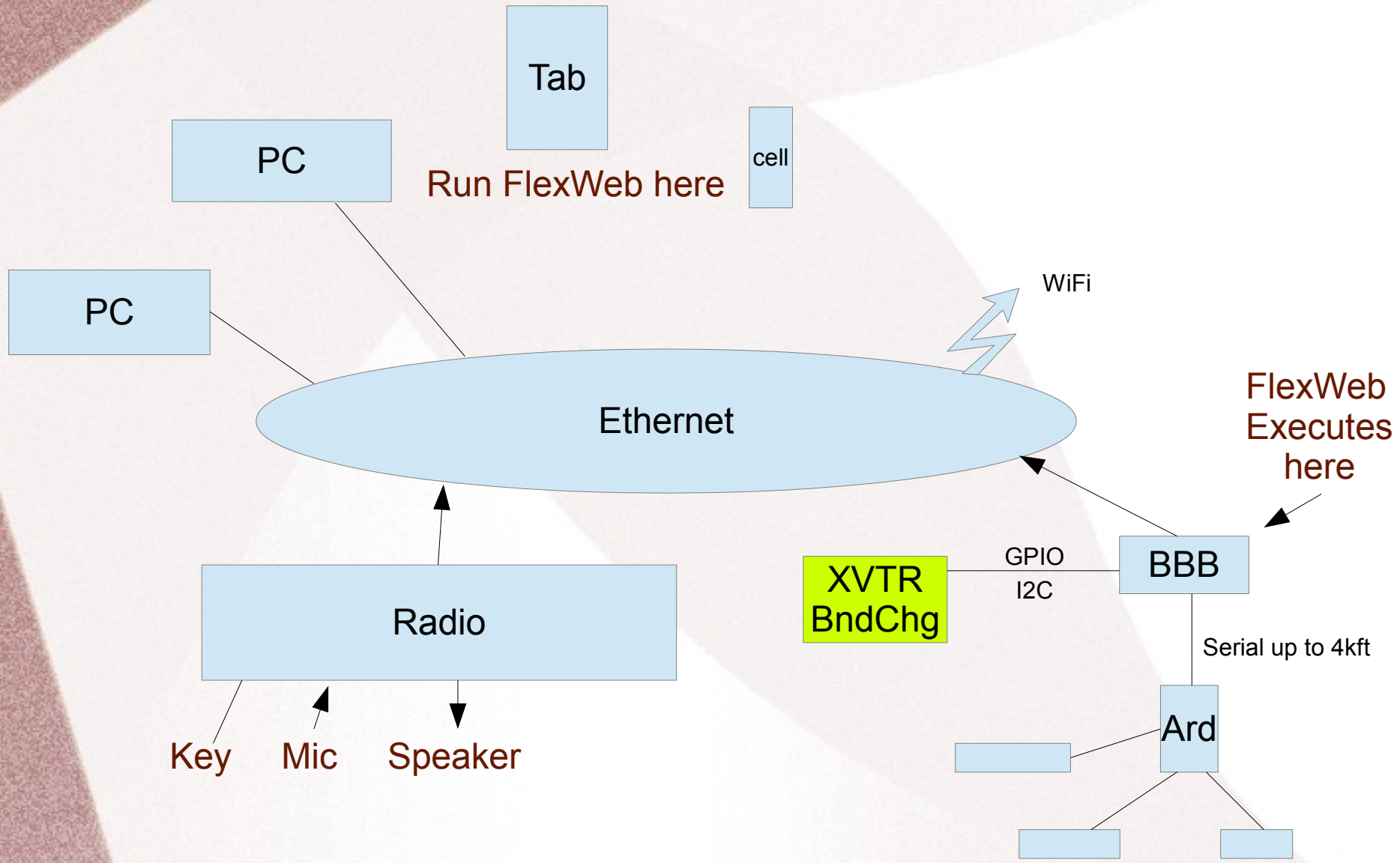
- HTML Hyper Text Markup Language
- AJAX Asynchronous JavaScript and XML
- DOM The Document Object Model is a platform and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents
- Apache / PHP is a server-side scripting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

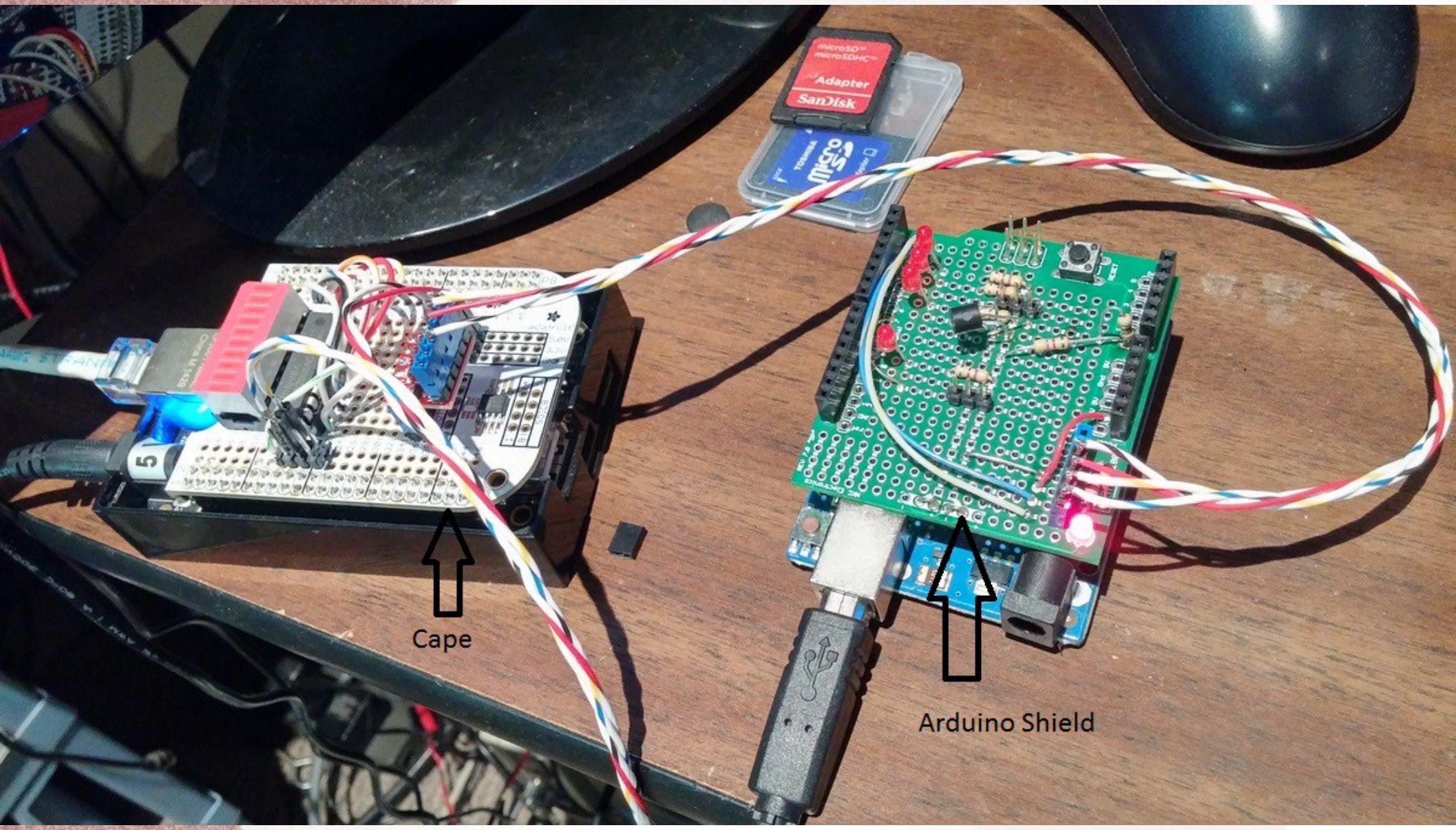
Technology: Languages cont'd

- 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

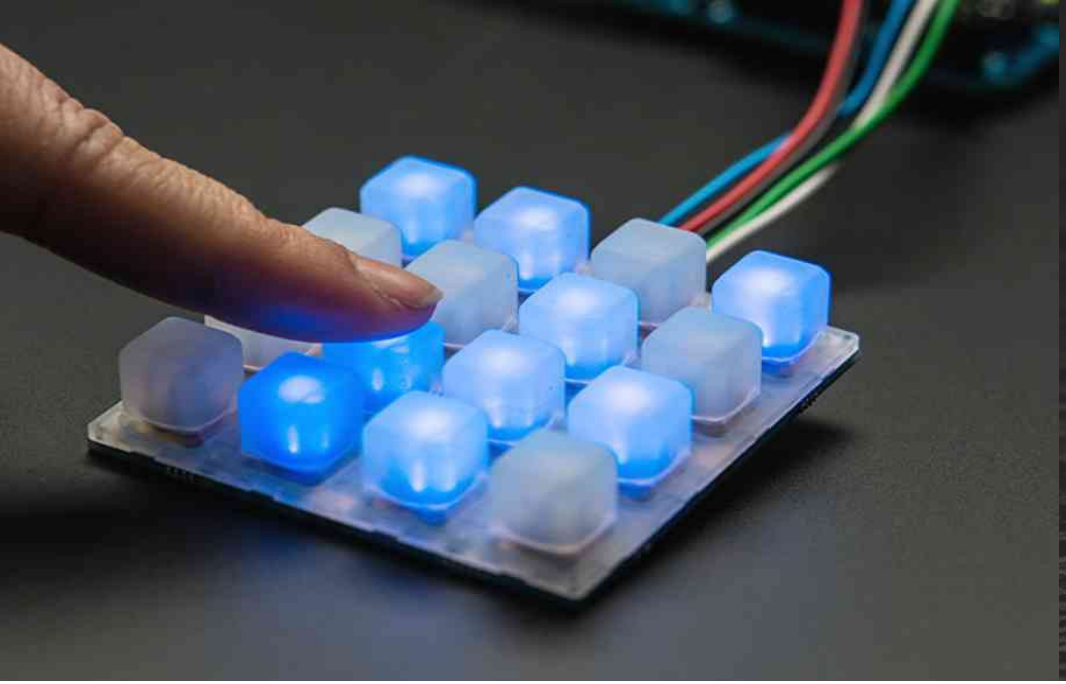
My Port 80 Plan

(Web Access)




Cape
Arduino Shield

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