Project Overview

• Introduction
  o Motivation
  o Cognitive Radio
  o TV White Space

• Dynamic Spectrum Access
  o Spectrum Sensing
  o Spectrum Analysis
  o Spectrum Decision

• Implementation
  o Hardware
  o Software
Project Overview

• **Introduction**
  - Motivation
  - Cognitive Radio
  - TV White Space

• **Dynamic Spectrum Access**
  - Spectrum Sensing
  - Spectrum Analysis
  - Spectrum Decision

• **Implementation**
  - Hardware
  - Software
Motivation
Motivation

• Low utilization
  o Caused by inefficient fixed frequency allocations instead of physical shortage of spectrum.

• Dynamic Spectrum Access (DSA):
  o To increase spectrum efficiency via spectrum sensing, probing and connectivity in cognitive radio networks.

• Cognitive Radio (CR) networks:
  o Primary User (PU) and Secondary User (SU)
  o PUs (licensed) have priority over SUs (Secondary) when accessing the wireless channel.
Cognitive Radio Networks

![Diagram of Cognitive Radio Networks](image)

- **Power** axis
- **Frequency** axis
- **Time** axis
- **Spectrum holes**
- **Dynamic Spectrum Access (DSA)**
- **Spectrum occupied by primary users**
Cognitive Radio Networks

Diagram:
- Radio Environment
- Transmitted Signal
- Spectrum Decision
- Spectrum Sensing
- Spectrum Holes Information
- Spectrum Analysis
- Channel Capacity
- RF Stimuli

The diagram illustrates the cycle of cognitive radio networks with the interaction between each component.
TV White Space (TVWS)

- TVWS, defined by FCC, means unused TV spectrum

**NTSC Channel Spectrum**
National Television System Committee (NTSC)
An analogy television system

**DTV-ATSC Spectrum**
Advanced Television System Committee (ATSC)
A digital television system
Spectrum Sensing Function Stack

- **Network**
  - Cooperative/Distributed Topology Management

- **Link**
  - Spectrum Sensing Policies
  - Interference Temperature

- **MAC**
  - Spectrum Sensing Management

- **Physical**
  - Sensing Algorithm
  - Wideband/narrow band sensing
  - Low SNR Detection
Project Overview

• Introduction
  o Motivation
  o Cognitive Radio
  o TV White Space

• Dynamic Spectrum Access
  o Spectrum Sensing
  o Spectrum Analysis
  o Spectrum Decision

• Implementation
  o Hardware
  o Software
Dynamic Spectrum Access (DSA)

- Step 1. Spectrum Sensing
- Step 2. Spectrum Analysis
- Step 3. Spectrum Decision
DSA – Spectrum Sensing

• The secondary users (SU) need to detect the presence of primary users (PU) in a licensed spectrum.
• If a PU emerges, the SU should quit ASAP in order to avoid interference to PUs.
DSA – Spectrum Sensing

• Energy Detection (ED):

  - The PSD of the signal is passed through a Band Pass Filter to select the channel.
  - Then integrated over time interval, i.e., the observation interval.
  - The output of the integrator is compared to a predefined threshold (H₀/H₁)
Comments on Energy Detection (ED)

• Advantages:
  o Simple
  o Is not required to know the primary user signal in advance.

• Disadvantages:
  o A pure energy detection scheme is confounded by the in-band interference because it is not robust against spread spectrum signals
  o Its performance severely suffers under fading conditions
DSA – Spectrum Analysis

• $S_0$: Scenario when only noise exists
  
  \[ S_0 : y(n) = w(n) \]

• $S_1$: Scenario when both noise and signal exist.
  
  \[ S_1 : y(n) = s(n) + w(n) \]

• The decision metric for the energy detector:

  \[ M = \sum_{n=0}^{N} |y(n)|^2 \]

  M is used in Spectrum decision step by comparing M with the threshold $\lambda_E$. 
DSA – Spectrum Decision

• Spectrum Decision is implemented by comparing $M$ with $\lambda_E$

• Probability of detection: $P_D = P_r(M > \lambda_E | S_1)$

• Probability of false alarm: $P_F = P_r(M > \lambda_E | S_0)$

So: Scenario when only noise exists
S1: Scenario when both noise and signal exist.
Project Overview

• Introduction
  o Motivation
  o Cognitive Radio
  o TV White Space

• Dynamic Spectrum Access
  o Spectrum Sensing
  o Spectrum Analysis
  o Spectrum Decision

• Implementation
  o Hardware
  o Software
Hardware – Raspberry Pi B 2

<table>
<thead>
<tr>
<th>Model</th>
<th>Raspberry Pi 2 Model B</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>900 MHz quad-core ARM Cortex-A7</td>
</tr>
<tr>
<td>Memory (SDRAM)</td>
<td>1 GB (Shared with GPU)</td>
</tr>
<tr>
<td>GPU</td>
<td>Broadcom Video Core IV @ 250 MHz</td>
</tr>
</tbody>
</table>
Hardware – 3.5” PiTFT Display

480 x 320 resolution; Touch screen; Use SPI and GPIO pins.
Hardware – RTL-SDR

RTL2832u

REALTEK

R820T Tuner: 24 - 1766 MHz;
Sample Rate: 2.56 MSamples/s.
Software

• Linux Modified Kernel
• FreqShow
• Scan Function
• Report
Software – Linux Modified Kernel

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Rasbian Wheezy (Debian 7.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>Linux 4.1.6</td>
</tr>
<tr>
<td>GCC</td>
<td>4.6.3</td>
</tr>
<tr>
<td>CMAKE</td>
<td>2.8.9</td>
</tr>
<tr>
<td>Python</td>
<td>2.7.3</td>
</tr>
</tbody>
</table>
Software – FreqShow

Waterfall Spectrum from 89.1 MHz to 91.5 MHz

Spectrum from 89.1 MHz to 91.5
Software – Add New Function: Scan
Scan Screen

- Previous Channel
- Next Channel
- Create Report
Scan Screen

Pause the data

Quit the program
Scan Screen
Scan Screen

High Peak Value

Low Peak Value

Energy
Scan Screen

Lower Frequency
Channel Number
Higher Frequency
## Report Formatted

<table>
<thead>
<tr>
<th>Channel Number</th>
<th>Energy in dB</th>
<th>Average Power in dB</th>
<th>Peak Value in dB</th>
<th>Frequency of peak in Mhz</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>13.68</td>
<td>6.94</td>
<td>19.2</td>
<td>470.95</td>
</tr>
<tr>
<td>15</td>
<td>14.27</td>
<td>8.48</td>
<td>20.74</td>
<td>476.68</td>
</tr>
<tr>
<td>16</td>
<td>12.33</td>
<td>6.63</td>
<td>19.19</td>
<td>482.94</td>
</tr>
<tr>
<td>17</td>
<td>14.24</td>
<td>8.4</td>
<td>20.1</td>
<td>488.95</td>
</tr>
<tr>
<td>18</td>
<td>16.23</td>
<td>10.17</td>
<td>22.53</td>
<td>494.63</td>
</tr>
<tr>
<td>19</td>
<td>16.44</td>
<td>9.88</td>
<td>21.49</td>
<td>500.47</td>
</tr>
</tbody>
</table>
Contribution

1. Learned:
   Dynamic Spectrum Control, Cognitive Radio, TV White Space, etc.

2. Configured:
   Raspberry Pi 2 with 3.5” TFT touch screen and RTL-SDR

3. Implemented:
   Energy Detection function in Python
Live Demonstration
Show the sensor working