RasHAWK

Distributed EM Situational Awareness Based on Raspberry Pi and REDHAWK

AOC Susquehanna Tech Challenge 2013
Final Report
March 7, 2014

Geon Technologies, LLC

www.geontech.com

MORGAN STATE UNIVERSITY
GATEWAY TO OPPORTUNITY • STAIRWAY TO EXCELLENCE

Contents

Figures

Tables

1 Introduction

The RasHAWK team has used a Raspberry Pi as the basis for a networked RF sensor capable of supporting spectrum monitoring, signal intercept and direction finding (DF) operations. All of these are fundamental capabilities of any ISR and EW system. For our AOC Tech Challenge demonstration, several RasHAWK sensors are deployed in a distributed sensor grid, wirelessly tethered to a command and control (C2) laptop. The system has the following key features and capabilities:

- A simple operator interface to configure the sensors

- Falling raster and PSD displays to monitor the spectrum for signal activity

- Demodulate FM signals from target FRS radios and play audio on selected channels

- Perform coarse DF on target emitters

- Display a map of the surrounding terrain that is annotated with the positions of the sensors, the target emitter and calculated lines of bearing (LOB) to the target. The map provides a RF Common Operating Picture (COP) with can be viewed on WiFi enabled tablets or smartphones.

This effort was made feasible by leveraging the existing capabilities of the open source REDHAWK software defined radio (SDR) framework. In addition to its inherent networked processing capabilities, REDHAWK provided a common software development framework that allowed a far-flung team to independently develop and test portions of the system software, then (relatively) easily integrate their contributions into the final system.

The motivation for this particular implementation was twofold: First, it grew out of a curiosity around the growing body of open source capabilities centered on the RTL USB radio dongle. Secondly, there is currently much interest in the REDHAWK SDR framework but unfortunately there is an absence of easily sharable applications that use REDHAWK in any significant way. A Raspberry Pi based sensor network using the RTL SDR dongle and REDHAWK seemed a good way to learn more about the RTL dongle while providing a meaningful reference design demonstrating REDHAWK capabilities for distributed processing.

This report presents an overview of the system architecture followed by a detailed description of all of the main sub-systems with particular emphasis on the hardware and software that comprise the Raspberry Pi sensor. Since a basic knowledge of the REDHAWK framework and terminology is necessary in order to understand how it is applied in the sensor and system, a brief REDHAWK overview is provided.

2 Team Composition

The RasHAWK sensor development was a joint effort between Geon Technologies, LLC and Morgan State University's Center of Excellence for Tactical and Advanced Communications Technologies (CETACT). A list of contributors including their affiliation and area of responsibility is provided in Table 1.

Table 1: List of Contributors

Individual	Affiliation	Area of Responsibility
Randeep Pannu	Graduate Student, Morgan State	DF antenna switch analysis, design, fabrication and construction
Antonio Samuel	Student, Morgan State	Antenna switching software
Dr. Willie Thompson	Research Professor, Morgan State	Technical guidance on the DF subsystem
Thomas Goodman	Geon Technologies, LLC	Software development on Raspberry Pi
Patrick Wolfram	Geon Technologies, LLC	Port of REDHAWK framework onto the Raspberry Pi ARM processor
Brandon Luquette	Geon Technologies, LLC	DF software component development
Chris Hinkey	Geon Technologies, LLC	RasHAWK User Interface
Joe Reinhart	Geon Technologies, LLC	Mapping software
Jim McCloskey	Geon Technologies, LLC	System architecture, integration & testing

3 System Description

Since the RasHAWK system architecture makes extensive use of the REDHAWK Software Defined Radio (SDR) framework, a brief description of the REDHAWK framework is provided. Much more information can be found at REDHAWK SDR FOSS website (http://redhawksdr.org).

3.1 REDHAWK in 30 Seconds (maybe more)

REDHAWK is a free and open-source software (FOSS) package that provides a robust architecture for distributed computing. Some key REDHAWK definitions and concepts are provided below:

Domain – A Domain can be thought of as the central controller in a REDHAWK based SDR system. The Domain is responsible for launching and controlling radio applications (called waveforms) along with maintaining a registry (called the Naming Service) of all resources that are available in the system. In the RasHAWK system, our Command and Control (C2) laptop serves as the Domain Manager, orchestrating all of the operations conducted by the system. As we'll see later, by remotely accessing the Domain Manager, we can navigate the entire system architecture and examine and control any device or software component.

Device – A Device is a software abstraction (or proxy) for a piece of hardware in the system that allows us to get status, control and data from that hardware. For example, a receiver Device will have software configurable control variables (called Properties) such

as center frequency, bandwidth, gain, etc. that an application can set and get. Setting these software properties in the Device results in the corresponding parameters being changed on the hardware. In the RasHAWK system, we wrote several Devices which reside on the Raspberry Pi to manage and control the hardware devices (GPS, RTL Receiver, Antenna Controller) connected to the Pi's USB and GPIO ports.

Node – A Node is a collection of Devices under the control of a Device Manager. The Device Manager runs on the node's host processor and starts each Device proxy in the node. The Device Manager is also responsible for contacting the system's Domain Manager and registering the nodes Devices with the Naming Service. In the RasHAWK system, each Raspberry Pi has its own Device Manager that contacts the C2 laptop and joins the system Domain on power-up.

Components – From the REDHAWK User's Manual – "A Component is a modular software building block that can be inserted into any number of signal processing applications to perform a specific and reusable function. A Component is fully defined by its interfaces, Properties, and functionality." Many of the software Components used in the RasHAWK system came with the standard installation of the REDHAWK framework. Others, such as the component that calculates Line of Bearing (LOB) were developed specifically for RasHAWK.

Waveforms – Waveforms (also referred to as applications) are sets of interconnected Components that are launched together to perform some specific signal processing function. In the RasHAWK system, all of the signal processing waveforms are launched on the C2 laptop's host processor. These waveforms connect to Devices in the remote Nodes in order to ingest data from the Node's receivers.

CORBA (Common Object Request Broker Architecture) – CORBA is the middleware in REDHAWK that enables the interconnection of Components and Devices seamlessly across a network. A programmer does not need to know specifically where a Component or Device resides on the network - CORBA and the Naming Service seamlessly manage this location information, greatly simplifying programming on a distributed system. In the RasHAWK system, the C2 Laptop and each sensor Node launch their own a CORBA Object Request Broker (ORB) to manage data transfers within the system.

Events – Events are an asynchronous mechanism used to disseminate status information throughout a REDHAWK system. Components and devices can subscribe to event topics of interest so that they are aware of a significant change in the system, such as a parameter update in another component. In the RasHAWK Sensor, we use events to signal changes in the antenna switching mode between the antenna control device and the RTL receiver device.

3.2 System Architecture

An overview of the RasHAWK system architecture is shown in Figure 1. We'll introduce each major component of the system here. More detailed descriptions the hardware and software for each subsystem will be presented in subsequent sections.

Figure 1 RasHAWK System Architecture

RasHAWK Sensor – The bulk of the work on this project involved development of the RasHAWK Sensors. The REDHAWK core framework was ported to the ARM processor on the Raspberry Pi and several REDHAWK Devices were written to remotely control the hardware connected to the Pi. Each sensor becomes a REDHAWK node that registers its capabilities with the Domain Manager on the C2 Laptop and relays position information and complex baseband data samples to the laptop for processing. A four antenna array and RF switch was constructed to perform coarse LOB calculations to target emitters.

Command & Control (C2) Laptop – The C2 Laptop is the main controller for the RasHAWK system. It runs the REDHAWK Domain Manager and keeps track of all the sensors that are currently online via the Naming Service. The waveforms which perform the signal processing for each sensor feed are launched from the REDHAWK Integrated Development Environment (IDE) along with spectrum and audio plots. A custom User Interface was developed that permits an operator to control a minimal set of key system parameters. The laptop also has a USB GPS dongle to provide its position to the mapping web server (also running on the laptop). Finally, an iPhone is used as a WiFi hotspot to tether the laptop to the Internet for real-time access to the online Google Maps server.

Location Mapping Server – A web server application is implemented on the laptop that can connect to the running REDHAWK Domain and probe all of the running devices and components for status. The mapping server polls the active sensors and laptop for their locations and plots them on a terrain map using calls to the Google Maps API. The server also polls the status of the LOB components running on the laptop and overlays active LOBs on the map. The map can be displayed in a web browser on any device connected to the RasHAWK WiFi network.

WiFi Router – RasHAWK uses a TP-LINK TL-WR1043ND 802.11N wireless router to establish an infrastructure WLAN for the sensor Nodes and Domain to remotely communicate. This model router was chosen primarily because its three antenna configuration looked impressive.

Transmitter Node – A late addition to the RasHAWK system is a Transmitter Node that allowed us to remotely key-up a slightly modified FRS Radio. The transmitter node is essentially a Raspberry Pi sensor node with an added transmit enable device that closes a relay to effectively press the radio's transmit key. An old MP3 played is used as an audio source so we didn't have to constantly say "Testing, testing, 1,2,3…". The position of the transmitter as determined by the GPS dongle is relayed to the laptop and plotted on the terrain map.

4 Subsystem Descriptions

This section describes each of the main RasHAWK subsystems in detail with particular emphasis on the use of the Raspberry Pi within the system.

4.1 RasHAWK Sensor

The centerpiece of the RasHAWK system is the Raspberry Pi based RF sensor shown in Figure 2. The sensor electronics are mounted on a plywood panel connected to a PVC pipe shaft that holds the four antenna DF array. The entire assembly is on a tripod and powered by a 12V 5.0Ah rechargeable battery. Alternately, the system can be powered from a 12V solar panel or car battery. The nominal current draw of the sensor is 330 mA at 12V. A block diagram of the sensor showing the interconnection of major components is shown in Figure 3.
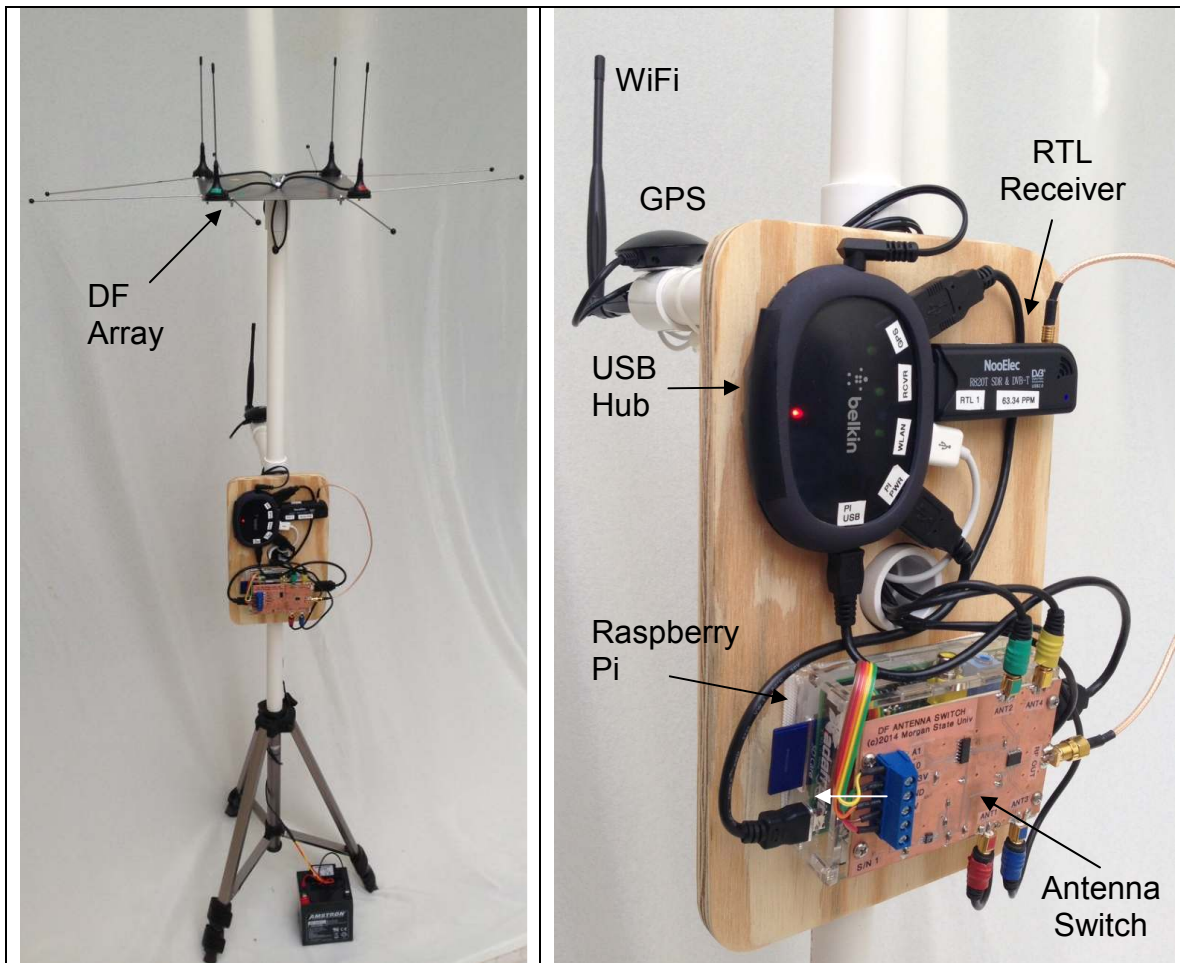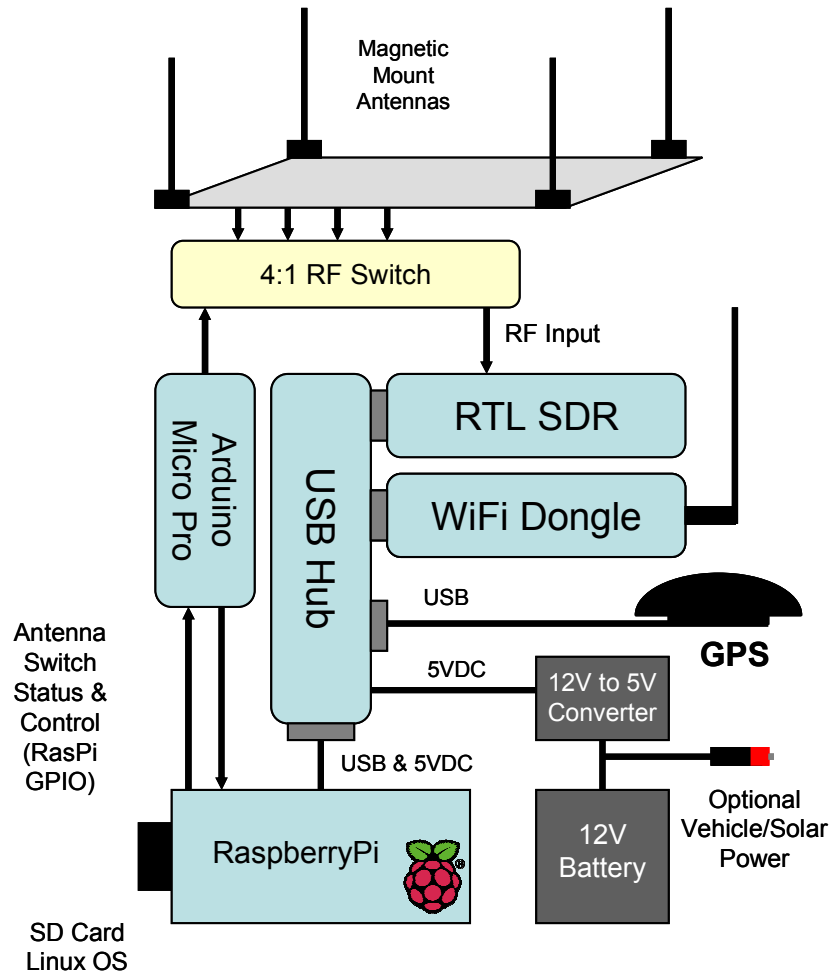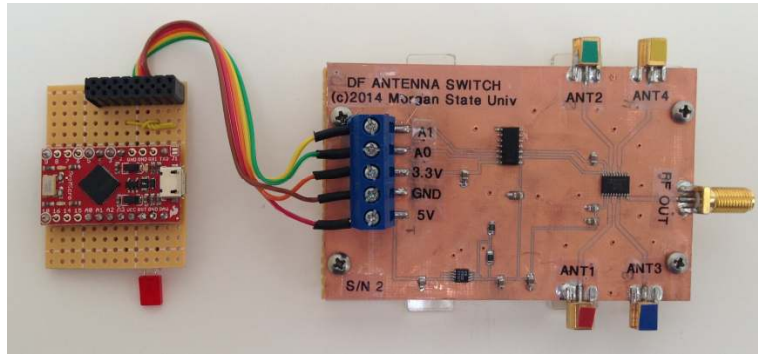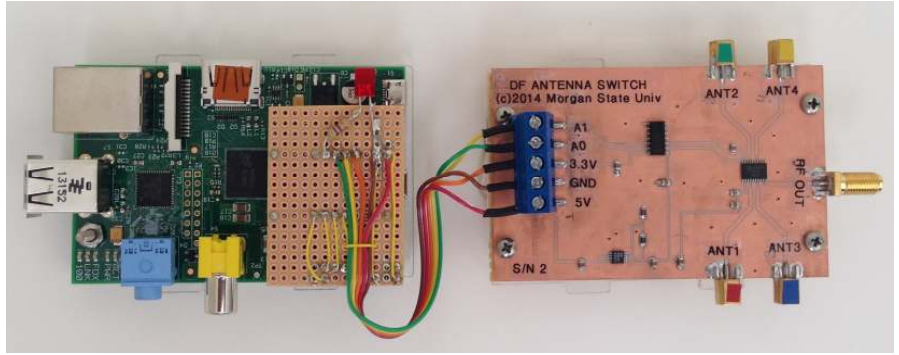

Figure 2 RasHAWK Sensor

Figure 3 RasHAWK Sensor Block Diagram

The USB GPS receiver used is a GlobalSat Model BU353-S4 with magnet base mount. The USB WiFi dongle used is a no-brand Model SL-1504N 802.11N ordered on eBay. It was selected mainly for its low price and impressive looking antenna. A 12VDC to 5 VDC voltage converter allows the sensor to be conveniently run from a car battery if needed. Each subsystem of the sensor is described in the following sections.

4.1.1 Rapsberry Pi

4.1.1.1 Hardware Configuration

The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC) which includes an single core ARM1176JZF-S 700 MHz processor and 512 MB of RAM. The Morgan State Antenna Switch is mounted on the outside of the unit's case. An 8 MHz Arduino Micro Pro microcontroller is mounted to the Raspberry Pi P1 connector to access the Pi's GPIO pins. The antenna switching is under the control of the Arduino which serves as an interface between the Pi and the antenna switch. This hardware configuration is shown in Figure 4. A block diagram of the configuration is shown in Figure 5.

Figure 4 Raspberry Pi Internal View (with Arduino)

Figure 5 Raspberry Pi Interconnections

The Arduino was included in the sensor design so that the antenna switching times could be more deterministic than directly using the Pi's GPIO lines under program control (due to the non-deterministic scheduling in the Pi's Linux OS). The Arduino is programmed to run "bare metal" with no operating system which produces near deterministic switching times. The configuration is a poor man's dual core processor with different operating environments on each processor.
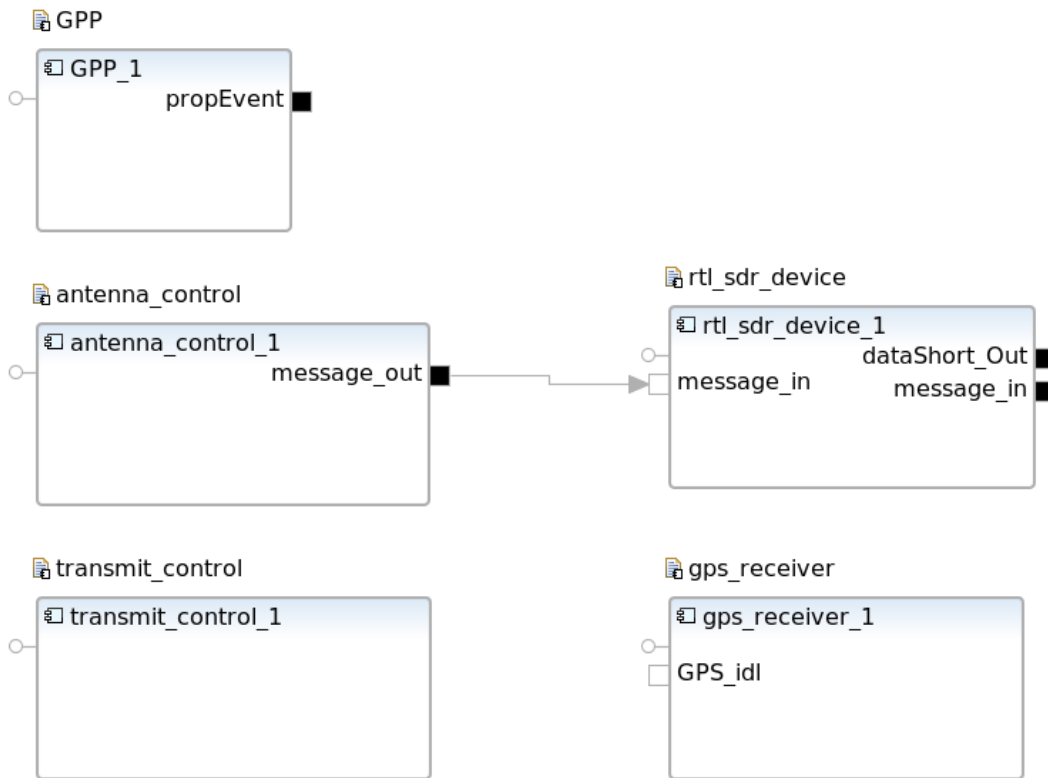
In operation, the Raspberry Pi sets the DF_MODE pin high to signal the Arduino to begin antenna switching. The red LED turns on when DF_MODE is high. The Arduino then controls the A0 and A1 port select lines to the switch to achieve the desired switching pattern. As it switches between antennas, the Arduino outputs the switching status to the Pi on the ANT_SEL0-3 lines. The Pi polls these pins (in the antenna_control Device) and the switching status is included in the SRI that is sent along with the baseband data from the sensor to the C2 Laptop.

The antenna array is made up of four magnetic mount antennas that come with the RTL radios mounted on a steel plate (extra antennas were ordered separately for this effort). The antennas are separated by approximately ¼ wavelength of the anticipated target signal frequency. Since our target signal for this demonstration is the unlicensed FRS radios, a nominal frequency of 462 MHz was used to calculate the ¼ wavelength spacing (approximately 6.4"). Antenna ground radials were constructed from lawn marker flags like the ones that were marking the locations of my neighbor's sprinkler heads until recently ;-). This hopefully provides a better ground plane for the antenna array than the metal plate alone. More detail on the DF technique used will be presented in a later section.

4.1.1.2 Software Configuration

For this effort, the Raspberry Pi based sensor runs the *Raspbian* distribution of the Linux Operating System (OS). The REDHAWK core framework was ported to this Linux distribution so that the Pi could act as a REDHAWK node and run the REDHAWK Device Manager.

Four new REDHAWK Devices were written to support the hardware in the RasHAWK sensor: rtl_sdr_device, gps_receiver, antenna_control and transmit_control. A new raspberry_pi node configuration that includes these devices was also created. The REDHAWK IDE graphical view of the raspberry_pi node is shown in Figure 6.

Figure 6 raspberry_pi Node and its Devices

Each new device developed for the RasHAWK Sensor is described below:

rtl_sdr_device Device - The rtl_sdr_device is a REDHAWK wrapper for the freely available librtlsdr library. The RTL Device's properties provide RasHAWK with direct access to altering the center frequency, sampling rate, tuner gains, etc. associated with this inexpensive SDR receiver. The Device provides a property event port for indicating when its hardware settings change and a messaging port to receive the switching pattern from the Antenna Control Device. Finally, it has a output data port for sending the received baseband data and metadata (called SRI – Signal Related Information) to the C2 laptop for processing.

antenna_control Device – The antenna_control Device leverages the open WiringPi library for directly manipulating the Raspberry Pi's GPIO port. This Device controls the Arduino board for relatively deterministic timing management of the antenna array switching. The Device provides properties for changing the antenna switching pattern and an Event port for signaling a change in the switching pattern to the RTL SDR Device.

gps_receiver Device - The gps_receiver Device provides serial bus access to the BU353-S4 USB GPS Receiver. Using the libnmea library, this Device translates the incoming NMEA-formatted messages into a variety of fields including latitude, longitude, signal

valid, etc. These fields are then applied to a FRONTEND GPS port ensuring future compatibility with other REDHAWK entities.

transmit_control Device - The transmit_control Device also makes use of the WiringPi library for directly manipulating the Raspberry Pi's GPIO port. In this case, the Device drives two pins 1) enabling the transmitter output and 2) enabling the audio input on an FRS radio. The state of each pin is controllable by the Device's properties that are then easily accessible by our control applications. Note that this device is only used on the Node that controls the FRS radio target emitter.

4.1.2 RTL SDR USB Radio Dongle

The RTL SDR (Figure 7) is a re-purposed USB DVB-T (terrestrial digital video broadcast) dongle that can be used as inexpensive radio receiver to stream downconverted and digitized I/Q baseband samples to a host computer for further processing. The RTL-SDR dongle used in RasHAWK sensors is capable of tuning over a 24 - 1766 MHz range with just over a 2 MHz bandwidth. Additional information on the RTL-SDR can be found at http://sdr.osmocom.org/trac/wiki/rtl-sdr.

Figure 7 RTL SDR USB Radio Dongle

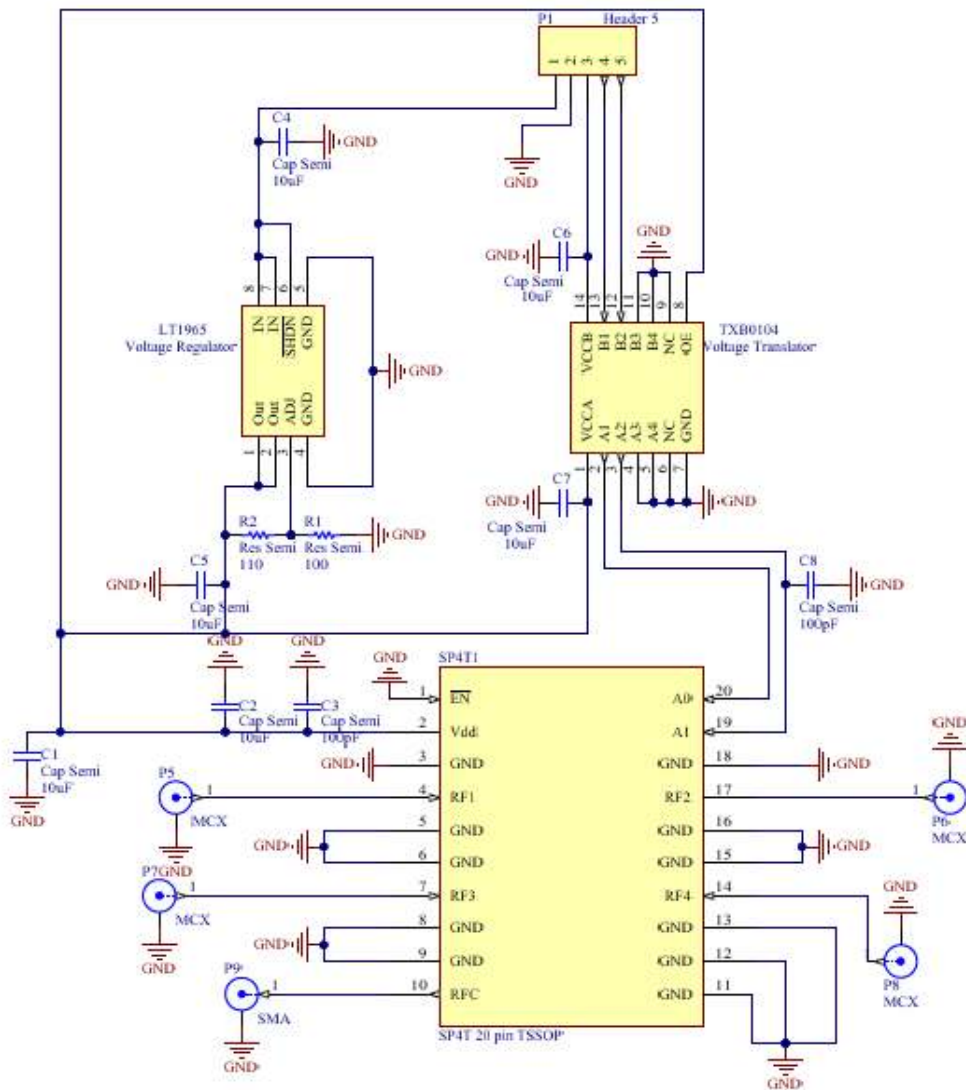## 4.1.3 Antenna Switching Subsystem

The Antenna Switching Subsystem (Figure 8) was designed, fabricated and assembled by team members at Morgan State University. The board was fabricated using a PCB milling machine in the University's RF design lab. The schematic for the design is shown in **Figure 9**.

Figure 8 Antenna Switch for Pseudo-Doppler DF

Figure 9 Antenna Switch Schematic

The antenna switch uses an Analog Devices ADG904 4:1 RF mux to switch between antennas under Arduino control. One design challenge was properly conditioning the supply and signaling voltages from the Arduino to be compatible with the 1.65 to 2.75V operating voltage range of the ADG904. This was done using a Linear Technologies LT1965 voltage regulator to convert the 5 V Arduino supply voltage to the nominal 2.5V required by the switch. A Texas Instruments TXB0104 level translator was used to convert the 3.3V Arduino control signals to 2.5V. RF characterization measurements conducted at the Morgan State RF Lab for the completed switch board are shown in Table 2.

Table 2 Characterization of the Antenna Switch PCB

RF Ports	Insertionn Loss (dB)	Off Isolation (dB)	Return Loss (dB)	Port to Port Isolation (dB)	Conditions
RF1	0.53	44.69	20.677	RF1-RF2= 40.05 RF1-RF3= 45.41 RF1-RF4= 42.5	VDD=2.5V, Fs= 462 MHz
RF2	0.53	44.08	20.741	RF2-RF1= 44.95 RF2-RF3= 46.11 RF2-RF4= 36.88	VDD=2.5V, Fs=462 MHz
RF3	0.44	44.98	20.689	RF3-RF1= 39.52 RF3-RF2= 45.98 RF3-RF4= 43.21	VDD=2.5V, Fs=462 MHz
RF4	0.41	43.9	20.958	RF4-RF1= 45.24 RF4-RF2= 39.76 RF4-RF3= 45.68	VDD=2.5V, Fs=462 MHz

A bill of materials for the antenna switch is provided in Table 3. The use of the Antenna Switch to perform the Pseudo Doppler DF technique will be discussed in Section 5.2.

Table 3 Antenna Switch Bill of Material

V3 Switch Board Part List (Per Board)						
Component	Manufacturer	Vendor	Description	Quantity	Unit ($)	Extended ($)
ADG904	Analog Devices	Digikey	SP4T Switch	1	3.78	3.78
TXB0104	Texas Instruments	Digikey	Voltage Level Translator	1	1.69	1.69
LT1965IMS8E	Linear Technology	Digikey	Linear Volatge Regulator	1	3.19	3.19
R04003C Dimension:12"x9" @ $9.90	Rogers Corporation	Rogers Corp	Board Material (DK=3.38+0.05, Diel Thickness=16.7 mils,TC Copper=0.5oz/sqft)	1	2.00	2.00
PCB Terminal Block	Philmore	Baynesville	Power Supply	1	1.20	1.20
MMCX SMT Conn	?	eBay	SMT micro coax connector	4	0.90	3.60
SMA SMT Conn	?	eBay	SMT SMA female connector	1	1.00	1.00
					TOTAL	16.46

5 Application Software Description

The following sections describe the processing application software on the C2 laptop.

5.1 Processing Waveform Description

After a sensor is powered on and registers with the laptop Domain's Naming Service, the processing waveform shown in Figure 10 is manually launched on the laptop from the REDHAWK IDE. The waveform ingests data from the sensor and performs the required signal processing functions to recover the audio and determine LOB to the transmitter.

Figure 10 RasHAWK Processing Waveform

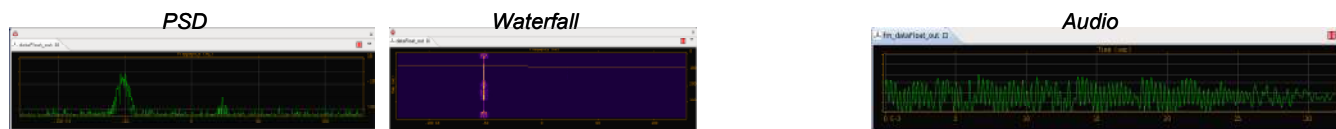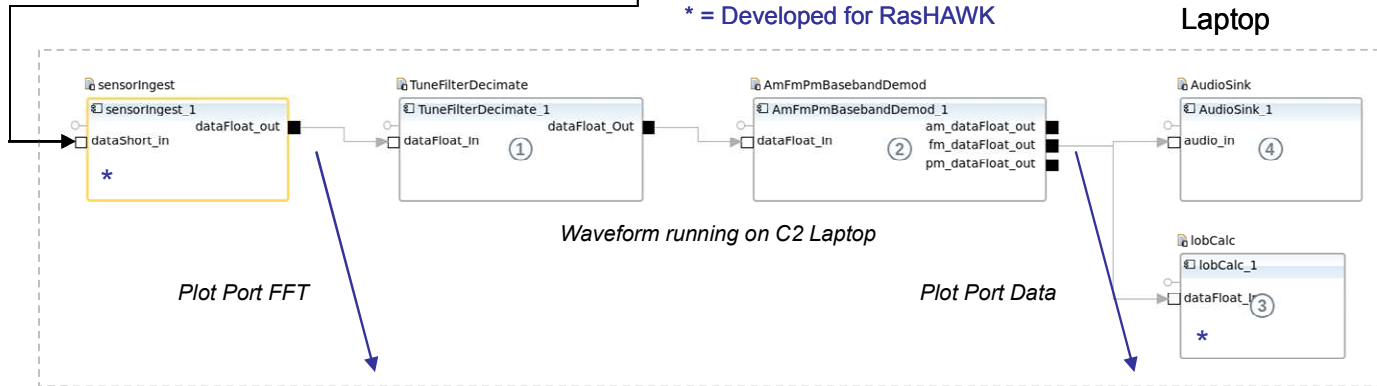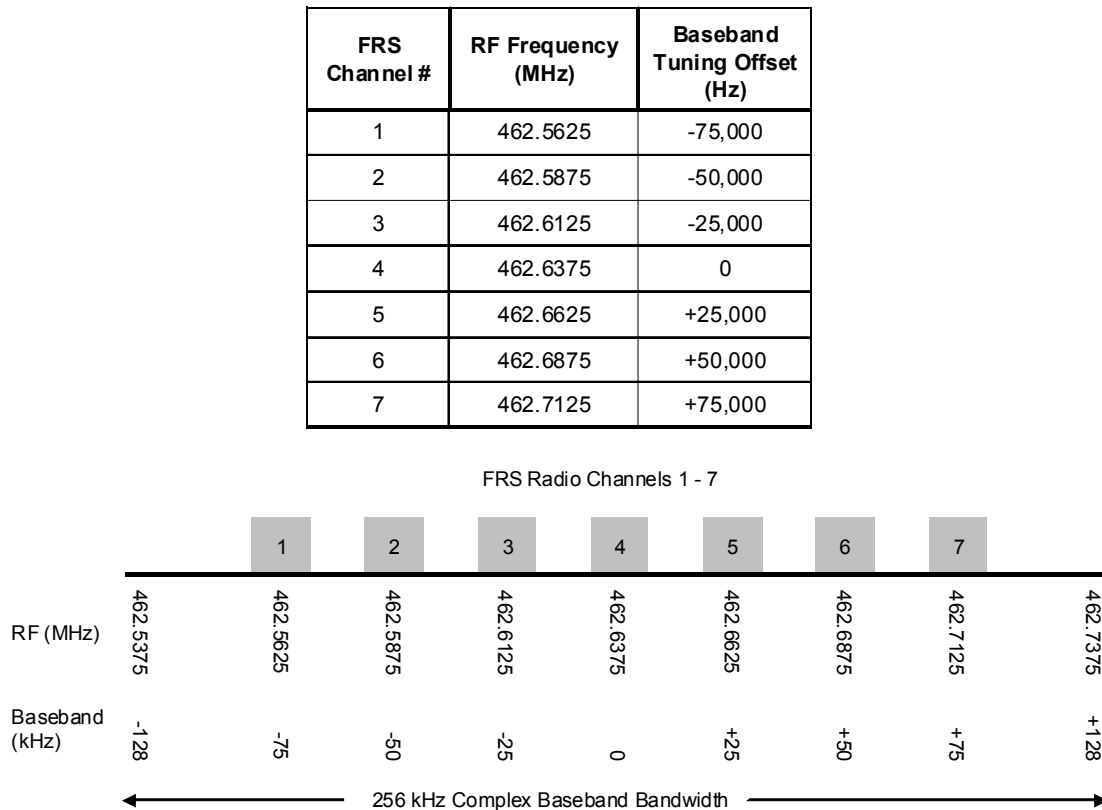The waveform automatically connects to an rtl_sdr_device using an allocation property in the waveform's Software Assembly Descriptor (SAD) XML file. A separate waveform is launched for each active sensor in the system. The component ports from which spectrum and audio plots are generated during the system demonstration are also shown in Figure 10. Each component in the application waveform is described below:

sensorIngest – Removes any residual DC bias from the RTL baseband data and converts the 16-bit signed integer (shorts) output from the RTL to floating point values compatible with the next input port in the chain.

TuneFilterDecimate – A stock REDHAWK component which does just that – performs a frequency translation of the complex input followed by an adjustable bandpass filter and finally decimates the oversampled filter output. For RasHAWK, we use the TuneIF mode which extracts the input sample rate from the SRI to determine the bandwidth of the input signal. The tuning limits in this mode are +/- sample_rate/2. For our demonstration we tune the RTL receiver to a center frequency of 462.6375 MHz (FRS Channel 4) with a sample rate of 256 kHz in order to provide sufficient bandwidth to cover FRS Channels 1 to 7. This gives us a baseband tuning bandwidth of +/- 128 kHz. The channel spacing for FRS radios is 25 kHz so the baseband tuning offset for FRS channels 1 to 7 is shown Figure 11.

FRS Channel #	RF Frequency (MHz)	Baseband Tuning Offset (Hz)
1	462.5625	-75,000
2	462.5875	-50,000
3	462.6125	-25,000
4	462.6375	0
5	462.6625	+25,000
6	462.6875	+50,000
7	462.7125	+75,000

FRS Radio Channels 1 - 7

	1	2	3	4	5	6	7	
RF (MHz)	462.5375	462.5625	462.5875	462.6125	462.6375	462.6625	462.6875	462.7125	462.7375
Baseband (kHz)	-128	-75	-50	-25	0	+25	+50	+75	+128

← 256 kHz Complex Baseband Bandwidth →

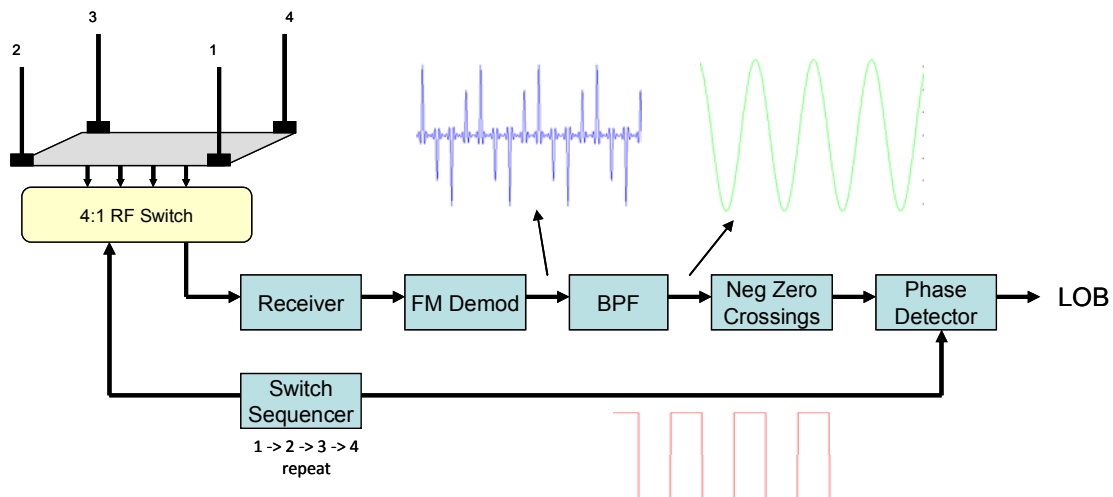**Figure 11 FRS Radio Channel Allocation and Baseband Tuning Offsets**

AmFmPmBasebandDemod – A stock REDHAWK component that ingests complex baseband data and outputs real demodulated samples. We only utilize the FM demodulation capability.

AudioSink – An existing component that ingests audio data samples from the FM demodulator output and plays the audio on the laptop soundcard. Downloaded from the Axios Engineering Git repository.

lobCalc – Developed specifically for the RasHAWK application, this component ingests FM demodulated data and performs the line of bearing calculation using a variation on the Pseudo Doppler DF technique. The next section describes this technique in detail.

5.2 DF Technique Description

From the beginning, the RasHAWK effort had the stretch goal of performing a line of bearing from each sensor to the target emitter for determining coarse emitter location. We use a modification of the Pseudo Doppler DF (PDDF) technique popular among ham radio "fox hunters" and used in some commercial DF system (such as LoJack Vehicle Recovery System). A notional PDDF system is shown in Figure 12.

Figure 12 Notional Pseudo Doppler DF System

In this case an RF switch is used to sequence between 4 antennas in a square array with ¼ wavelength separation (based on target signal frequency). The switching frequency for typical PDDF systems is in the 500Hz to 1 kHz range. For a planar wavefront arriving at an angle to the array, each antenna receives a slightly delayed or advanced version of the signal relative to the other antennas. Switching between antennas produces a phase discontinuity in the carrier sinusoid at the receiver input. The step discontinuity in phase produces an impulse out of the FM demodulator. These impulses are essentially sample points on a continuous sinusoid with a period equal to the time required to switch through

all 4 antennas. After some calibration, the phase difference between the sampled sinusoid and the antenna switch control signal can be mapped to the angle of arrival of the incident wavefront.

In a typical PDDF system, the antenna switching and receiver components are co-located with the FM demod and zero crossing detector making phase comparison of the switching signal and sampled sinusoid trivial. In the RasHAWK system, the receiver and antenna switcher are located in the remote sensor, away from the FM demod and any other processing components in the laptop making exact synchronization between the switcher and FM demod output difficult.

Our goal is to recreate the sampled sinusoid by measuring the magnitude of impulses out of the FM demod as the antennas switch. To accomplish that, we have the Arduino switch back and forth between a pair of antennas for a period of time, thus producing a sequence of impulses of the same amplitude. The rtl_sdr_device in the sensor reads the state of the Arduino switching and encodes the current antenna switching pair in the SRI that is sent along with the data to the laptop. The lobCalc component takes an average of the peaks it detects for each pair of antennas. The result after all antenna pairs have been visited are 4 sample values of one period of the pseudo-doppler sinusoid. Based on the relative amplitudes of these values, we can determine the coarse line of bearing of the incident signal.

5.3 Control User Interface

The RasHAWK Control User Interface (UI) provides a simple menu driven operator interface for basic control of sensors in the RasHAWK network. The UI is written using the pygtk python graphical user interface library and uses the REDHAWK python Application Programmer Interface (API) attach to a running REDHAWK Domain, discover which resources are currently online and control selected devices and components on demand.
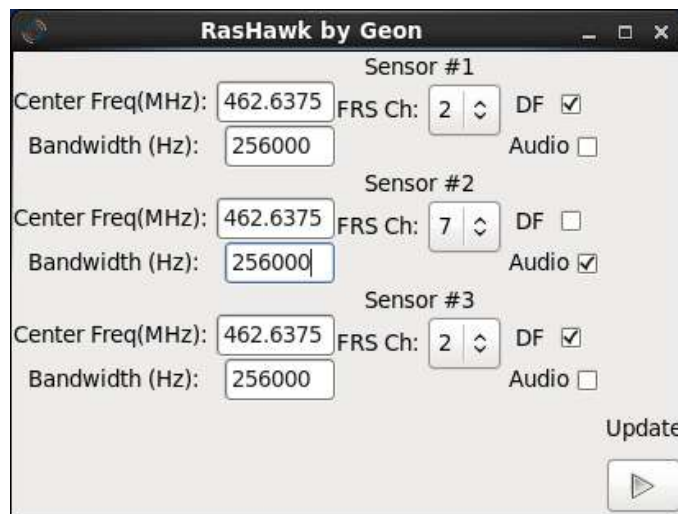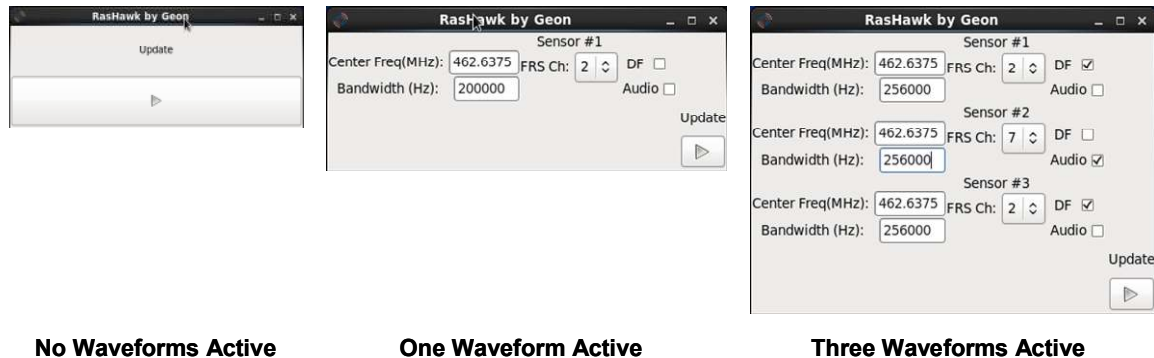

Figure 13 RasHAWK Control UI

When the UI is first launched it attaches to the running REDHAWK Domain and searches for active waveforms every 5 seconds. If no waveforms are active, then no control entries are displayed on the UI. As new waveforms are launched (by the REDHAWK IDE in our case), the control entries are displayed for the new waveform. This sequence is illustrated in Figure 14. As waveforms are released (destroyed), the corresponding entries are deleted from the UI.

No Waveforms Active   One Waveform Active   Three Waveforms Active

Figure 14 UI Display Updates as New Waveforms are Launched

Using UI, the operator has control of the following key processing parameters in the system:

- Center frequency and bandwidth of each sensor
- FRS radio channel currently being processed by a waveform
- Enabling FM demodulated audio output on the laptop soundcard
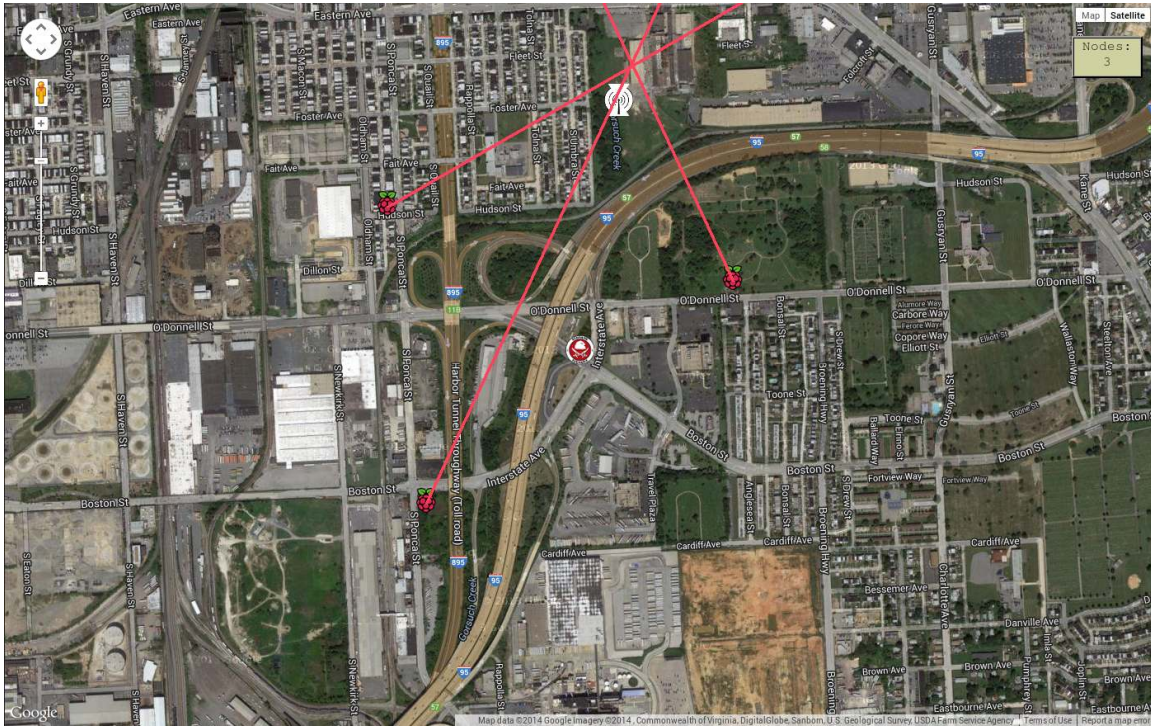- The DF function can be turned on/off on a per sensor basis

The user must click the Update button in order for any parameter changes to take effect.

5.4 Mapping Interface

The Mapping Interface utilizes standard web technologies and open source software to display sensor information in near real time without the need for additional plug-ins or proprietary software packages. Common HTML5 design practices are used to create an RF Common Operating Picture (COP) that will be supported by most modern browsers, including Firefox and Chrome, and on any platform supporting a browser, including laptops, tablet computers, and smartphones.

HTML, JavaScript, and CSS are used to create a webpage-based frontend UI that requires no plug-ins or proprietary software, only a modern browser that supports the Websockets protocols. The Google Maps and JQuery Javascript libraries are used to create a more dynamic user interface. The backend uses the open source Python Tornado framework to instantiate an asynchronous webserver on a host laptop.

This framework is based on the idea of asynchronous connections between the client and server, allowing for a persistent bi-directional connection between the server and Mapping Interface via Websockets. This allows Information updates to be sent in near real time from the server to the client browser. Nodes registering with the Domain, changes in position, and line-of-bearing (LOB) data are passed from the server to the Mapping Interface to be updated on the browser display. Figure 15 shows an example of the Mapping Interface with node locations and LOBs to the emitter.

Figure 15 Mapping Interface Example with Plotted LOBs

6 Application Scenarios

In the system demonstration, the goal is to demonstrate the following three use cases for the RasHAWK sensor network:

- Distributed Sensor Deployment
- Spectrum Situational Awareness and Monitoring
- Coarse Emitter Location

These scenarios will be explained briefly in the following sections.

6.1 Distributed Sensor Deployment

The RasHAWK sensors form a distributed sensor network connected to a central C2 Laptop node with wireless connectivity (via WiFi). On power-up, sensors connect via WiFi automatically and register with a REDHAWK C2 node on a laptop then begin forwarding signal and GPS coordinates. The sensor locations will be displayed on a satellite view map using Google Maps.

6.2 Spectrum Situational Awareness and Monitoring

Once the sensor network is established, an operator at the C2 laptop is able to monitor a sensor's spectrum and listen to target audio from a selected FRS channel using the Control UI described earlier. Spectrum plots and an audio waveform time domain display will be active for each sensor during the demonstration (see Figure 16).

6.3 Coarse Emitter Location

Once a signal is observed, the operator can command the sensors to perform a line of bearing (LOB) determination on the target. The LOB calculations will be performed on the C2 Laptop using data forwarded by the sensors. LOBs from each sensor will be displayed on the map for coarse emitter location determination via intersecting LOBs.
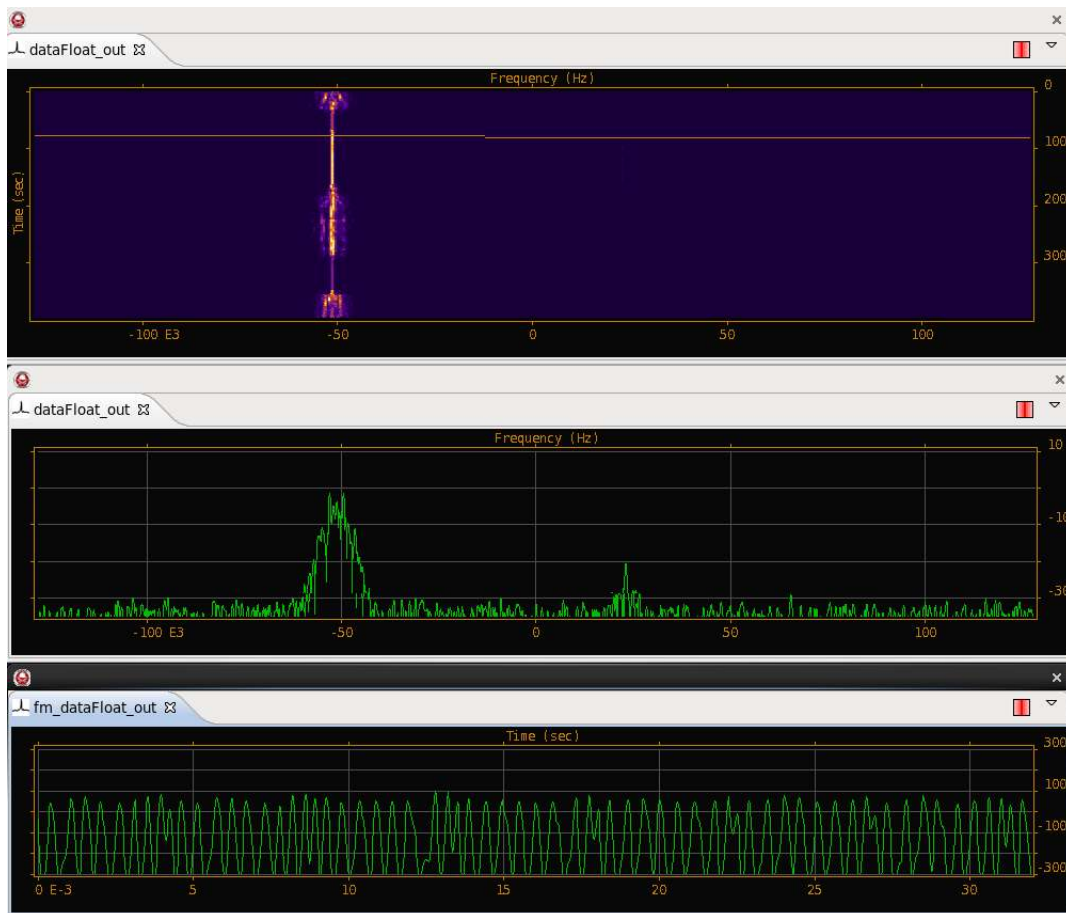

Figure 16 Spectrum and Audio Plots Displayed for Each Sensor

7 Demonstration Plan

Given that the Showcase demonstrations are indoors, a full demonstration of the system will not be feasible. Our submission to the judges will be a short presentation on the system followed by a brief video of an actual outdoor test of the system. The outdoor test will show the system performing the three application scenarios listed earlier. If possible, we would like to bring a sensor and laptop into the conference room if possible to show the judges. Note that both the laptop and sensor will have WiFi capability although we can disable that if required.

While the system is capable of monitoring frequencies from 24 to 1766 MHz, for practical reasons, demonstration and testing will use unlicensed Family Radio Service (FRS) radios as the RF target "rabbit". The rabbit emitters will be GPS-enabled networked REDHAWK nodes with the transmitters controlled from the C2 laptop. The system demonstration will use FRS radio channels 1 - 7 covering UHF frequencies from 462.5625 to 462.7125 MHz. A grid of three RF sensors will be deployed over an area of approximately 1/2 square mile.

8 Known Limitations of the System

The RasHAWK system is intended to demonstrate the utility of inexpensive software and components to perform operations typically requiring much more expensive equipment and software. That said, RasHAWK does not pretend to be a fully capable ISR system – it is a demonstration targeted at one particular class of radio transmitter in one frequency band. The REDHAWK framework underpinnings of the system are sound and given the correct equipment and software components, could fulfill the role of a fully capable distributed ISR or JEMSO support system.

9 Sensor Bill of Material

The final Bill of Material with pricing for the RasHAWK Sensor is shown in Table 4.

Table 4 RasHAWK Sensor Final Bill of Material

Item #	Description	Part #	Manufacturer	Source	Cost
1	USB Radio - DVB-T USB Stick (R280T)	TV28T-V2	UNK	NooElec.com	$17.95
2	Hi-Speed USB 2.0 4-Port Powered Hub		Belkin	OfficeMax	$27.99
3	Raspberry Pi Model B 512MB Ram			adafruit.com	$39.95
4	USB GPS Nav Receiver - magnetic mount	BU-353	GlobalSat	amazon.com	$28.87
5	802.11g/n USM WiFi Adapter, 5dBi antenna	SL-1504N	UNK	amazon.com	$14.69
6	4GB SD Card for RasPi			adafruit.com	$7.95
7	12V to 5V DC-DC Converter	UNK	Chuangruifa	ebay	$4.69
8	12V 5AH Sealed Led Acid Battery	AP-1250 F1	Amstron	atBatt.com	$13.99
9	Extra Mag Mount Antennas ($1.90 ea, 4 total)		UNK	ebay	$7.60
10	Misc Hardware (PVC pipe, metal plate, etc)			Lowes	$8.00
11	Misc USB Cables			Dollar Tree	$4.00
12	4:1 RF Switch Board		Morgan State	various	$16.46

GRAND TOTAL $192.14